



## Durham E-Theses

---

### *An investigation into the use of on-line computing in the solution of business problem*

Shergold, J. R.

#### How to cite:

---

Shergold, J. R. (1972) *An investigation into the use of on-line computing in the solution of business problem*, Durham theses, Durham University. Available at Durham E-Theses Online: <http://etheses.dur.ac.uk/9850/>

#### Use policy

---

The full-text may be used and/or reproduced, and given to third parties in any format or medium, without prior permission or charge, for personal research or study, educational, or not-for-profit purposes provided that:

- a full bibliographic reference is made to the original source
- a [link](#) is made to the metadata record in Durham E-Theses
- the full-text is not changed in any way

The full-text must not be sold in any format or medium without the formal permission of the copyright holders.

Please consult the [full Durham E-Theses policy](#) for further details.

AN INVESTIGATION INTO THE USE OF  
ON-LINE COMPUTING IN THE  
SOLUTION OF BUSINESS PROBLEMS

BY

J.R. SHERGOLD, B.Sc.

A thesis submitted for the Degree of Master  
of Arts in the University of Durham.

APRIL 1972

The copyright of this thesis rests with the author.  
No quotation from it should be published without  
his prior written consent and information derived  
from it should be acknowledged.



## ABSTRACT

The computer, with its power to perform lengthy calculations quickly and its ability to handle large amounts of information efficiently, is an obvious aid in the solution of business problems. With these abilities, the computer can free the manager from the tedious tasks of searching and calculating. Thus the computer-manager partnership is an excellent one in the business field.

The rapid development of time-sharing within the last few years, however, has greatly enhanced this partnership. The manager and computer can interact in a way never before possible. This research intends to investigate some of the problems involved and the most efficient way in which this interaction can take place.

## TABLE OF CONTENTS

	<u>Page</u>
Table of Contents	i
Acknowledgements	xi
1. Introduction	1
1.0 Historical background	1
1.1 Structure	1
1.2 What is on-line computing?	2
1.3 Some features of present-day systems	4
1.4 The NUMAC system	11
1.5 Summary	14
2. The Uses of On-line Computing in Business	15
2.0 Introduction	15
2.1 Why on-line computing?	15
2.2 Classification	17
2.3 Conclusion	34
3. Development of Research Areas	36
<u>SECTION 1</u> Research into Applications of Type 1	38
4. Conversational Packages - The Problems	39
4.0 Introduction	39
4.1 Response time	39
4.2 Conversational level	43
4.3 Errors	44
4.4 Input formats	45
4.5 Explanation techniques	47
4.6 Conclusion	47

	<u>Page</u>
5. Experiment on the Design of Conversational Packages for Managers	49
5.0 Introduction	49
5.1 Design of packages	49
5.2 The subjects	51
5.3 The problems	52
5.4 The facilities	56
5.5 Experimental procedure	57
5.6 Performance measures	58
5.7 Results	60
5.8 Conclusion	71
<u>SECTION 2</u> Research into Applications of Type 2	72
6. The Problems of Larger Scale Applications (type 2)	74
6.0 Introduction	74
6.1 Management involvement	74
6.2 Organisational problems	75
6.3 The human element	76
6.4 The development policy	77
6.5 Costs	78
6.6 Fear of the machine	80
6.7 Staff problems	80
6.8 Hardware problems	81
6.9 Lack of 'know-how'	82
6.10 Summary	82
7. A Survey of On-Line Computing - The Applications	84
7.0 Introduction	84
7.1 Procedure	84
7.2 Facilities	85

	<u>Page</u>
7.3 Why on-line?	86
7.4 Applications	87
7.5 Conclusions	96
8. A Survey of On-line Computing - The Benefits And Problems	97
8.0 Introduction	97
8.1 The role of standard library packages	97
8.2 Advantages of on-line computer applications	99
8.3 Disadvantages of on-line computer applications	105
8.4 General considerations	110
8.5 Conclusion	119
9. Conclusions	121
9.1 Summary	121
9.2 The future	124
BIBLIOGRAPHY	127
10. Appendix 1. Illustration of Use of Both Packages	I
11. Appendix 2. Questionnaires Used in Experiment	IV
12. Appendix 3. Instructions and Problems Given to Subjects before Experiment	VII
13. Appendix 4. Routines Used in Experiment	XV
14. Appendix 5. Statistical Results for Student Think Times with Error Times	XXIX
15. Appendix 6. Statistical Results for Student Think Times without Error Times	XXXII
16. Appendix 7. Error Scores from Experiment	XXXV
17. Appendix 8. Documents Used in the Survey	XXXVII

## FURTHER BREAKDOWN OF CONTENTS

	<u>Page</u>
1. Introduction	1
1.0 Historical background	1
1.1 Structure	1
1.2 What is on-line computing?	2
1.3 Some features of present-day systems	4
1.3.0 Introduction	4
1.3.1 Basic forms of time-sharing	5
1.3.2 Paging	5
1.3.3 Segmentation	7
1.3.4 Associative memory	7
1.3.5 Virtual memory	8
1.3.6 Memory protection	8
1.3.7 Satellite computers	9
1.3.8 Filing systems	9
1.3.9 User priority	10
1.4 The NUMAC system	11
1.4.0 Introduction	11
1.4.1 Paging and Segmentation	11
1.4.2 Associative memory	12
1.4.3 Virtual memory	12
1.4.4 Satellite computer	12
1.4.5 Filing system	13
1.4.6 User priority	13
1.5 Summary	14

## FURTHER BREAKDOWN OF CONTENTS

	<u>Page</u>
2. The Uses of On-line Computing in Business	15
2.0 Introduction	15
2.1 Why on-line computing?	15
2.2 Classification	17
2.2.1 Financial and Administrative	18
2.2.2 Management Planning and Control	20
2.2.3 Marketing and Sales Operations	25
2.2.4 Distribution Operations	27
2.2.5 Factory Operations	29
2.2.6 Research and Development and Engineering	31
2.3 Conclusion	34



## FURTHER BREAKDOWN OF CONTENTS

	<u>Page</u>
5. Experiment on the Design of Conversational Packages for Managers	49
5.0 Introduction	49
5.1 Design of Packages	49
5.1.0 Introduction	49
5.1.1 Conversational level	49
5.1.2 Input formats	50
5.1.3 Errors	50
5.1.4 Explanation techniques	51
5.2 The subjects	51
5.3 The problems	52
5.3.0 Introduction	52
5.3.1 General description	52
5.3.2 Analytic solutions of inventory problems	53
5.4 The facilities	56
5.5 Experimental procedure	57
5.6 Performance measures	58
5.6.0 Introduction	58
5.6.1 Time	58
5.6.2 Errors	59
5.6.3 Questionnaire responses	60
5.7 Results	60
5.7.1 Timing	60
5.7.2 Errors	65
5.7.3 Questionnaire responses	69

FURTHER BREAKDOWN OF CONTENTS

	<u>Page</u>
5.7.4 General points	69
5.8 Conclusion	71

## FURTHER BREAKDOWN OF CONTENTS

	<u>Page</u>
7. A Survey of On-line Computing - The Applications	84
7.0 Introduction	84
7.1 Procedure	84
7.2 Facilities	85
7.3 Why on-line?	86
7.4 Applications	87
7.4.0 Classification	87
7.4.1 Applications of Type 1	88
7.4.2 Applications of Type 2	89
7.5 Conclusions	96

## FURTHER BREAKDOWN OF CONTENTS

	<u>Page</u>
8. A Survey of On-line Computing - The Benefits and Problems	97
8.0 Introduction	97
8.1 The role of standard library packages	97
8.2 Advantages of on-line computer applications	99
8.2.0 Introduction	99
8.2.1 Speedier information for decisions	99
8.2.2 Ability to try more possibilities	100
8.2.3 Ease of information retrieval and file updating	100
8.2.4 Convenience	100
8.2.5 Applications previously impossible	101
8.2.6 Training staff	101
8.2.7 Remote computing	102
8.2.8 Fast turnaround	102
8.2.9 Speed and ease of programming	102
8.2.10 Exception reporting	103
8.2.11 Improved communications	103
8.2.12 Stimulates user	103
8.2.13 Facilitates data-input	104
8.2.14 Bureaux advantages	104
8.3 Disadvantages of on-line computing applications	105
8.3.0 Introduction	105
8.3.1 Costs	105

FURTHER BREAKDOWN OF CONTENTS

	<u>Page</u>
8.3.2 Input devices unsatisfactory	106
8.3.3 Lack of experienced computer personnel	106
8.3.4 Programming difficulties	107
8.3.5 Transmission faults	107
8.3.6 Cost-justification	107
8.3.7 Error recovery messy and difficult	108
8.3.8 Centralisation problems	108
8.3.9 Training problems	109
8.3.10 Lack of standards	109
8.3.11 System requirements outgrow equipment	109
8.3.12 Loss of control	110
8.4 General considerations	110
8.5 Conclusion	119

## ACKNOWLEDGEMENTS

The author wishes to acknowledge the help of the following in the research described in this thesis:-

1. His supervisors, initially C.J. Constable and later D.A. Longbottom.
2. The Director, H.C. Baker and other members of the Business School for their help and advice on various occasions.
3. Professor B. Shackel, K. Eason and G. Corney, Department of Ergonomics and Cybernetics, University of Technology, Loughborough, for their advice in designing the experiment.
4. Dr. Hawgood, A.A. Young, B. Lander of the Computer Unit, University of Durham, for the use of the computer facilities in conducting the experiment.
5. All the company personnel who co-operated in the survey.

## 1. INTRODUCTION

### 1.0 Historical background.

This research project was conducted in the Durham University Business School (DUBS) using the on-line facilities provided by the Northumbrian Universities Multiple Access Computer System (NUMAC). The project arose after several members of DUBS staff had visited American Universities and noted the large emphasis placed on the teaching of on-line computing in U.S. business schools. The indication, at that time, was that on-line computing would very shortly become a powerful tool in business and that most managers would be using some kind of interactive device, such as a display terminal, in their problem-solving and decision-making. Whilst it is true that on-line computing has succeeded in aiding management in many ways, these original expectations have not materialised. Certain members of DUBS staff felt that it would be useful to investigate into why these early promises have not been fulfilled and, as finance was available, a research post was created for that purpose. This thesis contains the results of that research up to the time of writing.

### 1.1 Structure.

The thesis begins with a discussion of the development and techniques of on-line computing. Previous published work is analysed to determine the applications of on-line computing in business and this indicates 2 research needs:-



(i) an investigation into the problems (and their solutions) in those applications that are primarily concerned with the conversational capabilities of time-sharing. This is done in Section 1 through literature searches and experimentation,

(ii) an examination of the advantages and disadvantages of those applications that are more concerned with the data storage and information retrieval facilities of on-line computing. This is done in Section 2 through literature searches and a survey of organisations using on-line computing.

Finally the conclusion summarises the results and indicates future developments.

## 1.2 What is on-line computing?

On-line computing, known otherwise as time-sharing or real-time, is the method by which several users can 'share' the power of the computer. Users are normally connected to the computer via some on-line device such as a typewriter terminal and the computer allocates each of these users in turn, a small quantum of time. The processing speed of the computer is so high that each user is given the impression that he has the full attention of the computer. By using the terminal, the user can have access to a computer which is many miles away. Before going on to discuss the effects of this kind of system on management, it would be useful to look at the development of time-sharing.

Programming the first computers was such a difficult task that normally only a handful of people could use the computer. In order to run and test programs it was necessary to be able to stop the machine,



take readings and reset counters etc. Thus, in the early days, a few people would act as programmers and operators and these people were in a 'hands-on' capacity with the machine. As systems grew and the writing of software became simpler, the programmer's task soon became separated from that of the operator. With the development of high-level languages such as COBOL, FORTRAN and ALGOL, more and more people were able to write their own programs. Thus the number of people wishing to use the computer increased and larger and faster computers were developed. As these machines grew in size and speed, it was no longer possible, because of the expense involved, to allow them to remain idle for several minutes whilst the user debugged his program. This led to the introduction of techniques such as multi-programming, where several users share the CPU in order to keep it active during input-output waits. These developments, together with the increase in the number of users, made it impossible for everyone to have a 'hands-on' relationship with the machine and the familiar 'batch' system was introduced. In this system, the programmer gives his input to an operator, and receives his output sometime later, without having any contact with the machine itself. As the number of machine users increased, however, delays became greater and waiting periods of several days became commonplace.

Originally computers were used to perform complex calculations, mainly in the scientific area. As software developed, however, other applications were discovered where the computer was used for control and information retrieval. It soon became obvious that

some of these applications would require almost immediate access to the computer which the 'batch' system could not provide, except at very high cost. This led to the development of time-sharing systems as we know them today.

The very first time-sharing system was developed at the Lincoln Laboratory at M.I.T. in the U.S. defence system S.A.G.E.<sup>1</sup> This was in the middle and late 1950's and involved men such as Forrester, Licklider and McCarthy who were to play such an important role in the development of many of the early systems. Initially only special-purpose systems were developed, as in the S.A.G.E. project, where each user performed the same operation. Then came the development of general-purpose systems on special purpose computers. As a result of his work on the S.A.G.E. system, McCarthy in 1961, proposed the following features for general purpose computers.

1. A large primary core memory.
2. An interrupt system with error traps and I/O functions.
3. Non-stop operation.
4. Some memory protection.
5. Large secondary storage.

These proposals finally led to the first time-sharing computers in 1965 which were the GE265 and the UNIVAC 491.

### 1.3 Some features of present-day systems.

#### 1.3.0 Introduction.

The following sections will deal with the features of

present-day time-sharing systems and those facilities available in the NUMAC system on which all of the experimental work was carried out. These descriptions (see Wilkes<sup>2</sup>) however, should not be regarded as being general since most systems vary in their methods.

### 1.3.1 Basic forms of time-sharing.

The simplest method of organizing a time-sharing system is to divide the available core store into N parts and have N users each possessing the same amount of core i.e. (available core size) / N. Processing can then take place on a circular basis, each user program being allocated a quantum of time in turn. However the amount of core available to each user is normally so small that a more efficient method is required and a technique known as 'rollin-rollout' is more popular.

In the 'rollin-rollout' system, each user is allocated a partition, which may be part or all of the available core store, for a time-slice. After his time has elapsed his copy of core store is 'rolled out' on to a disc or drum and another user's program 'rolled in'. If there is enough available core, several users may be able to have partitions in core while the others form a queue on disc or drum. However, this method requires a large amount of secondary storage such as disc or drum and, in addition, the whole of a user's program and data are rolled in or out each time-slice even though only small parts of the program may be accessed during this time. To overcome this disadvantage some systems use a technique known as 'paging' with time-slicing.

### 1.3.2 Paging.

As stated, paging is a technique which enables us to efficiently utilise the available core store by only loading those parts of the user's

program that are required in a time-slice. Paging also provides a method in which consecutive pieces of program or data need not necessarily occupy consecutive areas of core. This is accomplished by dividing an address into a page number and a line number. Thus a program will be divided into areas known as pages and the core memory will also be divided into pages. A page is normally set to equal a convenient number of bytes. Thus in the IBM 360/67 computer a page is equal to 4096 bytes. The Supervisor is a resident program in a computer memory which handles all user requests. To enable consecutive pages of program to be loaded anywhere into core, the Supervisor forms a table containing the physical page number into which each page of program or data has been loaded. This table is then updated each time a change is made in the contents of core memory.

To access a word in core, the page number in the address is used to locate the physical page number given in the page table. This page number when combined with the line number, gives the physical address of the word in core. If the page table is kept in core, however, a severe time penalty is incurred as 2 core cycles would then be required to access a word instead of 1. Because of this time factor, fast hardware registers are usually provided to contain the page table.

In a time sharing system, users' programs are normally stored on some relatively slow secondary storage such as a disc until the user wishes to access them. When this occurs, the programs are transferred to core until the user reaches the end of his time-slice when they are 'swapped' to some faster secondary storage which is usually a drum. In such a system therefore, pages are continually being 'swapped' from core to drum and paging allows this traffic to be reduced considerably since a page need not be loaded until it is needed. This traffic between core and drum can be even further reduced by using a 'bit' (the smallest unit of memory) to represent whether or not a page has been altered. If it has not, then the Supervisor need not write back this page to the drum. The disadvantage of this method

is that the programmer must allow enough memory space for his program and data before run-time and he must also allow for any logical areas of program that he wishes to share space. This problem is overcome by a technique known as segmentation, described below.

### 1.3.3 Segmentation.

Segmentation is a method by which hardware is defined so that the binding or fixing of address parameters can be deferred until run-time. To achieve this, an extra set of digits are required in addition to the page and line digits in the memory address. These are known as the segment number digits and they are used to enter a segment table which gives the address of the base of the page table for the segment in question. The page table is then entered using the page number and this, when combined with the line number, gives the actual address. Both the segment tables and page tables are regarded as permanently resident in core memory when the user's program is running. The Supervisor loads a hardware register containing the address of the base of the segment table belonging to the user whose program is active at that instant. This number is then added to the segment number and used to enter the segment table. This technique enables users to share coding simply by having their respective segment tables pointing to the same page table. However, the disadvantage of the system is that 3 memory cycles are required instead of 1, thus making it impracticable. To overcome this, an associative memory, constructed from very fast registers, is provided which enables the access procedure described to be short-circuited on the majority of occasions. This associative memory also enables programs to grow dynamically as described earlier.

### 1.3.4 Associative Memory.

The associative memory (A.M.) consists of a set of fast registers containing number pairs made up of the segment number and page numbers

from an address combined to form a single number together with the corresponding physical page number. Whenever a memory access is to be made, the A.M. is searched to find whether a number pair exists corresponding to that contained in the address. If this is the case, then the physical page number can be obtained directly and combined with the line number to give the address. Hence only 1 core cycle would be required. If the corresponding number pair is not contained in the A.M., then the full procedure is executed and an entry is made in the A.M. in order that the 3 cycles can be short-circuited next time. The drawback to this method is that the fast registers required are very expensive and normally only 8 or 16 are provided thus making it necessary to suppress items of information in favour of others. Simulation tests<sup>2</sup> indicate that 8 registers are usually enough to cut down the full procedure on 90% of the occasions.

#### 1.3.5 Virtual Memory.

The virtual memory concept is a consequence of the techniques previously described. As only part of the user's program is in core at any one time, there is no necessity for the virtual memory size to be the same as the program size. Indeed the virtual memory can be several times as large as the actual program size. In a system with paging and segmentation as described above, the address is composed of a segment number, a page number and a line number and the various segments can be regarded as being contained in a very large virtual memory. This memory can then, be very large compared with the lengths of various segments held in it and this means that there can be space between segments, allowing them to grow dynamically. Since the memory is virtual, these gaps do not represent a physical loss of core or drum space.

#### 1.3.6 Memory Protection.

In any multi-programming environment, safeguards are required to protect one user's program area from that of others in core at that

instant. This memory protection can easily be incorporated, in a time-sharing system as described above, by extending the hardware register containing the base of the user's segment table so that it also accommodates the value of the length of this table. A check can then be made to verify that the segment number in the instruction is a valid one, i.e. one within the user's own program areas. Similarly, the segment table can be made to yield the length of the page table in addition to the base of the page table for that segment. Finally, complete memory protection can be obtained by the addition of 2 extra digits in the segment table which can indicate the class of access permitted i.e. read only, write only, read and write.

#### 1.3.7. Satellite computers.

Normally users interact with time-sharing systems, such as the one described, via some device as a typewriter terminal or graphical display. An extremely powerful method of interaction is through a satellite computer which can be connected to the central computer in the system. These computers are usually much smaller than the central machine and the connection is often a direct channel to channel one, e.g. IBM ASP system. These satellite computers are normally used to handle all the slow input-output to the central machine, such as consoles, leaving the larger computer free to perform the 'number crunching'. A great advantage of these satellite machines is that they enable remote users to enter large jobs in the on-line system by providing them with card reader and printer facilities.

#### 1.3.8 Filing systems.

Most modern filing systems derive from that designed by Corbato in the Compatible Time-Sharing System (CTSS) which was incorporated at M.I.T. in 1963. Space on the discs is usually divided into blocks and, in a paging system, there is normally a simple relation between page-size and

block size. For files of variable length, a chaining system is used. A table or storage map is usually kept, with an entry for each block on the disc. These tables then enable administration of blocks to be performed without accessing them. The number of entries can, however, be typically large and it becomes difficult to keep all the tables in core. With some sort of paging system however, the tables can be divided into sections which would then have a good chance of being in core at any instant.

Once space has been allocated, attention must then be paid to the design of the file directories, of which there must be one for each user. These directories are normally kept in a hierarchy i.e. the entries in the higher level being directories, the lower level being a mixture of files and directories and files alone in the lowest level. Files normally have conditions imposed on them, i.e. permanent or temporary, password protection, read only etc. and these can be incorporated in one nameword which is contained in the user's file directory. It is then possible, by providing links between user directories, to allow certain users access to other people's files.

### 1.3.9 User Priority.

In any time-sharing system a set of software tools are required to allow for different priorities for different users. These tools are normally contained in 3 of the Supervisor routines.

1. The Scheduling Algorithm.
2. The Logging in and Logging out program.
3. The Accounting Routine.

#### 1. The Scheduling Algorithm.

This is responsible for giving effect to priority decisions but always subject to the preservation of the overall efficiency of the system. This algorithm represents the point where the ideas of the system programmer



meet with those of the management.

## 2. The Logging in and Logging out program.

This program can also control user priorities by setting conditions on the logging in or out process.

## 3. The Accounting Routine.

As stated, both the scheduling algorithm and the logging in/out routine must consider the overall efficiency of the system in dealing with user priorities. The Accounting Routine provides the statistics required to determine whether this efficiency is achieved.

### 1.4 The NUMAC System.

#### 1.4.0 Introduction.

The Northumbrian Universities Multiple Access Computer System is used by both the University of Durham and the University of Newcastle. The central computer, at Newcastle, is an IBM 360 model 67 which possesses a set of hardware registers enabling the techniques of paging, virtual memory and associative memory to be employed. The central installation also consists of:

- 3/4 million bytes of core store
- 12 x 2314 discs
- 2 x card reader/punches
- 1 x paper tape reader
- 1 x 2301 drum for paging

The input-output units most commonly used on the system are IBM 2741 typewriter terminals and several 2260 video display units have recently been installed. The time-sharing system used is the Michigan Terminal System (MTS) developed at the University of Michigan.

#### 1.4.1 Paging and Segmentation.

As stated, the 360/67 computer possesses a set of hardware registers enabling paging to be used. The page size on the NUMAC system is 4K bytes.

The NUMAC system also possesses 16 segment tables, of which only 14 are actually used and only 1 of these for user's own coding.

#### 1.4.2 Associative memory.

The NUMAC system possesses 8 fast registers forming an A.M. If the number pair, containing a combination of segment and page numbers, is present in the A.M. then the total address translation time is 0.15 micro secs. However the NUMAC system also possesses an additional register which is used to hold the last translated address. If consecutive instructions are executed, and these are all within the same page, then the 0.15 micro sec. penalty is incurred only on the first reference to the A.M. but succeeding instructions in the sequence use the translated address from the extra register thus saving this time penalty. In this system, all 3 searches, that is the full translation process, the A.M. search and the reference to the additional register, all begin simultaneously and the first successful reference causes the others to terminate.

#### 1.4.3 Virtual Memory.

The IBM 360/67 uses a combination of  $\frac{1}{4}$  million bytes of core and 2301 drum to provide the user with a large one-level storage system which is used as if it were core store. Addresses on the 360 are held in words of 24 bits so that the largest address is  $2^{24} - 1 = 16777215$ . This is the upper limit of addressable virtual memory as far as the user is concerned. However MTS imposes a limit of 1 million bytes on the amount of this virtual memory that the user can have.

#### 1.4.4 Satellite computer.

The NUMAC system possesses a satellite computer (IBM 1130) at Durham connected to the central 360/67 at Newcastle via a G.P.O. line and a multiplexer channel. The 1130 is used with a card reader and a line printer to enable the remote users at Durham to submit longer jobs to MTS.

#### 1.4.5 Filing System.

This is basically the same as the IBM 360 Operating System, i.e. the user must first set-up the space on disc he requires and he will be allowed up to 15 extra extents. There are 2 types of files - normal sequential files and line files. The latter possesses a line directory at the beginning of each file, the rest of the file being broken down into lines.

#### 1.4.6 User Priority.

##### (a) Scheduling algorithm

In the system at Newcastle and Durham the basic algorithm is as follows:-

An ordered queue of jobs wait for the processor. A job can lose the CPU on either of three conditions:-

1. It has finished its time-slice,
2. It requires I/O,
3. Some other job has finished its I/O and given an interrupt.

The job that is highest up the queue gets the CPU first. If a job has finished its time-slice then it returns to the bottom of the queue. A job that requires I/O loses the processor until it has finished its I/O. When a job has finished its I/O it is given the CPU immediately until it reaches the end of its time-slice when it returns to the bottom of the queue. Time spent in I/O operations are not included in the time-slice.

##### (b) Accounting Routine

Users are allocated a certain amount of funds per week. These funds include charges for CPU time used, virtual memory used, measured in page-hours since the last time that user was connected with the system, and disc space (again measured in page-hours). In MTS, page-size on disc is the

same as that in virtual memory, i.e. 4K bytes. Finally, charges are also made for input and output (cards read, pages printed etc.).

(c) Logging in and Logging out.

In MTS, a user is logged in provided he has not exceeded his allotted amount of units for the current week. Logging in is almost always within 5 minutes of a request. A user is logged out whenever he wishes to be.

1.5 Summary.

The project then, arose as a result of certain numbers of DUBS staff visiting the U.S. and noting the emphasis placed on real-time computing. Time-sharing itself, developed initially at M.I.T. in the S.A.G.E. defence system. Modern systems possess many important features including paging, segmentation and associative memories, for efficient utilisation of core, together with memory protection, file-handling and priority techniques for user's themselves. The NUMAC system, on which the practical work for the research was carried out, also possesses many of these facilities.

## 2. THE USES OF ON-LINE COMPUTING

### IN BUSINESS

#### 2.0 Introduction.

In order to determine the most important areas for research, it was necessary to investigate whether particular business uses of on-line computing can be classified by specific characteristics of time-sharing. A classification which correlates the applications of on-line computing in business with the time-sharing facilities that they utilise, should help in determining the problem areas to investigate. A literature search was therefore conducted to discover the major uses of time-sharing in business.

#### 2.1 Why on-line computing?

Greenlees<sup>3</sup> has described four advantages of using computers on-line.

##### (A) Immediacy of information.

Information is available immediately to help the manager in his decision-making.

##### (B) Development of new decision areas.

When connected to the computer with an on-line device such as a display terminal, the manager is relieved of the time consuming tasks of calculating and information searching. He is thus able to explore more alternatives when making a decision and he can attempt solutions previously impracticable.

##### (C) Creative interaction.

Using simulation models, the manager can ask 'what-if' type questions

in his decision making and obtain immediate answers concerning the affects of solutions on various factors.

(D) Miscellaneous economic factors.

These include savings made in time and money due to the implementation of successful on-line systems. R.K. Pearce<sup>4</sup> has shown an example of the savings made by transferring keying operations in one company to on-line. Both data entry time and errors were reduced because programs could check errors on-line (saving verification) and, in certain cases, correct them (reducing error rate).

Obviously then, time-sharing can be advantageous whenever timing is critical but speed is not the only criterion for using computers on-line. Allen<sup>5</sup>, in a survey of about 100 companies, has found that time-sharing is also successful whenever a trial and error process is required. On-line computing can often stimulate the user in his problem-solving process. As Cox and Good<sup>6</sup> have shown in a survey of 15 major U.S. Corporations, 'split second responses increase a managers willingness to formulate questions and get data on which to make decisions.'

Not all applications, however, are suitable for on-line use. Indeed, as Schilling<sup>7</sup> has indicated, time-sharing should not be used when large amounts of calculation and input-output are required unless very powerful interactive devices, such as displays, are available.

## 2.2 Classification.

The sub-divisions selected for the applications of on-line computing in business were the functional areas themselves. It was felt that applications within each functional area would benefit differently from using time-sharing, e.g. process control applications would require immediate action and would therefore benefit from the speed of on-line computing. Management planning applications, however, will generally be trial and error processes and would therefore benefit from the computational and information searching capabilities of on-line computing. The following is, therefore, the classification chosen:-

- (A) Financial and Administrative
- (B) Management Planning and Control
- (C) Marketing and Sales Operations
- (D) Distribution Operations
- (E) Factory Operations
- (F) Research and Development and Engineering

Before dealing with each category, it is interesting to investigate the amounts of computer time spent in these areas. This has been estimated by Booz, Hamilton Inc.<sup>8</sup> in a survey of 38 U.S. companies and the following are the average distributions of total computer time for each functional area:-

- (A) 47%
- (B) 6%
- (C) 12%
- (D) 11%
- (E) 16%
- (F) 8%

As these figures are based on only one survey and all the firms involved were manufacturing companies, not too much emphasis should be placed on them. These results do show, however, that almost half of the total computer activity is accounted for by the financial and administrative area. This is, of course, as one would expect since most firms begin their computer activities in accounting.

#### 2.2.1 Financial and Administrative.

This category includes financial reporting and analysis, accounting, payroll, invoice billing and personnel. Applications such as payroll, in which the benefits of computer use are most obvious, usually account for most firms starting their computer activities and, as McKinsey<sup>9</sup> has stated, 'the cream has already been skimmed'. Indeed, in his survey of 36 U.S. companies, 30 had finished computerising their accounting work.

Although speed is not important for activities such as preparation of payroll, there are certain applications which benefit greatly from being on-line. On the administrative side, Argenti<sup>11</sup> has outlined several applications suitable for on-line use.

##### (1) Reporting by responsibility

Managers are often swamped by too much information. By using some kind of report generator operating on a common data base a time-sharing system can enable the manager to restructure information so that he is



only given what he needs.

(2) Management by exception

Again, the object is to avoid giving the manager useless information - they do not wish to know when things are going to plan but only when things go wrong. With on-line systems one can have continuous checking to indicate when faults occur and the production of reports so that exceptions can be quickly identified.

(3) Information retrieval

A time-sharing system is capable of storing and retrieving data rapidly so that literature can be quickly referenced.

(4) Resource allocation

Available resources are allocated to minimise time and money etc. needed to complete a task. On-line computing can help here particularly in immediate decision-making cases. Skandera<sup>10</sup> has indicated how, with the aid of simulation models, the consequences of various choices can be analysed.

Whilst the personnel department does not seem to have any particularly important use for on-line computing apart from information retrieval, Argenti<sup>11</sup> has mentioned several uses in improving training methods. Using display terminals one can simulate programmed learning methods and display the appropriate page according to the user-response. This removes most of the tedium from the learning task.

Business games can also be simulated using terminals and one can even simulate other competitors. Nanus<sup>12</sup> also mentions the suitability of simulation in training methods. However, these applications are costly and should not be the sole reason for developing an on-line system but more an extension of existing systems with such interactive devices as displays.

On the financial side, terminals can be used for obtaining rapid answers to mathematical calculations on payroll etc. On-line computing

can be used in costing where decisions are short-term in their nature. McKinsey<sup>9</sup> mentions the use of time-sharing in one company for maintaining cost schedules in various plants. These schedules work in conjunction with some manufacturing plan which is updated regularly and they make possible intelligent work-load levelling and allocation among plants. As Argenti and Pope<sup>11</sup> have stated, the main advantage of having an on-line costing system is that one can get immediate answers to costs of systems and can update information regularly.

### 2.2.2 Management Planning and Control.

This area involves those applications which aid management in its planning and control, e.g. capital investment analysis, resource allocation and mathematical model simulation. Those planning and control applications which are closely related to their functional area, e.g. sales forecasting, have been dealt with in that category, e.g. marketing and sales operations. The planning and control functions probably account for the smallest use of computers and this fact is supported by Booz, Hamilton Incs.<sup>8</sup> survey of 38 firms. This is not surprising if we consider Greenlees's<sup>3</sup> analysis of long-range planning or intermediate-term control where decisions in these areas are often highly unstructured and difficult to program. Even if this point is accepted, time-sharing can still be helpful in supplying the background information required and can assist in decisions involving interactions of many variables.

Applications in this area can, in general, be divided into four parts:-

#### (I) Simulation

McKinsey<sup>9</sup> has stated that simulation is perhaps the greatest aid to top management decision-making and having models on-line can be advantageous in several ways.

(a) Primarily, it is a trial and error process and time-sharing can

enable one to attempt more solutions and also reduce the overall problem-solving time.

(b) The use of time-sharing can also aid in the building of the model, particularly in reducing its development time, as Greenlees<sup>3</sup> has indicated.

(c) Using on-line devices such as terminals, the model can be made to report on 'what-if' situations during solution and enable the user to stop its progress if results are not satisfactory. This is a powerful advantage and justifies the use of time-sharing even for large, complex models which produce vast amounts of output. In these instances, the model can be operated on-line until results appear to be satisfactory after a series of 'what-if' analyses and the final run can be performed off-line.

(d) The advantage of simulation is that it enables the manager to analyse the consequences of different courses of action without actually using his company's resources. As Miller<sup>13</sup> points out, the benefit of being able to do this via an on-line device is that the manager is free to try out his hunches alone before 'laying his cards on the table'.

14

Drucker however, warns that the results of a model are only as good as the information input to it. He illustrates this by referring to the example of the Ford Motor Co., in 1956, when the design of the Edsel motor car 'flopped' because the input, obtained from statistical analyses of past trends, was false due to a late change in these trends during processing.

There are numerous examples of how simulation has been successfully used of which a few are described here.

(i) Nanus<sup>12</sup> has described how demand can be simulated using Monte Carlo techniques thus forming queues when orders cannot be filled.

In conjunction with an inventory model, a manager can test different policies.

- (ii) Cox and Good<sup>6</sup> have shown how systematic testing of various plans can be performed in a marketing information system using simulation models and data base information in an on-line mode.
- (iii) Boulden and Buffa<sup>15</sup> show how simulation has aided top managers in their decision making in a large forests products company. Here, a real-time system comprising 22 models is used for financial planning. Complex changes due to inter-plant buying and selling and changing product mixes require assessment of these financial affects which produce lengthy time-delays. The on-line system reduces both the time and effort in evaluating alternative plans.
- (iv) A large steel company<sup>15</sup> developed a model for production processes. Costs incurred in converting ores to molten iron, molten iron to steel ingots, the processing of ingots etc. were simulated. Managers could then ask questions regarding amounts of raw material needed to meet production forecasts or the cost affects of various hot-metal to scrap ratios etc.

## (II) Inventory Control

This is another area in which computers have successfully aided management in its decision-making as has been illustrated by Tatham<sup>16</sup> in the Courage Breweries group. Inventory control systems can benefit enormously from being on-line. The Westinghouse Electric Company<sup>17</sup> uses terminals in their system to enter orders and check stocks. Southern Television<sup>17</sup> also possess terminals to handle requests for advertisement spots on T.V. The Miles Druce firm<sup>18</sup> of steel stockholders, with warehouses dotted throughout the country, have a real-time system processing sales orders and stock records. The most important benefit they find from being on-line is that it provides a method of cross-ordering from one warehouse to another.

Caminar<sup>19</sup> states that the most important benefit of an on-line inventory

system is that immediate decisions can be made. Conversational computing is particularly beneficial when dealing with "exception" cases when the system may want to ask managers 'can we delay this order, or shall we budget for overtime in one week or slack time 4 weeks later?' This "conversation" can take place via some on-line device such as a terminal enabling the manager to ask several questions himself before reaching a solution.

Other advantages arise when re-scheduling at short notice or when enquiring about the present state of orders in the face of some customer request.

### (III) Resource Allocation

Ingham<sup>20</sup> describes the matching of jobs to available resources as a most powerful application of computers in cases where the combination of jobs and resources are very large. McKinsey<sup>9</sup> illustrates how the computer was applied to resource allocation in one company in his survey. This particular firm maintained cumulative records on labour efficiency. By having this data on the work-content of each component by work centre they were able to schedule man-power requirements to meet a production schedule.

### (IV) Capital Investment Analysis

Greenlees<sup>3</sup> has investigated the on-line use of computers in the area of capital investment analysis in several banks. The greatest advantage of using terminals was found to be in the speedy revaluation of alternatives when the relevant factors changed.

By far the most common application in this field is in the use of DCF or NPV packages. As Allen<sup>5</sup> has shown, these are suited to on-line use because of the 'trial and error' process involved. In his survey of about 100 companies he found that they all used some kind of DCF or NPV package. Argenti<sup>21</sup> too, supports this view describing DCF as a 'major on-line use of computers in accounts.'

The use of risk-analysis in determining risks is also suitable for on-line use. Warner<sup>22</sup> describes the analysis of extremes and simulation in this area. Other applications described by Argenti and Pope<sup>11</sup> in the finance side involve replacement theory (estimating lifetimes of items of plant using mathematical and statistical techniques) and decision theory (embracing decision trees and games theory).

A major application which benefits from being on-line is that of costing. Morton et al<sup>23</sup> advocate the use of display terminals for this purpose. They mention 4 types of decision where it would help.

- a. Pricing
- b. Make or Buy
- c. Return on investment
- d. Short term sales emphasis

Displays are essential here as typewriters are slow and noisy and require typing ability. With displays however, one can have graphical output where trends and deviations can be readily seen. The use of methods such as CPM or PERT, particularly in construction models or marketing new products, enable the progress of administrative operations to be monitored and related to budgeted values.

In general, however, as Schilling<sup>7</sup> illustrates, managers in this area are mainly interested in those applications which are 'high leverage' in nature i.e. help with higher level decision making. Miller<sup>13</sup> has stated that only the most powerful on-line devices can assist managers in these cases, e.g. displays with keyboard and light pen. The advantages of these devices are that large amounts of data can be displayed immediately without the need for slow and noisy typing methods and, more important, the manager has available numerous decision possibilities at once with the combination

obtained with keyboard, buttons and light pen.

### 2.2.3 Marketing and Sales Operations.

This area embraces sales forecasting, sales analysis and control, market research and sales order processing. The literature suggests that there is not, at present, a great deal of computer activity in marketing but there is a high potential usage in the future.

Computers are particularly useful when applied to sales forecasting. Time-sharing can be used for most of the mathematical techniques employed in forecasting such as moving averages, exponential smoothing, regression analysis and correlation (see Argenti<sup>11</sup> and Argenti<sup>21</sup>). R. Taylor<sup>24</sup> has described how these techniques can be applied to computer generated records of past sales and incoming orders so that forecasts can be made by product lines, individual products and market. On-line devices such as terminals enable these calculations to be made quickly and simply and managers can easily modify records to include, for example, new products for which there is no sales history.

McKinsey<sup>9</sup> describes an application in one company which consolidates sales forecasts from 31 countries. These are broken down by region, product and model and correlated with figures for 7 previous years. Trends are established for each product group and any forecasts that appear doubtful can be reviewed further. These analyses are then used in annual budgeting discussions where time-sharing, particularly with displays, can assist in searching for trends and deviations. Dash and Berenson<sup>25</sup> have mentioned regression and correlation analysis as being susceptible to on-line forecasting, pricing, packaging product strategy and product planning. Another example of sales forecasting given by McKinsey<sup>9</sup> involves a manufacturer of high-styled clothing, who uses computers to forecast sales and thus establish preliminary cutting schedules at the beginning of each season.

In all these cases, the sales forecasting, planning and control systems are based on an updated information file and it is the ease with which information can be obtained from this file that makes on-line computing such an advantage in the marketing area. McKinsey<sup>9</sup> has described how a major consumer goods corporation gives its staff access to such a comprehensive and detailed sales history file. Cox and Good<sup>6</sup> have shown how this ease of information retrieval can enable executives to ask questions via an on-line device and pinpoint reasons for a sales decline in order to reach an action decision more quickly.

Simulation can be of considerable help in marketing decision-making. In corporate long-range planning for instance, models can be used in conjunction with data bank information to enable decisions to be taken whilst the order is being processed with on-line terminals. Argenti<sup>11</sup> has shown how a simulation model of the company's market could allow for such variables as competitor action or the affect of tariff reductions, etc. Linear programming is another powerful technique in marketing which is aided greatly by time-sharing because of the trial and error process required. As illustrated in section 2.2.2 on management planning and control, time-sharing can even benefit when complex LP models, requiring large amounts of output for analysis are used. In this case on-line computing can be used in developing the program itself and, by a series of what-if analyses, can ensure that the model is running satisfactorily on the input data before performing the final run off-line. Argenti<sup>21</sup> illustrates how LP can be used by advertising to select the best media for a given product in a given market. He also describes how it can be used in determining the best mix of products to make efficient use of a company's manufacturing facilities. LP can also be used to calculate effects on profits of certain factors



such as producing more units or changing mix. For this kind of application graphs are more useful and hence displays would be better suited.

The ease with which on-line systems enable information processing to be done simply and quickly is an important factor in their use in order processing. Kamman<sup>26</sup> describes how a leading container firm considers this to be such an advantage as to possess 30 terminals for order entry. Other major applications of this are at Miles Druce<sup>18</sup> and most large airport reservation systems (BOAC). Tesco supermarkets also have a system of 32 terminals used for order-accounting<sup>27</sup>. Displays are used to put sales information into computer files at a large data centre. The advantage of having on-line terminals operating on a common information base is that data files can be kept up to date so that immediate action can be taken when needed. Cox and Good<sup>6</sup> have emphasised how this can be important in the management of fashion goods. Faster moving items are reported daily for quick re-order and slow moving items are also reported for speedy price reductions.

Finally, on-line computers can assist on the customer service side. Many banks use computers in their marketing service departments to examine the need for new customer services and their impact on earnings, costs and profits. Kamman<sup>26</sup> describes an interesting application in insurance companies where one has a conversational mode of discussion leading to the design of an optimum policy for an individual or corporation, based on personal goals and objectives.

#### 2.2.4 Distribution Operations.

This section deals with warehouse operations, shipment order processing and traffic problems. Inventory control is, of course, another major aspect of distribution but this has been dealt with in section 2.2.2.

Distribution problems such as vehicle scheduling and siting warehouses are common to most large firms whose resources are scattered geographically. As Argenti and Pope<sup>11</sup> have indicated, on-line computing can be a great advantage in the solution of these complex mix problems, which often involve trial and error processes using LP models or simulation. Cases where a firm has several customers at different locations and requires to know such facts as production quantities, localities of production and methods of transportation are classical examples of this type. Time-sharing minimises the overall solution time and enables one to investigate more possibilities.

The Miles Druce firm<sup>18</sup> of steel stockholders, whose on-line inventory system has been described in 2.2.2, were also concerned about the size of their fleet of vehicles used in a nation-wide distribution service and this was one of the major applications of the computer that was suggested by the Atkins consultancy firm. Courages<sup>16</sup>, the brewery firm, are also undertaking several projects in this field. At their Bristol depot, they have a scheme to optimise transport routing and scheduling - a highly complex task benefiting greatly from time-sharing. Their ultimate aim is to design routes and schedules that use transport in the most time-saving and economical way, not only with regard to geography, but also customer requirements. To achieve this information concerning customer requirements, drivers' hours and loading time is needed. These factors can, of course, change suddenly due to illness of employees, breakdowns etc., and time-sharing can enable immediate action to be taken in these cases. Other projects planned are:

1. Finding the best locations for depots and warehouses with regard to position of breweries and the pattern of demand and delivery requirements.
2. An exercise to assess the advantage of one-of-a kind deliveries against mixed deliveries of beers, wines and spirits etc.

No company department is completely independent of other functional areas and distribution operations are often related to other activities such as production and sales. McKinsey<sup>9</sup> described how one company has a quarterly manufacturing plan for each of 13 plants. These plans are updated monthly by reconciling revised sales forecasts, finished goods inventory records and work in process in assembly. This has resulted in the lead time between customer order and delivery being reduced and the cost of shipping goods from depots in surplus to those with shortages has been cut drastically. Time-sharing can assist in this kind of application by keeping the plans up-to-date with the most recent changes and enabling immediate decisions to be made when 'exception' cases occur.

#### 2.2.5 Factory Operations.

This area includes materials controls, production scheduling and quality control and accounts for the second highest use of computers in Booz's<sup>8</sup> survey. This is not surprising if we consider Dearden's classification of management control systems which consists of:-

- (A) Strategic or long range planning,
- (B) Managerial or intermediate term control,
- (C) Operational or day to day control.

Factory operations fall into class C where timing is critical and decisions are usually repetitive and highly structured<sup>3</sup>. Because of the nature of these decisions, time-sharing can benefit factory operations.

Argenti<sup>21</sup> has described several techniques which can be used in this area, particularly in a time-sharing environment. These are described briefly below.

##### I Linear Programming.

It is possible to build computer models representing machine capacity, throughput rates, stock holding costs etc. The computer can then be used

to calculate optimum values for these variables. Time-sharing is useful in building the model and in reducing the overall solution time.

## II Simulation.

This application is similar in character to LP. Simulation models can account for uncertainty and it is possible therefore, to study the effects of ranges of values of input variables. The advantages of time-sharing in this application have already been outlined in section 2.2.2.

## III Heuristic Models.

Models of this type can use the computer to plan production and decide stock levels by trying out different sets of rules. Time-sharing is useful for 'exception' cases requiring quick decisions.

## IV CPM.

Critical path methods are most useful in the planning of large projects where the most critical jobs are easily identified. Time-sharing enables the information to be kept up-to-date and on the spot decisions to be made.

Order explosion consists of the breaking down of complex products into components, assemblies, sub-assemblies, etc., and lends itself to computerisation, especially when more than 10,000 items are involved (see Ingham<sup>20</sup>). This process is an even more powerful application when linked with production scheduling and inventory control systems. McKinsey<sup>9</sup> has shown how the availability of data on the work content of each component by work centre is invaluable for scheduling manpower requirements to meet a production schedule.

R. Taylor<sup>24</sup> has demonstrated how order explosion can be related to inventory. Using information from design about the product specifications and from production concerning assembly and fabrication, a detailed product

information file can be built up. The parts explosion process produces input information for inventory programs, and, in addition, a list of machine and manpower requirements broken down by plants, department and machine centres which again can be used in scheduling. As the explosion process, and its use in decision making, is a trial and error one, time-sharing can be of great benefit.

In the field of quality control, Argenti<sup>11</sup> advocates the use of on-line computers in improving product quality and reliability with on-line process controls. He describes how this can be achieved in several different ways.<sup>21</sup>

1. Data Logging

Only records plant conditions and print out the records.

2. Direct Digital Control

This is the use of on-line, real-time direct control over plant.

3. Adaptive Control

In this method, variables are automatically adjusted to optimise output or yield using a mathematical model.

4. Evolutionary Operation

This is a continuous process which makes small changes in operation and observes the effects produced. If output improves, the new readings are adopted, otherwise more changes are tried.

Process control systems of these types can make the most of time-sharing as decisions are purely quantitative and timing is critical. Ingham<sup>20</sup> has stated that the application of this type of system has been confined to larger companies such as paper and rubber manufacturers, but the advent of smaller and cheaper computers will make it more easily justifiable for smaller companies.

#### 2.2.6 Research and Development and Engineering.

This includes applications involving product testing, product design

and evaluation. As Wallace<sup>38</sup> has pointed out, the majority of research work is of a purely quantitative nature and terminals should, therefore, be particularly useful. Surprisingly, however, the results of Booz's<sup>8</sup> survey do not support this view.

The most obvious application of terminals to R. & D. is in the use of packages for statistical analyses in calculating errors and for scientific calculations in solving complex problems<sup>21</sup>. On-line computing can be of value however, in aiding R. & D. management in its planning and control. The uncertainty associated with research work, particularly in justifying costs in terms of future profits and in judging when projects should finish make this one of the most difficult areas to control and time-sharing can help management with these problems.

Groves<sup>29</sup> described how terminals can assist project managers in the evaluation of a number of projects. This is usually done in two ways, DCF calculations and sensitivity studies. The use of conversational packages for these applications is further justified by the fact that evaluation is not a once and for all exercise but continues throughout the life of the project. He also shows how computers can be of assistance in project control. If the computer is informed of costs to date, revised estimates of completion dates and a revised index of chances of technical success, managers can be notified of projects which are either ready for review or overdue. The system could provide the manager with 'exception' reports showing when projects are overspent or running late or experiencing technical problems which cannot be solved. These controls are most valuable when there are a large number of projects involved. The great advantage of time-sharing here is that the information file can be kept up to date

and querying can take place both easily and quickly. These advantages have been outlined by Argenti and Pope<sup>11</sup> who mention CPM as an important technique for this type of project control.

The design stage itself is the area where the facilities provided by interactive devices such as displays and terminals can be used to the full. Winsbury<sup>30</sup> has demonstrated the advantages of on-line computing in this area. One advantage is that this relieves the need for draftsmen as the product can be designed using the display and light pen and a combination of film camera and graph-plotters could draw up the necessary charts. Edwards<sup>31</sup> illustrates this technique in describing how General Motors design car bodies using displays. The user can add, alter or delete displayed data using light pens and keyboard. The final image is captured on film by camera and this in turn can be scanned and digitised and finally plotted. Another benefit of on-line computing in design is that it allows the designer to test his ideas without building prototypes and he can obtain greater accuracy because more possibilities can be tried, by building simulation models for this purpose. An outstanding example of this is in the design of electronic circuits where many hours spent by technicians on building test circuits have been reduced to minutes using displays and simulation models of the electronics. The greatest benefit, however, of on-line designing is that it 'bridges the notorious gap between design, development and production departments - usually a lengthy iterative process of shuttling between departments occurs for a new product'.<sup>30</sup>

Finally, on-line computing can assist the researcher in recording and analysing data and holding and searching published literature. The latter is an important application as Groves<sup>29</sup> estimates that approximately 40% of research work is repeated through lack of information.

### 2.3 Conclusion.

The preceding sections have illustrated many of the applications of on-line computing within each functional area. The assumption that applications within each area would use different characteristics of on-line computing is not valid. The benefits of immediate information are common to most of the on-line computer applications, particularly in exception cases. However, in relation to other characteristics of time-sharing, the applications can be divided into 2 types:-

#### Type 1

These applications are purely computational and self contained, relying very little or not at all, on the storage facilities of time-sharing. Rather they benefit enormously from the conversational capability of on-line computing (i.e. the ability to 'talk' to the computer in languages closely resembling English). Some examples of this type of application that have been described in this chapter are:-

DCF and NPV calculations

Scientific calculations

Simulation and business games  
for training

#### Type 2

These applications benefit greatly from the information storage and retrieval facilities of on-line computing. Although there is some interaction, this is not normally conversational to the extent of type 1 applications where the user wants results from the computer without involving himself in programming. On-line computing is



advantageous in applications of this type as it enables data bases to be set up and kept immediately up-to-date. By enabling many users (at different locations) and programs to access these data bases, on-line computing facilitates integration of systems of this type. Thus these applications are often complex and inter-dependant e.g. an inventory control system can be integrated with a sales information system, and production system. Some examples of applications of this type that have been discussed in this chapter are:-

Reporting by responsibility/exception reporting

Resource allocation

Sales forecasting, analysis and control

Order processing

This classification will now be used to determine the areas of research to be pursued.

### 3. DEVELOPMENT OF RESEARCH AREAS

The 2 types of application of on-line computing in business, described in Section 2.3, indicate 2 areas of research:-

(A) Research into applications of Type 1.

These applications rely on the conversational method of problem solving that on-line computing provides. It is proposed to investigate the problems connected with this manager-computer conversation through a literature search and to evaluate some methods of solution through experimentation. This research is described in Section 1 (Chapters 4 and 5).

(B) Research into applications of Type 2.

Applications of type 2, because of the common data base and remote computing aspects of on-line computing, tend to be complex and integrated computer systems. It would be impossible (because of the short time available) and not very profitable (because of the variety of applications of this type) to develop, implement and evaluate an on-line computing system of this type. The research can, however, indicate the benefits and problems of these applications to assist others in designing successful systems. It is proposed to discover these benefits and problems through a literature search and by conducting a survey of companies using on-line computing in their business. This research is discussed in Section 2 (Chapters 6, 7 and 8). The thesis then, is divided into 2 sections:-

## Section 1

This describes an investigation of the problems and techniques that can be used in designing those applications primarily concerned with the manager-computer interface.

## Section 2

This discusses those applications using the information storage facilities of time-sharing. The advantages and disadvantages of these applications are discussed and the problems in designing successful applications of this type are highlighted.

SECTION 1

RESEARCH INTO APPLICATIONS

OF TYPE 1

## 4. CONVERSATIONAL PACKAGES - THE PROBLEMS

### 4.0 Introduction.

A literature search was conducted to determine some of the problems encountered when managers use conversational packages (i.e. applications of type 1) in their problem-solving. The objective of the search was to indicate where experimental work can assist in designing the manager-computer interface. Based on this review, the 5 major problems were:-

1. Response time
2. Conversational level
3. Errors
4. Input formats
5. Explanation techniques

Although the following sections will deal with each category separately, there is a certain amount of interaction between them. Thus the level of conversation may affect the response time and the input format should be consistent with the conversational level etc. In writing a conversational package, all 5 categories should be treated so as to preserve the overall efficiency of the package.

### 4.1 Response time.

This is probably the most critical factor in a time-sharing system. 'Response time' refers to the time a person has to wait before he gets an acknowledgement from the system after issuing a request. This request may be the pressing of a hardware 'attention' button or the 'return' key after an input command. The 'acknowledgement' may be an indication that

the system is 'ready' or the results of a calculation.

When the manager is using a conversational package via an on-line device such as a terminal, he can be thought of as 'talking' with the computer. If the system has a high response time then there will be delays in the manager - computer conversation. As Carbonell<sup>32</sup> et al have pointed out in reference to human conversations, long delays are tolerant if the listener receives some indications that his partner is preparing a response i.e. repetition of keywords etc. Sackman<sup>33</sup> has also illustrated this point in the S.A.G.E. system in the U.S.A. Operators at S.A.G.E. used display terminals, into which they typed a request whenever they wished to take some action. Delays were often quite long and operators became very impatient and frustrated. To overcome this frustration, the operators would continuously type in the request until an acknowledgement was received, even though this was unnecessary. This gave the impression that it was the last request that prompted the reply. Thus the lengthy delays prevented the operators from turning to some other task. To overcome this, the system was made to reply 'OK' immediately upon receipt of the message and the operator was then able to prepare his next action whilst waiting for the system. This was such a success that other messages were incorporated e.g. 'BUSY', 'WAITING'. Shackel<sup>34</sup> has also suggested some means by which the user should be informed of a delay so he can occupy himself with other tasks - even simple warning lights and perhaps some statistics on current operating characteristics would be enough to ease frustration in most cases. It would appear, then, that lengthy delays are acceptable if the user is given time to turn to other tasks (see Carbonell<sup>32</sup>).

Eason and Corney<sup>35</sup>, in an experiment in which the manager of a small research unit used an interactive system in his work, have also found

that response time was a major problem. It was hoped that the computer would free the manager from the laborious tasks of computation and information searching and thus enable him to devote more time to task reviewing and sub-task operating (planning, executing and checking a number of sub-goals of the master plan). This was not found to be the case. In normal work, the manager's task was often broken up by frequent and lengthy delays in the form of other urgent matters or a need for information and calculations. However, when the manager is disengaged from the task in this manner, he can accept some delay before proceeding. When continuously engaged upon one task, as in the interactive system, delays of several minutes, whilst too short to allow other tasks to be tackled, were still long enough to frustrate the manager and affect his line of thought.

It is obvious, therefore, that high response times are detrimental to the manager's problem solving process. Before ensuring that the response time does not affect the problem solving process of the user, it is necessary to know something of critical lengths of these delays. Miller<sup>36</sup> has done a considerable amount of work on response times. He states that humans tend to organise activities into 'clumps' that are terminated by the completion of a subjective purpose or subpurpose. This subjective sense of completion is described by psychologists as a 'closure'. In general, Miller states, delays become more acceptable after a 'closure' than during the actual process of obtaining one. This theory suggests that very complex packages should be organised into 'clumps' or logical sections in order to minimise delays during the attainment of a closure. Another aspect of problem solving that Miller<sup>36</sup> refers to is the human short term memory. When engaged on continuous problem solving, the short term memory is heavily filled and delays of 10 seconds or more 'will not

permit the kind of thinking continuity essential to sustained problem solving'.<sup>36</sup>  
It can be shown that, rather than a straight line decrease in efficiency as the response time increases, sudden drops in mental efficiency occur when delays exceed a given limit.

Unfortunately, response time is mainly controlled by the characteristics of the system itself and there is not a great deal that the user can do to influence this aspect apart from the methods mentioned. Miller<sup>36</sup> has, however, obtained some useful classifications of response time and the approximate limits beyond which affects occur. At best, the writer of conversational packages can ensure that these limits are not exceeded. Some of the most important categories are:-

1. Response to 'system are you listening!'

This is a notification that the time-sharing system is operating. It may mean the pressing of an 'attention' button and receiving some kind of acknowledgement. A delay of up to 3 seconds is considered to be acceptable by Miller.

2. Response to 'system can you do work for me!'

This may be a request to load or execute a program. Delays up to 15 seconds are reasonable and delays up to one minute can be tolerated if the user can spend his time organising notes etc.

3. Response to identification.

This is when the system has to recognise the user and allow him to use it - usually termed 'signing on' the system. Delays of up to 5 or 7 seconds are acceptable.

4. Time recovery from errors and failures.

Miller states that the user should be informed of the error within 2 seconds and, if possible, should be presented with any information he needs to recover.



Not too much weight should be attached to these limits however, as Miller does not offer any experimental evidence in their support. A far more reliable guide is given by Yntema et al<sup>37</sup> who have conducted some very well controlled experimental work on response times and their affects on the problem-solving process. These experiments not only demonstrate the affects of lengthy delays on user-performance but also serve as evidence that carefully controlled experiments can be performed in this field. Yntema et al proceeded to vary the response delay received by the subjects when problem-solving. The variation in this delay was measured against the net completion time for the problem. Initial experiments showed that, if the subjects were doing profitable thinking during these delays, the net completion time would fall during these delays. Further experiments then showed that the net completion time, in normal problem-solving, increased when the delay interval became long, demonstrating that whatever the subject is doing during the delay is negatively profitable. Thus the delay does more than make the subject wait but it actually damages his performance in some way. A final experiment suggests that response delays of 10 - 15 seconds are enough to cause this deterioration in performance.

#### 4.2 Conversational level.

'Conversational level' refers to the verbosity and complexity of the computer replies to the user of a conversational package. The degree of verbosity employed can greatly affect the user. Eason and Corney<sup>35</sup>, in their evaluation of an interactive management information system, have found that 'lengthy and self-explanatory instructions can be as frustrating to the expert as terse, coded instructions can be to the beginner'. Nickerson et al<sup>39</sup> support this in stating that a time-sharing facility should satisfy both novice and experienced user, the former requiring lengthy, explanatory instructions whilst the latter requires only

abbreviated messages.

There have been several experiments to compare verbosity levels but without much success. Corney and Taylor<sup>38</sup> have performed such an experiment in which 3 different levels of verbosity were compared. The 3 levels were:-

1. Lengthy, self-explanatory level, requiring the minimum amount of instruction in using the package.
2. Shorter printout requiring further explanation.
3. Very brief, cryptic printout requiring a detailed manual for instruction purposes.

Level 1 was found to be the faster throughout the problem solving sessions, but also became frustrating. Level 3 was initially frustrating but became satisfactory after use. It also, however, remained the slowest throughout. The experiment suggested that a level of verbosity slightly lower than 2 was preferred - perhaps just one-word questions. These results, however, should not be regarded as conclusive because the experiment was performed in a simulated time-sharing environment.

Root and Sadacca<sup>41</sup> have conducted a similar experiment in which only two forms were used - full word and abbreviated. Here, however, no preference was observed.

#### 4.3 Errors.

In general, there are 3 types of error which can be made when using conversational packages. These are:-

1. Due to wrong concept.
2. Due to wrong entry.
3. Due to typing errors.

Typing errors are probably the most common of these and the package should contain some simple correction facilities.

Errors can be a major factor in the efficiency of a conversational system. Sackman<sup>33</sup>, in his description of the S.A.G.E. System, has shown that errors were the most costly and time-wasting factor in the system and a greater efficiency was obtained with the provision of certain error correction facilities. Eason and Corney<sup>35</sup> have also found that errors were particularly time-consuming in their management system. Searching for the cause of error was the main factor in the increase of non-productive activity that was measured (6.6% non-productive activity before the computer and 22.1% after 4 sessions with the system). Whilst error reviewing will improve with experience, it is important that the novice is given enough facilities to prevent him from rejecting the system initially.

Finally Miller<sup>36</sup> has shown that the user should be informed that he has made an error within 2 seconds and, even more important, he should be supplied with any information necessary to recover from the error.

#### 4.4 Input formats.

The choice of input format for a conversational package can influence both the user and the efficiency of the package itself. There are several ways in which data can be input.

##### (i) Fixed format.

The user must specify his values in a fixed form and fields must be of a fixed length. This method has the advantage that the program does not have to do any decoding and the response time can therefore be made quite small. As Eason and Corney<sup>35</sup> have shown, however, these methods are both time-wasting for the novice and error prone.

##### (ii) Free format.

The user can specify his input in any form he wishes and input fields can be of any length. This method has the advantages that the user can enter his data without consulting manuals on format specifications (a time-wasting

and often irritating task) and there need be no limit to the accuracy of his data as with fixed format. However, the program will need to do a considerable amount of decoding which could affect the response time, which, as Miller<sup>36</sup> has shown, can disrupt the thinking continuity necessary for interactive problem-solving.

### (iii) Tagged fields

The user enters his data in any order he wishes, his fields being tagged for identification. This method has the advantage that the user is forced to think more about his input and is therefore less prone to error. Furthermore, the use of tagged fields enables the package to perform various checking operations on the input data thus reducing errors. Response time can, however, be affected and the user is forced to perform extra typing which can lead to frustration and typing errors. This method is very similar to using the computer in a batch system and it does not make full use of the interactive capacity that time-sharing provides.

All three methods of inputting data have been compared in an interesting experiment conducted by Root and Sadacca<sup>41</sup>. No preference was found for any of these techniques.

Another factor in inputting data is the number of items required in reply to a computer-written question. Corney and Taylor<sup>38</sup> have shown that this can greatly affect user performance. In their experiment on input formatting, all subjects expressed a preference for only one item per question. They found that multiple items caused considerable errors 'one waits for a keyword to appear on the print-out and then looks at the data'. Where several items are asked for, there will be more than one keyword and the user enters the wrong one.

#### 4.5 Explanation techniques.

These are characterised by the verbosity of the program and the dependance on the user manual. Obviously the more verbose the program, the less detailed need be the user manual. A compromise must be obtained, however, between the frustration of having verbose printout and of searching through the manual for the correct explanation of a shorter printout.

A useful feature, which has been successfully incorporated with the TSS system at System Development Corporation, U.S.A., is the use of callable explanations. Here the user receives various explanations in reply to typing in some keyword such as 'HELP'.

#### 4.6 Conclusion.

The various problems encountered when using conversational packages in applications of type 1 nature have been outlined in the preceding sections together with some of the techniques that can be used.

Even when these different methods have been selected so as to produce an efficient package for a particular application, there is still no guarantee that the subject will use it. As Chasen and Seitz<sup>42</sup> have pointed out, one of the chief difficulties is that people are creatures of habit. When under pressure, they prefer to be under a known pressure rather than risk an unfamiliar approach. There should, therefore, be an indoctrination phase where the subject is encouraged to overcome this resistance to change. A great deal of work along these lines has been done by the National Physical Laboratory<sup>43</sup>. They have conducted an experiment in using on-line packages to diagnose illnesses of patients at a Scottish hospital. As most patients have never seen a computer, and are naturally nervous, they have tried to 'humanise' the package. This has been done with the use of

'chatty' phrases such as 'I am a computer and I only understand, etc', and 'There is no need to rush your answers, please take your time.' The results have been encouraging and further research is planned. A very amusing description of the personality that can be given to an interactive program is given by Berkeley.<sup>44</sup>

Conversational packages can enable the manager to invoke the computer in his problem solving without knowing anything of programming etc. The literature search shows that these packages must satisfy managers who are either novices or experts. The greatest problem in designing a conversational package is in overcoming any frustration that the user may experience through error-correcting, response delays, explanation techniques and the verbosity of the package's replies. Some experiments (notably the work of Carney and Taylor<sup>38</sup> and Root and Sadacca<sup>41</sup>) have been conducted to compare different methods of minimising this frustration (which can be detrimental to the manager's problem solving process) but without much success. However, these experiments are useful in providing a methodology for future work and in indicating the areas to investigate.

As a result of this literature search, an experiment was conducted to compare 2 different conversational packages. The aim in designing these conversational packages is to minimise any frustration that the manager may experience in their use and to enable him to solve his problems simply and quickly without possessing any computer knowledge. The next chapter describes this experiment.

## 5. EXPERIMENT ON THE DESIGN OF CONVERSATIONAL PACKAGES FOR MANAGERS

### 5.0 Introduction.

In an attempt to tackle the problems of the manager-computer partnership that have been described in Chapter 4, an experiment was designed in which 2 conversational packages were compared using groups of subjects who were, hopefully, representative of the typical manager. The experiment was concerned with typewriter terminals, as it was felt that these terminals would be the most popular interactive device for this type of application, being relatively simple and cheap.

### 5.1 Design of packages.

#### 5.1.0 Introduction.

Both packages were written with respect to conversational level, input formats, errors and explanation techniques. Listings of both packages are given in Appendix 4.

#### 5.1.1 Conversational level.

As stated in Chapter 3, the problem in designing a conversational package of this sort is that it must be acceptable to both the novice and the expert. The inexperienced user will require a high conversational level, whilst the expert may only need short abbreviated messages. The 2 packages were written with different ways of achieving this.

#### Package 1

This consisted of a series of questions, each question being made

up of 2 parts - an abbreviated part contained in the first 5 characters and representing the variable whose value was required, followed by a description of the abbreviation. Before solving each problem, the user could select the number of characters output in the question by replying to the query 'LENGTH?' with the number of characters he required. Thus a length of 80 would give him the whole question, whilst a length of 5 would mean he received only the abbreviation.

### Package 2

In this package, the series of computer-written questions were written in 3 different levels of verbosity. These were:-

- (a) A very verbose, supposedly self-explanatory, form.
- (b) A shorter version consisting of 1 or 2 word questions.
- (c) A very brief, abbreviated form consisting of up to 5 characters.

As with package 1 the user could select the level he required before each problem by replying to the question 'VERSION?'

### 5.1.2 Input formats.

As most managers who wish to use the computer in their problem-solving do not possess any technical knowledge of programming or input formats, it was decided that free formatted input would be used. Whenever questions required more than one item input, these items were input together, each value being separated by commas.

### 5.1.3 Errors.

As nothing was known of the most common types of error likely to be made, it was decided not to include lengthy error diagnostics or correction facilities, but to use the experiment as an indication of the errors that may arise. On making an error, each subject was merely given an 'ERROR' message



and returned to the beginning of the package to re-solve that particular problem.

#### 5.1.4 Explanation techniques.

As both packages contained a supposedly self-explanatory verbosity level, users were merely supplied with a brief set of notes describing the package and no attempt was made to describe, in detail, the meaning of each computer-written question. Appendix 1 contains an illustration of the use of both packages.

#### 5.2 The subjects.

Sackman<sup>45</sup> in describing 5 experiments comparing time-sharing and batch processing, stresses that subjects for experiments of this type should be selected with respect to 2 criteria:-

- (a) They should be representative of the 'real-life' users of the system being evaluated.
- (b) They should be comparable with respect to their skills in computing.

The subjects chosen for the experiment were a group of students from the University of Durham Business School who were all reading for the degree of M.Sc. in Management Studies. Because of their background (many have had industrial experience) and the nature of their studies, it was felt that they would be reasonably representative of managers who might use such packages for the purposes of the experiment.

To ensure that the subjects were of similar experience, the students were divided into 4 groups on the basis of their scores on a test designed to measure their knowledge of quantitative methods. This was done to enable comparisons to be made between people of similar problem-solving ability. In addition, each subject completed a questionnaire to ascertain

their previous computer experience and typing ability (see Appendix 2). The group having the greatest knowledge of quantitative techniques was also found to contain people with considerable computer experience and this group was therefore omitted from the experiment. Only 2 of the remaining subjects had some previous computer experience and all subjects possessed little or no typing experience.

Each of the 3 remaining groups was then sub-divided into 2 more groups, one of which was given package 1 and the other package 2. Thus group 1 consisted of 2 groups of 4 people of similar background, as did group 2. In group 3, however, there were only 6 people, forming 2 groups of 3.

### 5.3 The problems.

#### 5.3.0 Introduction.

As the experiment was concerned with evaluating problem-solving techniques for managers, a problem typical of a manager's task was required. However, Sackman<sup>45</sup> points out that problems selected for experiments using computers should be suited to the corresponding mode of use of the computer, i.e. for on-line solution, the problem should be one which does not require vast amounts of input-output and calculation (see Chapter 2). It was felt that a simple inventory ordering problem would meet with both these requirements.

#### 5.3.1 General description.

The aim of an inventory ordering package is to provide the manager with information to enable him to balance operating savings with the costs associated with larger stocks. The problem concerns the variation between production and distribution, the manner of production versus time and the size and location of physical stocks. These problems arise at almost every stage of the production process from purchasing to the distribution of finished products.

In controlling these processes, any fluctuations in inventory, i.e. building up or depletion of stock, must be avoided. Obviously it is not possible to completely eliminate these fluctuations as any forecasts on which production plans are based will often be incorrect. However, an inventory control model can keep these fluctuations from becoming extreme by fixing inventory levels and plans based on clear assessment and balancing of risks.

Some of the most typical problems are:-

#### Sales

This area is concerned with the cost of poor service in keeping customers waiting.

#### Production

This area is concerned with having long manufacturing runs in order to cut set-up and change over costs and the affect that variations in operating levels have on employment, overhead and facilities costs.

#### Financial

This recognises that inventories drain off cash which could be used elsewhere to make a profit.

The aim then, is an overall control method which serves to co-ordinate each of these goals into one which maximises the aim of each area.

#### 5.3.2 Analytic solutions of inventory problems.<sup>46</sup>

To keep the problems from being too complex, 3 simple cases in inventory control were considered. In each case a quantity known as the economic batch quantity was computed.

#### Economic batch quantity

Consider a firm which manufactures a certain item of which so many

are required for a certain number of days each year. The items will be taken, in batches, from the factory to the inventory. The problem is to select the size of the batches so that the total cost involved in buying, storing and moving is a minimum. This size or quantity is called the economic batch quantity.

The 3 cases of inventory control for which an economic batch quantity was calculated were the normal case, finite production rate case and several products case.

#### A. Normal Case.

It can be shown that the economic batch quantity (EBQ) is given by

$$EBQ = \sqrt{\frac{2 A S}{I}}$$

Where A = cost of placing or getting an order,

S = annual usage in units,

I = annual cost of holding one unit in stock in one year

This is the normal case of inventory control and assumes the following 3 facts:-

- (i) Withdrawal rate is constant over time,
- (ii) Replenishment is instantaneous,
- (iii) Quantity ordered does not affect other costs and operations.

#### B. Finite Production Rate Case.

This case accounts for the fact that replenishment is not instantaneous but the item is produced over some finite period of time. The expression for the economic batch quantity then becomes:-

$$EBQ = \sqrt{\frac{2 A S}{I (1 - S/P)}}$$

where S = rate of usage or sale,  
P = rate of production.

C. Several Products Case.

This case deals with the situation in which production is cycled among several products. There are 2 points to be considered:-

(i) Usage and production rates for all products should be expressed in common units.

(ii) Cycle lengths for all products should be set simultaneously.

The most convenient unit is hours of production, i.e. sales of each product in the sequence are represented by the number of hours of production sold. This makes the production rate the same for all products i.e. 1 'hour' of product is made each hour of operation on that product.

The cycle length must be set first because run quantities cannot be set independantly as they would then be inconsistent. In setting the cycle length, the object is to keep the total setup or changeover costs and inventory costs a minimum as in the one product case. When several products are involved, however, it is easier to begin with the number of cycles per time period and then compute the individual run length. For only one product, the frequency of orders or runs (N) can be obtained from the formula

$$N = \frac{S}{EBQ}$$

Where S = total sales or orders.

$$\therefore N = \frac{S}{\sqrt{\frac{2AS}{I(1 - S/P)}}}$$

$$= \sqrt{\frac{SI(1 - S/P)}{2A}}$$

For a number of products, the formula for the total number of cycles is given by:-

$$N = \sqrt{\frac{\sum_j S_j I_j (1 - S_j / P_j)}{2A}}$$

Where  $S_j$  = usage or sales rate for each of the products  $j$ .

$I_j$  = unit inventory cost for each of the products.

$P_j$  = production rate for each of the products.

$A$  = total set-up or changeover cost for one cycle.

The length of a run for any product  $j$  is then given by:

$$EBQ_j = \frac{S_j}{N}$$

A full description of production planning and inventory control and the derivation of all 3 formulas is given by Magee-Boodman<sup>46</sup>, page 350.

#### 5.4 Facilities.

The on-line devices used in the experiment were IBM 2741 typewriter terminals situated in Durham and connected to the IBM 360/67 computer at Newcastle. The experiment was performed under the normal operating conditions of the Michigan Terminal System. As the problems represent a typical manager's task, and the facilities are not simulated in any sense, conclusions from the experiment should be applicable to other such manager-computer interaction systems.

## 5.5 Experimental procedure.

The first problem in conducting the experiment was to ensure that the response time of the system was consistent with Yntema's figures in order that the subject's problem solving process would not be affected. Thus a series of trials were performed, using the packages at different times of the day, and the response times were noted in each case. The best response from the system was obtained after 17.00 hours when the average delay was about 2 seconds. All the experimental sessions were conducted after this time.

Both of the inventory ordering packages were written in the conversational mode. Subjects were asked to enter which of the 3 types of inventory problem they wished to solve and the appropriate values of variables when the package required them (see Appendix 1). All questions required only 1 item input except in the 'several products' case when all the items for a particular variable had to be input together, each one being separated by a comma. The package then calculated the E.B.Q. and the number of cycles.

The procedure that followed was based on similar experiments conducted by Corney and Taylor<sup>38</sup> and Root and Sadacca<sup>41</sup>. A set of instruction manuals were given to the subjects and any questions were answered. These manuals described the use of each package and contained a list of the inventory ordering problems that were to be solved and the corresponding answers (see Appendix 3). After being shown the typewriter keyboard, the subject was asked to solve the 10 inventory ordering problems using the package concerned. This set of 10 problems comprised 1 session. The problems were selected in such a way that the session would last approximately a half hour with an inexperienced

user. This was based on the average length of a session in the experiments of Corney and Taylor<sup>38</sup> and Root and Sadacca<sup>41</sup>. As each package was designed both for the novice and the expert, each subject was given 3 sessions, spread over a period of 3 weeks during which their competence would increase. The second and third sessions were merely repetitions of the first, i.e. they had to solve the same set of 10 problems. Most of the subjects expressed surprise at this. In solving the inventory ordering problems, the subjects were asked to repeat any problems with incorrect values of E.B.Q.

During each of the 3 sessions described above, the subject was kept under close supervision in order that, as Sackman<sup>45</sup> states, any discrepancies could be solved immediately. As also suggested by Sackman<sup>45</sup>, statistical tests were performed on the results and these are described later.

At the completion of the third session, subjects were given a questionnaire to obtain their final reactions on the package (see Appendix 2).

## 5.6 Performance measures.

### 5.6.0 Introduction.

The aim of the experiment was to find the 'best' package for use by managers in conversational problem solving. Both packages have been designed to minimise the problems described in Chapter 4. An ideal package can be defined, therefore, as that which minimises errors, minimises problem-solving time (particularly the user's think time between computer queries and his replies) and minimises frustration. The experiments summed up by Sackman<sup>45</sup> and those of Corney<sup>38</sup> and Root<sup>41</sup> indicate the use of time, errors and questionnaire responses as suitable evaluation factors for this purpose.

### 5.6.1 Time.

The experiment was concerned not only with measuring overall



solution time but with distinguishing between computer response time and subject 'think' time. Since using an observer to measure times with a stop-watch is both prone to error and liable to affect the subject's performance, it was decided to adopt some independent method of measuring this time. Sackman<sup>45</sup> states that computer recording of times, especially when the subject is on-line, provides an accurate account of performance and a timing routine was therefore incorporated into the software of each package. This routine recorded times at each I/O operation, either by computer or subject. From this timing data, both the 'think' times (the time in seconds from when the computer has typed out the question to when the user has input his reply) and response times (the time in seconds from when the user inputs his reply to when the computer begins to type the next question) could be deduced for each question throughout the session. In a series of trials with a stop-watch, these times were found to be accurate enough for the purposes of the experiment.

#### 5.6.2 Errors.

As stated in section 5.1.3, the object of the experiment in this respect was to discover what errors were made. As in the experiments conducted by Corney<sup>38</sup> and Root<sup>41</sup>, errors were scored from the typewriter output. All errors were classed into 3 categories:-

- (A) Due to wrong concept,
- (B) Due to wrong entry,
- (C) Due to typing errors.

Unlike the experiment conducted by Root and Sadacca<sup>41</sup>, both detected and undetected errors were scored.

### 5.6.3 Questionnaire responses.

Questionnaires were given to each subject immediately after the completion of the final session and these were designed to measure how frustrated they felt with the package, particularly with the verbosity level. The questionnaires consisted of 2 parts:

1. A set of adjective pairs forming a scale along which subjects had to select one of 7 points. These adjectives were selected especially to measure frustration. Various other adjective pairs were included to disguise the questionnaire's intention.
2. A set of questions to ascertain whether the subject's attitude to computers is favourable or unfavourable.

The second part of the questionnaire was included to distinguish between those subjects who felt frustrated because of a general dislike of computers and those whose frustration was a result of the design of the package. The questionnaire is shown in Appendix 2.

### 5.7. Results.

The discussion of the experimental results will be broken down into 4 parts:-

- a. Timing
- b. Errors
- c. Questionnaire responses
- d. General points

#### 5.7.1 Timing.

Using the timing data recorded by the software, it was possible to separate the subject's think times from the computer response time. The typewriter output from each subject's problem-solving session then enabled

these think times to be attributed to their particular user-response. Thus it was possible to deduce the total think time for each of the 10 problems in a session. This time did not include the response time of the computer which was always within 2 seconds and therefore had no effect on the problem-solving behaviour. This total think time for each problem was calculated in 2 parts:-

- (a) Total think time which included times to recover from error and times spent in re-solving those problems that gave incorrect answers.
- (b) Total think time ignoring any error times that occurred i.e. only those think times involved in solving the problem correctly.

There were, however, several problems involved in applying statistical techniques to the results for purposes of comparison. These were:-

(i) The sample sizes were small - 4 people were in groups 2 and 3 whilst group 1 possessed only 3 subjects for each package.

(ii) Eleven subjects, therefore, used each package for 3 problem-solving sessions, making a total of 66 subject-sessions in all. However, in 13 of these 66 subject-sessions, the time-sharing system went 'off the air' due to various faults and, on these occasions, the timing data was deleted from the system. As making the subjects repeat those problems for which the data was lost would only give them further experience and therefore affect the results, nothing could be done concerning this missing data. On these 13 occasions, therefore, the sample sizes would be still further reduced.

A comparison of the mean times taken in performing the problem-

solving was made between package 1 and package 2. The student t test for differences between means was used and the analysis was performed for 2 cases. All student t values were computed on the NUMAC system using the routine listed in Appendix 4. Initially only the times described in (a) above were used.

#### CASE 1

This case compared the mean times of group 1 with those of group 3 for both package 1 and package 2. Group 1 contained those people with the lowest scores on the quantitative test, whilst group 3 contained those with the highest. The purpose of this comparison was to test for any significant difference, with respect to mean think times, for the groups. As group 1 and group 3 were the most widely separated with respect to ability, the hypothesis was made that, if no significant difference was observed, all 3 groups could be treated together, i.e. they would all be of similar ability. This would then give a maximum sample size of 11 for each package. Student t values were computed for the difference between mean times of group 1 and group 3 on the first 3 problems of each session making a total of 18 t values in all (3 sessions, 3 problems, 2 packages). Only 1 of these t values indicated any statistical significance. The conclusion was, therefore, that the total think times were all from the same 'population'. These results are listed in Appendix 5, table 1.

#### CASE 2

As it would appear, from Case 1, that there is no difference between think times for groups, all subjects using package 1 were compared with those using package 2, for each of the 3 sessions. Because of the system failures described earlier, the numbers in each sample varied from 8 to 11.

As the problems themselves varied in their degree of complexity, comparisons were made on a problem by problem basis. A total of 30 t values, for differences between mean times on each package for each problem, were calculated, of which only 4 were significant at the 5% level and 8 were significant at the 10% level. Although no definite conclusions about the relative efficiency of the packages could be drawn from this statistical analysis, it was interesting to note that in 27 out of 30 comparisons, i.e. in 90% of the comparisons, package 1 produced a smaller mean think time than package 2, i.e. the t values were negative. Furthermore, of the 26% that were within 10% significance, all were negative. Thus, it would appear that package 1 produces faster think times when 'error times' are considered. These results are listed in Appendix 5, table 2.

N.B. The 2-tail test was used for all these cases since no prior assumption was made about which package was the more effective, that is the null hypothesis was that there was no difference between mean scores for each package.

Although the mean think times for package 1 were consistently lower than those for package 2, the variances of think times about the mean in each case were large and this, coupled with the fact that sample sizes were small, meant that statistical significance was unlikely to be observed. These variances might be attributed to 2 factors, error times and verbosity levels.

#### Error Times.

Whenever the subject made errors on a problem, his total solution time would, of course, be much greater than if no errors were made.

Thus, within each sample, there would be subjects with "error times" and those without, causing large variances about the mean.

#### Verbosity levels

Whenever a subject changed verbosity levels, his think time usually increased considerably whilst he adjusted to the new level. As subjects were left to make these changes when they wished, each sample contained a mixture of times for both those who had changed verbosity and those who had not. Again, this would cause large variances about the mean think times.

As a result of the above argument, it was decided to use a test which dealt only with score rankings and not actual values, and the Mann-Whitney U test was selected. This was the third case of comparison of the packages. Again the calculations were done on the NUMAC system using the routine listed in Appendix 4.

#### CASE 3

As with Case 2, the U value was calculated for all subjects on each session for each of the 10 problems. Of the 30 U's calculated, only 4 implied statistical significance. As in Case 2, however, 26 of these U values showed that package 1 had consistently lower times than package 2 and, in the 4 cases where significance was shown, all implied that package 1 was fastest. These results are listed in Appendix 5, table 3.

Thus it would appear that package 1 produces a lower mean think time than package 2. However, this may be due to the errors themselves, i.e. there may be more errors made on package 2 and, in order to investigate this further, the times described in (b) - that is, the total think

times without the time taken over errors - were treated for the same 3 cases.

#### CASE 1

Student t values for the differences in mean times of group 1 and group 3 again showed there was no significant difference between the times of each group. These results are shown in Appendix 6, table 1.

#### CASE 2

Thirty student t values were calculated as before and similar results were obtained. Only 5 of the t values were significant at the 5% level, and 8 at the 10% level. However, on this occasion, 29 of the t's were negative, showing that, on 97% of the comparisons, package 1 had a lower mean time than package 2. Again, all the significant comparisons were negative. In addition, of the 3 t's that were positive in the previous case for the times in (a), only 1 remained so. This indicates that when error times are omitted, package 1 appears to perform even better - this is consistent with the observed distribution of typing errors discussed in section 5.7.2. These results are shown in Appendix 6, table 2.

#### CASE 3

Again, in the Mann-Whitney U test, similar results were obtained. Only 6 of the U values implied statistical significance and of these, all showed that package 1 produced the lower think times. Only 5 of the comparisons indicated that package 2 produced lower times. (Appendix 6, table 3).

#### 5.7.2. Errors.

As mentioned earlier, errors were classed into 3 categories:

1. WC - due to wrong concept.
2. WE - due to wrong entry.
3. T - typing errors.

All errors could be placed in one of these categories, and the results are summarised in Appendix 7.

The only really significant difference between the packages was that package 2 produced 50% less typing errors than package 1. This is even more remarkable when we consider that 8 people had some typing experience on package 1 compared to 4 people on package 2. As both packages required identical input for all problems, it was assumed that the difference in typing errors was due to random variations and not to any assignable cause.

As one might expect, the number of errors made in each category decreased from session 1 to session 3, 50% of all errors occurring in the first session.

The most common type of error was class 2 (WE) which accounted for 66% of the total. Most wrong entry errors were due to the fact that subjects were anticipating the question with the result that the wrong item was input as the reply.

As stated previously, the purpose of the experiment with respect to errors was to ascertain what types of error were made, and how one could correct or avoid them. In this respect, the following points are important:

(a) The most striking feature of the scores was that the majority of errors occurred on two of the 10 problems - problem 3 and problem 7. Both these problems differed from the other 8 in that they required several items input together. The following is the percentage distribution of all errors for each category on these two problems:

1. WC - 33%
2. WE - 65%
3. T - 50%



Problem 3 required 5 items input together, whilst problem 7 required 10. Both the WC and WE errors were shared equally between the two problems but problem 7 had twice as many T errors as problem 3. It would appear that the majority of errors could, therefore, be eliminated if all questions required only one variable input per question. However, where many items are required per variable, this policy would make the package slow and repetitive and it would be more satisfactory to have a compromise between a higher error rate and a frustrating type of package. From these results, however, it would appear that the number of typing errors increases considerably with the number of values input together and the experiment suggests that this compromise should be no greater than 5. These results agree with the conclusions of Corney and Taylor<sup>38</sup> on a similar experiment.

b) The majority of wrong concept errors were a result of the question itself being misunderstood. This was true even in the supposedly self-explanatory verbosity level on package 2, and when the maximum length of 80 was given on package 1. Some subjects, surprisingly, found that certain questions posed an ambiguity. This misunderstanding arose more often, however, on package 2 when subjects changed verbosity levels and this is a possible explanation for the 14% more wrong entry errors on this package. Errors could be further reduced, therefore, by the following points:

1. More care should be taken in the wording of supposedly self-explanatory responses to ensure that no ambiguity arises.

2. When using programs similar to package 2, the user should be provided with some additional explanation techniques which could be used when difficulty is experienced after changing verbosity levels,

e.g. by typing in some keyword such as "HELP", the user could be provided with an explanation of the question causing concern.

(c) Wrong concept errors also arose from the way in which the input data was listed on the reference sheet. For all problems, this was listed in the order and form required by the package with one exception. In the case of problem 7, data was listed in table form (because there were 10 values per variable) and the majority of subjects were confused over this. Errors would be further reduced if the input data is set out in the form required by the package.

(d) Based on the views of each subject and the types of error made, the following correction facilities are suggested for packages of this type:

(i) For errors which can be detected by the program immediately, e.g. too many digits in field, incorrect character in reply, the question last output should be repeated, after informing the user of the type of error.

(ii) For other errors not so easy to detect, the user should be provided with a facility whereby he would be returned to the beginning of the package. The important point here is that the user should be protected from the rather frightening system messages that can arise on certain errors. Most systems to-day provide routines which will return control to the user on encountering these errors, and packages written for inexperienced computer-users should incorporate them.

In this experiment, the students were provided with such a facility for most errors. However, there were certain occasions when the error was not foreseen and the user was returned to the system. All subjects were considerably confused in these instances and had to be returned to the package by the instructor before continuing.

### 5.7.3 Questionnaire responses.

Of the 22 subjects who took part in the experiment, 20 had a favourable attitude to on-line computing, i.e. they thought that conversational packages were useful in solving business problems and that managers should be encouraged to use them.

There was no noticeable difference between the packages on the adjective-pair scores. These were designed to measure frustration, particularly with verbosity levels, but the majority of subjects scored at the neutral point. Of those who fell on either side of this point, the majority felt that the packages were more relaxing rather than annoying.

### 5.7.4 General points.

In addition to the 3 factors described above, several general points arose from the experiment:

#### Verbosity

It is interesting to investigate in detail the verbosity changes employed by each student throughout the experiment. On package 1 these involve a change in the number of characters output per question, whilst on package 2 they involve selecting 1 of the 3 levels of verbosity described earlier.

#### (i) Package 1

At the end of the initial session 8 of the 11 subjects had chosen a length of 5 or less by the third problem. This is equivalent to using version 3, i.e. the abbreviated one, on package 2. By the final session, 9 students had opted for a length of 5 or less.

#### (ii) Package 2

If we consider each session to consist of 110 subject-problems

(10 problems by 11 students) then approximately one-third of this figure was spent using each version for the first session. By the second session, however 87 subject-problems were solved using version 3 and, after the final session, this figure had risen to 107.

Thus, we can conclude that:

- (a) The most verbose level, although essential for inexperienced users of the package, is only required for the first few problems, after which most subjects felt competent enough to switch to a lower level.
- (b) Users very quickly progress to the abbreviated version in package 2 or its equivalent in package 1 (i.e. a length of 5 or less). This progress is so rapid that version 2 on package 2 is not really justified in this experiment.

#### Work Area

Each terminal possessed a work-table at the right of the keyboard upon which students could place their reference sheets. In spite of this, however, the majority of students attempted to balance the sheets on the left of the keyboard or held them in their left hand - often with great difficulty - when typing. A work area would have been better suited if placed at the left of the keyboard or, as one student suggested, some type of piano-music holder was provided at eye-level.

#### Explanation techniques

The majority of the students felt that the instruction sheets, together with the verbosity of the packages, provided enough initial instructions for the novice to be able to use the packages simply and efficiently. Only 2 of the subjects felt that they required more initial instruction.

## 5.8 Conclusion.

No significant difference in total 'think' times between the packages could be discovered using the three evaluation factors described. There are indications, however, that packages similar to package 1 have advantages over those of package 2 design. These are:

1. Package 1 consistently produced noticeably smaller mean think times than package 2 on over 90% of the cases treated.
2. The study of verbosity effects show that version 2 on package 2 is really a luxury and users only require a very verbose version and an abbreviated one. If we ignore version 2 on package 2 then both packages become very similar, i.e. the questions consist of abbreviated and very verbose versions. However, in the case of package 1, both these versions are combined i.e. the questions contain both an abbreviated part and a verbose explanation of this part (see Appendix 1). This is a distinct advantage over package 2 as no adjustment is necessary when verbosity levels are switched. As described in section 5.7, both errors and times often increased when these adjustments were made and this could possibly explain why package 1 produced shorter mean times.

In addition to these points, some useful guidelines are provided for future experiments in the field, these are:

(A) The measuring of times using software can be both accurate and free from human error. Care must be taken, however, to ensure that these times are not lost when system failures occur.

(B) The important point when dealing with errors is to protect the user from the system - a difficult task as it involves foreseeing the types of error that will be made.

(c) Errors should be classed while the subject is present and within 'recall-time' of the experiment. In this way, any doubts can be cleared up with his aid.

(d) The lack of statistical significance in the timing results was due to the small sample sizes and the large variances between users. Significance could be shown, therefore, by having larger samples. However, if the sample sizes are small, then variances can be decreased in 2 ways:

(i) By forcing users to change verbosity at similar problem stages on both packages. This is essential if timing comparisons are to be made, although information about verbosity affects is lost.

(ii) By conducting some previous trials on the keyboard involved to ensure that all groups are balanced with respect to typing speeds and keyboard familiarity.

Thus there are indications that, for applications of type 1, techniques such as those used in package 1 have advantages over those in package 2.

SECTION 2

RESEARCH INTO APPLICATIONS  
OF TYPE 2

## 6. THE PROBLEMS OF LARGER SCALE APPLICATIONS (type 2)

### 6.0 Introduction.

The last 3 chapters have dealt with those problems concerned with the man-machine interface. These problems are more relevant to applications of type 1 (i.e. relatively small, conversational packages) than to those of type 2 (i.e. more complex, often integrated applications). This is because the tasks of problem finding and problem solving are much clearer in type 1 applications than in the more complex and larger applications of type 2. Thus it is not difficult to discover what the user wants from a DCF package or a statistical package. However, to define what the manager requires from an information system is a much more complex problem. Whilst the man-machine interface problems discussed in the latter chapters are still relevant when dealing with type 2 applications, the nature of these applications give rise to completely different causes of dissatisfaction or failure. This chapter reviews the literature to discover these causes. As system problems (e.g. design, development etc.) are more important in applications of type 2, many of the limitations of off-line applications will also be applicable to on-line systems. A total of 9 major causes of dissatisfaction or failure were discovered and these are discussed in the following sections.

### 6.1 Management involvement.

Management involvement, particularly in the early stages of design, is essential for a successful system. In most systems the degree of success achieved is directly related to the level of management involvement. This factor is strongly supported in a survey of 25 U.S. corporations who have attempted to build marketing information systems. This survey, conducted



by Cox and Good<sup>6</sup>, discovered that the high level of management involvement was the most striking characteristic of the more successful systems. In all of these cases, top management had seen the possibilities of the system and followed its progress, devoting large amounts of time and effort in the development stages. Cox and Good<sup>6</sup> stress that the building of an information system cannot be left to a technical group because there are 5 important questions which can only be answered by managers. These are:

- (i) How can we organise the development of a better information system?
- (ii) How sophisticated should the system be?
- (iii) What development policy should be adopted?
- (iv) What are the chief characteristics of the system?
- (v) How much cash will be involved?

The aim of the management information system should be, of course, to supply the manager with the information he requires to make better decisions. However, assessing the manager's information needs is one of the system designer's most difficult tasks. If the system is going to succeed in this aim it is important that everyone concerned with the system should be consulted in making this assessment. As Harbridge and Shrimpton<sup>47</sup> stress, all managers, particularly line managers, should participate in defining their information needs.

## 6.2 Organisational problems.

As stated in section 2.3., computerised applications of type 2 are often linked to similar systems in other company areas. On-line computing, with a common data base, encourages this integrated system approach. Thus, in on-line applications of type 2, departmental boundaries often disappear and such systems will involve many different departments and individuals, e.g. top managers, systems analysts, operations research specialists, programmers, line managers etc. This gives

rise to a series of organisational problems in involving and coordinating the efforts of all these people to develop a successful system. These organisational problems have never been present in any pre-computer systems and most companies fail to find a solution, often designating the task to a technical group who lack the skills necessary to produce a successful system. Cox and Good<sup>6</sup> suggest that the best solution to this problem is to have a senior executive act as coordinator and to give him cost control of the project so that he can be effective. It is essential that this coordinator represents the department or division that will use the system so that he is aware of the problems and needs of the user.

### 6.3 The Human Element.

Many applications of type 2 involve a set of programmed decisions based on the manager's decision-making process. This decision process, as Cox and Good<sup>6</sup> point out, is not always quantifiable because managers' decision rules are often intuitive. This is where a great many of the information systems of the past have failed - a manager's hunches can often be more accurate than a decision based on management science techniques. One can never hope to simulate effectively this 'human element' of decision-making but care must be taken in developing a model of the decision process. It is important that the model should not make the decision but supply the manager with the alternatives and still allow him the flexibility to abandon these results if his intuition tells him to. In those instances, where the model actually makes the decision, the manager must be aware of the assumptions involved. The danger is that managers will put too much faith in the computer information and allow themselves to be cut off from reality. This point is well illustrated by Drucker<sup>14</sup> in the case of the Ford Motor Company and the Edsel motor car (Chapter 2). Here the assumptions

in the design process were incorrect and the car was a failure.

#### 6.4 The Development Policy.

The strategy adopted in the development of a system can greatly affect its chances of success. While it is not possible (and probably not desirable) for most companies to have a detailed development plan, there should be some future development policy. This is essential if the company is to avoid having a series of independent applications using different files of data which often contain a lot of common information (see Harbridge and Shrimpton<sup>47</sup>). Whilst a fully integrated information system is a very complex task and cannot be set up immediately, there should be some plans made for integration to be performed at a later date.

Even if it is possible to provide management with a sophisticated information system immediately this is often undesirable as the system only succeeds in confusing managers. This point is illustrated by Cox and Good<sup>6</sup> who stress that such systems should be developed gradually to enable the manager to get used to them and thus gain the confidence needed to overcome any initial fears and resistance to new methods.

Finally, all development plans should provide some flexibility as it is difficult to foresee the future direction of the system. One method of giving this flexibility is described by Cox and Good<sup>6</sup>. In this method, the system is based on a 'disaggregated data bank', i.e. rather than replacing data, new inputs are maintained along with existing data thus allowing the designers to make changes without losing information. However, this is very difficult in on-line systems because of the extra storage costs involved if the data is to be readily accessible.

## 6.5 Costs.

Costs have been a major complaint that managers have held against computers ever since their introduction. The 'electronic brain' image of the early machines, coupled with the success, particularly in savings, of early applications such as payroll, created an attitude to computers by managers that would not be tolerated with other innovations. Thus managers were prepared to devote huge amounts of cash on these machines, (probably for fear of falling behind in the 'race') without demanding any of the justifications that would be required for other projects of the same size. The advent of time-sharing, with the necessity for large amounts of secondary storage and program interrupt and memory protection features only served to increase the cost of in-house systems. In a survey of 15 computer manufacturers by Popell<sup>48</sup>, 7 thought that time-sharing would be 30% higher in cost on average than conventional systems. The survey indicated that only very large organisations would be able to afford their own systems. In a similar survey of 48 non-users, Popell et al<sup>48</sup> discovered that 42% quoted costs of equipment, software and operating as the major obstacle to their use of computers. However 25% were in favour of time-sharing on a bureau basis because of the lower costs involved. Miles Roman Ltd.<sup>49</sup>, in a survey of 300 computer systems, also found that these high costs of systems would indicate a shift to computing on a bureau basis.

One of the major causes of these high costs is the expense of the hardware itself, particularly in obtaining maximum efficiency from the equipment. Popell et al<sup>48</sup> have found that 'system requirements constantly outgrow equipment' and 'total potential throughput is difficult to achieve.'

Harbridge and Shrimpton<sup>47</sup> quote 80% as the highest level of efficiency that most computers can attain. They indicate that this is due to the fact that computer managers cannot get feedback on equipment performance to assess this efficiency. Even if some method of sampling the 'work rate' is used, the results are often too complicated to analyse. This is even more true of on-line systems where many users share the CPU, each performing often completely different tasks.

However, some of the blame for the high costs of computers must be placed on management itself. In many cases top managers assign the tasks of utilising the computer and controlling costs to a more junior level. This has the result that the machine is not applied to those business areas where the potential benefits lie (see Harbridge and Shrimpton<sup>47</sup>).

Another cause for managers' dissatisfaction with the high costs of computer applications is the difficulty of assessing the expense involved. Early estimates of costs of computer applications are often inaccurate because of the many factors involved, e.g. hardware, computer time, G.P.O. time, program development time. Many systems appear to 'fail' because they are often justified on the savings involved (e.g. less clerical staff needed for paperwork) and not on the improvement (if any) on the manager's performance. Miles Roman Ltd.,<sup>49</sup> in a survey of 2145 applications, indicate the danger of this 'lack of scientific audit of computer systems in favour of more conventional measures, for example, a relation to manual systems.' However, assessing the rather intangible benefits of 'better' decisions is a very difficult task especially as factors change during project development. P. Clout<sup>27</sup> has stated, for example, the problem of distinguishing between the effects of a new system or a new vice president. The problems of cost justification are obviously then very severe, and as Miles Roman Ltd's

survey showed 'there is a distinct lack of management techniques available for this purpose.'

#### 6.6 Fear of the machine.

As P. Clood<sup>27</sup> has noted in a survey of management information systems in the U.S., even when the system has been proven possible and cost justified, there is still no guarantee that the manager will use it. One reason for this lack of use by managers could be the inconvenience caused by the necessity to communicate with the machine via programmers, analysts and operators. On-line systems giving immediate access to the computer via some device such as a terminal enable managers to communicate directly.

There is no doubt that many managers have a fear of the computer which may have originated from the early 'electronic brain' image and the claims that the machine would 'take over' their role. G. Clarkson<sup>50</sup> recommends senior executives to use the machine via terminals as a decision-making aid when it would help to remove 'both the mystery and threat.' Allen<sup>5</sup> too, in a survey of 100 U.S. companies using time-sharing computers, advocates direct use for the manager leading to 'new ideas, insights and a clearer understanding of the relevant problem variables and their inter-relationships than would otherwise be possible through the use of conventional methods.'

#### 6.7 Staff Problems.

Staff problems are one of the most serious causes of dissatisfaction of computer systems among managers. Miles Roman<sup>49</sup>, in their survey discovered that all the major difficulties concerned staff problems. The first problem in setting up any computer system is the shortage of experienced computer staff. The nature of the computer industry is such that systems analysts,

programmers etc. benefit greatly from having varied experience and this, coupled with the higher salaries available, causes a very high turnover rate in computer staff, although there are indications, at the time of writing, that the industry is becoming more stable in this respect. On-line systems have even greater staff problems as the number of experienced people in this field are fewer still. The results of a survey of non-computer users show that even when time-sharing is used on a bureau basis the need for trained personnel will still not have been solved (see Popell<sup>48</sup>.)

However, these staff problems are not only related to computer staff. There are many obstacles caused by the training of user staff, their resistance to change and, in particular, changes in work structure and content. On-line applications of type 2, with their tendency to involve several departments, can cause tremendous problems concerning this latter point.

#### 6.8 Hardware Problems.

The most serious problem concerning the computer equipment itself is that of reliability. Allen<sup>5</sup>, in a survey of 100 time-sharing systems found that the failure of equipment was the most frequently quoted criticism of time-sharing. Popell<sup>48</sup> supports this in a survey of both computer users and non-users. With the latter group the lateness of reports induced by failure was quoted as one of the greatest deterrents to using a time-sharing system. As with many computer problems, hardware reliability becomes even more critical with on-line systems. This is because a fault in either the computer equipment, the communications link or the input-output device would be enough to prevent the manager from using the system. In many applications where speed is essential breakdowns of this nature can be critical.

Other hardware problems are a result of the nature of the computer industry itself. The many different computer manufacturers now produce a vast range of equipment much of which is incompatible. This incompatibility of manufacturer's hardware was quoted as a major disadvantage of computer systems by groups of computer users, non-users and management consultants in the surveys conducted by Popell et al<sup>48</sup>. In on-line systems, incompatibility can prevent users from linking up with other systems, i.e. a manager of an engineering division of a company may be unable to access research data at another company branch. Finally, the speedy developments in computer technology are also responsible for another serious cause of management dissatisfaction, namely the rapid obsolescence of hardware. In many installations, the manager has no sooner had equipment installed (probably with late delivery dates) when he is being persuaded to invest in some more efficient technique.

#### 6.9 Lack of 'know-how'.

One of the most obvious conclusions that can be drawn from the literature is that there is an inefficient use of most computer systems. Information from 48 non-computer users<sup>48</sup> indicates that this could be due to the shortage of information or assistance available to firms i.e. a lack of 'know-how'. This is particularly true of small firms which often do not possess any experienced computer personnel and have to rely solely on outside help. Surprisingly few of the firms in this predicament have resorted to using some of the countless software houses available - as Miles Roman Ltd.<sup>49</sup> discovered, there is an under-utilisation of professional software resources (it is interesting to note that this company has subsequently gone into liquidation!)

#### 6.10 Summary.

The important point for success in the more complex applications of type 2 would appear to be that managers should treat these applications with respect,



i.e. they should realise the complexity of these systems and not be content to hand over the project to some technical group who lack the skills to cope with the task. Management should be prepared to play an active part in the development stages of the system and should, where possible, use the computer directly. Applications should be developed gradually to allow the users to gain confidence and provision should be made for possible extension and integration. Careful consideration should be given to the organisational problems involved in these integrated systems, particularly the effects on the user in changes in work structure and content. Finally, managers should not expect an immediate return on investment but should be prepared to give applications a reasonable period in which to prove profitable, still, however, keeping a tight control over the costs involved and ensuring that any justifications are measured on the basis of improved performance.

In order to determine further information on the problems of applications of type 2, particularly in relation to the present state of on-line computing in companies in the United Kingdom, a survey of companies using time-sharing in their business was conducted. This survey was also used to gain information on the benefits of time-sharing in business. The following 2 chapters deal with this survey.

## 7. A SURVEY OF ON-LINE COMPUTING - THE APPLICATIONS

### 7.0 Introduction.

In an effort to obtain more information on the advantages and disadvantages of on-line computing applications of type 2 in business, particularly in the United Kingdom, a survey of companies using time-sharing was conducted. A discussion of these advantages and disadvantages is not possible, however, without first describing the applications themselves, (as revealed by the survey) and this chapter deals with those applications. Because of the nature of the research project (DUBS objectives are concerned with aiding management especially in the North-East region), the survey was restricted to firms within the North of England. From a total of 27 companies possessing on-line computing facilities, 15 replied favourably to an introductory circular about the survey (this circular is shown in Appendix 8). The 15 firms were classed as follows:

Manufacturing	5
Local Government	1
Financial	1
Distribution	1
Research & Engineering	6
<u>Mining</u>	<u>1</u>
Total	15

### 7.1 Procedure.

Initially a questionnaire was designed based on the problems discussed

in previous chapters. A separate set of questions was produced for each department within the company and for each terminal user (see Appendix 8). However, this procedure was soon abandoned when the variations in the number of departments and individuals using the computer systems became apparent (in many cases only one department used the on-line systems and several companies did not have a 'computer department' as such). Because of these problems and the desire not to inconvenience too much the firms co-operating in the survey, interviews were restricted to one person (who generally was responsible for the company's on-line computing efforts) and 1 or 2 actual users where this was possible. Furthermore, by keeping the interviews on a free discussion basis, a wider range of views could be obtained than would be possible if the manager was restricted to answering pre-set questions. However, the original questionnaire was used as a check list in these interviews.

## 7.2 Facilities.

The first noticeable feature concerning the on-line facilities of the firms taking part in the survey was that only 4 possessed their own time-sharing computers, even though all the 15 companies were large concerns. The other 11 firms were using on-line computing on a bureau basis although 2 had plans for their own systems in the very near future. This lack of commitment to in-house time-sharing was further stressed by the fact that 9 of the firms possessed only 1 terminal linked to the service bureau. Even more surprising was that all of the 11 companies using bureaux possessed their own batch computers. A discussion of the reasons for installing terminals will explain these facts and this is done in the following section.

### 7.3 Why on-line?

The most popular reason for installing a terminal was that the firm's in-house batch computer was overloaded. This factor was quoted by 7 of the 15 firms. Overloading of the batch computer was particularly detrimental to applications in research and engineering departments, 5 of the 7 companies obtaining terminals for these areas because the in-house machine gave priority to the more profitable commercial applications.

Three companies used terminals to provide remote computing power at various locations while one firm was particularly interested in this 'remote computing' aspect of time-sharing as it provided a means of having computer power without investing strongly in computers. Another organisation was attracted by the fact that the terminal offered the prospect of accessing a much larger computer than the in-house machine.

Two companies making extensive use of service bureaux demonstrated the 'bargaining power' that could be obtained with these systems. The first company had initially connected to a service bureau in order to use a standard package which it supplied because the task of programming was too expensive and time-consuming. As applications developed, the company began to look elsewhere for more efficient packages until they were eventually using the services of 8 different time-sharing bureaux. The result was that all of the 8 bureaux were constantly competing with each other and thus providing the company with a much improved service. If the manager using a package was not satisfied, he was able to use the resultant 'bargaining power' to persuade the bureau to improve the program. A more interesting example of this 'bargaining power' concerns a large association which provides a computer service for its members throughout

the country. The association developed packages for general use and stored the resultant library of programs with several different time-sharing bureaux thus giving each member-firm a choice of services when they wished to use a package. Each bureau offered a free service to the association in return for possible custom from member-firms and other users wishing to access the packages. The association even used a U.S. service bureau which provided both cheaper alternatives and much greater computer power.

Finally, another interesting use of time-sharing bureau was demonstrated by a research organisation in the development of a program for a manufacturing branch. The problem here was that the computers possessed by both concerns were different models and to write the program for the branch machine would involve several lengthy journeys to the manufacturing branch. A neat solution was to link to a service bureau possessing the same computer model and develop the program thus requiring a straight swap when completed.

#### 7.4 Applications.

##### 7.4.0 Classification.

Chapter 2 divided business applications of on-line computing into 2 types - applications relying on the conversational facilities of time-sharing and applications relying on the information storage characteristics of on-line computing. To facilitate discussion of the on-line computing applications within the companies in the survey, the applications of type 2 will now be further sub-divided into 4 classes: -

##### CLASS 1

This class involved packages which performed calculations based on

data bases which were generally relatively small.

#### CLASS 2

The second class of applications of type 2 discovered in the survey were those which used some form of management science technique.

#### CLASS 3

The third class of applications concerns those in which the terminal was used as a communications device, very often involving large data banks.

#### CLASS 4

The final class of applications involve those in which the terminal was used purely to update and query data bases.

#### 7.4.1 Applications of type 1.

All of the 15 firms had applications of this kind and one manufacturing company used the terminal for this purpose alone, even performing very simple calculations on-line because of the 100% accuracy that could be achieved. The application of this class that most frequently occurred was for the solution of statistical problems using packages. Some of the more specific uses were:

Statistical analysis of certain test results.

Calculating the costs of apprenticeships.

Bonus calculations.

Calculating the percentage scrap involved in a manufacturing process.

Statistical analysis of faults and long-term defect analysis.

Design calculations for mechanical and heat-transfer problems.

Stress analysis of pipes laid underground.

Numerical analysis.

D.C.F. and N.P.V. packages.

Calculating method study standards.

Surveying calculations.

Regression and correlation analysis.

Quality analysis.

#### 7.4.2 Applications of type 2.

##### CLASS 1

Unlike applications of type 1, applications of this type take advantage of the information storage capabilities of on-line computer systems. Examples of this kind of application are:

##### (i) Forecasting techniques

Forecasts are made using data from an information file which is updated regularly on-line. Thus a manufacturing company used statistical techniques such as regression and correlation analysis to produce very short control forecasts on how much is produced, whilst a mining concern updated weekly information concerning progress and used the resultant data base to produce monthly forecasts of stock required.

##### (ii) Quality control

There were 2 examples illustrating how on-line computing can help in decisions involving the quality of a product. One company used the file interrogation capabilities of on-line computing in determining product reliability by analysing a customer complaints file. The second example involved the marketing department of a company which used the terminal

to determine the affect of product quality on custom.

(iii) Complex design problems

The problems referred to here are large scale design problems involving the interaction of many variables. On-line computing is ideal for solving these problems because a data base can be used to store the values of variables so that the user need only input those factors whose value has changed. In addition, the data base can be used to hold results from previous runs or information on various design parts which can then be 'called up' at the design stage. The following 3 examples of this type of application were discovered in the survey:

(A) A shipbuilding firm was often required to produce tenders for prospective customers within one week. However, the many thousands of parts used in the building of a ship made the process of designing and estimating costs a very lengthy one. Without the aid of a computer only rough estimates, often very inaccurate, could be given. Even the batch computer system, because of its unreliability, could not guarantee the turnaround necessary and the tender was therefore prepared manually - a very tedious task. However by using a storage tube with a joystick providing a vector facility, the user could draw up the design of a hull and estimate storage requirements and costs. They also found that the device was sufficiently accurate and speedy enough to allow finer details of design to be investigated within the time available.

(B) A manufacturing firm used the terminal to calculate furnace blends from a number of different inputs. The package also calculated the costs involved in the process, breaking them down into their various parts, (i.e. costs of stock input). The data base in this case was used to hold



the inputs from the previous run and only those variables whose values had changed were input for future runs. On-line computing was essential for making speedy decisions when furnace conditions or stocks changed.

(C) The same manufacturing firm mentioned in (B) above was also faced with the difficulty of designing the furnaces which required many different types of brick. The problem here was that the bricks had to be ordered up to 18 months in advance of the actual building of the furnace. By using a terminal in the design stage, it was possible to have the order prepared in time. Information about the various types of brick was kept on a data base and used at the design stage.

(iv) Design-production links

There were 2 examples of applications in which the design and production stages were linked using the terminal. The first example concerns the use of the storage tube and joystick in the design of ships described earlier. A program was developed which generated paper tapes from the final displayed drawings. These tapes were then used to operate a drawing machine and hence the drafting function was eliminated. The second example arose in an engineering firm and was similar to the ship design application in that a paper tape was produced from input information concerning a drawing (i.e. tools, speeds etc.) In this case, however, the output tape produced was used to control the machine which actually manufactured the equipment. The use of a terminal reduced the time to manufacture the equipment from 3 days to about 12 hours.

(v) Inventory control

Only one company used their on-line computing for controlling their stock levels. The greatest advantage of being on-line was that frequent

and accurate lists of stock could be obtained.

## CLASS 2

Applications of class 2 used the following management science techniques:

### (i) Simulation.

This was used in 5 of the 15 firms taking part in the survey. However in only 2 of these cases was the model actually run from the terminal and used for immediate decision-making and policy testing. The major use of the terminal in simulation was in the development of the model and for updating purposes, the simulation run itself being performed off line. Some specific models were:-

(a) A simulation model of a reactor was updated on-line and used for operational decision-making and for emergency situations requiring immediate action.

(b) A manufacturing company used simulation to test out policies before implementing systems. The model itself was built from a set of standard packages and was used to help people think through a project - 'people never have a very clear objective but, by using the terminal, they can clarify what they mean.'

(c) A research establishment created a model of a large supply system to use in emergencies for making alternative plans. However the greatest benefits gained from the model were in the training of staff in both the theory of the system and computing itself. Initially subjects were very frightened of using the model via a terminal but after just 2 days they became quite efficient.

### (ii) Resource allocation.

Three firms used some form of resource allocation in their on-line

computing applications. One manufacturing firm used it to schedule manpower requirements in a production process involving many parts whilst another manufacturing concern used it to schedule maintenance work and to evaluate usage, inventory and plant reliability, thus obtaining information for future policies. A rather interesting use of resource allocation arose in a research organisation who used it to schedule the work load for the computer itself, the results making an impressive contribution to the efficiency of the system.

(iii) PERT and CPM.

There were 4 examples in which a terminal was used with PERT or CPM models. In one of these cases the terminal was only used for building and testing the model and for quick updating of files. However, an engineering concern had found the terminal provided a 'clear selective output' using a report-editing procedure - a vast improvement on the confused output from the batch model. A manufacturing firm had successfully used PERT and CPM for the project control of small O and M projects which were complicated because they involved many different departments. In addition, the terminal was used for network analysis of complicated assembly work. In the case of a research department however, the use of CPM was not successful because the schedules fell so far behind that the model eventually collapsed.

(iv) Business games.

Two companies used on-line computing for business games and found that they were very good for training staff both in company methods and computing. One national concern even had users at widely separated locations playing each other and had succeeded in simulating other competitors.

### CLASS 3

Examples of applications in which the computer terminals were used as communications devices are:-

(i) A manufacturing company had an on-line computing system linking 3 widely separated sites. A monitoring system was established by setting up a common data base containing the status of all projects. The file was updated whenever changes were made and people could therefore access information which was completely up-to-date. In the original communications system (using telephone, post etc), everyone had different ideas about the status of projects and there arose a confusion of errors. However, with the on-line computer system the data base was continually updated and any errors that were made were at least consistent.

(ii) The same manufacturing firm also used their on-line computing system as a data link rather like a normal telex facility. This was particularly advantageous for the processing of engineering information. A file was established which contained all changes made to drawings etc. and this could then be monitored.

(iii) A large financial concern had also developed a nationwide communications system using on-line terminals. A trained clerk at each branch entered transactions at regular intervals throughout the day, thus keeping the central files constantly up-to-date. A series of regional concentrator computers helped to cut down G.P.O. line costs to the central system and, by holding summary files at these 'concentrators', each branch was able to answer most customer requests immediately.

(iv) Finally, a distribution company used a terminal as a communications device in a computer controlled automated distribution system.

Checks were incorporated into the various electronic devices within the system (cranes, elevators etc.) and any faults were reported to the type-writer terminal. The terminal could then be used to find out more information about these faults and to send commands to stop activities or initiate others.

#### CLASS 4

There were 4 examples in which the terminal was used to make enquiries of and amendments to large data files. Three of these cases were in a local government concern who had developed the following on-line systems.

##### A. Housing accommodation system.

The data base in this application consisted of information about housing waiting lists. A terminal, installed in each local rent office, enabled clerks to make decisions on rehousing with attention paid to priorities such as medical cases, slum areas etc. The great advantage of this system was that one could look for complex chain transfers with the aid of the computer. In the previous system it was impossible to perform 3 transfers in one move.

##### B. Maintenance system.

By extending the data base mentioned above to include information concerning repairs etc, the system could easily allow maintenance to be handled.

##### C. Vaccination system.

In this case the data bank consisted of information on children at birth including past and future vaccination dates. The computer system automatically produced requests for vaccinations and sent the necessary cards to doctors, nurses etc. The terminal enabled immediate replies to be

obtained to a constant stream of requests for urgent medical information from hospitals and surgeries.

The 4th and final example of this class of application occurred in a large financial organisation where a terminal was used by the personnel department to make amendments to, and queries of, a data base containing staff records.

### 7.5 Conclusions.

The survey shows that the application of on-line computing, at least in the North of England, has not reached a very sophisticated level and few companies possess extensive time-sharing facilities. As section 7.3. shows, the decisions to use on-line computing by the firms in the survey were not motivated by the benefits described in Chapter 2 (immediate information, exploration of more alternatives in decision-making, stimulative interaction etc.), but more by the ability to tap remote computer power. In particular, the use of time-sharing bureaux, enabling companies to pay for just the actual computer resources used, persuaded many firms to use on-line computing to relieve the load on their own batch systems - even developing applications which would not normally be considered suitable for on-line use (heavy computational jobs requiring large amounts of input-output). It might be expected, therefore, that the advantages and disadvantages of these on-line computing applications will be closely related to the use of time-sharing on a bureau basis and these points are discussed in the next chapter.

## 8. A SURVEY OF ON-LINE COMPUTING - THE BENEFITS AND PROBLEMS

### 8.0 Introduction.

This chapter discusses the advantages and disadvantages of the on-line computing applications described in chapter 7. Several other interesting points arose out of the interviews in the 15 companies and these are also discussed. In the conclusions those factors which were a result of the nature of applications of type 2 are isolated and discussed.

Since most of the firms in the survey had a heavy dependence on standard library packages supplied by time-sharing bureaux, it is necessary to distinguish those factors which are a result of the characteristics of these library programs before entering into a more general discussion of the advantages and disadvantages of on-line computing as revealed by the survey.

### 8.1 The Role of Standard Library Packages.

Standard library packages consist of generalised programs supplied by the time-sharing bureau for use by many different customers. The survey revealed a very high usage of these packages, 7 firms applying them to scientific, technical and statistical problems and 3 firms making extensive use of such programs.

The time and expense of programming was the most common reason for resorting to standard packages. Thus a manufacturing firm was forced to use packages, even though they did not do quite what the user required, because there were no programmers available and users themselves did not have the time to write their own software. Although the majority of users interviewed were extremely dissatisfied with the packages supplied by the

bureau, they still felt that it was a better alternative than doing their own programming. This was particularly true of 'one-off' applications where the investment in programming could not be repaid by any repetitive use. In these cases, companies would invest considerable time and effort to secure standard packages. Thus a manufacturing company had a policy of approaching all service bureaux for a package before attempting to write their own program. One firm even linked to a United States bureau by satellite to use a highly complex, technical package - the only way in which the results could be obtained in time. A research company had found that using standard technical packages enabled them to compare methods of solution with their own techniques.

Some of the disadvantages of using these standard packages were:

(i) Explanation techniques were very poor. Many packages did not take advantage of the conversational mode of time-sharing thus requiring the user to search through a manual for instructions. The manuals themselves were difficult to follow being 'jargonised' and lacking worked examples. Typical comments were 'manuals need programmers to understand them', 'packages are written for computer people and try to be too clever,' and 'many create confusion by putting different meanings on words.' The general view was that the packages were necessary but the effort involved in finding out what a program does and how to use it was too great, e.g. with an L.P. package, the biggest difficulty in using it was discovering how to feed in the matrix.

(ii) Library packages often proved inefficient and expensive. This was mainly a result of their generality - they very often did not do quite what the user wanted and also gave additional information which the user did not require but nevertheless had to pay for.

(iii) Lack of flexibility was a major cause of dissatisfaction in using standard packages. Several firms expressed a desire for a building block approach to provide some flexibility but only one of the time-sharing



bureaux appeared to adopt this approach.

(iv) Packages gave too much irrelevant output and not enough exception information.

(v) Finally, problems arose not from the packages themselves but from the manner in which they were used. Many users had a tendency to run the programs 'blindly' and place too much faith in the results (one firm quoted this as a major advantage of standard packages, i.e. the manager does not need to know how the results are produced but only how to apply them). The dangers of mis-interpretation here are obvious.

There are then, obvious shortcomings in the use of library packages. However, unless the organisation is very large and the applications are needed in several branches enabling the programming costs to be shared, the lack of time, money and skills will force companies to use them despite being dissatisfied.

## 8.2 Advantages of on-line computer applications.

### 8.2.0 Introduction.

Each of the 15 firms was asked to describe any advantages they had obtained from using on-line computing. A total of 14 major benefits were stated and these are described in the following sections and summarised in table 8.2.

#### 8.2.1 Speedier information for decisions.

Over half (8) of the companies in the survey quoted this as a major advantage of using terminals. Two firms stressed that, even though speed was not critical with the majority of decisions made, the ability to obtain immediate information when required was important. One financial organisation found that the ability to answer customer enquiries immediately was a crucial factor in improving their customer service.

An interesting example from one company illustrates the advantages,

stemming from immediacy of information, that on-line systems have over batch computing. Before the introduction of computers, enquiries were answered by clerks from a filing system. However, with a 'human memory' system, many answers could be given immediately by the clerks themselves. When the batch computer system replaced this method, immediate replies were no longer possible until the introduction of a system based on computer terminals, which provided immediate access to the data base.

#### 8.2.2 Ability to try more possibilities.

A direct result of the speed with which information can be obtained by using terminals is that more possibilities can be tried in making a decision - a fact supported by 6 of the firms in the survey. This advantage was responsible for encouraging one company to try and help top level management in financial planning.

In enabling more alternatives to be explored, on-line computing can help the manager to make 'better' decisions. Thus a manufacturing firm discovered that the speed of time-sharing enabled them to obtain solutions approaching the critical limits of material used in obtaining furnace blends. Without the ability to explore many possibilities there was a tendency to overcompensate and hence incur extra costs.

#### 8.2.3 Ease of information retrieval and file updating.

The ease with which files could be interrogated and changes made using a device such as a terminal was praised by 5 of the firms in the survey. Indeed, one company initially obtained computer terminals purely for this purpose in order to relieve the load on their batch computer system.

#### 8.2.4 Convenience.

A third of the firms interviewed quoted convenience as a chief advantage of on-line computing. As one manager put it, 'information is literally at your fingertips.' The convenience of using computers via a device such as

a terminal had appealed to a research manager who found that having to communicate through a third party in order to use the in-house computer caused considerable confusion and delay.

#### 8.2.5. Applications previously impossible.

Four companies thought that on-line computing had made many applications possible. Some examples of these applications are:

A. A department of a manufacturing company thought that on-line computing had enabled some aspects of statistical analysis, such as regression analysis, to be performed in time - a task impossible on the batch system.

B. A large financial concern found that the increasing workload was becoming too great due to the amount of output that needed to be generated for each branch. On-line computing had succeeded in cutting down this workload to manageable proportions by providing each branch with a report editing facility.

C. A large corporation discovered that on-line computing had enabled complex chain transfers to be handled when making re-housing decisions.

D. A shipbuilding organisation found that on-line designing using storage tubes was the only way in which tenders could be given to prospective customers in time.

#### 8.2.6. Training staff.

Four companies discovered that the use of computer terminals was a distinct advantage in training staff both in computing, and with the aid of simulation models, business games etc, in the disciplines of business itself. The use of novel games such as 'hangman', 'noughts and crosses' etc. which can be played conversationally were ideal for overcoming the 'awe of computers' that most novices held. One particular manager who came to see the terminal

shortly after its installation became so fascinated 'playing the computer' that he was late for an important board meeting.

One company found that on-line computing had really influenced managers' attitudes to computers. This occurred because, in the absence of skilled staff, managers had to write their own programs and they had thus come to realise exactly what the computer could and could not do. Indeed they had become so keen that the response to learn a new language (introduced with a larger system) was overwhelming and even senior managers wanted to learn it. Terminals were particularly useful for training managers in programming because of the speed of development of programs and the absence of confusion created by characteristics of operating systems such as control languages.

#### 8.2.7. Remote computing.

The ability to 'tap' computer power at remote locations was mentioned by 4 of the 15 firms taking part in the survey. For small organisations on-line computing enabled them to use computers on a bureau basis without investing strongly in them. The remote computing aspect of time-sharing is a distinct advantage when applications are repeated in many different locations as it helps to cut the costs of duplicating systems.

#### 8.2.8. Fast turnaround.

The quicker turnaround time for jobs performed on-line was cited by 4 of the companies as a distinct advantage of using terminals. All 4 cases concerned design applications where the turnaround on the batch computer was too poor due to the priority given to commercial applications.

#### 8.2.9. Speed and ease of programming.

Whilst this factor is one of the major advantages that on-line computer systems have over their batch counterparts, only 3 firms in the survey quoted it as an important benefit. This is not surprising, however, considering



the high proportion of on-line systems of the bureau type and the dependence on standard packages mentioned in earlier sections.

#### 8.2.10. Exception reporting.

The ability of on-line computing to give the manager only the information he wants in output reports was praised by 3 of the firms interviewed. This was a distinct advantage over batch systems where the tendency was to swamp the manager with much useless information in the hope that he would eventually find what he wants. In 2 companies, the terminals were used to produce much clearer and more varied reports for managers. Finally, a financial concern found that, on the batch computer system, each branch had to be supplied with huge amounts of output as any customer might wish to know about his account. With the on-line system, output could be generated on a terminal at each branch only when a customer requested information.

#### 8.2.11. Improved communications.

Three companies stated that the use of on-line computing had reduced errors and faults arising through communication problems. This was done in 2 ways.

- (i) by providing a means of centralisation which reduced the risks of faults occurring and enabled regional sites to be monitored and controlled,
- (ii) by setting up a common data base which helped to produce consistent information unbiased by individual ideas.

#### 8.2.12. Stimulates user.

Only 2 firms felt that conversational computing helped the user's thought process. The first case involved design applications where a high degree of stimulation might be anticipated. However, in the second case, a company

had found that the use of computer terminals had helped many managers to progress into fields previously unthought of.

8.2.13 Facilitates data-input.

Two companies had found that the use of a teletype terminal with paper tape input was ideal for inputting data which could then be analysed on the terminal.

8.2.14 Bureaux advantages.

Finally one company stated that the major advantage of using a terminal on a bureau basis was that it 'lay somewhere between a large computer system and a calculating machine.'

Table 8.2. Advantages of on-line computing applications.

ADVANTAGE	Frequency (Number of Mentions)
Speedy information for decisions	8
Ability to try more alternatives	6
Ease of information retrieval and file updating	5
Convenience	5
Applications previously impossible	4
Training Staff	4
Remote Computing	4
Fast turnaround	4
Speed and ease of programming	3
Exception reporting	3
Improved communications	3
Stimulates user	2
Facilitates data-input	2
Bureaux advantages	1

## 8.3 Disadvantages of on-line computing applications.

### 8.3.0 Introduction.

A total of 12 causes of dissatisfaction with on-line computing applications were discovered in the survey and these are described in the following sections, and summarised in Table 8.3.

### 8.3.1 Costs.

Eleven of the firms taking part in the survey mentioned high costs as being the major obstacle to the efficient use of on-line computing. These costs are obviously very much higher per application when time-sharing is used on a bureau basis - the case with most of the companies in the survey. The main complaint was against secondary storage costs which were prohibitive for many applications involving large amounts of data. One research organisation resorted to the use of tape backup to minimise secondary storage costs. This resulted in delays for users, whilst files were recovered from tape, and became a major source of irritation.

G.P.O. charges often accounted for a considerable proportion of the total computing costs. Many of the firms interviewed used the cheaper, slower terminals which were connected to the normal STD network and therefore subject to standard telephone charges. However, only 2 companies recognised the extent of these transmission charges ( $\frac{1}{3}$  of the total computing costs in one firm before the introduction of a multiplexor), probably because the charges were often hidden with the normal telephone costs.

The main cause, however, of these high on-line computing costs, is the fact that terminal systems lay themselves open to the enthusiastic amateur. Companies need to keep strict control over the activities of each user to prevent them running away with wasted costs. Thus a manufacturing firm found that many users had reached the point where they were not getting what they

wanted from the system and had stopped using the terminals. This resulted in many files getting out of hand and incurring wasted secondary storage costs. To overcome this each user was held responsible for his own files. Another example of the need for strict discipline over terminal users involved a research organisation who found that it could not prevent users from running jobs on the terminal with large amounts of input-output, thereby increasing on-line time and communication costs.

### 8.3.2. Input devices unsatisfactory.

Input devices were the second largest cause of dissatisfaction, 9 firms complaining that they were noisy, slow and inefficient. One firm even used storage tubes as teletypes for certain applications as the noise-level of the teletype was high enough to disrupt thinking. In addition to the noise of these teletypes, the slow line transmission speeds increased the on-line 'connect' time which proved expensive on the bureau system. Two companies complained that the width of output paper on the teletype terminals limited the size and clarity of any tables that could be produced.

All these complaints were against the slow, cheaper form of teletype terminal used in 10 of the firms in the survey. Although there are faster, quieter and more efficient input devices available to overcome these problems, most of the companies found the present costs prohibitive.

### 8.3.3. Lack of experienced computer personnel.

Surprisingly, only 3 firms complained of a shortage of skilled computer staff even though 11 of the 15 firms were using on-line computing on a bureau basis, a situation in which such a shortage might be expected. As stated earlier, most companies coped with standard packages and dissatisfaction



was expressed only when something non-standard was required.

Although only 3 companies mentioned the lack of computer skills as a problem at least 6 of the terminal systems were considerably underused indicating that the problem is much more serious. None of the companies considered using consultants to overcome these problems, the general opinion being that they were expensive. Most firms were reluctant to divulge company information to 'outsiders' without a strong motivation.

#### 8.3.4 Programming difficulties.

Again, a surprisingly low number of users (3) complained about difficulties in programming tasks and this must be attributed to the high number of bureau users and the dependence on standard packages. However those firms who performed a considerable amount of their own programming discovered that a high level of control was needed if success was to be achieved. Thus a manufacturing firm kept a close surveillance on program development and, as soon as things went out of hand the project was either 'bulldozed through' or abandoned. Illustrating the same point, a financial organisation constantly attempted to evaluate programmer performance and insisted that all programmers were kept responsible for correcting and updating their own programs.

#### 8.3.5 Transmission faults.

Three firms complained of on-line interference and the terminal link breaking down, particularly in the early days of installation.

#### 8.3.6 Cost-justification.

The lack of available costing techniques for on-line computer systems was criticised by 3 companies. As a manager in a manufacturing firm stated, the problem is to isolate effects from the many complex interactions of systems

and attribute them to the use of the terminal. One computer manager indicated the dangers of attempting to cost-justify computer applications 'one cannot justify systems unless one gets one's fingers burned or learns from someone who has.'

### 8.3.7. Error recovery messy and difficult.

Errors were considered a serious cause of dissatisfaction in 3 of the companies using terminals, the detection of errors being a major problem. Thus one computer manager described how users were frustrated by the fact that errors in a data file were not detected until the program reached that point, very often necessitating a lengthy re-run. A research organisation providing a bureau service to its member firms found that a large amount of time was spent investigating user complaints that programs were incorrect when, in fact, the user's input data was invalid. Although many errors could be detected by the program, it was impossible to foresee all errors - a fact supporting the conclusions on the experiment described in Chapter 5. As one user complained, however, on those occasions when the bureau program was at fault it was difficult to convince the service bureau who often claimed that their programs were perfect.

Besides being frustrating and time-consuming, errors can be a severe problem in many applications such as the design of ships and motor cars where strict safety regulations are involved.

### 8.3.8. Centralisation problems.

Centralisation problems were limited due to the lack of any large-scale time-sharing systems amongst the companies in the survey, the majority possessing only 1 terminal connected to a service bureau. However 2 companies who were planning to have large time-sharing systems in the future had

encountered resistance from local managers who resented the loss of power even though they could appreciate the benefits of tighter control over regional branches that a central system would offer.

#### 8.3.9 Training problems.

Only 2 firms considered the training of staff to be a problem and, in both cases, the difficulty was in the training of the terminal operator. One company wished to use a 'data-secretary' to operate the terminal for repetitive applications. However the high G.P.O. charges necessitated the use of short, highly coded interaction languages rather than verbose messages and the operator would have to be trained to understand these codes. Another manager thought the man-machine interaction problems such as errors, response times etc., would be too great in training a secretary to use the terminal for report editing. In contrast to these 2 examples, 6 companies had successfully trained a data-secretary who, in many cases, had become very efficient.

#### 8.3.10 Lack of standards.

There were 2 complaints against the lack of any standardisation among the different service bureaux. One company mentioned the absence of standard transmission codes which prevented them linking with other service-bureaux using the same interactive device. The manager of a research department using a terminal complained that changing from one service to another was too difficult because of the effort involved in reformatting tapes, learning new languages etc. and the user was therefore content to remain with a bureau even though there was a better alternative.

#### 8.3.11 System requirements outgrow equipment.

Only one company found that their work load necessitated an ever increasing investment in computers. The remaining 14 firms recognised that early systems were oversold and had become cautious, making progress

in 'inches not strides'.

8.3.12 Loss of control.

One firm discovered that some terminal users felt uneasy because they did not know what actually happened during processing and they preferred, therefore, to use a calculating machine. As stated earlier, however, many managers were only too pleased to accept results blindly without having to understand the process that occurs.

Table 8.3 Disadvantages of on-line computing.

DISADVANTAGE	Frequency (Number of Mentions)
Costs	11
Input devices unsatisfactory	9
Shortage of skilled computer staff	3
Expense and time-consuming nature of programming	3
Transmission failures	3
Difficulty in cost-justifying	3
Errors	3
Centralisation problems	2
Training staff	2
No standardisation	2
Constantly increasing equipment requirements	1
Loss of control	1

8.4 General considerations.

In addition to the advantages and disadvantages discussed in the

previous sections there are several characteristics of on-line computer systems that are worth discussion.

#### A. Degree of Use.

Utilisation of the on-line computer systems varied considerably from one company to another but many systems were very under-used. Even within each company the departmental usage varied considerably. Many firms found that one department or area had made extensive use of the system whilst other areas had failed to take advantage of the facilities offered. This was generally a result of 1 or 2 people in a particular area who were keen users and had spread interest throughout that area.

There is no doubt that the shortage of skilled staff was responsible for many of the on-line systems being under-used. However several companies had obtained a high level of use by successfully encouraging managers to learn computing. By leaving all departments to their own initiative and with some influence from senior management, a reasonable balance of experience was obtained capable of dealing with most problems.

Surprisingly 2 companies mentioned the physical location of the terminal as being an important factor in its use. In one case, the device was situated in the computer room itself creating an effort for managers to reach it and a considerable amount of fear from the computer and peripherals themselves. To overcome this a portable VDU was planned for future use. Another company found that people preferred to use adding machines on their desks rather than make the journey (however short) to the terminal.

#### B. Success of applications.

All the managers interviewed expressed difficulty in gauging the success of the on-line computer applications due to the problems of costing, isolating effects from many complex interactions, etc. The general attitude was that results could not be expected immediately and terminal systems can

take anything from 6 months to 2 years before they begin to justify themselves. The definition of a successful system also proved difficult particularly for larger applications which are general in nature and stretch over several managerial functions. In these instances, the application has to please many different managers. One company using 8 different service bureaux stressed the importance of 'shopping around' before selecting a service as the many different standards can greatly influence the success of the computer system.

#### C. Rate of progress.

Three firms attached great importance to a policy of advancing in 'inches not strides' with computer applications. Managers in each of these firms stressed the fact that many systems fail because early applications are too ambitious (e.g. management information systems embracing whole company or an area), whereas it often proves better to develop a series of small projects which can be linked up eventually (it is important, therefore, that allowance is made for possible future integration).

#### D. Management involvement.

All the managers interviewed agreed that management involvement was crucial to success. Surprisingly, each company was satisfied with the level of management participation in their on-line computer applications. There were 2 ways in which managers were successfully involved in the design of applications:

(i) In the larger on-line computer systems investigated, those managers who were responsible for the applications did not, in general, use the system directly. Great care was taken, however, in ensuring that the system was

specified correctly to the computer staff before the actual programming. In addition, feedback from the programmers and systems analysts to the managers was ensured, allowing changes and additions to be made as the systems were developed. A financial concern was particularly pleased with their system design team which consisted of a successful combination of computer experts and systems and methods people who were non-computer personnel recruited from within the company.

(ii) Many of the on-line computer systems based on service bureaux computers had successfully overcome the problem of involving managers. This was due to the fact that the managers were encouraged to write their own programs and use the terminals themselves - hence the majority of the terminal applications were user-orientated.

However, in these systems the influence of senior management is of vital importance if success is to be achieved. In those systems where managers had become accomplished terminal users, senior management had encouraged users and taken an active interest in the on-line computer system, applying gentle pressure to those departments who fell behind in their anticipated use. Because of the difficulty in justifying on-line computer applications, many companies had a policy that senior management should be prepared to allow any project to be attempted but be quick to abandon those applications which begin to look uneconomic.

#### E. Communication problems.

One of the greatest obstacles to a successful computer application is the difficulty that managers and computer staff have in communicating with each other effectively. On the one hand the manager knows a great deal about business problems and little of computers and their capabilities, whilst the systems analyst generally lacks an understanding of management problems.

Managers in 4 companies mentioned this communication barrier as a problem in the use of their on-line computer system.

The computer staff claimed that managers only had general ideas about what they wanted and could not specify their problems exactly. One computer manager had become particularly frustrated in attempting to help his managers who had complained of too much useless information from the computer. His requests to managers for a clearer definition of their information needs were refused because of a fear that the computer would eventually replace the managers concerned. One person had even stated at a board meeting that 'the computer isn't taking over my job'.

Terminal systems can help to dispel this kind of mis-informed attitude provided computer staff encourage managers to interact directly with the computer and thus help them to become aware of the computer's potential and its limitations. However the use of coded interaction languages and computer jargon must be avoided if the manager's hostility and fear that the information he is given is not what he really wants is to be overcome.

A good example of how to overcome these communication problems was demonstrated by a large organisation where the computer was used as a bureau by each department. All the departments using the machine had a computer liaison officer who was familiar with the business of that department. It was the task of this liaison officer to sort out any communication problems by asking pertinent questions of the computer staff as to how information was obtained, i.e. 'did we get this result using last weeks figures?'

#### F. Decision models.

Four of the managers interviewed stressed that, if computerised decision models are to be successful, then it is important that the manager



is allowed to make the final judgement. They stated that the computer should only be used to supply the alternatives and the manager should be left to use his own skills, hunches etc. in reaching a solution - even if this means disregarding completely the computer's output. Thus one company split up its models into several different stages. At each stage a report was produced which was then analysed by the manager who was allowed to change factors according to his own judgement. The new report was then submitted for further processing until the final report was accepted. Another good example of allowing the manager to use his own initiative in decision-making was given by a local government concern in their computerised re-housing system. In this application, the computer provided the alternatives but left the final decision to the rent officer who was able to apply his own skills, (i.e. personal knowledge of tenant suitability etc.)

#### G. 'The personal touch'.

The importance of having personal contacts in business, as opposed to man-machine interactions, was well illustrated by one of the companies taking part in the survey. The operations research staff of the firm had proposed a method whereby orders sent to stores could be based on a sales analysis of products. By having an on-line recording method for sales, stores would be able to recognise and cash in on, the best selling items. Thus the buying process would be automatic. The system was rejected, however, because the personal touch associated with the salesman - store manager relationship would be lost.

#### H. Report editing.

While many firms recognise the advantages of on-line computing for report editing only one had actually implemented such an application .

Another company had attempted to provide managers with editing facilities but had failed because of 3 factors:-

(i) Simple interrogation, i.e. providing them with a set of questions, was inadequate.

(ii) Sophisticated interrogation involved managers having a knowledge of programming techniques which was undesirable in most cases.

(iii) Exception reporting was only a theory because the ultimate exception conditions could not be stated as there were many factors involved.

#### I. Displays.

Of the 15 firms co-operating in the survey only one used any form of display terminal and this consisted of a simple storage tube with a joystick to provide a vector facility. However, 3 companies planned to use display terminals in the near future. One of these companies was a large corporation who planned to install visual display units (VDU) in each local branch office for use by clerks in their decision-making. A more ambitious application was planned by a manufacturing company who intended to use VDU's at top level board meetings to provide graphs, tables etc. Finally, a research organisation hoped to mount a VDU on a portable trolley to enable staff to use the device whilst sitting at their own desks.

Although all firms recognised the advantages of display terminals for many applications, in particular the drafting function in design, the disadvantages far outweighed any potential benefits. The major deterrent to the use of display terminals was cost. As one manager put it, 'a person would have to be very brave to commit himself to displays when one can get half a dozen terminals for the same price as one display.' Other dissatisfactions concerned poor software support and the absence of a hard copy. Even in the field of design, where the benefits of interactive

displays are most obvious, these factors were great enough to prevent their use.

Some interesting facts were uncovered by a research team set up by a shipbuilding concern especially to investigate the use of interactive computer graphics in the industry. The conclusions were that the expense involved in using fully interactive displays was not justified and a cheaper form of display using an alpha-numeric scope with a joystick, although less efficient (one would have to re-build the picture continuously), would be more suitable. The major disadvantages of fully interactive displays found by the team were:

(i) Cost - a display would cost £10,000 compared to £600 for a teletype terminal.

(ii) The device required either a large, fast computer or a linked satellite computer to enable a high level of interaction to take place. The number of applications where people required this degree of interaction were minimal and did not justify the expense.

(iii) The display screen itself was too small for large scale design of this nature. In design, the user wishes to investigate small parts in detail and then look at the overall effect. In shipbuilding and other large scale problems, the variations involved in this 'blow-up' process are so great that the design stage would take too long. With the proposed storage tube system, the company were even considering screens 6 feet square.

(iv) In a trial and error process such as design, the user would be 'tying up' either a satellite computer or a large part of a main computer system thus making the task costly. The same task could be performed

much cheaper on a batch system or on a less interactive system with a bit of thought and effort.

(v) One application in design involves 'calling down' parts and building up a picture. For large jobs such as shipbuilding, the number of parts concerned is so great that the secondary storage costs become severe.

#### J. The terminal users.

It is interesting to discuss the degree of direct management use of the interactive devices available in the on-line computer systems investigated. Only 5 companies did not use any form of 'data-secretary', the managers interacting directly with the computer. However, 4 of these cases concerned research work of a highly technical nature for which a secretary would be unsuitable.

The most popular method was to use a 'data-secretary' for simple repetitive jobs, particularly when large amounts of input-output were involved, with the manager interacting directly for 'one-off' type applications. Seven firms adopted this approach and, in most cases, the secretary had developed into quite a sophisticated programmer.

Some typical uses of a 'data-secretary' were:-

- (i) Inputting standard information and interrogating files;
- (ii) using simpler statistical packages;
- (iii) in an automated distribution system, where the terminal was used as an exception reporting device, the 'data-secretary' handled simple faults and only called the manager when decisions were required;
- (iv) a clerk in an engineering firm was given the task of operating a critical path model which produced large amounts of output.

Surprisingly, 3 companies used a secretary for all applications of the terminal, the latter being regarded by managers very much as a

typewriter and therefore its operation was considered to be a secretarial task.

### 8.5 Conclusion.

The survey, although limited both geographically and in size, supports many of the points discussed in Chapter 6, with respect to applications of type 2 (data storage applications which may extend over several functional areas).

The most important points arising from the survey concerning applications of this type are:-

(a) Costs.

Chapter 6 discusses the costs associated with the company's own time-sharing computer system. The survey shows that these costs restrict such systems to only the very large organisations. The majority of companies used on-line computing on a bureau basis where they only pay for the resources used. However, these costs, particularly secondary storage costs, can also be severe. Applications of type 2, which often require large data bases, can prove particularly expensive in this respect. Because many of these applications extend over several departments and must, therefore, please several managers, it is difficult to gauge their success and justify these costs.

(b) Management involvement.

All the systems where managers were satisfied with the on-line computer applications were those in which senior management had taken an active interest.

(c) Development policy.

It is important to progress in stages and not to be over-ambitious initially.

(d) Decision-models.

The manager must be given the option of rejecting the computer results and rely on his own skills and hunches in decision-making.

(e) Fear of computers.

Many managers had a real fear of the computer. However on-line computing proved effective in overcoming this fear especially through the use of computer games. Before managers can be persuaded to use terminals directly, however, the costs of these devices and their standard will need to be improved. Input-output devices that are presently satisfactory, such as certain types of display, are much too expensive, whilst those devices that are reasonably priced are slow, noisy and resemble normal typewriters so closely that it is no surprise that many managers regard their use as a secretarial task.

The survey reveals that sophisticated, integrated applications of type 2 have not materialised in the region investigated, mainly because of the prohibitive cost of on-line computing on a bureau basis, especially the secondary storage costs of keeping large data bases on-line. However there is evidence to show that applications of this type, using common data bases, can help to produce a consistency of information and an improvement in communications within the company. By having these data bases on-line, managers can have immediate information for decisions and they can explore more alternatives and hence achieve 'better' solutions.

## 9. CONCLUSIONS

### 9.1 Summary.

#### Research into applications of Type 1.

On-line computing can enable managers to use the computational power of the computer without concerning themselves with programming techniques. This can be done through the use of packages which 'converse' with the manager via some interactive device such as a terminal. Chapter 4 shows that if the manager is to use the computer effectively in this way, these packages must be written to enable him to solve his problems simply and efficiently no matter what his level of experience. The experiment described in Chapter 5 shows that, for this interaction to take place efficiently, then the following points should be considered:-

(a) packages similar to package 1 (see Appendix 1), have some advantages over those of package 2 design (see Appendix 1). That is, where packages contain several different levels of verbosity to match the user's experience there are benefits from having these levels in one form thus:

(i) SC - What is the set-up cost?

as opposed to

(ii) What is the set-up cost?

and SC?

In (i) the user does not have to adjust when changing verbosity levels and this helps to minimise confusion arising from a change in verbosity. This confusion can increase both the problem solving time and the error rate.

(b) The frequency of verbosity changes by each subject, indicates that there is no need for an intermediate level on packages similar in

design to package 2.

(c) The combination of a very verbose level and the instruction sheets (Appendix 3) provided enough initial explanation for the user.

(d) The use of free formatted input enables the manager to enter his data efficiently (the decoding process, in this case, did not noticeably effect the response time). However, data input errors can be reduced by:-

(i) Listing the input data in the order and form required by the package.

(ii) Where several items are entered at once, the number of these items should be less than 5. The experiment indicates that the error rate when 5 or 10 items are input together is too high but further work is required to determine the optimum limit, i.e. that which optimises errors and frustration at being 'over-conversational'.

Finally, the survey described in Chapters 7 and 8 shows that many companies use standard library packages for this type of application (statistical packages, DCF and NPV calcs, scientific calculations). These interactive problems need to be considered by the time-sharing bureau when designing these packages - the survey shows that the present packages are very limited in these respects, particularly concerning conversational level and explanation techniques (see section 8.1). However, there are additional limitations to be considered from the user's point of view, i.e. high verbosity levels increase data transfer volumes (and hence G.P.O. charges) and program size (and hence secondary storage costs). These costs are critical when using a time-sharing bureau and a compromise



must be obtained.

### Research into applications of Type 2.

As stated in section 2.3, the objectives of the research in relation to applications of type 2 was to highlight the difficulties to be tackled in designing successful systems of this type. This was done through searching the literature (Chapter 6) and surveying companies using on-line computing in their business (Chapters 7 and 8). The important conclusions drawn from this research are:-

(a) The influence of senior management is crucial to success. They must take an active interest in the development of systems, encourage managers to use them and ensure that all line managers involved are interviewed to determine their information needs.

(b) It is important to determine the benefits of these applications in relation to improved management performance, as far as this can be measured, and not merely in terms of direct savings such as reduced clerical staff. Strict control is required over users and their files if costs are not to become excessive.

(c) Attention must be paid to the organisational problems in applications of this type, which tend to involve several departments. In particular, user considerations involving resistance to change, changes in work structure and content must be planned for. In very large organisations this may involve the use of a human factors specialist to indicate how these problems can be accounted for in system design.

(d) Applications must be developed gradually to allow management to gain confidence in the new techniques and to facilitate organisational changes. Flexibility must be incorporated into applications to allow for

future development, particularly integration with other systems.

(e) On-line computer models must still allow the manager to use his own skills and judgement in decision-making. The manager must be aware of the assumptions in the model and not be content to accept results blindly.

The survey reveals that complex, integrated applications have not materialised, at least in the region investigated, because on-line computing was mainly used on a bureau basis and the costs were prohibitive. However, the survey shows that applications of this type can provide benefits in 2 ways:-

(i) By giving managers immediate access to data files - the manager has speedy information for decisions, can try more alternatives and reach better solutions, perform tasks previously impossible and, through exception reporting, he receives only the information he requires.

(ii) By providing a common data base - this produces consistent and up-to-date information.

## 9.2 The Future.

The research described in this thesis shows that on-line computing has not developed to a stage where many companies possess large, centralised time-sharing systems and managers regularly use consoles at their desks in decision-making. High costs, particularly of primary and secondary storage of on-line information (i.e. information that may be required immediately), have restricted such systems to only the very large organisations. Because of these high costs, time-sharing bureaux are presently playing a major role in on-line computing by supplying companies with remote-access facilities, charging only for the resources utilised. Even companies possessing batch machines find that a bureau can be used in performing tasks beyond the capacity

or capability of their own systems. Time-sharing bureaux enable the user to have immediate access to, and therefore control over, his information (a comfort to many managers using external resources), allowing him to develop his own on-line computing skills whilst still relying on the resources of the bureau (such as standard packages). However, too heavy a dependance on standard packages when using a time-sharing bureau can be dangerous if the company eventually installs its own computer system. This arises because it is often difficult, or even impossible, for the user to transfer the packages to his own system when the change-over occurs. The time-sharing bureau should be used, therefore, to develop the company's own on-line computing skills and programs which can readily be applied to their own systems when this break-off is effected.

Future developments should affect on-line computing in business in 2 ways:-

(a) Cheaper and more efficient computer storage will help to make on-line computer systems more feasible for large organisations and enable more companies to benefit from time-sharing. Secondary storage units such as discs and drums are already becoming cheaper per unit of information and are capable of holding more data. Smaller businesses can continue to gain the benefits of on-line computing through the use of a time-sharing bureau as this will prove more economical for their computer needs. Present trends indicate that many time-sharing bureaux will tend to specialise in specific applications or specific businesses where they have already demonstrated a particular skill. This should enable small firms to select a service more appropriate to their business needs.

(b) Cheaper and more sophisticated integrated circuitry will give

added power to input-output devices such as VDU's and make them as popular as the present teletypes without many of the associated problems.

Improvements in the design of terminals should encourage managers to use them directly rather than through a data secretary. Advances in electronic circuitry are presently enabling many different types of I/O devices to be connected to a single central computer. Thus the problems of standardisation and incompatibility should disappear and the user will have a wide range of peripheral equipment to select from rather than be confined to the compatible range supplied by the central computer manufacturer.

If on-line computing develops in these ways, then the points discussed in 9.1 concerning applications of type 1 and type 2 will become even more important for success. The growth in the number and size of time-sharing systems within companies will deepen the problems of management involvement, organisational problems, cost-justification etc, whilst the wider range of input-output devices compatible with any one system will necessitate further work on the interface problems - considering the ergonomic design of the input-output device, the software for interaction and user requirements in relation to system limitations.

## BIBLIOGRAPHY

1. GLAUTHIER           Computers and Automation, 1967.
2. WILKES, M.V.       Time-sharing computer systems, Macdonald, 1968.
3. GREENLEES, M.     Time-sharing computers in business,  
Cost and Management, July/August 1969.
4. PEARCE, R.K.      Data entry techniques, The Commercial  
Accountant, July 1970.
5. ALLEN              Time-sharing takes off, Harvard Business  
Review, March - April 1969.
6. COX, D. &  
GOOD, R.            How to build a marketing information system,  
Harvard Business Review, May - June 1967.
7. SCHILLING, D.     Time-sharing/Batch processing, Data Processing,  
May - June 1970.
8. BOOZ,  
HAMILTON INC.       The computers role in manufacturing industry,  
Computers and Automation, December 1966.
9. MCKINSEY & CO.    Unlocking the computers profit potential,  
Computers and Automation, December 1966.
10. SKANDERA, R.     Process control & interactive computing,  
Management International Review, No. 6, 1969.
11. ARGENTI, J. &  
POPE, C.            Glossary of management techniques,  
Management Today, P.78, September 1966.
12. NANUS, B.         The management uses of simulation,  
Data Processing, 7, P.65, 1965.
13. MILLER, J.        Computer graphics for decision-making,  
Harvard Business Review, November - December 1969.
14. DRUCKER, P.      Managing the computer, Management Today,  
January 1967.
15. BOULDEN & BUFFA   Corporate models - on-line, real-time systems,  
Harvard Business Review, July - August 1970.
16. TATHAM, L.        Computing for the Courage Group, Data Processing,  
9, P. 184, 1967.
17. ORR, D.F.         The changing face of real-time, Data Systems,  
P.50, March 1969.

18. ATKINS, W. Real-time stock control systems, Data Processing, 10, P.126, 1968.
19. CAMINAR, D. The Computer Age, Management Today, 1969.
20. INGHAM, R. Automation & the Medium Sized Company, Data Systems, April 1968.
21. ARGENTI The Computer Age, Management Today, 1969.
22. WARNER, B. Reducing the capital risk, The Computer Age, Management Today, 1969.
23. MORTON M, & McCOSH, A. Terminal costing for better decisions, Harvard Business Review, May-June 1968.
24. TAYLOR, R. The Computer in Mainstream Management, Data Systems, P.28, July 1968.
25. DASH, J. & BERENSON, C. Techniques in marketing research, Harvard Business Review, September - October 1969.
26. KANNAN, A. The use of display terminals for business applications, Computers & Automation, April 1970.
27. CLOOT, P. The unhelpful computer, Management Today, P.105, February 1970.
28. WALLACE A survey of data communications and costs etc, Data Communications, May 1969.
29. GROVES, H. The computer in research & development, The Computer Age, Management Today 1969.
30. WINSBURY, R. The designing computer, Management Today, January 1968.
31. EDWARDS, J.A. Designing car bodies by computer, Data Processing, 7, P.171, 1965.
32. CARBONELL, ELKIND & NICKERSON. On the psychological importance of time in a time-sharing system, Human Factors, 10, P. 135, 1968.
33. SACKMAN Computers, Systems Science & Evolving Society, Wiley & Sons Inc., New York, 1967.
34. SHACKEL, B. Multi-access systems - some needs for human factors studies, EMI Electronics report, IMP-F1-2-2. 1965.

35. EASON & CORNEY                    The evaluation of an interactive management information system, E.M.I. Electronics Ergonomics Memo No. 275, 1972.
36. MILLER                            Response time in man-computer conversational transactions, AFIPS Conference Proceedings 33, 1968.
37. YNTEMA, MORFIELD, WILSON & GROSSEBERG                    Effect of response delay on on-line problem solving, International Symposium on man-machine systems, September 1969.
38. CORNEY & TAYLOR                            Experiment in input-formatting, EMI Electronics Ergonomics Memo No. 268, 1970.
39. CARBONELL, ELKIND & NICKERSON                        Human factors and the design of time-sharing computer systems, Human Factors, 10, 1968.
40. DEININGER, BILLINGTON & MICHAELS                        IEE Trans Man-machine Systems MMS-9-10-14, 1968.
41. ROOT & SADACCA                    Man-computer communication techniques - 2 experiments, Human Factors, 9, 1967.
42. CHASEN & SEITZ                    On-line systems & man-computer graphics, Computers & Automation, 1967.
43. NATIONAL PHYSICAL LAB.                    Say 1100011, Data Systems, June, 1970.
44. BERKELEY                        The personality of the interactive programmed computer, Computers & Automation, December 1969.
45. SACKMAN                        Time-sharing v batch processing, the experimental evidence, AFIPS Conference Proceedings, 1968.
46. MAGEE-BOODMAN                    Production planning and inventory control, McGraw-Hill, 1967.
47. HARBRIDGE & SHRIMPTON                    How to maximise computers, Management Today, P.35, August 1970.
48. POPELL, S.D.                    Computer time-sharing, Prentice Hall, 1966.
49. MILES ROMAN LTD.                Computer Survey, 1971.
50. CLARKSON, G.                    How to master the machine, Management Today Annual Review, P.36, 1969.

1 Illustration of use of Package 2

er identifies himself with system.

S (LA31-0032)

signon brt9 pw=john

\*LAST SIGNON WAS: 10:10.54 08-27-71

USER "BRT9" SIGNED ON AT 10:50.24 ON 08-27-71

Subject types in command to run Package 2.

run invent2 sprint=data 6=\*sink\* 7=data

EXECUTION BEGINS

VERSION?

IS IT NORMAL, FINITE PRODUCTION RATE, OR SEVERAL PRODUCTS CASE?

normal

WHAT IS THE VALUE OF PLACING ORDER OR SETUP COST?

0

WHAT ARE THE TOTAL SALES OR ORDERS?

000

WHAT IS THE INVENTORY HOLDING COST PER UNIT?

BQ= 464.7578 NUMBER OF RUNS= 12.9099

DO YOU WISH TO FINISH?

0

VERSION?

CASE?

finite

PLACING ORDER/SETUP COST?

0

TOTAL SALES/ORDERS?

000

INVENTORY COST?

PRODUCTION RATE?

0000

BQ= 489.8977 NUMBER OF RUNS= 12.2475

FINISH?

0

VERSION?

?

several

In this case there are several values per variable and they

are all input together.

.C?

4.4

.P?

.S?

0, 50, 25, 75, 35

.C?

16.5, 6.4, 26.4, 5

.P?

50, 250, 250, 250, 250

= 4

BQ

12.5000

12.5000

6.2500

18.7500

8.7500

?



Subject types in command to run package 1

```
srin invent1 sprint=data 0=*sink* /=data
#EXECUTION BEGINS
LENGTH?
0
CASE?-NORMAL,FINITE PRODUCTION RATE OR SEVERAL PRODUCTS?

normal
S.C?-PLACING ORDER OR SETUP COST?

90
T.S?-TOTAL SALES OR ORDERS?

6250
I.H.C?-INVENTORY HOLDING COST?

5
EBQ= 474.5416 NUMBER OF RUNS= 13.1762
F?-DO YOU WISH TO FINISH?

no
LENGTH?
20
CASE?-NORMAL,FINITE
finite
S.C?-PLACING ORDER 0
90
T.S?-TOTAL SALES OR
6250
I.H.C?-INVENTORY HOL
5
P.R?-RATE OF PRODUCT
62500
EBQ= 500.0000 NUMBER OF RUNS= 12.5000
F?-DO YOU WISH TO FI

no
LENGTH?
5
CASE?
several
N.P?-
5
S.C?-
54.4
T.S?-
50,50,25,75,35
I.H.C
8,16.5,6.4,26.4,5
P.R?-
250,250,250,250,250
NUMBER OF CYCLES= 4
PRODUCT 1 EBQ= 12.5000
PRODUCT 2 EBQ= 12.5000
PRODUCT 3 EBQ= 6.2500
PRODUCT 4 EBQ= 18.7500
PRODUCT 5 EBQ= 8.7500
F?-DO
yes
```

### 10.3 EXAMPLES OF ERRORS AND RECOVERY PROCEDURES

Command to run package 2.

```
#Srun invent2 srint=data 6==sink* 7=data  
#EXECUTION BEGINS  
VERSION?
```

1

IS IT NORMAL, FINITE PRODUCTION RATE, OR SEVERAL PRODUCTS CASE?  
normal

**The subject discovers here that he has entered the wrong reply. To recover, he presses the 'ATTN' key and is returned to the beginning of the package.**

WHAT IS THE VALUE OF PLACING ORDER OR SETUP COST?

'ATTN' key pressed here.

ERROR

VERSION?

1

IS IT NORMAL, FINITE PRODUCTION RATE, OR SEVERAL PRODUCTS CASE?  
finite

WHAT IS THE VALUE OF PLACING ORDER OR SETUP COST?

90

WHAT ARE THE TOTAL SALES OR ORDERS?

6250

WHAT IS THE INVENTORY HOLDING COST PER UNIT?

5

WHAT IS THE RATE OF PRODUCTION?

6250

ERROR

VERSION?

**The subject has caused a divide check by entering the incorrect rate of production. He is merely informed of an error and returned to the beginning of the package.**

2

CASE?

finite

PLACING ORDER/SETUP COST?

90

TOTAL SALES/ORDERS?

6250

INVENTORY COST?

**Here the user has typed '6230' in error and has backspaced and replaced by '6250'.**

5

PRODUCTION RATE?

62500

EBQ= 500.0000 NUMBER OF RUNS= 12.5000

FINISH?

no

VERSION?

normal

**This is an example of a serious error and the user is returned to the system and given a system error message. He must re-submit the run command.**

FCVTH-INVALID CHARACTER IN NUMERIC FIELD

```
#Srun invent2 srint=dat1 6==sink* 7=dat1
```

## 11 APPENDIX 2

### QUESTIONNAIRES USED IN EXPERIMENT

#### 11.1 Questionnaire to ascertain subject's previous experience.

1. Have you had any previous programming experience? Please specify what languages and, if the programs were tested, please describe the machine and the method of input.
  
2. Have you had any previous typing experience? This question refers to familiarity with any keyboard layout, e.g. punchcard machine, paper tape punch, etc. Please specify type of keyboard.
  
3. Have you had any previous computer experience? Mention particularly if you have experienced 'direct' or 'on-line' computer input-output and describe it.

11.2 Questionnaire to obtain subject's final reactions.

Below is a list of word-pairs describing two extremes of an affect of the package on the user. You are asked to choose one of seven points on the scale from one extreme to the other, which more closely affects your feelings, by placing a tick in the appropriate point.

	.	.	.	.	.	.	.	.
Satisfying	.	.	.	.	.	.	.	. Dissatisfying
Boring	.	.	.	.	.	.	.	. Interesting
Patient	.	.	.	.	.	.	.	. Impatient
Annoying	.	.	.	.	.	.	.	. Pleasing
Confusing	.	.	.	.	.	.	.	. Clear
Unfriendly	.	.	.	.	.	.	.	. Friendly
Relaxing	.	.	.	.	.	.	.	. Aggravating
Helpful	.	.	.	.	.	.	.	. Unhelpful
Frustrating	.	.	.	.	.	.	.	. Rewarding
Quick	.	.	.	.	.	.	.	. Slow
Irritation	.	.	.	.	.	.	.	. Calm
Bad	.	.	.	.	.	.	.	. Good
Efficient	.	.	.	.	.	.	.	. Inefficient
Relieving	.	.	.	.	.	.	.	. Anxious
Irrational	.	.	.	.	.	.	.	. Rational

Please answer the following questions:

I think that this type of package is useful for solving business problems:

Agree

Disagree

I would prefer this method to any other method:

Agree

Disagree

I think that this type of package is easy to use:

Agree

Disagree

If you disagree with any of the above, please clarify -

Have you any suggestions which you think would improve the package in any way?

Do you think Managers should use computers in their work?

Yes

No

Do you think 'conversational' computing is needed?

Yes

No

Do you think computers can help managers in their decision-making?

Yes

No

Do you think computers have tended to be misused in business?

Yes

No

Are you afraid of computers?

Yes

No

Do you think computers should be used only by the 'experts'?

Yes

No

## 12 APPENDIX 3

### INSTRUCTIONS AND PROBLEMS GIVEN TO SUBJECTS BEFORE EXPERIMENT

#### 12.1 Introductory Notes.

##### Inventory Package 1

This package will ask the user several questions in solving his inventory problem. The first five characters in each question consists of an abbreviation for the variable that the package requires. The rest of the question is a full description of that variable. After several sessions, the experienced user may find he only needs the abbreviated part of the question and he may find that, if given the whole question, he becomes frustrated. To overcome this, the package has been written so that the user can vary the length of questions he is given. To do this he replies to the question 'LENGTH?', with any value from 1 to 80. This value is the number of characters he wishes to be given in the question. Thus a length of 80 will ensure the whole question, whilst a length of 5 will mean only the abbreviated part. A length of 20 will give the user the abbreviation, followed by one or two keywords explaining the abbreviation. On his first attempt, the user is recommended to use a length of 80 but as soon as he feels he understands the questions he may vary this value.

The object of the exercise is for the user to solve his problems in the shortest time possible and, if any of his answers are incorrect to resolve them.

##### Inventory Package 2

This package will ask the user several questions in solving his inventory problems. These questions are written in 3 different forms - a very verbose supposedly self-explanatory form - a shorter version

containing one or two words and an abbreviated version. The package will allow the user to choose any form he wishes before he solves each problem.

The user is recommended to choose version 1 if it is his first attempt with the package. After he has solved several problems and become familiar with the questions, he may find it becomes frustrating and time-consuming to wait for the computer to type out these verbose questions. If this is the case he should choose version 2 and perhaps later version 3. The object of the exercise is for the user to match his experience with the most suitable verbosity level and to finish solving his problems in the shortest possible time. Please feel free to switch backwards and forwards through the verbosity levels until you find a satisfactory one.

## 12.2 Quantitative Description.

This package is capable of solving 3 different classes of inventory problems.

### 1. Normal

This is the simple case where the economic batch quantity is calculated thus

$$EBQ = \sqrt{\frac{2 A S}{I}}$$

where A = set up cost or the cost of placing an order,

S = total sales or orders,

I = annual cost of holding one unit in inventory.

This assumes 3 facts

1. Withdrawal rate is constant over time.
2. Replenishment is instantaneous.
3. Quantity ordered does not affect other costs and operations.

## 2. Finite production rates

This is the case when a finite length of time is required to produce an order. The EBQ is given by

$$EBQ = \sqrt{\frac{2 A S}{I (1 - S/P)}}$$

where P = rate of production

## 3. Several products

This is the case when production is cycled among several products. The number of cycles required is given by

$$N = \sqrt{\frac{\sum_j S_j I_j (1 - S_j/P_j)}{2A}}$$

where j refers to each product.

The EBQ for any one run is then given by

$$EBQ = \frac{S_j}{N}$$

## 12.3 Operational Instructions.

### Package 1

#### How to 'talk' to the package

The package will first ask the user what length of question he requires. He should reply with the number of characters he wishes to be given. Thus a length of 5 will mean only 5 characters will be displayed to him. This length must be between 1 and 80.

The package will then ask the user what case of inventory he wishes to deal with. He should reply with either 'NORMAL', 'FINITE' or 'SEVERAL'. The package will then request for the values it needs,



to calculate the EBQ. These should be entered in either integer or decimal form. In the 'several' case, where there is more than one value per variable, these should all be entered simultaneously, each value being separated by commas.

e.g. 1  
1.0  
1,2.3124,4

are all valid methods of input.

Finally, when the EBQ has been calculated, the package will ask the user if he wishes to finish to which he replies either 'YES' or 'NO'.

## Package 2

### How to 'Talk' to the package

The package will first ask the user which version he wishes to use. He should reply with either '1', '2', or '3' for the verbose, moderate or abbreviated versions respectively.

The package will then request the user to enter the case of inventory for that particular problem to which the user replies 'NORMAL' 'FINITE' or 'SEVERAL' according to which of the described cases it happens to be.

The package will then ask for the values it requires to calculate the EBQ. The user can enter his input in either integer or decimal form. In the 'several' case, where there is more than one value per variable, the user must enter all values together, in the correct order, separated by commas.

e.g. 1  
1.0  
1, 2, 3.124, 4

but not ~~10~~

NB If the user is using version 3, he need only reply with either 'N', 'F' or 'S' for the case of inventory.

When the EBQ has been calculated, the package will ask the user if he wishes to finish to which he replies either 'YES' or 'NO' or, in the case of version 3, 'Y' or 'N'.

#### 12.4 Errors

If the user discovers he has made an error after pressing the 'RETURN' key, he can recover by pressing the 'ATTN' key. This will inform him of an error and return him to the beginning of that particular problem. Thus if the user discovers an error at the end of his problem and presses the 'ATTN' button, he will have to submit all the data for that problem again.

There are certain things the computer cannot do, e.g. divide by zero etc. Occasionally the user may force the computer to do these things if he mis-types his data. If this happens, the user will be informed of his error and returned to the beginning of that problem. Therefore, if the user receives an error indication and is returned to the beginning of that problem for no apparent reason, he should look for mistyped data.

However, on certain occasions, the user may cause more serious errors by mis-typing his data. When this occurs he will be given some other messages apart from 'ERROR'. This will be followed by a ~~≠~~ sign. When this occurs the user must re-submit the line given on the page on running the package and he will be able to resolve the problem. However, the user must replace the words DATA in this line

by DAT1. If this happens on subsequent occasions then he must use DAT2 etc.

Any other input errors will become apparent when the user checks his answers.

## 12.5 Running the package

### Package 1

To do this the user should type in the following line:

```
§ RUN ␣ INVENT1 ␣ SPRINT = DATA ␣ 6 = *SINK* ␣ 7 = DATA
```

where

␣ = blank which is the horizontal bar at the bottom of the keyboard.

### Package 2

To do this the user should type in the following line:

```
§ RUN ␣ INVENT 2 ␣ SPRINT = DATA ␣ 6 = *SINK* ␣ 7 = DATA
```

where

␣ = blank which is the long bar at the bottom of the keyboard.

## 12.6 Data

TS = Total sales

SC = Setup cost

IHC = Inventory holding cost

PR = Production rate

NP = Number of products

### 1. Normal

SC = 90    TS = 6250    IHC = 5

### 2. Finite

SC = 90    TS = 6250    IHC = 5    PR = 6250

3. Several

SC = 54.4          NP = 5

TS = 50, 50, 25, 75, 35

IHC = 8, 16.5, 6.4, 26.4, 5

PR = 250, 250, 250, 250, 250

4. Normal

SC = 10    TS = 250    IHC = 0.5

5. Finite

SC = 36    TS = 8,000    IHC = 0.4    PR = 200,000

6. Finite

SC = 43    TS = 15,000    IHC = 0.02    PR = 125,000

7. Several

SC = 312    NP = 10

TS	IHC	PR
8,000	.4	200,000
15,000	.02	125,000
7,000	.3	175,000
1,000	.1	50,000
3,000	.35	75,000
10,000	.05	50,000
20,000	.1	225,000
5,000	.15	375,000
12,000	.02	125,000
18,000	.05	1,125,000

8. Normal

SC = 5    TS = 2,000    IHC = 1

9. Finite

$$SC = 15 \quad TS = 3,000 \quad IHC = .35 \quad PR = 75,000$$

10. Finite

$$SC = 40 \quad TS = 5,000 \quad IHC = .15 \quad PR = 37,500$$

12.7 Answers

1.  $EBQ = 474.3416$        $No. Runs = 13.1762$

2.  $EBQ = 500$        $No. Runs = 12.5$

3.  $No. Cycles = 4$

$$EBQ = 12.5, 12.5, 6.25, 18.75, 8.75$$

4.  $EBQ = 100$        $No. Runs = 2.5$

5.  $EBQ = 1224.7449$        $No. Runs = 6.5320$

6.  $EBQ = 8561.2734$        $No. Runs = 1.7521$

7.  $No. Runs = 4$

$$EBQ = 2,000$$

$$3,750$$

$$1,750$$

$$250$$

$$750$$

$$2,500$$

$$5,000$$

$$1,250$$

$$3,000$$

$$4,500$$

8.  $EBQ = 141.4214$        $No. Runs = 14.1421$

9.  $EBQ = 517.5491$        $No. Runs = 5.7966$

10.  $EBQ = 1754.1162$        $No. Runs = 2.8504$

ROUTINES USED IN EXPERIMENT13.1 Fortran Listing of Inventory Package 1

```

BLOCK DATA
COMMON/HOLD/MARK
DATA MARK/0/
END
LOGICAL*1 MES1(80),MES2(80),MES3(80),MES4(80),MES5(80),MES6(80),
1 INPUT(80),MES7(80)
REAL INV
DATA
1MES1/' CASE $\epsilon$ -NORMAL,FINITE PRODUCTION RATE OR SEVERAL PRODUCTS $\epsilon$ ',2
23*' '/,
3MES2/' S.C $\epsilon$ -PLACING ORDER OR SETUP COST $\epsilon$ ',46*' '/,
4MES3/' T.S $\epsilon$ -TOTAL SALES OR ORDERS $\epsilon$ ',52*' '/,
5MES4/' I.H.C $\epsilon$ -INVENTORY HOLDING COST $\epsilon$ ',49*' '/,
6MES5/' P.R $\epsilon$ -RATE OF PRODUCTION $\epsilon$ ',55*' '/,
7MES6/' N.P $\epsilon$ -NUMBER OF PRODUCTS $\epsilon$ ',55*' '/,
8MES7/' F $\epsilon$ -DO YOU WISH TO FINISH $\epsilon$ ',54*' '/
DATA MODE1/'NORM'/,MODE2/'FINI'/,MODE3/'SEVE'/
DATA FIN/'YES '/
DIMENSION VAL(10),S(20),INV(20),P(20),EBQ(20)
CALL INTRP(0)
CALL ERROR
77 CONTINUE
CALL INTRPT
25 WRITE(6,20)
CALL TIME(4,1)
20 FORMAT(' LENGTH $\epsilon$ ')
READ(5,21)(INPUT(I),I=1,10)
CALL TIME(4,1)
21 FORMAT(10A1)
CALL DECOD(INPUT,VAL)
LEN=VAL(1)+1
1 WRITE(6,2)(MES1(I),I=1,LEN)
CALL TIME(4,1)
2 FORMAT(80A1)
READ(5,3) MODE
CALL TIME(4,1)
3 FORMAT(A4)
M=3
IF(MODE.EQ.MODE1)M=0
IF(MODE.EQ.MODE2)M=1
IF(MODE.EQ.MODE3)M=2
IF(M.NE.3)GO TO 4
80 CONTINUE
WRITE(6,5)
CALL TIME(4,1)
5 FORMAT(' ERROR')
GO TO 25
4 N=1
IF(M.NE.2)GO TO 10
CALL IO(MES6,INPUT,LEN)
CALL DECOD(INPUT,VAL)
N=VAL(1)
10 CALL IO(MES2,INPUT,LEN)
CALL DECOD(INPUT,VAL)
A=VAL(1)
CALL IO(MES3,INPUT,LEN)
CALL DECOD(INPUT,VAL)
DO 6 I=1,N

```

```

S(I)=VAL(I)
CALL IO(MES4, INPUT, LEN)
CALL DECOD(INPUT, VAL)
DO 7 I=1, N
INV(I)=VAL(I)
IF(M.EQ.0)GO TO 13
CALL IO(MES5, INPUT, LEN)
CALL DECOD(INPUT, VAL)
DO 8 I=1, N
P(I)=VAL(I)
NN=N
CALL BATCH(A, S, INV, P, EBQ, T, N, M)
IF(M.EQ.2)GO TO 9
WRITE(6, 11)EBQ(1), T
CALL TIME(4, 1)
FORMAT(' EBQ= ', F10.4, ' NUMBER OF RUNS= ', F10.4)
GO TO 30
WRITE(6, 17)N
CALL TIME(4, 1)
FORMAT(' NUMBER OF CYCLES= ', I4)
DO 35 J=1, NN
WRITE(6, 12)J, EBQ(J)
CALL TIME(4, 1)
FORMAT(' PRODUCT ', I4, ' EBQ= ', F10.4)
IF(LEN.GT.80)GO TO 80
WRITE(6, 2)(MES7(I), I=1, LEN)
CALL TIME(4, 1)
READ(5, 32)ANS
CALL TIME(4, 1)
FORMAT(A4)
IF(ANS.EQ.FIN)CALL RESULT
GO TO 25
CONTINUE
END
SUBROUTINE IO(J, IA, L)
LOGICAL*1 J(80), IA(80)
WRITE(6, 1)(J(I), I=1, L)
CALL TIME(4, 1)
FORMAT(8CA1)
READ(5, 1)(IA(I), I=1, 80)
CALL TIME(4, 1)
RETURN
END
SUBROUTINE DECOD(IA, B)
DIMENSION IA(80), B(10), VAL1(10), VAL2(10)
LOGICAL*1 MINUS/'-'/
LOGICAL*1 IA, IBLANK, IPOINT, ICOMMA
DATA IBLANK/' ', IPOINT/'.'/, ICOMMA/'', '/'
LOGICAL*1 NUMB(4)
EQUIVALENCE(K, NUMB(1))
N=1
IF(COMP(IA(N), IBLANK))16, 14, 16
CONTINUE
K=0
ISIGN=0
M=1
IEND=0
IBEF=1
IAFT=0
IF(COMP(IA(N), MINUS))17, 18, 17

```

```

17 CONTINUE
3 K=0
  NUMB(4)=IA(N)
  K=K-240
  IF(COMP(IA(N), IPOINT))11,4,11
11 IF(COMP(IA(N), IBLANK))12,1,12
12 IF(COMP(IA(N), ICOMMA))13,2,13
13 IF(IAFT.GT.0)GO TO 5
  VAL1(IBEFF)=K
  IBEFF=IBEFF+1
  N=N+1
  GO TO 3
4 IAFT=IAFT+1
  N=N+1
  GO TO 3
5 VAL2(IAFT)=K
  IAFT=IAFT+1
  N=N+1
  GO TO 3
2 VALUE=0.
  IF(IBEFF.EQ.1)GO TO 8
  IBEFF=IBEFF-1
  DO 6 I=1, IBEFF
  VALUE=10.*VALUE
  VALUE=VALUE+VAL1(I)
6 CONTINUE
3 IF(IAFT.EQ.0)GO TO 9
  REM=C.
  IAFT=IAFT-1
  DO 7 I=1, IAFT
  REM=REM*10.
  REM=REM+VAL2(I)
7 CONTINUE
  REM=REM/(10.**IAFT)
  VALUE=VALUE+REM
9 B(M)=VALUE
  IF(ISIGN.EQ.1)B(M)=-B(M)
  ISIGN=0
  M=M+1
  N=N+1
  IBEFF=1
  IAFT=0
  IF(IENC.EQ.1)RETURN
  GO TO 3
1 IEND=1
  GO TO 2
14 N=N+1
  GO TO 15
18 ISIGN=1
  N=N+1
  GO TO 3
10 CONTINUE
  END
  SUBROUTINE BATCH(A,S,I,P,EBQ,T,N,M)
  REAL I
  DIMENSION S(20),I(20),P(20),EBQ(20)
  IF(M.GT.0)CC TO 1
  EBQ(1)=SQRT((2*A*S(1))/I(1))
  T=S(1)/EBQ(1)
6 CONTINUE

```



```

RETURN
1 IF(M.GT.1)GO TO 2
ERQ(1)=SQRT((2*A*S(1))/(I(1)*(1-S(1)/P(1))))
I=S(1)/ERQ(1)
RETURN
2 SUM=0.
DO 3 J=1,N
SUM=SUM+S(J)*I(J)*(1-S(J)/P(J))
3 CONTINUE
NN=N
N=SQRT(SUM/(2*A))
DO 4 J=1,NN
EBQ(J)=S(J)/N
4 CONTINUE
RETURN
END
SUBROUTINE RESULT
CALL TIME(4,1)
READ(7,1)I,J,K
1 FORMAT(1X,I2,1X,I2,1X,I2)
N1=3600*I+60*J+K
3 READ(7,1,END=4)I,J,K
N2=3600*I+60*J+K
N=N2-N1
WRITE(6,2)N
2 FORMAT(I4)
N1=N2
GO TO 3
4 CONTINUE
STOP
END
SUBROUTINE ERROR
WRITE(6,1)
CALL TIME(4,1)
1 FORMAT(' ERROR')
CALL ENTRY1
RETURN
END

```

## 5.2 Fortran Listing of Inventory Package 2.

```
LL INTRPT
LL INTRP(0)
LL ERRCR
NTINUE
ITE(6,1)
LL TIME(4,1)
RMAT(' VERSION#')
AC(5,2)N
LL TIME(4,1)
RMAT(11)
(N.NE.1)GO TO 3
LL SUB1(IFIN)
(IFIN.EQ.1)GO TO 9
  TO 4
(N.NE.2)GO TO 6
LL SUB2(IFIN)
  TO 5
(N.NE.3)GO TO 7
LL SUB3(IFIN)
  TO 5
ITE(6,8)
LL TIME(4,1)
RMAT(' ERROR')
  TO 4
LL RESULT
OP
ID
SUBROUTINE SUB1(IFIN)
LOGICAL*1 MES1(80),MES2(80),MES3(80),MES4(80),MES5(80),MES6(80),
IPUT(80)
REAL INV
DATA
MES1/' IS IT NORMAL,FINITE PRODUCTION RATE,OR SEVERAL PRODUCTS CAS
',18*' '//,
MES2/' WHAT IS THE VALUE OF PLACING ORDER OR SETUP COST$',30*' '//,
MES3/' WHAT ARE THE TOTAL SALES OR ORDER$',44*' '//,
MES4/' WHAT IS THE INVENTORY HOLDING COST PER UNIT$',35*' '//,
MES5/' WHAT IS THE RATE OF PRODUCTION$',48*' '//,
MES6/' WHAT IS THE NUMBER OF PRODUCTS$',48*' '//
DATA MODE1/'NORM'//,MODE2/'FINI'//,MODE3/'SEVE'//
DATA FIN/'YES'//
DIMENSION VAL(10),S(20),INV(20),P(20),EBQ(20)
WRITE(6,2)(MES1(I),I=1,80)
LL TIME(4,1)
RMAT(80A1)
READ(5,3) MODE
LL TIME(4,1)
RMAT(A4)
```

```

M=3
IF(MODE.EQ.MODE1)M=0
IF(MODE.EQ.MODE2)M=1
IF(MODE.EQ.MODE3)M=2
IF(M.NE.3)GO TO 4
WRITE(6,5)
CALL TIME(4,1)
FORMAT(' ERROR')
GO TO 1
N=1
IF(M.NE.2)GO TO 10
CALL IO(MES6,INPUT)
CALL DECOD(INPUT,VAL)
N=VAL(1)
CALL IO(MES2,INPUT)
CALL DECOD(INPUT,VAL)
A=VAL(1)
CALL IO(MES3,INPUT)
CALL DECOD(INPUT,VAL)
DO 6 I=1,N
S(I)=VAL(I)
CALL IO(MES4,INPUT)
CALL DECOD(INPUT,VAL)
DO 7 I=1,N
INV(I)=VAL(I)
IF(M.EQ.0)GO TO 13
CALL IO(MES5,INPUT)
CALL DECOD(INPUT,VAL)
DO 8 I=1,N
P(I)=VAL(I)
NN=N
CALL BATCH(A,S,INV,P,EBQ,T,N,M)
IF(M.EQ.2)GO TO 9
WRITE(6,11)EBQ(1),T
CALL TIME(4,1)
FORMAT(' EBQ= ',F10.4,' NUMBER OF RUNS= ',F10.4)
GO TO 30
WRITE(6,17)N
CALL TIME(4,1)
FORMAT(' NUMBER OF CYCLES= ',I4)
DO 35 J=1,NN
WRITE(6,12)J,EBQ(J)
CALL TIME(4,1)
FORMAT(' PRODUCT ',I4,' EBQ= ',F10.4)
WRITE(6,31)
CALL TIME(4,1)
FORMAT(' DO YOU WISH TO FINISH?')
READ(5,32)ANS
CALL TIME(4,1)
FORMAT(A4)
IFIN=C
IF(ANS.EQ.FIN)IFIN=1
RETURN
CONTINUE
END
SUBROUTINE IO(J,IA)
LOGICAL*1 J(80),IA(80)
WRITE(6,1)(J(I),I=1,80)
CALL TIME(4,1)
FORMAT(80A1)

```

```

READ(5,1)(IA(I),I=1,80)
CALL TIME(4,1)
RETURN
END
SUBROUTINE CLCOD(IA,B)
DIMENSION IA(80),B(10),VAL1(10),VAL2(10)
LOGICAL*1 MINUS/'-'/
LOGICAL*1 IA,IBLANK,IPOINT,ICOMMA
DATA IBLANK/' '/,IPOINT/'.'/,ICOMMA/','/'
LOGICAL*1 NUMB(4)
EQUIVALENCE(K,NUMB(1))
N=1
5 IF(COMP(IA(N),IBLANK))16,14,16
6 CONTINUE
K=0
ISIGN=0
M=1
IENC=0
IBEF=1
IAFT=0
IF(COMP(IA(N),MINUS))17,18,17
7 CONTINUE
3 K=0
NUMB(4)=IA(N)
K=K-240
IF(COMP(IA(N),IPOINT))11,4,11
1 IF(COMP(IA(N),IBLANK))12,1,12
2 IF(COMP(IA(N),ICOMMA))13,2,13
3 IF(IAFT.GT.0)GO TO 5
VAL1(IBEFF)=K
IBEF=IBEF+1
N=N+1
GO TO 3
4 IAFT=IAFT+1
N=N+1
GO TO 3
5 VAL2(IAFT)=K
IAFT=IAFT+1
N=N+1
GO TO 3
2 VALUE=0.
IF(IBEFF.EQ.1)GO TO 8
IBEF=IBEF-1
DO 6 I=1,IBEF
VALUE=10.*VALUE
VALUE=VALUE+VAL1(I)
6 CONTINUE
IF(IAFT.EQ.0)GO TO 9
REM=0.
IAFT=IAFT-1
DO 7 I=1,IAFT
REM=REM*10.
REM=REM+VAL2(I)
7 CONTINUE
REM=REM/(10.**IAFT)
VALUE=VALUE+REM
9 B(M)=VALUE
IF(ISIGN.EQ.1)B(M)=-B(M)
ISIGN=0
M=M+1

```

```

N=N+1
IDBF=1
IAFT=0
IF(IENC.EQ.1)RETURN
GO TO 3
IEND=1
GO TO 2
N=N+1
GO TO 15
) ISIGN=1
N=N+1
GO TO 3
) CONTINUE
END
SUBROUTINE BATCH(A,S,I,P,EBQ,T,N,M)
REAL I
DIMENSION S(20),I(20),P(20),EBQ(20)
IF(M.GT.0)GO TO 1
EBQ(1)=SQRT((2*A*S(1))/I(1))
T=S(1)/EBQ(1)
) CONTINUE
RETURN
1 IF(M.GT.1)GO TO 2
EBQ(1)=SQRT((2*A*S(1))/(I(1)*(1-S(1)/P(1))))
T=S(1)/EBQ(1)
RETURN
2 SUM=0.
DO 3 J=1,N
SUM=SUM+S(J)*I(J)*(1-S(J)/P(J))
3 CONTINUE
NN=N
N=SQRT(SUM/(2*A))
DO 4 J=1,NN
EBQ(J)=S(J)/N
4 CONTINUE
RETURN
END
SUBROUTINE SUB2(IFIN)
LOGICAL*1 MES1(80),MES2(80),MES3(80),MES4(80),MES5(80),MES6(80),
1 INPUT(80)
REAL INV
DATA
1MES1/' CASE$',74*' '//,
2MES2/' PLACING ORDER/SETUP COST$',53*' '//,
3MES3/' TOTAL SALES/ORDER$',60*' '//,
4MES4/' INVENTORY COST$',64*' '//,
5MES5/' PRODUCTION RATE$',63*' '//,
6MES6/' NUMBER PRODUCTS$',63*' '//
DATA MODE1/'NORM'//,MODE2/'FINI'//,MODE3/'SEVE'//
DATA FIN/'YES'//
DIMENSION VAL(10),S(20),INV(20),P(20),EBQ(20)
0 CONTINUE
INDEX=0
1 WRITE(6,2)(MES1(I),I=1,80)
CALL TIME(4,1)
2 FORMAT(80A1)
READ(5,3) MODE
CALL TIME(4,1)
3 FORMAT(A4)
M=3

```

```

IF(MODE.EQ.MODE1)M=0
IF(MODE.EQ.MODE2)M=1
IF(MODE.EQ.MODE3)M=2
IF(M.NE.3)GO TO 4
WRITE(6,5)
CALL TIME(4,1)
5 FORMAT(' ERROR')
GO TO 1
4 N=1
IF(M.NE.2)GO TO 10
CALL IO(MES6,INPUT)
CALL DECOD(INPUT,VAL)
N=VAL(1)
0 CALL IO(MES2,INPUT)
CALL DECOD(INPUT,VAL)
A=VAL(1)
CALL IO(MES3,INPUT)
CALL DECOD(INPUT,VAL)
DO 6 I=1,N
6 S(I)=VAL(I)
CALL IO(MES4,INPUT)
CALL DECOD(INPUT,VAL)
DO 7 I=1,N
7 INV(I)=VAL(I)
IF(M.EQ.0)GO TO 13
CALL IO(MES5,INPUT)
CALL DECOD(INPUT,VAL)
DO 8 I=1,N
8 P(I)=VAL(I)
NN=N
3 CALL BATCH(A,S,INV,P,EBQ,T,N,M)
IF(M.EQ.2)GO TO 9
WRITE(6,11)EBQ(1),T
CALL TIME(4,1)
11 FORMAT(' EBQ= ',F10.4,' NUMBER OF RUNS= ',F10.4)
GO TO 30
9 WRITE(6,17)N
CALL TIME(4,1)
17 FORMAT(' NUMBER OF CYCLES= ',I4)
DO 35 J=1,NN
35 WRITE(6,12)J,EBQ(J)
CALL TIME(4,1)
12 FORMAT(' PRODUCT ',I4,' EBQ= ',F10.4)
30 WRITE(6,31)
CALL TIME(4,1)
31 FORMAT(' FINISH')
READ(5,32)ANS
CALL TIME(4,1)
32 FFORMAT(A4)
IFIN=0
IF(ANS.EQ.FIN)IFIN=1
RETURN
33 CONTINUE
ENC
SUBROUTINE SUB3(IFIN)
LOGICAL*1 FINIS,ANS,MODE,MODE1,MODE2,MODE3
LOGICAL*1 IA(80)
DIMENSION U(10)
DATA
1MODE1/'N'//,MODE2/'F'//,MODE3/'S'//,FINIS/'Y'//

```

```

DIMENSION INV(20),EBO(20),P(20),S(20)
REAL INV
5 WRITE(6,1)
CALL TIME(4,1)
1 FORMAT(' C1')
READ(5,2)MOCE
CALL TIME(4,1)
2 FORMAT(A1)
N=1
M=3
WRITE(6,6)
CALL TIME(4,1)
5 FORMAT(' S.C1')
READ(5,5)(IA(I),I=1,80)
CALL TIME(4,1)
5 FORMAT(80A1)
CALL DECOD(IA,B)
A=B(1)
IF(COMP(MOCE,MOCE3))7,8,7
8 WRITE(6,9)
CALL TIME(4,1)
M=2
9 FORMAT(' N.P1')
READ(5,5)(IA(I),I=1,80)
CALL TIME(4,1)
CALL DECOD(IA,B)
N=B(1)
7 WRITE(6,11)
CALL TIME(4,1)
1 FORMAT(' T.S1')
READ(5,5)(IA(I),I=1,80)
CALL TIME(4,1)
CALL DECOD(IA,B)
DO 10 I=1,N
) S(I)=B(I)
WRITE(6,12)
CALL TIME(4,1)
2 FORMAT(' I.C1')
READ(5,5)(IA(I),I=1,80)
CALL TIME(4,1)
CALL DECOD(IA,B)
DO 13 I=1,N
3 INV(I)=B(I)
IF(COMP(MOCE,MOCE1))16,17,16
5 CONTINUE
5 IF(M.NE.2)M=1
WRITE(6,14)
CALL TIME(4,1)
4 FORMAT(' P.R1')
READ(5,5)(IA(I),I=1,80)
CALL TIME(4,1)
CALL DECOD(IA,B)
DO 15 I=1,N
5 P(I)=B(I)
GO TO 18
7 M=0
3 IF(M.EQ.3)GO TO 30
NN=N
CALL PATCH(A,S,INV,P,EBO,I,T,M)
IF(M.EQ.2)GO TO 31

```

```

WRITE(6,32)EBQ(1),T
CALL TIME(4,1)
FORMAT(' EQ= ',F10.4,' NO=',F10.4)
GO TO 33
WRITE(6,40)
CALL TIME(4,1)
FORMAT(' ERROR')
GO TO 25
WRITE(6,17)N
CALL TIME(4,1)
FORMAT(' N=',I4)
WRITE(6,21)
CALL TIME(4,1)
FORMAT(' EBQ')
DO 20 J=1,NN
WRITE(6,22)EBQ(J)
CALL TIME(4,1)
FORMAT(F10.4)
CONTINUE
WRITE(6,23)
CALL TIME(4,1)
FORMAT(' F')
READ(5,24)ANS
CALL TIME(4,1)
FORMAT(A1)
IF(COMP(FINIS,ANS))27,26,27
IFIN=1
RETURN
  IFIN=0
RETURN
END
SUBROUTINE RESULT
CALL TIME(4,1)
READ(7,1)I,J,K
FORMAT(1X,I2,1X,I2,1X,I2)
N1=3600*I+60*J+K
READ(7,1,END=4)I,J,K
N2=3600*I+60*J+K
N=N2-N1
WRITE(6,2)N
FORMAT(I4)
N1=N2
GO TO 3
CONTINUE
STOP
END
SUBROUTINE ERROR
WRITE(6,1)
CALL TIME(4,1)
FORMAT(' ERROR')
CALL ENTRY1
RETURN
END

```



### 3.3 Assembler Routines used by both Packages.

#### 3.3.1 Routine to Compare Logical Variables.

```
OMP      CSECT
         SAVE (14,12)
         PALR 12,0
         USING *,12
         LR   2,1
         L    3,0(2)
         L    4,=F'0'
         L    5,=F'0'
         IC   4,0(3)
         L    3,4(2)
         IC   5,0(3)
         XR   4,5
         BZ   ZERO
IE       LE   0,=D'1'
         B    FINIS
ERO      LE   0,=D'0'
NIS      RETURN (14,12)
         END
```

#### 3.3.2 Error Handling Routine to Return user to Beginning of Package.

```
NTRPT   CSECT
         LR   11,15
         USING INTRPT,11
         STM  13,12,SAVE1
         ENTRY ENTRY1
         L    15,=V(PGNTTRP)
         L    0,=V(ERROR)
         LA   1,SAVE
         BALR 14,15
         LM  13,12,SAVE1
         BCR  15,14
         DROP 11
ENTRY1   LR   11,15
         USING *-2,11
         L    15,=V(MAIN)
         L    13,SAVE1
         L    13,4(13)
         L    14,12(13)
         LM  0,12,20(13)
         BCR  15,15
SAVE1    DS   8D
SAVE     DS   9D
         END
```

## 1.4 Statistical Routines used with Experimental Timing Data.

### 1.4.1 Routine to Compute Student t Values.

```

      DIMENSION IX1(20),IX2(20)
      DATA Y/'YES '/
      T=0.
      T2=0.
      TOT1=0.
      TOT2=0.
      READ(5,1) I1, I2
      FORMAT(2I2)
      READ(5,2)(IX1(I), I=1, I1)
      FORMAT(20I4)
      READ(5,2)(IX2(I), I=1, I2)
      DO 3 I=1, I1
      TOT1=TOT1+IX1(I)
      CONTINUE
      DO 4 I=1, I2
      TOT2=TOT2+IX2(I)
      CONTINUE
      AVE1=TOT1/I1
      AVE2=TOT2/I2
      DO 5 I=1, I1
      S1=(IX1(I)-AVE1)**2+T1
      CONTINUE
      DO 6 I=1, I2
      S2=T2+(IX2(I)-AVE2)**2
      CONTINUE
      I=I1
      J=I2
      SIG1=(T1/I1)/(I1-1)
      SIG2=(T2/I2)/(I2-1)
      SIG=SQRT(SIG1+SIG2)
      S=((SIG1+SIG2)**2)/(((SIG1**2)/(I1+1)+(SIG2**2)/(I2+1))
      S=DF-2
      S=(AVE1-AVE2)/SIG
      WRITE(6,7)T
      WRITE(6,7)DF
      FORMAT(F10.4)
      READ(5,10)R
      FORMAT(A4)
      S=(R.EQ.Y)STOP
      GO TO 8
      STOP
      END
```

### 13.4.2 Routine to Compute Mann-Whitney U Values.

```
DIMENSION IX1(20),IX(40)
DATA S/'YES '/
READ(5,1)N1,N2
FORMAT(2I2)
N=N1+N2
READ(5,2)(IX(I),I=1,N)
FORMAT(40I4)
DO 3 I=1,N1
IX1(I)=IX(I)
DO 4 I=1,N
DO 4 K=I,N
IF (IX(I).LT.IX(K))GO TO 4
IT=IX(I)
IX(I)=IX(K)
IX(K)=IT
CONTINUE
WRITE(6,2)(IX(I),I=1,N)
T1=0.
DO 10 I=1,N1
DO 5 K=1,N
IF (IX(K).NE.IX1(I))GO TO 5
SL=K
RK=K
NN=1
L=K
IF (IX(L+1).NE.IX(L))GO TO 13
SL=SL+L+1
NN=NN+1
L=L+1
GO TO 14
RR=NN
RK=SL/RR
T1=T1+RK
GO TO 10
CONTINUE
CONTINUE
R1=N1
R2=N2
U=R1*R2+(R1*(R1+1.))/2.-T1
Y=(R1*R2)/2.
IF (U.LE.Y)GO TO 15
U=R1*R2-U
WRITE(6,16)U
FORMAT(' UDASH',F10.4)
GO TO 17
CONTINUE
WRITE(6,6)U
FORMAT(F10.4)
CONTINUE
READ(5,7)R
FORMAT(A4)
IF (R.EQ.S)STOP
GO TO 8
RETURN
END
```

14 APPENDIX 5

STATISTICAL RESULTS FOR STUDENT THINK TIMES WITH ERROR TIMES

14.1 Table 1. A comparison of Group 1 with Group 3 for the first 3 problems of each session using the think times including 'error times' - Student t values.

14.1.1 For package 1.

Problem	Session	t value	Degrees freedom	approximate probability
1	1	- 0.9710	2	0.50
1	2	+ 0.9396	1	0.50
1	3	+ 0.8427	2	0.50
2	1	- 0.0331	2	0.90
2	2	- 0.7845	1	0.70
2	3	- 0.7939	5	0.50
3	1	- 0.3637	2	0.70
3	2	- 0.2113	4	0.85
3	3	- 0.1300	6	0.90

14.1.2 For package 2.

Problem	Session	t value	Degrees freedom	approximate probability
1	1	+ 0.7748	3	0.50
1	2	- 0.3563	7	0.70
1	3	- 0.5353	6	0.60
2	1	+ 2.8274	3	0.07
2	2	+ 1.4190	2	0.30
2	3	- 0.8470	4	0.40
3	1	+ 2.1679	1	0.25
3	2	+ 1.7037	5	0.18
3	3	+ 1.3593	2	0.23

14.2 Table 2. A comparison of package 1 with package 2 for all subjects together using the think times including 'error times' - Student t values.

Problem	Session	t value	Degrees freedom	Approximate Probability
1	1	- 0.6625	11	0.50
1	2	- 2.4451	18	0.02
1	3	- 1.9618	10	0.10
2	1	- 2.2381	11	0.05
2	2	- 0.9386	13	0.50
2	3	- 2.4271	9	0.50
3	1	+ 0.6128	10	0.50
3	2	- 0.2193	12	0.80
3	3	- 2.6448	10	0.02
4	1	- 0.9000	17	0.35
4	2	- 0.7521	11	0.40
4	3	+ 1.2167	17	0.20
5	1	- 0.4653	15	0.60
5	2	- 1.5926	12	0.10
5	3	- 0.9962	13	0.35
6	1	- 1.443	10	0.20
6	2	- 0.2309	18	0.80
6	3	- 0.1991	21	0.80
7	1	- 1.6745	15	0.10
7	2	- 0.1890	12	0.80
7	3	- 1.6155	11	0.10
8	1	- 0.4686	19	0.60
8	2	- 0.8446	10	0.40
8	3	- 0.6159	16	0.50
9	1	- 0.6590	19	0.50
9	2	- 0.1490	21	0.80
9	3	- 1.0676	16	0.40
10	1	- 0.4342	14	0.60
10	2	- 0.9349	22	0.35
10	3	+ 1.1613	15	0.25

14.3 Table 3. A comparison of package 1 with package 2 for all subjects together using the think times including 'error times' - Mann-Whitney U values. (Negative values indicate when package 2 has lower mean.)

Problem	Session	U value	Approximate probability NS = not significant
1	1	29.0	NS
1	2	11.0	.01
1	3	25.0	.05
2	1	12.5	NS
2	2	40.0	NS
2	3	42.5	NS
3	1	36.0	NS
3	2	27.5	NS
3	3	18.5	.01
4	1	37.5	NS
4	2	44.5	NS
4	3	- 36.5	NS
5	1	37.5	NS
5	2	35.5	NS
5	3	40.5	NS
6	1	25.0	NS
6	2	- 42.5	NS
6	3	48.5	NS
7	1	23.0	.05
7	2	35.0	NS
7	3	44.5	NS
8	1	45.0	NS
8	2	- 39.0	NS
8	3	57.0	NS
9	1	37.5	NS
9	2	55.0	NS
9	3	48.5	NS
10	1	46.5	NS
10	2	45.0	NS
10	3	- 51.5	NS

15 APPENDIX 6

STATISTICAL RESULTS FOR STUDENT THINK TIMES WITHOUT ERROR TIMES

15.1 Table 1. A comparison of group 1 with group 3 for the first 3 problems of each session using the think times without 'error times' - Student t values.

15.1.1 For package 1.

Problem	Session	t value	Degrees freedom	Approximate probability
1	1	+ 0.2375	3	0.80
1	2	+ 0.9396	1	0.60
1	3	- 0.0376	2	0.90
2	1	- 0.5512	1	0.70
2	2	- 0.7845	1	0.60
2	3	- 0.7939	5	0.50
3	1	- 0.4877	1	0.70
3	2	+ 0.0637	4	0.90
3	1	+ 0.4961	7	0.65

15.1.2 For package 2.

Problem	Session	t value	Degrees freedom	Approximate probability
1	1	+ 2.0014	2	0.20
1	2	- 0.3563	7	0.70
1	3	- 0.5353	6	0.60
2	1	+ 2.0455	5	0.10
2	2	+ 1.4190	2	0.30
2	3	- 0.0814	7	0.90
3	1	+ 2.1679	1	0.30
3	2	+ 1.6737	5	0.20
3	3	+ 0.5529	2	0.60

15.2 Table 2. A comparison of package 1 with package 2 for all subjects together using the think times without 'error times' - Student t values.

Problem	Session	t value	Degrees freedom	Approximate probability
1	1	- 0.4202	13	0.65
1	2	- 2.4451	18	0.02
1	3	- 2.2077	10	0.05
2	1	- 2.4183	17	0.02
2	2	- 0.9386	13	0.40
2	3	- 0.4964	18	0.60
3	1	- 2.1740	13	0.05
3	2	- 1.3013	15	0.20
3	3	- 1.9642	9	0.10
4	1	- 0.3603	18	0.70
4	2	- 0.0855	14	0.90
4	3	+ 0.2223	20	0.80
5	1	- 0.2596	17	0.80
5	2	- 1.9981	10	0.10
5	3	- 0.7512	12	0.40
6	1	- 1.2185	15	0.30
6	2	- 0.7492	15	0.40
6	3	- 1.2945	13	0.30
7	1	- 1.1236	16	0.30
7	2	- 1.2111	15	0.30
7	3	- 1.2925	11	0.30
8	1	- 1.2758	12	0.30
8	2	- 0.7835	10	0.40
8	3	- 0.6159	16	0.50
9	1	- 2.6752	12	0.02
9	2	- 1.6330	13	0.10
9	3	- 1.1172	15	0.30
10	1	- 0.5697	21	0.40
10	2	- 0.9408	22	0.40
10	3	- 0.5397	21	0.60



15. 3 Table 3. A comparison of package 1 with package 2 for all subjects together using the think times without 'error times' - Mann-Whitney U values. (Negative values indicate when package 2 has a lower mean.)

Problem	Session	U value	Approximate probability NS = not significant
1	1	25.5	NS
1	2	11.0	.01
1	3	21.0	.25
2	1	17.5	.05
2	2	- 40.0	NS
2	3	43.5	NS
3	1	13.0	.01
3	2	30.0	NS
3	3	27.5	.05
4	1	42.0	NS
4	2	- 42.5	NS
4	3	49.0	NS
5	1	- 40.0	NS
5	2	32.5	NS
5	3	45.5	NS
6	1	31.5	NS
6	2	43.5	NS
6	3	41.5	NS
7	1	29.5	NS
7	2	- 35.0	NS
7	3	45.0	NS
8	1	37.0	NS
8	2	- 37.5	NS
8	3	57.0	NS
9	1	25.0	.025
9	2	44.5	NS
9	3	48.5	NS
10	1	41.0	NS
10	2	43.5	NS
10	3	53.0	NS

16 APPENDIX 7

ERROR SCORES FROM EXPERIMENT

(WC = WRONG CONCEPT WE = WRONG ENTRY T = TYPEWRITING)

16.1 Error Scores for Package 1.

WC	WE	T	Session	Group
4	8	13	1	1
2	19	4	1	2
1	14	4	1	3
2	15	4	2	1
2	6	6	2	2
3	11	5	2	3
0	4	6	3	1
0	2	3	3	2
2	9	4	3	3
16	88	49		

16.2 Error Scores for Package 2.

WC	WE	T	Session	Group
2	18	8	1	1
4	6	1	1	2
2	19	4	1	3
2	9	1	2	1
0	3	3	2	2
1	17	2	2	3
0	5	0	3	1
2	10	4	3	2
0	14	1	3	3
13	101	24		

16.3 Error Scores on Problem 3 and Problem 7 Only.

WC	WE	T	Package
6	59	19	1
4	68	16	2

17. APPENDIX 8

DOCUMENTS USED IN THE SURVEY

17.1. Introductory Circular.

Dear Sir,

Durham University Business School is currently engaged in research concerning on-line computing and its applications to business problems, and a literature survey of the uses of on-line computing has just been completed. It is now proposed to investigate the problems which managers have encountered with these applications by conducting a series of interviews in several companies. We would, therefore, be extremely grateful if you could fill in the attached slip, as soon as possible, and return it to:-

J.R. Shergold,  
Business School,  
30 Old Elvet,  
Durham City.

We will ensure that these interviews will not take up too much of your time and that all companies will remain anonymous in any reports published, so that any information you may kindly give us will be treated in the strictest confidence.

Yours sincerely,

.....

From:

Company:

- |  | Yes                      | No                       |
|--|--------------------------|--------------------------|
| 1. Does your company possess any on-line computing facilities? | <input type="checkbox"/> | <input type="checkbox"/> |
| 2. Would you be willing to co-operate in this survey?          | <input type="checkbox"/> | <input type="checkbox"/> |

17.2. Original Questionnaire used as a Checklist in Interviews.

17.2.2. Computer Department Section.

1. What type of time-sharing computer do you use?

Manufacturer (IBM, ICL etc)

Model (360/67, 19054 etc)

System (TSS, GEORGE 3 etc)

2. What input-output devices do you have?

Typewriter terminals

Teletype terminals

V.D.U.'s

Interactive displays

Graph plotter

Satellite Computer

Other

3. What percentage of your computer time do you estimate is spent in the following areas?

Financial and Administrative

Management Planning and Control

Marketing and Sales Operations

Distribution Operations

Factory Operations

Research and Development and

Engineering

Other

### 17.2.3. Other Departments Section.

1. Please indicate what on-line computer applications that you have and your degree of satisfaction with them.

Here the applications described in Chapter 2 for each functional area were used as a checklist.

2. What dissatisfactions have you had with these applications?

Not enough management involvement in development.

Lack of useful exception information.

Information not produced quickly enough.

Output is not what management wants.

Programming expensive and takes too long.

System requirements outgrow equipment

i.e. may have many improved applications in mind but limited by resources (cost of extra core too high etc.)

Other.

3. What advantages have you found with these applications?

Speedier information for decisions.

Ability to try more possibilities.

Ease of information retrieval.

Other.

4. Are there any applications which you have now that were impossible before time-sharing?

### 17.2.4. Computer Users Section.

1. Do you use the computer directly or via some kind of data-secretary such as a programmer?

Directly

Indirectly

2. What kind of interactive device do you use?
3. What advantages do you find from using the computer in this way?

Easier access to information.

Speedy response.

Stimulates user in problem-solving process.

Convenience.

Other.

4. What dissatisfactions do you have with this method?

Lack of input-output devices for the user.

Inflexibility in programming.

Poor information retrieval.

Response time slow and irritates user.

Input devices slow and inefficient.

Languages not conversational enough.

Error recovery too difficult.

Other.

5. How would you as a manager like to use the computer in your problem solving?

Via a programmer or systems analyst.

Using some interactive device via a data-secretary.

Using some interactive device directly.

What device.

What language.

