MODELLING END-USER SUPPORT

JACQUELINE MARY MEW

Master of Philosophy

1990

I wish to thank the following people for their support.

Professor P. Henderson,
University of Southampton.

Mr. R. Ruel and Mr. B. Dowding,
Southampton Institute of Higher Education.

Professor M. Bramer
Portsmouth Polytechnic.

Mr. S. Adams, Ms D. Callaghan and Mr. G. Farrow
Southampton City Council.

TABLE OF CONTENTS.

		Page
1.	Why Research into End-User Support?	1
	1.1 Introduction	1
	1.2 A discussion on end-user definitions.	2
	1.3 An Outline of this Thesis	6
2.	End User Perspective	8
	2.1 Introduction	8
	2.2 The General Situation 1984 and beyond	8
	2.3 Southampton City Council	13
	2.3.1 Up to 1985	14
	2.3.2 After 1987	17
	2.3.3 1985 to 1987	19

3. System Dynamics and STELLA. 25

3.1 Introduction 25
3.2 System Dynamics. 25
3.3 A History of Experiments with System Dynamic Tools. 34
3.3.1 By Paper,Pencil and Rubber. 34
3.3.2 LISP. 34
3.3.3 LOTUS 1-2-3. 35
3.3.4 COSMIC. 35
3.3.5 STELLA. 36
3.4 STELLA 37
3.4.1 Good Points Noted 38
3.4.2 Problems Noted 39
3.4.3 Experimenting with Models 40

4. The Models 46

4.1 Introduction 46
4.2 First Model : Simple Model 48
4.3 Second Model : Alternative Support 50
4.4 Third Model : Support 52
4.5 Summary 56

5. Results from the Models 59

5.1 Introduction 59
5.2 The First Model : Simple Model 59
5.2.1 The Equations used in the First Model 59
5.2.2 The First Model as a Whole 66
5.3 The Second Model : Alternative Support 69
5.3.1 The Equations used in the Second Model 69
5.3.2 The Second Model as a Whole 82
5.4 The Third Model : Support 84
5.4.1 Support Influence Diagram 84
5.4.2 Equations used in the Third Model 86
5.4.3 The Third Model as a Whole 97

6. Discussion of relationships between reality and the models. 102
6.1 Introduction 102
6.2 The models as a starting point 102
6.3 A Discussion of Originality within this Thesis 106
6.4 Summary 108

7. Conclusions & Future Work 110
7.1 Conclusions 110
7.2 to 7.20 Suggestions for Future Research 110
7.21 Summary of Future Work 117
7.22 The Next Stage 118

APPENDICES

A. Examples of Experiments with System Dynamic Tools. 120
A1. By Paper, Pencil and Rubber. 120
A2. By LISP 121
A3. By LOTUS 122

B. Southampton City Council Problem Logs Summary of Questions Asked. 124
B.1 1985 124
B.2 1986 126
B.3 1987 126

C. Sample Questions from Southampton City Council Problem Log July 1985 127

D. Sample Problem Log for July 1985, Southampton City Council 129

E. Extract from End-User Support's Diary July 1985. 130

F. Consultations 132

G. Taxonomy of Problems Encountered 134

H. A Charging Policy Example and how it can relate to justifying staff 135

BIBLIOGRAPHY 137

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING AND APPLIED SCIENCE

ELECTRONICS AND COMPUTER SCIENCE

Master of Philosophy

MODELLING END-USER SUPPORT

by Jacqueline Mary Mew

The concepts and practicalities of End-User Support are explored before selecting one aspect : the factors influencing the rate at which End-User Support answers questions posed by End-Users. System Dynamics is selected as a means of modelling the asking and answering of questions, and the way that queues of unanswered questions grow and diminish. System Dynamics is outlined. Several tools are evaluated for use with System Dynamics, before selecting Stella, a Systems Dynamics package. Stella is described. Three models are produced: A Simple Model, Alternative Support and Support. The Simple Model reflects just the asking and answering of questions. Alternative Support explores the switch by End-Users to a secondary source of Support. Support explores some of the influences on the rate at which End-User Support answers questions. The models are selected as complex enough to illustrate interesting aspects of the problem, whilst being simple enough to be able to produce actual working models. They have been used to clarify thoughts on the provision of End-User Support. The relationship between the models and the real world is dicussed, using figures from Southampton City Council. The creation of a model in conjunction with an End-User Support Manager would help to highlight areas of concern. There is scope to develop more sophisticated models which could be used as a planning tool. The conclusion includes several ideas for future research.

1. WHY RESEARCH INTO END-USER SUPPORT?

1.1 Introduction.

Much attention has been focused on large-scale computer systems: their cost to an organisation is immediately apparent. However, consider the myriad small systems that abound in corporations. Each microcomputer may cost a few thousand pounds and each package a few hundred. Let us assume that the hardware cost is in the region of £2,500, a conservative estimate, for a microcomputer and printer. Let us further assume that the software averages to about £500 per machine (LOTUS 1-2-3 was then (1984/85) £375 retail, but Southampton City Council could negotiate discounts and most users had more than one package on their Personal Computer.) In 1986 Southampton City Council had over 50 microcomputers, so, at a rough estimate £150,000 was invested in end-user computing. Personal Computing on the mainframe, an IBM 4381 has not been included in these estimates, although there were over 70 Virtual Machines set up for end-users to use APL (A Programming Language), ADI (APL Departmental Interface, a file interrogation language) and ADRS (A Departmental Reporting System).

The funding for these items usually lies within departmental budget, and is therefore hidden from view in terms of corporate budgetting.

So, from a financial viewpoint alone, end-user computing bears further investigation. In addition, simply because it is a relatively new area of computing, it becomes an object of curiosity.

1.2 A Discussion on end-user definitions.

A phrase central to this thesis is 'end-user' either as end-user support or as end-user computing.

Winkler [1] argues for the inclusion within the term 'end-user' everyone who has need of computing resources in order to carry out tasks. He includes computer professionals as at some point even programmers will use tools defined as for end-users, such as a word processor. This was borne out by experience at Southampton City Council; for example the operations manager made extensive use of ADRS for his statistics. Taking this concept, end-user computing is the direct hands-on use of computers, by users of computing facilities. This includes the users of traditional systems, which is too broad a definition, in the context of this thesis. It is suggested, then, that end-user computing is determined by the facilities and the way in which they are used, rather than the people using them.

Within the category of end-user there are several sub-categories.

Martin [3] has identified five categories of end-user of computers. The first three do not fall into the scope of this thesis as they use traditional applications, however it could be argued that the third category could if it was amended to include applications written in end-user languages (such as dBASE III, APL) by support personnel.

a) Indirect users:
   are people who use computers via a third party such as a bank teller.

b) Off-line users:
   those who use computer listings.

c) Use but do not create applications:
   including both applications written as a result of traditional development as well as applications developed using end-user languages.

d) Use and create applications employing the use of application generators:
   these end-users are non-programming users.

e) Use and create applications using APL, BASIC, dBASE III etc.:
   these end-users are programming users.

These definitions do not acknowledge the role of the Computer Services Department.Carr [2] has classified six categories;

a) Non-programming end-users who use software provided by others.

b) Command-level end-users who perform simple enquiries and calculations.

c) End-user programmers who use both command and procedural languages in order to satisfy their own information needs.

d) Functional support personnel are end-user programmers who support other end-users within their particular work area. At Southampton City Council they were referred to as liaison officers.

e) End-user computing support personnel provide support for end-users and are usually located within Computer Services, or an Information Centre.

f) DP programmers are analogous to traditional COBOL programmers, except that they program in end-user languages such as dBASE III.

This does acknowledge the role of end-user support. Whether end-user computing is regarded as a necessary evil, or whether it is regarded as something to be nurtured, end-user computing is here to stay, but not, perhaps, in its current form. It is a system, and like all systems that survive, adapts. A discussion centered around the growth pattern is in

Section 2.2. Having spent the estimated 150,000 pounds on end-user computing equipment, it is in the company's interest to see that the money is well spent. There is a lovely image of a microcomputer being used to wedge open a door in one of Douglas Adam's works of fiction [4]. Although this example is only a work of fiction, too many micro computers are bought and under-used. Alternatively too many are bought and over-used by people re-inventing the wheel.

This misuse of time is even more hidden than the cost of purchasing the equipment. Using a computer can be regarded as only 'doing one's job'. This is where the role of support comes in. It is a role of coordination and advice, drawing on practicalities learnt in traditional development.

If it may be accepted that support is useful in an organisation that has end-users, then end-user support itself must come under scrutiny.

An image that springs to mind is that of a juggler with many plates on the end of long sticks. The juggler is trying to keep the plates spinning in the air. End-user support feels like this to the practitioner.

At what point is a second juggler needed to keep the plates in the air? At what point can end-user support justify an additional member of staff?

If this member of staff is not forthcoming, what can be done to make the service more efficient and effective?

Should one keep adding more and more plates (end-users) without the resources to support them?

1.3 An outline of this thesis.

This dissertation uses System Dynamics models to start to clarify ideas on the end-user support. The subject of end-user support is wide, so attention has been focused on the way in which end-user support answers questions raised by users, as a rough and ready guide as to the effectiveness of the support provided. In the chapters that follow, the role of end-user support is discussed more fully. System dynamics is outlined, together with an analysis of the tools used; why Stella was chosen and how it was used. The models themselves are described and then discussed in detail. The models are then applied to the real world, and the question is asked as to how well they stand up to scrutiny. Finally conclusions are drawn and pointers are left for future research.

References.

[1] S. Winkler
The Information Resource Center, A Strategy for End User Support.
CH2022-2/84/0000/0504 IEEE

[2] J. Martin
Application Development without Programmers
pp 102-106
Prentice Hall 1982

[3] H.H.Carr
Managing End User Computing
pp 61-62
Prentice Hall 1988

[4] D. Adams
Dirk Gently's Holistic Detective Agency
pp 79-80
Pan

2. END USER PERSPECTIVE.

2.1 Introduction.

In order to put the experiments performed into context, this chapter concentrates on the type of work done by end-user support in general, before homing in on the area covered by the models. To begin with reference will be made to literature relating to end-user support, and the work being done in this field. An analysis follows of the type of activity carried out by Southampton City Council.

Records kept by Southampton City from the years 1985 to 1987 inclusive were kindly made available. As well as providing a basis for the models, this data can be used to draw a picture during these years. Additional information has been included in order to provide a fuller picture. The primary source of the data were work diaries and problem logs kept by end-user support. Notes were made of discussions with support staff.

2.2 The General Situation 1984 and Beyond.

Winkler [1] explains that the central problem for Information Systems is to provide access to information by end-users. This can be satisfied by the computer department providing reports and on-line screens, it can also be provided by the users

themselves, given appropriate tools and training. He goes on to explain the problem from the viewpoint of the user, the Computer Services Department and Management. He suggests that the users felt that the Computer Services Department was non-responsive. The Computer Services Department felt that the users did not understand the impact of 'unreasonable' requests. (How many times have we heard how the task could be done in half-an-hour on the user's BBC micro?) Perceptive management realised the mismatch between the user requirements and the Computer Services capability to deliver. Accordingly a solution was presented in terms of an Information Centre (a form of end-user support.) In this instance an Information Centre explicitly refers to information found on a computer. The term Information Centre is frequently used to mean the services provided by a library [2,3]. Winkler argues that an Information Centre would be able to provide the following types of service:

a) Analyse end-user requirements.

b) Help select hardware and software.

c) Provide access to training.

d) Provide help for user problems.

e) Know where the data is, and how to access it.

f) Encourage and facilitate end-user computing.

It may also provide any, or all, of the following services:

g) Demonstrations and hands-on access to software and hardware.

h) Train users.

i) Centralised buying and installation of equipment.

j) Centralised inventory of equipment.

k) Hot-line for end-user problem solving.

l) Maintenance of hardware and software.

m) Networking and micro-mainframe links.

n) Long range planning

Points (d) and (k) above have been highlighted as they explicitly refer to the area being modelled.

In 1984 Southampton City Council were providing this type of service. The use of the term Information Centre has generally been avoided as this implies a formal arrangement with a distinct section. Not all organisations who provided the facilities outlined above had set up an 'Information Centre'. Southampton City Council provided these facilities as a distinct

role, but in two separate sections. Hardware support was provided by Operations, software support was provided from the Development Team. The split was a matter of history. At that period Portsmouth City Council was providing support for both software and hardware from the same team. Winkler discusses the positioning of end-user support as does Martin [4]. Winkler suggests three places for end-user support:

a) An independant centre away from Computer Services, reporting through a separate chain of command.

b) A separate centre, parallel to the Computer Services, reporting to the same manager.

c) Within Computer Services.

Martin favours placing the Information Centre within Computer Services, but separate from the database administration and conventional application development. Central to the whole issue of end-user computing is that it complements traditional development, it is not a substitute for it.

Another area to consider is the staffing of end-user support. Winkler suggests:

a) A manager experienced in end-user support.

b) People who find it easy to communicate with both users and technicians, who act as a first contact with the users.

c) Training specialists.

d) Hardware specialists.

e) Software specialists.

f) Administrative, clerical and secretarial support.

This is ideal when the support centre is flourishing. When starting up, these qualities have to be found in the one person to take on all these roles.

Martin [5] suggests two areas of functions to be carried out by support staff: consultants and technical specialists.

a) Consultants: train and encourage users; consult on end-user problems; determine suitability of application to end-user development; communicate at all levels of the organisation etc.

b) Technical Specialists: set up systems; deal with hardware and software problems; monitor system usage and suggest improvements in efficiency, communicate with dealers etc.

Carr [6] discusses the staffing in terms of how many users the staff can support and the backgrounds of the staff. He suggests that the number of users to one member of staff should be kept to below twenty to one until liaison officers can be established (1.2).

It is difficult to give precise figures from Southampton City Council as users were not registered as such and reliance was put on several liaison officers. An estimate would be in the region of one member of support staff to one hundred and ten users in 1985. From the analysis of the 1985 Problem Log, there were questions from 110 different users. Carr suggests that two-thirds of the end-user support staff had Data Processing backgrounds, but that there are excellent reasons for using people from user-backgrounds. At Southampton City Council all of the full-time end-user support staff had computing backgrounds.

2.3 Southampton City Council.

Names are immaterial in this section, I will denote various people providing support as Support-X, Support-Y etc. Similarly any end-users mentioned will be User-A, User-B etc.

To put the years 1985 to 1987 into context this section begins with a brief history up to 1985, followed by an outline of developments since 1987. Some of the projects from 1985 to 1987 will be described to show the scope of work undertaken by end-user support.

Southampton City Council (SCC) used the term Personal Computing rather than End-User Support. The concept is similar. Personal Computing was descriptive as there were no checks on the work done by users, it was

'personal'. With the coming of Data Protection Legislation and the Corporate Information Systems Strategy this changed. The Data Protection Act meant that any personal data on any computer had to be registered: many of the personal computing activities were manipulating personal data. The Information Systems Strategy required knowledge of all computing activities in order to enable the most effective use of resources.

2.3.1 Up to 1985.

a) Support-X.

Until he left in July 1984, Support-X provided an informal service to users without an understudy, although it was acknowledged that the area was vulnerable.

From 1980 until October 1982 end-user computing was centered on the IBM mainframe (4381) exploiting the properties of the Virtual Machine (VM) environment and using APL, ADRS and ADI. The users were a core of enthusiasts, primarily accountants and engineers. One enthusiast was a top manager who provided much of the thrust instrumental in getting acceptance of end-user computing in Southampton City Council. This reflects the experience noted by Martin [7] in that in the most successful cases motivation is provided by top

management.

From October 1982 end-user computing on the mainframe continued, but began providing a smaller proportion of the total growing end-user environment. In October 1982 the first IBM PC was purchased by accountancy, and this marked the switch towards micro-based, end-user computing. Two machines were purchased in November, by City Transport (at Portswood) and Direct Works (at Town Depot by the Itchen Bridge), both sites were approximately 2 miles from the central support site.

Remote sites add a complication to support as it is impossible to just 'pop along the corridor' to visit the user when a problem arises. One technique is to recreate the problem on a terminal at the support end and then talk through the solution by telephone. This assumes that support has a good range of equipment and packages and that the machines at both ends are situated near telephones.

Between October 1982 and July 1984, 22 IBM PC's and compatibles were installed in 10 departments. Support did not receive its own PC until March 1984. There were 3 Compaqs, 2 XT's and the rest were PC's.

In July 1984 Support-X left. For 5 to 6 weeks Support-Y worked alongside Support-X, in order

to gain experience in end-user support. Both Support-X and Support-Y had similar computing backgrounds. Their career paths were Programming then Systems Design working in both Development and Maintenance. Support-X began as an Operator.

b) Support-Y.

Three months were given to Support-Y to find her feet. Users do not wait for this. Activities handled during August and September 1984 included:

i) Training users in LOTUS 1-2-3 and Chartman.

ii) Demonstration of ADRS II

iii) Word processing evaluation of DisplayWrite 2 and Symphony. The packages were tested against an application provided by Archeology. Several archeologists group together to write a report on their dig. Their requirements included tables, indexing and footnotes.

iv) Accompanying a user to a Dealer to view a package: Typing Tutor.

v) Finding a suitable end-user database for Port Health Authority. ADRS II, Tomorrow's Office and Symphony came under scrutiny. One of the requirements was to do a string search to

find 'prawns' hidden in Cold Prawns, prawns-cooked, etc.

vi) Attending a union consultation (Appendix F) representing computing, so that accountants could use Symphony. It was approved but only typists were allowed to use wordprocessing facilities.

In October 1984 the rapid obsolescence of some specialist equipment by Adler caused City Transport to request an investigation into the practicality of balancing tickets sold against money collected on a spreadsheet. LOTUS 1-2-3 templates were created and documented for use, and everyone who was to use the package was trained by end-user support. Southampton City Council had not, at that time, adopted a micro-database package for general use. A database package may have been a better tool to work with for this application.

2.3.2 After 1987.

By April 1988 several changes had been made to the organisation of end-user support. To begin with the name of the support group had changed to Small Systems as it was felt that this title reflected more accurately the nature of the work done.

The user groups that had been set up with enthusiasm initially (2.3.3) dwindled into apathy. It was decided by the Information Strategy group that they need no longer be held in the lunchtime.

A group drawn from managers of several departments (the IS Liaison Group) were to vet potential new systems with regard to compatibility with strategy and existing products.

The range of products supported had grown to incorporate networking for several departments, the evaluation of an expert system shell,and an E-Mail/Office Automation system.

Rather than charging out time for jobs as they are done, other ways of charging for services were offered: there was still the standard rate per hour for work done; there was a fixed sum per annum or there was a combination of a smaller fixed rate per annum coupled with a lower rate per hour for the actual work done. This third option was not popular. Most users opted for the fixed charge per annum. This charging could provide a mechanism for determining the acquisition of further staff (Appendix H).

The task of looking after hardware was combined with the role of supporting software.

APL was still used but not encouraged, continuing the trend noted (2.3.2.)

There were now 120 to 130 PC's in use within the council. With the advent of the PS/2 family of machines, the users were being encouraged early on to swap to them, rather than continue to use the 5.25 inch disks. External drives and data migration facilities were offered in order to assist the transfer. There was still an in-house policy to tend towards IBM equipment.

2.3.3 1985 to 1987.

These years were chosen to detail last as the Problem Logs were made available for these years. From three years of work there were many projects and tasks, it is not the role of this thesis to describe them in detail, nor even to simply list them. What has been attempted is to pick out a few illustrative tasks. During this time several small systems were written, usually by borrowing someone from another team.

One of the first projects in 1985 was to continue to evaluate an end-user database (2.3.2,b,v). There was a particular application in mind: the Port Health Authority wanted to log cargoes on ships. Over the years a relationship had built up with the County Council, and they allowed someone from Southampton City Council to spend a few hours

evaluating dBASE III. The principle was used that if support can easily get reasonable results from just trying the package, then users would be able to with training and support. Within half-an-hour data had been input and extracted. Continuing experiments proved that the specific enquiries requested could be successfully performed, and some tests were run to gauge the effect of size on the running of the system. It may have been better to try out several packages, but this was a market leader at the time, and seemed a reasonable choice as it appeared to fulfill the analysed requirements.

Part of the problem in providing end-user support was the lack of established procedures such as there were in conventional development. It was important to support to know what others were doing. Southampton City Council end-user support joined the London Information Centre User Group, organised by IBM. It provided a forum for exchange of ideas and discussion, as well as an opportunity to find out about latest products. It was comforting to know that other organisations were using dBASE III, for example.

Another activity performed by End-User Support in Southampton City Council was to arrange demonstrations of equipment, as a laser printer and the IBM 6750, an electronic typewriter/printer. The ICL One-per-Desk was evaluated, as well as the Olivetti M24 which was

adopted.

During this period, user groups were set up, at the request of some of the users. End-user support chaired the meeting, and organised equipment and rooms, but the provision of the topics was determined by the users themselves. Perhaps they wanted to know about a new release, perhaps one user would demonstrate his/her system or a workshop session was arranged. They were held over the lunch-time, in users' own time.

Technical Support, in conjuction with Support-X had set up a method of controlling access to inverted files. Because of this the data could be quite private, or controlled access could be allowed. Inherent to this was the idea that data could be 'owned' by a department, therefore, if another department wanted access to this data, it could be allowed, but only with the permission of the data owner, and with the knowledge of audit. Users were beginning to realise the advantages of access to data via end-user computing resources.

At the beginning of 1986 a second member joined the support team on a permanent full-time basis, Support-Z. The split of the work was along the lines of that suggested by Martin [5]. Support-Y fronted the section acting as a consultant, Support-Z was naturally more orientated to the technical side.

A major problem during this time was the inexplicable crashing of the 5520 shared logic wordprocessor, that was attributed to using PC's to emulate a 5523 workstation. Help was sought. IBM were helpful from the 5520 end, but approaching the problem of the emulating PC's was deemed to be the province of dealers. Dealers did not have experience of connecting PC's to 5520's. The result was an impasse.

This is a negative note on which to leave this chapter. It does perhaps convey a feeling of the frustrations experienced by end-user support. There are some ironies to the job. One such irony, is to ask the user to explain the problem to you. Whilst they are explaining to you, the answer clicks. They think you're wonderful, but they found the answer themselves.

Models were produced working from experience gained at Southampton City Council, based on the diaries and problem logs. However, before these models are examined in detail, there is a need to study methods used to develop these models.

References.

[1] S. Winkler.
'The Information Resource Center: A Strategy for End User Support.'
CH2022-2/84/0000/0504 1984 IEEE

[2] F. Devadson.
'Manpower Planning for a Library or an Information Center: a micro computer based operational and instructional systems'.
Application of Microcomputers in Information, Documentation and Libraries. Proceedings of the Second Internation Conference. Baden-Baden. West Germany, 17-21 March 1986. (Amsterdam, Netherlands, North Holland 1987) p 354-361.

[3] J. Rush
'Automation Support for the Management Function in Libraries and Information Centres'
Application of Microcomputers in Information, Documentation and Libraries. Proceedings of the Second Internation Conference. Baden-Baden. West Germany, 17-21 March 1986. (Amsterdam, Netherlands, North Holland 1987) p 339-346.

[4] J. Martin
'Applications Development without Programmers'
p 302-303

Prentice-Hall

[5] J. Martin
'Applications Development without Programmers'
p 306-307
Prentice-Hall

[6] H. Carr
'Managing End User Computing'
p 160 ff
Prentice-Hall

[7] J. Martin
'Applications Development without Programmers'
p 236
Prentice-Hall

3. SYSTEM DYNAMICS and STELLA.

3.1 Introduction.

From the idea that systems are inherent in any business activity, there must be a system behind End-User Support. The work is dynamic in the sense that varied requests for work [1] come in at a generally uneven rate, when viewed on a day-to-day basis. When viewed over a longer term patterns can be seen to develop, but there is a need to reflect the dynamical nature of supporting end-users. System Dynamics provided an opportunity to model support (3.2). After investigating several tools for implementing the models (3.3) STELLA was selected as the most useful (3.4).

3.2 System Dynamics.

Expanding the idea that there must be a system behind end-user support, there must also be several factors influencing the way in which support can be provided, and the way in which the influences affect the system over time. Therefore a way of modelling is needed. System Dynamics provides a mechanism for modelling. It provides a means in describing the influences on a system, and the way in which they interact with each other. A model of the system is produced, based on the influence diagram and experiments are made using various figures. As a result of running these models,

it often becomes apparent that the influences used were not the ones that altered the system, and that there were other influences on the system. The modeller therefore returns to the influence diagram. The process is iterative.

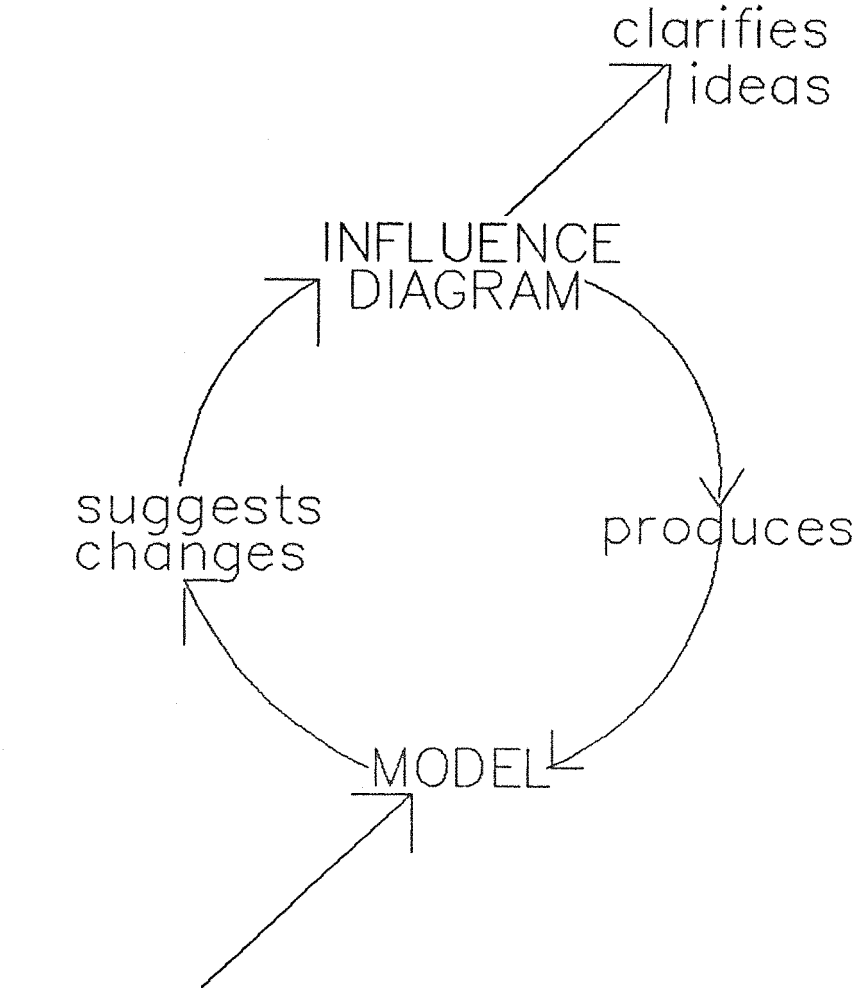


FIGURE 3.1
The Modelling Process

Of course there are other choices for modelling support, but System Dynamics was used as it reflected a reasonable approach to the problem of modelling end-user support's answering of questions. The problem has to be looked at over a period of time; the variables in Systems Dynamics change with respect to time. The variables in System Dynamics represent component parts of a system, and to analyse any system one needs to examine the component parts of a system and their interconnections. Systems Dynamics enables the modeller to look at the way in which the components influence each other.

To begin with lists are drawn up of factors within the systems and their relationships to each other are mapped. The column in Figure 3.2 headed LIST is a list of the main system components. Underlying these activities are events that immediately affect this list of components. This is the FIRST EXTENSION to the list. The events in this first extension are, in turn, affected by events themselves, the SECOND EXTENSION, and these could be influenced by other variables. It is possible to produce several extensions to the list. System Dynamics convention begins the list on the right-hand-side of the page, and works towards the left with each subsequent extension. Finally, arrows are drawn to indicate the influences of the variables on each other.

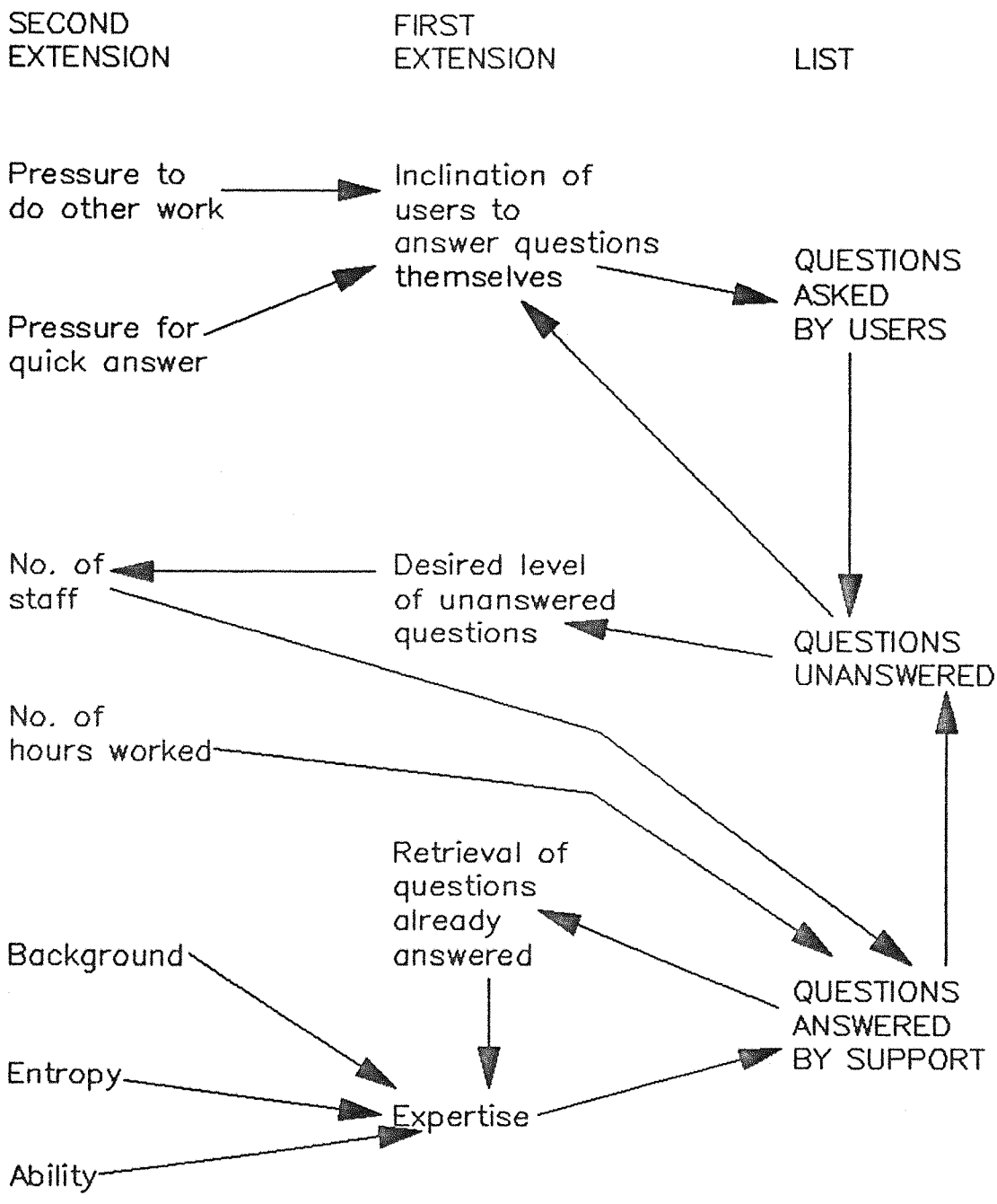


FIGURE 3.2
LAUNDRY LIST

It becomes apparent that loops are emerging creating positive and/or negative feedback loops.

From these feedback loops Influence or Causal Diagrams emerge, showing cause and effect relationships between the component elements.

A '+' by an arrow head indicate that the influence causes both elements to move in the same direction: the more questions asked by the user, the more questions there are unanswered. Conversely a '-' by an arrow head implies the opposite: eg the more questions answered by support, the fewer questions remain unanswered.

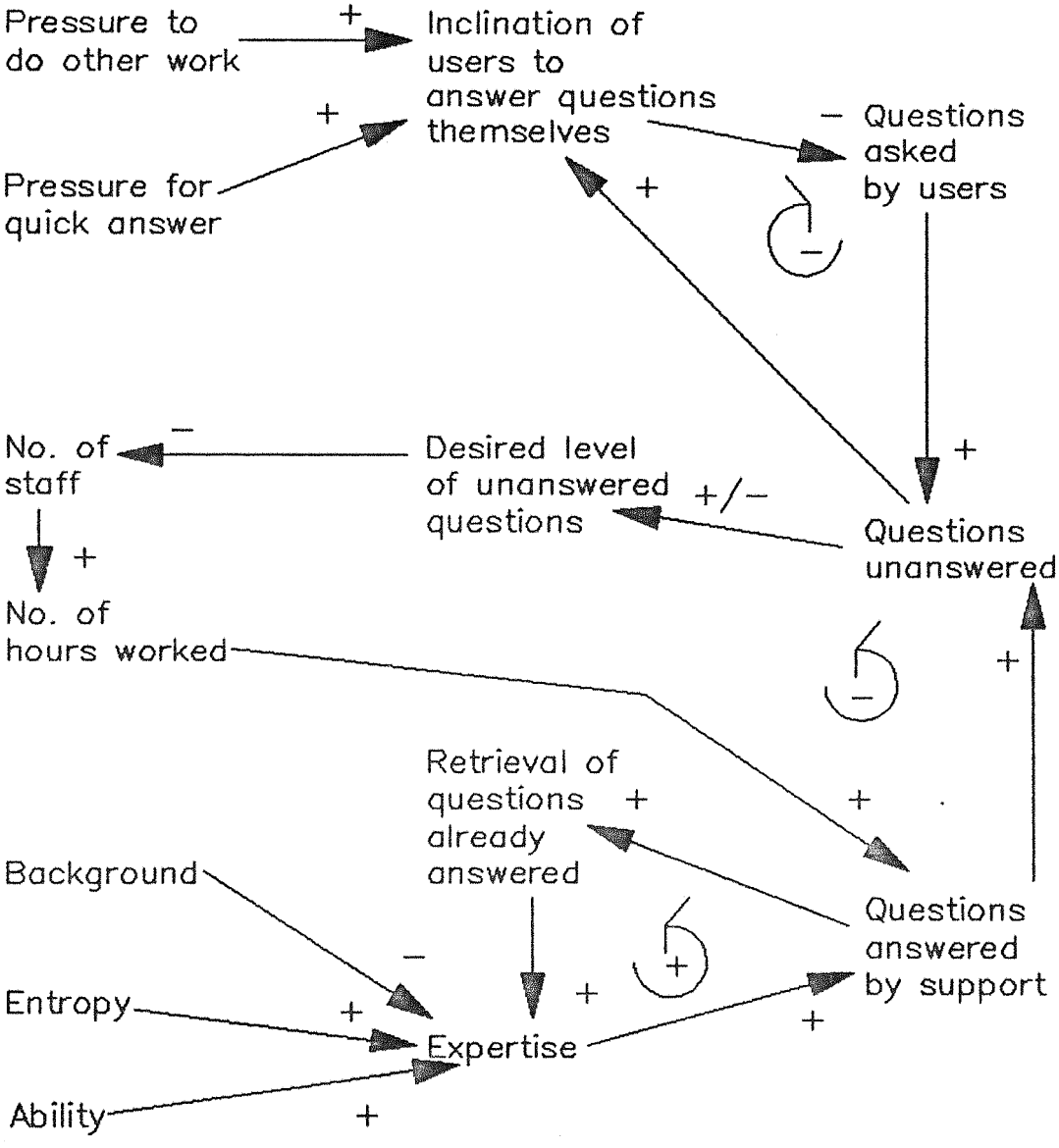


FIGURE 3.3
INFLUENCE DIAGRAM

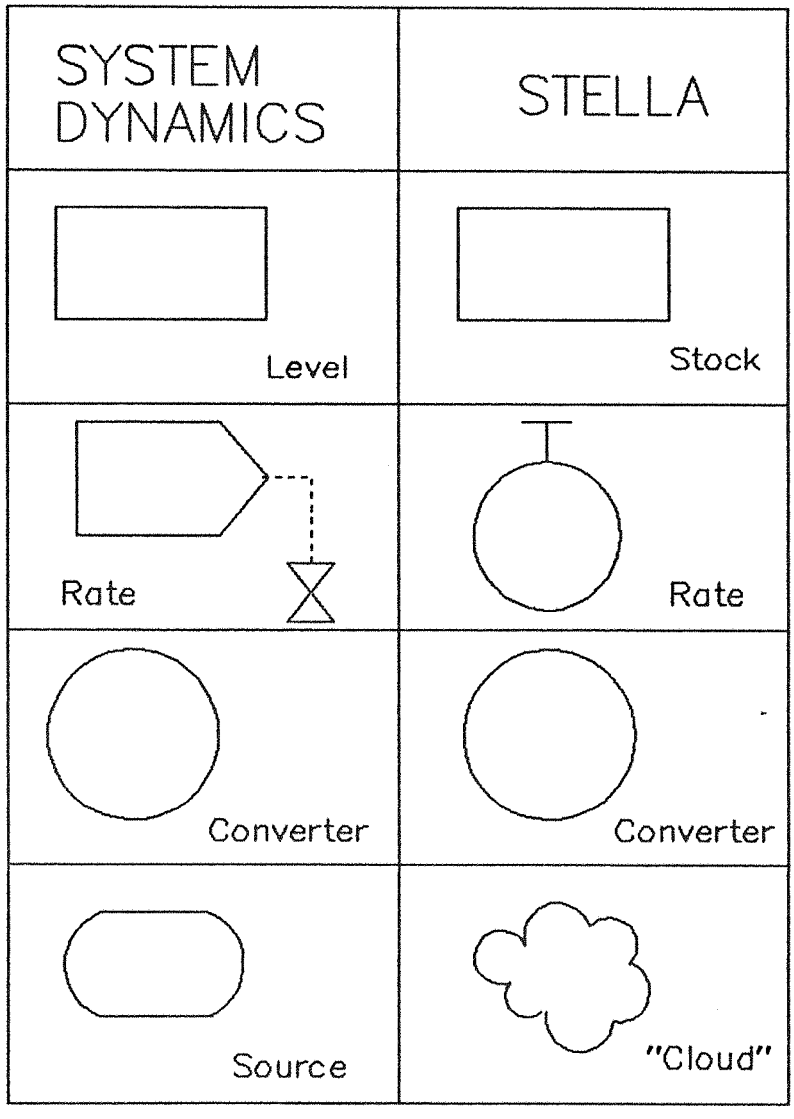


FIGURE 3.4
SYMBOLS AND NAMES

Some of the factors emerge as accumulations (levels or stocks). In the user-support problem, there is a stock/level of unanswered questions. In another system, it could be the stock of knitted jumpers, waiting to be sold. Some variables are flows in or out of these levels/stocks. These flows represent a rate of change to a stock/level. In the case of end-user support a rate could be the rate at which questions are asked. Alternatively it could be the rate at which jumpers are sold. Thirdly there are converters which enable the representation of other information within the model. For example it enables the modelling of the percentage of questions duplicated within the system. In the alternative system, one could use a converter to alter the size of knitting needle used: the bigger the needle, the quicker the jumper is knitted.

The symbols and names conventionally used by System Dynamics are shown in the left-hand column of Figure 3.4. Stella uses different symbols and names. These are shown in the right-hand column.

Finally one is ready to produce a model. Taking the Influence Diagram a model is drawn and subsequently run. A simple model is drawn in Figure 3.5. A worked example is given in Appendix A.1.

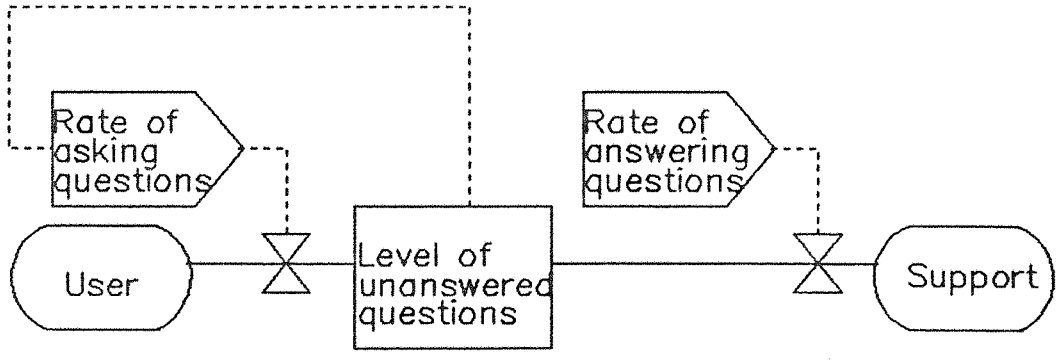


FIGURE 3.5
AN EXAMPLE SYSTEM DYNAMICS MODEL

For futher reading on System Dynamics you are referred to the Bibliography.

It was quickly apparent that it would be too cumbersome and potentially inaccurate to produce the models by hand. A tool was needed to automate the process. Although Stella was the package eventually chosen, several options were considered.

3.3 A History of Experiments with System Dynamic Tools.

3.3.1 By Paper, Pencil and Rubber.

To begin with the models were developed by hand, (an example is given in Appendix A.1.) It was soon apparent that they would be too cumbersome, lengthy and too prone to errors for this to be a serious method of producing System Dynamic models.

3.3.2 LISP.

As part of another research project a framework was written in LISP to enter variables, constants and formulaes.

The variables etc were entered using LISP syntax. It proved to be difficult to use the syntax correctly, without undergoing a training in LISP.

The framework itself was at an experimental stage and there needed to be more flexibilty in terms of the output, graphs were needed as was a neat tabular output. (Appendix A.2) It was not deemed an effective/efficient use of resources to continue along this route as there were System Dynamic packages commercially available.

3.3.3 LOTUS 1-2-3

The nature of the output suggested graphs and tables. Spreadsheets handle graphs and tables. Models were set up in LOTUS 1-2-3 (see Appendix A.3) and reasonable tables and graphs were produced. However it was apparent that LOTUS did not reflect the dynamic nature of the problem, neither would it cope easily with more complex models. This again pointed to the necessity of using a package especially designed for use with System Dynamics.

3.3.4 COSMIC

Professor Coyle at Shrivenham was instrumental in developing a System Dynamic Package called COSMIC. He kindly demonstrated the package using an end-user support example the switching from one source of support to another. (This is shown as modelled in in its final form in STELLA in sections 4.3 and 5.3.) Although COSMIC handled the

problem well the package did not display all the characteristics that were now emerging as required from the modelling tool:

i) conformity to System Dynamics rules

ii) diagramatic representation of the model

iii) the ability to easily input the underlying mathematics coupled with the package being able to make elementary deductions about the nature of some of the formulae

iv) user control over clear graphical and tabular output

v) the ability to automate the model. This need is discussed by Fishwick [2].

It fulfilled (i) but did not enable (ii), and did not appear to fully fulfil (iii) and (iv).

3.3.5 STELLA.

The next package to be tried was STELLA, running on an Apple Macintosh.

After following the manufacturers instructions for making a backup copy of Stella, the program is loaded by inserting the disk into the drive. Double click with the mouse on the Stella disk icon and the Stella

window will load. Select Stella or the required model, again with a double click. If working with diskettes one is asked to insert the original disk before proceeding, as part of the copy protection of the software. Once in Stella, System Dynamics icons, on the right-hand side of the screen, are selected and positioned by use of the mouse. The mouse can be used to move the icons around. There is a set of pull-down menus at the top of the screen which are used to change windows, print, save etc. To input formulae, one selects the icon with the mouse, clicks and is presented with an associated screen in which to add the information. It does appear to satisfy he criterion outlined in 3.3.4. A more critical review follows.

3.4 STELLA.

This package was found to be easy to learn and and straightforward to use.

The first impression was of a sophisticated package able to create the underlying equations from the diagram by making some elementary assumptions. For example, it recognises flows in and out of a level to create a fundemental equation for that level. A change in the direction of flow is mirrored instantaneously by a change in the associated equation. By using menus and pointers, entering equations is very quick.

3.4.1. Good points noted.

i) When a connection is made between two symbols, the second symbol changes colour if it is a vaild connection. This is a simple, visual confirmation that you have successfully connected two icons, and is an improvement on some C.A.S.E. tools. It does not happen if you make an illegal connection, such as trying to join two levels together.

ii) If you alter the direction of the flow (say from a level instead of to it) there is an automatic corresponding change to the underlying equations.

iii) When naming icons spaces automatically convert to underscores to satisfy naming conventions. It is more natural to type spaces than underscores.

iv) There is an expert and a novice mode.
eg. Expert: <command> + <N> to move name
Novice: selection from menus.

v) A question mark appears on an icon to show that you still have an equation to complete.

vi) When you complete the underlying equations the influences are listed and you must include

all of them in the equations. If you do not you are warned as you attempt to leave the screen.

3.4.2 Problems noted.

i) There are several instances when the documentation does not correspond with menu names on the screen.
eg. Define Selection is in Edit menu on the screen and Define menu in the manual.[5].

ii) The command key did not appear to retain the selected structural element, as mentioned in the manual [3].

iii) There is a revert-to-save menu option which enables a recovery from a bad mistake. Like all computing, it is good practice to make regular saves. However it seemed fairly simple to 'save' your mistake, rather than 'revert-to-save'.

iv) The manual described the editing options, rather than giving examples to work through step by step, as it did in the first part of the tutorial. Therefore, in general, the first part of the tutorial appeared to be more effective than the second part.

3.4.3 Experimenting with models.

Before work can begin on modelling, some ground rules have to be established. For example, the scale chosen was integers as the timescale that would be worked in would be days, weeks, or months. Ultimately months were chosen, as this was the natural unit that Southampton City Council worked in. The drawback to using months was that months do not have the same number of days: February has 20 working days, some have 22.

Parts 2, 3 and 4 of the Stella manual [6] were worked through, applying the theory to examples from End-user Support. Each generic process was modelled with an example from end-user support and so was each generic subsystem.

a) Every model is made up of a combination of building blocks called generic processes:

i) External Resource Production
when some external influence provides a basis for producing a flow and when you want to show this influence. For example, the intervention of Personnel in setting staff limits.

ii) Compounding
whenever you want a level to feed upon itself. The characteristic behaviour pattern is one of compounded growth. For example, a compounding

fraction can be used to show that the more questions answered the quicker one gets at answering questions.

iii) Draining
used to show the draining, passive decay or aging of levels. In End-user support this could represent the loss of motivation of those providing support.

iv) Stock Adjustment
to represent a flow of activity which adjusts a stock to a desired level. For example, the artificial backlog of questions in order to dampen user expectation of the speed of service. (4.4, 5.4, the variable backlog).

v) Implicit Goal Seeking
used whenever you want to represent a flow of activity that adjusts a level towards an implicit goal. Typically it creates a relationship between two conditions. In End-user support, if the rate of questions asked is too small, and there is a charging policy, then perhaps an adjustment could be made to the charging policy.

vi) Co-flow
is an abbreviation for 'coincident flow' when the flow is an artifact of another flow, then the two flows are identical exept for a conversion co-efficient. An example from End-user support is of the flow to two different supports. (4.3, 5.3)

vii) Ordering
the general rule is one generic process per flow, but there will be times when you want to enable different processes to interact with each other. In End-user support an example could be the blending of the factors influencing the use the users make of support with the factors influencing the provision of support.

For a fuller description of the generic processes you are referred to the Stella user manual [6].

Where the examples have been incorporated into worked models, the associated reference is given above. For simplification of the models the setting of limits by personnel (i) was not included. A simple compunding (ii) was felt not to enable analysis within a model. The draining (iii), loss of motivation, was regarded as outside of the scope of the thesis as it is behavioural. Charging policy (v) was not explored, again for reasons of unnecessary complication of models, similarly the joining of elements to make one large model (vii) was not incorporated at this stage of the research. All of these could be taken further in additional research (7).

b) These generic processes combine to make generic sub-systems:

i) S-shaped growth.

ii) Overshoot and collapse.

iii) Oscillation - sustained oscillation
- dampened oscillation
- expanding oscillation.

Again you are referred to the Stella manual [8] for fuller descriptions.

Many experiments were conducted to force End-user support into the above patterns. It was possible to push the results to conform to these patterns, but, other than the intellectual fun of doing so, it was not helping to clarify understanding of the end-user support situation. This approach was therefore shelved, and work was begun on the eventual models.

To gain an appreciation of the number and type of questions asked the Southampton City Council Problem logs were put onto dBASE III to facilitate analysis. Then the modelling of end-user support began in earnest.

References.

[1] H. Carr.
'Managing End-User Computing'
p 98
Prentice-Hall

[2] P. Fishwick
The role of process abstraction in Simulation.'
p25
IEEE Transactions on Systems, Man and Cybernetics
Vol 18, No 1, Jan/Feb 1988.

[3] B. Richmond, S. Peterson, P. Vescuso
'An Academic User's Guide to STELLA'
p 14
High Performance Systems

[4] B. Richmond, S, Peterson, P. Vescuso
'An Academic User's Guide to STELLA'
Chapter 2, pp 9 to 34
High Performance Systems

[5] B. Richmond, S. Peterson, P. Vescuso
'An Academic User's Guide to STELLA'
p 19
High Performance Systems

[6] B. Richmond, S. Peterson, P. Vescuso
'An Academic User's Guide to STELLA'
Parts 2,3 and 4. pp 45 to 199

High Performance Systems

[7] B. Richmond, S. Peterson, P. Vescuso
'An Academic User's Guide to STELLA'
Chapter 8. pp 99 to 111
High Performance Systems

[8] B. Richmond, S. Peterson, P. Vescuso
'An Academic User's Guide to STELLA'
Chapter 9. pp 113 to 127
High Performance Systems

4. THE MODELS.

4.1 Introduction.

In Chapter 2 we looked at the provision of End-User Support, now the models themselves are considered.

All of the following models are based on this underlying concept:-

> From time to time users ask questions of End-user Support about problems they have using packages and micro-computers. End-user Support has many roles, one of which is to answer the questions, correctly, when asked. If the rate at which the users ask questions is greater than the rate at which End-user Support can answer them, then a queue of unanswered questions builds up.

For the models I am assuming that:-

a) Each question is of approximately the same level of difficulty.

b) The questions come in at a steady rate.

c) The questions are taken on a 'first come first served' basis.

d) End-user Support answers questions at a steady rate.

e) The objective is to keep the queue of unanswered questions as close as possible to zero questions outstanding.

These assumptions have been made in order to keep the models simple enough to be able to model, one of the first stages in modelling [1, 2]. However each of the models are either interesting in themselves in terms of the aspect of the problem that they illustrate, or they are interesting in the research that they could lead onto. Each of the above points could form the basis of further research (7.1 to 7.6), but it was decided to limit the scope of this exploratory work to using System Dynamics to clarify thought, rather than produce the ultimate model that brings in every conceivable nuance and variation on answering end-user questions. A football-pitch sized model does not help to clarify anything.

Working from the basic idea that users ask questions and End-user Support answers them, a series of models were built using STELLA. Several models were built and each model was experimented upon with variations of data. The models that have been included, have been chosen for their ability to illustrate a point, or because they seem to be representative of the problem. Each model began simply. Once the basis of the model was established, additional items were added to the model, until the models began to suggest answers, or

raise even more questions.

4.2 First Model : A Simple Model.

The fundamental principle that underlies all the subsequent models is that users ask questions of End-user Support and End-user Support answers the questions, correctly, in strict rotation. This is discussed in the Introduction to this chapter (4.1).

This simple model has been included as it is the starting point for the subsequent models [2] and as a means on confirming that STELLA is acting in a deterministic manner. With only two variables it is possible to mentally calculate the outcome, something that is not so easy once more variables are involved. Therefore if STELLA is behaving as expected at this level, it can be assumed with some degree of confidence that it will behave as expected with more complex models. If it does not react well at a simple level, then no confidence can be placed in it at a complex level. As far as the experiments went it performed consistently and well.

The model itself "A Simple Model" is included and described in section (5.1).

As well as confirming that Stella would be an appropriate tool for the experiments, it also confirmed some underlying principles of the model:

a) If the rate at which questions are asked exceeds the rate at which questions are answered, then a backlog of questions builds up.

b) If the rate at which questions are asked is less than the rate at which questions are answered, then End-user Support has excess capacity.

c) If the rates at which questions are asked and answered are equal then there is no queue and no excess capacity.

These principles are self-evident. To illustrate, take point (a) and apply common-sense. If you cannot answer the questions quickly enough, then of course a queue builds up.

To take point (b) further. The excess capacity mentioned should be made use of. If the primary objective of user support is answering questions then steps should be taken to improve the efficiency/accuracy of the work. One way is to 'play' (to experiment with packages using non-critical or hypothetical examples as opposed to using the package for a critical, reasonably urgent task) with packages. Leslie R. Port [3] discusses the significance of 'play' from the end-user angle. If users benefit from 'playing' then so must End-user Support. If End-user Support uses this time to develop applications for themselves or for users then they gain the benefit of using this package in an actual task and so push it

further than they would if merely 'playing' with it to see what it can do.

However it does not address the question of what happens when the queue builds up. This is modelled in 'Alternative Support'.

4.3 Second Model : Alternative Support.

Users would not continue asking questions indefinitely if they are getting no reply for what can prove to be weeks. They would either find an alternative means of End-user Support, or they would stop using the computer altogether.

Let us assume that users continue to use the computer and that they do not stop needing help just because End-user Support cannot answer questions fast enough. The users turn to a second source; a dealer, an expert within their department or the organisation generally, or even answer the questions themselves.

So now users have two options, Support-A or Support-B. Let us assume that Support-A is the original means of End-user Support and the one favoured by corporate policy, however an external source of End-user Support (external to Support-A, not necessarily external to the company) may be used.

The users would not continue asking the same rate of questions of Support-A indefinitely. From how long it

has been taking recently to obtain an answer they would know how long they have to wait for an answer. In 'Alternative Support' (5.2) the length of the queue indicates how long users would have to wait for an answer. So there is a link from each of the queues to the rate at which questions are split between the two queues. Once the queue to Support-A reaches a certain length fewer questions are asked until the queue drops sufficiently to restore user confidence in Support-A, and the users return to asking of Support-A the original rate of questions.

Support-B also has a limit on the number of questions it can answer, and its queue also grows. If users have to pay more for the service Support-B provides their tolerance may be lower for Support-B than Support-A, so it is possible that users swop back quickly to Support-A, causing the queue to gradually grow over the months.

So, 'Alternative Support' shows a switching mechanism from one End-user Support to another. It is possible that the user checks the queue size before placing the question with one or other End-user Support. What is more likely is that sometimes the user becomes fed-up waiting and switches the question to the other End-user Support without informing the original End-user Support that this has been done. As a result, although there may only be 100 questions in the system, there are in effect 110 because some have been duplicated. There is therefore an element of wasted effort. This is reflected in the model which is

included in section (5.2) together with a description and some results.

Not only could wasted effort arise from the duplication of questions, it could be that the answer itself may no longer be relevant; the user has answered the question himself or found another way around the problem, or the deadline for the answer has now passed. The onus is on Support-A to provide an effective and efficient service to try to prevent this duplication occurring. 'Support' investigates the interaction of variables to enable End-user Support to provide a useful service.

4.4 Third Model : Support.

The main area of concern are the factors influencing the ways in which End-user Support can answer questions more effectively and efficiently. Section (5.3) 'Support' includes an Influence Diagram as well as the model 'Support' and a description.

The rate at which questions are asked by the users has been kept at a steady rate. Although various factors affect the rate at which questions are asked, (for example, user expertise) they are regarded as outside of the boundaries of this thesis (7.7). For the purposes of this model, it is assumed that homogenous questions come in at a steady rate.

In the preceeding models it has been assumed that the objective is to keep the queue down as close as possible to zero. This may not be in the best interests of user-End-user Support.

a) It may have been found in the past that if users receive an answer almost instantaneously, then End-user Support is deluged by trivial questions. This does not make the best use of End-user Support's scarce resources. (It must be assumed that user-End-user Support does not have infinite resources.) Suppose the users have to wait, say, half-a-day for an answer, then they will look more readily for an answer themselves.

b) Similarly, if most of the questions are answered within one hour, then user expectation is raised to expect an hourly response. If End-user Support then takes half-a-day to answer a question (which is not unreasonable) users complain of a poor service.

This has been included in the model as an acceptable backlog. This acceptable_backlog in turn affects the rate that staff can be added to the End-user Support team. If more people are available to answer questions, then more questions should be answered. It is possible to add another member of the team to work full-time as a permanent member of staff. Which is practicable until demand drops.

a) An alternative is to 'borrow' someone from another team to make use of their slack time from another

project. It does take time for someone to become a fully effective member of the team (a further sophistication of the model would be to have differing, increasing rates for new End-user Support-team members until they become fully operational (7.8)).

b) End-user End-user Support can be an excellent training ground as interpersonal skills are gained from the experience of working with user-End-user Support. Carr [4] discusses the characteristics required of End-user Support personnel.

In this model it has been assumed that there is one permanent member of staff, and any number of additional people can be added half-a-person at a time on a temporary basis. It would be of value to pinpoint the set of circumstances for appointing second and subsequent permanent members of staff (7.9 and Appendix H). For example, one such circumstance could be a sustained growth in questions asked.

A short-term response to the increase in the queue is to increase the amount of flexi-time worked. This is limited by the inevitability of taking time off later in order to redress the balance. The larger the End-user Support team, potetially the more flexibility in hours worked. It is modelled as an additional numbers of questions answered rather than being expressed in terms of hours worked. Practically the actual additional hours worked could be spent on other work, thus freeing time during the normal working day

for dealing with users.

It is unlikely, and undesirable, that End-user Support will spend all of its time answering questions. Other work is done, such as writing applications, training, consultancy [5]. The relationship that additional work has to the the ability to answer questions effectively and efficiently needs exploring further (7.10). Here it is a device to note that whilst other work is being done, questions are not being answered, and this results in a time split between answering questions and carrying out other tasks.

This model is only the tip of the iceberg, at least nine-tenths of the research is still to be done. Only three main areas have begun to explored in this model;-

a) the affect of additional staff on the rate at which questions are asked,

b) the rate at which other work done takes time away from answering questions,

c) the use of flexi-time to give a degree of flexibility in the rate that questions are answered,

Suggestions for additional research are included in Chapter 7 'Conclusion and Future Work'.

4.5 Summary.

These three models can only be regarded as starters. The more complex the model (ie. the more rates, levels and variables) the harder it is to isolate the cause and effect of actions. It is easier to conceptualise the system if it is partitioned (as in the theory behind Structured Systems Analysis [6]), but the real world works in a series of complex inter-relationships. The relationship of the models with the real world is discussed in Chapter 6, but first the models and some results follow in Chapter 5.

References.

[1] B. Richmond, S. Peterson, P. Vescuso
'An Academic Users Guide to Stella'
High Performance Systems

[2] P. Fishwick
'The Role of Process Abstraction in Simulation'
p18
IEEE Transactions on Systems, Man and Cybernetics, Vol 18, No 1, Jan/Feb 1988

[3] Leslie R. Port.
'Managing the diffusion of End-User Computing Techniques: a Fifties Mindset with Eighties Tools'.
p 65
from 'Managers, Micros and Mainframes. Integrating Systems for End-Users.'

[4] H. H. Carr
'Managing End-User Computing'
pp 94-100
Prentice-Hall

[5] H. H. Carr
'Managing End-User Computing'
pp 100-112
Prentice-Hall

[6] T. de Marco
'Structured Analysis and System specification'
p 82
Yourdon Press

5. RESULTS.

5.1 Introduction.

This chapter contains printed output from the models (diagram, equations, graph, and table) together with discussions of the equations used, and the models as a whole.

5.2 The First Model : Simple Model.

5.2.1 The Equations used in the First Model.

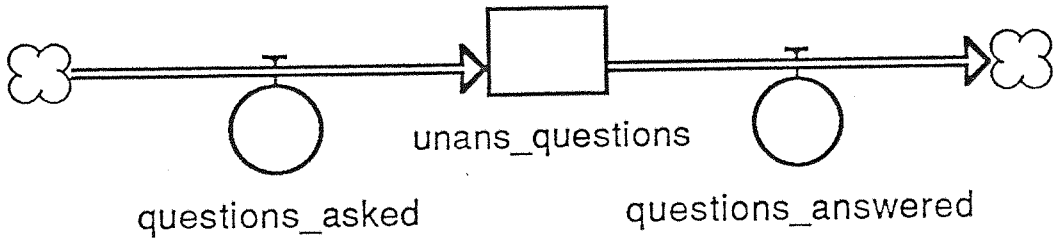
□ unans_questions = unans_questions + dt *
(questions_asked - questions_answered)

The first part of the equation for the level unans_questions is generated by Stella. Working from the diagram, the package notes the input to the level, (questions_asked,) and subtracts the output (questions_answered). Stella takes into account the passage of time (dt) and accumulates the results. All the levels have this part of the equation automatically generated.

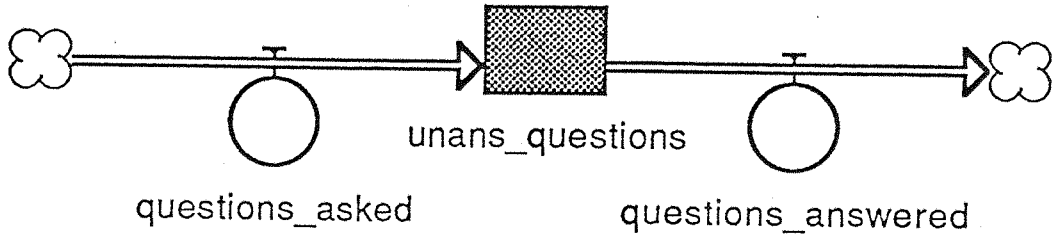
MODEL 1 : A SIMPLE MODEL

The Diagram, before the run.

Figure 5.2.1



MODEL 1 : A SIMPLE MODEL
The Diagram, after the run.
Figure 5.2.2



MODEL 1 : A SIMPLE MODEL

The Equations

Figure 5.2.3

unans_questions = unans_questions + dt * (questions_asked - questions_answered)

INIT(unans_questions) = 0

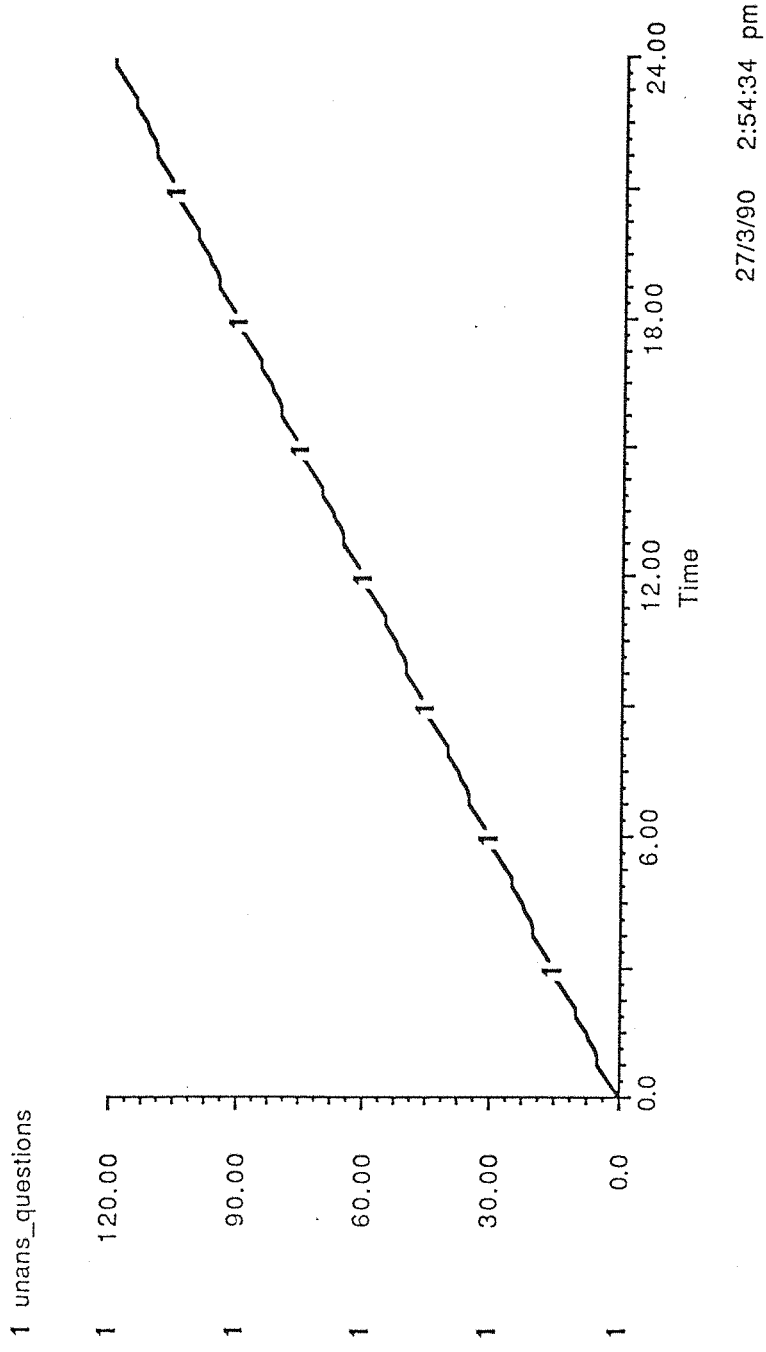
questions_answered = 25

questions_asked = 30

MODEL 1 : A SIMPLE MODEL

The Graph.

Figure 5.2.4



MODEL 1 : A SIMPLE MODEL

The Table.

Figure 5.2.5

Time	unans_questions	questions_answered	questions_asked
0.0	0.0	25.00	30.00
1.00	5.00	25.00	30.00
2.00	10.00	25.00	30.00
3.00	15.00	25.00	30.00
4.00	20.00	25.00	30.00
5.00	25.00	25.00	30.00
6.00	30.00	25.00	30.00
7.00	35.00	25.00	30.00
8.00	40.00	25.00	30.00
9.00	45.00	25.00	30.00
10.00	50.00	25.00	30.00
11.00	55.00	25.00	30.00
12.00	60.00	25.00	30.00
13.00	65.00	25.00	30.00
14.00	70.00	25.00	30.00
15.00	75.00	25.00	30.00
16.00	80.00	25.00	30.00
17.00	85.00	25.00	30.00
18.00	90.00	25.00	30.00
19.00	95.00	25.00	30.00
20.00	100.00	25.00	30.00
21.00	105.00	25.00	30.00
22.00	110.00	25.00	30.00
23.00	115.00	25.00	30.00
24.00	120.00	25.00	30.00

INIT(unans_questions) = 0

This second part of the equation, initialising the level, is set by the modeller. Zero has been chosen, as, in this particular case the assumption is that end-user support is starting up. If the model is exploring the effect of a backlog, then this figure could be set to 100, 50, or whatever the backlog is deemed to be.

questions_answered = 25

The rate at which questions are answered has been set to 25 per unit of time (monthly) for this model. It is purposely lower than the rate that questions are asked in order to illustrate the way in which a queue of unanswered questions can build up. No record was kept at Southampton City Council of the rate at which questions were answered, however from experience it was just under the rate at which questions were asked. It is within the range of averages for questions asked 1985 to 1987, and tends towards the lower end, as the rate at which questions were answered was lower than the rate at which questions were asked.

The rate at which questions were asked lay between 5 and 93 (Appendices B) between 1985 and 1987. The averages per year ranged from 18.25 and 53. Whilst 30 is a nice round number to use, it is feasible as it is within the actual figures recorded at Southampton City Council.

5.2.2 The First Model as a whole.

The graph shows a steady increase of 5 questions per day added to the backlog of unans_questions. This is also shown by the table in the unans_questions column. It is what is expected, given that the questions are answered at a slower rate than they are asked.

This is a model of the problem at its highest level of definition. To go any higher would not say anything about what is being modelled. To summarise what is being modelled:

> Users ask questions of support. Support answers the questions. Sometimes support answers questions too slowly and a queue builds up. Sometimes support answers the questions too quickly and has excess capacity. Sometimes the rate at which

support answers questions and the rate at which questions are asked coincide.

There are three elements: users, a queue and support. From experience at Southampton City Council you need these three elements.

a) Users are needed to generate work. Without users there is no need for support. Southampton City Council was not the only organisation with end-users. IBM organised the London Information Centre User Group, of which Southampton City Council was a member. Delegates from many organisations in the South of England attended these sessions. Houston H. Carr argues the existence of end-users and their corresponding need for support [1]. References to end-users are made in contemporary literature [2,3,4,5,6,7].

b) Again drawing on practical experience at Southampton City Council, there was always a queue of outstanding work. There is a well-documented and acknowledged backlog of work for traditional development [8]. Switching from traditional development to end-user development can be seen as a way of by-passing the queue. Jobs which would not normally be requested as they would have been regarded as too trivial to ask traditional development teams to do [8,1] now crawl out of the woodwork. All of which add to the queue of work to be done.

c) Finally the organisation succumbs to demand and sets up an end-user support function. [1].

An extract of data recorded at Southampton City Council in the yearly Problem Log is included as Appendix D. The date the questions were received were recorded as well as the recipient of the question. The problem itself was recorded. From this can be derived the rate at which questions were asked. It is represented in the model as a monthly flow as this is how it was regarded within Southampton City Council. The time taken to answer the question, nor the time the reply was given was recorded. Therefore the rate at which questions were answered can only be surmised. From recollection, most questions were answered within one day, with many being answered on the spot. The rate at which questions were answered could perhaps be the focus of the collection of more data (7.11). In some cases the answer to the question was also recorded. The asking rate of 30 questions per month given in the model is slightly lower than the average of 34.08 during 1984 (Appendix B.1). Figures for 1985 and 1986 are given in appendices B.2 and B.3.

5.3 The Second Model : Alternative Support.

5.3.1 The Equations used in the Second Model.

□ questions = questions + dt * (basic_question_rate - questions_to_a - questions_to_b)

The level questions accumulates the questions asked as they come into the system. Stella automatically generates this first part of the equation from the diagram. It has noted that there are two flows going out of the level, represented by the rates questions_to_a and questions_to_b. Similarly, it has noted the one flow in, basic_question_rate.

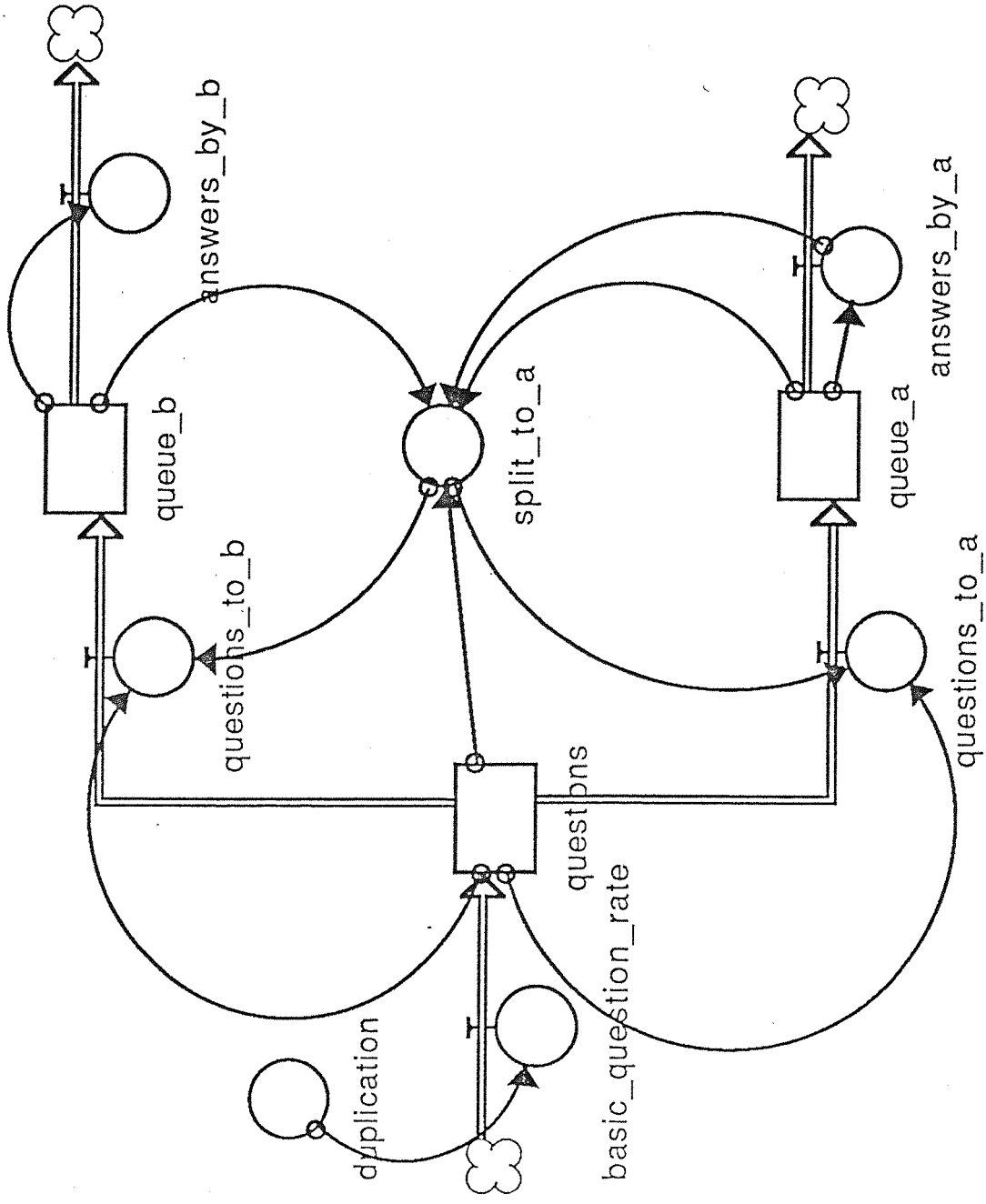
INIT(questions) = 100

The second part of the equation has been set up by the modeller. On this occasion the level has been initialised to 100, which is the same as the basic_question_rate. This is in order to show activity from the very start of the model. It may have been more natural to set it to zero, on the premise that it is a new system, and one is unlikely to find an existing collection of questions as soon as the service is begun. However, it has been suggested [1] end-user

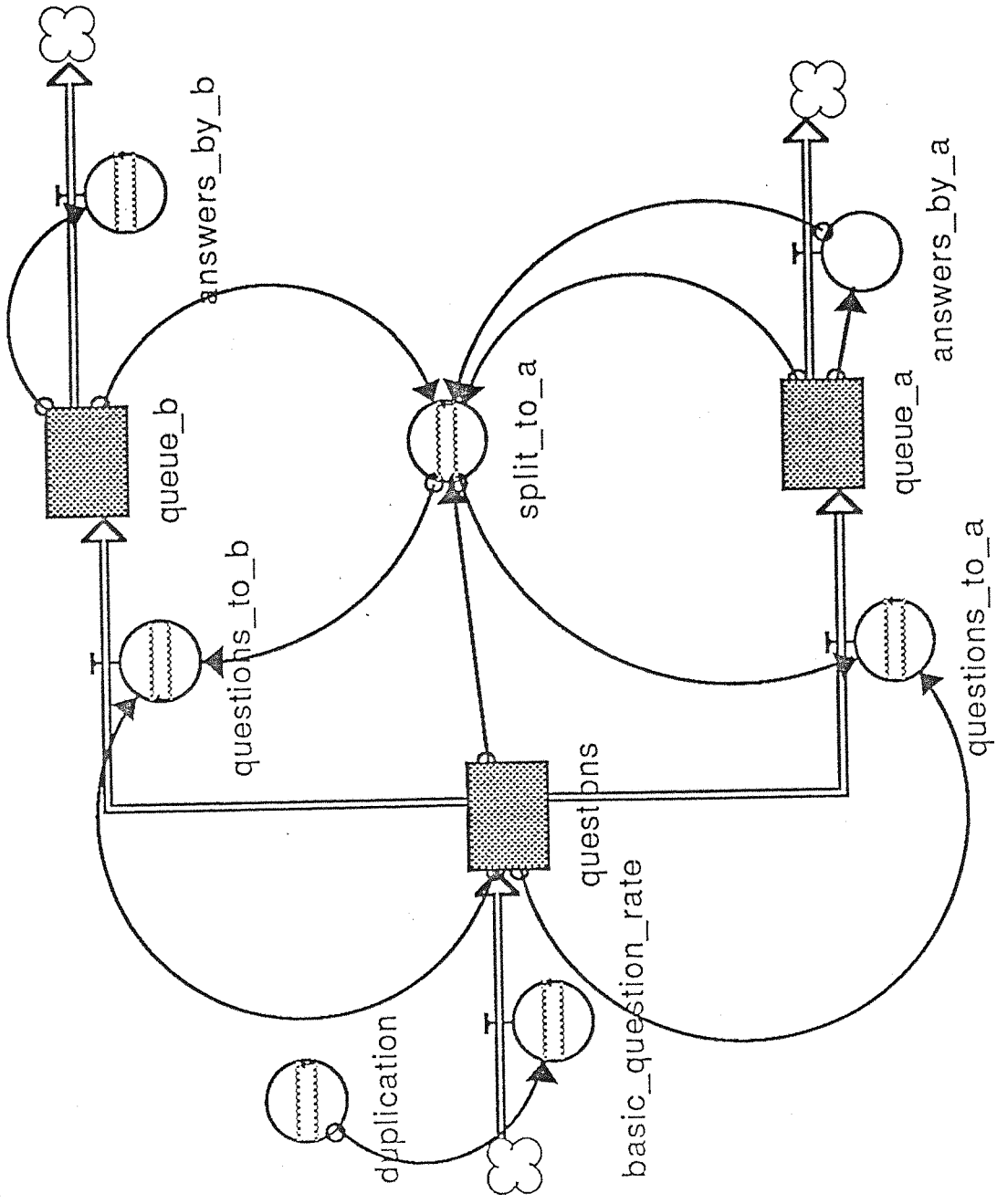
MODEL 2 : ALTERNATIVE SUPPORT.

The Diagram, before the run.

Figure 5.3.1



MODEL 2 : ALTERNATIVE SUPPORT.
The Diagram, after the run.
Figure 5.3.2



MODEL 2 : ALTERNATIVE SUPPORT.

The Equations.

Figure 5.3.3

□ questions = questions + dt * (basic_question_rate - questions_to_a - questions_to_b)
INIT(questions) = 100

□ queue_a = queue_a + dt * (questions_to_a - answers_by_a)
INIT(queue_a) = 0

□ queue_b = queue_b + dt * (questions_to_b - answers_by_b)
INIT(queue_b) = 0

○ answers_by_a = IF DT=1 THEN 30 ELSE IF queue_a>30 THEN 30 ELSE queue_a

○ answers_by_b = IF queue_b > 15 THEN 15 ELSE queue_b

○ basic_question_rate = 100*duplication

○ duplication = IF TIME>5 THEN 1.1 ELSE 1

○ questions_to_a = questions * split_to_a

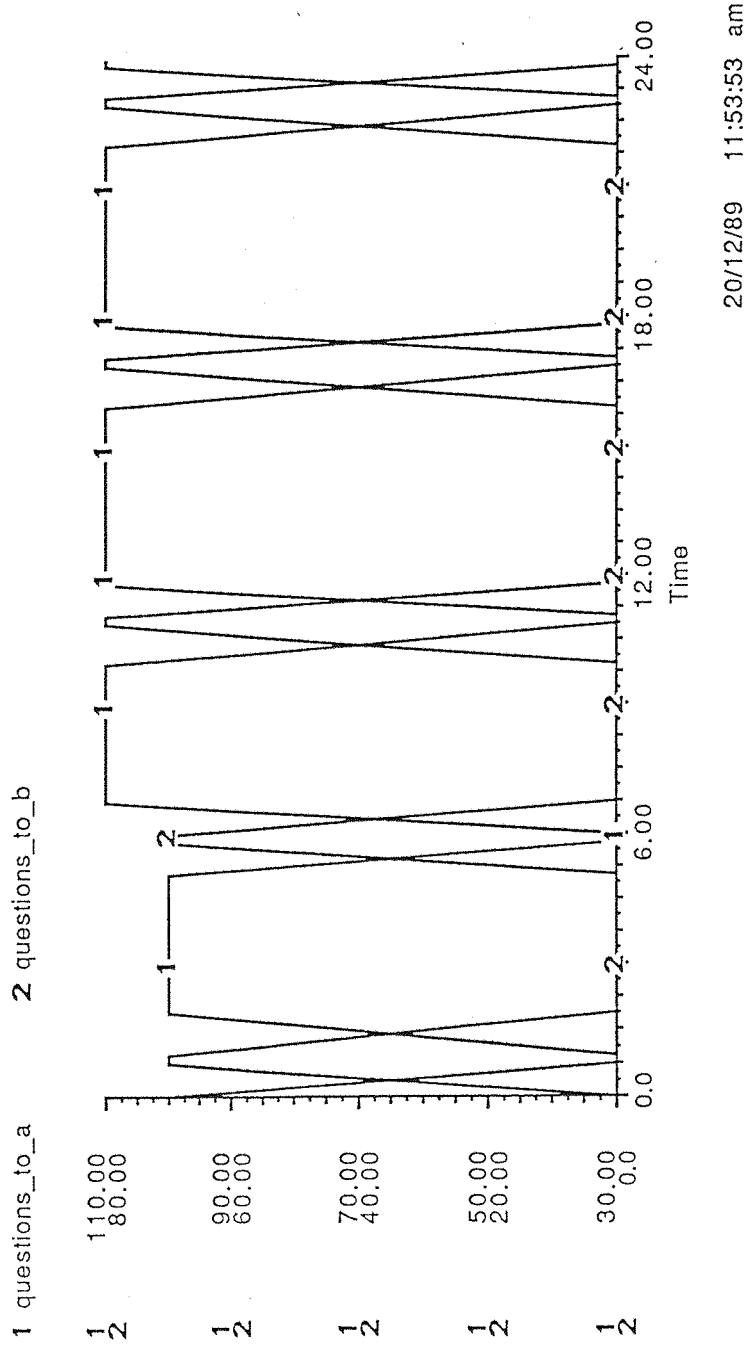
○ questions_to_b = questions * (1-split_to_a)

○ split_to_a = IF queue_b > 10 THEN 1 ELSE IF queue_a > 0 THEN answers_by_a / questions ELSE 1

MODEL 2 : ALTERNATIVE SUPPORT.

Graph 1.

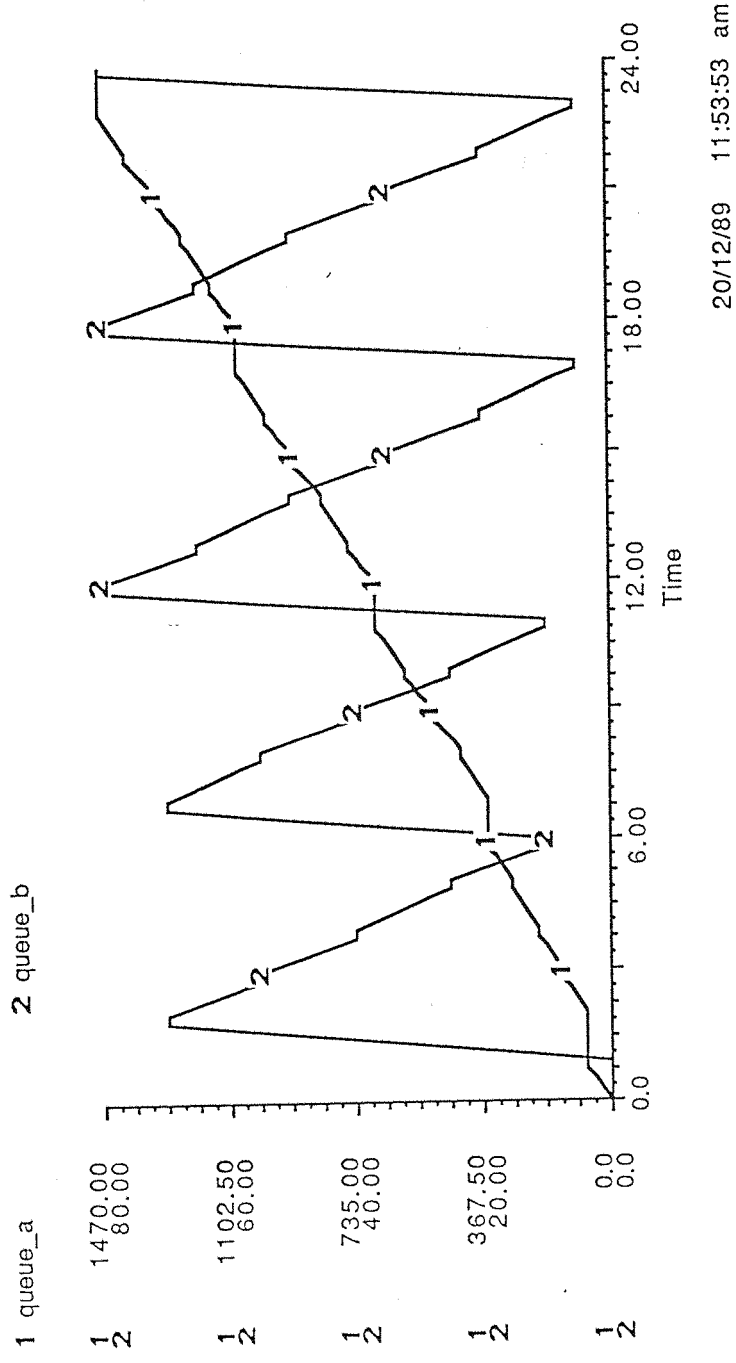
Figure 5.3.4



MODEL 2 : ALTERNATIVE SUPPORT.

Graph 2.

Figure 5.3.5



MODEL 2 : ALTERNATIVE SUPPORT.

The Table

Figure 5.3.6

Time	queue_a	queue_b	questions_to_a	questions_to_b	answers_by_b
0.0	0.0	0.0	100.00	0.0	0.0
1.00	70.00	0.0	30.00	70.00	0.0
2.00	70.00	70.00	100.00	0.0	15.00
3.00	140.00	55.00	100.00	0.0	15.00
4.00	210.00	40.00	100.00	0.0	15.00
5.00	280.00	25.00	100.00	0.0	15.00
6.00	350.00	10.00	30.00	70.00	10.00
7.00	350.00	70.00	110.00	0.0	15.00
8.00	430.00	55.00	110.00	0.0	15.00
9.00	510.00	40.00	110.00	0.0	15.00
10.00	590.00	25.00	110.00	0.0	15.00
11.00	670.00	10.00	30.00	80.00	10.00
12.00	670.00	80.00	110.00	0.0	15.00
13.00	750.00	65.00	110.00	0.0	15.00
14.00	830.00	50.00	110.00	0.0	15.00
15.00	910.00	35.00	110.00	0.0	15.00
16.00	990.00	20.00	110.00	0.0	15.00
17.00	1070.00	5.00	30.00	80.00	5.00
18.00	1070.00	80.00	110.00	0.0	15.00
19.00	1150.00	65.00	110.00	0.0	15.00
20.00	1230.00	50.00	110.00	0.0	15.00
21.00	1310.00	35.00	110.00	0.0	15.00
22.00	1390.00	20.00	110.00	0.0	15.00
23.00	1470.00	5.00	30.00	80.00	5.00
24.00	1470.00	80.00	110.00	0.0	15.00

support is only provided if there is already an unsatisfied demand, in which case it would be acceptable to say that, from the word go, there is a pile of questions waiting to be answered.

queue_a = queue_a + dt * (questions_to_a - answers_by_a)

Again as this equation is for a level, Stella has made some assumptions about the nature of this equation. From the diagram it has been automatically noted that the queue is affected by the rate at which questions are sent to Support-A and by the rate at which Support-A can answer the questions. The whole equation is an accumulation and takes into account the passage of time, like all the levels.

INIT(queue_a) = 0

Queue-a has been initialised to zero as this model is based on the assumption that, although the questions flow in from the very start, there are no existing unanswered questions on support-A's desk.

queue_b = queue_b + dt * (questions_to_b - answers_by_b)

Queue_b follows a similar pattern to the accumulation generated in queue_a. Stella has created the equation from the diagram, noting that queue_b is influenced by the rate at which questions are asked of Support-B minus the rate at which Support-B can answer the questions. The whole takes into account the passage of time (dt) and accumulates the answer over a period of time.

INIT(queue_b) = 0

Again, Support-B is not expected to have any questions outstanding when the modelling begins, therefore it is initialised to zero.

answers_by_a = IF DT=1 THEN 30
ELSE IF queue_a>30 THEN 30 ELSE queue_a

The rate at which questions are answered by Support-A is set at a standard 30 per month, unless this causes the queue to be represented by a negative number. It has been kept constant, puposely to limit the factors influencing the model. The third model, Support, which is examined later (5.4), looks at ways in which the rate that questions are answered can be varied. 30 per month is higher than the figure used in the first model for the answering of questions, but then the rate at which questions ae asked (basic_question_rate) is substantially higher. It is within the figure obtained from Southampton

City Council (Appendices B) but is still lower than the rate at which questions are asked in order to see the patterns emerging from the switching of questions between two supports.

The rate remains steady, unless by keeping at that level, Support-A starts answering questions that are not there. This is typified in the model by the level becoming negative. Therefore there is a check on the size of the queue before the questions are answered, and the rate that questions are answered is limited by the number of questions in the queue. This reflects, perhaps a behavioural pattern in end-user support, then once the rush of questions is over, resources can be switched away from this activity to another.

The final part of the equation, so far undiscussed, is the that on the first pass, 30 questions are answered. Otherwise there will be no questions answered on the first pass, because there are no questions in the queue.

answers_by_b = IF queue_b > 15
THEN 15 ELSE queue_b

The rate at which Support-B answers questions has arbitrarily been set to 15. The rate at which alternative sources of support answer questions could be the subject of further research (7.12). If it may be accepted, as previously argued, that

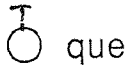
30 is a reasonable rate for the main source of support to answer questions, then surely that rate at which an alternative support can answer questions must be less? The alternative source of support will have other work to do. If the source is a dealer, then the dealership will have other clients. If the alternative source is the liaison officer, then this role is often a proportion of the their total time available at work. Therefore it is argued that a reduction in the rate that questions are answered is rational. Halving the rate seems to be appropriate.

Like answers_by_a support-B cannot answer questions that have not been asked, therefore Support-B is limited by the size of the queue of outstanding questions.

The basic_questions_rate has been set at 100. According to the figures from Southampton City Council (Appendices B) this is high, but in April 1986 93 questions were asked, so it is not outside of the realms of possiblility. The setting of the rate needed to be high enough to send questions to both Support-A and Support-B. The duplication is discussed next.

○ duplication = IF TIME>5 THEN 1.1 ELSE 1

It would be perfect if, when users changed support, they notified the original support that they no longer required an answer. The real world is not perfect. What happens is that the user simply places the same question with the alternative source of support, forgetting to mention it to the original. Duplication models this by duplicating some of the questions within the system. It is suggested that there is a time delay before this somewhat chaotic behaviour emerges, perhaps somewhere in the region of 5 months. There were no actual figures recorded at Southampton City Council, but from experience 10% seems a reasonable proportion, from the number of times Support comes up with the answer, only to be told that it is no longer needed. 10% may be a trifle high, but it is used to represent this type of activity within the organisation.



The rate at which questions are sent to Support-A depends on the number of questions that there are in the system and the way in which they are split between the two alternative sources of support.

questions_to_b = questions * (1-split_to_a)

The comment above also applies to the rate at which questions are sent to Support-B.

split_to_a = IF queue_b > 10 THEN 1 ELSE
IF queue_a > 0 THEN answers_by_a / questions ELSE 1

It was decided to make the model more sensitve to the level of unanswered questions outstanding in queue_b as the alternative source of support is usually more expensive, therefore users will be more critical of the service provided by Support-B. Let us assume that the alternative source of support is a Dealer. At the time that Southampton City Council was making an internal charge of £15 per hour, one dealer was charging £40 per hour. Therfore queue_b is checked first. If the level of unanswered questions is unnacceptable (ie. the users would have to wait too long for a reply, in this instance just under a week) then all the questions would go to Support-A (split_to_a = 1). Otherwise Support-A receives all the questions it can answer in a week (30) and the rest go to Support-B.

5.3.2 The Second Model as a Whole.

The graphs start to show a developing pattern.

Graph 1 plots the rate at which questions are asked of the various areas of support. Please note the different scales on the graph; a maximum of 110 questions are sent to Support_A whilst 80 questions are sent to Support_B. A steady oscillation builds up. This is confirmed by the figures given on the table. The differing scales have been used in order to emphasise the pattern. It is possible to set the scales to the same range.

Graph 2 plots the effect on the two queues. The length of queue_a builds up steadily, because of the bias towards in-house support as outlined in (4.3). The length of queue_b oscillates, again please note the different scales on the y-axis.

Note the increase in month 5 as a result of the duplication of questions within the system.

At Southampton City Council there were instances when the users went outside of end-user support as provided by the Compter Section. Sometimes with its blessing, sometimes without:

a) Audit were a source of advice to users with suggestions as to the appropriate use of end-user computing.

b) Liaison Officers were encouraged as a first port-of-call for their own departments. the accountants became expert in the use of LOTUS 1-2-3 and were a source of support for end-user support itself on occasion.

c) Another source of support was the company grapevine. Council officers shared lifts in, caught the same train in, ate together in tha canteen etc. They shared ideas. For example a menu written in BASIC by one department emerged around the building at various points.

d) Dealers usually preferred to have one point of contact within an organistaion. From time to time users approached them directly. They provided word-processing training.

e) Southampton Institute of Higher Education ran a series of courses for and on behalf of Southampton City Council.

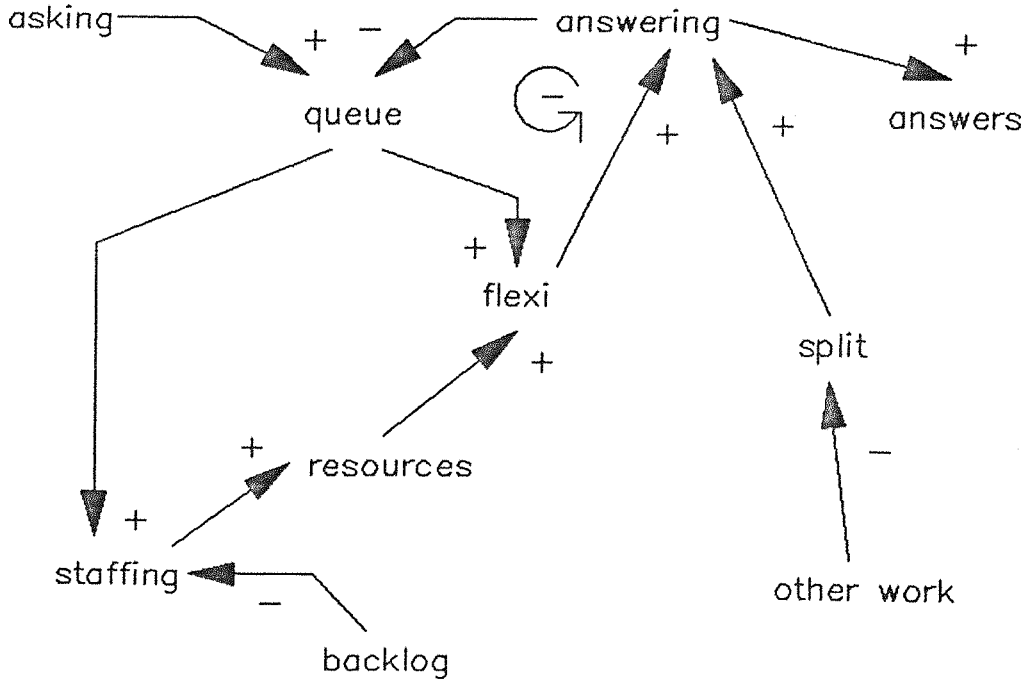
Users, therefore, at Southampton City Council at least, had access to support other than that formally provided by the Computer Section.

This model suggests that users may become fed-up waiting for a response from the primary source of support and go elsewhere.

5.4 The Third Model : Support.

5.4.1. Support Influence Diagram.

This leads us to the main exploration of this thesis: the factors influencing the ways in which support can answer questions more efficiently and effectively. The third and final model therefore, concentrates on some of the factors inluencing the way in which support answers questions.



To begin with the fundamental starting point for all of the models: users ask questions of support, and a queue builds up if support cannot or will not answer questions quickly enough.

This time, however, a pool of answers builds up. This pool of answers could provide another starting point for reasearch (7.12). If the queue builds up more staff can be added to support, so that there are more people doing the job. This is limited by an backlog,(it may be deemed necessary to make users wait before answering the question.) The larger the backlog the slower the rate that staff(ing) are added to the resources of the support team. The bigger the queue and the greater the resources the more flexi time is worked. Support has other work to do. the more other work is done, the smaller the split to answering questions.

There is one negative feedback loop around queue, flexi and answering. The more questions that are answered the smaller the queue, which in turn decreases the the amount of flexi-time worked, which means that fewer questions are answered, so the queue grows again ... . This loop acts in a short time scale. It is reactive. Support sees the queue growing so works more flexi-time as a front-line defence. A longer term solution is to add more staff.

There are two limiting factors.

a) The acceptable backlog. This enables the addition of staff only as a positive response to the size of the queue. So user-expactation of the service can be controlled.

b) Other_work. It should be expected that support does other_work. If other_work is done, this limits the time available to answer questions.

5.4.2. Equations used in the Third Model.

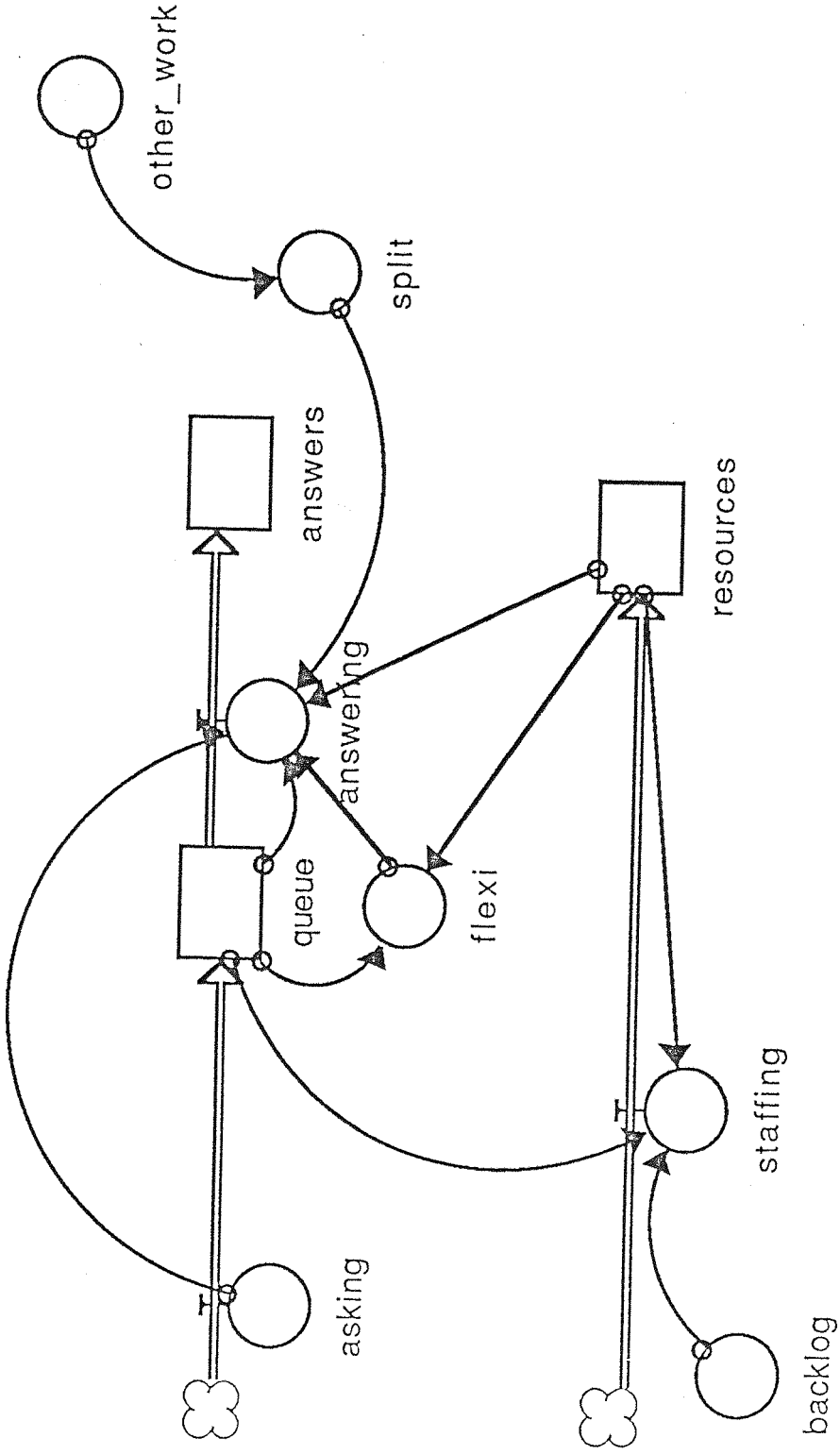
□ answers = answers + dt * (answering)

The equation for the level answers was created by Stella. It is a straight accumulation, over the months, of all the answers given by Support.

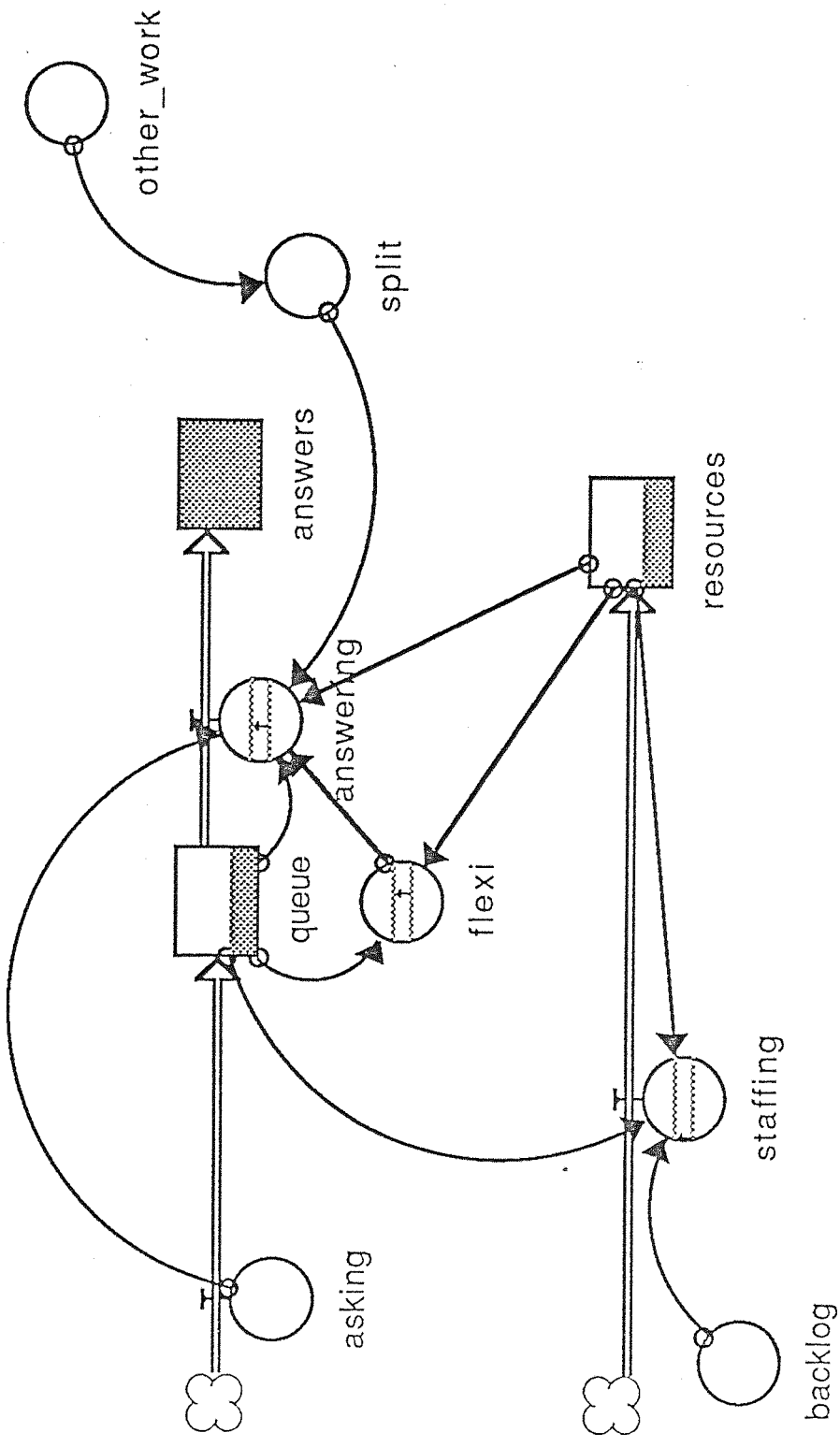
INIT(answers) = 0

It was initialised to zero signifying that the system was beginning from nothing. If this level is represented in the real world by a database of answers, it would be opened by the first answer given. This is not to say that support has no answers at all to their fingertips, it is just

MODEL 3 : SUPPORT.
The Diagram, before the run.
Figure 5.4.2.



MODEL 3 : SUPPORT.
The Diagram, after the run.
Figure 5.4.3.



MODEL 3 : SUPPORT.

The Equations.

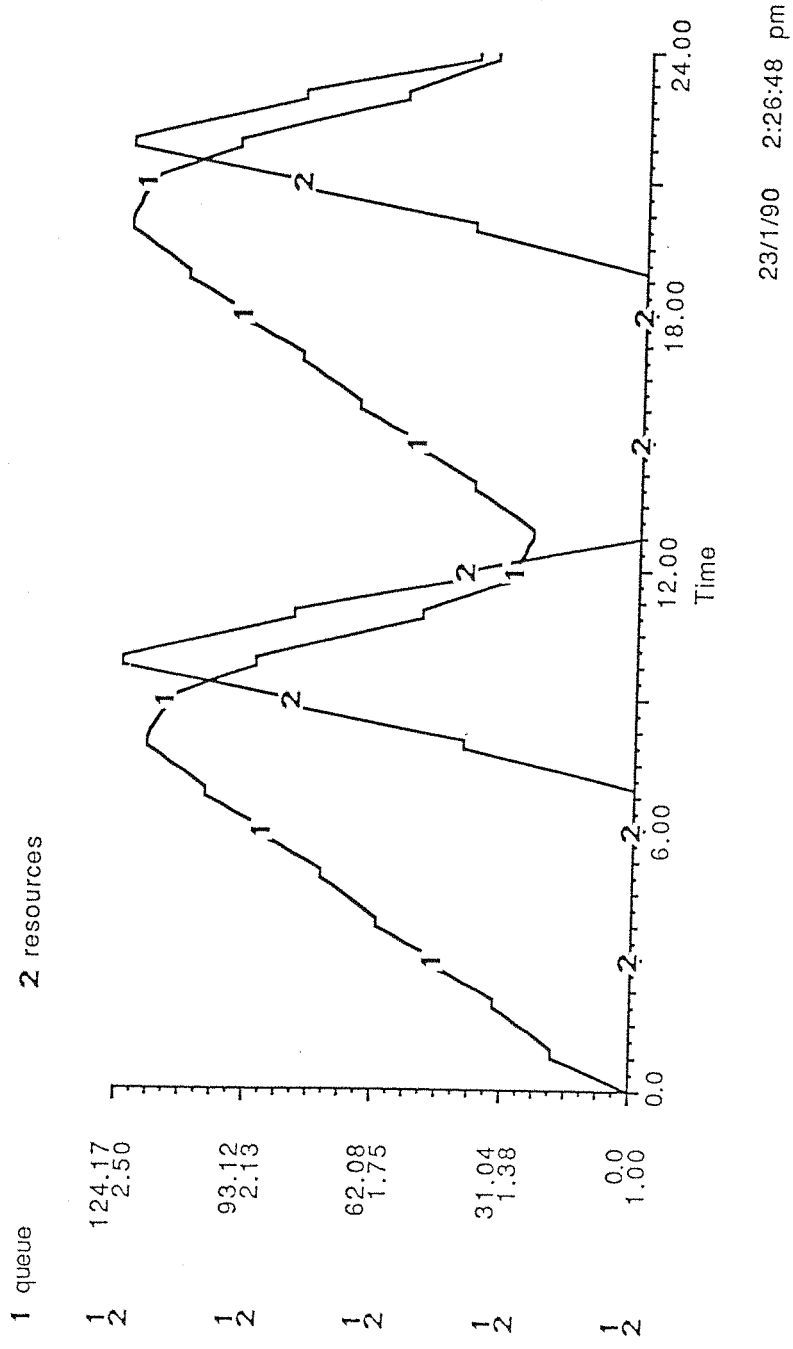
Figure 5.4.4

□ answers = answers + dt * (answering)
  INIT(answers) = 0
□ queue = queue + dt * (asking - answering)
  INIT(queue) = 0
□ resources = resources + dt * (staffing)
  INIT(resources) = 1
○ answering = IF queue +asking - (50 *resources*split+flexi) < 0 THEN queue +
  asking ELSE 50*resources*split+flexi
○ asking = 50
○ backlog = 100
○ flexi = IF queue > 10 THEN 5 * resources ELSE 0
○ other_work = 8
○ split = 1 - other_work / 21
○ staffing = IF queue > backlog THEN .5 ELSE IF resources -.5 < 1 THEN 0 ELSE -.5

MODEL 3 : SUPPORT.

The Graph.

Figure 5.4.5



MODEL 3 : SUPPORT.

The Table

Figure 5.4.6

Time	queue	resources	answering	asking	staffing
0.0	0.0	1.00	30.95	50.00	0.0
1.00	19.05	1.00	35.95	50.00	0.0
2.00	33.10	1.00	35.95	50.00	0.0
3.00	47.14	1.00	35.95	50.00	0.0
4.00	61.19	1.00	35.95	50.00	0.0
5.00	75.24	1.00	35.95	50.00	0.0
6.00	89.29	1.00	35.95	50.00	0.0
7.00	103.33	1.00	35.95	50.00	0.500
8.00	117.38	1.50	53.93	50.00	0.500
9.00	113.45	2.00	71.90	50.00	0.500
10.00	91.55	2.50	89.88	50.00	-0.500
11.00	51.67	2.00	71.90	50.00	-0.500
12.00	29.76	1.50	53.93	50.00	-0.500
13.00	25.83	1.00	35.95	50.00	0.0
14.00	39.88	1.00	35.95	50.00	0.0
15.00	53.93	1.00	35.95	50.00	0.0
16.00	67.98	1.00	35.95	50.00	0.0
17.00	82.02	1.00	35.95	50.00	0.0
18.00	96.07	1.00	35.95	50.00	0.0
19.00	110.12	1.00	35.95	50.00	0.500
20.00	124.17	1.50	53.93	50.00	0.500
21.00	120.24	2.00	71.90	50.00	0.500
22.00	98.33	2.50	89.88	50.00	-0.500
23.00	58.45	2.00	71.90	50.00	-0.500
24.00	36.55	1.50	53.93	50.00	-0.500

that there is no formal repository of answers when the system starts.

queue = queue + dt * (asking - answering)

The queue is built up over time as an accumulation of the differences between that rate that questions are asked and the rate at which questions are answered. This part of the equation, is as usual generated by Stella.

INIT(queue) = 0

As the system is deemed to be starting afresh, the level of the queue is initialised to zero.

resources = resources + dt * (staffing)

The third level is that of resources: the number of staff. This equation, determined by Stella from the diagram simply reflects the rate at which staff are added or taken away over a period of time.

INIT(resources) = 1 |

It is initialised to 1, as there must be one person to provide support, before support is provided. Although staff may be added a half at a time (see below) the underlying assumption is that there must always be one whole member of staff supporting the users.

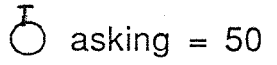
answering = IF queue +asking -
(50 *resources*split+flexi) < 0 THEN queue +
asking ELSE 50*resources*split+flexi |

The rate answering is affected by many influences.

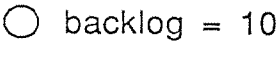
In order to never answer more questions than there are available in the system, support is limited to the number of questions in the queue plus the number of questions asked at that point in time, which is why it is linked to both asking and queue. This is a sophistication above that offered in the Second Model, which takes into account the size of the queue only, not the questions coming in as well. In this model, a calculation is performed in order to see that the queue does not become negative, before the actual calculation takes place. A further sophistication would be to link the chacking to the rate asking, rather than hard-code it in.

Another influence taken into account is the proportion of time allocated to answering questions (split). How split is derived is

discussed below. The more resources (staff) thaere are, the more questions that can be answered. This is reflected by including resources as a mulitiplier in the equation. The final influence in this equation is the amount of flexi-time worked. It is argued that this is a further mechanism for increasing the number of questions answered.



The rate that questions are asked is set at 50 in this model. This is within the figures noted at Southampton City Council, and is large enough to give scope to the experiments.



The backlog is set to 100. this is the acceptable backlog of questions, which is a mechanism for controlling user expectations. Until the queue reaches this figure then the response is deemed reasonable. In this instance it has been assumed that two months waiting for an answer would not try the patience of the users. This may be a little extreme, but it does enable the mechanism to be seen working within the model. If the time scale was read in terms of days, then double the input rate would not seem so bad. However, it has

been set at two months in order to illustrate the affect it has on the model.

It has been set high for another reason. The adding of staff is a major decision, because of all the employment implications. therefore management must be reasonably sure that the demand is there before employing additional staff, or removing staff from other projects.

○ flexi = IF queue > 10 THEN 5 * resources ELSE 0

The working of additional hours under flexi-time is more flexible. The queue has only to grow slightly for the support staff to react and increase the hours worked. It is shown in the system as a multiplication of the number of staff (resources) that there are. If one person answers 50 questions a month, and may work an extra ten hours each month (in a normal month 147 hours are worked. 21 * 7 = 147, an average of 21 days per month, 7 hours per day.) 10 is one fourteenth of the total hours per month, 5 is one tenth of the questions answered per month, then an extra 5 questions per month per support staff if flexi-time is worked is not unreasonable, given that the effort is probably concentrated if there is a special task to do.

○ other_work = 8

It is assumed that approximately two-fifths of the time spent at work, is spent on activities other than answering users' questions. If there are, on average 21 working days per month (February has 20 unless it is a leap year, in 1990, March had 22 working days) then 8 is approximately two fifths of this time.

○ split = 1 - other_work / 21

The split takes into account the time spent on other work and calculates a fraction making the time spent on other work as a proportion of the whole. 21, for the reasons given immediately above, is taken as the average number of working days in a month.

○ staffing = IF queue > backlog THEN .5
ELSE IF resources -.5 < 1 THEN 0 ELSE -.5

This equation is where staff are added to the number of staff available to support in the form of resources. If the queue is greater than the acceptable backlog then staff are added at the rate of a half a person at a time in order to bring down the number of unanswered questions. It

is feasible to add half-a-person at a time, as one could employ part-timers or split their work between this and conventional development, say. Similarly if the queue falls under the acceptable backlog, then the additional staff could be redeployed elsewhere. No comment is made on the morale of staff constantly switched from one job to another. The only limitation on this equation is, that it always checks to make sure that at least one person remains to answer questions.

5.4.3 The Third Model as a Whole.

The diagrams and equations are included as usual.

The graph shows both the size of the queue, and the resources that used by support. (Again note the discrepancy in scales, which have been kept so as to emphasise the pattern.) Once the queue has passed the acceptable backlog of 100, more staff are added to clear the backlog. There is a quick effect on the queue which rapidly drops to zero and stabilises. Equally, the number of staff reaches a peak, then also stabilises. This is also reflected in the table.

Various factors affect the way in which support can improve its service. Many factors such as morale have been purposely ommitted from the model as they have their roots in behavioural science and are therefore outside of the scope of the

model.However there are ways of representing this type of factor.

Three major elements have been included in this model and all of these elements are of keen concern to the provider of end-user support, as they are first-aid attempts at improving the suituation.

a) A short-term solution is to increase the amount of flexi-time worked. This was done by support in July 1984. Work done was recorded in a diary, an extract is included in Appendix E. It can be compared with the increase in questions in July 1984 (Appendix B.1). Although the actual number of questions answered in that time span is not explicitly recorded, it is in the same region as the number of questions asked.

b) Another area of concern was to know when to increase the number of staff in the support area. One had to be certain that the upsurge in questions asked was a trend that was likely to continue, which is why the problem log was begun in the first instance: proof of the level of need of support, and that the service was being used. A solution was to borrow members of other teams in their slack periods.

c) The third area of concern was that of other work to be done. Although requests for small systems could be logged as a question, the work

itself had to be scheduled. This caused a conflict between providing an answering service and producing small systems. The writing of small systems was regarded as beneficial to the support function as, coincidentally, support was storing up answers for future questions. What often happened was 'thrashing', where no progress was made on work. Support would start to work on a small system, then be interrupted by a question, then return to the work, only to be interrupted by another question. The answer lies in 'time management': setting aside alloted time for different tasks. This time management is acknowledged in the model by the splitting of time between answering questions and other work done.

An attempt has been made to include these three factors in 'support'. The split denotes time allocated to other work. The temporary borrowing of other staff is reflected in staffing and resources. Flexi shows the additional questions answered by increasing the amount of the time in the office. (It could also acknowledge the flow of adrenalin in having a lot of work to do.)

References

[1] H. H. Carr
'Managing End-User Computing'
p 157
Prentice-Hall

[2] H. H. Carr
'Managing End-User Computing'
pp 61 - 81
Prentice-Hall

[3] W. B. Rouse, N. M. Morris
'Understanding and Enhancing User Acceptance of Technology'
IEEE Transactions on Systems, Man and Cybernetics, Vol SMC-16, No.16 November/December 1986

[4] J. W. Spence
'End-User Computing - the Human Interface'
Journal of Systems Management, February 1988

[5] M. Sumner, R. Klepper
'Information Systems Strategy and End-User Application Development' Data Base. Summer 1987.

[6] R. Tagg
'Software Engineering from the End-User Angle'
Computer Bulletin. December 1987.

[7] S. Winkler.
'The Information Resource Centre. A Strategy for End-User Support'
CH2022-2/84/0000/0504 IEEE 1984

[8] J. Martin
'Application Development without Programmers'
p 4
Prentice-Hall

6. DISCUSSION of RELATIONSHIPS BETWEEN REALITY AND THE MODELS.

6.1 Introduction.

Within this dissertation the objective was to use the models as a means of clarifying thought around end-user support, and in particular the way that end-user support handles the questions.

Of necessity a model is a simplification and an abstraction [1] of the real situation. If the abstraction has its foundations in reality, then it could be argued that the model is valid.

However, as noted in Chapters 4 and 5, the models are merely a starting point, the second part of this chapter discusses ways in which the models may be used and built upon. Chapter 7 takes these ideas further, suggesting paths for future work.

6.2 The Models as a Starting Point.

The First Model is an overview of the problem under investigation reduced to its simplest form. Its value is in the bounding of the problem and the setting of the problem into context. Perhaps it has a similar role to that of the Context Diagram in Structured Systems Analysis. It is submitted that a valid

approach to modelling in Systems Dynamics is to start simple at a high level and become progressively more detailed. This first model, in fact, equates with the column headed 'list' in Figure 3.2 (Chapter 3).

The Second Model grew as a result of observing a phenomenon at Southampton City Council: users had recourse to alternative forms of support. It is possible that the use of alternative forms of support could be an indication of the satisfaction of users within the service officially provided. If end-user computing tools are regarded as useful within the culture of the organisation, then help of some sort is generally required. The alternative is not to use the tools. If it is known that end-user computing tools are used within an organisation, but few questions are asked of support, this could be an indication of a poor level of support from within the official source of support. Alternatively, it could mean that the liaison officer network has worked. It would, perhaps, be interesting to explore this topic further; to find out what support is actually used by users, to find measures or indicators of how well support is performing, how users are affected by pricing mechanisms, etc. This second model begins to explore the way in which users switch between sources of support, and the way in which questions can be duplicated to the different sources of support.

The third model 'support' is the one that offers the most scope as a starting point for further development. Whilst it is argued that it is a

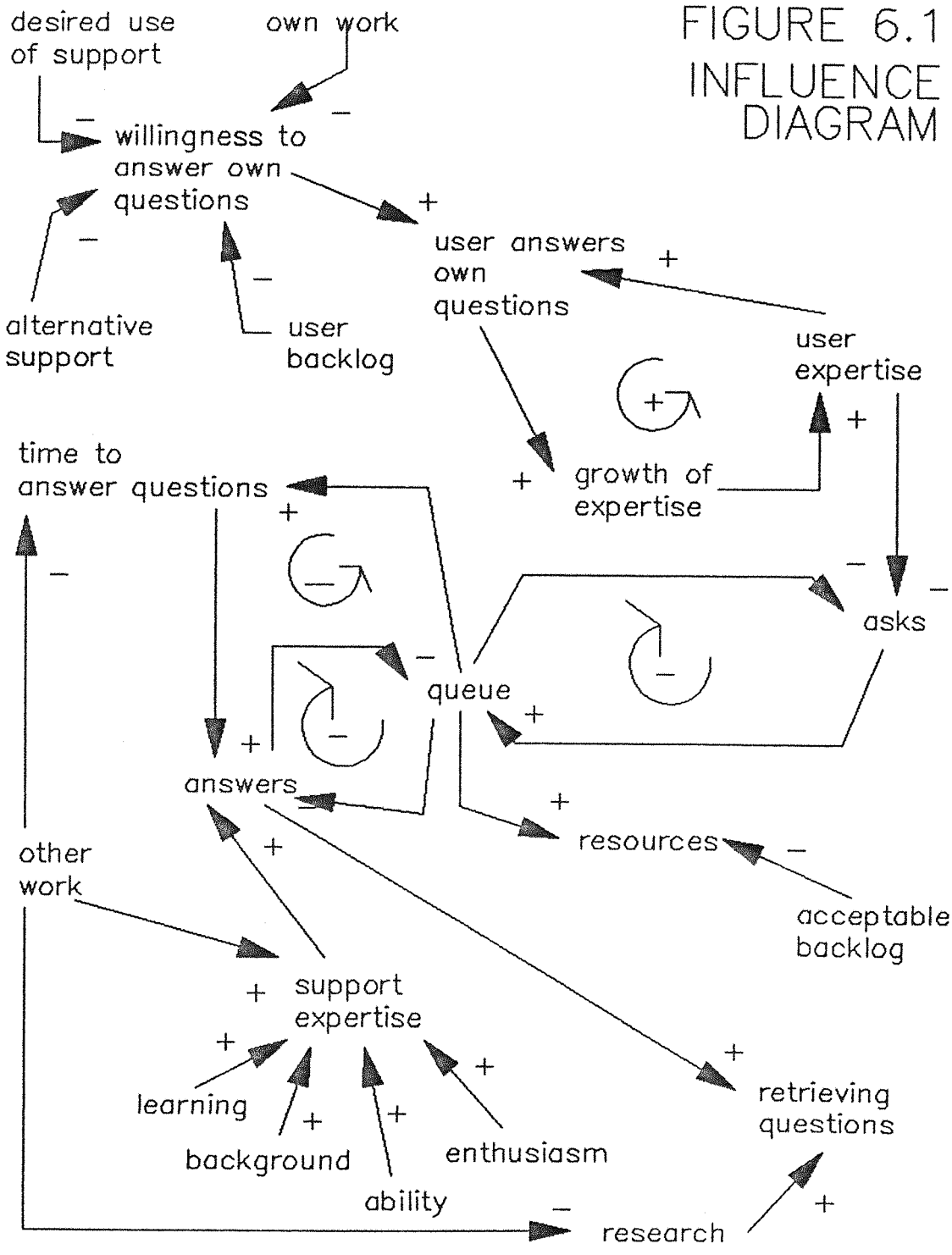


FIGURE 6.1
INFLUENCE DIAGRAM

reasonably accurate reflection of the situation as far as it goes, there are more factors to be taken into consideration, as one would expect in a model as simple as this. However it was as complex as was achievable given the timescale of this project. Therefore more experimentation needs to be done in order to use these models with absolute confidence. In fact the experimentation was taken further, as shown in Figure 6.1, but it was apparent that much work was needed to produce a robust working model. It was therefore decided to reduce the level of complexity in order to produce reasonably robust but simple models. This demonstrates the applicability of Systems Dynamics to the problem, whilst clarifying ideas on the provision of end-user support. Figure 6.1 is an Influence Diagram taking into account some additional factors as well as some already modelled. It could be further developed into a more detailed model, beginning to move towards a more realistic picture of events in end-user support. It also expands the user side, another area that must be taken into account to gain a fuller picture of the situation. The influence diagram in turn reflects the taxonomy given in Appendix G, which was the first attempt at clarification of understanding of end-user support. We turn full circle.

The models are a reasonable representation of the way that end-user support was provided, but at a high level of detail. During the experiments it became apparent that the relationship between the complexity of the model and the resources needed to

build it was not linear. In fact there seemed to be an explosion in the relationship between the complexity of the model and the resources needed to model progressively more complex models. It was necessary to put a stop at some point.The models produced were complex enough to be emulating some aspects of the real world, without taking every possible nuance into account. For example, homogeneous questions have been used. This makes the model small enough to handle, but it also enables a realistic feel to the model.

6.3. A Discussion on Aspects of Originality within this Thesis.

Within this thesis it is submitted that there is some original work: the application of the techniques of System Dynamics to the problems of end-user support. It is the application of System Dynamics that enables the problem to be considered analytically. Problems such as the provision of end-user support are generally analysed in a more ad-hoc way. It is therefore suggested that System Dynamics is an additional approach, enabling new light to be shed on the problem.

This approach also lends itself to the partitioning of the problem. If the first model produced is an overview of the problem, then it can be successively refined into more and more detail. Each factor can be examined in turn for their influences and

corresponding models produced. By limiting the scope of each exploration (partition) it is easier to verify the component parts, as the fewer the factors the easier it is to identify and isolate the effect. If the components are believed to be sound, then one is in theory only testing the interconnections when the building blocks are joined together. It is necessary to join the components together to reach an understanding of the way in which the system operates as a whole.

If it is accepted as suggested earlier in this chapter, that to produce complex models will take much time and effort, then are there any benefits to be gained from exploring the problem at a simpler level?

It is submitted that the value of this technique lies in the actual process of modelling itself. Part of the modelling process is to critically review the activities of end-user support section. One would discuss the provision of support with the section manager, with the practitioners of providing support and with the users. For example it could lead to the realisation that expertise as represented by a member of the support team is a positive asset, one of the section's strengths. However it could also be regarded as a liability as the section could become reliant on a member that could be lost (2.3.2). Presumably, having identified a vulnerable area, managerial corrective action could be taken.

It is therefore suggested that herein lies the real strength of the technique. It is further suggested that the development of full, detailed models, although desirable as a management planning tool, are almost a by-product of the technique itself.

6.4 Summary of the Models.

a) The 'simple model' and 'alternative support', although simplistic are modelled on actual considerations based on experience at Southampton City Council and supported by literature.

b) 'Support' covers three main factors: flexi-time, staffing and other work done which were areas of concern to Southampton City Council.

c) 'Support' needs expanding to consider additional factors to create a more representative model.

d) The factors affecting the rate at which users ask questions needs exploring. This should then be joined to a fuller model of influences on the provision of support to produce a composite model.

e) This composite model then needs exploring with actual figures from several organisations before an element of confidence can be placed on its predictions.

f) As it stands, the strength of the technique, is that the models may be used by end-user support management as a focus for ideas about improving the provision of support.

References.

[1] P. Fishwick
'The role of process abstraction in simulation'
p 25
IEEE Transactions on Systems, Man and Cybernetics
Vol 18, No 1, Jan/Feb 1988

7. CONCLUSIONS AND FUTURE WORK.

7.1 Conclusions.

As has been mentioned previously, these models have been prepared as a means of starting to clarify understanding of the problems of end-user support. Eventually a model could be created that could be used as a management planning tool for the provision of support. As the work here has been based on a Local Authority, in the first instance a tool could evolve for use in a Local Authority, but with additional research, it could broaden out to being applicable to any organisation. Surely there must be areas of overlap and similarity. There is certainly plenty of scope for additional research, and some suggestions follow.

7.2 to 7.20 Suggestions for Future Research.

7.2 One assumption in the preceeding models is that the questions are homogenous. This is a gross simplification of the real world. For example questions range from: which way does the diskette go in?

to: How do I get a pound sign in LOTUS 1-2-3?

The answer to the diskette question can be answered over the telephone. The answer to the

LOTUS 1-2-3 question is more complex. When it was first asked support went cap-in-hand to an Accountancy LOTUS expert. the second time it was asked, support had an answer to hand: a five minute demonstration with the user, plus an A4 hand-out and a copy of a file on diskette.

For further examples of the types of questions asked see Appendix C.

Further work could be done on discovering criteria on which to judge the complexity of a question, which in turn could be applied to incorporating this within a system dynamic model.

Additionally it would be of interest to know how the level of complexity changes as the users become more expert. From observation only, it would seem that questions become less frequent as the user becomes more expert, but when the questions are asked they take longer to find the answer to as the questions themselves become more complex. It would be interesting to find out whether or not this is the case.

7.3 The appendices B give sample figures of the flow in of questions to support. It would be interesting to find out how representative this type of flow is? One would need to examine figures from several organisations, perhaps first

SOUTHAMPTON UNIVERSITY LIBRARY

to contrast Local Authorities, then later other organisations.

It has been assumed that questions flow in at a steady rate. If the figures in Appendix B.1 are analysed, then it can be seen that 20 questions were asked in September and 57 in July. The figures in the second table have been adjusted to take into account holidays etc., but it would be of value to know how holidays etc. affect the flow of questions in, in order to be able to plan for the return or to know when to provide cover.

7.4 In the models it has been assumed that the questions are answered in strict rotation. This is too simplistic. Suppose a question can be answered simply over the telephone, then this question in effect jumps the queue.

The model needs to be able to handle different priorities assigned to individual questions as they come in.

7.5 In the third model 'Support' only three areas were explored with repect to factors influencing the provision of support. There must be several more. For a more complete picture of the problem and a more useful model these additional factors need identifying, and ways found of modelling them.

7.6 Implicit in the models so far has been the assumption that zero outstanding questions is the ideal state. As suggested by backlog in 'Support' (5.4.2) this may not always be the case. Some research need to be done to find out ways of measuring user satisfaction with the service provided. It is a crude supposition that the length of the queue is a measure of satisfaction or dissatisfaction with the service. The longer the queue the more dissatisfied the users are because the longer they have to wait for an answer. Alternatively the measure is the rate at which users ask questions of end-user support. The more questions asked, the more satisfied the users must be, otherwise they would stop asking questions.

7.7 All the way through this thesis it has been assumed that factors influencing users is outside of the scope of the study. As the research has been centered on the provision of end-user support, then this is probably reasonable. However, for a fuller picture, and for model that could be even remotely useful as a planning tool, then this area needs exploring.

7.8 An area that has been touched on, (4.4) but bears further research is the idea that it takes some

time for new team members to become fully operational. It must surely be quite reasonable to expect to add members to the team if the end-user support function is growing, but one also needs to know the avearge lead time before they can become fully effective. Again this would be needed to be included on a planning tool.

7.9 Another subject, allied to 7.8, is the question of when to introduce a new member of staff. It has been suggested that this can be a matter of finance (Appendix H and section 4.4), but there must be other rules of thumb to determine when the time is ripe. These need finding out.

7.10 It has also been suggested that doing work other than simply answering questions would be of value (4.4), but would it? Is there a point at which it would become counter-productive? What sort of ratio is recommended?

7.11 One set of figures that was not kept with any detail at southampton City Council were the figures associated with the provision of answers. To enable a more accurate model figures relating to the time taken between placing the question and receiving a reply could be analysed along with the actual time spent working on an answer. The first could be associated with a measure of

satisfaction with support (7.6) and the second part of determining the complexity of a question (7.2).

7.12 This suggestion may lead to the writing of software as well as inclusion in a model. Once a question is answered, then the question, along with the answer should be stored in a way that facilitates easy retrieval of the answer. Once a question is answered, it is usually quicker to find the answer a second time. It is even quicker if one can remember the answer. It should also enable the retrieval of similar questions. As a corollary to this it would be of value to support to be able to spot emerging trends. Trends that could suggest more training is needed, say, or that there is a recurring fault on a piece of software or haardware.

Over and above the ideas so far raised in this thesis, there are many more areas, connected with end-user support, that could perhaps bear further exploration.

7.13 There is a general need for the provision of education and training of users. How do organisations meet this need? What are their resources? How effective are these resources?

7.14 How much help is given to users for writing their own applications? What are the inherent problems? What guidelines can be given, are given, should be given, in terms of standardisation, generalisation and documentation.

7.15 What is the role of liaison officers? (2.2) Is there a career path? What is the career path for those embarking on end-user support?

7.16 Several organsations have set up Information Shops, including Hampshire County Coucil. These are areas set aside for users to come in and see what is available in terms of hardware and software, and also to obtain advice. They are costly in terms of staffing, as someone should be at hand to answer questions, as well as space and equipment. One copy of every device and package should be there, plus additional equipment to act as a resource centre.

7.17 Ways of modelling the effect that policy decisions have on the running of end-user support could be explored, together with a study of the types of policy in operation.

7.18 Areas that could be descibed as behavioural could be investigated and ways of modelling found. The area of study here could be the effect of motivation on end-user support staff.

7.19 Alternative ways of modelling such as making use of statistical techniques, catastrophe theory, behavioural science could be explored for their applicability. Results from these other techniques could be compared with the results from System Dynamics.

7.20 Finally another area for further research could be the provision of support by other sources such as dealers.

7.21 Summary of Future Work.

It is submitted that this research has potential for future expansion in terms of:

a) Understanding of the problem

b) Provision of a management planning support tool for use especially within a Local Authority environment.

c) The manager can then perform 'what-if' analysis on the problem.

7.22 The Next Stage.

Suppose that resources for producing a model to solve an actual problem for a company or Local Authority become available. What data should be collected? How can this model be tested, to see if the solutions suggested actually help?

a) Collection of data.

The data collection must be as simple as possible, as it should not add unduly to the workload of the support staff. Wherever possible use a computer to record data and generate items such as data and time.

The data collected should include:

i) The date and time that the question is received.
ii) A brief description of the question.
iii) Who recieves the question.
iv) Who answers it.
v) The answer.
vi) Date and time of the answer
vii) The time taken to actually work out the answer.

additional data that should be recorded:

viii) The hours worked each week

ix) The other work done, together with a brief description.

x) The staff associated with support.

and this data should be collected over an appropriate timescale, say three months.

b) Run the model.

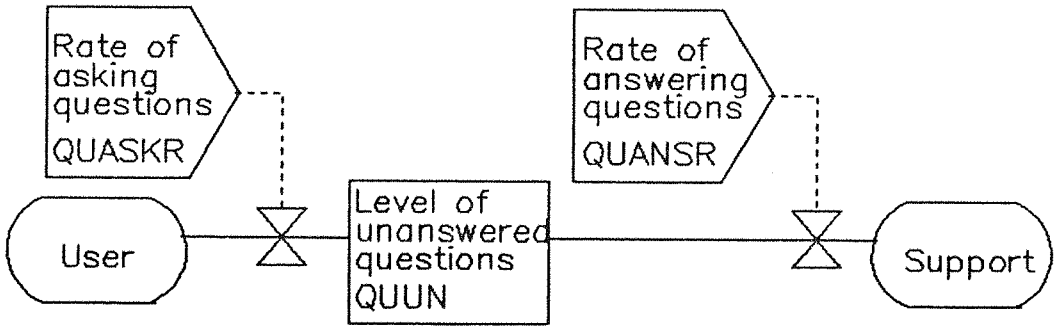
The data suggested above would support an enhanced model 3. Run the model using the collected data, and look for possible solutions. Perhaps a busy period is noted at the beginning of the month, so someone is drafted in to work for just that period.

c) Change the system, and continue to collect the data for a further period, say three months.

d) Has there been an improvement? Could anything else be done?

The model should be validated at the end of the first stage against what is actually happening. The model should reflect changes in the system as they occur. If there is a discrepancy, then the model should come under careful scrutiny, but it is also possible that the imagined problem is not what is actually happening. By putting the system under analysis a greater understanding of the problem emerges.

APPENDIX A.1
Experiment by Paper, Pencil and Rubber.



DESCRIPTION.

1. There are no outstanding questions to begin with.
2. Assume questions are being asked at a steady 10 per week unless it is week 3, when 30 questions are asked.
3. Assume questions are answered at a steady 10 per week.

TABLE

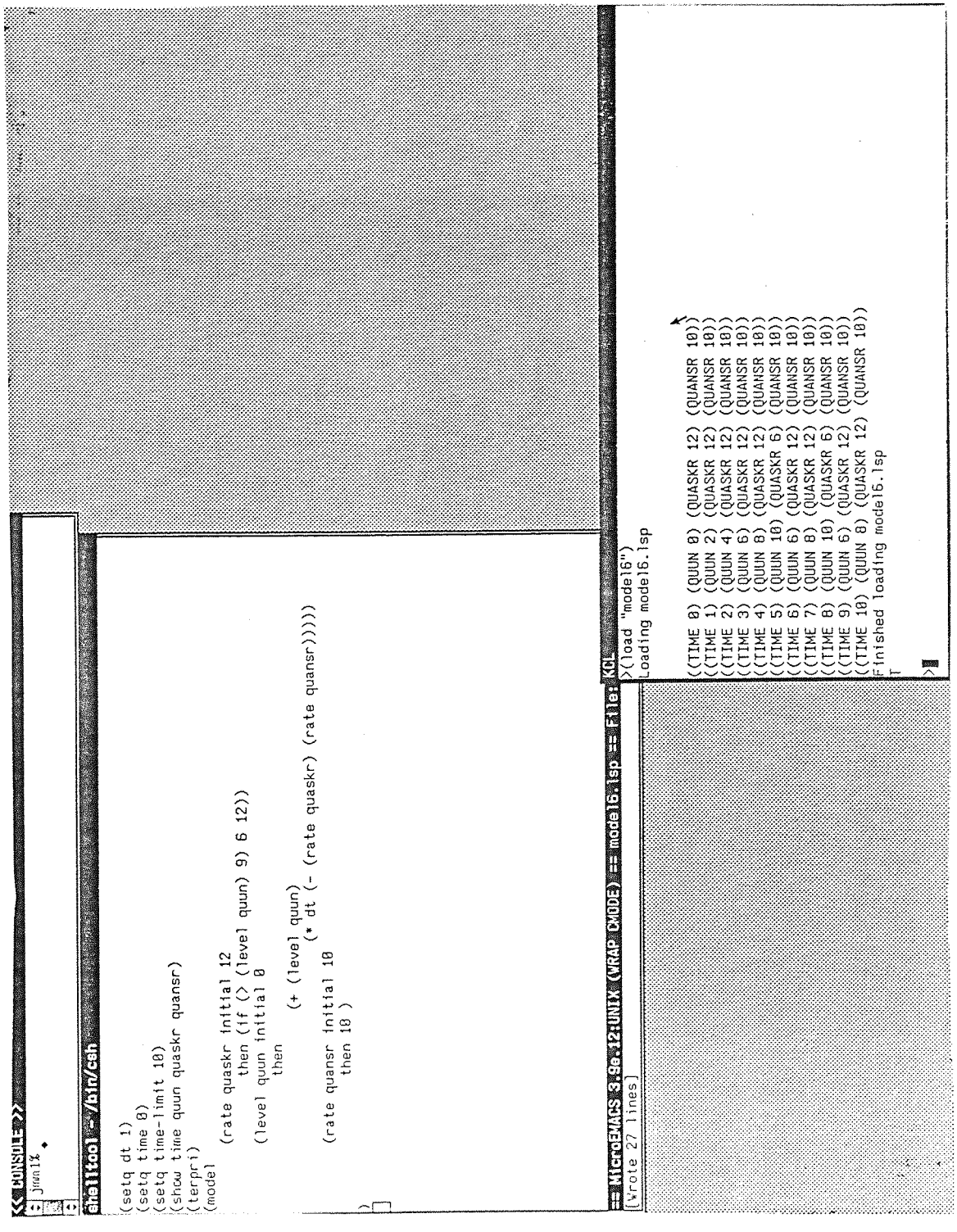
TIME	QUUN	QUASKR	QUANSR
0	0	10	10
1	0	10	10
2	0	10	10
3	0	30	10
4	20	10	10
5	20	10	10

EQUATIONS

```
QUUN.J = 0
QUUN.K = QUUN.J + DT * (QUASKR.JK - QUANSR.JK)
QUASKR.J = 10
QUASKR.K = 10 IF TIME <> 3
         = 30 IF TIME =  3
QUANSR.J = 10
QUANSR.K = 10
```

APPENDIX A.2.

Experiment by LISP.



Appendix A3. Experiment by LOTUS.

```
RESW.J =          1
RESW.K =          0 IF QUUANA >          9 ELSE RESW
                  1 IF QUUANB >          9 ELSE RESW

QUASKAR.JK =             12
QUASKAR.KL =              6 IF RESW.J =          0
                         12 IF RESW.J =          1

QUANSAR.JK =             10
QUANSAR.KL =             10

QUUNA.J =         0
QUUNA.K = QUUNA.J + DT * (QUASKAR.JK - QUANSAR.JK)

QUASKBR.JK =              0
QUASKBR.KL =              6 IF RESW.J =          0
                          0 IF RESW.J =          1

QUANSBR.JK =              4
QUANSBR.KL =              4

QUUNB.J =         0
QUUNB.K = QUUNB.J + DT * (QUASKBR.JK - QUANSBR.JK)
```

TIME	QUUNA	QUASKAR	QUANSAR	QUUNB	QUASKBR	QUANSBR	RESW
0	0	12	10	0	0	4	1
1	2	12	10	0	0	4	1
2	4	12	10	0	0	4	1
3	6	12	10	0	0	4	1
4	8	12	10	0	0	4	1
5	10	12	10	0	0	4	0
6	12	6	10	0	6	4	0
7	8	6	10	2	6	4	0
8	4	6	10	4	6	4	0
9	0	6	10	6	6	4	0
10	0	6	10	8	6	4	0
11	0	6	10	10	6	4	1
12	0	12	10	12	0	4	1
13	2	12	10	8	0	4	1
14	4	12	10	4	0	4	1
15	6	12	10	0	0	4	1
16	8	12	10	0	0	4	1
17	10	12	10	0	0	4	0
18	12	6	10	0	6	4	0
19	8	6	10	2	6	4	0
20	4	6	10	4	6	4	0
21	0	6	10	6	6	4	0
22	0	6	10	8	6	4	0
23	0	6	10	10	6	4	1
24	0	12	10	12	0	4	1
25	2	12	10	8	0	4	1
26	4	12	10	4	0	4	1
27	6	12	10	0	0	4	1
28	8	12	10	0	0	4	1
29	10	12	10	0	0	4	0
30	12	6	10	0	6	4	0
31	8	6	10	2	6	4	0
32	4	6	10	4	6	4	0
33	0	6	10	6	6	4	0
34	0	6	10	8	6	4	0
35	0	6	10	10	6	4	1
36	0	12	10	12	0	4	1
37	2	12	10	8	0	4	1
38	4	12	10	4	0	4	1
39	6	12	10	0	0	4	1
40	8	12	10	0	0	4	1

APPENDIX B.1

SOUTHAMPTON CITY COUNCIL PROBLEM LOG 1985
SUMMARY OF QUESTIONS ASKED

RECIPIENT	A	B	C	TOTAL
JANUARY	27	6	3	36
FEBRUARY	20	2	2	24
MARCH	27	3	1	31
APRIL	34	1	2	37
MAY	40	4	2	46
JUNE	22			22
JULY	56		1	57
AUGUST	30		1	31
SEPTEMBER	19	1		20
OCTOBER	38			38
NOVEMBER	39			39
DECEMBER	28			28
	380	17	12	409
AVG/MONTH	31.67	1.42	1.00	34.08
AVG/WEEK	7.31	.33	.23	7.87

NOTES
*****

1. Recipient A was working full-time on user-support.
   Recipients B and C helped out from time to time.
2. These figures could be adjusted, to allow for sick leave and holidays etc.

RECIPIENT	TOTAL	HOLIDAY/ SICK	ADJUSTED TOTAL
JANUARY	36		36
FEBRUARY	24		24
MARCH	31		31
APRIL	37		37
MAY	46		46
JUNE	22	7 days (3/8)	35
JULY	57		57
AUGUST	31		31
SEPTEMBER	20	10 days	40
OCTOBER	38		38
NOVEMBER *	39	10 days	78
DECEMBER	28	5 days	37
	409		490
AVG/MONTH	34.08		40.83
AVG/WEEK	7.87		9.42

* 39 was probably accurate as an influx of questions could be expected following support's sick leave

APPENDIX B.2

SOUTHAMPTON CITY COUNCIL PROBLEM LOG 1986
SUMMARY OF QUESTIONS ASKED

	RECIPIENT	A	J	K	OTHERS	TOTAL
1	JANUARY	55	1		2	58
2	FEBRUARY	51	4		2	57
3	MARCH	81	9		1	91
4	APRIL	80	12		1	93
5	MAY	42	5		1	48
6	JUNE	49	11			60
7	JULY	22	30		1	53
8	AUGUST		29		2	31
9	SEPTEMBER		37		2	39
10	OCTOBER		40	8	1	49
11	NOVEMBER		20	19		39
12	DECEMBER		8	9	1	18
		380	206	36	14	636

AVG/MONTH 53.00
AVG/WEEK 12.23

APPENDIX B.3

SOUTHAMPTON CITY COUNCIL PROBLEM LOG 1987
SUMMARY OF QUESTIONS ASKED

	RECIPIENT	I	J	K	TRAINEES	OTHERS	TOTAL
1	JANUARY	3	5	9			17
2	FEBRUARY	8	1	5	3	1	18
3	MARCH	5	4	9	2	1	21
4	APRIL	9	8	5	8	2	32
5	MAY	1	3	1			5
6	JUNE	1	12	3	1	1	18
7	JULY	6	9	3	5		23
8	AUGUST	3	4	1			8
9	SEPTEMBER	5	9	2	5	1	22
10	OCTOBER	8	9	12	8		37
11	NOVEMBER	7	2	1			10
12	DECEMBER	2	3	2	1		8
		58	69	53	33	6	219

AVG/MONTH 18.25
AVG/WEEK 4.21

APPENDIX C.

Sample Questions from Southampton City Council Problem Log July 1985.

1 When will the equipment ordered arrive?
2 Wanted information on the Lotus and Symphony upgrades
3 Information given as to when a machine could be purchased by another department
4 Handouts requested on dBASE III
5 Help requested in installing a new package.
6 The new logo has caused the loss of the buzzer.
7 Unable to access VM. Still using old IPF menus from last Autumn.
8 Can Support invert files to separate VM's?
9 Library still full after workspace deleted.
10 Where can you obtain diskettes?
11 Wanted DisplayWrite 2 training
12 Wanted Support to look at work done in dBASE III
13 Has sidekick arrived?
14 Any chance of recovering lost data?
15 An ADI function was not working.
16 Informed that a user was looking at a package, Norton Utilities, the following Monday.
17 Lent out the plotter
18 Would he run out of memory using LOTUS?
19 When will the machine be mended?
20 Asked if Sidekick had arrived yet.
21 Wanted to book the plotter

22 Additional fields required on Inverted files
23 Could he book our machine?
24 How does one swap between CPM and DOS?
25 How do you print out more than 80 characters in LOTUS
26 Problems with the negative sign in APL
27 Is it possible to transfer an APL variable to LOTUS?
28 Is the new Inverted file ready?
29 Why is the printer not printing?
30 Problem transferring data
31 How to print the result of an enquiry to a line printer?
33 Unable to set up an ADRS file
34 Could not get the correct logon menu
35 PF keys playing up
36 When is dBASE III available?
37 Will there be enough memory for the LOTUS task?
38 User is stuck in shift/APL mode
39 When will the machine and software arrive?
40 Is there a light pen facility on a PC?
41 Is dBASE III corrupting LOTUS
42 When could Support write a system for them?
43 She will be doing some more questionnaires
44 Could Support organise some outside training?
45 Is there a waterproof coat for the PC?
46 Is dBASE III corrupting files?
47 A request for a VM
48 Why isn't an Inverted file available?
49 How to print to printer from ADRS
50 Borrowed book.

APPENDIX D
SAMPLE PROBLEM LOG FOR JULY 1985, SOUTHAMPTON CITY COUNCIL

NO	DATE	DEPT	CONT	H/W	S/W	REC.	PROBLEM & ACTION
224	15/07/85	RA	AR	3179	ADI	A	ADDITIONAL FIELDS REQUIRED ON INVERTED FILES AMENDED TABLE AND PASSED TO CURRENT SYSTEMS
225	18/07/85	CTR	U	XT	LOTUS	A	COULD THEY USE OUR XT? YES
226	18/07/85	CTR	U	XT		A	HOW TO SWAP BETWEEN CPM AND DOS TOLD
227	18/07/85	LEI	AW	M24	LOTUS	A	HOW TO PRINT MORE THAN 80 CHARACTERS TOLD
228	18/07/85	LEI	AW	M24	APL	A	PROBLEMS WITH NEGATIVE SIGN NO APL CHIP
229	18/07/85	ENG	Z	3179	ADRS	A	IS IT POSSIBLE TO TRANSFER AN APL VARIABLE TO LOTUS? YES, VIA ADRS
230	18/07/85	RA	AR	PC	ADI	A	IS NEW FILE READY? YES, DOCUMENTATION ETC. HANDED OVER
231	19/07/85	ENG	Z	3179	ADRS	A	no question recorded GIVEN ADRS TRANSFER FUNCTION TO TRANSFER TO LOTUS

NB. The departments are an abbreviation, but the contact has been arbitarily assigned letter in order to protect privacy.

CONT is the support person contacted.
H/W = Hardware
S/W = Software
REC. is the recipient of the question.

APPENDIX E.

EXTRACT FROM END-USER SUPPORT'S DIARY JULY 1985

Monday 15th July 1985	8.45-11.45	DWD
	11.45-12.15	Administration
	14.00-15.00	Museums
	15.00-15.30	Housing-valuers
	15.30-17.00	Administration
Tuesday 16th July 1985	9.15-15.45	'PROFS', at IBM Basingstoke
	15.45-16.15	Administration
	16.15-18.15	City Transport
Wednesday 17th July 1985		
	8.45-10.45	DWD
	10.45-12.15	Administration
	14.00-16.00	Leisure
	16.00-18.00	Administration
Thursday 18th July 1985	8.00-10.30	Administration
	10.30-12.00	'One per Desk'
	12.00-12.30	Administration
	13.30-14.00	Norton Utlity
	14.00-15.00	Rates
	15.00-16.00	Engineers
	16.00-16.30	Administration
	16.30-17.00	Housing-valuers

Friday 19th July 1985	8.00- 8.30	Administration
	8.30- 9.00	Preparation for demonstration
	9.00- 9.30	Engineers
	9.30-10.00	External demonstration
	10.00-11.30	Meeting with IBM
	11.30-12.30	Administration-schedules
	13.30-14.00	Port Health
	14.00-17.30	Administration

NB. Although some of the administration was administration, it was also the way of recording all those tasks that only take a few minutes, but because they only took a few minutes were difficult to accurately attribute to a particular department, and were therefore easier to class as overheads

Where a department is recorded, then this implies an activity that took over half-an-hour. For example it may have involved developing a system.

APPENDIX F

CONSULTATIONS

This is an outline of a New Technology agreement, of the kind that could have been around during 1985.

In principle a meeting is held a fortnight before the introduction of new technology. 'New technology' generally meant computer equipment, both hardware and/or software. The meeting is attended by management, the people who will be using the equipment, union representatives and someone from the computer section.

The reason for this meeting is for users of new technology to find out what will happen and to raise any questions or objections. A typical objection could be the siting of a printer, from both the noise and privacy angles. Following this meeting there is a two week 'cooling-off' period, during which time answers can be found to problems etc.

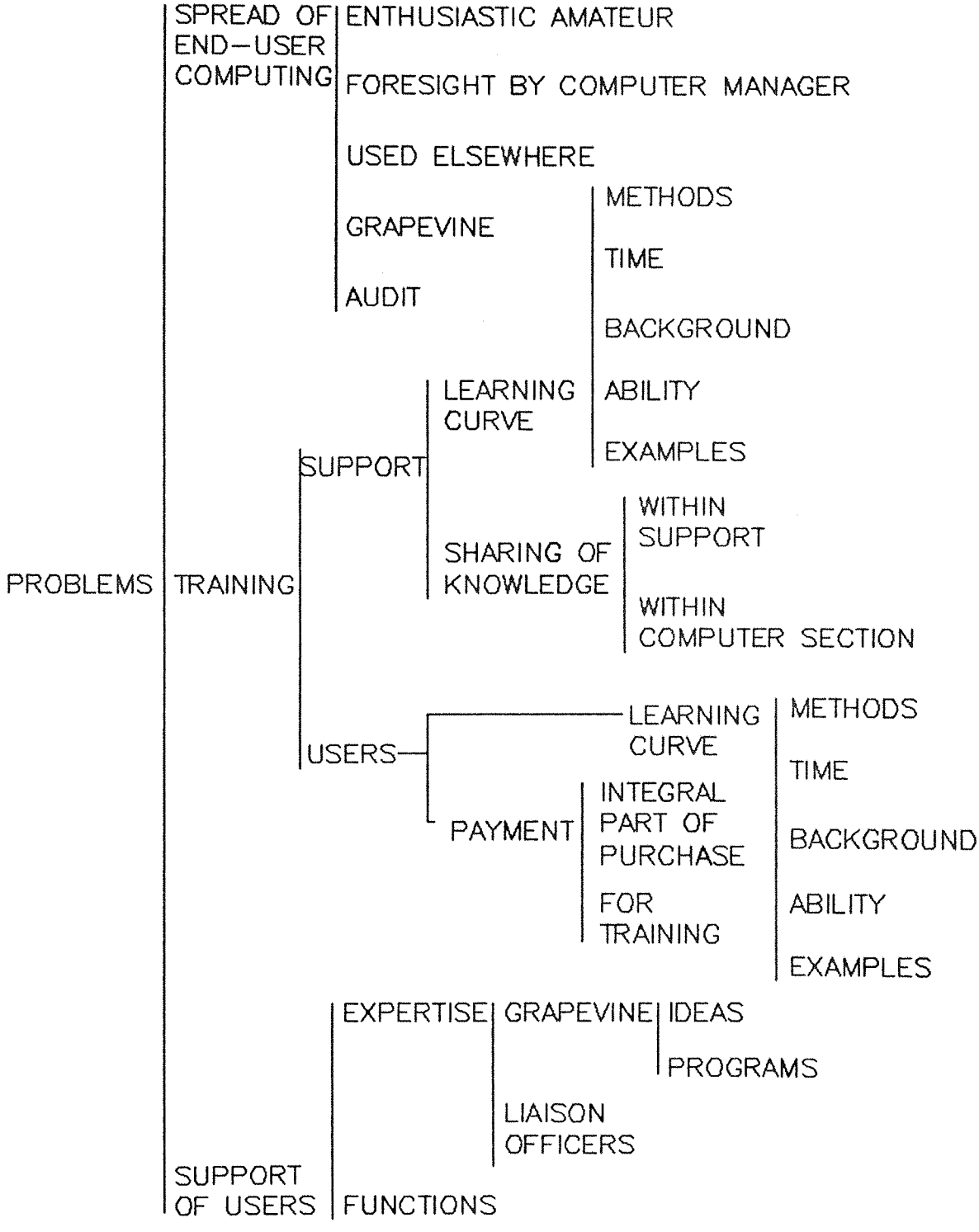
If a section already used a microcomputer, but wanted to use an additional package, then the consultation procedure would be followed but only in respect of the package.

If a new package or micro-computer was introduced to the organisation, then general approval for its use

would be obtained, prior to individual consultations for the sections.

The onus on arranging the consultations lay with the users.

APPENDIX G
TAXONOMY OF PROBLEMS SOLVED



APPENDIX H

A CHARGING POLICY AND HOW IT CAN RELATE TO THE ACQUISITION OF STAFF

Let us assume that a charging policy is in operation as follows. The alternatives are:

Option 1. £25 per hour.
Option 2. £250 per annum per machine.
Option 3. £100 per annum plus £15 per hour.

The supposition was that experts would opt for the hourly rate, in the confidence that they would not need to ask for help many times in a year. Novices would go for a fixed fee, as end-user support could be called on many times without incurring additional costs. Everyone else would go for the third option.

In practice nearly everyone went for the fixed fee option.

How many users could be supported on the fixed fee option? The main constraint is time. Let us assume that support staff are scheduled for 30 hours a week for 45 weeks of the year, to allow for sickness and

holidays etc.

```
    30 hours/week
x   45 weeks/year
  ----
   150
  1200
  ----
  1350 hours/year
  ====
```

Based on the figures in the options above, if the hourly rate is £25 and the fixed fee is £250, it is assumed that users have 10 hours of support per year, therefore, if there are 1350 hours available, 135 users can be supported by each member of the end-user support team.

1350 hours/year divided by 10 hours per user

=

135 users per support person.

In terms of money: 135 users bring in £33,750 (135x£250). In 1988, when these notes were recorded, a programmer would be earning around £15,000. This leaves £18,750 for overheads and profit. Therefore if 10 hours of support time per machine is a reasonable level of support, then £250 is a reasonable charge, and can be used as a means of employing additional staff, as the income for the year is known in advance.

BIBLIOGRAPHY

Books.

Backhouse J.K., Houldsworth S.P.T.
'Pure Mathematics, A First Course'
Longman 1978

Brooks F.
'The Mythical Man Month'
Addison-Wesley 1972

Burgess R., St.John Bate J.
'Office Automation using the IBM Personal Computer Systems'
Collins 1986

Carr H.H.
'Managing End-User Computing'
Prentice-Hall

Christie B. (editor)
'Human Factors of Information Technology in the Office'
Wiley

Coyle R.G.
'Management System Dynamics'

John Wiley

Forrester J.W.
'Industrial Dynamics'
MIT Press

Heigl L.
'The Electronic Office and You'
Sphere Reference 1985

Hirschheim R.A.
'Office Automation, Concepts Technologies and Issues'
International Computer Science Series. 1985

Jarke M.
Managers, Micros & Mainframes
New York University

Lang K., Lang T.
'Managing Your Business with the IBM PC'
Holt, Rinehart & Winston 1985

Martin J.
'Applications Development without Programmers'
Prentice-Hall

Megarry J.
'Computers Mean Business'
Pan. 1984

Megarry J.

'Inside Information, Computers, Communication & People'
BBC 1985

'Operations Management - A Systems Concept
Chapter 13. System Dynamics

Richmond B., Peterson S., Vescuso P.
'An Academic Users Guide to Stella'
High Performance Systems

Roberts N., Anderson D., Deal R., Garet M., Shaffer W.
'Introduction to Computer Simulation. A Systems Dynamics Modelling Approach'
Addison-Wesley

Willcocks L., Mason D.
'Computerising Work. People, Systems design and Workplace Relations'
Paradigm

Zorkoczy P.I. (ed)
'Oxford Surveys in Information Technology, Vol 2'
Oxford 1985

Papers.

Abdel-Hamid T.K. 'The Dynamics of Software Project Staffig: A System Dynamics Based Simulation

Approach'
IEEE Transactions on Software Engineering. Vol 15. No 2. February 1989

Baroudi J.J., Orlikowski W.J.
'A Short-Form Measure of User Information Satisfaction: A Psychometric Evaluation and Notes on Use'
Journal of Management Information Systems/ Spring 1988 Vol 4 No. 4

Brehmer B.
'Systems Design and the Psychology of Complex Systems'

Bylander T.
'A Critique of Qualitative Simulation from a Consolidation Viewpoint'
IEEE Transactions on Man, Systems and cybernetics, Vol 18, No. 2, March/April 1988

Cobelli C., Lepschy A., Romanin Jacur G., Viaro U.
'On the Relationship Between Forrester's Schematics and Compartmental Graphs.'
IEEE Transactions on Systems, Man and Cybernetics, Vol SMC-16, No. 5, September/October 1986.

Coyle R.G.
'Assessing the Controllability of a Productions and Raw Materials System'
IEEE Transactions on Systems, Man and Cybernetics, Vol SMC-12, No. 6, November/December 1982

Coyle R.G.
'A System Description of Counter Insurgency Warfare'
Policy Sciences 18 (1985) 55-78

Coyle R.G.
'A Systems Approach to the Management of a Hospital for Short-term Patients'
Socio-Econ. Plan. Sci. Vol 18, No 4, pp 219-226, 1984

Coyle R.G.
'The Technical Elements of the Systems Dynamics Approach'
European Journal of Operation Research 14 (1983) 359-370

Coyle R.G
'The Use of Optimisation Methods for Policy Design in a Systems Dynamics Model'
Systems Dynamics Review 1 (No 1, Summer 1985) 81-91

Fishwick P.A.
'The Role of Process Abstraction in Simulation'
IEEE Transactions on Man, Systems and Cybernetics, Vol 18, No 1, January/February 1988

Forman F.L., Inmon W.H.
'Using extracted Samples Improves Information Centre'
Data Management Vol 26. Part 1, pp 23-27,1988

Hirschheim R., Newman M.

'Information Systems and User Resistance:Theory and Practice'
The Computer Journal, Vol 31, No. 5 1988 pp398-408

Homer J.B.
'Worker Burnout: a Dynamic Model with Implications for Prevention and Control'
System Dynamics Review Vol.1 No. 1 Summer 1985

Hopper G.W.
'Decision Aiding Dangers: The Law of Hammer and Other Maxims'
IEEE Transactions on Systems, Man and Cybernetics, Vol SMC-16, No. 6, November/December 1984

Jagodzinski A.P., Clarke D.D.
'A Review of Methods for Measuring and Describing Users' Attitudes as an Essential Constituent of Systems Analysis and Design'
The Computer Journal, Vol 29, No 2, 1986, pp 97-102

James P.N.
'The User Interfact:The Support Infrastructure'
Jounal Information Systems Management. Vol.4,Part 4, pp66-67. Fall 1987

Keller L.R., Ho J.L.
'Decision Problem Structuring : Generating Options'
IEEE Transactions on Systems, Man and Cybernetics, Vol 18, No.5, September/October 1988

Krishnamurthy R., Hochgesang G.T.
Architecture for a Universal Office System
CH2022-2/84/0000/0554 IEEE 1984

Licker P.S.
'Human Resource Management and Business Success in Small, Hi-Tech Firms: Some Lessons for Data Processing'
The Computer Journal, Vol 32, No. 6 1989. pp 533-540

Lucas H.C.
'Educating Managers to Cope with Computers'
CH2022-2/84 0406 IEEE 1984

Magal S.H., Carr H.H.
'An Investigation of the Effects of Age, Size and Hardware Option on the Critical Success Factors Applicable to Information Centres.'
Journal of Management Information Systems/ Spring 1988 Vol 4 No. 4

Rege A., Agogino A.M.
'Topological Framework for representing and Solving Probabilistic Inference Problems in Expert Systems.
IEEE Transactions on Systems, Man and Cybernetics, Vol 18, No. 3, May/June 1988

Rouse W.B., Morris N.M.
'Understanding and Enhancing User Acceptance of Computer Technology'

IEEE Transactions on Systems, Man and Cybernetics, Vol SMC-16, No. 16, November/December 1986

Sandstrom G.
Users' Opinion of their Work, Language and Computerised System.'

Spence J.W.
End-User Computing - the Human Interface
Journal of Systems Management, February 1988.

Sumner M., Klepper R.
Information Systems Strategy and End-User Application Development'
Data Base. Summer 1987

Tagg R.
'Software Engineering from the End-User Angle'
Computer Bulletin. December 1987

Winkler S.
'The Information Resource Centre. A Strategy for End-User Support.'
CH2022-2/84/0000/0504 IEEE 1984

Zinser K., Henneman
'Development and Evaluation of a Model of Human Performance in a Large-Scale System
IEEE Transcations on Man, Systems and Cybernetics, Vol 18, No. 3, May/June 1988

Zloofe M.M.
'Office Software : Capabilities and Drawbacks'
CH2022-2/84/0000/0540 IEEEE