HUMAN INTERACTION WITH INFORMATION: NEW DISPLAY AND INTERACTION METHODS, AND THEIR IMPLICATION FOR COMMUNICATION NETWORK DESIGN AND PERFORMANCE

by

Ioannis V. Tzavaras

Thesis submitted for the degree of
Doctor of Philosophy
in the Faculty of Engineering,
University of London

July 1984

Department of Electrical Engineering
Imperial College of Science and Technology

ABSTRACT

Though concerned with a single theme - human interaction with information - this thesis divides naturally into two parts, each concerned with a well-defined topic.

The first part deals with the human factors of information handling. First, a review of currently available techniques for accessing information, and a study of human memory processes, enables those methods of display and interaction leading to ease of use to be established. An outcome of this review is the proposal of an approach that exploits instinctive information handling skills.

This 'Bifocal' technique for information presentation and interaction is described, and comparatively evaluated with other known methods of database access. Design steps for its implementation, and an operating prototype model, are presented.

The second part of this thesis examines the impact that such new database interaction techniques will have on supporting communications network design philosophies, and on the enhancement of network performance.

An examination of the way a local area database would operate in such a network reveals a degree of topology independence. This is used as a basis for deriving a simplified model describing the operation of file servers in such a system. It is proposed that dynamic job routing is an effective, and easily implemented performance enhancement strategy.

A queuing network model is constructed and found to be similar to models of dynamic routing nodes in Packet Switched Networks. A

new approximate analysis technique for this model is proposed and developed. A survey and performance analysis, shows that it is unique in providing an efficient analysis approach for a wide range of model types, and that it is usable as part of a network performance optimization scheme, based on file allocation/ replication.

ACKNOWLEDGEMENTS

I wish to express my gratitude to my supervisor Dr. R. Spence for his help and guidance throughout the duration of this work, even when I had drifted into research areas not of his direct interest.

In addition, I would like to thank Dr. M. Apperley who together with Dr. R. Spence, proposed the basis for the first part of this research.

Thanks are due to all my friends, and the members of the Information Engineering section, who have supported me during this period.

This thesis is dedicated to my family with gratitude, without whose moral and financial support this work would not have been possible.

TO MY FAMILY

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

β - mean of a diffusion process

β_{ij} - transfer rate from queue i to queue j

γ - variance of a diffusion process

Ci - storage capacity of server i

f - index

F - number of files

{Fi} - set of files stored in server i

i - index

j - index

k - index

λ - overall arrival rate

λ^i_k - arrival rate when server i is in state k

l_f - length of file f

Li - mean queue length at server i

μ_i - service rate of server i

M - number of servers

n_i - queue size at server i

N - maximum queue size

p^i_k - equilibrium state probability of server k being in state i

q_f - probability of file f being requested

Qij - query rate of file j by user i

Qi - auxiliary variable in Appendix 3

ρ - utilization

r - index

Ri - auxiliary variable in Appendix 3

Si - maximum size of queue i

S_i' - auxiliary variable in Appendix 3

$t_{j,i}$ - a threshold state

Ti - auxiliary variable in Appendix 3

Uj - update rate of file j

Xij - 0-1 allocation variable

STATEMENT OF ORIGINALITY

The original contributions of this thesis are deemed to be the following:

1) The identification of significant, instinctive human factors in the context of human interaction with information. Presentation of the Bifocal concept. A relative assessment with other similar methods, based on the above factors.

2) Design guidelines and implementation of a prototype graphics processor supporting a Bifocal display.

3) The modeling of a local area information network and the proposal of a distributed data management technique based on the assumption of local area topology independence. Presentation of the file allocation/replication problem based on a dynamic routing, central server model. Similarity of model to models of adaptive routing Packet Switched Network nodes.

4) Presentation and evaluation of a new, approximate analysis technique for dynamic routing networks. The proposal and investigation of an efficient iterative technique for solving the system of equations formed. Statistical analysis of performance. Applications to the optimal file allocation replication problem, and extensions.

THESIS ORGANIZATION

In Chapter 1, psychological factors important in the human perception of information are presented. Human memory processes and the concept of spatiality, in the context of information cognition and the use of these factors in database systems are investigated. Currently available database access methods are presented and comparative evaluation results quoted.

In Chapter 2, recent attempts at exploiting spatiality and other instinctive human factors in database systems are presented. The concept of the Bifocal display is presented and justified. Design guidelines and a prototype model are described. Database access systems based on spatiality are comparatively evaluated, based on the factors suggested in Chapter 1.

In Chapter 3, attention is shifted onto a typical example of a local area, integrated, information network that is expected to support intelligent workstations employing spatiality, and other high data content media. An investigation of the data management techniques for such a distributed network indicates that a virtual, centralised controller is an efficient approach. Presentation and justification of a macro and micro data allocation problem. The dynamic routing, queuing model of the central controller.

In Chapter 4, a backround of queuing analysis and results into the benefits of adaptive routing are presented. A critical review and assessment of currently available, exact and approximate analysis techniques for dynamic routing, central server systems.

In Chapter 5, a new approximation technique is proposed and applied in several networks of different characteristics. Statistical analysis of performance. Investigation of convergence. Applications to the problem of optimal file allocation/replication are examined. Extensions to the technique are proposed.

In the conclusions, the main results of this thesis are reviewed, and suggestions for further research presented.

CHAPTER 1

HUMAN FACTORS IN THE DISPLAY AND INTERACTION WITH INFORMATION

1.1 Introduction

As questions relating to the technical performance of automated information (database) systems, and of their economic viability are being resolved, increasing attention is being paid to human factors. There is a widespread recognition that future systems will be commercially viable only if the user interface is in harmony with user skills and task requirements [SHNE78].

This growth in the development and use of automated information sytems has led naturally to a number of attempts at producing user friendly man-machine interfaces, i.e. the interface between a human user and the data management processes within a computer. In past years when the main users of computers were skilled professionals, this man-machine interface operated at a low level adequate for precisely defined tasks needing a minimum of assistance and subsequent analysis by the processing machine. The advances however of computer technology, and the subsequent introduction of computers to familiar environments such as the home and the office, have created a new class of potential users. This class of users will work with computers by choice, using them as tools rather than as the hub of their task.

This proliferation of new users has led to the large amount of interest spent on the re-evaluation of the man-machine interface. Man-machine issues of a generalised sense abound, such as workstation design, and operator fatigue. Here, we will attempt to provide a framework for the issues and factors important in the development of interfaces designed for natural and therefore efficient interaction with information.

The type of usage expected of information systems is diverse. The tasks which are performed may range from the retrieval of for example, an inter-office digitized verbal message, to complex database queries requiring precise answers. The functions or operations which users perform on data may be approximately classified under three main headings:

a) the retrieval of information
b) insertion/deletion of items from the database
c) structuring the presentation of information

The large number of conflicting requirements and differences between task types has led to a large number of task specific and generalized database query techniques. Conventional query techniques operating mainly on alphanumeric data and the relational data model are briefly described in sections 1.2.2 and 1.4.1. The psychological backround of human interaction with externally presented information is developed in section 1.3 and shown to advocate the use of instinctive information handling skills. A number of systems which exploit such human skills have been recently proposed. These are presented and comparatively evaluated in chapter 2, while an evaluation of 'conventional' techniques is presented in section 1.4.

Two user types will be distinguished in these discussions. One is the trained computer professional, skilled in formulating precise, syntactically correct queries in a conventional database query method. The other, and the one on which these discussions concentrate, is the intermittent user, representing the most rapidly expanding class of users. He may be a trained professional, and will use computers as a tool in aiding the completion of a task. He will probably have minimal typing and syntactic query formulation skills, and it is possible that he will have no desire to learn either.

It is the accommodation of such users in the existing database/information interaction environment that presents a new set of problems in the design of human compatible interfaces.

1.2 Types of Interaction

It is the diversity in the types of tasks and groups of people performing these tasks which is the main factor behind the existence of a large number of user specific and 'generalized' query languages. A question exists as to whether one or a limited set of systems can satisfy all user requirements. A user group sharing a common task is likely to have a set of specific job requirements. It seems reasonable therefore that a database query system should be organized in a manner compatible to the user sets and the task at hand. This requires the use of a number of job specific systems rather than few generalized ones.

The range of interaction modes covered by current and proposed systems is large, and a brief classification is attempted here:

a) menu driven systems

b) precise query formulation

c) natural language or non-procedural systems

d) graphics assisted systems

The main features of the above will be described in the following four sections.

1.2.1 Menu driven systems

This mode of interaction is, in general, computer guided and usually offers a limited set of options. It is simple to interact with, and has been used successfully in the interface between new users and databases, to produce simple tree-type searches. The range of options and the extent to which a search can be made is usually limited. Succesful systems have however been implemented [ROBE79], and a number of studies available on improving search performance (see section 1.4.1.2). A similar set of techniques -to the extent that they are also computer assisted- are the 'fill-in-the-blank' approaches. Users respond by providing a word, number or phrase to a line of text. Some training is required as users are required to be aware of response formats. Both of these techniques, even though limited in their range of applications, have been found acceptable, new users quickly becoming proficient in their use, and are likely to remain with us as long as their restricted application domain exists.

1.2.2 Precise query techniques

These techniques are mainly used in the formulation of exact queries in response to precise questions such as, "what is the number of employees of a company? what is their average salary?", etc. They are very powerful if used correctly and indeed are the only approaches which will interface to many of the databases currently available. Using a categorization taken from programming languages, these query techniques operate at a low level, requiring the minimum of assistance and subsequent query analysis from the supporting processor. As a consequence, however, they are difficult to learn, a typical value being 5 hours of training time, while perfect knowledge is rarely attained, syntactic and semantic errors being frequent. The proficiency with which such exact languages are used, tends to depend on the frequency of usage [ZLOO75, REIS81].

1.2.3 Non-procedural languages

A number of attempts have been made at producing natural language interfaces for database systems [LEAVE74, OGDE83]. The motivation behind these developments is the fact that we do not currently have any way of ensuring that a user states exactly and precisely what he wishes to accomplish. This inaccuracy occurs at two levels; one is the lack of a suitable language in which to express objectives, and the other is the tendency humans have for being imprecise in knowing what is wanted. Impressive results have been obtained, but a number of criticisms have been raised. Such language types, although supposedly simple to learn syntactically, do not ensure that users are aware of the semantics of database interaction or the semantics of the information stored in the database. Users also may assume unrealistic intelligence from the computer, or request data which this pseudo-intelligence suggests is available. However the domain for application of such natural language interfaces exists, though it may not be as wide as first expectations have suggested.

In general, such database access techniques, like programming lang'ages provide a powerful and precise means of interaction, but they are inappropriate for the vast majority of non-programmer users. Menu selection and similar approaches are appropriate for novices and users with minimal training, but are generally too slow and contain too much explanatory text for skilled operators.

1.2.4 Graphics-assisted systems

A novel approach to the problem of finding one's way in large databases has recently been suggested [NEGR81]. The main concept of this technique, and of its subsequent variations, is their ability to allow the user to interact with a pictorially represented database in a more natural manner. This is accomplished by presenting to the user a graphical view of a database in combination with various cues/icons, in order to enable him to orient himself easily and naturally. The user is provided with a means of manipulating his position in the data space, or the data space itself, allowing for the natural exploration of the items and options offered. The range of interaction possible, although limited in the first implementations, has been enlarged in the later versions to include conventional database access.

The concept that this type of interaction is exploiting, is the natural abilities humans possess in interacting with their environment in spatial terms. Attempts are made at producing graphical representations of familiar environments (e.g. an office desk, an in-tray, a telephone, etc.) in order to smoothly bridge the gaps between conventional and electronic data manipulation. The considerable promise shown in graphics-assisted information display/interaction techniques will focus these discussions on an analysis of the human factors and concepts involved in the successful implementation of these approaches.

1.3 Human factors in the management and interaction with data

It is only recently that the bottleneck in the efficient use of database systems has been identified as located at the man-machine interface. The trend towards faster processors and larger databases has neglected the possibility of a severe mismatch between an operator's needs and requirements, and those offered by improperly designed systems. It is however generally accepted that we have moved away from the concept of introducing automation for its own sake, and that in order to achieve an efficient man-machine symbiosis, it has been accepted that human cognitive capabilities should be supported and supplemented rather than replaced [GILO76].

It is a fact that the vast majority of systems in use today require a certain amount of skills on the part of the operator. The reasons behind this are the design of database systems under machine efficiency goals and constraints, which rarely match those dictated from a human factors point of view. For example, a hierarchical tree-structured data model may be efficient in terms of storage and maintenance, but may become hopelessly confusing if the stored data cannot be readily conceptualized in tree-type form.

A question arises as to whether one or a limited number of systems can economically satisfy user requirements. A user group undertaking the same task is likely to have similar models of the task and the nature of the information involved. It may therefore be appropriate to organize information in a manner compatible to specific needs, thereby producing a number of user specific systems, rather than few and flexible ones. This is one of the reasons behind the diversity of database systems in use today.

The requirements of achieving efficient man-machine interfaces are manifold, encompassing a number of disciplines. Martin [MART73], Foley [FOLE74], Cole [COLE80], and Norman [NORM83], provide a set of general guidelines. Card [CARD83], suggests a model of the human being consisting of a number of parallel and serial processors with different characteristics, useful for predicting reactions and performance in low level tasks.

In these sections however, we will concentrate on the issues and problems of attaining efficient human interaction with database systems, while focussing on the professional but computer-naive user.

1.3.1 The human cognition of information

Data is not information, and information not necessarily informative [SHAC79, NISB76]; these simple facts have however been neglected in the current flood of data and the attempt to make as much as possible of it available to users. As a first step in our human factors oriented examination it would be useful to examine the meaning of information, in the context of how it is perceived, evaluated and used by humans. In the dictionary sense, information is defined as: "that which can potentially increase knowledge", the emphasis being on the word potentially.

Much of what people do everyday is information processing. Thinking, perceiving, remembering, judging, and the like, all involve operations on sensory data or on mental representations in order to classify, organize, or produce from them conclusions or decisions. It is clear on biological grounds alone, that people have a limited capacity to handle information. This limited capacity can however be allocated in different ways to perform different types of operations or to cope with varied conditions. The situation is similar to a general purpose processor whose organization depends upon the task to be done. Human information processing capacity however increases with practice at a steady but reducing rate. An office worker for example, will have a higher apparent information processing capacity for familiar tasks rather than for seemingly arbitrary ones which he will need to learn in order to interact with a conventional database access system.

Just as computers, human beings have their own architectures and operating systems that determine how they process information. Advances in cognitive psychology over the past several decades have revealed the varieties of architecture and operating systems underlying the information processing tasks humans usually perform [HABE82]. We now have available a number of principles that specify the optimal organization of information for various types of human information processing.

Information theory as developed by Shannon [SHAN48], was an attempt to quantitatively measure the amount of information involved in any communication. Attempts to apply conventional information theory to problems beyond communications have, in the large, failed. The early developers stressed that the information measure used was dependent only on the probabilistic structure of the communication process. It is obvious however that information value cannot be measured by a merely probabilistic measure, as this restricts its application to one narrow range of ideas. "Information theory is a deep mine comprising more than one stratum of ideas, all of which must be worked if we are to win enough intellectual material for the building of information systems, and not just telephone systems" [STAM71].

In general, the real level of information is characterized by the interaction between the information transmitted, the interpreters or users of information with their systems of value judgements, and of the goals and objectives which information serves to promote. This in turn, is directly related to the concepts of information value and utility [ZUND71]. For example, considering a professional user, the amount of information contained in a message is highly sensitive to his own current pre-occupations, his present knowledge, his confidence in the source of the message, the amount of attention he can give the message, etc. If a system is intended to provide important, relevant information to professional users, then ideally it can achieve maximum efficiency by keeping track of the shifting concerns of users themselves. They, the users, are responsible for providing much of the most important information in an organization.

A body of statements or statistics has an information content, when measured on a subjective scale, which varies as a result of the changing needs and perception of the user. The amount of information, which these data contain, is not only a function of their relative frequencies of statements or of their logical precision, but also of how they are judged by the user. Without actually attempting to measure subjective probability and derive measurements of information from it, the concept alone should force us to pay greater attention to this source of information centred in the user's perceptions. Perhaps we should pay greater attention to methods of keeping track of what users require, and devising systems that are capable of having just that information on hand.

A more thorough study of the complex information processing steps required even for simple tasks, and a recognition of the fact that the brain is organized in such a fashion that some processes can be carried out simultaneously, has led to the abandoning of the hope for a simple, all-inclusive measure. Attempts have been made to formally develop an information value theory to the level where it can be used in the design of systems where the reduction of overall uncertainty (i.e. by combining event probabilities and values), or an increase in information value, is necessary [HOWA66, ZUND71, ISHI75, STAM71].

It is clear therefore that interaction with information (say by means of an advanced database system), should incorporate some means of evaluating the importance or value of that information perhaps by monitoring its usage, in order to reduce uncertainty and therefore increase interaction efficiency. This could be achieved by an efficient utilization of available resources in an interactive man-machine dialogue where, for example, important tasks would be obvious and/or highlighted, while the support of certain functions or items could be optimized at the expense of, for example, less frequently used ones.

1.3.2 Human Memory

Any study hoping to address itself to the human perception of externally presented information, progresses naturally into the study of human memory performance. An understanding of the way information is stored and organized in human memory would contribute guidelines which might partially bridge the conceptual mismatch between the computer and the user.The study of human memory is a large subject best left to specialists in experimental psychology. Here we will concern ourselves with available results concerning the role and characteristics of human memory in the context of perception and interaction with a computer assisted information access/display system, human memory being the essential component in external information retrieval.

In experimental studies designed to test the power of recall for words, phrases, and sentences as a function of their attributes, a common and potentially useful finding has been the recognition that words and concepts with a high imagery context are recalled most successfully. The superiority of pictorial stimuli as compared to verbal ones is well known [LIND72, COHE77, EYSE77, HABE82]. In addition, the results reported suggest that:

a) the amount of detail contained in a stimulus is not a critical factor
b) recall is considerably higher when an interpretation of pictorial stimuli is provided, indicating that we must distinguish between interpreted and uninterpreted images.

World knowledge interaction is responsible for the latter effect; there exists currently an accepted general notion that the encoded representation of information incorporates a considerable amount of information not immediately present in the stimulus material, and which interacts with existing concepts stored in memory to produce more meaningful and thereby easier to recall material. For example, an image (icon) of a VDU would be more easily recognized by a computer expert than, say, a totally computer naive person. Similar results have been reported in [LIND72, CARD83] in the context of short term memory storage of meaningful items versus nonsensical ones. Enough material must however be stored in memory, in the form of past experiences, for a person to be sensitive to the organization of the externally stored/presented information; access of particular items would otherwise be impossible [COLE80, HABE82], the storage of concepts in memory being highly organized to aid their retrieval.

If human memory storage were to consist of highly specific, uninterpreted images or other items, then the retrieval of information would be extremely difficult, analogous to trying to find a photograph in a trunk full of randomly distributed photographs [EYSE77]. In an information retrieval context, the analogy would be the case of a large unorganised (spatially unencoded) presentation of a menu type query, where a verbal description of an item would offer no immediate clues as to its attributes or location [SNOW83]. With practice world knowledge accumulates in the form of the spatial positioning of the item (if unchanged), to make subsequent interaction faster.

It is also known that recognition of information is much easier than total recall. An investigation into the tip-of-the-tongue phenomenon [EYSE77], has shown that in tasks where subjects had a 'feeling' for the answer but could not completely recall it, a large proportion were able to recall correctly when the answer had been presented as a member of a multiple-choice type questionnaire, suggesting that reminding should be an important function of an information system. Such a conclusion is strengthened by investigations into the way in which individuals organize their desks [MALO83] where it is reported that an important function of the spatial location of items on desks is to act as a reminder of tasks to be done.

1.3.3 Information display

In this section we will touch briefly on some of the issues important in the design of human compatible information displays. There are numerous approaches to this task, but a human factors oriented viewpoint will be taken.

As Cole [COLE80] suggests, the human component places a set of constraints on the way in which data is displayed. Among other factors, these include spatial reference, order, association, simplicity, accessibility, responsiveness, control, reliability and support. In addition to the above, studies have been carried out into human visual perception and how it can be linked to the design of efficient information displays. Lindsay [LIND72], provides a psychologist's view of the human visual system, and of the mental processes associated with visual perception; Budrikis [BUDR72], discusses measures of visual fidelity as criteria in visual communication; Roufs [ROUF80], also discusses image quality criteria and their link to visual processes in the context of alphanumeric display design; Haber [HABE82], discusses the main human perceptual mechanisms, pointing out the well developed human capabilities of storing and interacting with visual images. Kinney [KINN79], Krebs [KREB79], and Teichner [TEIC79], suggest guidelines for the design of efficient displays, while current trends in displays and display technology are reviewed in [CHAN80A, NEWM76, MACH77].

Some general guidelines for the presentation of information on graphic displays are [COLE80]:

a) logical sequencing: the information must be presented to the user in a logical sequence
b) spaciousness: avoid cluttering the display
c) relevance: irrelevant information confuses and misleads users
d) consistency: users develop expectations by forming conceptual models of display functions, which they expect to be met

The use of graphics and graphic symbols, is increasing as icon type displays can be used to significantly compress information. Graphic symbols represent a lot of information in little space, space being at a premium on display screens. The use of graphics/icons, has initiated a number of attempts at evaluating their optimum use [ISHI75, HEME82].

The main conclusions from studies such as above, center on the superior abilities humans possess for visual processing, remembering and spatial organization. The particular strengths of the human visual system lie in its abilities to encode multidimensional information rapidly and enduringly. It must be remembered however that whatever display method is used, what is 'simple enough' to be quickly and easily assimilated is very much relative to the particular user.

1.3.4 Information Retrieval

Perhaps the best way of thinking about information retrieval is to consider retrieval systems as communication systems. Unlike television, which is one-to-many, or telephone, which is one-to-one, retrieval systems are few-to-few. The contributor to a computer information system knows that he/she is addressing some set of people, but does not know who they are. The user suspects that in a gigantic mass of irrelevant information there are a few items that meet his/her interests, and often the task of retrieval is made difficult by the need for the user to assess and combine probabilistic evidence in addition to the use of a coding scheme. Both the contributor and user must share a coding scheme that allows the user to find the contributor's record [MART80].

Athough it is important to understand how users conceptualize the functions and operations of a computer, we also need to know how these concepts interact with the user's mental processes, and how these mental processes interact with the system environment. This would allow us to be judicious in terms of which concepts to concentrate on. Without this knowledge it is possible that, in trying to fulfil all the previous conceptual conditions, we may end up with a system which is too unwieldy to use.

The retrieval process can be modelled as a matching process detecting the resemblance or the similarity between requests and actual information objects [NEGO76, BOOK76]. Therefore, information retrieval has two facets: language for description and search strategy for retrieval. In indexing, we are replacing the content with a few simple descriptors, and these descriptors serve as the basis for making decisions on what information objects we want to retrieve.

In essence we can view the storage and retrieval of information as a special kind of problem solving behaviour. Typically, in a job we must achieve some goal, but there is a state of uncertainty as to how to go about it. It is often necessary to retrieve a document from the filing system to assist us; the problem is, which information is relevant and where is it? We must interpret our mental model of the filing system in order to answer these questions [COLE80].

Regardless of the memory system we contemplate, be it the human brain, a library card catalog, a collection of file folders, or a computerised information system, there are types of queries for which the organizational structures of the memory are inappropriate, even though one may know the information is there. The question to ask then is, how does one design an information system such that we may find the information we want, and what kind of retrieval strategies are required? If the type of questions to be asked are known beforehand, then the task may not be too difficult. But what about questions that were not anticipated? It might at first appear that the objective [LIND72] is "to build an information system that like human memory, can answer almost anything that comes to mind about the stored information".

At present, information systems are devised the other way round. They are built upon a foundation of clerical systems which are further extended in the hope that, by having larger and larger files, we shall eventually have all the answers a user may require.

The reason we use filing systems is to compensate for our own inadequacy in retrieving large quantities of information from memory. As with memory we can easily retrieve familiar information, but with unfamiliar information we have to have a rigid retrieval structure to enable us to negotiate the storage structure and access the desired information. In this situation, the storage structure is well defined and is actually open to view and therefore easier to use, with information access more reliable than is the case with memory.

In imposing a structure on information in our memory we form categories; these categories will depend on the context in which information is sought, a fact which tends to suggest that we should concentrate on retrieval structure rather than storage structure. Although in filing systems we impose a basic storage structure which doesn't change, the retrieval strategy for particular information will depend on the context under which it is being sought.

In fact the performance of human memory has been noted to depend more on the retrieval strategy adopted (as for example in word/image association), than on the amount of information stored in memory [EYSE77]. It would therefore seem correct to assume that an intuitively satisfying coding scheme (perhaps using associative, memory coding schemes, similar to the ones used by humans) would be better matched to the information retrieval task: an example would be the use of a telephone icon to access an inter-office message function in an office information system.

1.3.5 Data models and conceptual models

The types of data manipulated in database systems are numerous. In conventional systems the data may be alphanumeric and arranged in the form of tables, adequate for a particular set of applications such as information on the inventory of a company. In an office type situation the data may include digitized verbal messages, inter-office letters, documents, etc. In simpler consumer type applications, alphanumeric data assisted by simple graphic displays may be adequate. In pictorial systems, a large number of high quality images may require efficient storage, retrieval and manipulation.

The way in which data is represented, the data model, varies considerably. A data model is defined [MCGE76], as: "any formally definable class of data structures which can be used as the basis for data processing applications". Tree type data models appear to be most succesful when the data are perceived to have a natural tree structure, but are cumbersome otherwise. The relational model is more elegant, but critics have complained that it is too 'syntactic' and that models which can represent more semantic information are preferred. Network models permit sophisticated structures to be described, but this complexity introduces difficulties [SHNE78].

The type of data and the data model used depend therefore very much on the type of application, combinations of these and other data models within a single application being possible. Although machine efficiency issues may intervene in the selection of a data model, every effort should be made to provide the schema easiest to conceptualize.

McGee [MCGE76] provides a set of criteria for evaluating data models based on use criteria, i.e. a measure of the useability of the model; these include:

1. Simplicity: a data model should have the smallest number of structure types, composition rules and attributes
2. Elegance: a model should be as simple as possible for a given direct modeling capability
3. Picturability: model structures should be in pictorial form or lend themselves easily to visualization
4. Modeling directness: a model should not provide equivalent direct modeling techniques

Although these are a starting point, they remain to be universally accepted. A problem with these criteria is that we may obtain conflicting impressions from different users for, say, the relative simplicity of two data models [SHNE78].

In addition to data models, it has been noted that users tend to form conceptual models of computer functions and dialogue structures [COLE8Ø, BEWL83, HALA83, GENT83, CARD83, CARE82]. It may, for example, be advantageous to regard a menu driven dialogue in terms of a tree model or to think of a relational data model in terms of relational tables, even though the internal machine representation may be very different, a conceptual (mental) model being a representation of a system's internal mechanics, i.e. its components and their behaviour.

The reason for the formation of these conceptual models is to allow users to formulate correct queries and to approximately predict system responses, the type of model formed depending on the way in which the data manipulation operations are perceived by a user. In spite of the logical appeal of mental models, there is presently no detailed psychological evidence that they are actually used in reasoning about systems. It has however been found experimentally [HALA83], that when an explicit model of a simple calculator's (RPN) stack operations are provided, then for innovative (i.e. novel) tasks, users to which this model had been explained performed better than the no-model users. Such results tend to reinforce the importance of mental model formation in system use and understanding.

Conceptual models are also used in navigating, i.e. when used properly, can tell the user what stages he has reached, for example, in a dialogue structure. The appropriateness and compatibility of such models depends on how system functions and operations are perceived by users, and this in turn depends on past experiences gained by users in the field of information handling. It is logical therefore to assume that interaction with computers would be enhanced if this interaction were in terms of concepts with which users are already familiar and have gained through previous information handling experience. With current systems in use, this is not always the case (see review of conventional approaches in section 1.4), and is the major reason why naive users must be taught these concepts as they do not have them stored in memory due to the lack of previous experience with computers. These concepts are as follows [COLE80]:

1. verbal: verbal concepts include the language of interaction
2. spatial: refers to the way in which information is displayed and conceptualized as flowing in the system
3. temporal: refers to the sequence of actions which must be followed in an interaction mode
4. task specific: refers to the way users visualize their task
5. abstract functional: this includes all available functions for manipulating data

Summarizing, the more compatible we can make the above concepts with a user's past experiences and abilities in information handling, the more easily they will be attained, and the easier the system will be to use. As regards database systems, the following question therefore arises: How do we organize the coordinated storage and access to data such as to enable subsequent retrieval to be efficient and meaningful, or in other words in terms of concepts which are already familiar and meaningful to users? Cole [COLE80], provides some of the answers:

1. The conceptual structure of the database should conform to the semantic relationship between its elements
2. The language used to interrogate the database should allow for the direct expression of the different types of relationships

Approaches to database access systems design which attempt to satisfy some of the above requirements and criteria will be presented in later sections.

1.3.6 Spatiality and the use of instinctive skills

In the previous sections, the need has been mentioned of users to interact with computerized information systems in terms of concepts with which they are already familiar and understand. The aim is to facilitate interaction by making compatible a system's functions and operations and a user's experiences and previously attained skills.

There is one major difference between computer database access and library information access. Whereas in computer aided access it is necessary to enter an appropriate set of descriptors to access the required information, in a library situation the descriptors have to be converted into a classification number, which is in turn converted into a spatial location containing constituent items in alphabetical order. Therefore, in the library, users develop a spatial model of information in addition to the one based on the relationship between information descriptors. This means that if the location of the desired section of information can be remembered, a user can go straight to it, without index mediation, and scan for the required book.

More specifically, human beings code and remember the spatial positions of significant objects. They also appear to encode and remember certain characteristics of the change of positions of objects (movements). Experimental studies [EYSE77,PAIV71] indicate that imagery is a powerful predictor of recall, and therefore high imagery concepts have been better retained than abstract ones in item recollection types of studies. Comparisons between spatial and temporal memory have also produced higher scores when pictures were used as compared to words. These, and other results suggest that in addition to the apparent parallel processing of visual images, imagery facilitates recall either by providing a second means of storage in addition to the verbal, or because retrieval from a visual store is more efficient than retrieval from a verbal store. Pictorial information is commonly processed thoroughly, both imaginally and verbally, whereas concrete and abstract words and phrases are weakly processed if at all.

Indications exist that visual imagery is specialized for parallel processing in a spatial sense [PAIV71]. The recall of imagery can be improved by an organization of the material. The presence or absence of spatial organization, is an important characteristic, and has an effect on its subsequent recall [EYSE77].

The everyday information of the world we live in is organised in terms of various conceptual 'labels', and also in terms of the conceptual 'labels' of other information items, both by virtue of meaning (semantics) and often by its relative position in time and space. Information handled every day (except perhaps for previously learnt factual knowledge) is rarely independent of location and time. Consequently, without knowledge that we had received a particular item at some time, and without a 'cognitive map' of where we put it in relation to other items, we could not find it. Conversely, without a concept of identity we would not 'know' what we were looking for.

The extensive use of 'cognitive maps' as an aid in the orientation and recall of information is known [COLE80]. Cognitive maps are ubiquitous. We all know what to expect around the corner on the way home; we can all plan trips to destinations and check our expectations along the way if we undertake one. These abilities are commonplace but by no means uninteresting.

This concept of spatiality is in use in everyday life. It refers to the way in which humans use imagery to aid the learning and interaction with the outside world. It is natural for humans to interact with various items in terms of their visual attributes relative to one another and/or a backround. With respect to the retrieval of information it is known that it is easier for humans to recognize information rather than recall the same information accurately from memory. This is one of the reasons behind the success of menu type interactions with untrained users [CARD82, NORM83].

As mentioned in section 1.3.2 there are indications that while users may have an idea for the answer to a question, they are able to find the answer precisely with greater speed and accuracy when it is presented as part of a multiple choice format, typical of menu type queries. In summary, when one knows what to look for in terms of item descriptors (e.g. spatial location, colour, etc) then it is easier to find it [LIND72].

As we learn an information-processing task, e.g. organizing our desk, our measured capacity increases with continued practice, although at a reduced rate. This isn't surprising for it is very much as one would arrange a computer: hard-wired often performed functions in the interest of speed but using software routines to do tasks that are infrequent or subject to change. This results in efficient use of time and available resources [SHER74]. Any theory of cognition in relation to information handling must give due consideration to these points.

This sense of spatiality is instinctive and well developed in humans through the training of a lifetime in negotiating real space. By applying such a concept to information systems we may appeal to the class of users for whom directness and immediacy are essential qualities in an interactive interface. Spatiality may serve as a tool for correcting the mismatch between our instinctive skills and the non-intuitive, symbolic space used thus far in conventional filing systems. Spatiality must not be considered as being in competition with data handling on a symbolic basis, but as complementary, interfacing system functions which are beyond our immediate perception.

Summarizing, it is certain that imagery can have a profound effect on the way we conceptualize and store information, this strategy of tapping the user's natural ability to organize and reference information spatially having until recently been neglected. The economic reasons related to the use of graphics, have decreased in importance, and a number of systems utilizing the principle have been proposed. An overview and assessment is presented in chapter 2.

1.4 Current Approaches

1.4.1 Conventional Query Languages, Descriptions and Comparisons

By conventional query languages we take to mean those based on alphanumeric queries or menu selections operating on well defined databases. Users of such databases are usually confident as to the nature of their task, the formulation of their query, and operate on an accurate conceptual model of the database. These three factors are however contrasted in new users of database systems. Many conventional database query techniques are subject to a strict set of syntactic rules, and in general query formulation becomes more complex as the questions asked become more complicated. The extent to which this complication occurs depends on the database structure, and on how well it conforms to the type of questions being asked. It is ususally the rule rather than the exception that lengthy training times of the order of hours are necessary for new users to attain a complete level of proficiency, even though many languages such as Query By Example (QBE) allow users to learn up to the point they regard as useful. The term query language as used here, may be taken to mean a language system with extensive hardware and software support of the relevant facilities as well as a transportable piece of software.

1.4.1.1 Some Examples

1. SEQUEL (renamed SQL): SQL is an acronym for Structured Query Language. SQL utilises english keywords with which to express queries, and provides a range of update as well as retrieval functions. The level of the language is comparable to that of relational algebra.

2. SQUARE (Specifying Queries As Relational Expressions): SQUARE utilizes a mathematical notation to express queries against a database. The central concept in SQUARE and SQL is that of a 'mapping' which returns the data values in some columns which are associated with a known data value in another column, in a relational table.

3. QUEL: QUEL uses the relational calculus concept and has been implemented in the INGRES system [HELD75]. It uses a 'tuple variable' ranging over a relation whose permitted values are tuples of that relation.

4. TABLET: TABLET is also based on the relational model of data and is similar to SEQUEL (SQL) but more procedural [WELT81]. It specifies the operations that are to be performed on base relations in order to produce the desired results.

Two other important techniques exist which operate on relational data structures. Their main difference from the above is that they use a simple graphic display to aid query formulation.

5. QBE (Query By Example): proposed by M. Zloof [ZLOO75], it is aimed at the professional and the casual users who may learn the language only up to the point dictated by their requirements. QBE operates on the relational model of data and formulates a query by presenting to a user relational table(s), and asking him to fill in the appropriate table rows with an example of a possible answer. In fact, when filling in queries, two entities are distinguished: the example elements, and the constant elements. Linking variables are used to join relation tables. QBE may also operate on hierarchical databases [ZLOO76], and has been modified to work in an office environment [ZLOO81].

6. CUPID: CUPID is a language implemented on top of QUEL, but which uses graphic representations to aid the construction of queries. The user constructs queries by simple light pen manipulations of a small number of standard symbols.

Many more query languages have been proposed for the relational data model tailored to a specific application area. Taking an example from the computer graphics/image processing field: QPE (Query By Pictorial Example), GRAIN, IQ, IDMS and many more. Each has its own special attributes adapting it to a specific task, and these are reviewed in [CHAN81]. The above is a limited selection of the query languages (systems?) available in relational database applications. Two examples of languages operating on different data models are listed belcw:

7. ZOG: ZOG is a menu type approach to database interaction [ROBE79]. The computer guides the user through the database by

supplying a list of the options available. Although easy to learn it suffers from a common disadvantage of such menu driven techniques of not being able to answer 'what if' questions easily. Such techniques are however successful when applied to the appropriate domains.

8. Other menu type dialogues. One example [APPE83, SPEN77], was not originally intended for database access but has evolved from ideas developed in a Computer-Aided Circuit Design system.

9. MIDAS (a Multi Sensor Image Database System): MIDAS operates on a hierarchical data structure in order to store symbolic representations of images in a pictorial database system.

The relational approach may be combined with the hierarchical one in an attempt at simplifying the data model. An example applied to pictorial databases is presented in [CHAN80B], where a set of logical pictures stored and manipulated in relational tables is represented by a hierarchical structured collection of real picture objects, corresponding to the logical pictures manipulated.

1.4.1.2 A comparison of conventional query techniques

The design of any query language (system) is in part an attempt to match the capabilities of a supporting system to an estimate of a user's needs. This is the main reason behind the existence of a large number of different approaches to essentially the same problem. The conflicting requirements of efficiency of information retrieval, and the need to reduce training times while increasing the 'naturalness' of the language have resulted in a number of compromised solutions. Systems which are easy to learn are usually limited in their flexibility and in their ability to respond to a diverse range of queries. More complex systems however tend to be limited as to the ease of interfacing to new users. Proficient use of these also tends to depend on the frequency of usage.

Comparisons between existing query language systems are few, and tend to concentrate on systems intended for similar tasks. SQUARE and SQL have been compared [REIS75] with respect to ease of use by non-programmer professionals and usability. An investigation was made on the type of errors frequently made and on how they may be corrected from the system side, improving the user interface. The results indicated that about 12 to 14 hours of academic instruction are necessary, programmers being the quickest learners as compared to newly introduced users. A small advantage of SQL over SQUARE has been suggested in terms of errors made in formulating queries. Having however learned both languages, the subjects used attained an approximately equal level of proficiency. These tests indicate that human beings tend to make repetitive errors in syntax and spelling when corrective feedback is not used, and therefore techniques such as stem matching or synonym directories may be of use in alphanumeric queries.

QBE, SQL and an algebraic language have also been compared in relative terms [GREE78]. The comparisons concentrate on comparing the time, accuracy, and confidence of subjects in formulating queries, and in comparing the learning and application potential of each approach. The results indicate that Query By Example is superior in terms of training time, time taken to formulate a query, and proportion of successful queries formed. The subjects used indicated that they felt more confident in queries formulated in Query By Example. It is suggested that QBE's explicit tabular organization aids in query formulation particularly when cross referencing of columns and/or tables is required. Enhancement of QBE with english-like commands is suggested.

SQL and TABLET have been compared in order to examine the effects of language procedurality [WELT81], the languages being similar in other respects. The results support the hypothesis that people often write difficult queries correctly using a procedural query language, than when using a nonprocedural one. For easier queries, no significant differences were reported.

In an isolated psychological study of QBE [THOM75], the time taken to learn the language and the time, accuracy, and confidence in translating stated questions into QBE was assessed. The results indicate that QBE is simple to learn requiring about a third of the training time of SQL and SQUARE, while about equally accurate in query formulation. In addition, it is suggested that QBE is relatively robust against forgetting, and that in some cases it helps the understanding of complex queries as compared to their english language counterparts.

An overall description, summary, and experimental comments of these studies, pointing out possible failing points and offering guidelines for future experimentation is available in [REIS77, REIS81].

The main criticism with the use of exact query techniques was the 'unnaturalness' of the dialogues used. Other complaints center on the small error tolerance offered, with respect to grammatical and minor syntactic errors. Simple, computer assisted instruction such as stem matching or query verification has been suggested as a possible improvement.

Similar comparative studies for menu orientea interactions have not been performed on a similar scale. There exist however isolated suggestions on the optimum approach for designing menu type dialogues. Savage [SAVA82], suggests that menus should have no more than 9 options and that levels should be restricted to 4. He concludes that users prefer shorter menus with more levels, and that help texts should be available at each level. Interestingly, in [TOMB82] it has been suggested that in some cases, users preferred to search from within an alphabetical directory than use a menu. Brown [BROW82], indicates that if the underlying information has a stable well defined structure, then we should allow the menu network to reflect that structure. If the need for shortcuts in the network is obvious, they should be made available.

In Miller's [MILL81] exploration of the depth/breadth tradeoff in menus, an optimum value of 2 levels and 8 choices per level was found as optimal for his particular example. Miller recommends an expansion in breadth over depth which possibly contradicts [SAVA82], but agrees with the recommendations in [TOMB82]. In Snowberry [SNOW83], results similar to [MILL81] were reported. However when testing the extreme case of one level with a breadth of 64 and with options ordered according to category, this was found as the optimal case. Such a result would agree with the study in [TOMB82]. It is obvious however that practical limits in breadth size must be investigated.

A problem with this type of study lies in the sensitivity of users to the particular menu dialogue example used, in terms of actual content and menu presentation, which accounts for some apparent contradictions.

A problem with menus is the fact that users readily get lost [ROBE79], indicating that in some cases the amount of information and help facilities should be increased. A number of features desirable in menu dialogues has been put forward by Apperley [APPE83]. The dialogue should be stable, users should at all times be aware of their state, default values (possibly context dependent) may be used, and that the dialogue should inspire familiarity and confidence to the user.

A brief comparison between menu and other command languages is available in [NORM83], while general criticisms and descriptions of dialogues based on the use of graphics assisted interaction are available in [FOLE74, SPEN77].

Summarizing the results for menu type dialogues, we find the need for providing the user with a familiar environment supporting extensive help and state awareness facilities. The use of uncluttered displays with a limited number of well specified, easily understandable options is obvious.

In spite of some criticisms, the QBE approach seems to be the most favourably accepted in its application domain. Simple, well structured menu dialogues are acceptable in certain applications. The need of users to communicate with computers in terms of familiar concepts can partly explain these first results. The small advantage SQL claims over SQUARE is partly due to the use of English keywords, these being more familiar and instinctive than relational algebra constructions. The better results obtained when procedurality is tested would indicate that for complex tasks, humans feel the need to break a task down in ordered parts which can be tackled more easily in isolation. As regards QBE, one would expect that a person who understands a question will be capable of easily producing an example answer. This is the basic principle which QBE adopts, for simple queries. In the words of M. Zloof [ZLOO76], "QBE gives the user the sense of manually manipulating objects. QBE also requires the user to master only a few simple concepts initially". More complex queries can be more difficult to formulate. The basis however of: using easily constructed examples as the foundation in all query formulations, the use of simple tabular graphics to aid the visualization of relational tables, and the near wordlessness of the system, make it less easily forgettable, giving QBE an advantage over similar methods.

1.4.2 The Rigidity of Conventional Approaches

The origin of a number of problems associated with current database interaction systems can be traced to their similarity with conventional filing systems as found in libraries for example, and the ignorance to a large extent of the presence of the human element. The reasons behind this initial use of conventional filing systems are straightforward. They were designed to compensate for our own inadequacy in retrieving from memory large quantities of information. Even though we may retrieve familiar information from memory or elsewhere easily, rigid retrieval structures allow us to negotiate storage structures such that information whose location or even existence is unknown, may be located. This emulation of filing structures by computers has caused a number of problems. For example, having retrieved an item of information one usually knows by assigning cues such as colour and location where that item is located, such that subsequent retrieval is efficient. Rigid filing systems frustrate this natural tendency, by penetrating all levels of the man-machine interface. It should be possible to allow for the easy location of known and frequently used items, without having to negotiate a complex access structure irrespectively of the type of information being retrieved.

Browsing, i.e. the leisurely search through a database without a precise idea of what is being sought is rarely provided, although some recommendations for systems designed for browsing have been made [LECL82].

The use of symbolism and rigid access strategies are certainly necessary, when used in the appropriate context. The point is that human information handling activities cover a broad spectrum, of which only a small part is concerned with the handling of unfamiliar information. Thus instead of using one particular technique to cater for all cases of possible activities, we should allow a degree of adaptation. In particular, when a user is dealing with familiar information, the use of strict symbolism is unnecessary. Natural cues such as the relative position of an item, its shape and size are more useful and efficent in such a context. Conventional systems do not usually accommodate such techniques. Keyboards are used which are in general accepted as a low capacity medium of the wrong type and mode. Humans feel the need to interact with information directly, for example by annotating or by personalizing the organization and presentation of information. Another failing of conventional systems is that rarely a global view of the data space and the structures one is interacting with are presented in a meaningful way to allow users to assess their current state in the system; nor do the majority of systems cater for the natural tendency humans have for being imprecise in forming queries. It is also rarely possible to test or examine options in order for example, to verify a selection and its consequences, without complex and/or tedious reversive action.

1.5 Conclusions

This survey on the human cognition of information has pointed out the importance of understanding human memory processes. It is known that human information processing and memory capacity are limited, though this capacity increases with practice. The superiority of human memory in interacting with visually presented data, as compared to abstract symbolic, and the higher ability for item recognition than recall, have been noted, and are well documented. The use of imagery, however, should be restricted to the use of concepts meaningful to the user, as his past experiences and knowledge play an important role in the cognition, memorizing and subsequent use of information. Results such as the above would point towards the use of meaningful information interaction techniques dealing with visually familiar information.

It would therefore be desirable to interact with a computer in terms of concepts that users have already available through previous information handling experience, as naive users will not have concepts gained through previous training. One first step towards this direction would be to design an information system to emulate the familiar environment of the user such as, for example, his office desk, and indeed some initial approaches have attempted this [SMIT82, BEWL83, WILL83].

It has been mentioned that humans have a tendency to form conceptual models of the computer system they interact with, and therefore as an aid to this conceptual model formation a simple, consistent pictorial representation of the interaction dialogue has been suggested as a necessary feature of information systems [MCGE76]. The successful formation of such mental models can enhance dialogue effectiveness, as users would thereby be able to approximately predict system responses and gain confidence in interaction. The usefulness of a dialogue structure may be increased as has been suggested in [BROW82, APPE83], by monitoring usage and using default paths to adapt the system to the operator's increasing interaction expertise.

In addition, and possibly as a result of the highly developed human capability of interacting with visual information, there exists a well developed sense of interacting with our environment in spatial terms. Such abilities have however until recently not been exploited in information systems design. Humans use spatiality to form cognitive maps of their environment and of their office surroundings. The approximate positioning of items is recalled through their attributes, such as for example relative positioning. With reference to office desktop organization [MALO83], it has been noted that: "the method of accessing information on the basis of its spatial location, instead of its logical classification, is an important feature of the way people organize their desktops that might profitably be incorporated into computer based information systems". As Cole [COLE80] notes: "spatial imagery obviously plays an important role in our conceptual model of our real world information environment".

On a computer, however, this is reduced to a partial view of the information environment through the two-dimensional window of the VDU. Consequently, people find it difficult to 'navigate' about a database, especially as they are forced to do this in terms of abstract, modified verbal codes. If ways could be found to reproduce a simulated 'spatial environment' on a computer, it might be of benefit to naive users in terms of accessing information; they would have a concept of 'where' instead of trying to remember verbal codes. The concept of spatiality, however powerful, should nevertheless be employed with an awareness of the manifold requirements that other research considers important in designing a succesful man-machine interface, ranging from physiological to psychological factors other than spatiality.

Some of the proposals to be presented and evaluated in the following chapter depart from the idea of emulating a familiar environment such as a desktop. Instead, an equally comfortable, instinctive abilities based environment is provided, which may therefore be designed with the objective of matching human factors based requirements such as those presented in this chapter, removing design constraints based on conventionality.

CHAPTER 2

OFFICE INFORMATION SYSTEMS EMPLOYING SPATIALITY

2.1 Introduction

This chapter will summarize the suggestions made in chapter 1, and examine their effect on the design of practical information display/interaction interfaces, with an emphasis into their integration in office information systems.

An office is defined as a place that primarily deals with the acquisition, storage, transformation and presentation of information. An integrated office information system will have to handle all such information processing functions effectively, to achieve any measure of success. Firstly, some of the fundamental results available on the user's psychology of interaction with information are reviewed in the context of office information systems design.

2.2 Issues in Office Information Systems design

The design of office information systems, in contrast to those designed for use by computer experts, requires the acquisition of a different viewpoint by the system designer. A number of factors are responsible for the slow acceptance of computational assistance in offices. This is particularly true for systems designed for use by professionals rather than secretaries, clerks, etc, or in general, personnel whose job functions are more well defined. Many of these factors are associated with human attitudes and behaviour rather than limitations set by technology. For example, constraints are imposed by the lack of enthusiasm by the typical professional to be trained in the use of complex command languages. Another impediment is the sensitivity of professionals to the appearance of their office and particularly to the introduction of keyboards [SPEN82]. Interestingly, it is the professional class of workers who have the most to gain from an assistance to their information handling functions, since they may then concentrate on the more important tasks of processing that information.

The potential of the computer and other electronic techniques to assist these information handling activities is large but underexploited [SPEN82]. A number of factors needs to be taken into account in the automation of an office environment. The slow rate at which office workers are willing to accept changes in their environment indicates that the 'office of the future' is a journey and not a destination, in the sense that changes will probably need to occur incrementally. Towards that end, it is however useful to include, in our assessment of possible implementations, proposals involving drastic alterations to the conventional ideas of an office environment, such that a range of useful ideas and proposals may be selected and evaluated.

In terms of the advantages of automating information handling tasks in an office, a recent report has examined a number of professionals and approximately evaluated the benefits to be gained from the introduction of efficient information retrieval technology [HAYM80]. In almost all cases the questioned professionals were able to assess the amount of their time which can be saved to within 10-30%.

The 'ideal' model office, exemplified by the professional who spends most of his day dictating letters, and reading incoming correspondence does not exist. Professionals require a comfortable environment providing more functions than just the pure display and transmission/reception of text files. It is true to say however that people want to escape from the mass of paperwork, 90% of which seems extraneous, hindering rather than helping the more important goal of being able to reach good decisions and communicate them with precision.

It is generally accepted that people do not always access information through strict symbolic means. In fact, many office workers prefer to browse through the available information in order to locate items. Such facilities are not normally offered in current systems.

We can conveniently divide the professional person's activities into thinking phases and acting phases. The action phase may be easily improved, is within current technological capabilities, and conceptually comfortable. The thinking phase however requires more care, as we may easily hinder it through distractions and constrained capabilities; it presents a potentially more serious hurdle. We must therefore concentrate on the concepts of information handling rather than implementation details, and make these seem comfortable to the user. As quoted in [COUL79]: "users need to accept office systems before they will use them, and to do so they must understand them". A new viewpoint is therefore required when examining the problem of efficient office information systems design. This would necessitate the examination of basic human perceptual mechanisms and their effects on efficient information systems design, as reviewed in chapter 1 of this thesis.

2.3 Information systems exploiting spatiality

One of the human factors mentioned in the previous chapter is the sense of spatiality, a potentially very powerful medium for use in the design of man-machine interfaces. The strategy of tapping this natural ability to organize experience spatially, has however not been exploited on any scale due to the high cost of computer graphics hardware, and because typical systems have not yet offered modes for presenting the user with a sufficiently commodious and interactive virtual space. Typically, systems offer information retrieval on a symbolic or name basis, and an intuitive sense of spatiality cannot play an organizational role. Recently however, the falling cost of electronics hardware has reduced the importance of the cost factor, and as a result, a number of systems exploiting the sense of spatiality have been proposed. They are mainly intended for the class of user for whom directness and immediacy are of importance, as for example in military or office information systems. Extensions which allow these systems to be used in a more generalized set of applications have been proposed.

A number of common approaches is shared in these novel approaches to database interaction. Visual and auditory cues are used to indicate to users their state in a dialogue structure and/or to draw attention to important items. Tree type data structures with individual variations are mainly employed, a multiplicity of data access ports being available. Users can 'move' in the data spaces in a real sense, going to where data is located rather than by relying on the referencing of items by symbolic names. 'Personalization' of the data space is allowed, enabling users to create an environment suited to their individual needs. Emulation of conventional data handling tasks can be provided in an attempt at exploiting existing data handling skills. Conversely, recent suggestions [SPEN82], do not always accept that complete emulation of familiar environments is the optimal approach, relying instead on the provision of other interactive facilities which are however within immediate intuitive perception.

In summary, the underlying aim of this new direction in information systems development, though one which should be common to all approaches, is to provide systems that are suited for use by the person with little computer knowledge, and with no time or inclination to learn specialized skills. The objective is to allow users to concentrate on the information processing function as such, simply, and without a significant intrusion to their perception of conventional information handling activities.

The following section will describe the primary features of current, interactive information systems employing spatiality, a first order relative assessment being attempted based on initial user reactions and an estimate of future usage requirements.

2.3.1 SDMS and related approaches

The first database access system to make full use of spatiality by employing extensive graphical representations is the Spatial Data Management System, developed at MIT [FIEL77, DONE78, BOLT79, BOLT81, NEGR81]. It consists of a large, wall sized graphics display and a pair of two smaller graphics screens arranged on either side of a user as in Figure 2.1.

The user interacts with a large, graphically represented database by 'navigating' within this graphic space. A close up view of his current position is available on the large screen, and a global view of the entire data space is presented in one of the smaller screens, the other serving to allow for more detailed interaction. Sound cues are available to aid the location of items. The user is therefore aware of the contents of the data space near his current position, and of his position relative to the whole of the data space. The database structure used is a form of tree structure (Figure 2.2), with later variations that allowed transitions within levels without needing to backtrack to parent nodes, the model being termed a laminar tree structure. A pressure sensitive joystick allows the user to manipulate his position in the data space, allowing him to 'fly' through it. This 'flight' is in 2½D, as full 3D movement it is commented, would probably confuse.

The data space consists of several 'iconized' items such as photographs, letters, maps, etc, that upon a closer examination (zooming) acquire more and more detail making available the full extent of their contents. The user is able to move across the deeper levels of the data structure, while his position relative to other items is maintained in the smaller screen. The user is therefore aware of items through their visual attributes, spatial location, context, etc.

SDMS is a milestone, as it was the first system to offer a fully spatially oriented database to users, doing away with keyboards. It was however mainly intended as a prototype, to stimulate interest in the field, and is not offered as an alternative in competition with data handling on a symbolic basis, but as complementary thereto. SDMS designers have in short, attempted to "enlist the user's understanding of space for the purposes of managing a very large database, accessed in a comfortable and natural manner".

An interesting result obtained from SDMS is the readiness with which users discuss data in a spatial manner, the spatial idiom taking hold immediately, and persisting beyond live demonstrations of the system. If we define as command language the set of 'instructions' needed to interact satisfactorily with a database system, then new users of SDMS were reported as mastering the 'command language' within 2 minutes of sitting down. They "rapidly acquired the analogues to their own understanding of space, and exhibited all the behaviour we would expect of persons familiar with a particular environment" [DONE78].

Some problems with the system were the tree-structured hierarchy, and its lack of size and flexibility. When users zoomed further down into the tree structure, orientation became a difficult task, as spatial orientation was not available in the lower levels of the logical tree structure. There seems to exist a tradeoff between item visual simplicity and item information content. For example, items looking complex visually were hard to interpret while the opposite is true for visually simpler items. The implication is that the more simple the data space the better, and that the data space in terms of its logical arrangement, its extent, etc, must be directly and perceptually plausible to the user. This would agree with the comment made in chapter 1, that meaningful images are more efficiently retained and recognized by humans.

A variation of the SDMS approach has been developed at the Computer Corporation of America (CCA) [HERO80A, HERO80B, FRIE82]. The central, expensive large screen and the two smaller monitors have been replaced by a workstation containing three equally sized monitors, serving basically the same functions as in SDMS. This approach is again reported as easy to learn [HERO80B], but the comment is made that on its own it is not well suited to circumstances which require locating a number of entities meeting a stated criterion.

Later versions of SDMS include provisions for more formal queries, the prototype implementation being used to access a database of ships and being currently in active service aboard a U.S. aircraft carrier. Access is made by a combination of graphical commands and formal query language accesses, presenting a combination of iconic and symbolic data presentation, access and manipulation. The database viewed through the spatial data management system is claimed as more accessible and its structure more apparent than through a conventional DBMS, requiring less prior knowledge of its structure and contents. Browsing is encouraged and queries are allowed to be generated graphically. However it is acknowledged that the cost of representing a large database graphically is high.

Within the CCA system the formal query language used is SQUEL, a combination of QUEL, and additions designed for the graphical environment. As a tuple is selected from the relational tables it is passed to SDMS which performs an indicated graphic action, such as for example highlighting or flashing.

2.3.2 Single Display approaches

A number of alternative approaches to graphics-assisted database interaction have been proposed [SMIT82, BEWL83, WILL83]. The central concept in these approaches has been to reduce the cost associated with the use and support of multiple colour graphics display units as in the SDMS and CCA systems, while maintaining some of their desirable features. A single display unit is used, on which are displayed various items or their iconic representations, as shown in Figure 2.3. Means are provided for the easy manipulation and access of items. For example, items which emulate documents may be moved, stored, stacked, retrieved, operated upon, etc, in an attempt at emulating familiar information handling activities of a conventional office. Iconic representations are used to visually identify information processing functions such as the transmission, reception, deletion, and manipulation of documents or messages. The limited display area available calls for a sophisticated graphics controller allowing the use of 'windows' which may be scrolled, overlapped, etc. However, the limited space available on a single display does not allow the global visual assessment of the current state of an information handling task, particularly in the case of complex ones involving the simultaneous availability of a number of 'documents' or windows.

The use of a single display is convenient in a real workstation environment, and reactions to the recent introduction of these approaches in real office environments have been favourable.

2.4 Spatiality based systems in office applications

In attempting to apply spatiality to an office environment, it would be profitable to briefly examine some of the conventional approaches used in office organization and information retrieval. In particular, we are interested in the activities of the professional worker. He is frequently characterized by a job function which is not always well defined. Typically, he will need to have efficient access to large amounts of information, to allow its effective use in reaching decisions and in producing more information. His responsibilities are fuzzily defined, and he will require access to an efficient communication medium to hold discussions and exchange ideas. He will have little or no computer orientation nor any desire to be trained, and may not be skilled in information organization.

Another example of potential users would be the class of workers who need to be kept up to date in a particular field, such as engineers, directors of research, research students, etc. The conventional approach to this need has been the browsing of a 'what's new' section in a library. The objective is simple: to browse quickly and efficiently through journals, books and abstracts. The human being is well equipped instinctively to browse and libraries on the whole cater for this need. There exist however no applications of available technology which cater for this requirement.

The commodity crucial to professional office workers is information. It includes printed text, diagrams, speech, etc. It is however the large volume of information relevant to the professional's task that requires the use of computer assistance. Taking the example quoted in [SPEN82], we see that a typical professional would require access to upwards of a million pages.

These requirements are in contrast to those required, say, by a secretary or a clerk whose activities are normally restricted to handling information generated by others, is not normally required to exercise a great deal of initiative, is expected to be familiar with basic office automation equipment such as word processors, and is often capable of the highly efficient organization of information.

An office worker requiring access to an item of information (e.g. a book or a journal), would rarely specify it in a purely symbolic manner (i.e. author(s), title, ISBN, etc). In order to retrieve an item of information in a conventional office, one would first locate the book or file containing that item. In doing so, people rely upon memory of where the item was last located as well as spatial and other cues and symbolic or textual labels. This memory will be enhanced if the items can be seen, for it is therefore necessary only to follow the recall of the approximate location, by a rapid browsing or scanning action, in order to locate. Where items are continuously visible, memory will be continually enhanced, by the constant awareness of spatial location.

As a complement to the finite capacity of their memory processes, humans have therefore developed an efficient visual recognition capability, this being manifested in the extensive use of browsing types of actions in interacting with their environment. Office workers, in addition, feel the need to be reminded of urgent or important tasks, and this is most commonly done by spatially coding the location of items which require their attention (e.g. the pile of papers to the left, or the in-tray) [MALO83].

An office worker, by organizing his work environment to his needs, attempts to optimally match the facilities offered by his office to his most efficient information processing capabilities. It is important not to underestimate the importance of maintaining the user's unconscious awareness of a spatially organized, browsing oriented environment, and his confidence in an efficient browse-recognize-select action. It follows, therefore, that the highly efficient spatial memory and browsing actions should be exploited in information systems design.

With information handling it is useful to distinguish between two essentially different tasks, as suggested in [SPEN82]: "One is the specification of an item of information that is of interest, the other is the retrieval of that item from wherever it may be located. By contrast with information retrieval, which is already the subject of extensive research, and where efficiency is largely dictated by hardware and software considerations, the human-computer interaction associated with information specification has received little attention".

It is a fact that automated assistance in any form is rarely available in conventional offices, the failings of conventional computerised information systems being partly responsible for this situation. There are a number of complaints associated with the use of a conventional database access system, and VDU's in general, by professional workers. We may approximately categorize these as follows:

1) Foremost is the problem of the need to train individuals in the symbolic, non-intuitive command dialogue necessary. Training times of the order of 5hrs to learn a command dialogue and the need for constant use in order to refresh learned knowledge is not acceptable by most professionals.
2) An intrinsic problem, associated with conventional VDU's is their limited display area, effectively performing the function of a window onto the available data. They thus tend to blinker

the user from the extent or nature of the data, depriving him of an awareness of its extent and contents, and do not offer means for comfortable, instinctive interaction. The windowing problem [SPEN82] has been compared to: "attempting to peruse a newspaper through a 2-column-inch sized keyhole, ('How long is this article ?', 'Where is the crossword ?', 'What other headlines are there ?') or of attempting to locate a needed diagram or equation within a microfiched article".

3) An additional and equally important factor for the slow acceptance of office automation, though one which will not concern us here, being already the subject of extensive research, has been the large number of physiological complaints associated with the use of VDU's, such as eye fatigue. This would indicate the need to take into account user physiology as well as psychology of interaction in the design of office automation equipment.

Some of the spatiality based approaches mentioned above attempt to offer some facilities necessary for a more instinctive, interactive environment. Their common feature, is the spatially oriented database provided, allowing items to be referenced in terms of their visual attributes, reducing training times (item 1 above), and making them more comfortable to use.

A number of additional features would be necessary however in applying such techniques to office information systems. An important office function is to browse, or skim through a list of items in order to quickly assess their urgency, importance, and significance. Such would be the case for the professional who wants to examine his in-tray for the current day's items, quickly and efficiently, allowing him to rank items in order of relative urgency. With conventional displays, some items will be necessarily masked, and the 'window' would have to move to cover all items thereby losing the global perspective of the data space (item 2 above). A 'spreading out' function is therefore required that would allow this assessment to take place by displaying the contents of an option simultaneously and in a meaningful form. In addition, we would need (as in a real office),to simultaneously keep track of all other options available so that we may return to, or work with them simultaneously (an example in a real office would be to answer a letter referring to perhaps other letters and technical material). In single screen approaches, display area is a commodity which needs to be conserved, thereby limiting the amount of cues that can be displayed. Normally, items are allowed to overlap one another, their identities preserved mainly in the form of textual labels, and can be 'pulled' to cover a larger part of the working area, thereby however obstructing the view of the 'context' of other items. Cost considerations and the intrusion to the office dictate against the large screen and multiple screen approaches which can partly overcome this limitation.

2.5 The Bifocal Solution

This recent approach to database interaction has been developed as an answer to the main criticisms raised against other spatiality based approaches [SPEN82, APPE82]. Some of these are related to the way with which professionals interact with data in their office, while others have been dictated by a number of human factors considerations. This new technique for information display and interaction, has been suggested as part of a complete office of the professional [SPEN82]; it retains the concept of spatiality but incorporates a number of new features as will be described later.

By realizing that the first level representation of a data space is usually static, a non-computer assisted presentation of this first data level is proposed; in the particular example quoted, the space available on a user's office wall. Options from this first level display are chosen by simple pointing actions, while techniques for dynamically changing its contents are possible.

One of the problems with current graphics assisted systems, as mentioned above, has been the limited display area available in conventional graphics screens. The use of multiple screens is clearly expensive, and a single conventional display cannot always contain all of the information desired at any one time. The Bifocal technique proposes a unique solution to this problem.

The proposed approach is to focus on the area of interest rather than use a window, thereby combining an overview of the entire data space with a detailed view of a small portion. This approach, by dividing a VDU screen into three separate viewports achieves two objectives. First, it permits two or three items to be displayed in the central region with sufficient detail to be read in full. Second, and at the same time, two outer 'demagnified' regions retain adequate detail of the remaining items, in the form of colour, shape, tags, initial letters, etc, to indicate important attributes such as number, size, urgency, nature and origin. The data space is still considered to be continuous, and can be conceptualized as a long horizontal strip of data space. A representation of the concept is shown in Figure 2.4 while examples of the Bifocal display applied to different items is shown in Figure 2.5. By using a touch screen, any item in either of the two regions can be examined in detail by touching that item and 'pulling' it into the central viewport. By this action, the whole strip representing the data space is moved across the screen, preserving the -cognitevely useful- spatial relationships between individual items, and always retaining an overall view of the entire space. Having brought an item (an icon or an option) into the central viewport, we may interact with it in detail.

The effectiveness of the representations displayed in the outer demagnified regions can be enhanced by the following method: As mentioned in the previous chapter, humans perceive visual images or their meaningful representations more effectively than text. For this reason, graphic attributes can be used to encode the items in the outer regions. These include colour, shape, size, tags, etc. The use of alphanumerics is restricted to a few letters. The use of such attributes should however be exercised with caution, as one of the conclusions reached from systems such as SDMS, is that too many cues can be confusing, and lead to cluttered rather than informative displays. The representation therefore of items in the outer regions is not merely a demagnified version of their central region representations, but a representation more appropriate to the lower resolution of these regions, retaining clues as to their meaning and nature.

By allocating the first levels of the data space to a largely static display, (for example an office wall), the amount of information which needs to be displayed has been considerably reduced. At the same time, the user is aware of the extent of the data space available with the Bifocal display, and of the items contained, while interacting with an item or option in detail.

In traversing a tree type data structure, the Bifocal technique retains the identifying features of the parent nodes traversed. This is accomplished by compressing a representative description of nodes, in the edges of the central region, as each node is intersected. The user can therefore maintain a perspective view of his path through the data structure, while allowing for a global assessment of his current state to be continuously available. This feature is included to cater for the need of providing a pictorial, consistent representation of the data model as has been suggested in [MCGE76].

Applications of the Bifocal display can make use of its ability to 'spread out' the contents of an option, allowing for the leisurely examination of offered items. Such would be the case for example, in browsing a library section of 'recent additions'. In combination with symbolic searches on precise target word(s), the Bifocal display would be ideal in containing the results of such a search (normally producing a lot of irrelevant material) such that they could be quickly examined and the desired items selected.

Variations to the type of movement in the display are possible. For the simple 'spreading out' operations, such as examining an in-tray's contents, a simple scrolling action and two levels of detail within the Bifocal screens are probably adequate. For the examination of, say, a scientific journal we would need to scan through years, then perhaps months, and finally articles, requiring therefore three levels of detail (Figure 2.5a). Examination of a journal or article within the central viewport may be facilitated with the provision of a simulated page turning, or zooming action, whereas in the case of a diary representation, a combination of horizontal and vertical movements within the display are desirable [SPEN82, APPE82]; horizontal movements would be used for large jumps of the order of weeks, and vertical movements for, say, movement within the current week.

The techniques for creating, transmitting and annotating items or documents are similar to those of other graphics based approaches. The range of applications need not be restricted to offices but have applications in library search, military command and control, and in public and private viewdata systems. All normal information processing functions are possible but, for the system described, details have been given only of the process of specifying the item required in a search operation. Extensions to include precise database query languages can be implemented, the approaches by which information is retrieved from a database being a complementary problem common with that of other experimental approaches, requiring considerable research.

In summary, the Bifocal display presents a solution to the difficult problem of windowing, by providing an awareness of the entire context of an item of information while simultaneously providing a detailed view of that item. The main advantage proposed with this new display technique is that it is possible, using one display unit, to provide the user with many simultaneous viewpoints and cues to allow for the easy assessment of his location in relation to other items in large data spaces, while maintaining the possibility of detailed interaction with an item. It makes use of natural human actions such as pointing and touching which are instinctive and comfortable to all humans and demand little specialized knowledge. The representation of items iconically on a wall provides an inexpensive, extensive first level of interaction which is always available, allowing the user to become familiar with item location and spatial relationships. In addition, users are not required to have any specialized skills, using a number of instinctive actions such as pointing and touching. The Bifocal technique, in contrast with the approach adopted by single display-based systems, does not attempt to emulate the appearance of a conventional desktop. All normal information processing functions are still available, but it is believed that as the technology is available, we should not necessarily be constrained by previous decisions based on past technological performance constraints. In fact, we should adapt the new technological capabilities (or a reasonable estimate of them), to functions which match and exploit our highly developed and thereby efficient, instinctive perceptual capabilities; at present the reverse is true, i.e. humans have been constrained to readapt their capabilities to the limited, symbolic range of interaction available with current technology.

2.5.1 Design guidelines

Here, we will attempt to present a number of design guidelines for the realization of a Bifocal display [APPE82] incorporating the features described above. These features are summarized as follows:

a) the ability to present data in different viewports at different levels of magnification
b) an easy scrolling capability allowing for the easy movement of items across the display
c) a zooming and/or windowing capability on items in the central region
d) a compression of node descriptors on the edges of the central area.

These functions must be executed efficiently (i.e. operation should be transparent to users), and independently of ancillary functions. Such ancillary functions include:

a) management of the virtual data space
b) management of other peripherals
c) the interaction dialogue

Such requirements initially point towards the provision of a separate graphics processor implementing the as above functions insulating the main processor from the management of the possibly complex Bifocal display.

It is noted that the items present in the outer regions are presented at a lower level of 'magnification' in the form of a representational description appropriate to the lower level of detail which nevertheless retains salient features of items such as colour, shape, and a textual tag, indicating the nature of the item. It is therefore obvious that at least two levels of description for each item are required. As an item is chosen, and examined, more detail can be added.

The implementation of the Bifocal display can be achieved by providing three separate viewports, each with an individually programmed origin which can be moved continuously through display memory, as in Figure 2.6. This memory must be larger than one screen page, but a virtual memory management system can be used to provide for any size of data-space. It would seem logical to suggest that the required functions must be implemented by specialized display hardware and this is in fact, the approach upon which the design principles to be described are based.

For the majority of data spaces, the detail provided at level II will be sufficient (e.g. reading items of an in-tray). To obtain further detail (where available) yet still retain the advantages of the Bifocal display, a 'zoom' action can be carried out in the central region, replacing the level II image with a more detailed representation (level III). Further zooms can be executed if desirable and appropriate. The image at level I remains in the outer viewports throughout, ensuring that the entire data space is always visible.

To move the data space through the viewports, different horizontal scrolling modes may be appropriate in different circumstances. For example, with an essentially continuous data-space, a smooth continuous scrolling action allows any item to be drawn into the central region. With a paginated data space, such as the periodical mentioned above, a simulated page turning action, would be more appropriate. Here, the text does not move, but a page is simply peeled back by the scrolling action of a finger, to reveal the next page underneath. A third mode is possible, that of a combined vertical and horizontal movement, where for example horizontal movement would be used for large jumps, and vertical ones for a vernier type of movement as with the calendar mentioned above.

2.5.1.1 Horizontal Scrolling

We shall begin our design description with the continuous horizontal scrolling action [APPE82]. For this mode, the level II description can be considered as a continuous strip without page boundaries. This movement, fundamental to the concept of a Bifocal display, involves dividing the screen into three boundaries. A convenient division, and one with benefits in a binary environment, is to have the widths of these viewports in the ratio 1:2:1. Each viewport is mapped into virtual display memory by adding the contents of a corrresponding window origin register to the horizontal screen coordinate (Figure 2.6). Thus, each viewport can display any windowed portion of virtual display memory.

Virtual display memory must be large enough to contain both the level I (low detail) and the entire level II (high detail) images of the data-space simultaneously. In practice, physical display memory will normally be smaller than virtual display memory, so a virtual memory management system in addition to the bifocal management system is required to map virtual space into physical space. In a simple implementation with a limited data space, the virtual memory management system can be dispensed with, as a full data space can be kept in physical memory continuously. For larger spaces, page faults can be anticipated by monitoring the scrolling movement, and requesting the relevant data from disk. To prevent 'freeze-outs' overlapped disk I/O is necessary.

Horizontal scrolling can be achieved by a simple adjustment of the contents of the three window origin registers. The rate of change however will be larger for the central viewport as the magnification provided by that viewport is larger. The ratio of velocities of the level I and II images are as the ratio of the widths of the data space at the two levels.

Defining a display memory page as the contents of the central viewport, it is observed (Figure 2.7) that physical display memory should have a capacity of at least 3 pages of the level II image. This would allow for direct scrolling across either side of the central image, and thereby provide time for predicting page faults and initiating data transfers from disk. Another requirement for a Bifocal display is that the entire extent of the data space be visible from the level I viewports. This would require half a display memory page, plus a blank half page on either side of the level I image, to fill in the spaces when the data space is scrolled to its extremes. As the level I image will be used frequently, and does not consume large amounts of memory it should be permanently resident in physical display memory.

2.5.1.2 Combined Horizontal and Vertical Scrolling

For a combination of continuous horizontal and vertical scrolling, used to support the same type of movement but with differing increments, we need only to extend the one dimensional strip of memory to a 2-dimensional arrangement as in Figure 2.8. Horizontal motion is exactly as described in the previous section. Vertical movement however in this space is also possible. If, for example, the central window were to move diagonally, then we are faced with transforming the memory pages being intersected in our conceptual 2-dimensional memory to the linearly organized virtual display memory. This can be done by, in effect, splitting the window as in Figure 2.8. This operation is trivial in a binary environment as simple remainder/quotient operations are all that is necessary [APPE82].

Virtual memory management is as for the previous case, but a greater investment in physical display memory is now necessary as four pages may be required for satisfactory buffering of the scrolling action, as can be seen from Figure 2.8.

2.5.1.3 Horizontal Page Turning

The page turning action is used to examine an item in detail, by simulating the way a book is read. The current page on the central screen is peeled back, with no movement of text, to reveal a page underneath, as in Figure 2.9. A simple method of executing this action is to think of an additional peeling page viewport on the data. As the page is peeled back, we need to compare the moving page boundary with the X scan coordinate, and use the result to select between the new and previous viewports. Again, when this facility is requested by the user, it is desirable to bypass the virtual memory management system and provide a temporary dedicated physical display page(s) to avoid undesirable delays. For further page peeling we can use the (old) central region viewport as the new peeling page viewport, if physical display memory is at a premium, and so on.

2.5.1.4 Virtual Memory Management

The virtual memory management system can be implemented in a conventional manner, with available memory management circuitry. There are however a number of factors suggesting a simpler solution [APPE82]. Memory management is required for the Bifocal display processor, not the CPU. This implies that some of the virtual memory management can be allocated to the CPU rather than specialized hardware. Conventional memory management hardware cannot anticipate faults by monitoring the direction of the scrolling movement, a necessary feature for smooth scrolling, and unique in graphics-based database systems. In addition, graphics images are relocatable; that is, the actual physical memory location does not affect image characteristics. The above considerations simplify the task of virtual memory management to the extent where all that is required is that there exists a strip of memory apparently extending continuously in either direction. This is obvious from Figure 2.10; as the displayed image approaches page 3, and is about to extend to the empty space to the right of page 3, we require the following: the central processor to request page 4, to be read from disk and replace page 0, while the Bifocal display generating unit, as soon as page 4 has been transferred, will allow the displayed image to be wrapped around from page 3 to page 4 (previously 0). In effect, we are requesting a modulo {4xpage width} operation on the horizontal screen coordinate. The process can be inverted for a scroll in the opposite direction.

It is obvious therefore that continuity can easily be maintained, and large data spaces thereby accommodated, in smaller machines with limited memory.

2.5.1.5 Data node compression

The feature of compressing data nodes on the outer edges of the viewport is conceptually not different from the viewporting operations described above. It is only necessary to define an extra pair of viewports on either side of the central one, in order to accommodate the new viewports, and so on for every new level. These new viewports will point to appropriate virtual memory locations, these pointers again changing according to the scrolling action.

Extra complication is however introduced by the need to introduce these viewport management facilities. This process will also, eventually, be limited by the size and effective resolution of the physical display device.

2.5.2 A prototype implementation

As a first step towards a practical implementation, the design considerations described in the previous sections were applied to the display generating circuitry of an Apple II microcomputer. The criteria for choosing such a machine were ease of modification and low cost, rather than any inherent attributes favourable to the support of a Bifocal display.

For this prototype implementation it was decided to implement the continuous horizontal scrolling and page turning modes of operation, leaving vertical scrolling as a later option. The display memory of the microcomputer chosen occupies 8k bytes of memory, and by splitting the display into three viewports, in the ratio 1:2:1, we set the size of the central viewport display to 4k bytes. Four such pages have been allocated to the central viewport level II image, and two pages for the level I image. The entire data space is kept on disk, and can be transferred to display memory as desired, in accordance with the direction of movement through the data space.

The implementation is rather complicated, mainly due to the difficulty of interfacing to existing display generating hardware, to critical timing problems, to colour display peculiarities, and to the need to keep dynamic memory refreshed during the display cycle. A more detailed description of the design approach taken is available in Appendix 1. Multiplexing of the Bifocal display is achieved by disabling the normal video generation circuitry during the display cycle, and using the new memory addresses generated. Window origin registers and the peeling page register are accessed via an I/O port. The completed Bifocal display generating unit, and its interface to the Apple II is shown operating in Figure 2.11. Scrolling is controlled with a joystick, and can be easily replaced by other devices such as a touch screen, as the microprocessor's CPU, programmed in a fairly high-level language (interpreted BASIC) is responsible for the scrolling action, thereby adding flexibility to the system. The microprocessor is also responsible for managing the simple virtual memory system proposed.

The prototype implementation was used to display the contents of a hypothetical in-tray containing various items that are available for examination, by scrolling the data space underneath the central viewport. The system implemented performed as desired, providing the horizontal scrolling and page turning modes. Scrolling speed was effectively unlimited, and had to be artificially constrained through waiting loops. Problems encountered included colour display inconsistencies particular to the Apple II, and 'freeze-outs', the latter as a consequence of the fact that the disk operating system provided does not permit overlapped I/O operations. This is not a serious problem even at this stage, as we may easily isolate the microprocessor from the management of the scrolling action. Such types of display, however, consume large amounts of memory and therefore a larger physical memory would be desirable.

It was subsequently discovered that the CPU is powerful enough to execute a Bifocal display purely through the use of graphics software operations, by moving, for example, object descriptions across unmodified display memory in a coordinated fashion, detecting intersections with the central viewport, and substituting the magnified object description. The basic requirement for such a process would be to move items across the screen smoothly. At level I, we might as an example, wish to move a 50 by 50 pixel square across the screen. In order to achieve smooth animation we would be required to effect this in around 100mS. Assuming 4bits/pixel then the required (display) memory to memory transfer rate is about 100kbits/sec. A higher rate is required for smoothly scrolling the level II and higher images. It is believed that such memory to memory transfer rates ar well within the capabilities of current 16-32 bit microprocessor designs (clocked at about 10Mhz), while allowing sufficient spare capacity to manage the overhead of a Bifocal display; for example, detecting viewport boundaries, deciding length and type of movement, managing the virtual memory and I/O devices, etc.

Such an approach would relax the memory constraint and add flexibility to the system but conversely, may overload the CPU entirely with the management of the Bifocal display. Recent microprocessor designs, however, with more powerful CPU's and larger memories can easily overcome the latter problem, and it is in this way that software-only driven types of the Bifocal display are now contemplated.

2.5 A Relative Assessment and Conclusions

Primarily as a consequence of their recent introduction and continuing development, graphics assisted database systems have not been subjected to rigorous performance testing. The specialization involved in each system (for example, ship's databases in the CCA system, and office documents in the STAR and LISA approaches), make comparisons on an objective basis meaningless. Nevertheless in this section, by examining the main features of current graphics-assisted database interaction systems, a first-order relative assessment is attempted.

The criteria typically used for evaluating the performance of conventional database query systems have centered mainly on the ease of learning a particular command language, and the relative success with which accurate queries are formulated by newly taught subjects. The latter criterion's validity is questionable in the present context, since simple actions are all that is necessary for interaction with graphic data spaces, errors being an integral part of the navigational process, rather than disasters in an otherwise calm environment.

Training times, or the time taken to familiarize a user with the 'dialogue', have been quoted as being of the order of minutes for SDMS, similar results being claimed for the other graphics-based approaches, all based on initial user reactions.

Performance criteria such as training time, and percentage of correct query formulation used conventionally are obviously not valid in such comparisons, and a new set of criteria is necessary, based on relative assessments such as interaction efficiency, cost, ease of use, etc, regarding features which are commonly available in all approaches being evaluated.

Navigational cues

One such performance criterion is the ability to present to the user adequate navigational cues such that he may easily assess his relative position in the data space. This is an important factor in graphics-assisted systems, as it is one of the primary means through which spatiality is being exploited. No significant differences exist between current graphics based systems in terms of the amount of cues that can be presented, but it is obvious that in conventional single display implementations, the amount of cues is limited by display space and the need to avoid excessive 'cluttering'. Conversely, in multi-screen systems, the user is forced to shift his gaze across a number of displays with consequent tiredness and a possible loss of fixation. The bifocal technique presents a new approach to this confrontation by exploiting the best features of multi-screen and single screen approaches in one unit. The cost associated with supporting multiple displays is also obviously reduced.

Applications

In terms of the diversity of applications possible with each system there are no obvious restrictions, apart perhaps from cost limitations in using the multiple screen approach. The type of 'navigation' or motion allowed in each approach differs. In SDMS, the movement is similar to a 'flight' through the 2¼D data space, while in document-oriented single displays the motion is restricted to the items being manipulated. In the CCA and Bifocal approaches, the movement simulates a 'scrolling' of a data space under the user's control.

Browsing Support

An important facility offered by the Bifocal approach is the ability to 'spread out' the contents of an icon such that the user is aware of the content, type, number and importance of items at a glance, while retaining the capability of quickly examining and/or interacting with an item in detail. The objective of such a facility is to cater for the human need for browsing, not available with current symbolic oriented approaches which are geared towards precise definitions of the items being sought. The availability of such a facility makes the Bifocal display ideally suited to library types of search. Other graphics-based approaches have not yet offered such a convenient browsing feature, mainly due to the rigid nature of their display design.

Command Dialogue

A degree of adaptation to the command dialogue has been suggested [BROW82, APPE83] as being useful for experienced users. The system may recognize and store paths which a user commonly traverses in a dialogue, and can initiate default paths to avoid experienced users negotiating simple selections repeatedly. A useful feature in current systems would be the facility of being able to interact with a conventional DBMS, the CCA system being an example of a combination of graphics assisted and conventional database access techniques. This adaptability and incorporation of precise query techniques may be introduced in all of the approaches suggested, there being no obvious differences between them regarding the degree to which such enhancements to their dialogue structure are possible. Initial implementations are limited in terms of the offered facilities, but in this assessment, it is the potential for efficient access to information which is important, and in this respect the above approaches rate approximately equally.

Zooming (e.g. level change) through these data structures and detailed interaction with the offered items again differs in all approaches, the implementation and features offered depending on the application. As yet there exist no definitive results as to the relative preference by users of each approach.

Cost

Cost considerations, though not important in, for example, military applications, are likely to be a major deciding factor in lower budget implementations, such as office information systems. Mainly as a consequence of their multiple screen format, and in spite of the falling cost of graphics hardware, the SDMS and CCA approaches are thought to have an economic disadvantage as compared to single display approaches. In office environments the use of multiple or large screens is currently unacceptable. Single display, document handling oriented techniques in most implementations to date, have required the support of a minicomputer and specialized display hardware to execute the real time memory mapping operations necessary, although the increasing power of the latest microprocessors is reducing this constraint.

By contrast, the prototype implementation of the Bifocal approach, has been implemented at minimal cost by modifying the existing display hardware of a widely available microcomputer [APPE82]. In addition, it can be implemented on larger machines supporting bit-mapped displays entirely through the use of software graphics operations.

The database support units necessary are thought to be roughly similar in all approaches in terms of specifications and cost, the major difference being that systems such as SDMS and CCA, as currently implemented, emphasize real time, short response delay applications, and are therefore unlikely to lend themselves to a more cost effective, shared local-area database type of approach, as would be necessary in an integrated office information system.

The Data Model

In designing a data model for compatibility with human factors considerations, it has been pointed out [MCGE76], that a data model should be elegant, simple and, preferably, available in a visible form. The Bifocal approach has an advantage in this respect, since the nodes traversed in the data structure are always available in a compressed pictorial form, at the edges of the central magnified region, allowing for easy visualization. This facility is potentially possible in other systems but to a lesser extent. For example, one of the problems mentioned in SDMS, is the fact that users tend to become 'lost' in the deepest levels of the tree structure, as in these levels there are few navigational cues available. It is worth noting that the importance of accurate mental model formation of dialogue structures has been appreciated by the designers of the STAR approach [BEWL83], and has led to conscious attempts at maintaining the consistency of the dialogue as perceived by the user.

This limited set of comparisons is clearly inadequate if positive comparative results and final opinions are desired. In spite of this, it is believed that a number of weaknesses have been identified in these first attempts at exploiting spatiality. They concentrate mainly on the cost of providing a sufficiently commodious interactive graphics environment, and in observing that even though spatiality is an extremely powerful medium with which to bridge the gap between a user and a computer, its use should be directed by the consideration of other known human factors. It should be noted that most of the existing implementations of the above suggestions (with the possible exception of the LISA approach), exist as prototype models and do not encompass the full extent of enhancements which would be possible and/or desirable. The comparisons made concentrate therefore on an estimate of the potential, rather than on available system capabilities.

We conclude that the Bifocal technique, by utilizing the concept of spatial information management, by implementing a number of the proposals which current thinking on the user's psychology of interaction indicates as desirable, and by utilizing the successful features of other recent attempts at exploiting spatiality, proposes a potentially advantageous approach.

It ensures that the user is always fully aware of the context of displayed information, and of the entire context of a data space, while simultaneously providing a simple pictorial view of the data structure. It is proposed that the above requirements can be accomplished in a single display unit. General design principles for such a display have been presented, and an example design described, indicating that the concept may be implemented inexpensively with current technology. These principles have been applied to a prototype implementation, demonstrating the viability of the technique. Trends in current technology indicate that flexible and cost effective software based implementations are feasible. The Bifocal approach can be simply interfaced with a conventional DBMS to provide an intermediate step between existing approaches and future directions in information systems design.

It is believed that displays of this type have the potential to greatly enhance the capacity for information display and interaction, particularly with regard to the construction of integrated information systems offering a complete range of information manipulation capabilities.

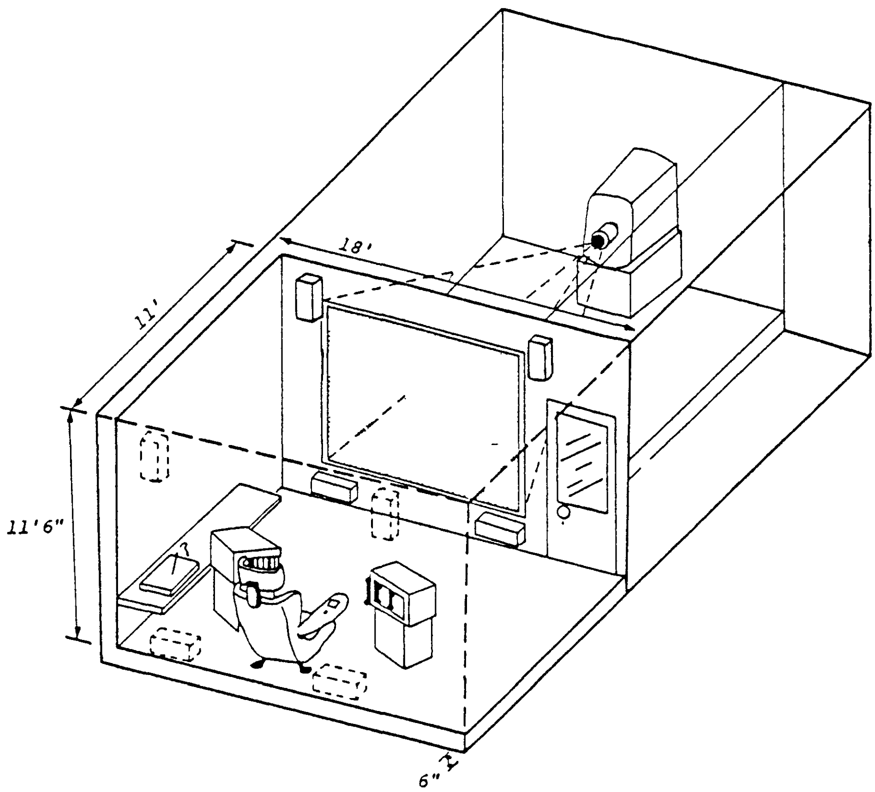


Figure 2.1 SDMS "media" room

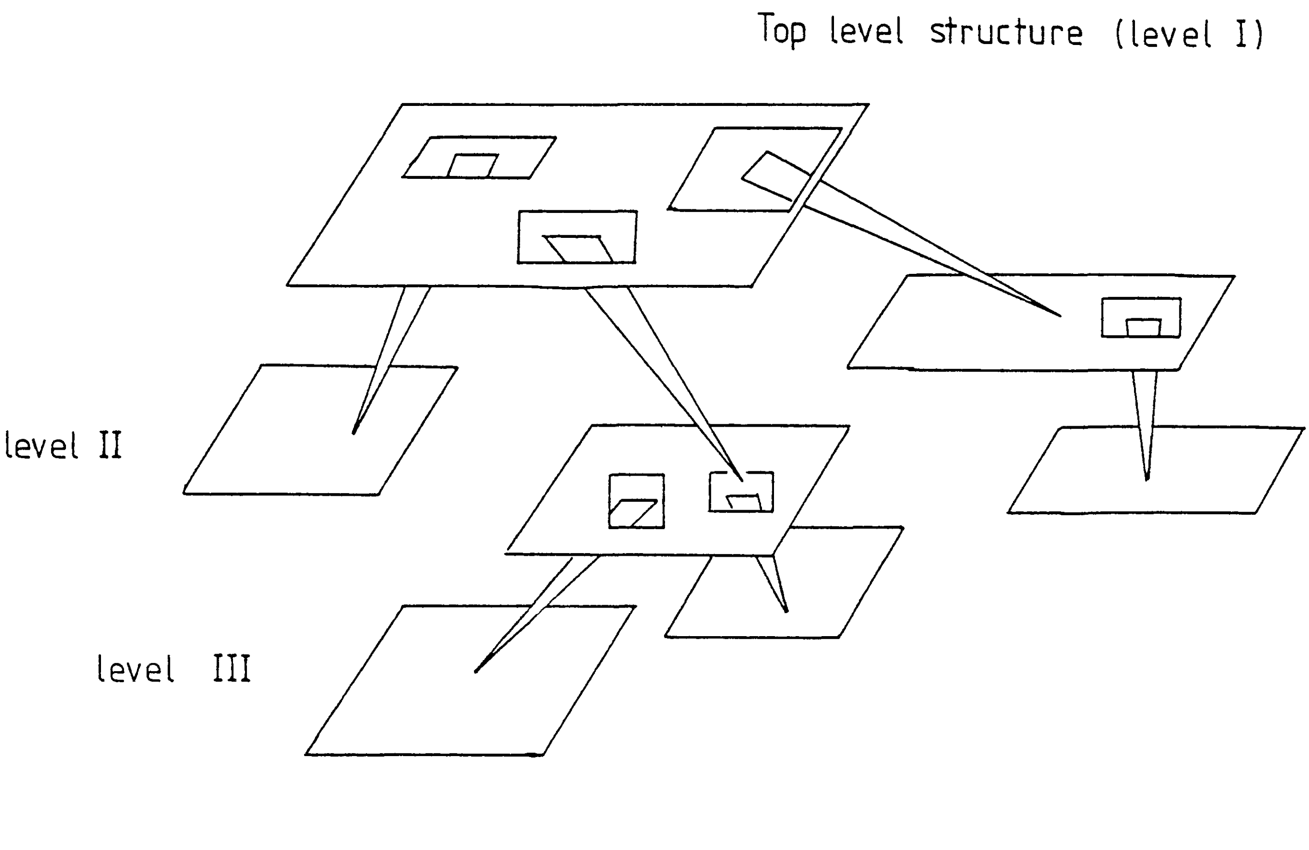


Figure 2.2 A laminar tree structure

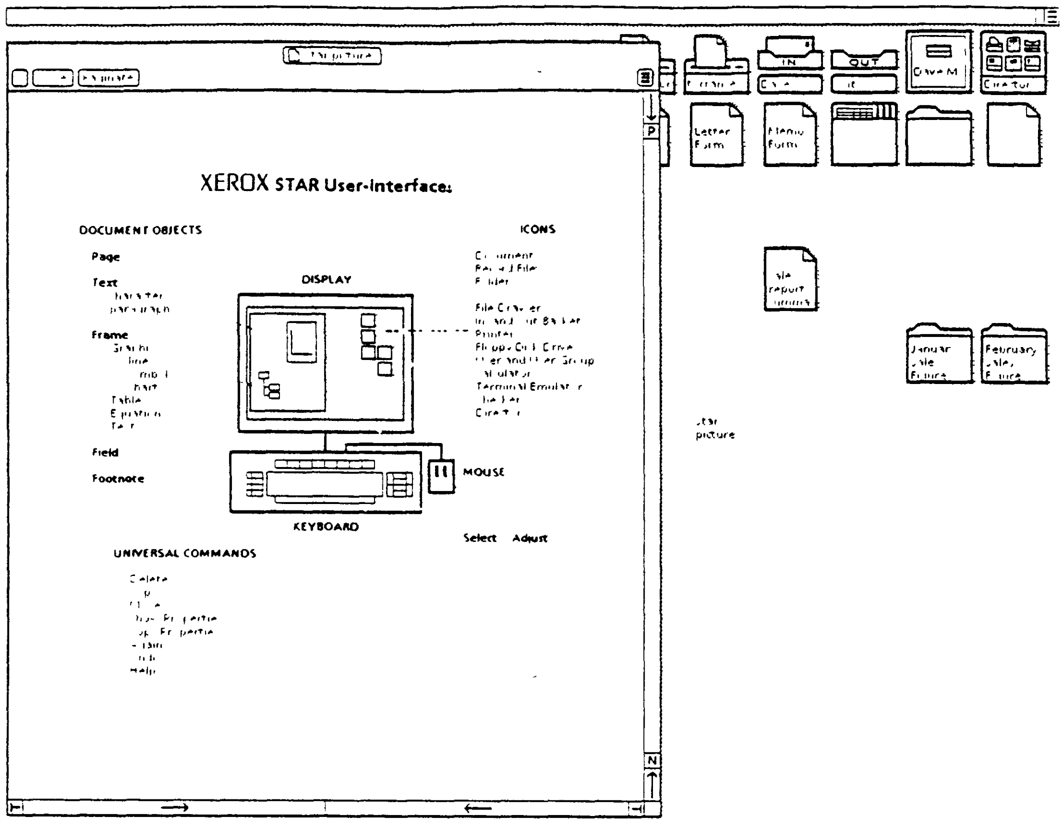


Figure 2.3 A Desktop as it appears on the Star screen

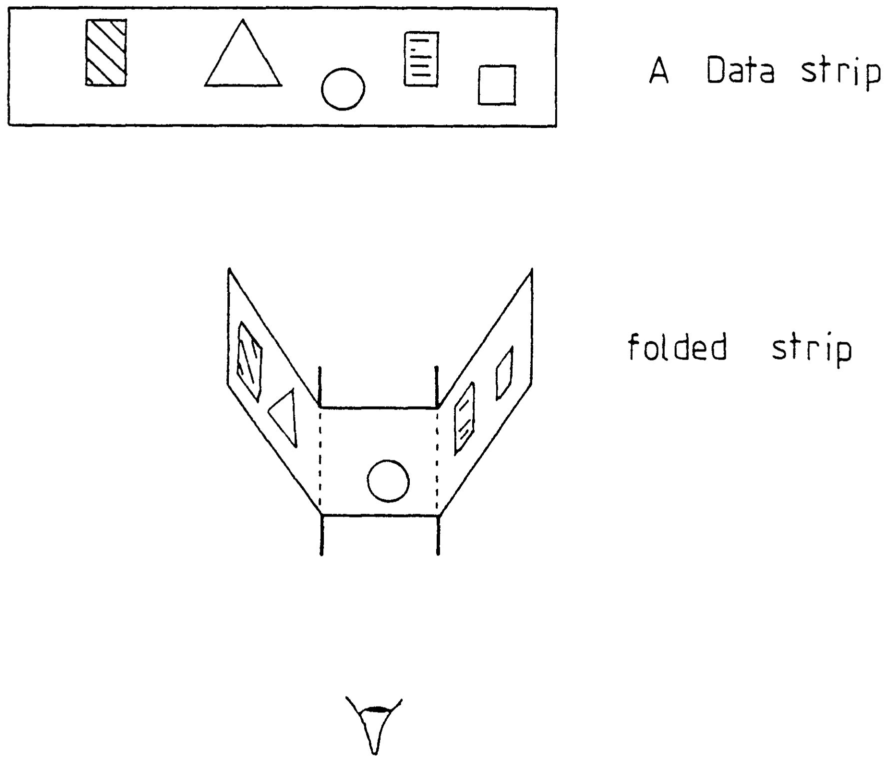
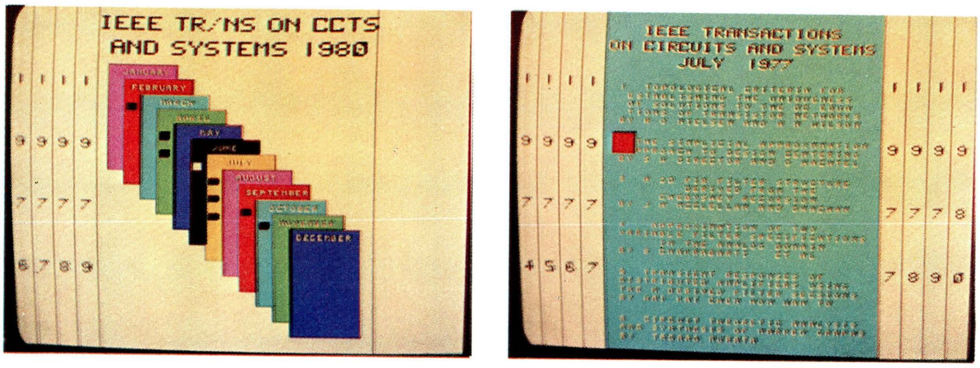
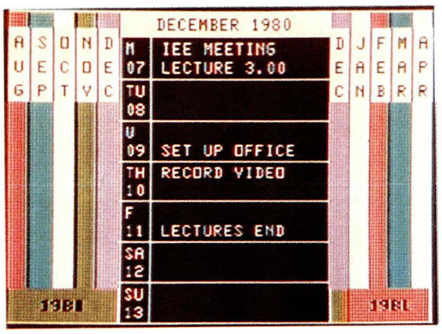


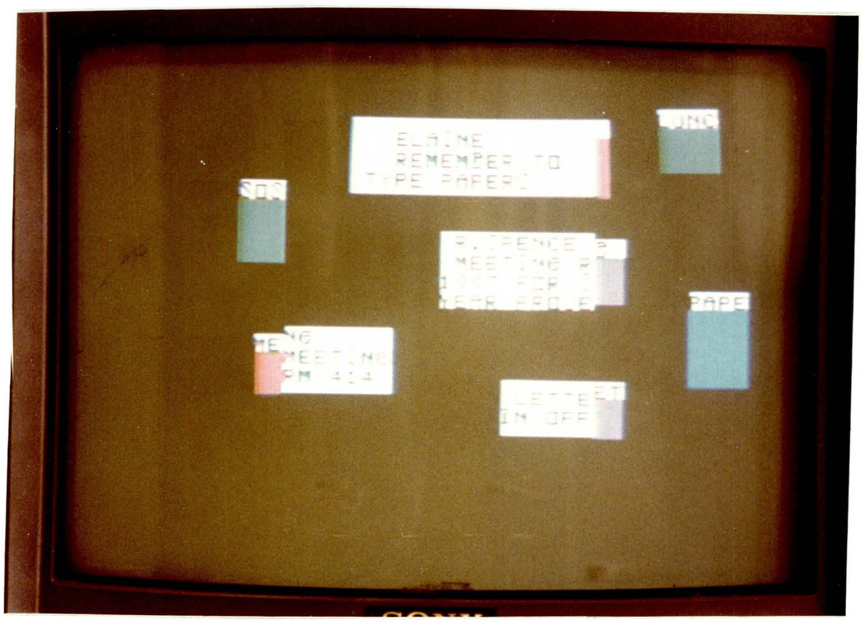
Figure 2.4 principle of the Bifocal display



(a) a journal

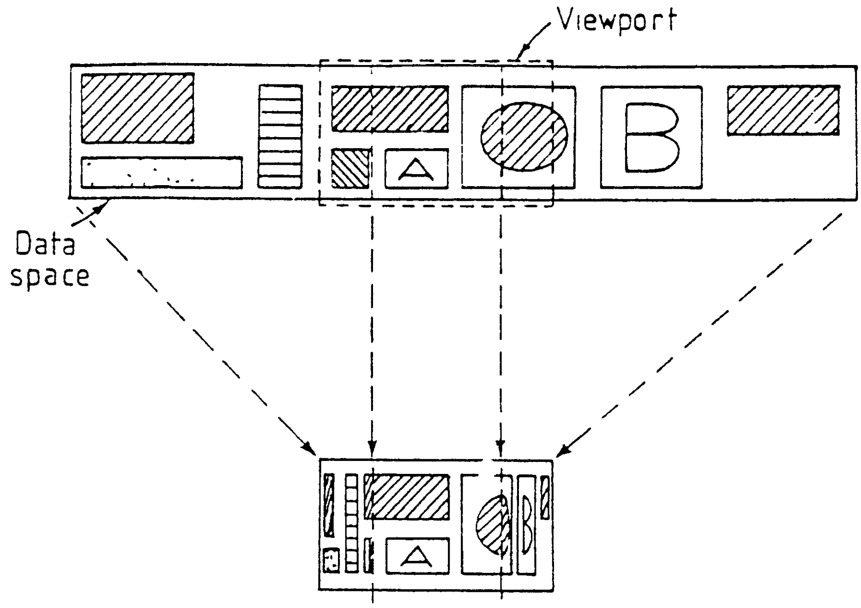


(b) Diary



(c) an in-tray

Figure 2.5 examples of the Bifocal display



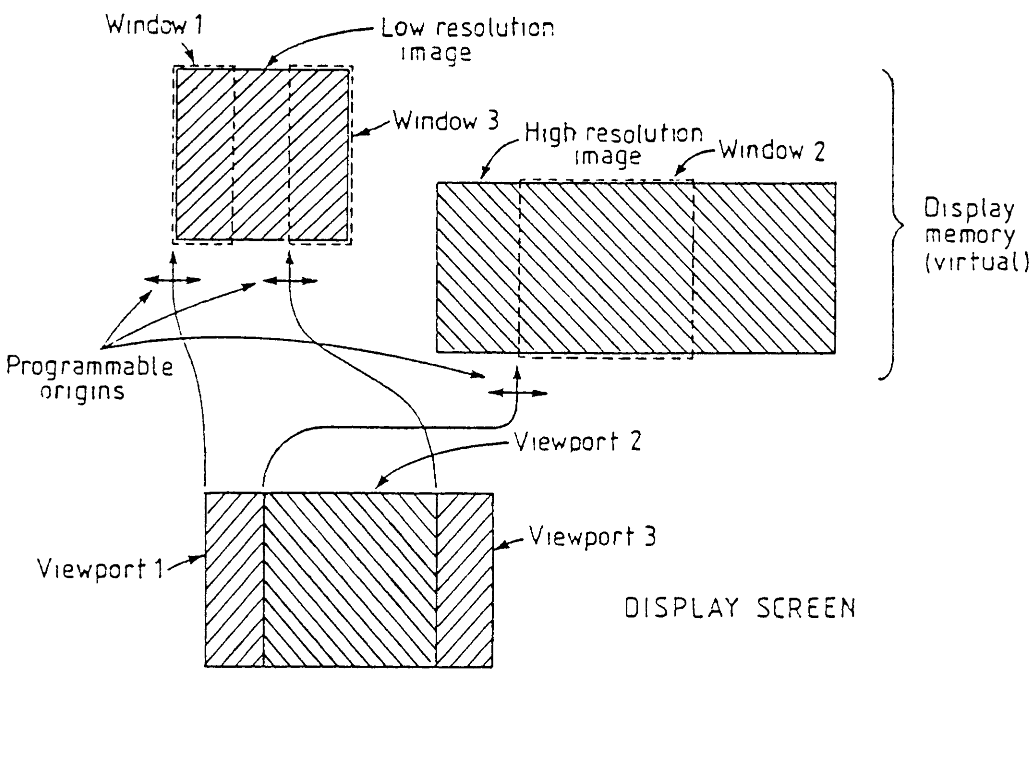


Figure 26 The Bifocal display mechanism

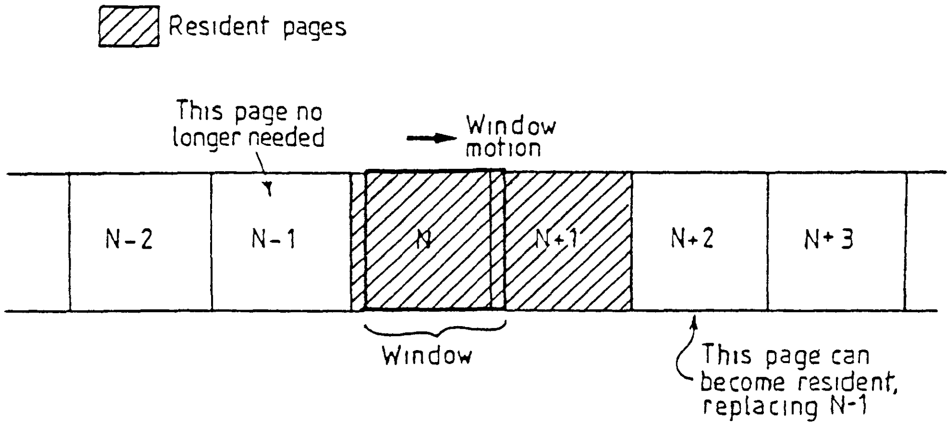


Figure 2.7 Physical Display memory capacity

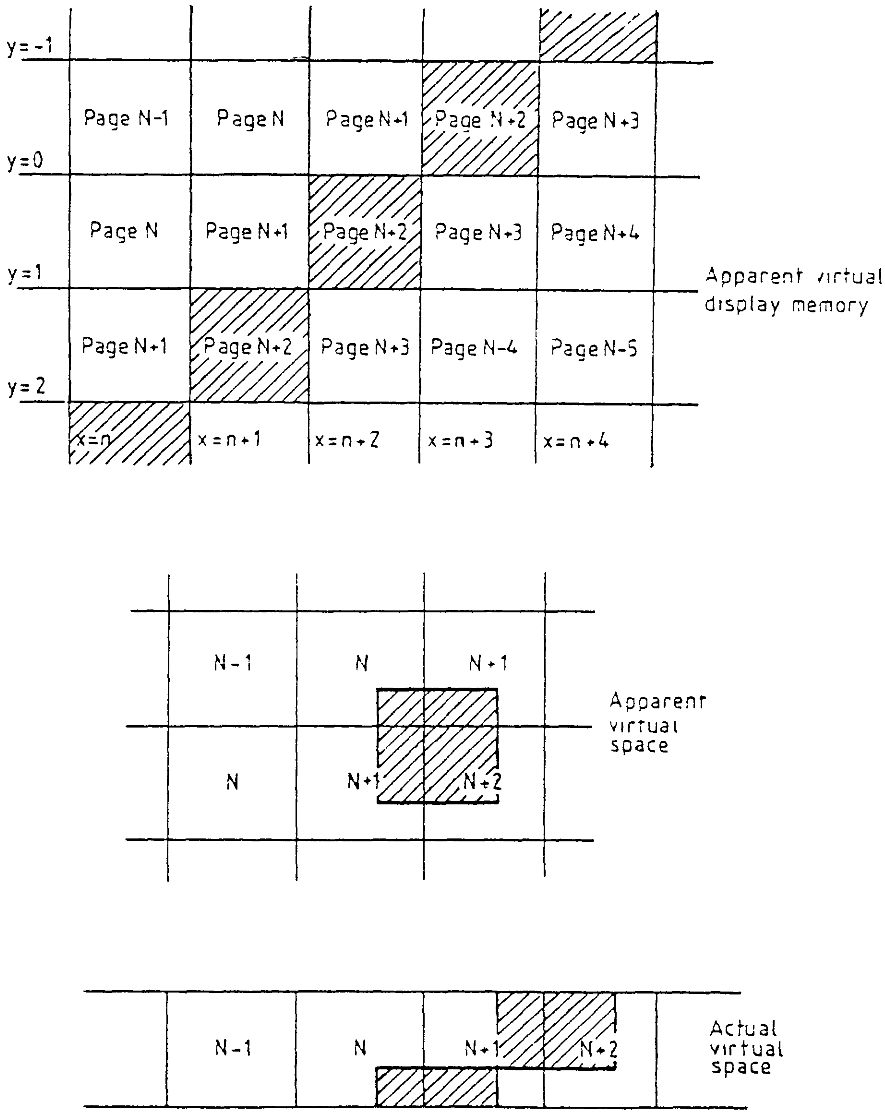


Figure 2.8 Horizontal and Vertical movement

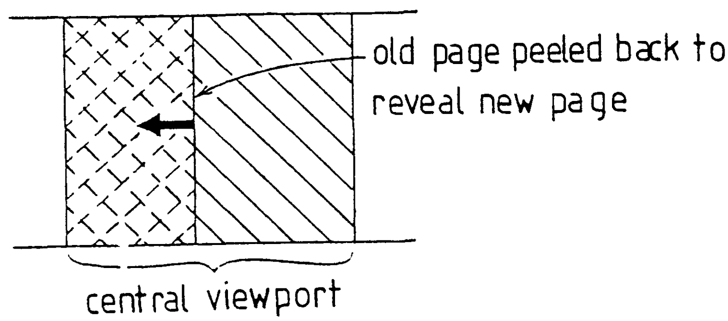


Figure 2 9 Page peeling mode

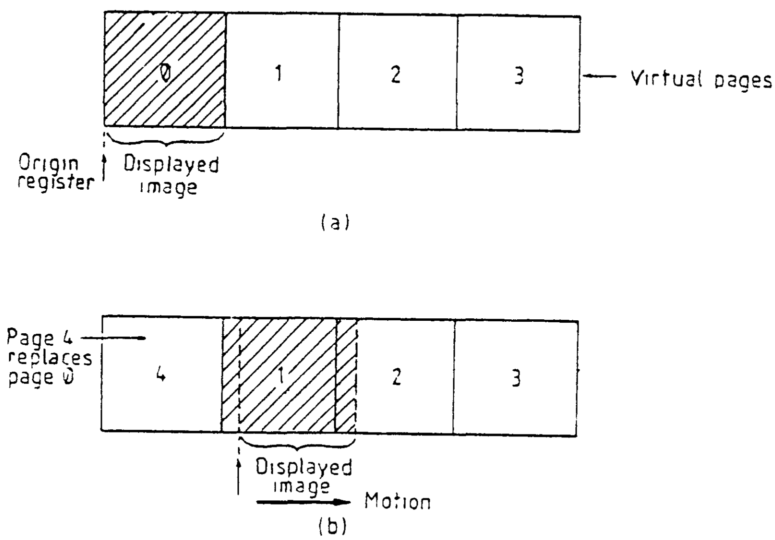


Figure 2 10 Virtual memory management

Figure 2.11 A Bifocal display implementation

CHAPTER 3

MODELING OF AN INTEGRATED, OFFICE INFORMATION SYSTEM

3.1 Introduction

In this chapter we shall investigate some aspects of the operation of an integrated, Local Area Information network, of which a typical example is a company's Office Automation system. We shall mainly be interested in the way in which component parts are interconnected, and the protocols under which they operate, in order to produce complete systems offering a wide range of facilities. The general form of such a network is represented in Figure 3.1. Its main features are a number of intelligent user workstations connected to a local area communications subnetwork, which also links data storage devices, controllers, printers, gateways to other networks, etc.

We examine such a Local Area Network (LAN) as the future implementation of intelligent workstations supporting graphics-based dialogues using extensive graphical representations and other high data content items (e.g. voice) is probable, as suggested in chapter 2. Such graphics-based man-machine interfaces however, demand new levels of performance from the supporting network [TZAV82, TZAV83A]. This is in part due to the requirement for an interactive dialogue characterised by short response times, and the large volumes of data associated with the images involved in graphics-based systems. The supporting network whose performance characteristics as viewed from an interacting user must satisfy these requirements, comprises the local area communications subnetwork, the database server units, controllers, operating protocols, etc.

3.2 System Description

Here, we will not attempt to concentrate on a specific implementation nor on an example system with a predeternined performance and topology. Instead, in view of the evolving nature of the design of such systems, we will consider an estimate of the characteristics and performance of network components based on current estimates of the direction such systems will follow in the near future.

With respect to the workstation, it is expected to support a man-machine interface of the form suggested in chapter 2. The extensive use of graphics in the presentation and interaction with information is therefore assumed. The workstation will also support a multitude of data types such as text and voice, and activities such as database search and electronic mail. It will provide facilities for storing data locally while also being able to access a larger remote data base. Facilities for the efficient communication and handling (i.e. coding/decoding), of these various data types are therefore necessary. As an example, coded/unencoded pages of A4 will require in the region of 20,000/200,000 bits per page, and an unencoded graphics image around 1Mbit. Voice messages will require from about 2400bits/sec to 64000bits/sec depending on the coding used. It will be assumed that the workstation data management protocol is intelligent enough to handle some of the data management activities such as forwarding file access requests and updates. The specific details of interaction however, other than the above, need not concern us further.

As systems will now be used by novice users as tools rather than as the hub of a task, adequate performance margins will be advisable. The typical users of an office information system are likely to be less patient than members of the professional computing community, since they will now be unaware of possible inherent system limits and will demand optimum performance at all times.

The communications subnetwork will be specifically designed for the local area environment, and based on one or a variant of the emerging successful topologies such as ETHERNET and RINGs. A disagreement exists currently as to the topology which will finally emerge -if ever- as the standard for the communications subnetwork. There are numerous trade-offs involved, each system having been developed for a particular location and application [CLAR78]. It would therefore be wise not to concern ourselves further with topology specific details if possible. This conclusion is strengthened by the emergence of technologies based on fiber optics with data rates approaching 100Mbit/s [RAWS78, IKED80, DAVI83], and small network access delays, allowing us in some cases, to neglect the effects of network access and transmission delays. However, although LA communication network technologies differ in concept, performance, reliability, and facilities offered, a common characteristic of the dominant examples to date has been the use of variants of broadcast-based, network access protocols [CLAR78, BUX81]. A resulting advantage is that system traffic may be simply monitored by the units connected to the communication subnetwork. Such monitoring allows inexpensive approximations of system state to be made, potentially an important feature as will be shown later. In contrast, system state estimation is difficult and expensive in larger scale computer networks where long point to point links are required.

The database server will consist of distributed file server devices such as disk drives of varying characteristics, tape drives, an intelligent controller, and a centralised or distributed operating system (protocol), organ.zing data access and the maintenance of its integrity. The protocol chosen will depend on network characteristics, and will be required to optimise data access by exploiting these characteristics. The access speed of current bulk data storage devices such as disk drives has not improved to the same extent as LA communication network capacities. As a result, "the communication problem has been pushed back to the operating system" [LIMB82]. The network will in addition, support features such as as gateways to the outside world, printers, and electronic mail but, their type and implementation details are not relevant in this examination.

It is our thesis that the use of graphics-based man-machine interfaces, together with other high data content media (e.g. voice) that need to be implementted efficiently, generates a new set of network performance issues not encountered in 'conventional' text based systems. The efficient support of interactive man-machine dialogues using such data types may therefore reveal inadequacies in current LA information systems design not previously visible. Briefly, our goal must be to identify the inadequacies and suggest means of remedying them, and thereby make the internal operation of the supporting LA network as transparent as possible to users; only in this way can we prevent the disruption of possibly complex interaction(s) by inherent system limitations.

3.3 The Data Management protocols

We will examine here the methodology by which data will be accessed and maintained within the network. As mentioned, workstations will be intelligent, and therefore capable of maintaining private or frequently accessed data or applications in a local data store of limited size. Data that is infrequently used or too extensive to store locally, will be retrieved from the remote database management system. This data management/operating system will operate on either a centralized or distributed protocol. Whichever protocol is used, it is suggested that such a distributed network should look like a centralised one to the user [DATE81], in the sense that the user need not (indeed should not) be aware of the physical location of data. In other words, the system should present a consistent and reliable interface. Although such a scenario appears initially simple, a number of complicating issues in fact intervene. Foremost is the need to maintain the integrity of data, while preserving fast access times. For example, assume a user wants to access and possibly alter a timetable of events; how would the system ascertain that all copies (possibly distributed) of the same timetable were kept up to date, and whether or not the user has permission to execute the alterations. This problem is exacerbated by the need to take into account the possibility of workstation or system failures. A description of an example LA network with similar design issues is available in [REED81], while general distributed database design considerations can be found in [CHU79].

The issue under examination is that of deciding the type of data access protocol to implement, and involves considerations which depend on the topology of the network, the type of data stores, communication links, and data access characteristics, of the type investigated in [CHU76, TOMI78]. Performance requirements in LA information networks differ however from those considered in larger scale computer networks. In these discussions the user's viewpoint is paramount in most considerations. Data privacy, security, and system reliability are prime considerations if users are to rely on a system to routinely perform information handling tasks. Efficiency of interaction with information, as well as system response delay to user commands, is equally important. These factors are different from those associated with larger systems not based on interactive information handling, where factors such as throughput and operating cost may be more important.

In these discussions we will name a collection of data items a file (e.g. a text page, a voice message), this being the smallest data unit of interest to us. The nature of its contents will not affect these discussions, only its size and access frequency. We will also only consider shareable files. The list of non-shared/private files will be available to the relevant, privileged users, and will not concern us further; we assume that these contribute a small proportion of the total network traffic. Data not stored in the network and accessed via the gateways will also not be considered here. The transfer, storage and manipulation of application programs need not be considered in isolation to those of the data files. We will assume that processes are mainly executed at the workstations and it is these, in effect, that are responsible for the access and updating of data in the system.

Before moving further, it is useful to reconsider some general characteristics of the network. We have assumed a LAN based on a high capacity communications subnetwork, perhaps using fiber optics. The amount of total distributed storage is assumed adequate for storing the totality of data in the system, and ideally with some spare capacity. Data access times are assumed to be significantly longer than data transmission times, and will therefore be considered as one of the bottlenecks. Data communication and data storage costs in a LAN are assumed fixed and small. The main objective of this treatment is therefore to reduce system data access delay. Intelligent workstations and data access controller(s) are assumed.

Data access takes place in the following steps: Upon a file access request from a user, the local workstation file store directory is searched. If the requested file is not found, a request is transmitted to the network operating system. At this stage, the following three generalized versions of remote data access are possible:

1) All requests are primarily routed to a centralized database controller. Upon receiving a data access request the central controller, which may itself be a storage device, examines a master file directory of appropriate file server units, and reroutes the request appropriately. The file server storing the requested data accesses that data and transmits it to the requesting workstation.

2) The user workstation interrogates all file servers in turn until the desired data is located. This method of access is termed the local directory approach.

3) The user's workstation maintains a copy of the master file directory and forwards a request directly to the appropriate file server. This method is termed the distributed directory approach

The approaches discussed above feature a number of advantages/disadvantages depending on network topology, file/directory update rates, communication/storage costs, etc. Combinations of the above are also possible, but we will consider these as the main examples of interest to us.

An important and desirable feature which must be mentioned at this stage is the desirability of maintaining multiple copies ot a file, as a result of our concentration on miminimum response delay. For example, if the state of the network is known, then the ability to route a file access request to a file server which at that instant can offer a better service time than, for example, another more heavily loaded one, may offer a significant improvement in performance. In effect, such a facility would tend to balance the effective system load, and in addition to reducing average delay, would reduce the variability of access delay, thereby improving the consistency of the interaction dialogue as perceived by the user. It would in addition, provide some protection against failures and congestion. According to Marcogliese [MARC81], "if such a (local area) system is provided with a mechanism for dynamic distribution of load among those hosts that possess the required capability, then replication of most frequently used modules is often mandatory for system efficiency". The benefits of adaptive routing, particularly when it does not involve additional costs in the form of extra traffic and/or operational costs, have been noted by other researchers in similar fields [CHOW77, FOSC78, YUM81, BOOR81] and are discussed in chapter 4; adaptive routing may also reduce system response delay variability [CHOW77], potentially an important factor in systems based on interactive information handling [GOOD82]. However, the provision of an adaptive routing facility leads to the problem of maintaining multiple copies of files [ELLI77, THOM78, GARC80, KOHL80]. Assuming, however, that a simple and reliable data maintanance protocol is available, then the proposed dynamic routing feature is desirable. The implications and approaches become clearer in the following discussions of the merits of the above mentioned protocols.

An important feature in any distributed database management system is the efficiency and reliability of its protocol with respect to the consistent implementation of file updates. Updates may be generated by the system or by the distributed users and, after being checked for validity, should be propagated to all copies of affected files. Problems of data consistency arise when an application or a user updates files at more than one node and when files at one node are updated by more than one user. The problem is also made difficult by the possibility of multiple copies of a file, locked files, system failures, etc, [CHU79, KOHL80].

In case 1, as only a single master data directory is maintained, file updating is simple. Update requests have to pass through the centralized controller, thereby avoiding update conflicts. The advantages of a central controller in such a network where database consistency becomes a major task are well known [KAMO81, ROSE78]. The network however relies totally on the central controller, posing reliability problems. Redundancy can however be employed as the controller is a pure processing device, possibly a microcomputer which can be replicated cheaply and, for example, a polling protocol implemented for reliability. The implementation details need not concern us; suffice it to say that centralized data management simplifies the problem of managing distributed data. Adaptive routing is simplified as only the centralized controller needs to be aware of network state, and may therefore make instantaneous optimal routing decisions.

In case 2, the successive interrogation of file servers introduces additional delays. Adaptive routing is a complicated and expensive process, as it would necessitate additional server interrogations. File updating would require a fail-safe distributed data management protocol, known to be difficult to implement succesfully.

In case 3, the problem exists of maintaining up-to-date multiple, consistent, master directories, a difficult problem in a distributed access protocol. Data maintenance problems are as in case 2. Adaptive routing requires that all workstations are aware of current system state, a more difficult proposition than in the case of a centralized controller.

Summarizing, protocols 2 and 3, are less suitable when required to support data updating and adaptive routing. Although these problems exist in conventional centralised database systems, they are potentially more critical in a distributed environment due to update time delays. Distributed data management is suitable mainly for networks with high data storage and transmission costs. Therefore, the central intelligent controller case will be examined here as an option offering the simplest and most reliable data management/operating system option in a LAN. In certain cases, the central controller offers, in addition, the least operating cost option [TOMI78]. It is known to simplify data maintenance considerably and allows a more robust (against failures) protocol [KAMO81]. In addition, it lends itself readily to adaptive routing techniques.

Alternatively, we may argue that, in a LAN, the cost of communicating with a neighbouring database is approximately equal to communicating with a remote one within that LA network [MARC81]. One such device situated anywhere in the network, may therefore, handle data access as efficiently as the distributed cases.

Advantages however exist in the form of a cheap, simple, more robust, data access and dynamic routing protocol.

The high data capacity expected of the communication subnetwork, and the selection of a centralized controller, allow us to conceptualize the file management section of the network in terms of a virtual STAR network, or central server system, as all requests to the network are routed and processed via a central node. Central controller processing and subsequent re-routing times are considered to be negligibly small. This may not be valid in all cases, but it is believed that the trend toward very high LA communication network data transfer rates supports these assumptions and arguments.

Adaptive routing in the central controller case is straightforward, involving the monitoring of network state and using this information to route file access requests dynamically. A good routing protocol [KLEI76] "should ensure minimum delay and adapt to failures and varying traffic loads by bypassing congested areas". Routing in a LAN is therefore based on a knowledge of file server directories, their mean access rates, and on whether they are operational.

This monitoring of system state is simple to implement in a LAN. In addition to the high data transmission capacities available, which may allow servers to directly signal their current loading, a common feature of current LAN's is their broadcast based network access protocols. This means that at the communication network access level, all stations listen to all transmissions and receive messages addressed to themselves. This broadcast operation is an attractive alternative to circuit switched networks for a LAN [BUX81]. In ETHERNET [METC76] for example, transmissions are monitored by all stations connected to the bus, including the receiving one. In RINGS, the data packet(s) can easily be made to reach all stations and the same is true of most TREE and STAR topologies [CLAR78]. Simply by monitoring network traffic therefore, the central controller may assess network state, either by timing service time intervals, or by reading piggy-back messages sent on top of the requested (serviced) files. Having assessed file server loadings and by being aware of their service characteristics, the central controller may then make an instantaneously optimal routing decision based on minimum expected service time. The term dynamic (adaptive) routing therefore describes techniques which are time-sensitive as opposed to time-fixed ones. The processing of a file request from its inception to the file's reception at the workstation is briefly outlined in Figure 3.2.

Summarizing, the centralized data access protocol has the following advantages:

a) a centralised master directory simplifying data access, consistency maintenance and robustness against failures
b) dynamic routing, as a powerful means of reducing response delay, can be implemented easily
c) data access as viewed from the workstations is simplified

3.4 Performance Enhancements

The problem we shall be examining here may be stated as follows: given the probable implementation of man-machine dialogues employing high data content media (e.g. graphics, voice), what can be done to improve user service quality?

User service quality has many facets [PYKE76, FOLE71, FOLE74], but here we will be concerned mainly with how to make the supporting system's operation as transparent as possible to users by minimising response delays and the extent of data maintenance problems. We will be attempting to alter the dynamic aspect of user quality, i.e. by analysing those factors which are variable in a designed, operating system. This is in contrast to approaches geared towards design time optimisation [FOLE71, COTT77], and which attempt to find (fixed) optimum system configurations based on cost/performance criteria. The network in the present case will be assumed to be fixed in terms of network capacities and file server characteristics. We will assume that response delay becomes significant in data access transactions which must be serviced remotely with respect to a workstation. As the network is assumed fixed, the available control variables are the location of files in the network.

For example, if we assume that a user accesses a particular file frequently, then perhaps it would be profitable -if that file is not updated too frequently- to maintain that file local with respect to that user, avoiding the need to continuously request data through the network and thereby improving response time. In addition, the total network traffic, would be reduced so that service times as viewed by other users would improve. As another example, if we assume that a certain data file is accessed by a lot of users, then a good strategy is to maintain several copies of that file, if spare storage capacity exists, distributed among the file servers. Other factors such as data privacy, congestion avoidance and the limited local data storage capacity, may intervene in deciding file placement.

This approach, of optimally allocating files in the network subject to a set of criteria, was first proposed by W. Chu [CHU69] in the context of large scale computer networks. A number of studies have followed, each dealing with a specific example network and exploiting its particular features; a good review is available in [DOWD82]. The file allocation problem is a non-trivial one [DOWD82, PEEB78] and most attempts at modeling produce a cumbersome 0-1 integer programming problem, with rarely an efficient and optimal solution available. The reason for this is that the variable order is of value MxN, where M is the number of file servers, and N the number of distinct files in the network, potentially a large number. The problem has been shown to be difficult to solve intuitively, and heuristic techniques exploiting individual network features offer an alternative [DOWD82, WAH84].

To summarize, we observe that, in the light of increased performance requirements, it is probable that there will exist a need for improving such LA Information systems performance, particularly with their intended use in demanding, office environments. In this presentation we have indicated that adaptive data allocation is an effective means for boosting performance. LA networks posess a number of characteristics favourable to the implementation of adaptive file allocation techniques. This is in contrast to large scale computer networks where the large overhead associated with transferring files between nodes may dictate against a file re-allocation.

File allocation studies such as the ones referenced above have mainly concentrated on large scale computer network file allocation, with performance objectives concentrating mainly on reducing operating costs, such as communication and data storage costs. However, once a network is installed, operating cost becomes a secondary issue, and techniques to allow a system to reach its full performance potential are required. Few studies have considered the problem of adaptive file allocation in small scale networks [DOWD78, FOST81, TZAV82, TZAV83A], where user's viewpoints (in the form of response delay) are primarily considered rather than operating cost. On a still smaller scale, there exists some work on distributing the workload between memory hierarchies in a simplified model of a computer system [CHEN73B], again with a view to reducing overall memory access delays in that computer system.

3.5 File Allocation between the system and the users

We first consider the macro (in the network sense) distribution of data. In particular, we will regard the central controller and the distributed file servers as a single unit, responsible for the coordinated access and maintenance of non-locally stored files.

The problem we shall consider here is: given the number, size and characteristics of shared data files in the system, which should we store locally with respect to users? We may define a 0-1 variable X_{ij}, where i denotes a file and j a user workstation. X_{ij} is 1 if file i is stored locally with respect to user workstation j and 0 otherwise. The parameters available, and which may be obtained through system monitoring, are the average access rate Q_{ij} of file i by user j, and the average update rate U_i of file i. Updates may originate from users or from the system, the overall update rate being of interest here. In a real environment, e.g. an office, it is probable that steady usage patterns exist and these may therefore be monitored in an attempt to assess the relative importance of data files to a user or group of users. The fraction of local storage space taken up, and the amount of network traffic generated by the maintenance of a file local to a user, are also directly measurable.

The generalized nature of these discussions does not allow the presentation of precise techniques for the system-users aspect of the file allocation problem. We will therefore only indicate the general issues and trade-offs involved in deciding file placement in a LA network. A global model would have as variables the location of shareable files in the form of the Xij's. The objective of the model is the reduction of overall response delay. We would assume that locally stored files offer small response delays for that particular workstation, whereas centrally stored ones offer delays which are a function of queuing delays due to network access, file access rates, network loading, etc. Network loading (offered traffic) is, in turn, a function of access/update rates, the length of files, background traffic, etc. System constraints are the capacity of the local server(s), the file server(s), permissible network load, etc. These can be used to form a set of weights/penalties to be used in deciding a file's allocation. A formulation will not be presented, numerous approaches and suggestions being available in the literature [DOWD82, CHU79]. In addition, the evolving nature of such systems design may invalidate attempts at a precise formulation.

We may however comment on the likely characteristics of such a formulation. By primarily considering overall response delay we are in danger of improving response delay for some users at the expense of others. This may be controlled through additional constraints, though these will complicate the model. The main problem with such models is the number of Xij variables involved, of the order of MxN, where M is the number of users and N the number of files. If, for example, we consider 10 users and 100 shared data files, by no means a worst case example, we are faced with 1000, 0-1 variables.

Available optimal 0-1 linear integer programming techniques are limited to linear problems with a maximum of about 100 variables [GUE68, TAHA82]. We may therefore conclude that such models, with the possible additional complication of non-linearities (e.g. delay vs network load), are prohibitively costly to solve optimally for all but the smallest examples. Heuristics can however be developed which exploit network characteristics; the main problem with these, however, is that of ascertaining the extent of optimality of the allocations arrived at, and the consistency of their performance [DOWD82, PEEB78].

Alternatively, we may propose an approach which considers each user in isolation, with an objective function that considers only files relevant to each user [TZAV83A]. This is made possible by considering network traffic as a commodity, shared by users of a LAN. Such a technique is conceptually similar to 'single-file' allocation approaches [DOWD82], although based on different assumptions. The constraints are workstation storage capacity and the amount of network capacity allocated to each user in addition to routine background traffic. The latter constraint is soft, and may therefore be included in the objective in the form of a penalty. By allowing to each user a predefined fraction of the network traffic capacity, we may isolate the model from the remainder of the network. This capacity allocation, however, may be a variable parameter responding to changes in usage and demand. The resulting model may be solved in isolation as it involves only one user. Usage statistics and other parameters are available by monitoring system use, allowing the model to be solved locally at the workstation. In effect, this would allow the global problem solution to be distributed among the intelligent workstations.

Problem order is thereby reduced by M, the number of users, and approximate solution heuristics may be applied. The cost of executing a new file re-allocation may be included in the model to balance the improvements possible versus the cost of achieving the new configuration. Incremental design, in the form of considering a single file at a time, is also possible.

An exact description will not be presented as these formulations are by necessity suggestions based on approximations. A more precise formulation would necessitate knowledge of precise network characteristics. The ability however to attempt such simplifications is confined to LAN's [TZAV83A]. This is in contrast to large scale computer networks where we may no longer consider a global network capacity allocation to users. The fact that Local Area networks are still evolving does not allow the verification of the assumptions made nor an estimation of system parameters. The description and methods presented therefore are intended only as guidelines rather than as exact approaches.

3.6 File allocation between the file servers

This section will consider the (micro) allocation of files between the file servers themselves, i.e. within the centralized data management system. The number and type of files to be stored in the system are assumed to be predetermined by procedures similar to those described in the previous section. A general description of the data access protocol is repeated for clarity (Figure 3.2). The central controller, upon receiving a request for a file, examines its master directory to ascertain the number of available copies of that file, and their location. If more than one copy is available, then by virtue of the ability to monitor loadings at each of the file servers, the controller routes the request to the file server which as well as storing that file, offers the best estimate for shortest response time. Such an arrangement has the advantage of reducing mean response delay and response delay variance by balancing effective server loadings. Our goal is to find the optimum location of files and also to determine which files must be replicated given a system and its access characteristics.

Queues build up at the file servers due to the randomness of service request arrivals, and the randomness of service times. Figure 3.2 is therefore redrawn as Figure 3.3. We will assume that data transmission time is small compared to data access time, by virtue of the use of a high data rate LA communications network. Node processing (e.g. routing) time is also assumed as small, the queues shown therefore being due to the variability of access times in the file servers. Alternatively, if no queuing is possible at the file servers, requests may build up at the central controller until such time that a server retrieves and transmits a previously requested file and signals that it is idle, as in Figure 3.4. If no queue jumping (jockeying) is allowed within the central controller, then this model is similar to the one described above (Figure 3.3). If movement between queues is allowed, then the problem is more complex. In this thesis we will primarily consider the case where inter-queue movement is not allowed or in other words where there is queuing at the servers.

By virtue of the assumed high data rate of the LA communications subnetwork as compared to data access rates [LIMB82], it may be removed from our model, producing the central server model of Figure 3.5. This removal is further justified by other researchers [MARC81]: "in local network topologies, we may assume that the average transmission delay through the channel does not depend on the distance between the transmitting and receiving station. Consequently, software distribution does not depend on the mutual distance between hosts (as it would for larger scale nets) but on the kind of services the network offers to users, on the structure of software modules, and on their interactions".

The reason for the existence of a number of different storage (file server) devices is the capacity/access rate/cost tradeoffs normally involved with such devices. A company or organization requiring the use of a LAN would therefore need to have access to a variety of such devices to cater for a variety of storage needs. For example, few fast access devices for frequently used data, while capacious but slow access devices for infrequently used, large, or backup files would be necessary.

In the above context, therefore, the problem we will be examining here is as follows: Given the performance characteristics of available storage devices, and assuming that the bottleneck occurs mainly due to data access rather than data transmission/processing, what is the best approach to the allocation and/or replication of shared data files between storage devices with the objective of reducing overall response delay?

The problem of the optimal allocation of files can in some cases be formulated as an optimal routing decision [DOWD78, CHEN73B]. The routing proposed in these studies is however considered as static and probabilistic, in the sense that fixed branching probabilities are attached to the server branches of Figure 3.5. The routing decision is equivalent to throwing a biased die for every arriving request. The advantage of this procedure is that the arrival streams remain Poisson with a consequent gain in analytic simplicity.

The problem has therefore been treated by obtaining these optimal branching probabilities and executing a file allocation which matches them as closely as possible. Dowdy [DOWD78], however, in his static routing analysis, reiterates the point that an instantaneous routing decision is desirable. He does not however attempt to analyse such a technique, proposing that: "device capacity constraints, excessive update traffic, system monitoring overhead, and the dynamic routing mechanism may prove crushing". Here, we have nevertheless suggested that the validity of these arguments is diminished with the advent of new LA networking developments, and any (small) overheads are offset by the considerable gains in performance, as will be described in chapter 4.

File allocation/replication in central server networks as far as is known, under the assumption of time sensitive dynamic routing has not been considered in the literature. In order to tackle such a problem we need to be aware of its structure and/or of an efficient analysis technique. Such knowledge would allow us to evaluate the effects of alternative allocations and with the aid of an optimizing procedure, suggest other potentially advantageous allocations until a satisfactory solution is reached. We are therefore interested in knowing the performance (in terms of average response delay) of networks of the form represented in Figure 3.5, for different file allocation/replication patterns.

3.7 Dynamic Routing networks

Re-iterating, the simplified central server model of the Local Area network arrived at (Figure 3.5), is based upon the following characteristics:

a) the suitability of a server for servicing a request (the service capability)

b) minimum expected response time routing

The servers are not necessarily identical, and may serve only a subset of service requests. More detailed characteristics suited to the queuing analysis of the model will be discussed in the following chapter.

This simplified model, is similar to models that have been used to model the performance of Packet Switched Networks [FOSC77, WONG78]. This is an interesting similarity, since by simplifying the complexity of a LAN database, we have shown it to be similar to a class of models used to analyse adaptive routing Packet Switched Network (PSN) nodes. Briefly, in the PSN model, the service request arrivals are equivalent to the packets reaching the PSN node from all links. The node may examine packets and check their destination, routing dynamically to an outgoing link which ultimately leads to the packet destination. The routing decision is based on link capacities, link loading, and suitability for reaching the required destination, considerations similar with those of the model of Figure 3.5. The objective in the PSN case is to predict the average delay of packets traversing this intelligent dynamic routing node. In general, queuing networks employing such dynamic (adaptive) routing have proved difficult to analyse [CHAND78, WONG78], their analysis therefore posing an interesting problem on its own, and applicable to other interesting cases in addition to the file allocation problem proposed here.

If dynamic routing is not allowed, and if all files are stored in all servers, then the problem has been solved analytically [CHEN73A]. The solution indicates the optimum partition of workload to the servers. File allocation based on numerical techniques, for the case of closed queuing networks where only a fixed number of jobs is allowed to circulate in the network, is available in [DOWD78]. These situations are not applicable in the present context of a dynamic routing network, but it is interesting to note that in simpler cases, exact solutions are available.

To conclude, we remark that the prediction of the optimum location of files in file servers devices requires an ability to analyse the dynamic routing network of Figure 3.5. Several researchers have in the past attempted this problem; their results and conclusions are critically reviewed in the following chapter.

3.8 Summary

A number of assumptions based on current research in interactive office information systems design allow us to propose simple system models. We have concentrated on user's viewpoints indicating that this leads to unique considerations and developments. The controlled factors which influence system operation have been outlined, and two data allocation problems proposed. Macro and micro models are covered with respect to the problem of optimally allocating data files in the network. Macro refers to the global network data distribution between the system and the users, and micro to data file allocation within the available file server units. The objective has been to indicate that adaptation techniques have a role to play in Local Area Office Information Systems design. The impetus for the use of such techniques has been the need to adapt current design philosophies to demanding real world environments which in combination with the probable use of graphics assisted information presentation/interaction, considerably tightens user service quality constraints. It is suggested that such techniques may be implemented more simply in LAN's rather than larger scale networks, though being of equally significant importance.

The absence, however, of sufficient usage information on such systems does not allow us to completely verify the assumptions made. The proposed models are therefore intended only as suggestions intended to initiate studies on the factors we believe critical with respect to the problem of allocating and/or replicating files efficiently in such a network. In the central file server allocation case, a network model also used in a Packet Switched Network context has emerged. This dynamic routing queuing network model has been identified by other researchers as difficult to analyse and poses therefore an interesting problem on its own, in addition to its use as a tool in deriving the optimum file allocation/replication pattern within the available file servers.

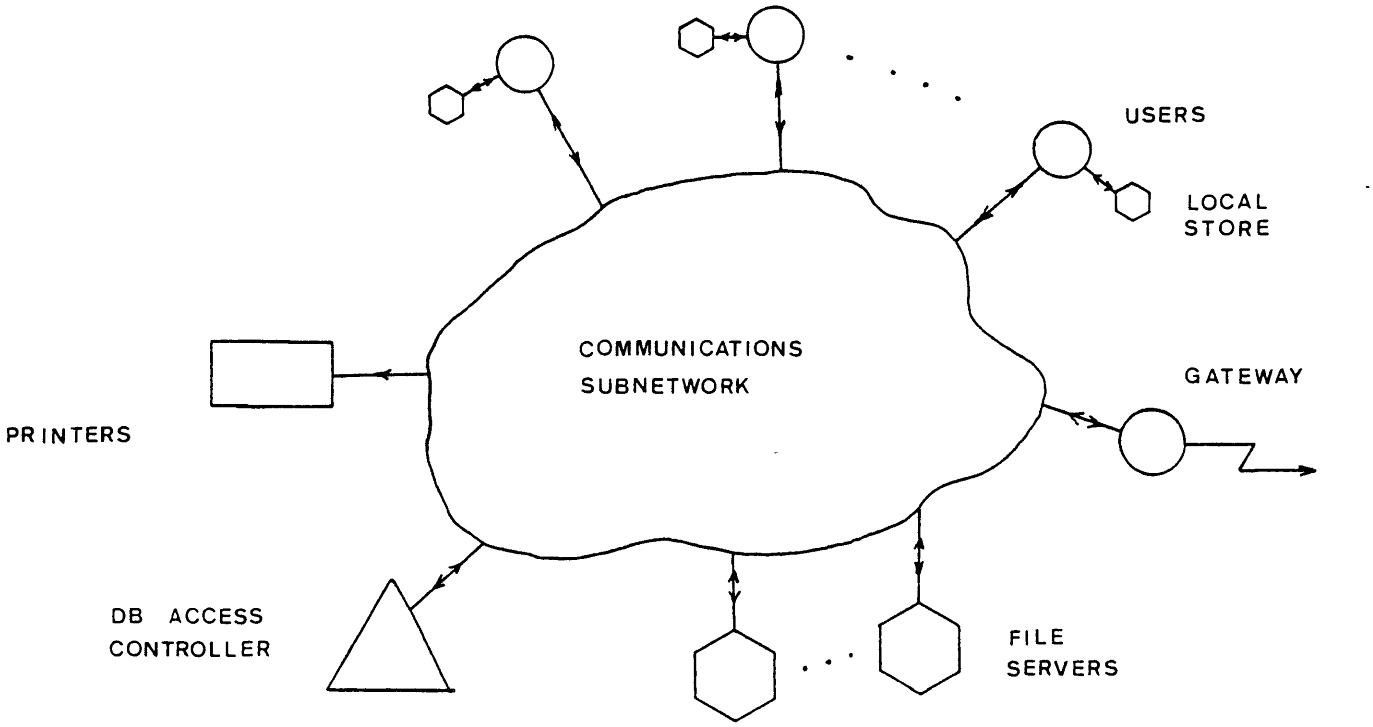


Figure 3.1 A general, LA information network

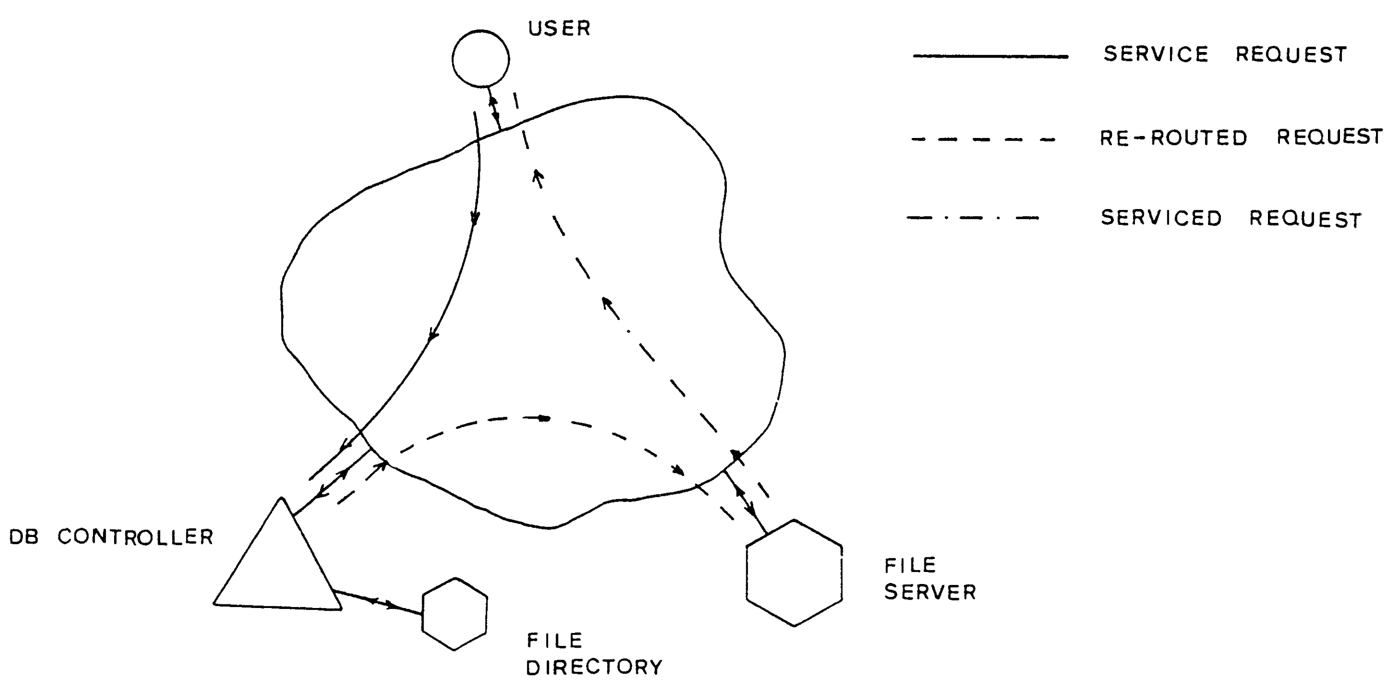


Figure 3.2 File access

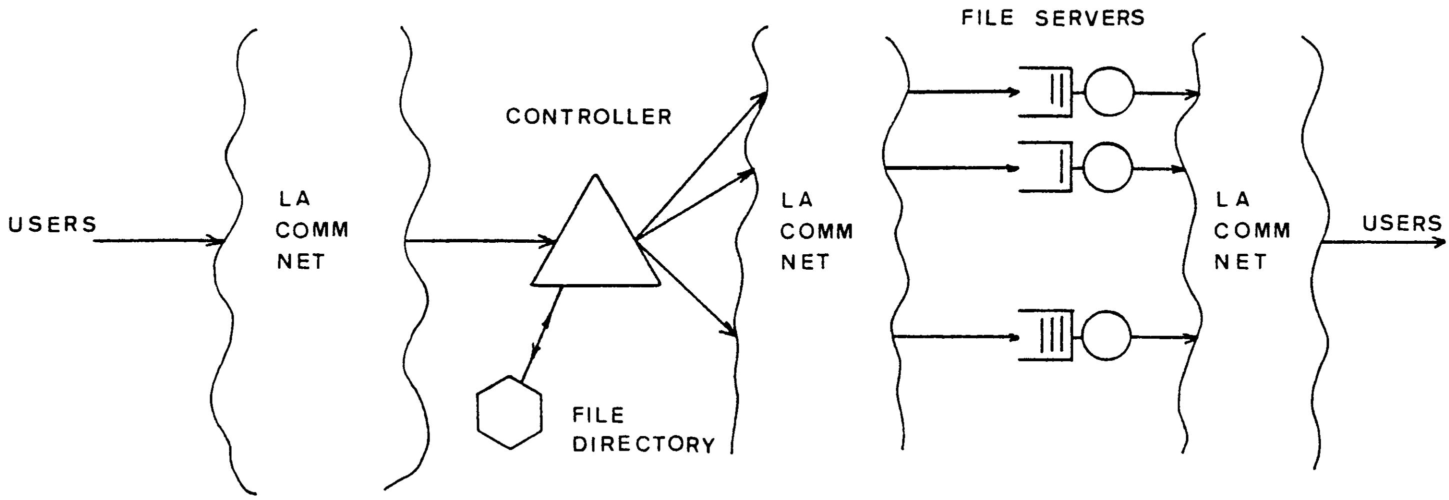


Figure 3.3 Queuing network model

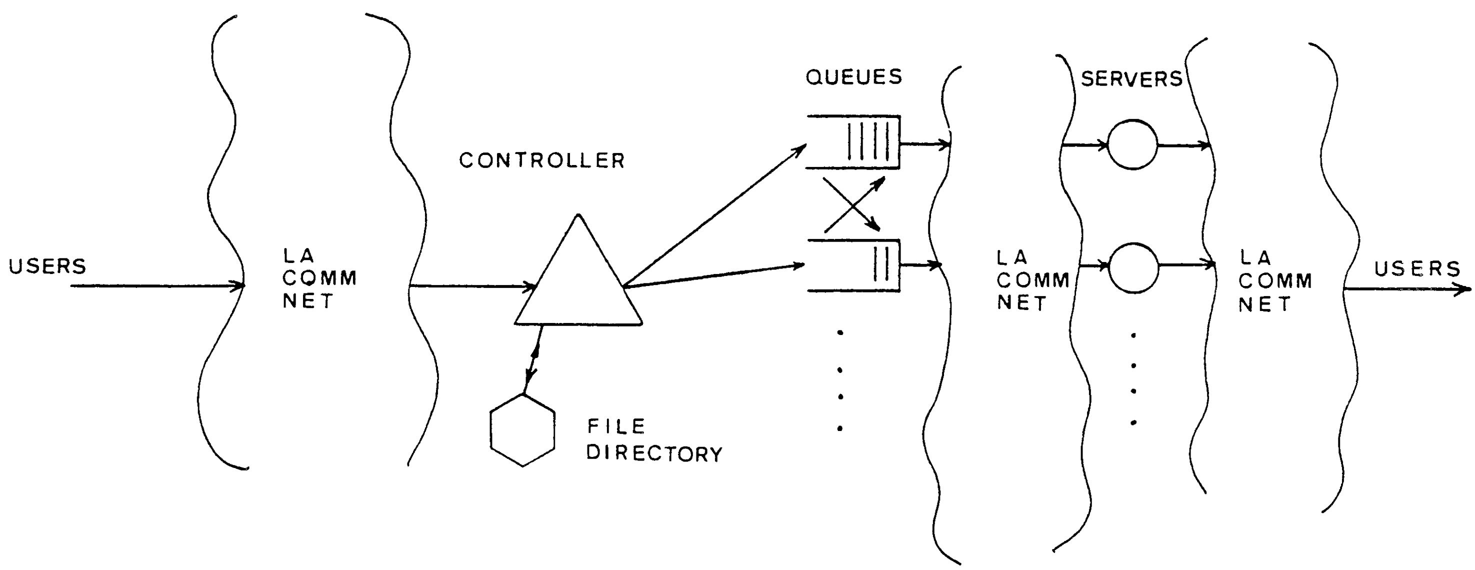


Figure 3 4 Queue jumping model

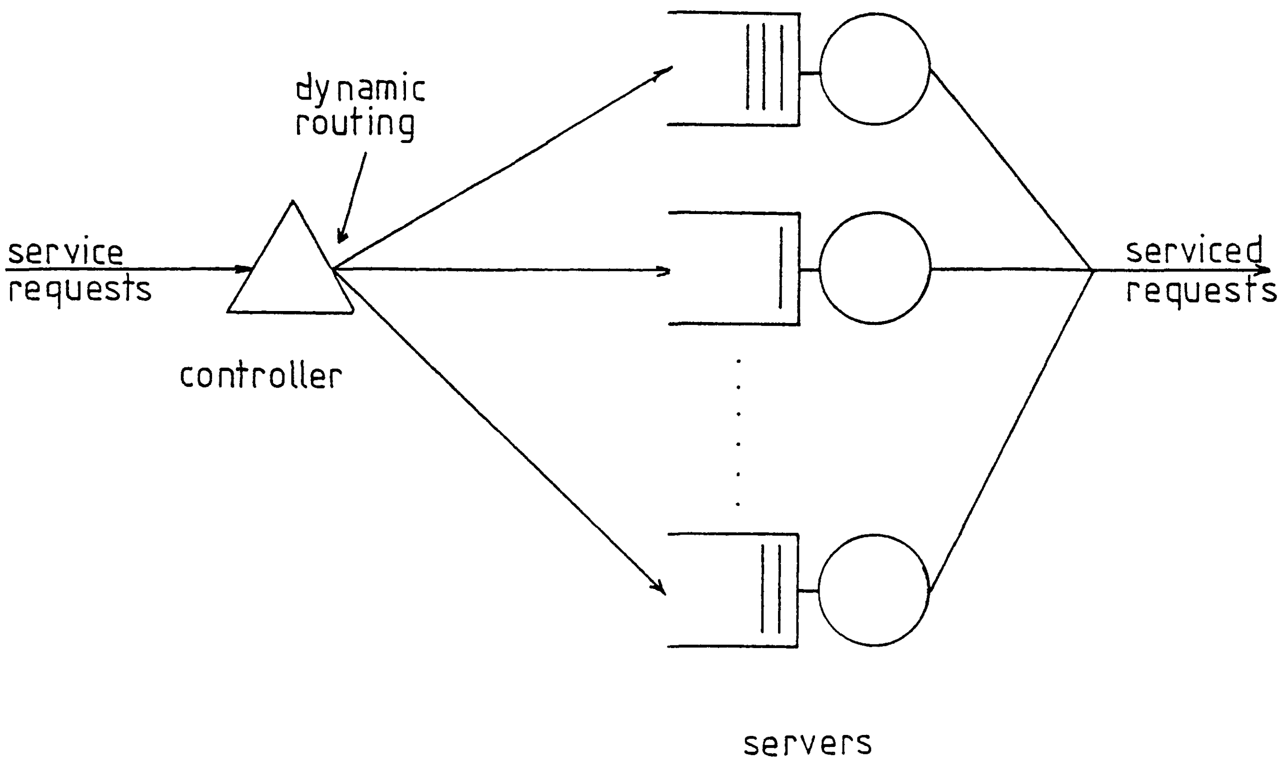


Figure 3.5 Central server model

CHAPTER 4

SURVEY OF TECHNIQUES FOR ANALYSING DYNAMIC ROUTING NETWORKS

4.1 Introduction

This chapter will investigate the available techniques for analysing dynamic routing central server models. The dynamic routing feature is a result of the possibility of maintaining multiple copies of files in these (file) servers. As represented in Figure 4.1, the model of interest consists cf an intelligent controller that examines externally arriving service requests, routing each request to one of the servers The routing criterion is based on minimum expected service time and capability of servers for handling requests. While an intermediate goal is to be able to analyse such a network, our final objective is to use an analysis technique in order to efficiently allocate and/or replicate files between the servers.

A queuing model is used to model this central server system, as it captures system behaviour as a function of the workload, job routing policy and system characteristics. The model has been developed from considerations of the operation of a Local Area information system. The same model has been used to model nodes of Packet Switched Networks (PSN's), and it is from this latter field that previous work on the analysis of the model has primarily been derived.

The reason for the existence of an intelligent controller is the need to balance the loading of the system servers; this balance is achieved by exploiting possible redundancies. Redundancy in a PSN context is manifested as the availability of several links leading (ultimately) to the same destination. In the file server context of interest to us here, redundancy is implemented by maintaining multiple copies of a file. When therefore, a request for a file arrives, it may for example, be routed to the server storing a copy of that file and offering the least expected response delay. It will be assumed throughout, that complete and up to date information is available to the central controller in the form of queue sizes, server characteristics, and the current file allocation.

4.2 Dynamic Routing

By a dynamic routing strategy we mean a policy that, for each arrival, bases the choice of a route for a request upon the information then available. A question naturally arises as to the extent to which improvements are possible with dynamic routing, particularly with regard to the trade-off between the necessarily increased complexity, and the gains in performance. Several studies have addressed this issue, both with regard to the best dynamic routing policy to implement, and the best system configuration that supports this routing policy. The performance measure of interest we will consider is average system response delay. We will assume that it is not possible to estimate in advance the amount of service time each request will require.

Chow and Kohler [CHOW77] have examined several system configurations for static and dynamic routing, where the routing policy in the dynamic case is simply to join the shortest queue of two identical servers. The objective is to achieve system balance by automatically directing jobs to processors (servers) perceived as more lightly loaded. Their conclusions are that for the configurations examined, which include 'queue jockeying' (i.e. queue jumping) types, there is a significant performance improvement with dynamic routing, of the order of 40-45% as compared to static (i.e. probabilistic) routing. Between the dynamic routing configurations examined, differences were small as compared to the improvement gained over the static routing case.

Chow and Kohler comment that the dynamic routing policy has, in effect, changed the inter-arrival times distribution from exponential to hypoexponential with the same mean. The hypoexponential distribution has a lower coefficient of variation than the exponential distribution, resulting in more consistent service times. This would be of particular importance for example, in an interactive man-machine dialogue environment where the above would be perceived as an improvement in the service time consistency of the supporting network. A reduction in response delay variance may have an effect on the success of an interaction dialogue. Although this variance effect was not found to be of considerable significance on its own [GOOD82], when combined with other factors (e.g. time of day) it can potentially have a detrimental effect on the effectiveness of a man-machine dialogue.

A number of different routing policies have been examined by Chow and Kohler [CHOW79], in order to assess their relative performance. These include policies based on minimum system time (overall), minimum instantaneous response time, static (probabilistic) routing, and a maximum system throughput policy.

This latter policy, in contrast with the other approaches, makes use of knowledge of the arrival rate. The system on which they were evaluated is a two-server system, with different service rates, the performance objective being minimum average response time. Significant improvements are reported over the static policy, with the best performance obtained from the maximum system throughput, and minimum instantaneous delay policies. It is commented that for high utilizations, the maximum system throughput policy achieves slightly better behaviour than the minimum response time policy, as it uses arrival rate information, and ensures that servers do not remain idle. In common with other researchers, the minimum expected response time policy will be primarily considered here as it is simple to implement, and near-optimal in the majority of cases.

An investigation based on the diffusion approximation approach [FOSC77, FOSC78], suggests that, under certain assumptions, the performance improvement that can be obtained from minimum delay based dynamic routing, approaches M, where M is the number of processors (servers). It was also found that even when servers of different rates and capabilities are used, then, under some constraints, a significant improvement in performance can still be obtained. Foschini notes that "a guiding principle should be that wise dynamic policies are those that balance, in the short term, the service delays corresponding to the various options".

An investigation into the relative trade-offs when dynamic routing is employed is available in [CHOU81]. The study assumes that dynamic routing carries a penalty (e.g. the cost of determining system state), and therefore is not always the least cost option. It is commented that a combination of deterministic and adaptive routing is desirable, the exact implementation depending on communication costs, the frequency with which traffic patterns change, etc. Adaptive routing is in general, desirable for unbalanced systems or systems in which chaotic (unexpected high traffic loads) conditions are possible.

Ephremides [EPHR80] has shown that when complete information is available on system state, then for two identical servers the optimal policy is to route to the shortest queue, this policy being optimal irrespective of the statistics of the job arrivals. A similar result is suggested for the case of several identical servers. Other disciplines are suggested if incomplete information is available.

Yum and Schwarz [YUM81], examine a variant of the join-the-shorter-queue policy (JSQ), the join-biased-queue policy (JBQ). The system examined is assumed to have proportions of dedicated and adaptively routed traffic flows. This situation, although presented in the context of a PSN, corresponds in the file server case, to the limited service capabilities feature. Markov chain analysis is used to analyse the networks, the results obtained indicating that different routing biases are required for different routing patterns, and that a 10-27% improvement in performance is possible over the best static probabilistic routing rule. The performance improvement reported here is lower, as the proportion of traffic that can be routed dynamically is not 100%.

Another study of dynamic routing [BOOR81], indicates that a large proportion of the improvement is still possible even when some servers cannot serve some of the requests, as long as 15-20% of the requests are not dedicated to one server. The improvement possible depends on the proportion of traffic having choice, and on how this choice is distributed among the servers. The analysis presented and resulting comments are based on simulation results. Boorstyn and Livne [BOOR81] make the comment that dynamic routing strategies have a profourd effect on a model's output process, which may affect the Poisson assumption for inputs to other nodes of a PSN. Some other problems of adaptive routing in large PSN's are discussed which are however not relevant in our isolated central server model.

We may at this point draw the conclusion that dynamic routing, when possible, is a desirable feature of the central server model. Performance improvements are reported as significant, while not being very sensitive to the implementation details of the dynamic routing policy.

4.3 The analysis of dynamic routing networks

The objective of this chapter is however to examine results of the analysis of such a system, rather than verify the gains possible through adaptive routing. The performance of interest to us is the average system response delay (turnaround time). All discussions from this point on will be based on the queuing model of a central server system of Figure 4.1.

Model characteristics are as follows: As the service requests are generated externally from a number of independent (in general) sources, they may be quite accurately assumed as forming a Poisson process, [BOOR81] with a mean arrival rate λ. In other words, the probability of an arrival in a small time interval dt, is λdt. This probability is independent of all past events (memoryless property), while dt is assumed to be small enough such that the probability of more than one arrival in dt is o(dt). The interarrival times of a Poisson process are as a consequence, exponentially distributed.

The service rates of the servers are also assumed as exponentially distributed with mean service rate μ . This approach has been conventionally taken mainly because of its modeling convenience and approximate resemblance to reality. The exponential service time distribution assumption, though not always a precise model for reality, can produce surprisingly accurate results [TRIVE78], having been widely used to model bulk data storage unit service times. It combines the effects of different file sizes, file access times (e.g. disk head seek time), and the data transfer rate of a mass storage device. In the PSN context, the equivalents are packet lengths, routing times and link data rates. These assumptions for the inter-arrival and service times are common to all of the approaches to be described.

We may now consider the transformation of our queuing model to a Markovian state diagram. Let n be the number of customers at server i. We may then define the state of our system as S, where $S=\{n_1,n_2,...,n_M\}$. As the interarrival and service times are assumed exponentially distributed, S completely defines the state of the system. An arrival or departure sends the system to a neighbouring state by incrementing or decrementing one of the n_i.

The rate with which states change depends on the arrival rate, service rates, routing protocol, etc. This discrete state, continuous time process, forms a Markov process as future states depend solely on the current states, and the service time distributions are assumed as exponential [CHOW77]. If the system is ergodic, i.e. assumed to attain an equilibrium distribution, the rate with which the system 'enters' a state S is equal to the rate with which it leaves that state S. By assigning a steady state occupancy probability P_i to each state S_i, we may equate the rate with which a process enters a state with the rate with which it departs from that state.

Consider for example the system of Figure 4.1. If it consisted of two servers only, and the maximum number of customers per server is N, then a possible (two-dimensional) state transition diagram is shown in Figure 4.2. The number of states is $(N+1)^2$, and we may therefore write down $(N+1)^2$ state balance equations, equating the flow into and out of each state. Due to the closed nature of the state diagram, one of the equations is redundant, and is replaced by the probability conservation condition $\sum_i \sum_j Pij=1$, where Pij is the state probability of state Sij. This system of $(N+1)^2$ linear equations in the Pij's may be solved to obtain these probabilities. Once these joint probabilities are known, many other probabilistic measures are readily established. A more detailed presentation of the philosophy and use of such Markov chain analysis is available in [KLEIN75, SAUE81].

The main advantages of this technique are that it is an exact, general approach, which can also be used in networks with non-exponential assumptions. The main drawback is the rapid expansion of the state space and consequent difficulty in solving a large number of simultaneous balance equations. For M servers constrained to a maximum of N customers each, the number of states is $(N+1)^M$. There exist however techniques that exploit the sparseness of the state transition matrix and can recursively obtain the joint equilibrium distribution [WALL66]. Such techniques can handle networks with several thousand states although the problem of forming the transition matrix for an arbitrary network remains.

For this reason, a number of attempts have been made at reducing the number of balance equations or of exploiting regularities in their structure. Notable among these are local balance methods [CHAN72, KLEIN75] where, for certain networks, we may equate the rate of arrival of customers in a single queue to the rate of their departure from that queue. This has the effect of significantly reducing the complexity of the balance equations so that they may be solved more efficiently. Unfortunately, dynamic routing networks do not in general, obey local balance conditions [FOSC77, TOWS80], necessitating a shift of attention to other approximate, but simpler to evaluate techniques.

Approximation techniques can also be employed to analyse queuing networks [CHAN78]. One of these is the diffusion approximation described in section 4.5, and the other is decomposition. Decomposition methods proceed generally in two steps: 1) analysis of a subsystem, and 2) analysis of a macro system composed of subsystems using the results of step 1. For example, a subsystem may consist of a set of interacting queues lumped together. In general, the rate of interaction among queues must be significantly higher than the rate of interaction of the subsystem with the remainder of the system. By parameterizing the subsystem(s) they may then be re-connected to form the original system which may then be analysed more simply. Such techniques have been used succesfully for a number of systems although in general, they have not as yet been rigorously analysed. For example there is no assurance besides empirical results that an approximation will give satisfactory answers, though approximations may quickly give values very close to the exact ones. Approximations usually use heuristic solution techniques. The philosophy is very similar to that of artificial intelligence search strategies. The methods are reasonable but, in general, it cannot be rigorously proven that the methods provide the desired solution.

Summarizing, our objective is to analyse (perhaps approximately) a dynamic routing central server network. The analysis technique must be efficient, and must handle any number of servers of different service rates and of constrained service capabilities. It should be consistent, simple to implement, and suited to an optimising procedure intended for optimally allocating/replicating files within a network. The following sections will summarize the main analysis techniques available, and examine to what extent the above requirement is met.

4.4 Direct Techniques

These are approaches which basically form and solve the system balance equations, exploiting symmetries where they are available. They are thus accurate, but due to the number of balance equations that need to be solved are confined to small (2-3 server) systems.

The work reported by Chow and Kohler [CHOW77, CHOW79], has been motivated by an investigation into the merits of dynamic routing in a two-processor central server system under a variety of routing protocols and connectivities. The analysis method used is based on a direct solution of the system balance equations.

In their first paper the routing protocol examined is a join-shorter-queue (JSQ), as the two servers are assumed as identical. The possibility of transfer between queues (jockeying) after the initial routing decision has been taken, is also examined. The objective has been to assess which configurations achieve better system balance and therefore better performance.

The method used for the analysis of this two server system is a recursive technique used to recursively solve the system balance equations. Chow and Kohler mention that in the absence of other suitable analysis tools at the time, an exact approach was inevitable. Such general networks unfortunately do not satisfy local balance assumptions, and hence no simple reduction of the number of equations is possible. Chow and Kohler assume a maximum number of customers per server. This is a reasonable approximation for light system loadings, if this maximum number is chosen carefully. In addition, by exploiting the symmetry of the system, it is possible to employ a recursive technique to solve a reduced system of balance equations.

In their second paper [CHOW79] Chow and Kohler analyse the performance of a number of different load balancing policies by routing requests dynamically to two different servers. The performance objective of interest is again mean turnaround time. The analysis technique is a generalization of the previous approach, employing recursion and taking advantage of the structure in the now non-symmetric, state diagram. Again, this may reduce the number of equations to be solved and simplify their solution, but the rapid expansion of state space is still there, even for systems of modest size. Chow and Kohler [CHOW79] concede that, "the generalization of the technique to three or more processors does not appear to be straightforward".

Such direct techniques offer the advantage of accuracy for light system loadings, and the wide variety of systems they can model if desired. They are however plagued by the common problem of similar methods of the size of the state space and of automatically forming the balance equations. State space expands rapidly with the number of servers and maximum allowed states per server, severely limiting such direct numerical techniques to systems of small dimensionality. In addition, the symmetry that Chow and Kohler have been able to exploit disappears in systems where each server can handle only a subset of the totality of service requests arriving at the system.

Some dynamic routing networks have homogeneous regions in their state transition diagrams in the sense that the balance equations are similar (only the indices vary) inside these regions. A technique that attempts to exploit these regions of homogeneity in the state transition diagram is presented in [YUM81]. This homogeneity allows a simplification of the equation forming process, but the problem of defining homogeneous areas for different systems and of state space size remains.

In general, it may be said that such balance equation solving techniques are cumbersome (i.e. difficult to implement and solve) and of limited practical value in analysing more than 2-3 server systems; they have been used mainly for the relative comparison of alternative routing policies where accurate analysis techniques are necessary. One may wonder therefore whether the conclusions thereby derived are valid for systems with larger state spaces.

4.5 Diffusion Approximations

One approach to the approximate analysis of queuing systems is through the use of the diffusion approximation. With this method, we consider a diffusion process as a Markov process with a continuous state space. Diffusion approximations are principally succesful in open networks with heavy traffic and have therefore been used to examine dynamic routing queuing networks. A good introduction to the diffusion process applied to queuing theory is available in [KOBA74, KLEI75, CHAN78, SAUE81]. Here we will briefly present an outline of the philosophy behind the diffusion approximation and its application to dynamic routing networks.

With the diffusion approximation, in essence, we approximate the movement of a particle d representing the state of a queue (its length) by a particle c, moving on the non-negative real line (Figure 4.3). Whereas d can only take values Ø,1,2 we let c take on values from the continuous non-negative real line. We now have to decide how particle c should move along the real line. Assume that the particle can only move at times Ø,T,2T,3T,..., T being a small constant time interval. In addition, c can only take steps of magnitude M. In each interval T, the particle takes a step M with probability p, and a step -M with a probability 1-p. In time nT the total displacement of the particle will be the sum of n independent, identically distributed random variables. As n gets large, the distribution of the particle displacement approaches that of a normal distribution [SAUE81].

If the position of c at time t is x(t), the displacement of the particle in an interval (t,t+dt) is dx(t) where

$$dx(t)=x(t+dt)-x(t)$$

We will assume that dx(t) is normally distributed with mean βdt and variance γdt. We denote as $p(x_o,x,t)$ the density function of the process x(t) given that $x_o=x(\emptyset)$.

Our objective however is to deduce the behaviour of particle d from the behaviour of particle c. To do this we simulate the behaviour of d by $d^\star$, which also jumps in discrete intervals but whose movement is dictated by c, as in Figure 4.3. Statistics regarding $d^\star$ are said to be diffusion approximations of those for d. The modeling accuracy of the method [SAUE81] depends on:

1. How values are assigned to β and γ.
2. How the real line is partitioned into intervals.
3. The boundary conditions applied to the diffusion process e.g. $x(t) \geq 0$ for all t.

As an example, assume that the arrival rate of a queue is λ, and its service rate is μ. Then the expected increase in queue size in an interval dt is $(\lambda-\mu)dt$. As the mean of the diffusion process is βdt, then it is reasonable to set

$$\beta = \lambda - \mu$$

By a similar argument [KLEI76], we may set

$$\gamma = CVa^2 \lambda + CVs^2 \mu$$

where CVa and CVs are the coefficients of variation of the interarrival and service time respectively. Having obtained estimates for β and γ (for the case of interest) we must set the boundary condition that $x(t) \geq 0$, for all t. There exist several ways of doing this [KLEI75, SAUE81], while in some cases [FOSC78] the condition is satisfied automatically.

It can be shown that the density function of a diffusion process satisfies a partial differential equation known as the Fokker-Planck (FP) equation. Into this we may substitute for β and γ and then solve the equation to obtain the required queue length probability distribution.

The diffusion approach is in general difficult to apply to new problems. It can however deal with cases where conventional approaches fail, such as by providing insight into the transient behaviour of queues [KLEI76]. It is mostly accurate for the heavy trafic case and can be thought of as the next level of modeling from average value analysis, using the variance of the arrival and service processes in addition to their means.

The application of heavy traffic diffusion analysis into dynamic routing networks was first introduced by R. Foschini [FOSC78]. The motivation behind Foschini's work is "to assess what dynamic routing policies have to offer over static policies, in terms of enhancing the resistance to delay performance degradation in a crisis situation when a node nears overload". Foschini concentrates on PSN networks, and the dynamic routing policy is that of routing (dispatching packets) to the feasible outgoing link that offers the least expected delay, i.e. $\min_i \{(n_i+1)/\mu_i\}$.

In the case of ties, these are resolved randomly with equal probability. Poisson arrivals and exponential service times are assumed. Foschini also comments, "under nominal operating conditions the stochastic nature of demand requires that a commercial computer network functions with substantial spare service capability". We comment that this spare capacity may be used to optimise performance under normal operation, or be used in cases of node breakdown, when other nodes may be overloaded, both being cases where it is necessary to employ dynamic routing.

In his earlier work [FOSC78], Foschini has attempted to analyse a two queue network with equal service rates and capabilities. He attempts this by treating the state transition diagram of a two-queue network as a random walk in two dimensions. By determining the dynamics of this random walk, he has been able to show that the 'equilibrium mass' of the random walk problem lies on the line where the queue lengths in each server are equal, a fact perhaps intuitively obvious through the symmetry of the system and its self-balancing action.

Having determined the differential mean and variance of the diffusion process, these may be substituted into the Fokker-Planck equation, which, it turns ou., can be solved by inspection while the solution automatically satisfies the boundary condition (i.e. positive queues). The results also show that in the heavy traffic case, the solution is the same as that of an M/M/2 queue with the same input. In other words, we may conclude that with minimum delay based routing we can approach the limiting case of a single queue served by two servers.

This advantage of a factor of 2, however, holds mainly in the limit of a heavy traffic load. By extending the argument to cover M queues, Foschini indicates that the maximum advantage of the dynamic over the static routing policy i.e. of breaking the input stream λ, into M separate ones of λ/M rate each, is M. It is also claimed that from the point of view of the routing protocol, "the minimum delay routing policy is asymptotically (heavy loading) optimal in terms of providing the greatest delay advantage possible", as it asymptotically approaches the limiting case of a single queue with a combined service rate, i.e. no server can ever go idle, while there is still work in the queue. Although an interesting insight into the behaviour and advantages of dynamic routing policies has been gained, the networks analyzed are not general enough to cover more realistic cases.

In the second paper [FOSC77], attention is shifted to servers with limited capabilities and of different service rates. However these effects are not considered simultaneously while the proposed analysis is based on systems of "minimum dimensionality".

In the limited service capability case, by considering the new dynamics of the state transition diagram, and by assuming that the incoming packet streams (by virtue of their ability to be re-routed) balance the loadings at the server, it is shown that for a 3 equal server example the performance gain is a factor of 3. The arguments and procedures are similar to those used in the previous case. However, the incoming packet streams are assumed as satisfying certain assumptions thereby losing generality, and indeed making the analysis impossible if the input streams are such that the servers are unbalanced.

In the case where two unequal servers are considered, then it is shown that the system now tends to balance itself on a new line of the state transition diagram. Foschini is able to propose that the delay advantage in the limiting, heavy traffic case, of a system of M, M/M/1 queues with service rates ($\mu_1, \mu_2, \ldots, \mu_M$) is:

$$\left(\sum_{k=1}^{k=M} \sqrt{\mu_k}\right)^2 \Big/ \sum_{k=1}^{k=M} \mu_k$$

The analysis however assumes that the traffic streams for a particular destination (or equivalently a file) do not 'degenerate' to values small enough to upset system balance.

In summary, heavy traffic diffusion analysis has yielded a useful insight into the advantages of adaptive routing. However, several restrictive assumptions have been made. In particular, restrictions have been placed on the nature of the arrival streams, a reasonable proposition in the presented context, as the goal is to compare two systems working under optimal conditions. As a result therefore, the method cannot be used to yield performance measures of a general system with general input streams. In addition, the analysis is useful only for the systems analysed, and has therefore to be repeated for new configurations, an impractical proposition. In addition, it is valid only for heavy traffic, and is practical for limited sized systems, under a number of restrictions. These restrictions are due to the goal of the analysis, i.e. that of assessing the potential improvement rather than as a precise performance description for each file allocation/replication pattern.

4.6 Local Balance methods

Local Balance refers to a property of some queuing networks that allows a considerable simplification of their global balance equations. It was first introduced in queuing networks by K. Chandy [CHAN72]. Local balance states that:
"the rate at which a process enters a given state due to the movement of a job into a given queue is equal to the rate at which the process leaves that state due to the movement of a job out of

that queue".

Unfortunately, queuing networks with dynamic (i.e. state dependent) routing in general do not satisfy local balance [CHAN78, CHOW77]. An exception is the class of routing policies identified by D. Towsley [TOWS80, SAUE83], which do satisfy local balance.

Towsley considers a closed network central server system with two queues and a fixed number of jobs circulating (Figure 4.4). The routing functions examined are rational functions of queue length. For even degrees of multiprogramming (number of jobs) N, a job leaving the CPU, goes to device i (i=1,2), with probability $(N/2-n_i)/(N-n_1-n_2)$ where n_1, n_2 are the number of jobs at devices 1 and 2. For odd N, the probability of going to device i is $((N+1)/2-n_i)/(N+1-n_1-n_2)$. Towsley then shows that the above network with such a routing satisfies local balance. It is claimed that these routing functions have a potential in modeling real-world problems of load balancing. These variable stochastic routing functions are not claimed to be optimal, since studies have shown that deterministic rather than variable stochastic policies are necessary.

In order to justify the routing policies analysed, Towsley has compared the following routing policies:

1. join shortest queue (JSQ)
2. static (probabilistic) routing
3. variable stochastic routing

Policy 2 is simple to analyse, while 3 can be analysed by virtue of its local balance property. Policy 1, the optimum case, is analysed through a direct solution of the system balance equations. Through a comparison of policies for various degrees of utilization, and of multiprogramming, of policies 1, and 3, it is concluded that the difference in performance between these policies is small. The work reported has been extended to cover more complex networks, constructed from 2 equal parallel queues.

This work is interesting from the point of view of discovering a class of adaptive routing strategies satisfying local balance. Local balance assumptions considerably simplify the solution of the balance equations, and therefore a direct analysis may be attempted. The networks analysed, however, are restricted to 2 parallel equal-rate queues. Limited service capabilities and optimal JSQ routing is not considered, as they invalidate the local balance basis of Towsley's treatment. Sauer [SAUE81] has developed a set of solution algorithms for treating networks such as those proposed by Towsley.

In summary, the as above approach is an attempt to find and analyse routing functions which conform to the available analysis tool (local balance). It must be concluded therefore that, although interesting, the methods presented lack generality, in the sense of efficiently analysing a central server network of any number of unequal servers of constrained capabilities.

4.7 Simulation

The failure of available techniques to efficiently treat the problem of analysing a general, central server network, naturally raises the prospect of employing simulation for the analysis. Simulation is a quite widely used tool when exact analysis techniques fail or when approximations are not available. Simulation models are more prevalent in practice as they can represent complex aspects of modeled systems faithfully. However, our main objective has been to obtain an optimal file allocation/replication pattern in a central server system. The use of a simulator is therefore inappropriate for the following reasons:

1. The computational cost of running one or more simulations, when a non-trivial number of file allocation patterns need assessing.
2. Statistical errors present in a simulator may affect the file allocation procedure, unless that procedure is designed specifically to take this effect into account.

4.8 Summary

This survey of available techniques for analysing dynamic routing networks has it is believed, revealed the need for new methods for analysing general dynamic routing networks.

The general system of interest here may contain any number of servers, each with a different service rate; servers may store copies of files thereby offering service alternatives to each arriving request. The routing protocol is of minimum expected response delay between the set of servers storing copies of a requested file.

The techniques presented in this chapter, address subsets of the as above, general requirement. In particular, exact techniques based on a solution of the system balance equations suffer from problems of state space size expansion even when their limited structural regularities can be exploited. They are thus used for small systems (2-3 servers) under light loadings; they are cumbersome to implement and difficult to solve. They offer the advantages of an exact solution and of general applicability.

Diffusion approximations have been used mainly to assess the improvements possible with dynamic routing. Although they have been presented as covering the different service rates and limited service capabilities case, this is done under restrictive assumptions, not valid in the general case.

Local balance methods offer a solution to the problem of large state space size. However the routing protocol examined is not optimal, and does not consider limited service capabilities or different service rates. The routing protocol, in addition, has been proposed and analysed only for 2 server queues.

Simulation is an alternative, albeit costly to execute. Its results can be difficult to analyse, and it is difficult to gain an insight into the system. In addition, it is not well suited to the problem of allocating/replicating files optimally.

It seems that efficient techniques for (approximately) analysing dynamic routing networks are yet to be developed. This is supported by the following comments from leading researchers, regarding the analysis of dynamic routing networks:

[FOSC77] "from a classical queuing theory standpoint, the dynamic routing problems subsume problems that have been long recognized as intractable".

[WONG78] "analytical results for models with complex features, e.g. adaptive routing algorithms and finite buffer space are not available".

[CHAN78] "a general approach to approximation methods for dynamic job routing is yet to be devised".

[FOSC78] "traditional analytic techniques are inadequate in dealing with networks with state sensitive dynamics. It appears that classical queuing approaches are too microscopic in their description, and lead to overwhelmingly complex system models. On the other hand, macroscopic models that deal with transmission of packets as average flows have not successfully represented the stochastic nature of the input while at the same time allowing meaningful performance evaluations".

[MARI79] "exact analytic solutions for complex networks are still limited to those which satisfy local balance equations, i.e.

networks in which the transition probabilities are fixed".

[TOWS8Ø] "the problem of analysing routing strategies in computer systems and communication networks is difficult".

In general [MUNT78], "the methods used to generate exact results for queuing networks have reached their limits. It is also fair to say that we have at this time no hint as to how to proceed to obtain exact solutions for general networks that do not have a product form solution. It seems, therefore, that the best hope for progress in the near future is through approximation techniques".

In the following chapter, a new approximation technique for such networks is proposed, developed and evaluated.

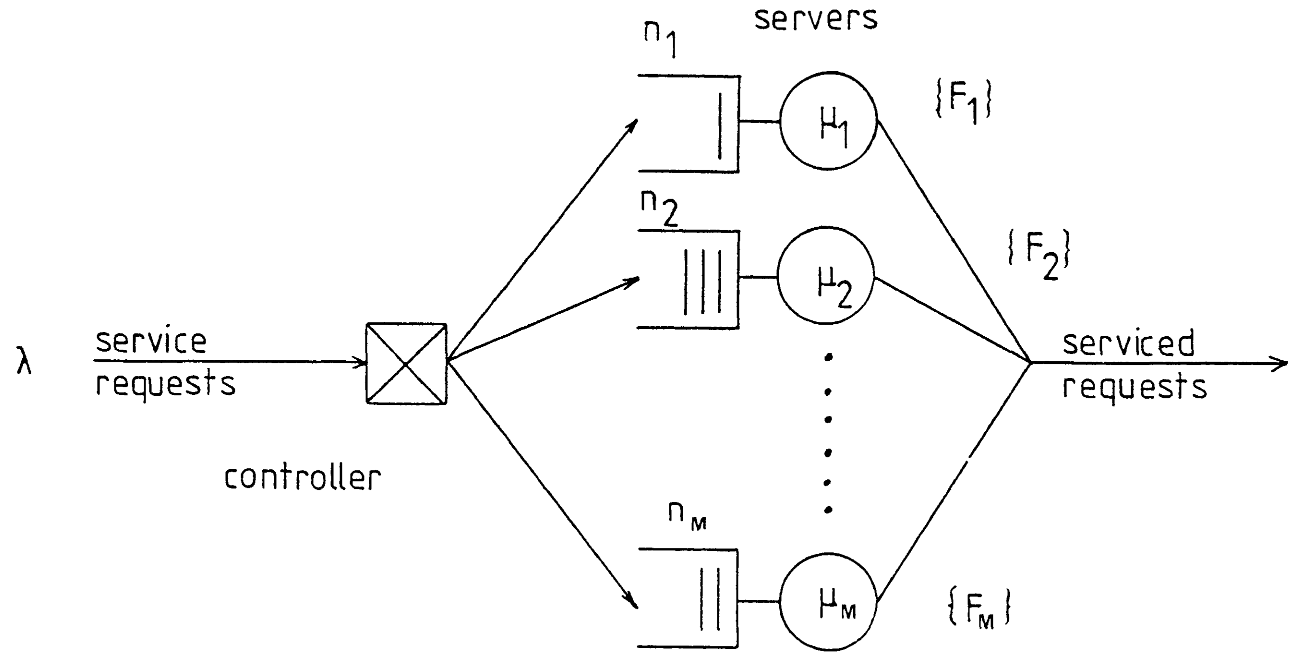


Figure 4.1 A central server network

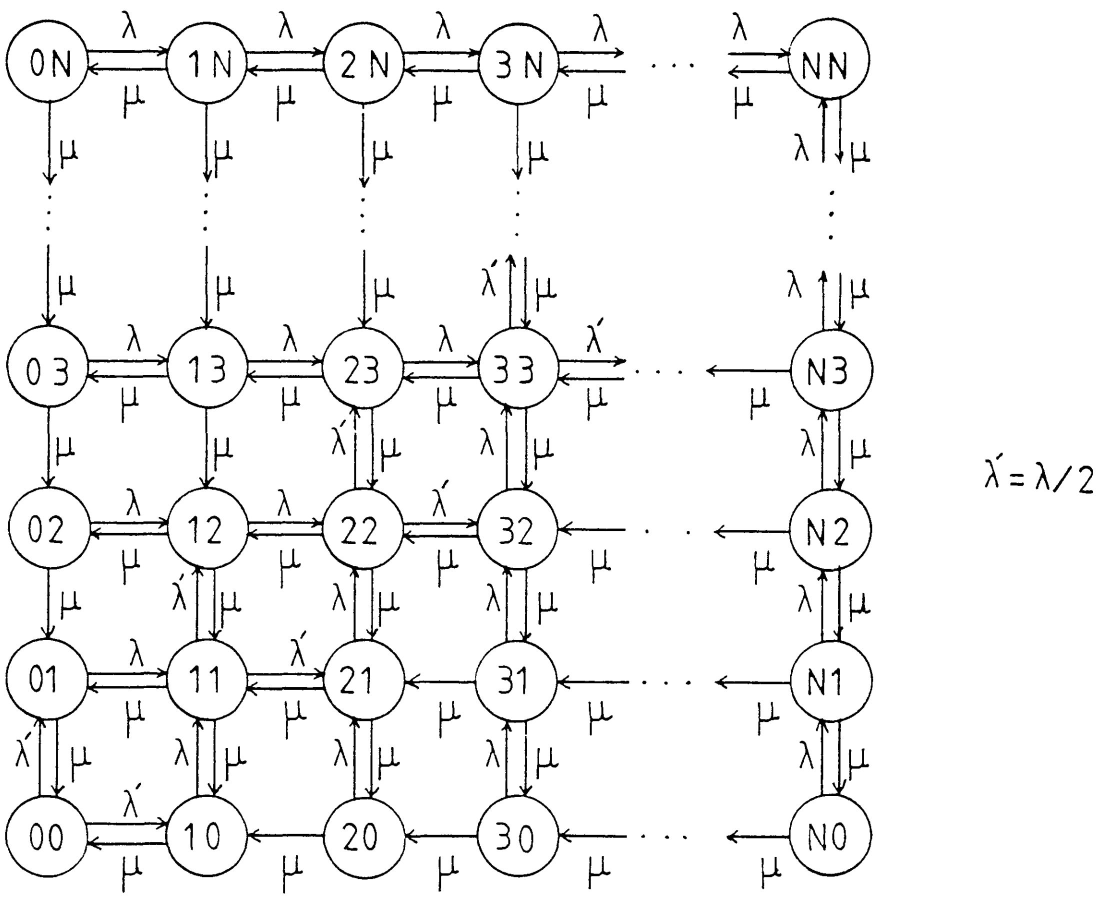


Figure 4.2 State transition diagram for two identical servers

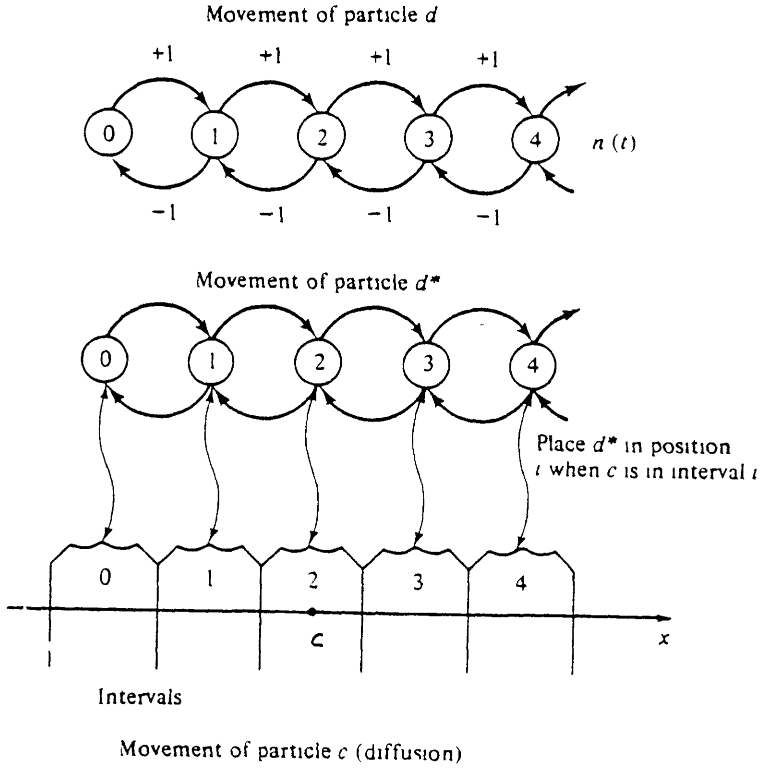


Figure 4.3 Diffusion Approximation

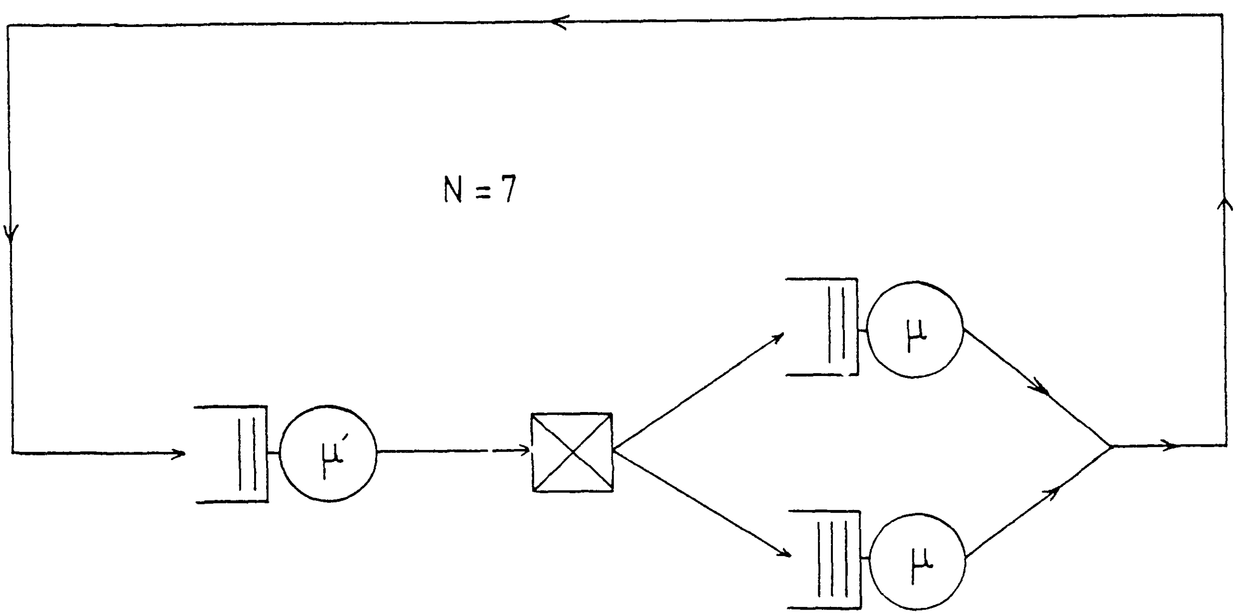


Figure 44 Closed, central server network

CHAPTER 5

A NEW APPROXIMATION TECHNIQUE

5.1 Introduction

In this chapter a new approximation technique for the analysis of open, dynamic routing central server networks is proposed [TZAV83B]. Such a network is depicted in Figure 4.1, and may be described as follows:

1) Arrivals to the network are Poisson with mean arrival rate λ.
2) Service time distributions are exponential with mean rate μ_i where i represents a server.
3) Arriving requests are for the files stored in the servers. Each server stores a set of files $\{F_i\}$. Arriving requests for file f therefore have the option of being routed to any of the servers storing that file.
4) The routing protocol is a minimum expected delay one, i.e. $\min_i(n_i+1)/\mu_i$, where n_i is the queue size at server i. In the case of ties, these are resolved arbitrarily by routing to the server with the smaller index.

The main interest is in finding an analysis technique for such a system. This technique is to be used for deciding the optimum allocation and/or replication of files in the system, and must therefore handle any combination of the above requirements efficiently.

5.2 Description of method

In order to present the philosophy behind the new technique, it is useful to consider the simplified case of Figure 4.1 in Figure 5.1. The servers are identical, and store all copies of files. Arriving service requests can therefore be routed to any server.

Consider the state transition diagram of each queue in isolation. This is a one dimensional birth-death process as has been used in the analysis of simpler queuing systems [KLEI75]. A representation of the birth-death diagram for each server is shown in Figure 5.2, assuming a maximum of 3 customers per server. In essence, this is a simplified form of the two dimensional state transition diagram of Figure 4.2. Consider the state transition rates of these diagrams due to service completions at each server. These are equal to the service rate for each server (μ), and can therefore be placed immediately beside the left pointing transitions in Figure 5.2. However, consider the state transitions due to arrivals at the servers. These will depend on the system arrival rate, and the current state of the servers.

For example, the transition rate from state Ø to state 1, for server 1 is equal to λ.This is justified as follows:
While server 1 is in state Ø (i.e. empty), server 2 can be in any other state. If server 2 is at any state other than empty (Ø), then it will be busier than server 1, and therefore an arriving request will be routed to server 1. If server 2 is empty (state Ø), then a tie situation exists, as both servers are empty, offering the same service time estimate (i.e. $1/\mu$). As it has been decided in the

case of such ties to route to the lower indexed server, then again a request is sent to server 1. Therefore irrespective of the state of server 2, the transition rate of server 1 from state Ø to state 1 is λ.

Consider the equivalent case for server 2, i.e. the transition from state Ø to state 1. If server 1 is busy, (i.e. any state other than Ø), then the request will be sent to server 2. If server 1 is empty, then a tie exists, and the request is sent to server 1 as explained above. The desired transition rate therefore is λ times the probability that server 1 is busy, i.e. λxP(server 1 busy).

The equilibrium state occupancy probabilities are defined as p_k^i , where i denotes a server, and k a state. The state transition rate for server i from state k to state k+1 is written as λ_k^i. Therefore λ_0^2 is written as $\lambda_0^2 = \lambda x(p_1^1 + p_2^1 + p_3^1)$ or $\lambda x(1-p_0^1)$.

The remaining state transition probabilities for the two birth-death processes are formed in a similar manner. In effect, the two queues have been isolated by describing them as single queues with inter-dependent arrival rates. These transition probabilities are substituted into the known state balance equations for such a process [KLEI75].

$$p_0^i = \left[p_1^i + p_2^i + p_3^i \right]^{-1} \qquad (5.1)$$

$$i = 1,2 \qquad k = 1,2,3$$

$$p_k^i = p_0^i \prod_{l=0}^{l=k-1} \frac{\lambda_k^i}{\mu_l} \qquad (5.2)$$

where the λ_k^i are functions of the p_k^i as in Figure 5.2. Eight non-linear equations are therefore obtained (5.1,5.2) in 8 variables (the p_k^i). Techniques such as Newton iteration may be employed to solve for these. However, such techniques are usually expensive to implement.

A more efficient approach can however be proposed for solving this system. It is observed that the state probabilities of server 1 are a function of the state probabilities of server 2 and vice-versa. This cyclic nature of the balance equations suggests an iterative solution. Initial values are chosen as $p_0^i = 1$, $p_k^i = 0$ for k=1,2,3. Equation 5.2 is evaluated for consecutive k=1,2,3, and these substituted into 5.1 to obtain p_0^1 . The p_k^1 (k=1,2,3) are then multiplied by p_0^1 to obtain the normalised values. These values are substituted into 5.1 and 5.2 for i=2, and the process repeated. The iteration is stopped when a steady iterate has been reached. The iteration stopping criterion examines six decimal digits of the solution and stops when these are steady between two iterations. By solution, we mean the average system delay as obtained from the state probabilities. It has been found experimentally that the required iterations are relatively few in number (less than 10 on average, depending on system parameters), producing good estimates for the state probabilities of this central server system. A more formal examination of convergence is presented in section 5.7.

5.3 Sources of Error

Before considering the general case of this method, the sources of error which make the technique an approximation rather than an exact approach must be examined. Firstly, the number of customers allowed in a queue has been limited to a maximum number N (N=3 in the example of Figure 5.2). The objective has been to bound the number of state balance equations that need to be solved. By doing so, it is assumed that the state probability for the higher states is small. This is true for low utilizations of the system. As the arrival rate is increased however, higher states must be included. One simple way of detecting whether more states are needed is to examine the value of $p^{1}_{S_1}$. If it is above a minimum threshold then it is an indication that more states are required in the state transition diagram, and these are added accordingly. This minimum threshold is chosen as .000001.

The second source of error is due to the use of unconditional probabilities. In particular, when forming the transition rates from one state to the next one up, the state occupancy probabilities of the other server are used. In fact, strictly, these should depend on the state of the current server as well. For example, λ_0^2 should be λx(1-P(server 1 is in state 0, <u>given</u> that server 2 is in state 0)). The use of conditional probabilities, however, would revert to the original state space model, thereby defeating the goal of this technique. By neglecting these conditional probabilities, the size of the original state space has been reduced to a more manageable size, using state dependent transition rates to model this dynamic routing network.

In networks satisfying local balance (i.e. fixed probabilistic routing) the equilibrium state probabilities of one queue are independent of the other, and therefore conditional probabilities are equal to the unconditional ones. In these cases, the technique is an exact one. It is therefore expected that this second source of error will depend on the degree of interaction between the queues. This interaction will increase with a larger arrival rate and with the number of options available for routing, as more load balancing will then take place. These effects will be further investigated in later sections.

5.4 General form of the method

Having presented an outline of the technique, the more general case of the proposed approximation is now considered [TZAV83B]. The general case will involve any number of servers of different service rates, storing files and copies of files. The central concept of the new technique is the formation of the state dependent transition probabilities due to arrivals λ'_k. The proposed form for this is as follows:

$$\lambda_k^i = \lambda \sum_{f \in F_i} \left\{ q_f \prod_j \left[\sum_{r = t_{j,i}}^{r = S_j} p_r^j \right] \right\} \qquad i = 1, \quad , M \quad k = 0, \ldots , S_i - 1 \tag{5.3}$$

where

$$j \in \Phi_{i,f} \cdot \Phi_{i,f} = \left\{ j \in \{1, \quad , M\} \quad f \in F_j \cdot j \neq i \right\}$$

$$t_{j,i} = \min \left\{ \xi \in \{0, \quad , S_j\} \quad \text{either } \frac{\xi+1}{\mu_j} > \frac{k+1}{\mu_i} \text{ if } i > j \quad \text{or } \frac{\xi+1}{\mu_j} \geq \frac{k+1}{\mu_i} \text{ if } i < j \right\}$$

where,

i,j - index a server

k - server state (number of customers in queue)

f - indexes a file

F_i - the set of files stored in server i

q_f - the probability of file f being requested

λ - overall system arrival rate

μ_i - service rate of the ith server

S - the maximum number of states of server j

p_r^j - the steady state probability of finding of server j in state r

$t_{j,i}$ - a threshold state signifying the state above and including which, server j offers a worse service delay estimate than server i.

$\Phi_{i,f}$ - the set of all servers storing file f, other than server i

Expession (5.3) is explained as follows: the transition rate λ^i_k, is the product of two factors; it is equal to the overall arrival rate λ, times the probability that an incoming request will be forwarded to server i. The probability that a request will be forwarded to server i, and therefore that server i will jump from state k to state k+1, is the first summation term in (5.3). The sum is for all files stored in server i ($f \in F_i$), the product of their probability of being accessed q_f and the probability that all other servers which also store file f offer a worse service delay estimate than the current server, i. This latter factor is simply the product $\prod_j$ over all these busy servers of the probabilities that they are busier than the current server, i. For a given server j, this probability is the summation over all of its states which offer a worse delay estimate than server i, beginning from a threshold state $t_{j,i}$. This threshold state is chosen to show that if equal service delay estimates are offered by two servers (i and j) then, arbitrarily, it has been chosen to route to the lower indexed one (i<j).

The state transition rates defined in (5.3) may be substituted into the standard expression for a birth-death queuing system [KLEI75], to obtain the steady state probability p^i_k of finding server i in state k:

$$p_k^i = p_0^i \prod_{l=0}^{l=k-1} \frac{\lambda_l^i}{\mu_i} \qquad k=1,..,S_i \,,\ i=1,\ ,M \tag{5.4}$$

$$p_0^i = \left(1 + \sum_{k=1}^{k=S_i} p_k^i \right)^{-1} \tag{5.5}$$

This is a system of $\sum_{i=1}^{i=M} S_i + 1$ non-linear equations, to be solved for the p_k^i .

5.5 Solution technique

The technique proposed for solving the above system of $\sum_{i=1}^{i=M} S_i + 1$ simultaneous nonlinear equations is a simple relaxation method based on successive substitution. The reasoning behind this choice is the fact that the state probabilities for each server, are functions of the state probabilities of the other servers by virtue of the dynamic routing feature, suggesting thereby an iterative approach.

This iteration is initiated by setting $p_0^i = 1$, and $p_k^i = 0$ for k=1,...,Si and for all servers, i=1,...,M. It is useful to consider equations (5.4) and (5.5) as consisting of M subsystems, of $S_i + 1$ equations each, for each server. Considering each subsystem (server) in turn, and starting from $k=1$, p_1^i is obtained. The process is repeated by finding the remaining p_k^i , from the following difference equation based on equation (5.4).

$$p_k^i = p_{k-1}^i \frac{\lambda_k^i}{\mu_i} \qquad k=1,\ ,S_i \tag{5.6}$$

In effect, this is a form which is simpler to calculate than (5.4). The values for p_k^i k=1,...,Si thus obtained must be normalised and are therefore substituted into the probability conservation equation (5.5). This will produce a value for p_0^i . All state probabilities for that server are then multiplied by this factor to obtain their normalized values. These are then substituted into the balance equations for the following server, and so on. The iteration is complete when all server probabilities have been evaluated.

At the end of each iteration mean queue lengths at each server are evaluated as:

$$L_i = \sum_{k=0}^{k=S_i} k p_k^i \qquad (5.7)$$

These mean queue lengths are added to find the average total number of customers (i.e. service requests) waiting for service in the central server system. The iterations are halted when, from one iteration to the next, this number remains constant up to its sixth decimal digit. This is somewhat arbitrary, but has been found to be a good compromise between solution speed and accuracy. In addition, at the end of each iteration, the normalised final states $p_{S_i}^i$ are examined. If these are greater than .000001 then an extra 10 states are added to that server for the next iteration. Again, this variable expansion of state space size was found extremely useful during experimental testing. It does not greatly compromise accuracy, and allows the state space to expand only up to the point necessary, without wasting calculations for near-empty states. The approximation technique was found to be simple to implement, and efficient to execute. In fact, it has been coded in compiled BASIC and runs on a microcomputer (Apple II). Run time for 10 iterations of a typical system is of the order of one minute. The amount of processing time increases with the number of serves (M), and the number of states per server (Si), the latter factor in turn, depending on the utilzation of the system.

5.6 Experimental Testing

A series of experiments has been conducted to test the performance of the approximation technique under a variety of system configurations. These were designed to be representative of a wide range of situations. The objective is to examine the important factors, and the extent to which they affect the accuracy of the proposed technique; it is also necessary to assess the potential of the method in a) analysing such systems and b) the possibility of using it to decide optimal file allocation patterns. Another property of interest is the speed of convergence of the technique, measured by the number of iterations necessary to arrive at a solution.

In order to obtain a common reference point for these comparisons it was decided to plot absolute, relative error versus system utilization. Utilization, however, is ill-defined in central server networks where service capabilities of servers are limited. Taking a worst-case example, it may be that some servers are overloaded while others remain idle, clearly an overloaded situation, which may however produce 'reasonable' values of utilization. If however, service capabilities are allocated in a not 'degenerate' manner, in the sense that they do not severely restrict the total service capacity of the system, then the definition proposed in [KLEI75] is adopted. In other words, utilization may be interpreted as the fraction of servers that are busy.

In practice, this was found to yield a satisfactory definition of utilization. By satisfactory, it is meant that utilization varies from Ø to 1 (as measured with the simulator), in a linear fashion, as the arrival rate is varied from Ø to $\sum_{i=1}^{i=M} \mu_i$. This last summation represents the total service capacity in the system. A linear regression equation was applied to the arrival rate and the measured utilization values. The parameter measuring the accuracy of fit, the coefficient of determination, is of the order of 95% for networks in which service capabilities were allocated in a 'reasonable' manner. For example, high capacity stations should not be restricted, while small capacity stations should not have allocated to them a large proportion of the requests. This allocation problem is made more complex by the provision of dynamic routing, and can only be approximately defined. In effect, of course, it is this service capability (file) allocation rule that is the final objective of this study and therefore, by definition, an approximation is necessary.

In order to simplify matters, therefore, utilization (ρ) is taken as the ratio between arrival rate and total service capacity, except where otherwise stated. The examples tested were exercised for levels of utilization up to approximately Ø.7 to Ø.8. The reason for this was that the number of states required in the approximation expands rapidly beyond these values thereby considerably increasing solution time, while longer simulation runs are necessary for the same levels of accuracy. In addition, it was later discovered that approximation accuracy deteriorates rapidly at utilization levels higher than these values.

5.6.1 Initial Results (Test 1)

As a first step in testing the method it was applied to eleven central server networks [TZAV83B]. These included 2-server examples, and larger 5- and 7-server examples. Various combinations of service rates and service capabilities were applied to the examples on test. These comparisons constituted a pilot run, and therefore precise details will not be presented. A brief summary of system characteristics is available in table 5.1. Examples 1 to 4, were compared with the exact values obtained from solving the system balance equations. The large number of equations formed (>30) forces us to limit the number of states to 6 per server in these four examples. The larger 5 and 7 server examples were tested against the output of a queuing network simulator. A flow chart and a description of this simulator is presented in Appendix 2. The result of interest is the relative error in average delay between the proposed approximation method and the reference results obtained through the balance equations or from the simulator.

The results of this first attempt at evaluating the usefulness of the technique are represented in Figure 5.3. The plot is of absolute relative error versus utilization. The error is the relative error between the approximation, and the reference, averaged over all 11 examples. The maximum and minimum values of this relative error are also plotted. It is evident that the method offers performance below the 5% average error level for utilizations up to Ø.5, while 1Ø% accuracy is maintained for utilizations of up to approximately Ø.7. The accuracy obtained is deemed acceptable in the sense that it is of the same order as other widely accepted queuing network approximation techniques [REIS74, CHAN75]. It is believed that the accuracy level is good enough to demand a further exploration of the technique.

The expected relationship between accuracy (error%) and utilization is evident in Figure 5.3. This is due to the increasing interaction between servers, as the number of arrivals increases. As this interaction becomes more intense, conditional state probabilities start to differ appreciably from their unconditional counterparts used in the approximation, and accuracy therefore deteriorates. This interaction is also a function of the number of options available for routing, i.e. the extent to which files have been replicated. The relative importance of these factors, i.e. utilization and proportion of replicated files, will therefore be investigated more thoroughly in the following sections.

5.6.2 Effect of number of servers (Test 2)

An important question regarding the performance of the approximation is whether it is affected by the size of the system (i.e. number of servers). To this end, 6 test examples have been designed (Table 5.2). These examples differed in the number of servers present in each central server system. The proportion of replicated files was fixed at 50% of the arriving requests, for all examples. The service rates of servers are different. The allocation of files is chosen to represent a 'reasonable' amount of load to each processor (of course this optimal allocation is one of the goals of this investigation, and therefore can only be approximated at this stage).

In Figure 5.4 the average, maximum and minimum errors in service delays for all examples are plotted. The similarity of this plot with that of Figure 5.3 is evident. In Figure 5.5, average delay as obtained from the simulator and from the approximation technique are plotted for each test system. The approximation seems to consistently underestimate average delay, a result also found in later experiments.

One question of importance at this stage is the significance of the number of servers and the utilization level on method accuracy. To this end, a two-way analysis of variance was conducted on the results of Test 2. As the proportion of replicated files is fixed to 50%, the factors tested here were the effect of the number of servers and utilization on method accuracy. The results of this analysis are available in Table 5.3. It was found that for a 99%

confidence level, the effect of the number of servers is insignificant, while the effect of system utilization is significant, a result perhaps obvious from Figure 5.5.

These results strongly suggest that the number of servers does not affect the accuracy of the method. As a consequence of these results therefore, this factor is removed from subsequent analyses by concentrating exclusively on two-server systems. A side advantage is that two-server systems can be approximated and simulated more efficiently, while a factor to be examined later -the proportion of replicated files- can be defined more precisely in two-server systems.

Before examining method accuracy further, it would be of interest to examine the number of iterations required by the approximation, as an approximate indication of computational time and approximation accuracy. Computational time will also depend on the number of equations that need to be solved in each iteration. This number of equations per iteration depends on the number of states and number of servers. Here, however, the interest is on the speed of convergence of the method itself, a factor measured by the number of iterations required to execute the approximation. It is believed that utilization affects the number of iterations required by a significant amount. For example, as utilization increases, the solution is further from the starting point ($p'_0=1$, $p'_k=0$, $k=1...Si$). In addition, as utilization increases, more adaptive state expansion is required during the solution. To test these effects, the number of iterations required to approximate the system(s) of Test 2 were also measured.

The results are plotted in Figure 5.6. The symbols identify the number of servers in the experimental system, and the graph shows the number of iterations required, for each utilization level, for different numbers of servers. From a visual examination of Figure 5.6 it seems that the effect of utilization is much stronger than that of the number of servers. This is tested as in the previous case by means of an analysis of variance whose results are presented in Table 5.4. The results indicate that in fact, the number of iterations required to execute the approximation depends on both the number of servers and the level of utilization, utilization being the more significant factor.

Summarizing, the accuracy of the approximation technique does not appear to depend on the number of servers in the system. As a result, in the further investigations into method accuracy, two-server networks only were analysed, these being more efficient to analyse and simulate. The number of iterations required by the method depends both on utilization and the number of servers, the former having a more significant effect.

5.6.3 Effect of degree of file replication and utilization (Test 3)

In this section the results of an investigation into the effects of utilization and the proportion of replicated files, on the accuracy of the proposed approximation technique are presented.

The system on test is a two-server, with service rates μ_1=1.5 and μ_2=0.8. It is assumed that three files are stored in this system, with access probabilities of 0.5, 0.25, and 0.25 respectively. By varying the allocation of files between servers, the proportion of requests that have two routing options is varied from 0% to 100% in five steps, the arrangements chosen being shown in Figure 5.8.

The levels of arrival rates were chosen to coincide with utilization levels of 0.1, 0.3, 0.5, and 0.7. There exist however 5 system configurations corresponding to each level of file replication, and for each of these, utilization may be different for the same arrival rate, due to the definition chosen for utilization. This definition measures the average proportion of active servers, and can therefore be affected by a particular allocation. In order to reduce these effects, utilization values obtained from the simulator were averaged over all system configurations used, and this averaged utilization plotted against the corresponding arrival rates as in Figure 5.9. A very linear relationship is evident, with a linear regression coefficient of determination of .95. The desired values of utilization (i.e. .1, .3, .5, .7) were plotted against this linear relationship to obtain a set of new values for the arrival rates, and these were then used as inputs for the simulator and the approximation.

This very linear relationship indicates that the approximation used for utilization, i.e. $\rho = \lambda / \sum_{i=1}^{i=M} \mu_i$ is a good one for systems with 'reasonable' file allocations, agreeing well with the definition proposed in [KLEI75], i.e. average fraction of busy servers, at least for the examples tested.

Method accuracy is measured as the percentage difference in average system delay as obtained by the approximation and the simulator. Each comparison between the approximation and the simulator was repeated 5 times with a different random number seed for the simulator. This percent error is averaged over these 5 repetitions and its absolute value is plotted against utilization, and the proportion of replicated files, as in Figure 5.10. It is obvious that this error depends both on utilization and the proportion of replicated files. Note that the 0% replicated files curve, corresponds to the case where the approximation technique produces exact results (since queues are now independent); the residual error is therefore due to the simulator itself. This increases slightly with utilization as a greater number of runs are required for the simulator for higher utilizations, to maintain the same level of accuracy.

In order to further test the factors affecting method accuracy, a two-way analysis of variance with interaction was carried out, and is presented in Table 5.5. As expected, the effect of utilization and the proportion of replicated files have a significant effect. In addition, the interaction between these two factors is found to be significant.

To test the significance of these factors, and in particular the effect of the interaction, a regression equation based on a linear and a product form is applied to the data plotted in Figure 5.10. The coefficients of this regression fit are available in Table 5.5. A test of significance of the four coefficients in this regression equation indicates that the constant term, and the utilization term on its own, are not statistically significant. This would indicate that method accuracy is mainly dependent on the proportion of replicated files, and the product of this term with utilization. The coefficient of determination of this regression fit (a measure of the accuracy of the fit) is approximately 90%. This is considered a high value, and allows us to suggest that there exist no other major sources of variation other than statistical errors.

The number of iterations required to find an approximate solution were also measured and have been plotted in Figure 5.7. The two-way analysis of variance presented in Table 5.6, indicates that the number of iterations depends on both utilization and the proportion of replicated files.

Acknowledgement

I would like to thank Dr. E. Frangoulis for his help and advice with the statistical analysis of these results.

5.7 Applications to file allocation and replication

Having assessed the general performance of the proposed analysis technique, the interest now focusses on its potential as an analysis tool in deciding the optimum allocation and/or replication of files within a dynamic routing central sever network.

To this end, the following results must be compared:

a) using the approximation technique to find this optimal allocation, and

b) using a reference method to find this allocation

Unfortunately, as has been made clear in chapter 4, no efficient techniques exist currently for analysing such general networks and therefore, as a consequence the use of simulation is the only alternative.

The use of a simulator to obtain an optimal file allocation introduces additional problems, as it cannot easily be used from within an optimization technique. The use of a brute-force approach is therefore inevitable. In other words, the reference optimal file allocation procedure enumerates all feasible solutions (file allocations), applying the simulator to each in turn, and deciding which offers the best performance. This file allocation is represented by a matrix of binary variables Xij. That is, Xij is 1 if server i stores file j, and 0 otherwise. The number of variables therefore is the number of files (F), times the number of servers. The possible number of combinations of this matrix is therefore $2^{M \times F}$, potentially a very large number. Constraints exist however. These take the form:

$$\sum_{j} l_j x_{ij} \leq c_i \qquad i=1, \ldots, M \tag{5.8}$$

where l_j, is the length (in bits) of file j, while Ci is the available storage capacity in server i.

In addition, all files must be allocated at least once. Therefore

$$\sum_{i} x_{ij} \geq 1 \qquad j=1,..,F \tag{5.9}$$

The fact that a complete enumeration of the solution space is unavoidable, combined with the inefficiency of the reference method used (i.e. simulation), forces us by necessity to use examples of smaller dimension, the expectation being that the results obtained will be equally applicable to larger scale cases.

The procedure for executing this comparison in order to test the potential of the Birth-Death approximation in executing optimal file allocations is therefore as follows. An example system is proposed, in terms of service rates, number of files, query rates/lengths of files and storage capacities. The allocation matrix consisting of variables Xij is then cycled through all possible combinations. Each combination is checked against constraints (5.8) and (5.9) and, if these are satisfied, then this combination is analysed by both the approximation technique, and through simulation. The results, in terms of average system delay are noted, and the procedure repeated until all feasible combinations of the allocation matrix have been exhausted.

Values for delay corresponding to each allocation are then searched, and the lowest found is chosen as corresponding to the optimal file allocation pattern. These two allocation patterns, i.e. as obtained through the approximation and through the simulator are then compared. Ideally they should be identical. This procedure is also repeated for different values of the arrival rate.

The examples chosen consist of 2- and 3-server systems with different service rates and different file storage capacities (this affects the number of feasible allocation patterns possible). Their characteristics are briefly outlined in Table 5.7. The results of this investigation are plotted in Figure 5.11. The plotted values are explained as follows: For each value of the arrival rate, two optimal file allocations among the set of feasible file allocations have been chosen. One is that chosen using the approximation, and the other using the simulator, both based on a minimum average delay criterion. The values of delay corresponding to these two allocations are then plotted against the arrival rate. The value of delay plotted for the allocation <u>chosen</u> by the Birth-Death approximation is however, that <u>evaluated</u> with the simulator for that allocation pattern. This is done in order to maintain consistency in these plots by presenting delay values for different allocations, evaluated with the same technique. In this study however, the objective is not the actual delay values obtained, but on whether the allocation obtained with the approximation technique coincides with that obtained using the reference (simulation) technique.

The coincidence in the majority (approximately 74%) of the two sets of file allocation patterns chosen is evident from Figure 5.11. In those cases where the two allocation patterns do not coincide, the difference is thought to be due to the statistical variation in the simulator, as it is obvious that this variation is small in all cases.

The optimal file allocation pattern obtained with the approximation technique was found to remain constant through the range of arrival rates for each example, indicating an independence on the arrival rate factor. This would indicate that an analysis based on small arrival rates would produce an allocation pattern that remains optimal for larger values of the arrival rate.

In addition, the approximation technique performs as expected, i.e. it responds to the variations in file allocation patterns, producing lower delays with every increase in the number of file copies stored in the system. These tables of delay as a function of allocation pattern and arrival rate are too large to present here, but the matching of the optimal file allocation patterns obtained with the reference analysis method (simulation), and the approximation technique offers evidence for this point.

The actual choice of a file allocation pattern would probably be carried out by an integer programming technique employing an analysis routine to evaluate and compare alternatives. In this respect, the following characteristics of the approximate analysis technique presented in this chapter are noted:

1) average system delay is monotonic with the number of file copies allocated. In other words, more file allocations always mean a lower system delay.
2) constraint functions are monotonic.
3) the optimal file allocation pattern appears to be invariant with increasing arrival rate.

These characteristics make the new approximate analysis technique suited to available Ø-1 integer programming techniques that do not require an analytic objective function [LAWL66]. Such techniques take advantage of the monotonic nature of the objective and constraint functions to bypass suboptimal file allocation patterns, thereby limiting the number of patterns that must be searched.

The above discussions have of necessity been limited to small dimensionality allocation patterns. Example applications to larger, more realistic examples are possible. The results however would be meaningless as there is a lack of a reference, comparison method. It is nevertheless expected that the use of the approximation in this aspect of the problem, can be generalized succesfully to larger problems with more files and servers.

5.8 Convergence of Solution

An equally important question that remains to be addressed is that of whether the iterative technique proposed for solving the set of non-linear equations (5.4,5.5) converges. The general absence of any underlying theory has been a major stumbling block in the widespread acceptance of approximation techniques based on iteration [CHAN78]. A side result would be to ascertain how quickly (in terms of the number of iterations) this convergence takes place.

The solution of the set of non-linear equations consists of the state probabilities p_k^i. These, by definition, lie in the range [0,1]. If, therefore, it can be shown that these state probabilities change monotonically during the iterations, then a solution must be reached. However, we cannot guarantee that this will take place in a finite number of iterations. Nevertheless, monotonicity if proven, will ensure the absence of oscillations, a potential problem with techniques such as Newton iteration.

Unfortunately, the number and generality of the proposed state balance equations prohibits an examination of the general case. A small example is therefore chosen, consisting of two servers and 3 states per server. The analysis therefore is not a general one, and is expected to be different for different systems, with possibly different results.

The details of this analysis are presented in Appendix 3. The results indicate that all six state probabilities -in the chosen example- vary monotonically as the iterative solution proceeds. An examination of the rate of convergence has not been carried out as it would not apply to the general case.

For the small example chosen, therefore, it has been shown that oscillation of the solution technique is not possible, and that a solution will eventually be obtained (as probabilities are bounded), though not necessarily in a finite number of steps.

In practice, an average of 10 iterations were found to be adequate for a stopping criterion of six stable decimal digits in average system delay. Typical values of the number of iterations required are plotted in Figures 5.6 and 5.7, for a variety of system configurations.

Overall, approximately 1500 systems were analysed with the proposed birth-death approximation technique, for a wide variety of system parameters. These parameters are the number of servers, the service rates, file allocation, and arrival rate. In all cases, a stable solution was obtained for less than 25 iterations. The expected exceptions are unstable situations where at least one server is loaded with a total arrival rate higher than its mean service rate, an impractical case.

A plot of the rate of convergence of the method with iterations is shown in Figure 5.12 for the 50% replicated files system of Figure 5.8, representing a typical case. The error is (absolute) relative error at each iteration as compared with the result obtained with the last iteration. The stopping criterion is a variation of less than .000001 (absolute) from one iteration to the next. The plot is repeated for different levels of utilization, but as indicated in the previous sections, the number of iterations appears to depend on other factors as well.

The starting point in all examples was an arbitrary one, i.e. $p_0^i=1$, $p_k^i=0$, $i=1,...,M$, $k=1,...,Si$. To test the stability of the technique, random starting points were chosen for a set of example systems. In all cases convergence was confirmed, demonstrating the robustness of the technique.

In common with other iterative techniques, it was observed that the number of iterations taken to solution may be reduced, by estimating better starting points for the state probabilities.

5.9 Extensions to the approximation

In this section a set of extensions to the proposed approximation technique is suggested. The objective is to suggest means by which method accuracy may be improved, and its range of applications extended.

5.9.1 Load dependent service rates

The fact that a set of Birth-Death processes is used to model a central server system allows us to easily include load (state) dependent service rates for each server, i.e. the service rates of servers may increase as queue sizes increase. This is simply done by substituting μ'_k in place of μ_1 in expressions (5.3) to (5.5). This load-dependent service rate model has been used to model some types of servers in computer systems, and as such offers a useful extension to the method. This suggestion however has not been implemented or tested.

5.9.2 Closed queuing networks

The proposed approximation has been mainly intended for modelling open queuing networks. However, it is possible to extend the technique to closed queuing networks of the type shown in Figure 4.4. Briefly, this can be done by asssuming that the output of each server i constitutes a Poisson process with mean rate P(server i is busy)xμ_1. This process is added to any other output process that may exist in parallel (by virtue of the Poisson assumption). This sum can then be considered as an arrival process fed back to the input, or to another set of servers. This suggestion can in fact potentially be applied to any open, closed or mixed queuing network, by assuming that the outputs of servers can be approximated as Poisson processes that can be added, or split, in both a probabilistic and adaptive routing manner.

The as above proposal introduces an additional approximation, as the output of a dynamic routing central server network is no longer a Poisson process [BOOR81] as in the case of M/M/1 systems [BURK56], since the input process can no longer be considered as Poisson [CHOW77].

This extension has been briefly tested on systems of the type shown in Figure 4.4, and found to produce adequately accurate results. The closed nature of the network did not affect the convergence of the system of equations (5.4) to (5.5), but the tests conducted are too few to allow general comments to be made.

5.9.3 Improvements in accuracy

In this section a suggestion is proposed to correct one of the sources of error in the approximation. This error, as already mentioned, is due to the neglect of conditional probabilities in the approximation. Referring to Figures 5.1 and 5.2 as a simple example, the following can be observed: the conditional probability that server 2 is in say, state 2, given that server 1 is also in state 2, is higher than the corresponding unconditional probability of server 2 being in state 2, due to the self-balancing action of the system. In other words it is more probable that the two servers occupy the same states rather than random ones. This correction is not simple to apply, as it is complicated by the effects of different service rates, the file allocation pattern, and the arrival rate. However if a good heuristic rule can be found for introducing these corrections to the state probabilities then, potentially, it may improve the accuracy of the method significantly.

5.9.4 General dynamic routing disciplines

In terms of the diversity in modeling applications, the birth-death approximation can be extended to analyse other routing disciplines. For example, the Join-Shorter-Queue discipline, is a simple extension of expression (5.3) where all μ_i are set to 1.

The way in which ties are broken in the suggested approximation is by routing to the server with the lower index. This is an arbitrary choice, chosen to simplify the presentation of expression (5.3). In fact, other arbitration techniques can be handled, such as equiprobable routing to servers involved in a tie. The modification of expression (5.3) to handle this case, although straightforward conceptually, is rather complex to present and explain, and will therefore not be presented here. The modification has however been briefly tested on a small system with good initial results.

Another adaptive routing rule, the Join Biased Queue (JBQ) suggested in [YUM81] can also be modeled. In this case, for two equal servers, a request is sent to a particular server if that server has a queue length equal to that of the other server minus an amount D. This parameter D has been introduced as a simple technique for balancing flows in these two servers when it is known that there exist dedicated flows to these two servers (i.e. arrival streams that have only one routing option). This adaptive routing rule can be introduced in expression (5.3) in a straightforward manner. The state dependent arrival rates λ_k^i are formed by adding the rates for the dedicated flows to the adaptively routed flows, formed in a manner similar to that of expression (5.3). Details will not be given here, as this modification has not been tested. In any case, the JBQ rule is a subset of the study presented in this chapter, i.e. that of analysing the performance of systems with restricted service capabilities (or, equivalently, dedicated traffic flows).

Another potentially useful extension is the ability to model queuing systems in which it is possible for jobs (service requests) to be transferred from one queue to another, once they have been routed to a server. Job transfer takes place when it is decided that by transferring a job from one queue to the end of another, its estimated service time can be improved. This would be done in cases where, due to the statistical nature of the service time processes, one server happens to reduce its load significantly as compared to other servers in the system, thereby offering spare service capacity. Such queuing systems are difficult to analyse, particularly when combined with a dynamic routing protocol. However, for the simplified case of equal servers and full service capabilities, a queue jockeying central server network is modelled as follows:

(continued next page)

$$\lambda_k^i = \lambda \prod_{\substack{j=1 \\ j \neq i}}^{j=M} \left[\sum_{r=t_{j,i}}^{r=S_j} p_r^j \right] + \sum_{\substack{j \neq i \\ j=1}}^{j=M} \left[\beta_{ji} \sum_{r=k+2}^{r=S_j} p_r^j \right] \qquad i = 1, \ldots, M \quad k = 0, \ldots, S_i - 1$$

$$\mu_i^k = \mu + \sum_{\substack{j \neq i \\ j=1}}^{j=M} \left[\beta_{ij} \sum_{r=0}^{r=k-2} p_r^j \right] \qquad (5.10)$$

where

$$t_{j,i} = \min \left\{ \xi \in \{0, \ldots, S_j\} \quad \begin{array}{ll} \text{either } \dfrac{\xi+1}{\mu} > \dfrac{k+1}{\mu} & \text{if } i > j \\ \text{or } \dfrac{\xi+1}{\mu} \geq \dfrac{k+1}{\mu} & \text{if } i < j \end{array} \right\}$$

β_{ij} is the rate with which jobs are transferred from server i to server j. Jobs are transferred if one queue becomes shorter than another by two jobs. If jobs cannot be transferred from one server to the other, then the relevant β_{ij} entry is zero. The case for different service rates and restricted service capabilities is an extension of the above expression. These will not be presented here as they have not been tested, being intended primarily as suggested extensions to the proposed techniques.

Extensions to cover the cases of non-Poisson arrivals and non-exponential service times have not been investigated, the method of stages [SAUE81, KLEI75] being a possible approach.

5.10 Conclusions

In this chapter, a new approximation technique for the analysis of open, central server, dynamic routing networks has been proposed and evaluated. Assumptions include Poisson arrivals and exponential service rates. The approximation method is based on separating the queues of each server, and on treating them as independent birth-death processes with state dependent arrival rates. As such, the method may be approximately classified under the category of decomposition techniques [CHAN78].

The state balance equations associated with these processes form a rather large system of simultaneous non-linear equations. It has been suggested that this system can be solved iteratively, considerably simplifying the solution. This iterative process is based on relaxation, and has been found to converge in a small number of iterations for a large number of examples with a wide variety of input conditions and starting points. A more detailed investigation of convergence for a simple example system confirms that the method will not oscillate.

Overall, it was found that the method, on average, is adequately accurate (average error<10%) for small to medium levels of utilization (0 to 0.7). As such, it can be considered as complementary to the heavy traffic diffusion approximation, while its performance is of the order of, and better than, that considered as typical of other widely accepted queuing network approximation methods [REIS74, CHAN75]. It has been found as simple to implement and executes efficiently on a microcomputer. It will model systems of any number of servers, range of service rates, and service capabilities.

Accuracy is independent of the number of servers and their service rates. Accuracy does however, depend on system utilization, and the proportion of requests that are routed dynamically.

The method has been tested on a mimimum expected response delay routing protocol, but it is believed that other routing disciplines can also be modeled, e.g. join shorter queue, join biased queue, and queue jockeying. This generality is an indication of the inherent flexibility of the technique.

This approximation method has also been used as the analysis component of a study into the optimal allocation and replication of files in a central file server system. The optimal replication of files in a central server system is a problem that as far as is known has not been previously investigated. The performance of the proposed approximate analysis technique is very satisfactory in this application, the optimal allocation patterns obtained with it agreeing, in the majority of cases, with the file allocation patterns found with a reference method. In the cases where no agreement between the two techniques was obtained, the discrepancy is thought to be due to the statistical fluctuation of the simulator used as the reference technique.

The optimal allocation patterns chosen did not change with increasing utilization. This consistency in performance and other factors make the approximate technique suited to available integer programming methods suited for handling larger problems.

Unfortunately, and due to the absence of other efficient, general, approaches for analysing these types of network, it has been necessary to use simulation as the reference method, restricting these example applications to small dimensionality examples. Nevertheless, the consistent performance of the method in the file allocation context prompts us to extrapolate that its use in higher dimensionality cases would be equally successful.

Concluding, and in the light of the review of chapter 4, it may be said that the proposed technique is unique in being able to approximately model a dynamic routing network consisting of any number of servers of different service rates, and limited service capabilities. It is simple to implement and execute, offering an adequate level of accuracy and has, it is believed, a useful potential for obtaining an optimal file allocation/replication pattern in a central file server system.

Table 5.1 Parameters in Test 1. Eleven test examples

Example	No. of servers	Equal service rates	Service capabilities
1 to 2	2	YES	FULL
2 to 4	2	YES	PARTIAL
5 to 7	5	YES	PARTIAL
8 to 9	7	YES	PARTIAL
1Ø to 11	7	NO	PARTIAL

Note: Examples 1-2, 2-4, 5-7, 8-9, 1Ø-11, differ in terms of the system parameters, e.g. file allocation pattern and service rates.

Table 5.2 Parameters in Test 2

Example	No. of servers	Service rates	%error at ρ=Ø.5
1	2	.8 1.5	6.1
2	3	.6 1.1 1.6	7.22
3	4	.6 1.1 1.6 2.1	1.55
4	5	.6 1.1 1.6 2.1 2.6	6.11
5	6	.5 1.1 1.6 2.1 2.51 3.Ø1	3.Ø3
6	7	.6 1.1 1.6 2.1 2.6 3.1 3.6	4.6

Table 5.3 Analysis of variance. Error (%) in Test-2

Source	Sum of squares	degrees of freedom	Mean squares	F-ratio
Number of servers	27.8	5	5.56	1.264
Utilization	19Ø.3	3	63.43	14.42
Residual	65.9	15	4.4	

$F_{01, 5, 15}$ =4.56
$F_{01, 3, 15}$ =5.42

Therefore, the effect of the number of servers is not significant, while the effect of utilization is significant.

Table 5.4 Analysis of variance. Number of iterations in Test-2

Source	Sum of squares	degrees of freedom	Mean squares	F-ratio
Number of servers	30.72	5	6.14	12.93
Utilization	349.125	3	116.4	245
Residual	7.125	15	.475	

$F_{01,5,15}$ =4.56
$F_{01,3,15}$ =5.42

Therefore, the effect of both the number of servers and utilization affects significantly the number of iterations required to approximate the system.

Table 5.5 Analysis of variance with interaction. Error (%) in Test-3. Number of repetitions=5.

Source	Sum of squares	degrees of freedom	Mean square	F-ratio
Utilization	1906.05	3	635.35	177.5
proportion replicated files	1726.472	4	431.62	120.56
Interaction	1435.125	12	119.6	33.4
Residual	286.73	80	3.58	
Total	5354.385	99		

$F_{01,3,80}$ =4
$F_{01,4,80}$ =3.5
$F_{01,12,80}$ =2.4

Therefore, utilization, and proportion replicated files are significant factors, while the interaction between these two factors is also significant at the 99% confidence level.

If we let x=proportion replicated files, and y=utilization, then a regression fit of the equation

$$error = a+bx+cy+dxy$$

can be attempted. The inclusion of the factor xy is by virtue of the significant interaction between x and y. The parameters a,b,c,d obtained are as follows:

a=-.8 b=.Ø69 c=4.15 d=-.46

The coefficient of determination of this regressional fit (a measure of the accuracy of fit) is 9Ø%.

An analysis to test the significance of these coefficients by obtaining their t-ratios, is executed.

Coefficient	t-ratio
a	-.59
b	3.11
c	1.4Ø
d	-9.56

degrees of freedom=15

$t_{01,15}$ =2.6

This would indicate that the significant factors (99% confidence level) are the proportion of replicated files and the product term.

Table 5.6 Analysis of variance. Number of iterations in Test-3

Source	Sum of squares	degrees of freedom	Mean square	F-ratio
Utilization	265	3	88.3	25.9
proportion replicated files	138.2	4	34.55	1Ø.13
Residual	41	12	3.41	
Total		19		

$F_{01,3,12}$ =5.95
$F_{01,4,12}$ =5.41

Therefore both effects are significant, i.e. utilization and proportion of replicated files in affecting the number of iterations required by the example being tested.

Table 5.7 System parameters of test in section 5.7 of the use of the approximation technique in optimal file allocation problems

Example	number of servers	number of feasible patterns	arrival rates tested	matched patterns
1	2	12	6	5
2	2	29	6	3
3	2	22	6	4
4	3	44	1Ø	6
5	3	19	1Ø	1Ø

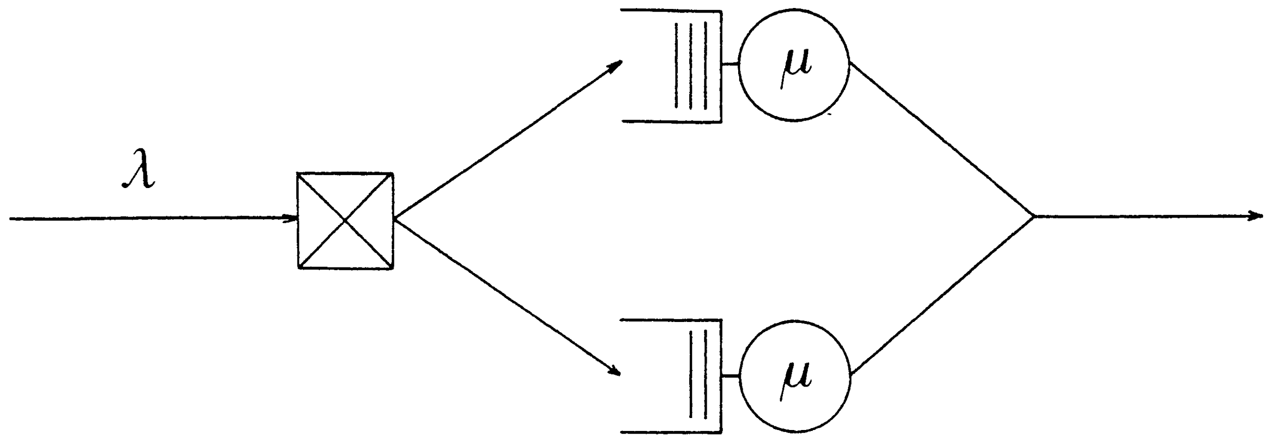


Figure 5.1 Two equal servers

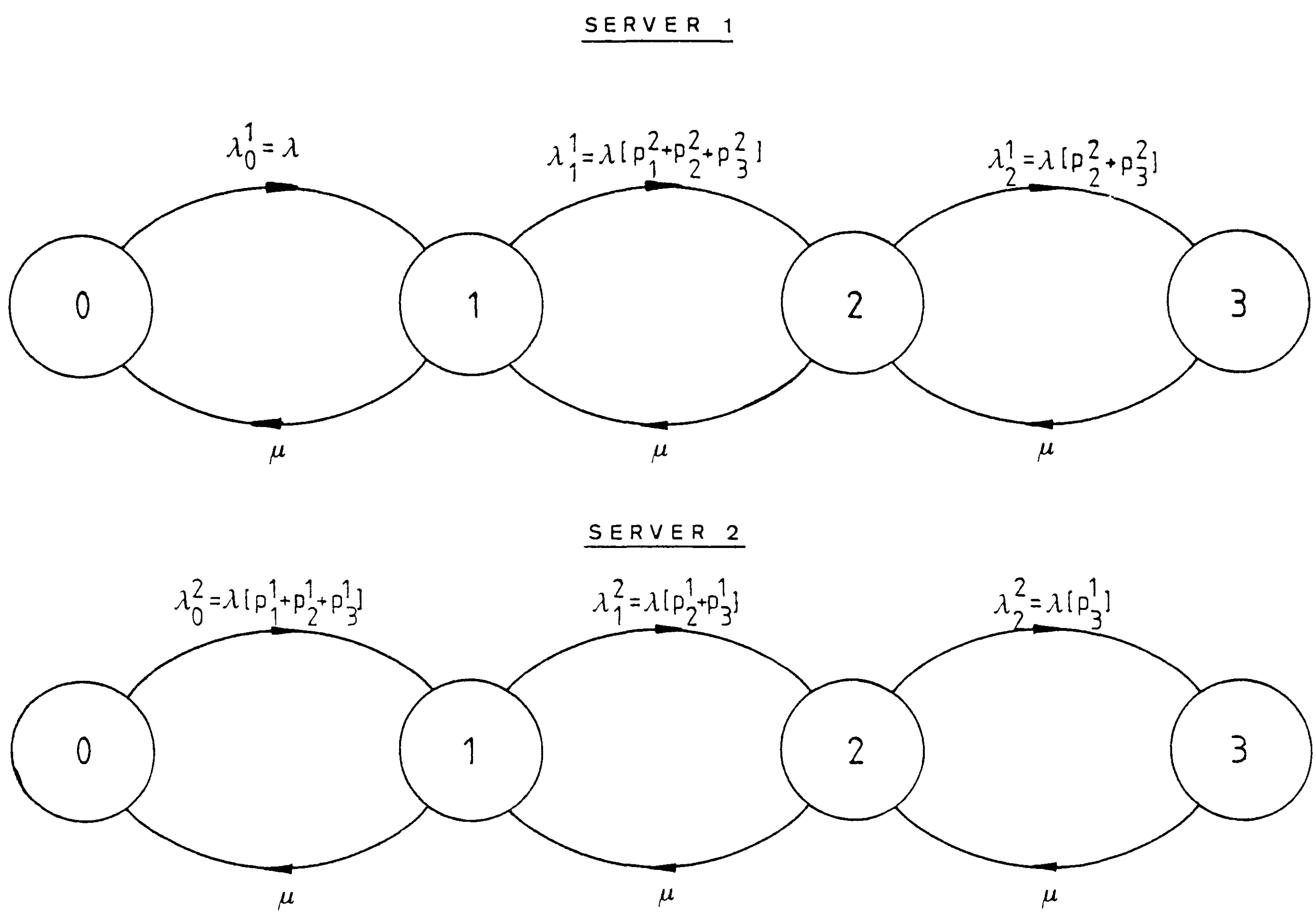


Figure 5.2 Birth - Death approximation

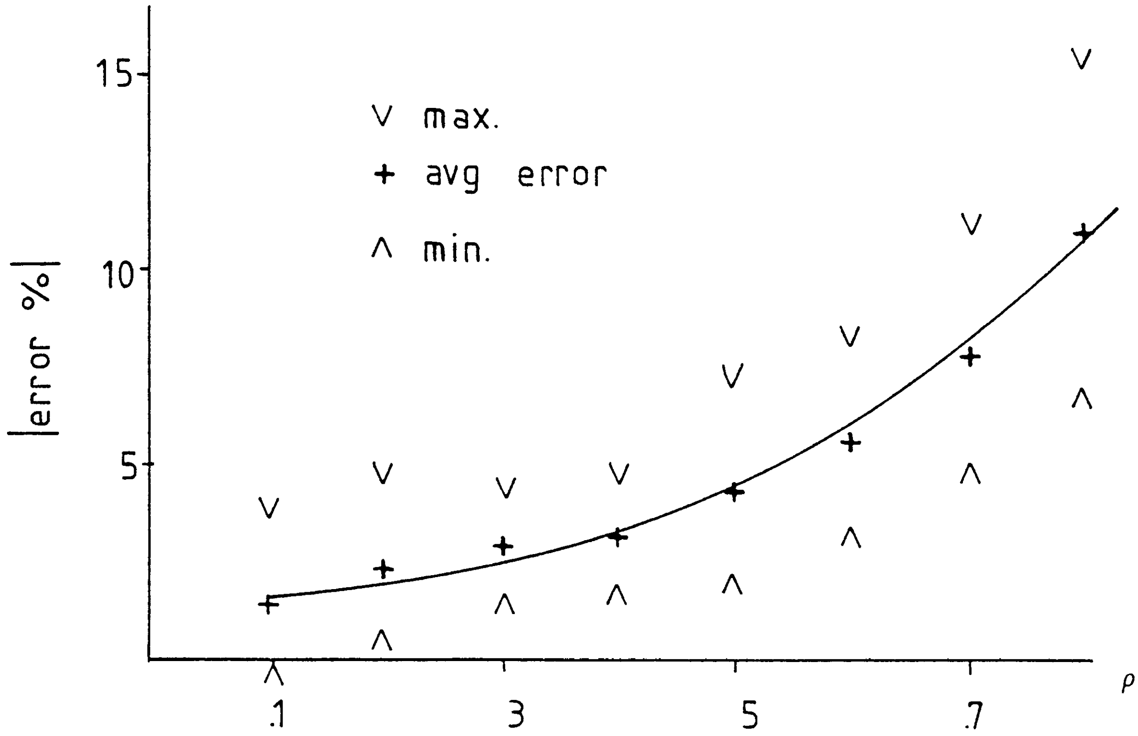


Figure 5.3 Error versus utilization (Test 1)

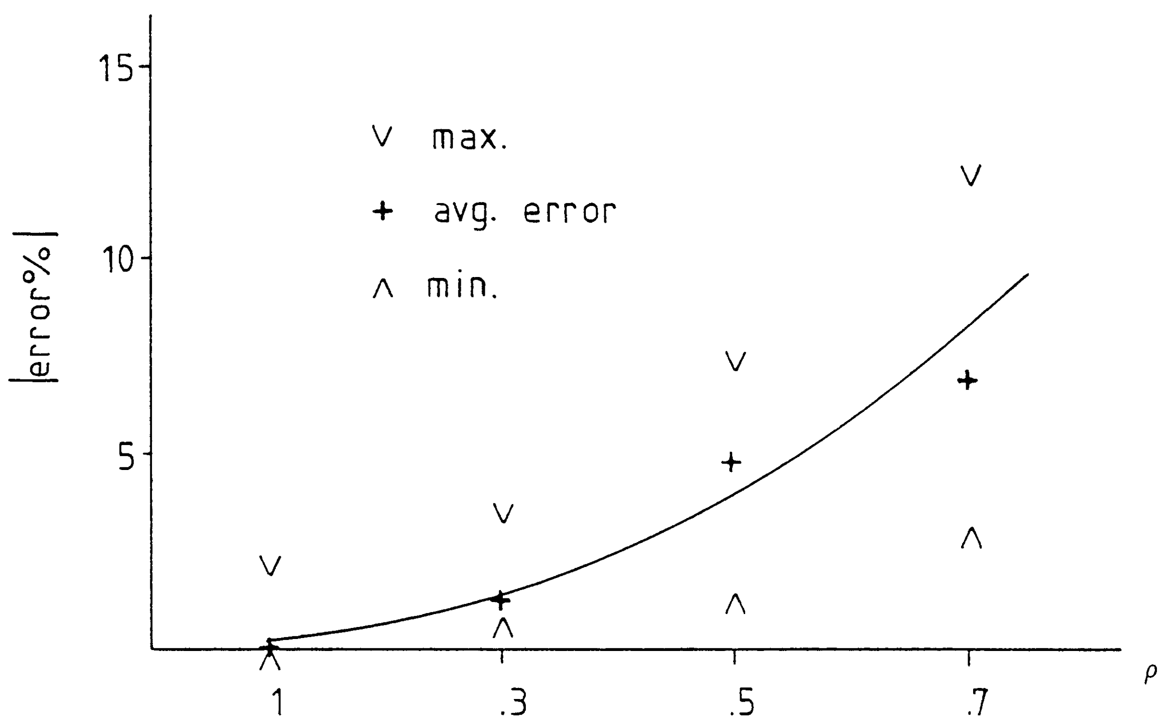


Figure 5.4 Error versus utilization (Test 2)

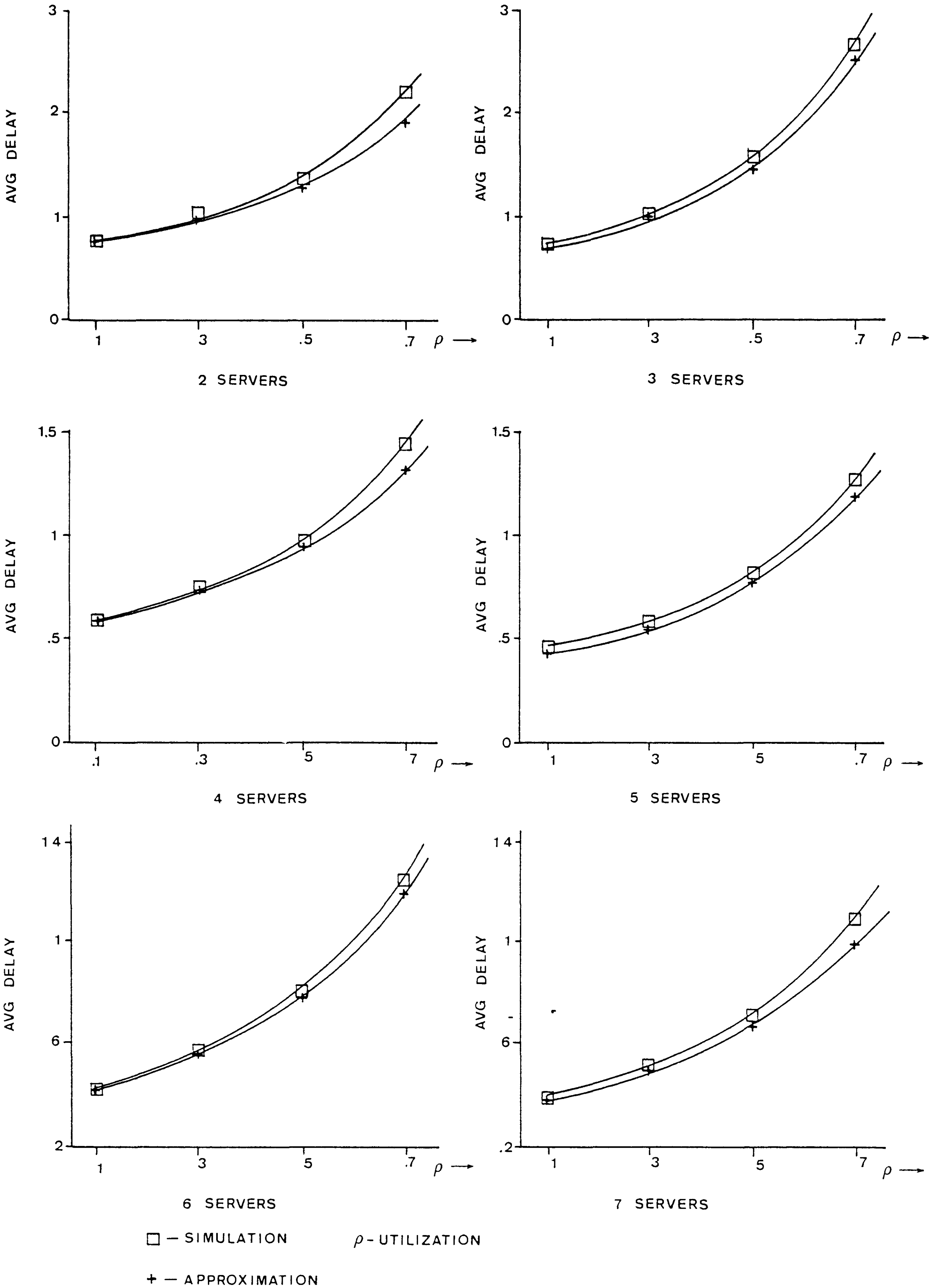


Figure 5 5 Avg delay obtained from simulation and the approximation, for different no of servers (Test 2)

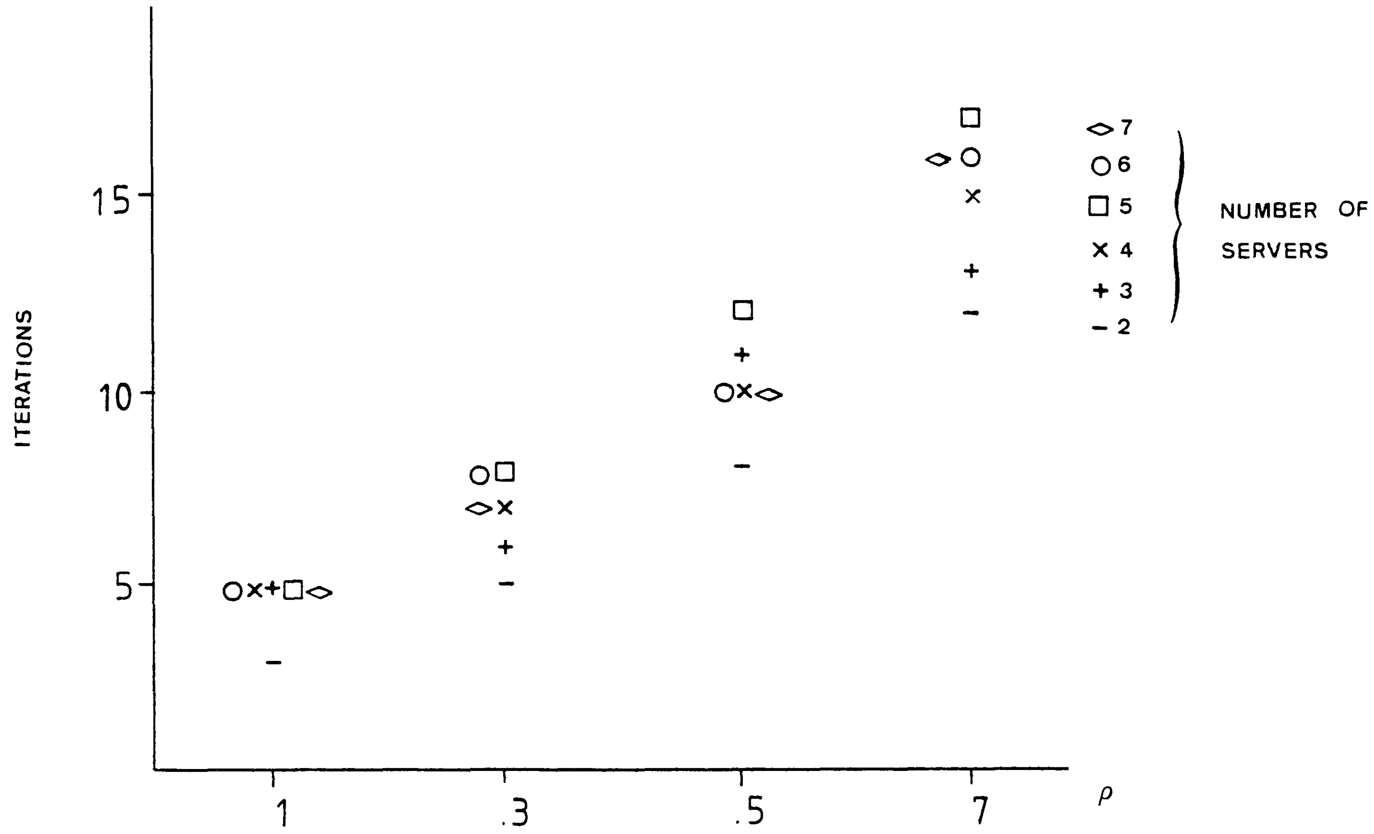


Figure 5.6 No. of iterations as a function of utilization (ρ) and number of servers (Test 2)

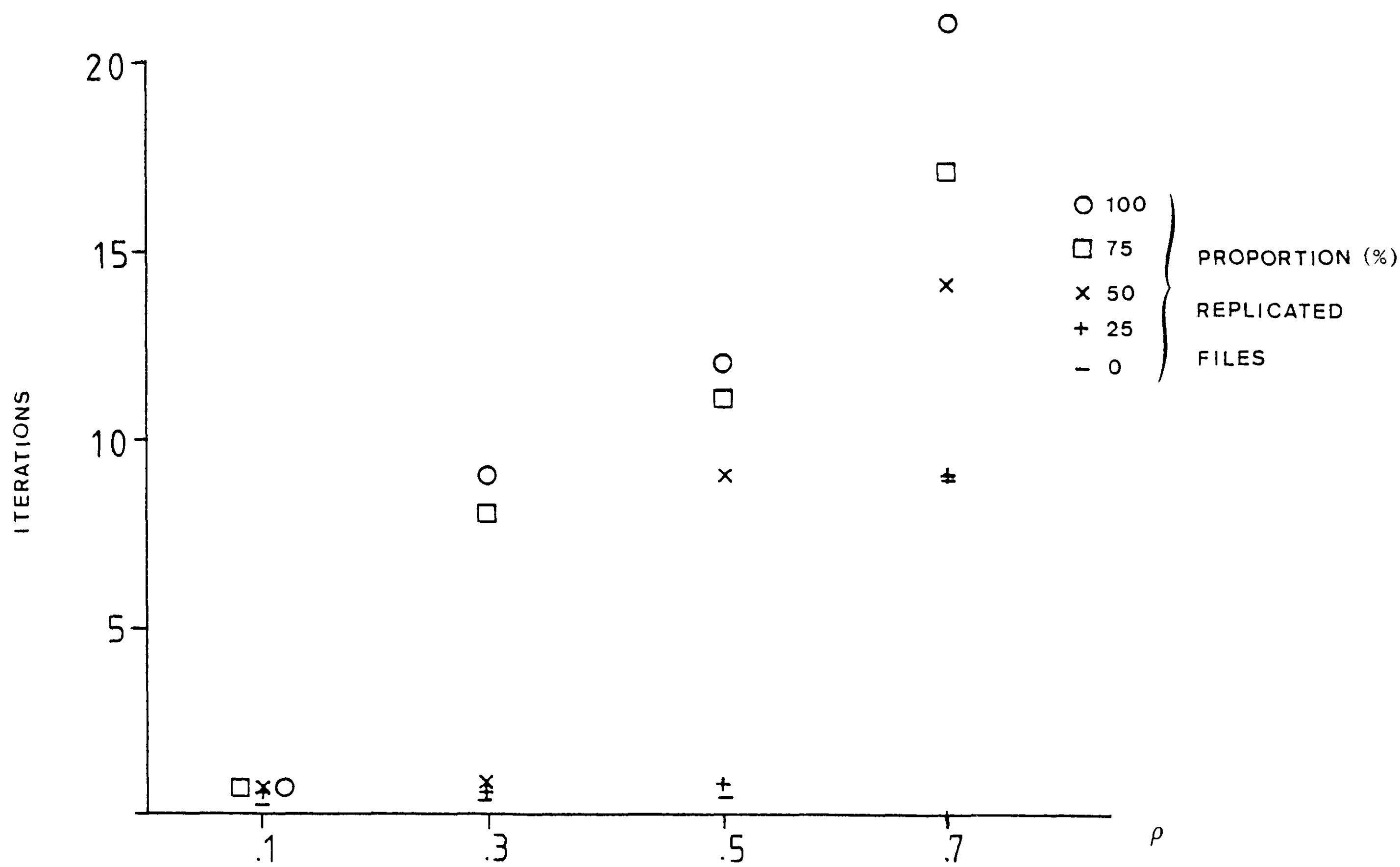


Figure 5.7 No. of iterations as a function of utilization (ρ) and proportion of replicated files (Test 3)

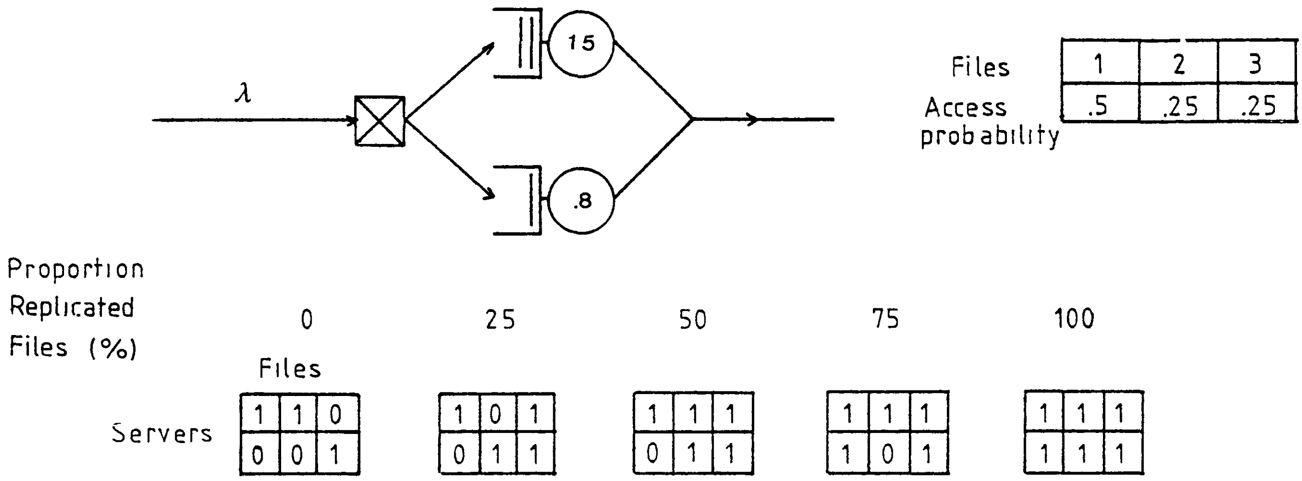


Figure 5.8 Set-up of Test 3

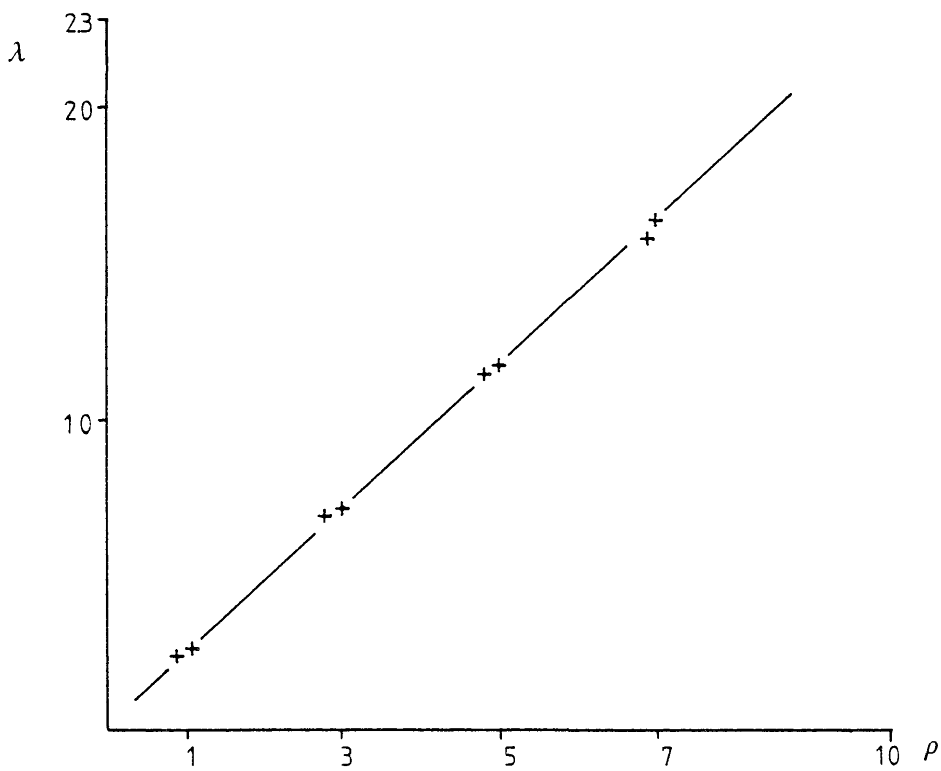


Figure 5.9 Relationship between arrival rate λ and utilization ρ (Test 3)

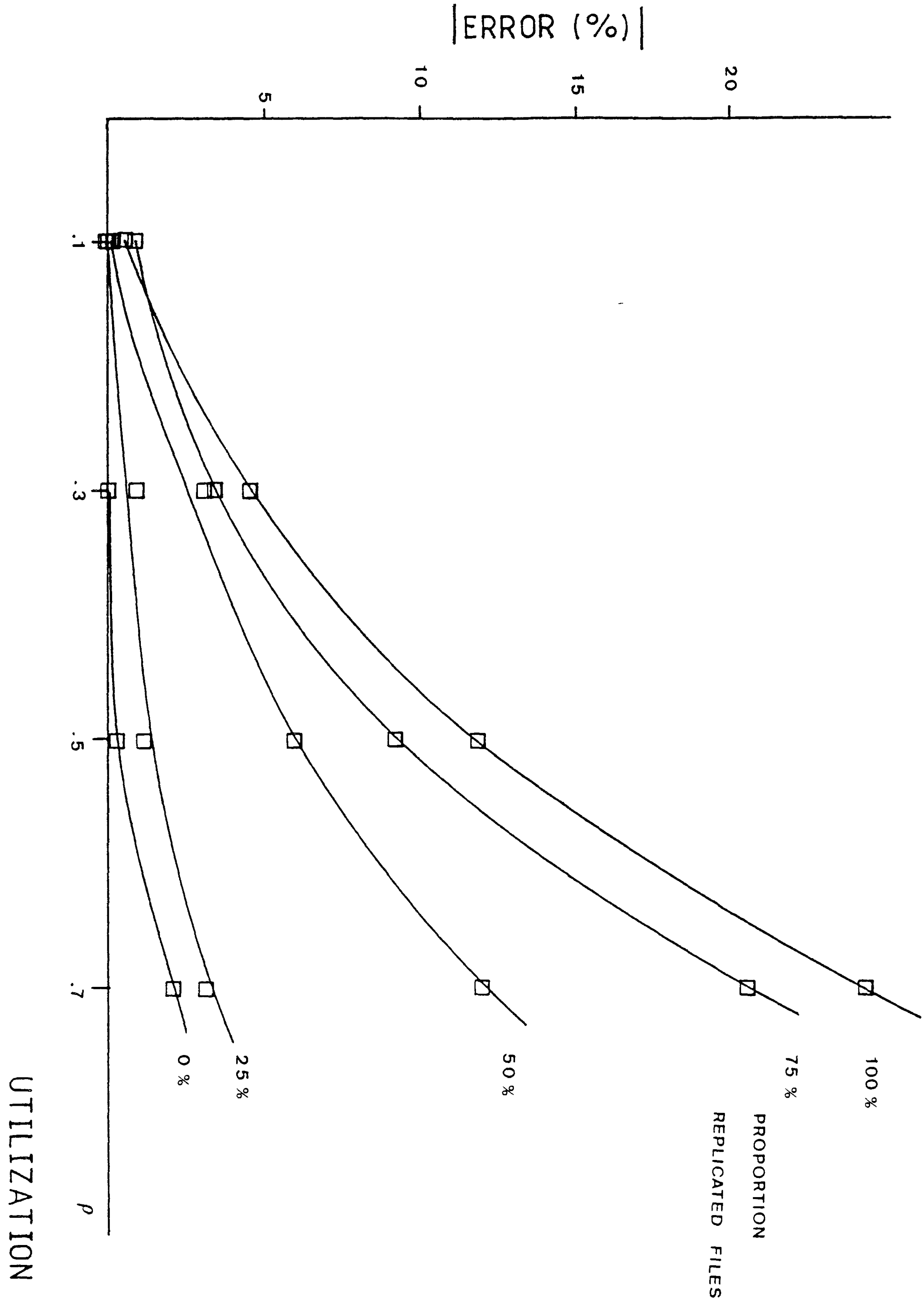


Figure 5.10 Error (%) as a function of utilization and proportion replicated files (Test3)

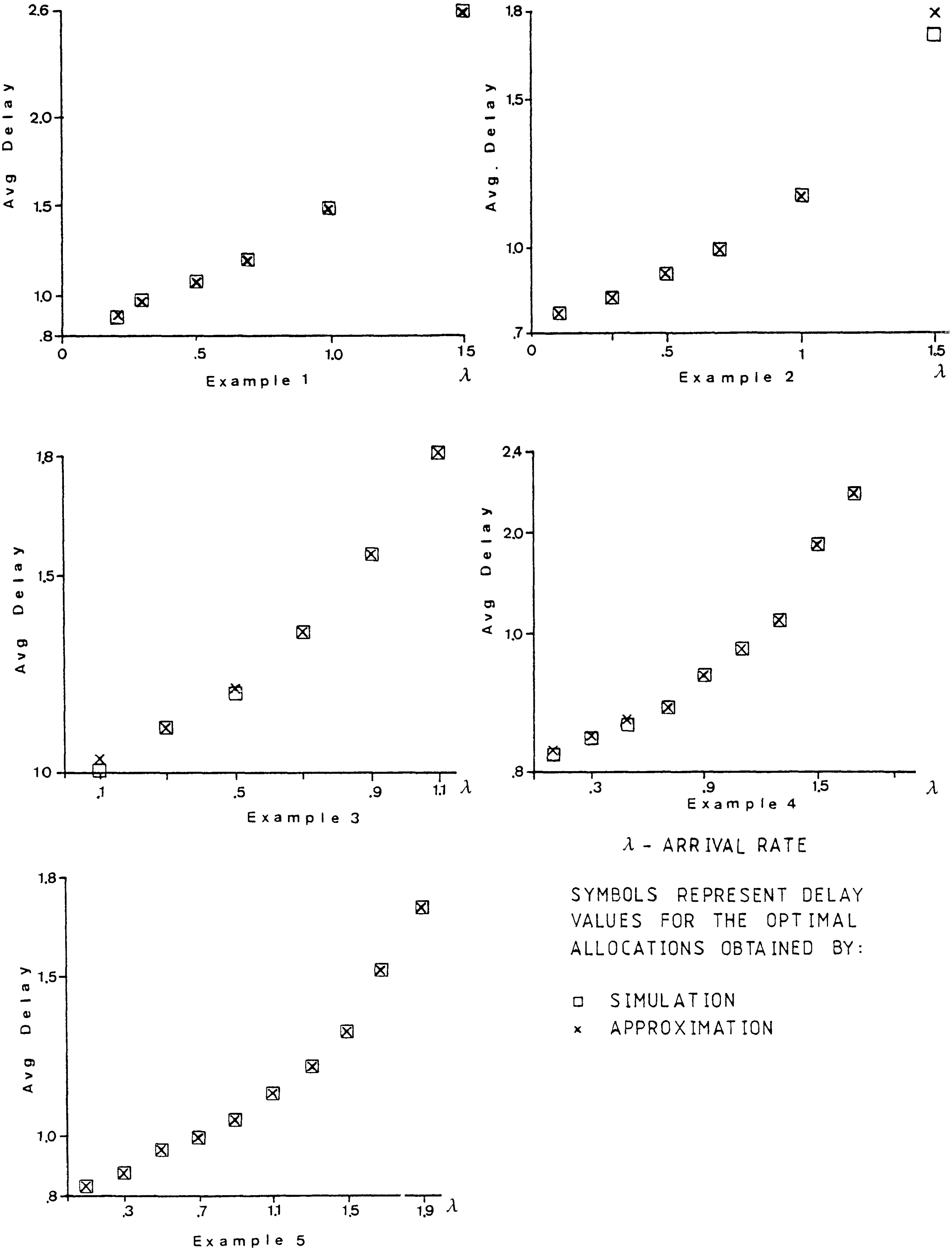


FIGURE 5.11 A comparative test of the approximation technique for the file allocation problem

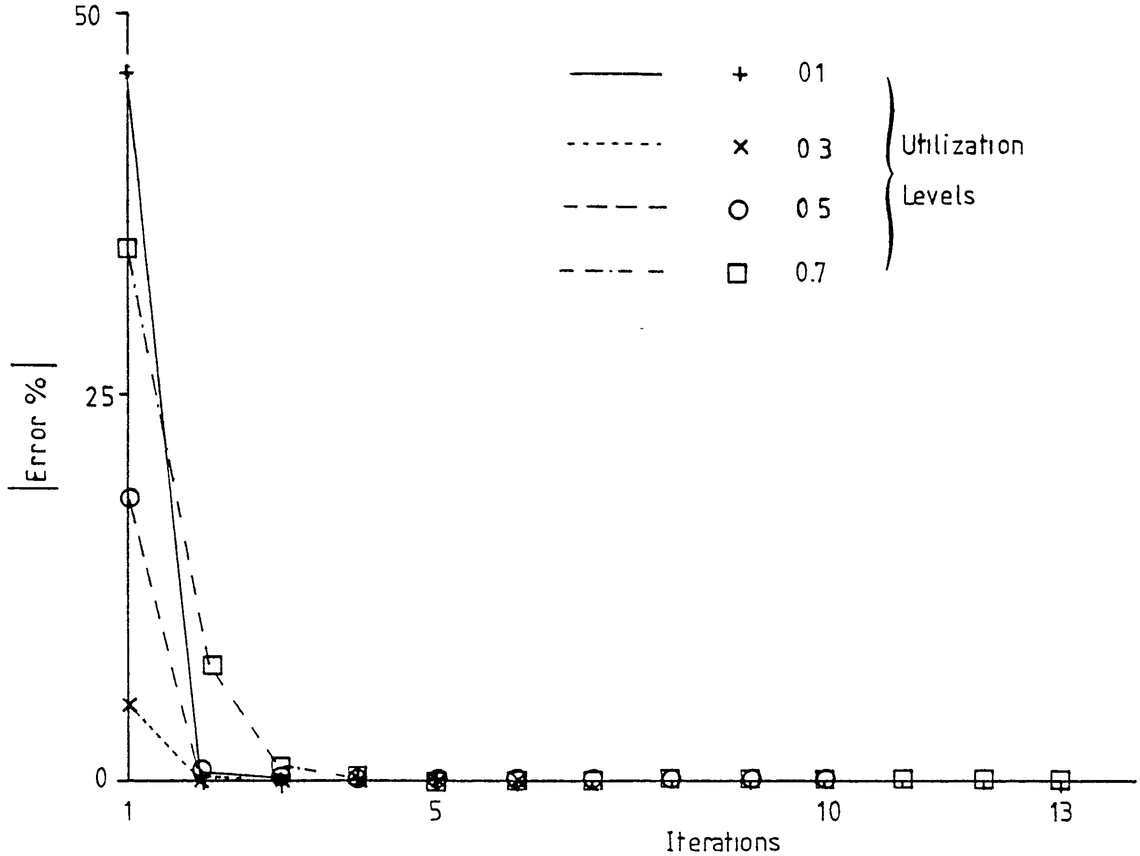


Figure 512 Rate of convergence for different utilizations

CONCLUSIONS AND FURTHER RESEARCH

In this thesis, a set of problems associated with the human interaction with information have been identified and analysed.

Chapters 1-2

The first part of this thesis (chapters 1-2) is concerned with the problem of human interaction with information. This is examined in the context of exploiting existing human information handling skills, thereby allowing systems to become more plausible to experienced and new users alike.

A survey of the human cognition of information has pointed out the importance of understanding human memory processes. The superiority of human memory in interacting with visually presented data, as compared to abstract symbolic, is noted. Imagery however must be restricted to the use of concepts meaningful to the user, as past experience and knowledge play an important role in the cognition, memorizing and subsequent use of information. A conclusion is that the use of meaningful, visually familiar information interaction techniques is desirable in the design of efficient man-machine interfaces.

It is well known that humans have a tendency to form conceptual models of systems they interact with; this process is known to aid interaction effectiveness. In order to enhance this conceptual model formation and use, it has been suggested that pictorial representations of dialogue and data structures are desirable. The highly developed human capability of interacting with the environment in spatial terms has been noted. Spatiality is in constant use in everyday activities -an example being the organization of a desktop- and is potentially a very powerful man-machine interface medium.

The concept of using instinctive human abilities and spatiality has, until recently, been neglected in the design of man-machine interfaces. A set of proposals for exploiting spatiality have however been recently implemented. These database access systems based on spatiality have been described and comparatively evaluated. Based on this comparison and on the factors thereby identified as important in chapter 1, a new information display and interaction technique is evaluated.

This Bifocal display technique employs spatiality while taking note of suggestions based on the user's psychology of interaction. It is deemed to posess several advantages over other comparable techniques. It ensures that the user is fully aware of the context of displayed information, while simultaneously providing a simple, pictorial view of the data structure. It is proposed that these facilities can be implemented in a single display unit, with consequent cost advantages. Other advantages of one display unit are that the user is not forced to shift his gaze across displays to obtain 'navigational' cues. The Bifocal display is ideally suited to browsing, a frequently used and important human information handling ability. The command dialogue presents nodes traversed in the data structure in an easily visible form allowing a user to easily assess his current position. General design principles for such a display have been presented, and the design of a prototype model described, indicating that the concept may be implemented inexpensively with current technology.

It is concluded that the Bifocal technique, by utilizing the concept of spatial information management, by implementing a number of proposals that current thinking on the user's psychology of interaction indicates as desirable, and by utilizing the successful features of other attempts at exploiting spatiality, presents a potentially advantageous approach. In general, it is believed that displays of this type have the potential to greatly enhance the capacity for information display and interaction, particularly with regard to the construction of integrated information systems offering a complete range of information manipulation capabilities.

Chapters 3-5

In the second part of this thesis (chapters 3-5) a generalized local area information system is examined. The motivation for this study is to assess the potential bottlenecks in such systems due to the use of graphics-based interactive information handling techniques. These studies have concentrated on the user's point of view, indicating that this leads to unique considerations and developments.

A number of assumptions based on current developments in the design of such systems allows the formation of simple system models. It is suggested that adaptive file allocation techniques have a role to play in local area information systems design. It is proposed that such adaptation techniques may be implemented more simply in LAN's than in larger scale networks, though of equally significant importance. It is proposed that there exists a need to adjust current network design philosophies to demanding real world environments which together with the newly proposed, high data content man-machine interface media, considerably tighten user service quality constraints.

The absence of sufficient usage information prevents us from completely verifying the assumptions made. Our proposals for network adaptation are therefore intended to initiate studies on the problem of allocating files optimally in such a network. An examination of possible techniques for distributed database management in such networks suggests that a centralized controller scheme, supporting a database that allows multiple copies of files to exist, is an efficient approach. The dynamic routing central server model we derive is similar to the network models used in Packet Switched Networks.

The task of optimally allocating and replicating files in such a network requires an analysis technique suited to dynamic routing central server networks. In chapter 4 therefore, available techniques for the queuing analysis of central server systems are critically reviewed. It has been found that exact techniques are limited to systems of small dimensionality, or constrained to limited configurations. Approximation techniques are difficult to apply and are not general enough. Simulation is an alternative. However simulation is costly to execute, and is not suited to the problem of optimally allocating/replicating files. It is concluded that techniques for efficiently analysing a general version of the dynamic routing central server problem are not available. These results are supported by other researchers who acknowledge the non-triviality of the analysis problem and the need for approximate techniques. In chapter 5, a new approximation technique for the analysis of open, central server, dynamic routing networks has been proposed and evaluated.

The new approximate analysis method is based on separating the queues of each server, and on treating them as independent birth-death processes with state-dependent arrival rates. An iterative technique for solving the system of non-linear balance equations thereby formed, is proposed. Convergence of the iterative technique is shown in Appendix 3 for an example system.

Overall, it was found that the method is accurate for small to medium system utilizations. It is simple to implement and executes efficiently, being able to handle systems of general complexity in terms of the number of servers, their service rates, and the file allocation/replication pattern. Accuracy appears to be independent of the number of servers and their service rates. Accuracy is, however, a function of system utilization and the extent of dynamic routing (i.e. the proportion of replicated files). The approximation method has been used to model a minimum delay protocol, but is flexible enough to allow the modeling of other dynamic routing protocols.

The approximation method has been applied to the problem of optimally allocating/replicating files in a central server network. As a consequence of the non-availability of reference analysis techniques, the allocation procedure employing the proposed approximation is a simple one, and has as a result, only been applied to problems of small dimensionality. The results are very satisfactory, and it is expected that this will also be the case for systems of larger dimensionality.

Contributions

The contributions of this thesis are twofold. First, a set of significant factors based on the user's psychology of interaction with information have been identified and their integration into efficient man-machine interfaces examined. A new proposal for the display of, and interaction with information, the Bifocal display, is presented and investigated with respect to these factors. A comparative, first order evaluation with other similar techniques indicates that displays of this type posess a number of advantages, and have a potential for enhancing the effectiveness of the man-machine interface in an information system. A prototype implementation has demonstrated the viability of the technique and its potential for low-cost implementation.

Secondly, the examination of a local area information network, under certain assumptions based on current developments, has revealed a degree of topology independence, and highlighted the fact that bottlenecks in future systems are likely to occur due to data access rather than data transmission. A data access model based on a central controller database management system is used to establish the problem of optimally allocating files between available file server units. The local area nature of the network allows the use of dynamic routing techniques as a means of enhancing performance. An efficient, general, approximate analysis technique for such dynamic routing networks has been proposed. Modeling accuracy as compared with other techniques is considered to be acceptable. The technique has been applied to several small dimensionality file allocation/replication problems with excellent results. By virtue of its efficiency and consistency, it is usable from within available optimization procedures to evaluate optimal allocation patterns in larger networks.

Further Research

It is believed that the work reported here, opens up a set of new fields that require investigation. In connection with graphics-assisted information handling, a second stage implementation of the Bifocal display is desirable. This implementation is expected to be totally software generated for reasons of flexibility. A full range of features and functions must be supported. These would include the ability to handle a variety of data spaces, the interfacing with existing databases, and the experimental support of a variety of data structures, man-machine dialogues, and graphic input devices.

Comparative evaluations for the optimum use of such displays is clearly necessary. There is an absence of adequate research in this field, mainly due to its recent introduction; for example, studies into optimal data structures for particular applications, the use of colour, the optimum use of icons, shapes, etc. Investigations into the optimum way of utilizing a display screen, for example by examining the limit beyond which displays begin to appear cluttered rather than informative, for various tasks. The examination of which types of icons are perceived most efficiently, and the optimum approaches to 'iconizing' material, given a user's backround. Further studies are required to assess acceptable response delay bounds for graphics operations such as screen scrolling, icon/window movement, and movement between levels of a data structure. Studies similar to those for text-based menus are also desirable, i.e. by examining the optimum extent of a data space and the number of data levels for a variety of applications.

With respect to the analysis of dynamic routing queuing networks, it would be of interest to examine techniques for improving the accuracy of the approximation technique. For example, by investigating techniques to take into account the extent to which conditional probabilities differ from their unconditional counterparts and suggesting techniques for approximately correcting this effect.

The general applicability of the approximate analysis technique must be investigated by examining the results of its application to other networks. These include mixed (i.e. open and closed) networks, queue jockeying networks, networks with arrival processes other than Poisson, and service disciplines other than exponential.

APPENDIX 1

Interfacing a Bifocal Display Generator to the Apple II

In order to examine techniques of bypassing the internal video generation circuitry of any machine, we must first understand the way in which the CPU, memory, and the display circuitry are interconnected and operate. A simplified block diagram for the case of the Apple II, is shown in Figure A1.1. The microprocessor used (6502), requires a 1 Mhz clock signal to operate. In fact, it is active only for 50% of the cycle of this clock, using the other half for internal housekeeping. During this housekeeping, the video display generator is active, producing an address which is fed to the memory bank through a multiplexor, the resulting 8-bit data used to drive either a character generator, or serialized to produce a 1 bit/pixel graphics display. The display generator simultaneously takes care of RAM refresh requirements. This is done by arranging the sequence of video display memory addresses generated to occur in a pattern which matches certain addressing requirements of the RAM chips. This technique has the advantage of saving time, and doing away with separate refresh circuitry or CPU intervention, freeing it for other operations. The penalty paid is the complicated display memory map, causing the bus on the upper side of figure A1.1 to be rather complicated.

The simplest approach to inserting a new display generator in this case is therefore to intercept the CPU address bus at point X, and introduce a new multiplexor, as in Figure A1.2. The original multiplexor is latched to the CPU only state, while the new one is now cycled between the CPU and the Bifocal display generator. By alternately latching the two multiplexors we may therefore switch between the Bifocal and normal modes of operation. The advantages of this approach is that the new display generator is transparent to the remaining circuitry, while we may now access and display any part of memory. In fact, we are constrained to methods of access which take into account the dynamic RAM refresh requirements. In this particular application 16k bytes are allocated to the level II image, and 8k bytes to the level I image.

The most difficult task at this stage was the need to address the chosen display memory pages while making certain that a fairly restrictive set of refresh requirements are simultaneously met. The problem was solved by alternately attempting various address mapping schemes and examining for the above requirements. The result was circuitry implementing a fairly complicated addressing system. Its inputs are virtual page numbers, and offsets within a page as produced by the Bifocal Display generator, and its output physical memory address locations cycled in a manner compatible with the refresh requirements. The circuit constructed also implements the wrap around feature mentioned in Chapter 2.

The remaining part, is a system that has as inputs the coordinates of the scanning beam at any moment across the display screen, comparing it to the desired viewport boundaries, the relative displacement of the data space, and the peeling page index, producing the number of the virtual memory display page which needs to be accessed, and the offset within that page, to be sent to the circuitry described immediately above. The time available for this operation is about 500 nS, which is approximately within the limits of an implementation in low power Scottky TTL. In order to increase the timing margin, it was decided to execute the as above operation on a buffered version of the desired inputs, rather than attempting to execute it during the (short) time period these inputs are valid.

About 40 TTL gates and registers were used for this implementation, the final result being shown in Figure 2.11. The system operated as expected, first time, due to the wide safety margins used. Simple routines were written to manage the scrolling action, and to take into account the complex display memory mapping function. As the CPU is freed from managing the Bifocal display, there is spare processing capacity available, allowing the system to be programmed in interpreted BASIC.

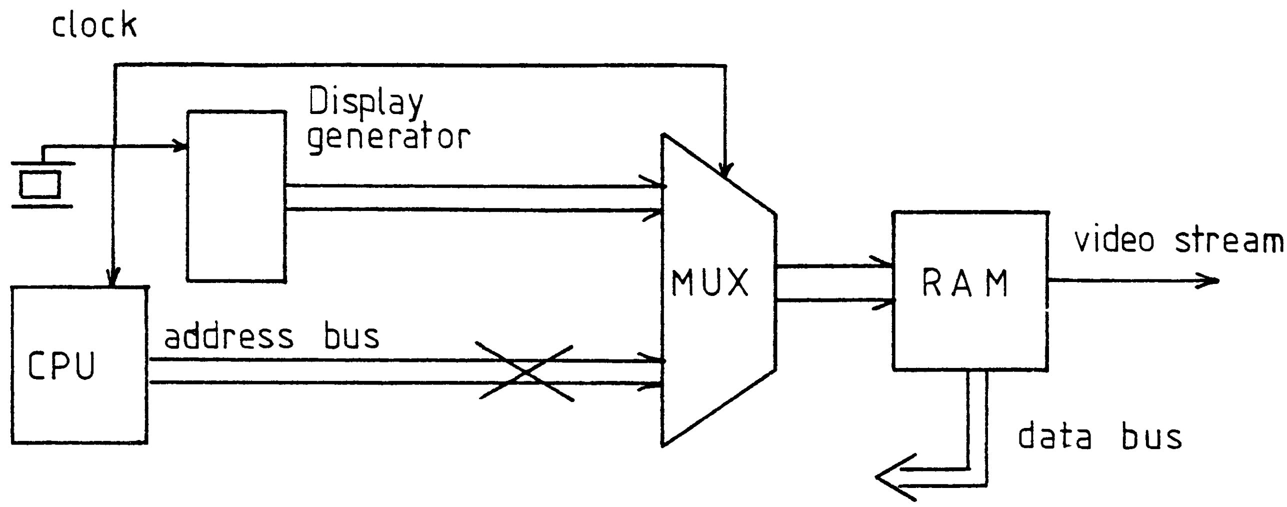


Figure A1.1 Simplified Apple II Block diagram

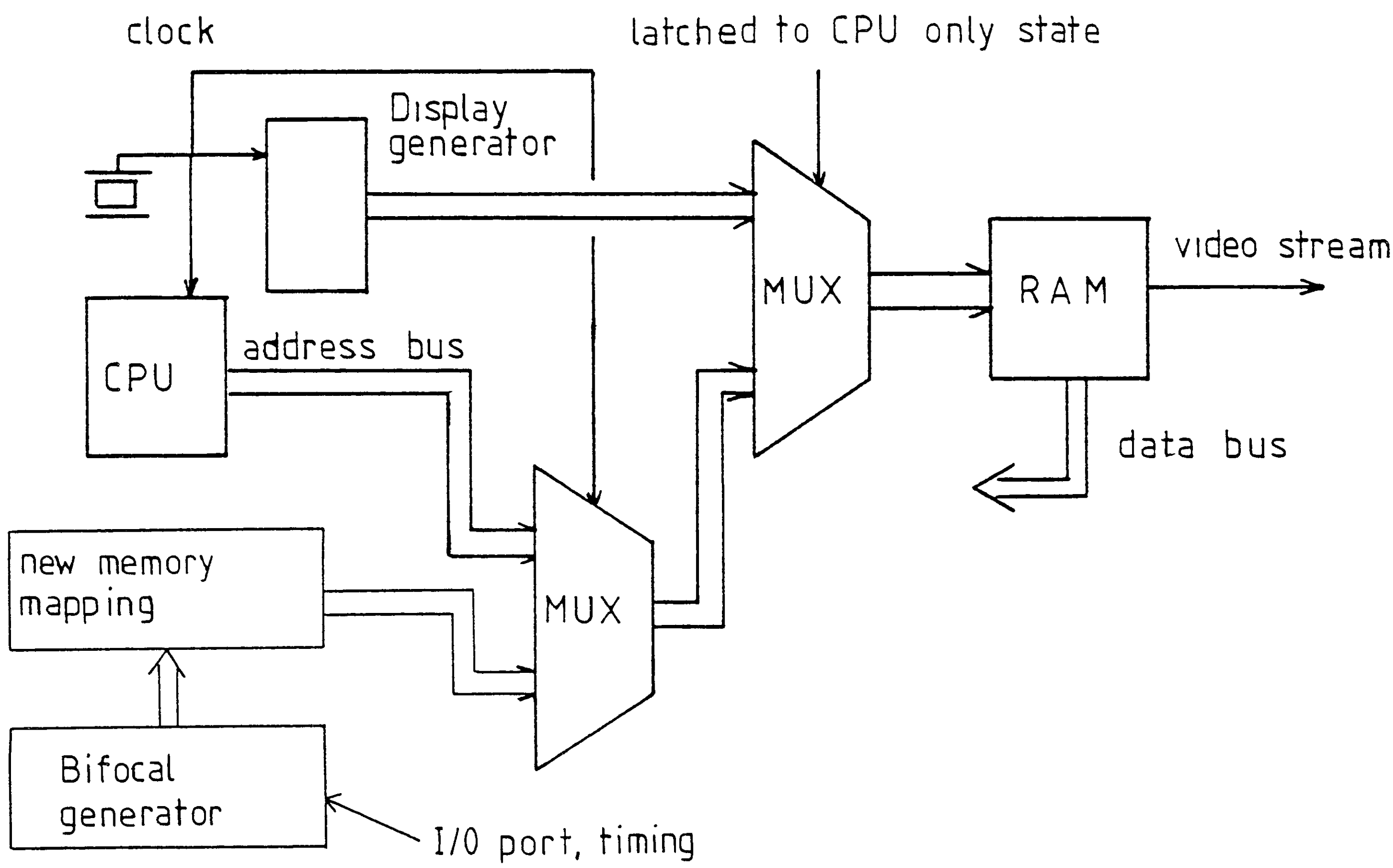


Figure A1 2 Introduction of a Bifocal Display Generator

Appendix 2

The Queuing Network Simulator

A flow chart description of the simulator used as the reference technique in the analysis of these dynamic routing central server networks is presented here. The flow chart is represented in Figure A2. The simulator is relatively easy to construct by virtue of the memoryless properties of the job arrival and service time processes. This allows us to treat each event independently of past events.

A complete statistical analysis of the simulator has not been executed, as its performance was seen to vary depending on the system being simulated. Approximate tests of its performance were obtained from results such as those plotted in Figure 5.10 (0% replicated files). In general, performance was approximately measured by running the simulator for systems for which exact results were available and assuming that performance remains the same for dynamic routing networks.

The simulator was written in FORTRAN and runs on a PDP 11/23. Run time for 1000 arrivals is of the order of 10 seconds. Typical run lengths for the examples under test in chapter 5, were 20000 to 50000 arrivals, this number remaining constant during each set of experiments.

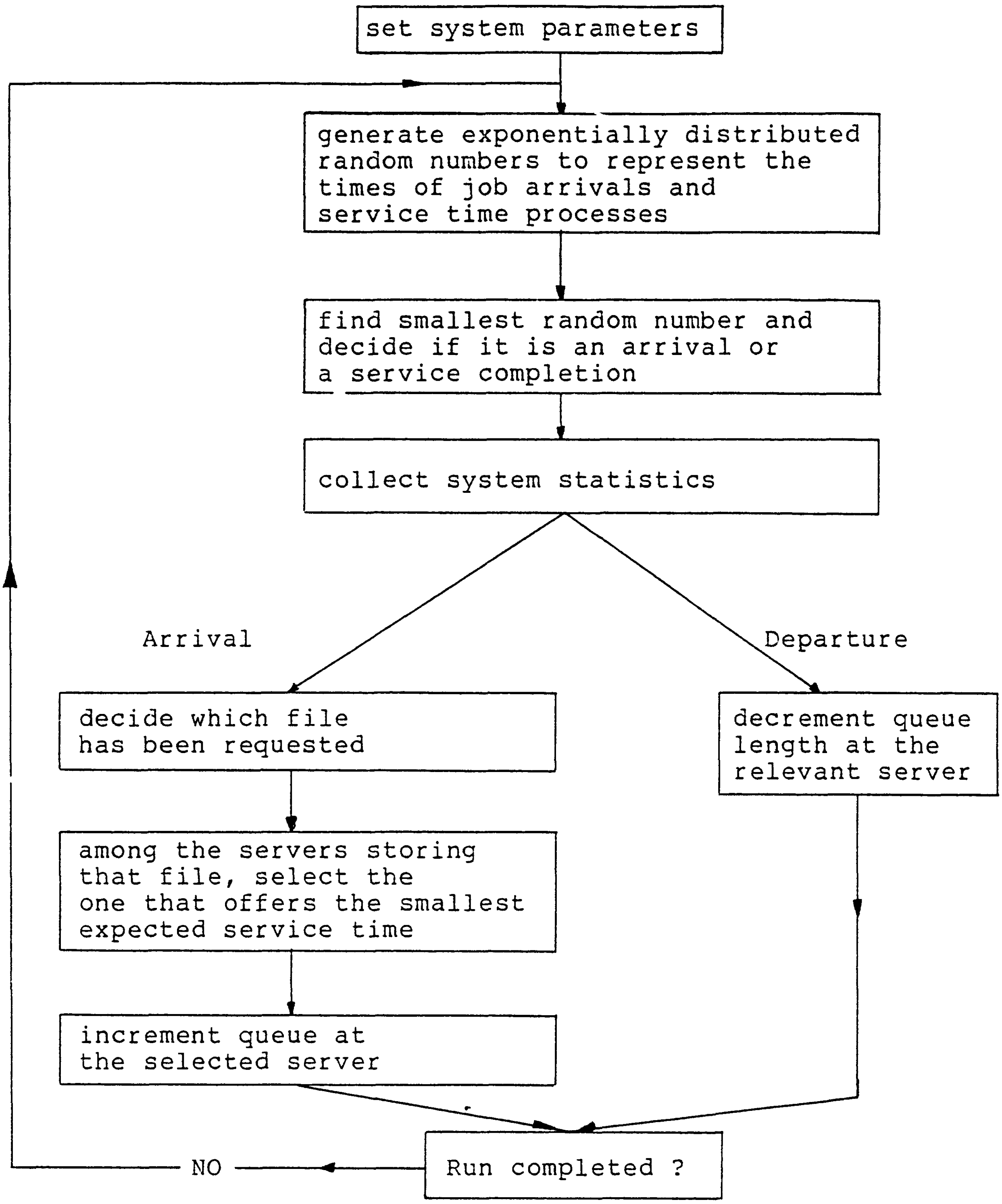


Figure A2 Simulator flow chart

APPENDIX 3

Investigation of Convergence of the Iterative Technique

Consider the two equal server example of Figure A3.1. The number of states per server is limited to 3 for simplicity. The state dependent arrival rates are as in Figure A3.2.

As service rates are equal, let $\lambda = \lambda'/\mu$. The steady state equilibrium probability iterates, are derived using equations (5.3 - 5.5). If $p^j_{k,i}$ denotes the i^{th} it rate of the k^{th} state probability of server j (j = 1,2, k = 0,1,2,), it follows that:

$$p'^{1}_{1,i} \triangleq \lambda$$

$$p'^{1}_{2,i} \triangleq \lambda[p^2_{1,i-1}+p^2_{2,i-1}]p^1_{1,i} = \lambda^2[p^2_{1,i-1}+p^2_{2,i-1}] .$$

Let $$Q_i \triangleq p^2_{1,i-1} + p^2_{2,i-1}, \quad \text{then} \tag{A3.1}$$

$$p^1_{0,i} \triangleq \frac{1}{1+p'^{1}_{1,i}+p'^{1}_{2,i}} = \frac{1}{1+\lambda+\lambda^2 Q_i} . \tag{A3.2}$$

Normalizing, i.e.

$$p^1_{1,i} \triangleq p'^{1}_{1,i} \; p^1_{0,i}$$
$$p^1_{2,i} \triangleq p'^{1}_{2,i} \; p^2_{0,i} .$$

Thus,

$$p^1_{1,i} = \frac{\lambda}{1+\lambda+\lambda^2 Q_i} \tag{A3.3}$$

and

$$p^1_{2,i} = \frac{\lambda^2 Q_i}{1+\lambda+\lambda^2 Q_i} . \tag{A3.4}$$

For the second server,

$$p'^2_{1,i} \triangleq \lambda[p^1_{1,i} + p^1_{2,i}] ,$$

$$p'^2_{2,i} \triangleq \lambda p^1_{2,i}\, p^2_{1,i} = \lambda^2 p^1_{2,i}[p^1_{1,i}+p^1_{2,i}]$$

$$= \lambda^2 \left(\frac{\lambda^2 Q_i}{1+\lambda+\lambda^2 Q_i} \right) R_i$$

where $R_i \triangleq [p^1_{1,i} + p^1_{2,i}]$. (A3.5)

$$p^2_{0,i} \triangleq \frac{1}{1+p'^2_{1,i}+p'^2_{2,i}} = \frac{1}{1+\lambda R_i+\lambda^2\left(\frac{\lambda^2 Q_i}{1+\lambda+\lambda^2 Q_i}\right) R_i} . \tag{A3.6}$$

Normalizing

$$p^2_{1,i} \triangleq p'^2_{1,i}\, p^2_{0,i}$$
$$p^2_{2,i} \triangleq p'^2_{2,i}\, p^2_{0,i} .$$

Thus,

$$p^2_{1,i} = \frac{\lambda R_i}{1+\lambda R_i+\lambda^2\left(\frac{\lambda^2 Q_i}{1+\lambda+\lambda^2 Q_i}\right) R_i} \tag{A3.7}$$

and

$$p^2_{2,i} = \frac{\lambda^2 \left(\dfrac{\lambda^2 Q_i}{1+\lambda+\lambda^2 Q_i}\right) R_i}{1+\lambda R_i+\lambda^2 \left(\dfrac{\lambda^2 Q_i}{1+\lambda+\lambda^2 Q_i}\right) R_i} \quad . \tag{A3.8}$$

Now, let

$$S_i \triangleq \lambda R_{i-1} + \lambda^2 \left(\frac{\lambda^2 Q_{i-1}}{1+\lambda+\lambda^2 Q_{i-1}} \right) R_{i-1} . \tag{A3.9}$$

From (A3.1) and (A3.7-9), we obtain

$$Q_i = \frac{S_i}{1+S_i} \quad , \; Q_{i-1} = \frac{S_{i-1}}{1+S_{i-1}} \; , \; Q_{i-2} = \frac{S_{i-2}}{1+S_{i-2}} \; . \tag{A3.10}$$

From (A3.5) and (A3.3-4) we obtain

$$R_i = \frac{\lambda+\lambda^2 Q_i}{1+\lambda+\lambda^2 Q_i} \qquad , \; R_{i-1} = \frac{\lambda+\lambda^2 Q_{i-1}}{1+\lambda+\lambda^2 Q_{i-1}} \quad . \tag{A3.11}$$

We proceed to investigate how R_i and Q_i vary between iterations.

<u>Proof by induction</u>

We begin by assuming that

$$R_{i-1} > R_{i-2} \qquad Q_{i-1} > Q_{i-2} \quad , \quad i = 3,4,\ldots$$

and attempt to show that :

$$Q_i > Q_{i-1} \quad \text{and} \quad R_i > R_{i-1} \quad , \quad i = 3,4,\ldots .$$

This will be investigated in two parts.

Part 1

If $R_{i-1} > R_{i-2}$ and $Q_{i-1} > Q_{i-2}$

then $Q_i > Q_{i-1}$ $\quad i = 3,4,\ldots .$ (A3.12)

From the assumption,

$$\lambda R_{i-1} > \lambda R_{i-2} . \tag{A3.13}$$

Also from the assumption,

$$Q_{i-1} > Q_{i-2}$$

this means that $\dfrac{\lambda^2 Q_{i-1}}{1+\lambda+\lambda^2 Q_{i-1}} > \dfrac{\lambda^2 Q_{i-2}}{1+\lambda+\lambda^2 Q_{i-2}}$ holds.

Or,
$$\lambda^2 \left(\frac{\lambda^2 Q_{i-1}}{1+\lambda+\lambda^2 Q_{i-1}} \right) R_{i-1} > \lambda^2 \left(\frac{\lambda^2 Q_{i-2}}{1+\lambda+\lambda^2 Q_{i-2}} \right) R_{i-2} \tag{A3.14}$$

since $R_{i-1} > R_{i-2}$ and $R_i > 0$, $\quad i = 1,2,\ldots .$

Adding inequalities (A3.13) and (A3.14) we obtain

$$\lambda R_{i-1} + \lambda^2 \left\{\frac{\lambda^2 Q_{i-1}}{1+\lambda+\lambda^2 Q_{i-1}}\right\} R_{i-1} > \lambda R_{i-2} + \lambda^2 \left\{\frac{\lambda^2 Q_{i-2}}{1+\lambda+\lambda^2 Q_{i-2}}\right\} R_{i-2} \quad \ldots.(A3.15)$$

or, using (A3.9)

$$S_i > S_{i-1} \,. \qquad (A3.16)$$

(A3.16) implies

$$\frac{S_i}{1+S_i} > \frac{S_{i-1}}{1+S_{i-1}} \quad \text{using (A3.10)} \Longrightarrow Q_i > Q_{i-1}, \quad i = 3,4,\ldots\,.$$

Therefore statement (A3.12) is proven.

<u>Part 2</u>

$$\text{If} \quad Q_i > Q_{i-1} \text{ then } R_i > R_{i-1}, \quad i = 2,3,\ldots\,. \qquad (A3.17)$$

This is easily proved using (A3.11).

Combining Parts 1 and 2,

if $\qquad R_{i-1} > R_{i-2}$ and $Q_{i-1} > Q_{i-2}$

then $\qquad Q_i > Q_{i-1}$ and $R_i > R_{i-1}$ $\quad i = 3,4,\ldots\,.$ $\qquad$ (A3.18)

This is the first part of an inductive argument.

With a starting point $p_0^j = 1$, $p_k^j = 0$, $j = 1,2$, $k = 1,2$ we may execute numerically the first iterations of (A3.2-4) and (A3.6-8) with the following results:

$$\left.\begin{array}{l} R_0 = 0,\ R_1 = \frac{\lambda}{1+\lambda},\ R_2 = \frac{\lambda+\lambda^2 Q_2}{1+\lambda+\lambda^2 Q_2} \Rightarrow R_2 > R_1 > R_0 \\ Q_0 \text{ is not defined.} \\ Q_1 = 0, \quad Q_2 = \frac{\lambda R_1}{1+\lambda R_1} \Rightarrow Q_2 > Q_1 \quad . \end{array}\right\} \quad (A3.19)$$

Using (A3.18) and (A3.19) and by induction,

$$\begin{array}{l} R_{i+1} > R_i \\ Q_{i+1} > Q_i \end{array} \qquad i = 1,2,\dots \quad . \qquad (A3.20)$$

We will now use the result (A3.20) to examine

$$p_{0,i}^1,\ p_{1,i}^1,\ p_{2,i}^1,\ p_{0,i}^2,\ p_{1,i}^2,\ p_{2,i}^2$$

and how they vary between iterations.

a) $\qquad p_{0,i}^1 = \dfrac{1}{1+\lambda+\lambda^2 Q_i}$ from (A3.2)

as $\qquad Q_i > Q_{i-1} \Rightarrow p_{0,i}^1 < p_{0,i-1}^1$, $i = 2,3,\dots$

i.e. $\qquad p_{0,i}^1$ is monotonically decreasing.

b) $$p^1_{1,i} = \frac{\lambda}{1+\lambda+\lambda^2 Q_i} \quad \text{from (A3.3)}$$

again as $Q_i > Q_{i-1} \implies p^1_{1,i} < p^1_{1,i-1}$, $i = 2,3,..$

i.e. $p^1_{1,i}$ is monotonically decreasing

c) $$p^1_{2,i} = \frac{\lambda^2 Q_i}{1+\lambda+\lambda^2 Q_i} \quad \text{from (A3.4)} .$$

Since $Q_i > Q_{i-1}$ then $(\lambda^2+\lambda^3)Q_i > (\lambda^2+\lambda^3)Q_{i-1}$

$$\implies \lambda^2 Q_i+\lambda^3 Q_i+\lambda^4 Q_i Q_{i-1} > \lambda^2 Q_{i-1}+\lambda^3 Q_{i-1}+\lambda^4 Q_i Q_{i-1}$$

$$\implies \lambda^2 Q_i(1+\lambda+\lambda^2 Q_{i-1}) > \lambda^2 Q_{i-1}(1+\lambda+\lambda^2 Q_i)$$

$$\implies \frac{\lambda^2 Q_i}{1+\lambda+\lambda^2 Q_i} > \frac{\lambda^2 Q_{i-1}}{1+\lambda+\lambda^2 Q_{i-1}} \quad \text{if } Q_i > Q_{i-1} \qquad \text{(A3.21)}$$

$$\implies p^1_{2,i} > p^1_{2,i-1}$$

i.e. $p^1_{2,i}$ is monotonically increasing.

d) $$p^2_{0,i} = \frac{1}{1+\lambda R_i+\lambda^2 \left(\frac{\lambda^2 Q_i}{1+\lambda+\lambda^2 Q_i}\right) R_i} \quad \text{from (A3.6)}$$

it is known that

$$\frac{\lambda^2 Q_i}{1+\lambda+\lambda^2 Q_i} > \frac{\lambda^2 Q_{i-1}}{1+\lambda+\lambda^2 Q_{i-1}} \quad \text{from (A3.21).}$$

$$\Rightarrow \qquad 1+\lambda R_i + \lambda^2 \left(\frac{\lambda^2 Q_i}{1+\lambda+\lambda^2 Q_i}\right) R_i > 1+\lambda R_{i-1} + \lambda^2 \left(\frac{\lambda^2 Q_{i-1}}{1+\lambda+\lambda^2 Q_i}\right) R_{i-1}$$

since $R_i > R_{i-1}$, $R_i > 0$, $\lambda > 0$, $i = 1,2, \ldots$

$$\Rightarrow \qquad \frac{1}{1+\lambda R_i+\lambda^2 \left(\frac{\lambda^2 Q_i}{1+\lambda+\lambda^2 Q_i}\right) R_i} < \frac{1}{1+\lambda R_{i-1}+\lambda^2 \left(\frac{\lambda^2 Q_{i-1}}{1+\lambda+\lambda^2 Q_{i-1}}\right) R_{i-1}}$$

or $\quad p^2_{0,i} < p^2_{0,\ -1} \qquad i = 1,2,\ldots$

i.e. $\quad p^2_{0,i}$ is monotonic decreasing.

e) $$p^2_{1,i} = \frac{\lambda R_i}{1+\lambda R_i+\lambda^2 \left(\frac{\lambda^2 Q_i}{1+\lambda+\lambda^2 Q_i}\right) R_i} \quad \text{from (A3.7)} .$$

In this case, it is not straightforward to find the direction in which this state probability varies. However we are, at this stage, only interested in whether monotonicity occurs, and not in the direction.

Assume for the moment that we prove the following : if $p^2_{1,i}$ ever decreases with increasing i (iterations), then it is monotonically decreasing from that point onwards.

Therefore:

a) if $p^2_{1,i}$ increases initially with i, it will either:

1) continue increasing or remain constant

2) decrease for some i

if case 2) occurs, then based on the above assumption, it will from that point onwards become monotonic decreasing.

b) It $p^2_{1,i}$ decreases with i, it will remain so by virtue of the assumption.

It is therefore evident that $p^2_{1,i}$ can develop, at the most, one turning point (increasing-decreasing), and is monotonic beyond that point.

The above argument is subject to the proof that if $p^2_{1,i}$ is decreasing at any stage, then it cannot increase again for any i.

<u>Proof</u> that $p^2_{1,i-1} > p^2_{1,i} \leq p^2_{1,i+1}$ (A3.22)

is false $i = 1,2,\ldots$

$$p^2_{1,i} = \frac{\lambda R_i}{1+\lambda R_i+\lambda^2 S_i R_i} \quad \text{from (A3.7)},$$

where $$S'_i \triangleq \frac{\lambda^2 Q_i}{1+\lambda+\lambda^2 Q_i} .$$

Note that $S'_{i+1} > S'_i$ as $Q_{i+1} > Q_i$ from (A3.21).

We shall first examine the R.H. part of (A3.22),

i.e. $p^2_{1,i} \leq p^2_{1,i+1}$ or

$$\frac{\lambda R_i}{1+\lambda R_i+\lambda^2 S_i' R_i} \leq \frac{\lambda R_{i+1}}{1+ R_{i+1}+\lambda^2 S_{i+1}' R_{i+1}} \ .$$

Expanding,

$$\lambda R_i + \lambda^2 R_i R_{i+1} + \lambda^3 R_i R_{i+1} S_{i+1}' \leq \lambda R_{i+1} + \lambda^2 R_i R_{i+1} + \lambda^3 R_i R_{i+1} S_i'$$

simplifying,

$$(R_{i+1}-R_i) \geq \lambda^2 R_i R_{i+1} (S_{i+1}'-S_i')$$

or,

$$\frac{R_{i+1}-R_i}{S_{i+1}'-S_i'} \geq \lambda^2 R_i R_{i+1} \qquad \text{(A3.23)}$$

as $\quad S_{i+1}' - S_i' > 0 \quad , \quad i = 1,2,\ldots$.

Substituting for R_{i+1}, R_i and S_{i+1}', S_i' in terms of Q_i, and after some algebraic manipulation it can be shown that:

$$\frac{R_{i+1}-R_i}{S_{i+1}'-S_i'} = \frac{1}{1+\lambda}$$; substituting into (A3.23),

$$\frac{1}{1+\lambda} \geq \lambda^2 R_i R_{i+1} \ . \qquad \text{(A3.24)}$$

In a similar manner for the L.H. side of (A3.22),

$$p^2_{1,i-1} > p^2_{1,i} \Rightarrow \frac{1}{1+\lambda} < \lambda^2 R_i R_{i-1} \quad . \tag{A3.25}$$

Combining (A3.24) and (A3.25),

$$\lambda^2 R_i R_{i-1} > \frac{1}{1+\lambda} \geq \lambda^2 R_i R_{i+1}$$

$$\Rightarrow \qquad \lambda^2 R_i R_{i-1} > \lambda^2 R_i R_{i+1}$$

$$\Rightarrow \qquad R_{i-1} > R_{i+1} .$$

This last inequality does not hold as it contradicts statement (A3.20). Therefore statement (A3.22) does not hold, indicating that if a decreasing step exists, then it must always be followed by another decreasing step.

f) $$p^2_{2,i} = \frac{\lambda^2 T_i R_i}{1+\lambda R_i + \lambda^2 T_i R_i} \quad \text{using (A3.8)} \ .$$

where

$$T_i \triangleq \frac{\lambda^2 Q_i}{1+\lambda+\lambda^2 Q_i} \quad .$$

We begin with

$$T_i R_i - T_{i+1} R_{i+1} < \lambda R_i R_{i+1} (T_{i+1} - T_i) \tag{A3.26}$$

as the L.H. side is negative and the R.H. side is positive

since $\quad R_{i+1} > R_i$

and $\quad T_{i+1} > T_i$ using $Q_{i+1} > Q_i$ as in (A3.21).

Re-arranging, (A3.26) becomes

$$T_iR_i+\lambda R_iR_{i+1}T_i < \lambda R_iR_{i+1}T_{i+1}+T_{i+1}R_{i+1},$$

adding $\lambda^2 T_iT_{i+1}R_iR_{i+1}$ to both sides, multiplying by λ^2 and factorizing,

$$\frac{\lambda^2 T_iR_i}{1+\lambda R_i+\lambda^2 T_iR_i} < \frac{\lambda^2 T_{i+1}R_{i+1}}{1+\lambda R_{i+1}+\lambda^2 R_{i+1}T_{i+1}} \text{ or } p^2_{2,i} < p^2_{2,i+1} \quad i=1,2,\ldots$$

i.e. $p^2_{2,i}$ is monotonic increasing.

Summarizing, this Appendix has shown that all state probabilities are monotone in the number of iterations taken to solve this simple two-server system. Note that all state probabilities lie in the region[0,1], and are therefore bounded. A finite ending point for the iterations is therefore necessary. The number of iterations required however, may be infinite. In addition, we have not examined the rate with which the iterative process converges to the solution.

The extension of these techniques to more general systems is not obvious at this stage.

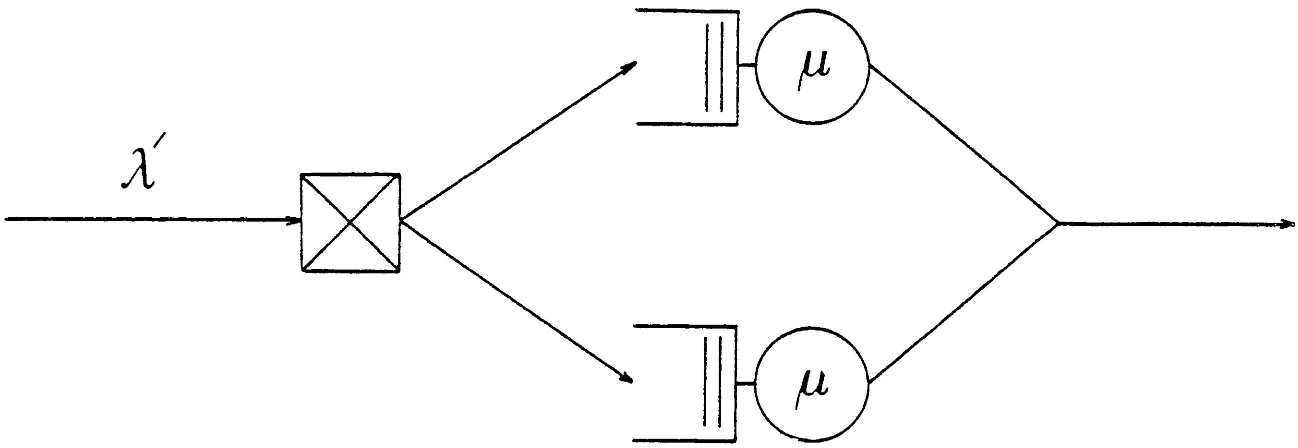


Figure A3.1 Two equal servers

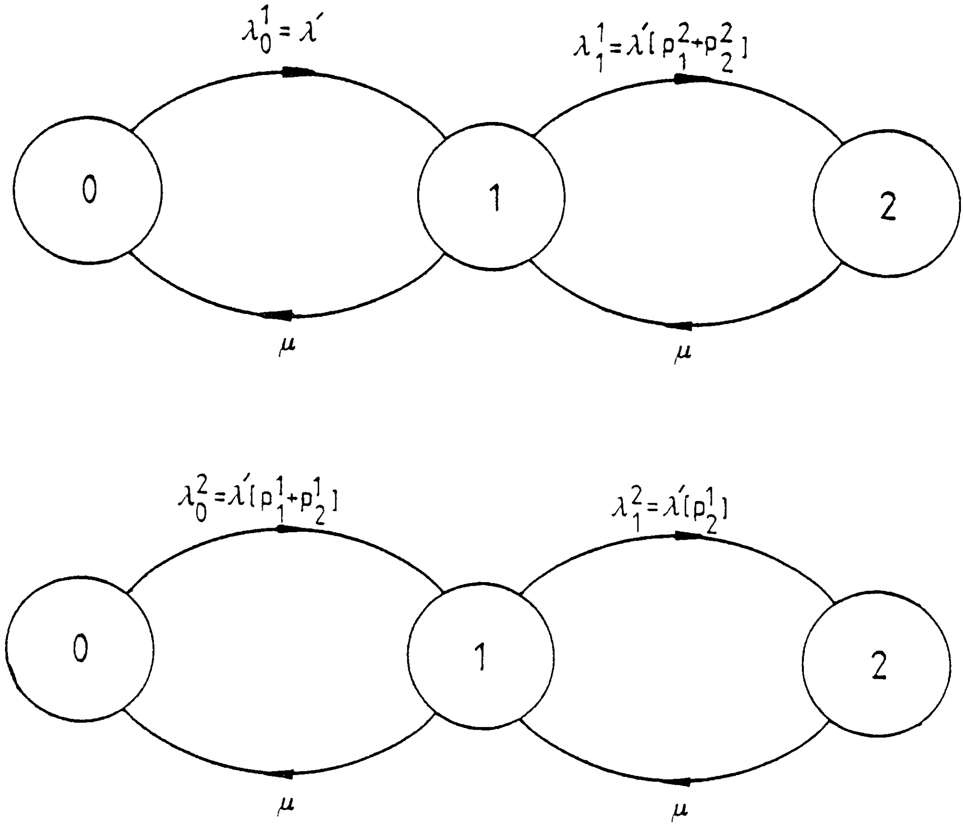


Figure A3.2 Arrival rates

REFERENCES

[APPE82] M.D. Apperley, I.V. Tzavaras, R. Spence, 'A Bifocal Display Technique for Data Presentation', in EUROGRAPHICS '82, edited by D.S. Greenaway, E.A. Warman, North-Holland, pp.27-43

[APPE83] M.D. Apperley, R. Spence, 'Hierarchical Dialogue Structures in Interactive Computer Systems', Software-Practice and Experience, vol.13, 1983, pp.777-790

[BEWL83] W.L. Bewley, T.L. Roberts, D. Schroit, W.L. Verplank, 'Human Factors Testing in the design of Xerox's 8010 'STAR' office Workstation', Proceedings of 'Human factors in Computer Systems', SIGCHI, 1983, ACM

[BOLT79] R.A. Bolt, 'Filing and retrieving in the future - Spatial Data Management', Infotech International state of the art report, vol.2, 1979

[BOLT81] R.A. Bolt, 'Gaze-orchetrated Dynamic Windows', Computer Graphics, vol.15, no.3, 1981, pp.109-119

[BOOK76] A. Bookstein, W. Cooper, 'A general mathematical model for information retrieval systems', Library Quarterly, vol.46, no.2, pp.153-167

[BOOR81] R.R. Boorstyn, A. Livne, 'A technique for adaptive routing in networks', IEEE Transactions on Communications, COM-29, no.4, April 1981, pp.474-480

[BROW82] J.W. Brown, 'Controlling the complexity of menu networks', Communications of the ACM, July 1982, vol.25, no.7, pp.412-418

[BURK56] P.J. Burke, 'The output of a queuing system', Operations Research, vol.4, 1956, pp.699-704

[BUDR72] Z.L. Budrikis, 'Visual Fidelity Criterion and Modeling', Proceedings of the IEEE, Vol. 60, no.7, 1972, pp.771-779

[BUX81] W. Bux, 'Local-Area subnetworks: a performance comparison', in 'Local networks for computer communications', edited by A. West, P. Janson, North-Holland 1981

[CANF82] D. Canfield Smith, C. Irby, R. Kimball, B. Verplank, 'Designing the Star User Interface', BYTE, April 1982, pp.242-282

[CARD82] S.K. Card, 'User perceptual mechanisms in the search of Computer Command Menus', in Proceedings of 'Human Factors in Computer Systems', March 15-17, 1982, Gaithersburg, Maryland

[CARD83] S.K. Card, T.P. Moran, A. Newell, 'The psychology of Human Computer Interaction', Lawrence Erlbaum Associates, Inc. Publishers 1983

[CARE82] T. Carey, 'User differences in interface design', IEEE Computer, November 1982, pp.14-20

[CHAN72] K.M. Chandy, 'The analysis and options for general queuing networks', 6th annual Princeton conference on Information Sciences and Systems, 1972, pp.224-228

[CHAN78] K.M. Chandy, C.H. Sauer, 'Approximate methods for analyzing queuing network models of computer systems', Computing Surveys, vol.10, no.3, September 1978, pp.281-316

[CHAN80A] I.F. Chang, 'Recent advances in display technologies', Proceedings of the S.I.D., vol.21/2, 1980, pp.45-53

[CHAN80B] N. Chang, K. Fu, 'Query by pictorial example (QPE)', IEEE Transactions on Software Engineering, vol.SE-6, 1980

[CHAN81] N. Chang, 'Picture query languages for pictorial information systems', IEEE Computer, November 1981

[CHEN73A] P.P.S. Chen, 'Optimal partition of input load to parallel exponential servers', Proceedings of the fifth Southeastern Symposium on System Theory, North-Carolina state University, March 1973

[CHEN73B] P.P.S. Chen, 'Optimal file allocation in a multi-level storage system', AFIPS, vol.42, 1973, pp.277-282

[CHOU81] W. Chou, A.W. Bragg, A.A. Nilssen, 'The need for adaptive routing in the chaotic and unbalanced traffic environment', IEEE Trans, COM-29, no.4, April 1981, pp. 481-490

[CHOW77] Y.C. Chow, W.H. Kohler, 'Dynamic load balancing in homogeneous two-processor distributed systems', in 'Computer Performance', edited by K.M. Chandy and M. Reiser, North-Holland 1977

[CHOW79] Y.C. Chow, W.H. Kohler, 'Models for dynamic load balancing in a heterogeneous multiple processor system', IEEE Transactions on Communications, C-28, no.5, May 1979, pp.354-361

[CHU69] W.W. Chu, 'Optimal file allocation in a multiple computer system', IEEE Transactions on Computers, October 1969

[CHU76] W.W. Chu, 'Performance of file directory systems for databases in star and distributed networks', Proceedings AFIPS, NCC, 1976, pp.428-438

[CHU79] W.W. Chu, P.P. Chen, 'Tutorial: Centralized and Distributed Data Base systems', IEEE Computer Society 1979, IEEE Catalog no.EH0154-5

[CLAR78] D.D. Clark, K.T. Pogran, D.P. Reed, 'An Introduction to Local Area Networks', Proceedings of the IEEE, November 1978, vol.66, no.11, pp.1497-1517

[COHE77] G. Cohen. 'The psychology of cognition; chapter 2: Visual Imagery in Thought', Academic Press 1977, pp.26-45

[COLE80] I. Cole, 'The role of memory in the external storage and retrieval of information', PhD thesis, Loughborough University 1980

[COTT77] I.W. Cotton, 'Cost-Benefit Analysis of Interactive systems', Computer Networks, 1, 1977, pp.311-324

[COUL79] G. Coulouris, 'The personal computer in the office of the future', Infotech international state of the art report on man-machine communication vol.2, 1979

[DATE81] C.J. Date, 'An introduction to database systems', Addison-Wesley 1981

[DAVI83] P.A. Davies, F.A. Ghani, 'Optical fibre computer communication networks', proceedings of the Medtiterranean Electrotechnical Conference (MELECON) 1983, vol I, Athens 1983

[DONE78] W.C. Donelson, 'Spatial Management of Information', Proceedings of the ACM/SIGGRAPH, 1978, pp.203-209

[DOWD78] C.W. Dowdy, 'Optimal branching probabilities and their relationship to computer file distribution', PhD Thesis, Duke University 1978

[DOWD82] L.W. Dowdy, D.V. Foster, 'comparative models of the file assignment problem', Computing Surveys, vol.14, no.2, June 1982, pp.287-313

[ELLI77] C.A. Ellis, ' A Robust Algorithm for Updating Duplicate Databases', Proceedings of the 2nd Berkeley workshop on Distributed Data Management and Computer Networks 1977, also in [CHU79]

[EPHR80] A. Ephremides, p. Varaiya, J. Walrand, 'A simple dynamic routing problem', IEEE Transactions on Automatic Control, Ac-25, no.4, 1980, pp.690-693

[EYSE77] M.W. Eysenck, 'Human Memory: Theory, Research, and Individual Differences', Pergamon Press 1977

[FIEL77] C. Fields, N. Negroponte, 'Using new clues to find data', Proceedings on Very Large Data Bases, Tokyo, Japan 1977 (IEEE), pp.156-158

[FOLE71] J.D. Foley, 'An approach to the optimum design of computer graphics systems', Communications of the ACM, June 1971, vol.14, no.6, pp.380-389

[FOLE74] J.D. Foley, V.L. Wallace, 'The art of Natural Graphic Man-Machine Conversation', Proceedings of the IEEE, vol.62, no.4, April 1974, pp.462-471

[FOSC77] G.J. Foschini, 'On Heavy Traffic Diffusion and Dynamic Routing in packet switched networks', in Computer Performance, edited by K.M. Chandy and M. Reiser, North-Holland 1977, pp.499-513

[FOSC78] G.J. Foschini, J. Salz, 'A basic Dynamic Routing problem and Diffusion', IEEE Transactions on Communications, COM-26, no.3, March 1978, pp.320-327

[FOST81] D.V. Foster, C.W. Dowdy, J.E. Ames, 'File assignment in a Computer Network', Computer Networks 5 (1981), pp.341-349

[FRIE82] M. Friedell, J. Barnett, D. Kramlich, 'Context-sensitive Graphic presentation of information', Computer Graphics, vol.16, no.3, 1982, pp.181-188

[GARC80] H. Garcia-Molina, 'Reliability issues for completely replicated distributed databases', Proceedings of COMPCON 80 on Distributed Computing, Washington, Fall 1980

[GENT83] D. Gentner, A.L. Stevens, 'Mental Models', Hillsdale NJ: Erlbaum, 1983

[GILO76] W.K. Giloi, 'Interactive Systems: Patterns and Prospects', in Lecture Notes in Computer Science, no.49, Interactive Systems, edited by A. Blaser, and C. Hackl, Springer Verlag 1976

[GOOD82] T. Goodman, R. Spence, 'The effects of potentiometer dimensionality, system response time, and time of day on interactive graphical problem solving', Human Factors, 1982, 24(4), pp.437-456

[GOUL83] J.D. Gould, 'Designing for usability - key principles and what designers think', Proceedings of 'Human factors in Computer Systems, SIGCHI, 1983, ACM

[GREE78] D. Greenblatt, J. Waxman, 'A study of three database query languages', in Databases: improving usability and responsiveness, edited by B. Shneiderman, Academic Press 1978

[GUE68] R.L. Gue, J.C. Liggett, K.C. Cain, 'Analysis of algorithms for the 0-1 programming problem', Communications of the ACM, vol.11, no.12, 1968, pp.837-844

[HABE82] R.N. Haber, L. Wilkinson, 'Perceptual Components of Computer Displays', IEEE Computer Graphics and Applications, 1982

[HALA83] F.G. Halasz, T.F. Moran, 'Mental Models and problem solving using a calculator', Proceedings of 'Human factors in Computer Systems, SIGCHI, 1983, ACM

[HAYM80] C. Hayman, 'Coping with the people factor', Management Today, November 1980, pp.131-136

[HELD75] G. Held, M.R. Stonebraker, E. Wong, 'INGRES: a relational database system', AFIPS, NCC, 1975

[HEME82] K. Hemenway, 'Psychological issues in the use of icons in command menus', in Proceedings of Human Factors in Computer Systems', March 15-17, 1982, Gaithersburg, Maryland

[HERO80A] C.F. Herot, 'Spatial Management of Data', ACM Transactions on Database systems, vol.5, no.4, 1980, pp.493-514

[HERO80B] C.F. Herot, R. Carling, M. Friedell, D. Kramlich, 'A Prototype Spatial Data Management System', Proceedings of ACM/SIGGRAPH, 14, 3, 1980, pp.63-70

[HOWA66] R.A. Howard, 'Information Value Theory', IEEE Trans on Systems Science and Cybernetics, vol.SSC-2, no.1, 1966, pp.22-34

[IKED80] K. Ikeda et al,'Computer Network coupled by 100MBPS optical fiber using bus -system planning and ring bus subsystem description-', Proceedings of COMPCON 1980, Fall, on Distributed Computing, pp.159-165

[ISHI75] A. Ishikawa, H. Mieno, 'Design of video information system and the fuzzy information theory', Proceedings of EUROCOMP 1975

[KAMO81] F. Kamoun, L. Kleinrock, R. Muntz, 'Queuing analysis of the ordering issue in a Distributed Database Concurrency control mechanism', 2nd International Conference on Distributed Computing systems, Paris 1981

[KINN79] J.A.S. Kinney, 'The use of colour in wide angle displays', Proceedings of the S.I.D., vol.20/1, 1979, pp.33-40

[KLEI75] L. Kleinrock, 'Queuing Systems, vol I:Theory', Wiley 1975

[KLEI76] L. Kleinrock, 'Queuing Systems, vol II:Computer Applications', Wiley 1976

[KOBA74] H. Kobayashi, 'Applications of the diffusion approximation to Queuing Networks I: Equilibrium Queue Distributions' Journal of the ACM, vol.21, no.2, 1974, pp.316-328

[KOHL80] W.H. Kohler, 'Overview of synchronization and recovery problems in distributed databases', Proceedings of COMPCON 80, on Distributed Computing, Washington Fall 1980

[KREB79] M.J. Krebs, 'Design Principles for the use of color in displays', Proceedings of the S.I.D., vol.20/1, 1979, pp.10-15

[LAWL66] E.L. Lawler, M.D. Bell, 'A method for solving discrete optimization problems', Operations Research, vol.14, 1966, pp.1098-1112

[LEAV74] B.M. Leavenworth, 'An overview of non-procedural languages', SIGPLAN notices, April 1974, 9, pp.1-12

[LECL82] Y. Leclerc, S.W. Zucker, D. Leclerc, 'A browsing approach to documentation', IEEE Computer, June 1982, pp.46-49

[LIMB82] J.O. Limb, 'Communications in the Automated Office', Guest editor's prologue, IEEE Transactions on Communications, special issue, vol COM-30, no.1, pp.1-5

[LIND72] P.H. Lindsay, D.A. Norman, 'Human Information Processing: an Introduction to Psychology', Academic Press, 1972

[MACH77] C. Machover, M. Neighbors, C. Stuart, 'Graphics Displays', IEEE Spectrum, 1977, pp.24-32

[MALO83] T.W. Malone, 'How do people organize their desks? Implications for the design of Office Information Systems', ACM Transactions on Office Information Systems, vol.1, no.1, 1983, pp.99-112

[MARC81] R. Marcogliese, R. Novarese, 'Module and Data allocation methods in distributed systems', 2nd International Conference on Distributed Computing systems, Paris, April 1981

[MARI79] R.A. Marie, 'An approximate analytical method for general queuing networks', IEEE Trans, SE-5, no.5, 1979, pp.530-538

[MART73] J. Martin, 'Design of Man-Computer Dialogues', Prentice-Hall, 1973

[MART80] T. Martin, 'Information Retrieval', in 'Human Interaction with Computers' edited by H.T. Smith, and T.P.G. Green, Academic Press 1980

[METC76] R.M. Metcalfe, D.R. Boggs, 'Ethernet: Distributed Packet Switching for Local Computer Networks', Communications of the ACM, July 1976, vol.19, no.7, pp.395-403

[MCGE76] W.C. Mc Gee, 'On User Criteria for Data Model Evaluation', ACM Transactions on Database Systems, vol.1, no.4, 1976, pp.370-387

[MILL81] D.P. Miller, 'The depth/breadth tradeoff in hierarchicall computer menus', Proceedings of the Human Factors Society, 25th annual meeting, 1981, pp.296-300

[MUNT78] R.R. Muntz, 'Queuing Networks: a critique of the state of the state of the art and directions for the future', Computing Surveys, vol.10, no.3, 1978, pp.353-359

[NEGO76] C.V. Negoita, P. Flondor, 'On fuzziness in information retrieval', Int. J. Man-Machine Studies, 1976, 8, pp.711-716

[NEGR81] N. Negroponte, 'Media Room', Proceedings of the S.I.D., vol.22/2, 1981

[NEWM76] W.M. Newman, 'Trends in graphic display design', IEEE trans. on Computers, vol.C-25, no.12, 1976, pp.1321-1325

[NISB76] R.E. Nisbett, R. Crandall, H. Reed, 'Popular Induction: Information is not Necessarily Informative', in 'Cognition and Social Behaviour', edited by J.S. Carroll and J.W. Payne, Lawrence Erlbaum Associates, Inc., Publishers, 1976

[NORM83] D. A. Norman, 'Design Principles for human computer interfaces', Proceedings of 'Human factors in Computer Systems', SIGCHI, 1983, ACM

[OGDE83] W.C. Ogden, S.R. Brooks, 'Query languages for the casual user; exploring the middle ground between formal and natural languages', Proceedings of 'Human factors in Computer Systems', SIGCHI, 1983, ACM

[PAIV71] A. Paivio, 'Imagery and Verbal processes', Holt, Rinehart and Winston, London 1971, referenced in EYSE77

[PEEB78] R. Peebles, E. Manning, 'System architecture for ditributed data management', Computer, January 1978, pp.40-47

[PYKE76] T.N. Pyke Jr, 'Assuring User Service Quality in a Distributed Computer Network', Digest of Papers, COMPCON 76, Spring

[RAWS78] E.G. Rawson, R.M. Metcalfe, 'Fibernet: Multimode optical fibers for local networks', IEEE Transactions on Communications, July 1978, pp.983-990

[REED81] D. Reed, L. Svobodova, 'Swallow: a distributed data storage system for a local network', in 'Local networks for computer communications, edited by A. West, P. Janson, North-Holland 1981, pp.355-373

[REIS74] M. Reiser, H. Kobayashi, 'Accuracy of the diffusion approximation for some queuing systems', IBM Journal of Research and Development, 18, 2, March 1974, pp.110-124

[REIS75] P. Reisner, R.F. Boyce, D.D. Chamberlin, 'Human factors evaluation of two database query languages, SQUARE and SEQUEL', AFIPS, NCC, 1975

[REIS77] P. Reisner, 'Use of psychological experimentation as an aid to development of a Query Language', IEEE Transactions on Software Engineering, vol.SE-3, no.3, May 1977, pp.218-229

[REIS81] P. Reisner, 'Human Factors Studies of Database Query Languages: A Survey and Assessment', Computing Surveys, vol.13, no.1, 1981

[ROBE79] G. Robertson, D. McCracken, A. Newell, 'The Zog approach to man/machine communication', Rep. no.CMU-CS-79-148, Computer Science Dept, Carnegie-Mellon University

[ROSE78] D.J. Rosenkrantz, 'System level concurrency control for distributed database systems', ACM transactions on Database systems, vol.3, no.2, June 1978

[ROUF80] J.A.J. Roufs, H. Bouma, 'Towards Linking Perception Research and Image Quality', Proceedings of the S.I.D., vol.21/3, 1980

[SAUE81] C.H. Sauer, K.M. Chandy, 'Computer Systems Performance Modeling', Prentice-HAll 1981

[SAUE83] C.H. Sauer, ' Computational algorthms for state-dependent queuing networks', ACM Transactions on Computer Systems, vol.1, no.1, 1983, pp.67-92

[SAVA82] R.E. Savage, J.K. Hakinek, T.W. Barnhart, 'The design, simulation and evaluation of a menu driven user interface', in Proceedings of 'Human Factors in Computer Systems', March 15-17, 1982, Gaithersburg, Maryland

[SHAC79] B. Shackel, 'Man/Computer Communication', Infotech International, State of the Art report, vol 1, 1979

[SHAN48] C. Shannon, 'A mathematical theory of communication', Bell System Technical Journal, vol.27, pp.379-423, and 623-656, 1948

[SHER74] T.B. Sheridan, W.R. Ferrell, 'Man-Mahine Systems: Information, Control, and Decision Models, of Human Performance', MIT Press 1974

[SHNE78] B. Shneiderman, 'Improving the Human Factors Aspect of Database Interactions', ACM Transactions on Database Systems, vol.3, no.4, 1978, pp.417-439

[SMIT82] D.C. Smith, C. Irby, R. Kimball, B. Verplank, E. Harslem, 'Designing the Star User Interface', BYTE, April 1982, pp.242-282

[SNOW83] K. Snowberry, S.R. Parkinson, N. Sisson, 'Computer display menus', Ergonomics, vol.26, no.7, 1983, pp.699-712

[SPEN77] R. Spence, M.D. Apperley, 'The interactive-graphic man-computer dialogue in computer-aided circuit design', IEEE Transactions on Circuits and Systems, vol.CAS-24, no.2, 1977, pp.49-61

[SPEN82] R. Spence, M.D. Apperley, ' Data base navigation: an office environment for the professional', Behaviour and Information Technology, vol.1, no.1, 1982, pp.43-54

[STAM71] R.K. Stamper, 'Some ways of measuring information', The Computer Bulletin, December 1971, pp.432-436

[TAHA82] H.A. Taha, 'Operations Research: an Introduction', third edition, Macmillan 1982

[TEIC79] W.H. Teichner, 'Color and Visual Information Coding', Proceedings of the S.I.D., vol.20/1, 1979

[THOM75] J. Thomas, J. Gould, ' A psychological study of query by example', Proceedings of AFIPS, NCC, vol.44, 1975

[THOM78] R.H. Thomas, 'A solution to the concurrency problem for multiple copy data bases', COMPCON 1978, Spring, also in [CHU79]

[TOMB82] J.W. Tombaugh, S.A. McEwen, 'Comparison of two information retrieval methods on videotex: Tree structure vs. Alphabetical directory', in Proceedings of 'Human Factors in Computer Systems', March 15-17, 1982, Gaithersburg, Maryland

[TOMI78] H. Tominaga, S. Tajima, K. Saito, 'Trade-off of File Directory systems for Data Base', Evolutions in Computer Communications, North-Holland 1978, Kyoto, Japan 1978

[TOWS80] D. Towsley, 'Queuing network models with state-dependent routing', Journal of the ACM, vol.27, no.2, April 1980, pp.323-337

[TRIV78] K.S. Trivedi, 'Analytical modeling of computer systems', IEEE Computer, October 1978

[TZAV82] I. Tzavaras, R. Spence 'Specifications on a graphic database navigation system imposed by human factor considerations', Proceedings of the International Zurich Seminar (IZS) on Digital Communications, Man-Machine Interaction Zurich 1982, pp.117-120

[TZAV83A] I. Tzavaras, 'Resource Allocation in Interactive Information Systems', Proceedings of the Mediterranean Electrotechnical Conference (MELECON) 1983, vol I, Athens 1983

[TZAV83B] I. Tzavaras, 'Predicting the performance of queuing networks with dynamic job routing', Electronics Letters, vol.19, no.21, October 1983, pp.899-901

[WAH84] B.W. Wah, 'File placement on distributed computer systems', Computer, January 1984, pp.23-32

[WALL66] V.L. Wallace, R.S. Rosenberg, 'Markovian models and numerical analysis of computer system behaviour', AFIPS, SJCC, 1966, pp.141-148

[WELT81] C. Welty, D.W. Stemple, 'Human Factors Comparison of a Procedural and a Nonprocedural Query Language', ACM Transactions on Database Systems, vol.6, no.4, pp.626-649

[WILL83] G. Williams, 'The Lisa computer system', BYTE, February 1983, pp.33-50

[WONG78] J.W. Wong, 'Queuing Network modeling of Computer Communication Networks', Computing Surveys, vol.10, no.3, pp.343-351

[YUM81] T.S.P. Yum, M. Schwartz, 'The Join-Biased-Queue rule and its application to Routing in Computer Communication Networks', IEEE Transactions on Communications, vol.COM-29, no.4, April 1981

[ZLOO75] M. Zloof, 'Query By Example', Proceedings of AFIPS, NCC, vol.44, 1975, pp.431-438

[ZLOO76] M. Zloof, 'Query By Example - Operations on hierarchical data bases', Proceedings of AFIPS, NCC, 1976, pp.845-853

[ZLOO81] M. Zloof, 'QBE/OBE: a language for Office and Business Automation', IEEE Computer, May 1981, pp.13-22

[ZUND71] P. Zunde, 'On Signs, Information, and Information Measures', ISA Transactions, vol.10, no.2, 1971, pp.189-193