

AD-A195 054

INTEGRATING IMAGES APPLICATIONS AND COMMUNICATIONS
NETWORKS VOLUME 5(U) MASSACHUSETTS INST OF TECH
CAMBRIDGE A GUPTA ET AL. DEC 87 MIT-KBIISE-5

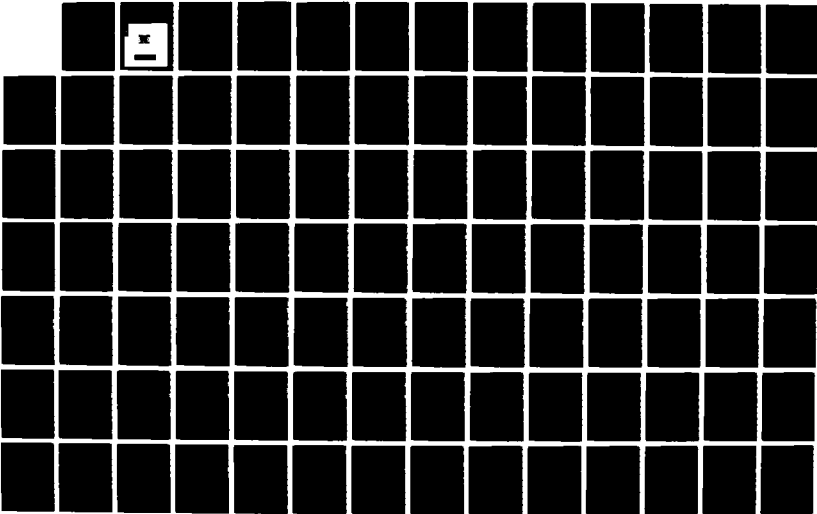
1/4

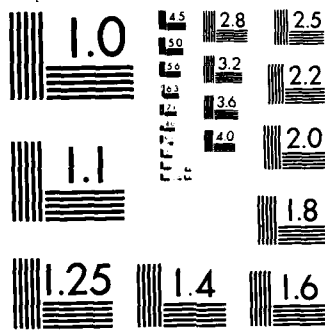
UNCLASSIFIED

DTR557-85-C-00003

F/G 5/2

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

DTIC FILE COPY



Massachusetts
Institute of
Technology

Knowledge-Based
Integrated Information
Systems Engineering
(KBIISE) Project

Volume 5

Integrating Images, Applications, and Communications Networks

2

Amar Gupta
Stuart Madnick

SERIES EDITORS

AD-A195 854

DTIC
SELECTED
JUL 19 1988
S D
66 D

DISTRIBUTION STATEMENT A
Approved for public release
Distribution Unlimited

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER KBIISE-5	2. GOVT ACCESSION NO. A175-854	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Integrating Images, Applications and Communications Networks	5. TYPE OF REPORT & PERIOD COVERED Part of final report: September 86-Jan 88	
	6. PERFORMING ORG. REPORT NUMBER KBIISE-5	
7. AUTHOR(s) Amar Gupta and Stuart Madnick (editors)	8. CONTRACT OR GRANT NUMBER(s) DTRS57-85-C-00083	
9. PERFORMING ORGANIZATION NAME AND ADDRESS Massachusetts Institute of Technology Cambridge, MA 02139	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS	
11. CONTROLLING OFFICE NAME AND ADDRESS Transportation Systems Center Broadway St. Cambridge, MA 02142 (In conjunction with USAF)	12. REPORT DATE December 1987	
	13. NUMBER OF PAGES 342	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Transportation Systems Center Broadway Street Cambridge, MA 02142	15. SECURITY CLASS. (of this report) Unclassified	
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE na	
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release: distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES This volume is one of eight volumes prepared by MIT for the Department of Transportation and Department of Defense (US Air Force).		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Integration, knowledge, databases, systems engineering, methodologies, in- formation modeling, heterogeneous database systems.		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This volume presents practical ideas and prototype systems for integrating existing information and communication resources. It is divided into four parts. The first part, "Knowledge-Based Pictorial Information Systems" describes a prototype system that enables users to manage all types of information, especially pictorial information. The Image Database Manage- ment System (IDBM) provides an integrated standard that can be used to spec- ify the information to be retrieved, in a format acceptable to all		

participating computers.

The second part, "Storage and Retrieval of Pictorial Information in Heterogeneous Computing Systems" studies existing techniques for storing pictorial images, such as bit-mapped mechanisms, vector-based techniques, and quadtree and pyramid-oriented approaches. Recent advances in compression techniques are also discussed.

The third part, "An Expert System for Accessing and Integrating Design Analysis Knowledge" presents an approach for integrating information from multiple design environments. Mechanical design packages such as CADAM and CATIA, thermal design packages such as ITAM and PHOENIS, and other specialized packages can be linked together through a common expert system.

The fourth part highlights a number of critical connectivity issues in the context of data communication networks maintained by two large organizations. The heterogeneity of the user community, their decentralized management structure, the advent of voice, pictorial, and graphical oriented applications are some of the factors that complicate the decision-making process.

INTEGRATING IMAGES, APPLICATIONS, AND COMMUNICATIONS NETWORKS

Amar Gupta
Stuart Madnick
Series Editors

Knowledge-Based Integrated Information Systems
Engineering (KBIIE) Report: Volume 5

Massachusetts Institute of Technology



Accession	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
DTIC /	
DTIC	
DTIC	

A-1

Copyright © 1987 by
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
77 Massachusetts Avenue
Cambridge, MA 02139
All rights reserved.

This book is one of eight volumes published by M.I.T. as part of the Knowledge-Based Integrated Information Systems Engineering (KBIISE) Report. The contents of these eight volumes reflect the work performed by M.I.T. employees and students, as well as by a number of contractors and subcontractors. All requests for permission for photocopying and distribution of these volumes and individuals parts should be directed to the series' editors at M.I.T.

- 3 -

INTEGRATING IMAGES, APPLICATIONS, AND COMMUNICATIONS NETWORKS

About This Volume

This volume presents practical ideas and prototype systems for integrating existing information and communications resources. It is divided into four parts. The first part, *"Knowledge-Based Pictorial Information Systems,"* describes a prototype system that enables users to manage all types of information, especially pictorial information. The Image Database Management System (IDBM) provides an integrated standard that can be used to specify the information to be retrieved, in a format acceptable to all participating computers.

The second part, *"Storage and Retrieval of Pictorial Information in Heterogeneous Computing Systems,"* studies existing techniques for storing pictorial images, such as bit-mapped mechanisms, vector-based techniques, and quadtree and pyramid-oriented approaches. Recent advances in compression techniques are also discussed.

The third part, *"An Expert System for Accessing and Integrating Design Analysis Knowledge,"* presents an approach for integrating information from multiple design environments. Mechanical design packages such as CADAM and CATIA, thermal design packages such as ITAM and PHOENIS, and other specialized packages can be linked together through a common expert system.

The fourth part highlights a number of critical connectivity issues in the context of data communication networks maintained by two large organizations. The heterogeneity of the user community, their decentralized management structure, the advent of voice, pictorial, and graphical oriented applications are some of the factors that complicate the decision-making process.

Table of Contents

	Page
SERIES EDITORS' NOTE	1
KNOWLEDGE-BASED PICTORIAL INFORMATION SYSTEMS (Technical Report #17)	5
STORAGE AND RETRIEVAL OF PICTORIAL INFORMATION IN HETEROGENEOUS COMPUTING SYSTEMS (Technical Report #18)	69
AN EXPERT SYSTEM FOR ACCESSING AND INTEGRATING DESIGN ANALYSIS KNOWLEDGE (Technical Report #10)	111
DATA COMMUNICATION NETWORKS: A COMPARATIVE EVALUATION (Technical Report #15)	259

DEDICATED
TO
THE
NEXT
GENERATION
OF
PROFESSIONALS

SERIES EDITORS' NOTE

This book is one of eight volumes published by MIT as part of the Knowledge-Based Integrated Information Systems Engineering Project (KBIISE). In order to appreciate the papers in this book, it is necessary to be aware about the theme of the KBIISE project, its major objectives, and the different documents that summarize the research accomplishments to date.

Goal

The primary goal of the KBIISE project is to integrate islands of disparate information systems that characterize virtually all large organizations. The number and the size of these islands has grown over years and decades as organizations have invested in an increasing number of computer systems to support their growing reliance on computerized data. This has made the problem of integration more pronounced, complex, and challenging.

The need for multiple systems in large organizations is dictated by a combination of technical reasons (such as the desired level of processing power and the amount of storage space), organizational reasons (such as each department obtaining its own computer based on its function), and strategic reasons (such as the level of reliability, connectivity, and backup capabilities). Further, underlying trends in the information technology area have led to a situation where most organizations now depend on a portfolio of information processing machines, ranging from mainframes to minicomputers and from general purpose workstations to sophisticated CAD/CAM systems, to support their computational requirements. The tremendous diversity and the large size of the different systems make it difficult to integrate these systems.

Key Participants

The above problem is becoming increasingly evident in all large government agencies and in large development programs. In the fall of 1986, the U.S. Air Force (USAF) and the Transportation Systems Center (TSC) of the U.S. Department of Transportation approached M.I.T. to conduct and to coordinate research activity in this area in order "to develop the framework for a comprehensive methodology for large scale distributed, heterogeneous information systems which will provide: (i) the necessary structure and standards for an evolving top down global framework; (ii) simultaneous bottom up systems development; and (iii) migratory paths for existing systems."

Both USAF and TSC provided sustained assistance to members of our research team. In addition, Citibank and IBM provided some funds for research in very specific areas. One advantage of our corporate links was the opportunity to analyze and to generate case studies of actual decentralized organizational environments.

The research sponsors and MIT agreed that in order to deal with the heterogeneity issue in a meaningful way, it was important that a critical mass of influential individuals participate in the development of solutions. Only through widespread discussion and acceptance of a proposed strategy would it become feasible to deal with the major problems. For these reasons, a Technical Advisory Panel (TAP) was constituted. Nominees to the TAP included experts from academic and research organizations, government agencies, computer companies, and other corporations. In addition, several subcontractors, the primary one being Texas A&M University, provided assistance in specific areas.

Technical Outputs

The scope of the work included (i) technical issues; (ii) organizational issues; and (iii) strategic issues. On the basis of exploratory research efforts in all these areas, 24 technical reports were prepared. Eighteen of these reports were generated by MIT research personnel, and their respective areas of investigation are summarized in the figure on the opposite page.

The five technical reports, not represented in the figure, are as follows:

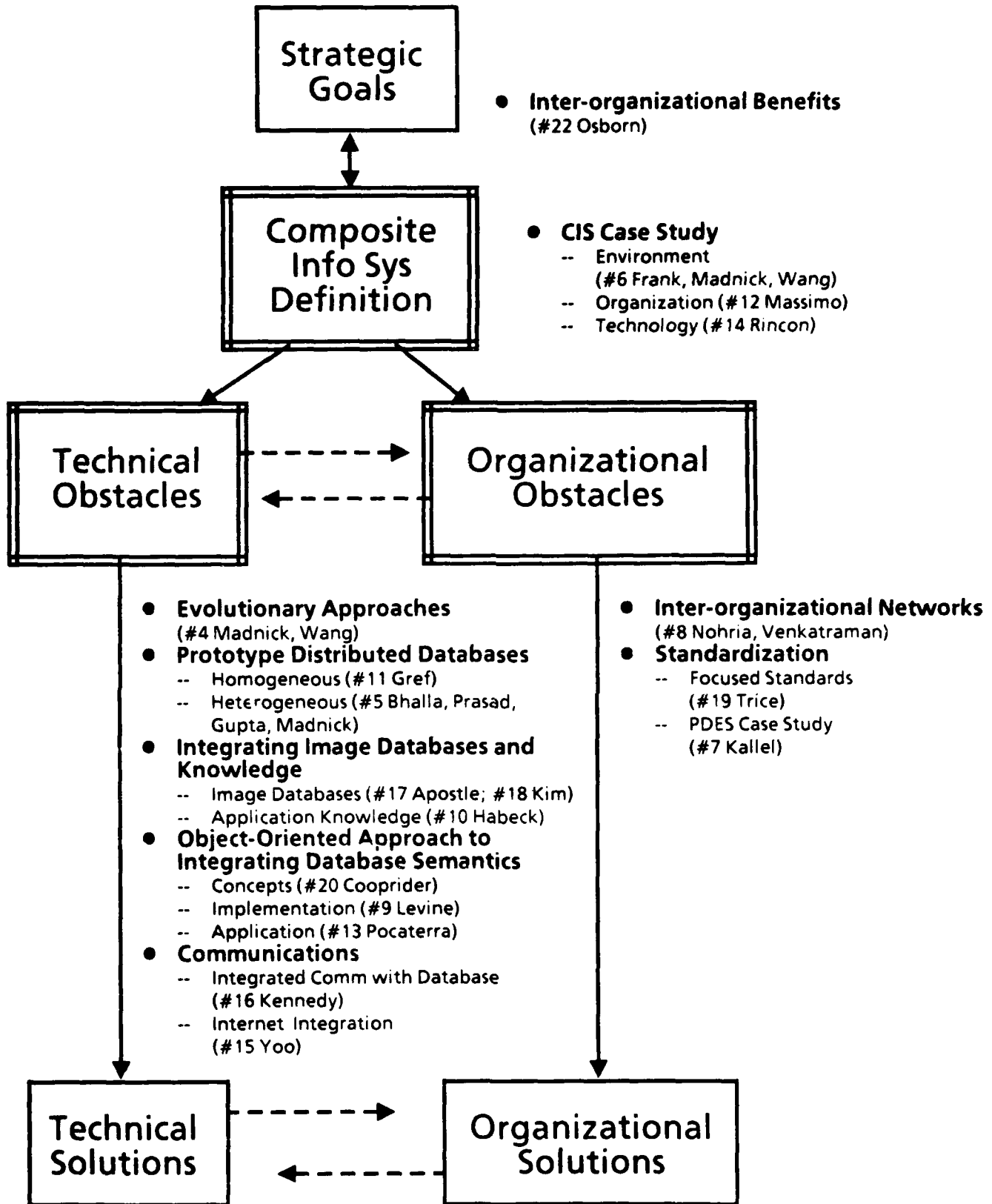
- #1. Summary.
- #2. Record of discussions held at the first meeting of the Technical Advisory Panel (TAP) on February 17, 1987.
- #3. Consolidated report submitted by Texas A&M University.
- #21. Annotated Bibliography.
- #23. Record of discussions held at the second meeting of the Technical Advisory Panel (TAP) on May 21 and 22, 1987.
- #24. Contributions received from members of the TAP highlighting their views on various aspects of the problem.

All the 24 technical reports have been edited and reorganized as an eight-volume set. The titles of the different volumes are as under:

- 1. KNOWLEDGE-BASED INTEGRATED INFORMATION SYSTEMS ENGINEERING-HIGHLIGHTS AND BIBLIOGRAPHY
- 2. KNOWLEDGE-BASED INTEGRATED INFORMATION SYSTEMS DEVELOPMENT METHODOLOGIES PLAN
- 3. INTEGRATING DISTRIBUTED HOMOGENEOUS AND HETEROGENEOUS DATABASES - PROTOTYPES
- 4. OBJECT-ORIENTED APPROACH TO INTEGRATING DATABASE SEMANTICS
- 5. INTEGRATING IMAGES, APPLICATIONS, AND COMMUNICATIONS NETWORKS
- 6. STRATEGIC, ORGANIZATIONAL, AND STANDARDIZATION ASPECTS OF INTEGRATED INFORMATION SYSTEMS
- 7. INTEGRATING INFORMATION SYSTEMS IN A MAJOR DECENTRALIZED INTERNATIONAL ORGANIZATION
- 8. TECHNICAL OPINIONS REGARDING KNOWLEDGE-BASED INTEGRATED INFORMATION SYSTEMS ENGINEERING

Volume 2 contains the report submitted by Texas A&M and Volume 8 highlights the views of members of the TAP. Activities described in the other 6 volumes have been conducted at MIT.

EXPLORATORY RESEARCH EFFORTS



Acknowledgments

Funds for this project have been provided by U.S. Air Force, U.S. Department of Transportation (Contract Number DTRS57-85-C-00083), IBM, and Citibank. We thank all these organizations and their representatives for their support. In particular, we are indebted to Major Paul Condit of U.S. Air Force for his initiative in sponsoring this project, to Dr. Frank Hassler, Bud Giangrande, and Bob Berk of the Transportation Systems Center (TSC) for their support and assistance, to Professor Joseph Sussman, Director, Center for Transportation Studies (CTS) at MIT for his help and encouragement, and to all the individuals whose results have been published in this book.

We would welcome receiving feedback from readers of this book.

Amar Gupta and S.E. Madnick
Massachusetts Institute of Technology
Cambridge, Massachusetts.

KNOWLEDGE-BASED PICTORIAL INFORMATION SYSTEMS

GEORGE APOSTOL, JR.

In a distributed computing environment, it becomes difficult for a user to specify the information to be retrieved, in a format acceptable to all participating computers. This problem is even worse for pictorial data, where a user may recall the contents of a particular image but neither its title nor its source.

To mitigate the above problem, the Image Database Management System (IDBM) has been designed, developed, and implemented at MIT. IDBM has been written in C language and it can operate in a personal computer environment. It contains four modules, whose names and functions are described below.

The specification database contains information about each image. Each image is assigned a unique ID. Because a user is more likely to specify a list of attributes than a list of image IDs an inverted file structure has been used. A five level structure is used to optimize response times.

The syntactic database is used to validate and to decompose a user query. Since the person retrieving images is usually different from the person who initially stores the images, it is unrealistic to assume that the two persons would define a particular image in an identical manner. To mitigate the problem of functionally equivalent words, or synonyms, two dictionaries have been set up. One dictionary contains words of universal importance, and the other dictionary is user defined.

The pictorial database consists of pictures, images, graphs, photographs, maps, and virtually anything that can be displayed on the screen. The system can be easily tailored to accept inputs from all standard computing environments.

The user interface is entirely menu driven. Pull down menus are used. Further, there is an explanation facility that allows users to make minor changes in the selection criteria to increase or to decrease the number of image retrievals of numeric and textual information. The initial prototype was designed to run on a single personal computer. It is now being expanded to run in a multicomputer environment.

Chapter 1

Introduction

1.1 Knowledge and Communication

Throughout history, man has been in search of knowledge. Whether by curiosity or by circumstance, he has gained experience providing him with a range of information, a clear perception of the truth (whatever that may be), and perhaps a simple understanding. However, the range and broadness of the knowledge was far too vast to be kept in the mind, and he was forced to develop ways by which the education could be stored and shared. Thus, languages were developed, number systems were generated, and alphabets were created to assist in saving and communicating ideas and experiences.

Consequently, the knowledge was documented, and people were given the opportunity to learn from others. This eventually led to an information explosion making it nearly impossible for one to acquire and retain even but a fraction of today's enormous amount of information. Hence, once again faced with the problem of information management, man developed the computer to assist in managing what he had learned.

1.2 Computer-Based Information

From the mainframes of the 1960's to the microcomputers of the 1980's, computers have become excellent tools for use in handling large amounts of information. Today, computer-based information plays an important role in our society [20]. In the business environment, for example, computers are used to assist

in many tasks. Focusing on numerical information at the dawn of the computer era and evolving to include processing of textual information, computers have become important and effective additions to the office [29].

Currently, spreadsheet programs are used to analyze data and make forecasts; word processing programs aid in preparing written documents in a quick, elegant manner; and graphics packages help in constructing meaningful charts, graphs, and plots [27]. However, "the ability to process information does not generally aid in the successful communication of information..." [29], and, rather than increased computing power, what is now needed is more sophisticated and effective communication tools. Effective communication pivots around the ability to integrate data imagery with textual, numeric, and graphical representations, and systems should be developed with this in mind. After all, people communicate information graphically [20], and a picture is worth a thousand words.

1.3 The Graphical Presentation Problem

One reason why computers have so far failed to serve as effective communication tools is because traditional computer application software has maintained a strict distinction between different types of information, and, at present, separate application packages are required for processing of numbers, text, and images. A crucial missing component is the ability to present and manipulate visual, pictorial data [29].

The graphical presentation problem is to create a graphical design that effectively communicates the knowledge contained in a computer. The design can be created utilizing the computer's power to manipulate and organize relevant information into a meaningful presentation. Although effective communication

involves a balance between generation and management of images, the focus here is primarily related to the management issues. This includes the efficient storage and retrieval of pictorial information. With this comes the hope of bridging together the gap between current technologies and transforming current systems to support business communications with sophisticated data imagery.

1.4 Overview

This thesis consists of six chapters. Chapter 2 gives a brief summary of current image systems. Chapter 3 describes the Image Database Management System (IDBMS) used in storing and retrieving images in a personal computer environment. Design specifications and implementations are explained and illustrated. Chapter 4 focuses on traditional expert system models, paying particular attention to knowledge representation. Chapter 5 draws parallels and makes inferences between IDBMS and expert systems. Finally, Chapter 6 contains conclusions and directions for future work.

Chapter 2

Current Image Systems

2.1 Introduction

Pictorial information systems are currently entering the mainstream of computer science and engineering. With the advances of new applications in picture processing, such as computed tomography, whole-body scanner, satellite image processing, medical applications, etc., the problem of efficient, economical storage and retrieval of vast amounts of data becomes more important and requires more careful attention. The rapid advances in computer graphics have also given additional encouragement for development of sophisticated pictorial information systems capable of handling picture data.

In designing image systems, there are many factors to be considered. Among these are the problems of storage and retrieval of a large number of pictures, and the storage and retrieval of large pictures or pictures of great complexity. Also to be considered is the intended usage of a pictorial information system -- whether it is intended mainly for the retrieval of pictures, or processing and manipulation of pictures is the main purpose. These considerations can lead to the design of entirely different pictorial information systems. This chapter examines various pictorial information systems and their applications [12, 22].

2.2 Pictorial Information Systems

2.2.1 Computer Aided Design and Manufacturing (CAD/CAM)

CAD/CAM systems automate many analysis and drafting operations associated with product design. The CAD/CAM engineer interacts with a graphics workstation to develop, modify, manipulate, and refine a particular design. As the design develops, the system accumulates and stores geometric and character descriptions of every design element. The design process is speeded up since documentation is systematized and the redrafting of commonly used components is simplified. The graphic data base replaces the paper drawing as the design record. Eventually, a data description may be fed directly to an automated factory which would manufacture a part to its specification.

Currently there are various CAD/CAM systems available commercially. The application areas covered by these systems range from Computer Aided drafting in building architecture [18] to design and manufacture of sculpture surfaces [11]. Another application area is the CAD/CAM of engineering components for the production and mechanical engineering industry. Some better known research and commercially available systems addressing the needs of the engineering industry are PADL [1] and ROMULUS [2]. These systems are rapidly becoming commonplace in the engineering world.

2.2.2 Computer Animation

Animation for engineering provides important pictorial information, and is very different from traditional animation for entertainment. Realistic images are not required, and the high cost and time of production makes them unaffordable. At the same time, there are other requirements which must be met. In engineering

animation, it must be possible to identify, unambiguously, each separate mechanical part in a scene. Also, the animation must be produced quickly: ideally, in real time.

Animation is one of the best ways to visualize the movement of an object. Thus, such systems are useful for determining motion of humans, animals, and robots. By constructing a dynamic model, movement of an object can be analyzed carefully. This assists greatly in the engineering process of designing artificial limbs, examining human mobility, and as a means of communicating the results of assembly simulation [31, 32].

2.2.3 Medical Diagnosis

In the medical field, image processing is fast becoming an effective means for diagnosis. X-ray imaging has become standard for examining broken bones, internal organs, such as lungs, and with the CAT scan, the human brain. However, X-rays are a proven carcinogen and overexposure to radiation is harmful. Hence, other methods of imaging are being explored. Among these is Magnetic Resonance Imaging, a process whereby images are created by detecting the response of isotopes under a magnetic field. This system, not known to have side-effects, is soon becoming a reliable alternative to X-rays.

Image processing is also useful in medical research. Currently, studies are being done at the Brigham and Women's Hospital, Boston, MA, using state-of-the-art image technology. Images are used in analyzing Alzheimer's disease, vertebrae, blood vessels, heart valves, and brain cancer. Although still in the experimental stage, such processes are beginning to show promising results [19, 16].

2.2.4 Geographic Imaging

Computer mapping systems also use pictorial information and are becoming more abundant and useful, especially in the regional planning field. Planners in local governments often use thematic maps in order to explain some ideas. For example, population density maps, housing maps, and geological features maps are often used in explaining the regional planning. Planners also use thematic maps to emphasize their ideas and effectively convey the planners' intentions to map viewers [5].

2.3 Hardware

2.3.1 Large Systems

Traditionally, computer manipulation of images was a task that only large systems could accomplish. Enhanced with the ability of handling huge amounts of data (several million bits per image), image processing was restricted to mainframes - 32-bit computer systems used for supporting large integrated databases [29]. Furthermore, microcomputers were limited by speed, memory capacity, and the cost for permanent storage of large amounts of information [30].

Mainframe computers, packed with computing power, also had more sophisticated imaging systems aided by supporting hardware including laser technology storage media and parallel computing capabilities. Some systems offered 1024x1024 resolution unequalled by microcomputers [27].

2.3.2 Personal Computer Systems

Personal computers, however, have benefited greatly from advances in technology and now offer more basic computing power than the largest computer

did 25 years ago [29]. Also, personal computers add the convenience of portability making them more attractive to the office environment.

Current systems have screen resolutions 800x600 pixels which can show both characters and bit-mapped images. Affordable laser printers and plotters make high quality output possible, and the growing acceptance of hard disks have made feasible the storage of many images. The commercial availability of inexpensive optical disks with storage capacities of 100 Mbytes and more have also contributed to the explosive growth in the use of personal computers [29].

In addition, personal computers systems have been displacing calculators, typewriters, and word processors. Also, with the current technological evolutions, personal computers, supported with graphics capabilities, are replacing overhead and slide projectors. The integration of personal computer technology and efficient graphic methodologies will undoubtedly add to the realm of personal computer applications, as seen thus far in the field of presentation graphics [29].

As price/performance continues to improve, image processing will become an integral part of the office information processing environment; thus, adding a new dimension to the power of personal computers [30].

2.4 Image Retrieval

There are a number of methods of image retrieval. The simplest method is storing the images within the operating system's file structure. This method, however, does not offer easy access to related images, as all are stored randomly.

In answer to this problem, the use of a relational database is becoming a more desirable alternative. Image retrieval under this process, however, can be used only in a narrow domain. Each image is given a fixed number of attributes

that contain particular information specific to each image. But, this method lacks generality since attributes must be determined prior to storage [27].

In pictorial information systems, a large quantity of data is gathered to cover all possible inferences that can be drawn in the future. A user is usually interested in retrieving or updating a very small subset of data, but the probing of a database is usually expensive. The user is also unaware of potential information available in the database, and presents his intent in a set of unstructured queries. It is undoubtedly true that an intelligent query system will be useful, if the system can help the user to locate the information he needs. How to effectively store the information and design a knowledge-based pictorial information system which can support user's retrieval needs remains to be a challenging research problem. The Image Database Management System is one possible solution.

Chapter 3

Image Database Management System (IDBMS)

3.1 Introduction

The development of a general purpose image database is a difficult task. An image database is more than a collection of images. In order to be classified as a database, the set of images must exhibit management qualities. Furthermore, since images contain varied amounts of information, categorizing images into related groups is virtually impossible. Some systems have attempted to extend conventional alphanumeric databases to include images as a data type. This approach, however, proved to be ill-suited since basic attribute-value pairs did not fully describe the range of information found in images [23].

Other difficulties include the problem of subsets of images. Images contain much information on many different levels. An effective database should provide for this multi-layered data. Also, a mechanism must exist enabling the system to address these issues in a microcomputer environment, which inherently possesses lesser computing power and smaller storage capacity than available on mainframe computers and minicomputers [23].

3.2 Design Philosophy

3.2.1 Slides/Pixes

Designed to operate in a microcomputer environment, the Image Database Management System (IDBMS) is an excellent tool for storing and retrieving pictorial information useful for lectures and presentations. The system allows

images with similar characteristics to be grouped together to form a library. Each slide, an image which occupies the entire screen, is associated with a filename. In addition, since a slide may be composed of many subsets, the user is given the capability of defining boundaries to distinguish components of a slide. Each subset is referred to as a pix, and, if the slide contains more than one pix, the pixes are numbered. The unique ID attributed to each image is comprised of the slide name summed with the pix number.

3.2.2 Attributes

Each ID is then described by a set of attributes. Presently, the named attributes are SUBJECTS, EMOTION, ACTION, and PHYSICAL CHARACTERISTICS. The system also allows a modifier to precede each attribute to better describe an element, i.e. old man, new car, etc...

For example, suppose a slide is composed of two pictures - one of a car and one of a boat. Consistent with this systems terminology, the slide contains two pixes and, consequently, two specific IDs. The user identifies the two pixes and describes each using the named attributes. The inputted data is collected and placed in a temporary file. Once the user is satisfied with the descriptions of all the slides/pixes in the temporary file, the data is entered into the database.

3.2.3 Retrieval

Presently, there are three ways by which images can be retrieved. The first is by specifying the library number to view all of the slides/pixes contained in a library. The second is by specifying the unique ID number to display a known slide/pix. Lastly, slides/pixes may be retrieved by specifying a list of attributes describing a particular set desired by the user. Only those slides/pixes satisfying all the conditions are displayed.

The first two methods of acquisition are the easiest - specifying the library number or the image ID. This requires only a simple lookup in the appropriate table to locate the particular slide/pix. The third process, however, is more complex. Selective retrieval of images based on a list of descriptions, although difficult, is the power of this system. However, before describing this process, one must first become familiar with the structure of the system.

3.3 Structure of the IDBMS

The IDBMS consists of four main modules: (i) the specification database which contains information about each image; (ii) the syntactic database used to validate and decompose a user query; (iii) the pictorial database containing the images; and (iv) the user interface routines. The next few sections further describe these modules.

3.3.1 Specification Database

The specification database is comprised of files that contain the information about each slide/pix in the image library. It is organized in five levels using a fully inverted file structure. Level 1 contains the library, image ID, and attributes; level 2 contains the descriptors; level 3 consists of the modifiers; level 4 has the accession list; and level 5 is comprised of the slide IDs. The inverted file structure is used because a user is more likely to specify a list of attributes rather than an image ID to retrieve an image.

Given the typical memory sizes available on microcomputers, the structure of this database offers the best response time despite the limitations.

3.3.2 Syntactic Database

This section contains the vocabulary of the system. The words used in describing the images are contained in one of two dictionaries. The first is the standard dictionary. This is an invariant file that contains a list of words most commonly used and of universal importance.

The other file is known as the application dictionary. This dictionary is a dynamic file that gradually grows in size. It contains words that are not in the standard dictionary that the user wishes to use. Each time the user enters a word describing a particular slide/pix, the standard dictionary is searched. If the word is not found, the application dictionary is searched. If the word is not contained in either of the two dictionaries, the user is given the opportunity to add it to the application dictionary along with any synonyms.

The two dictionaries mentioned above make up the syntactic database. Each entry in the database is of the form (n,w,p) , where 'n' is the unique number assigned to the word, 'w' is the word, and 'p' is the pointer to the synonym. The set of words characterized by 'p' equal to zero are called the base words. Upon entry, the images are coded with the unique number associated with such words. For all other words, the corresponding basic words are assigned as well as the words themselves.

As an example, assume COMPUTING and COMPUTATIONAL are desired to be stored as synonyms. Then the two entries in the dictionary may well appear as $(150,COMPUTING,0)$ and $(180,COMPUTATIONAL,150)$ implying that COMPUTING is the base word and COMPUTATIONAL is a synonym to COMPUTING. In coding, the unique number 150 is used for each image described by either of these two words.

Thus, rock and stone, two functionally-equivalent words, can be used to describe/select the same image. This alleviates the problems that could arise from different people storing and retrieving images. After all, it is unrealistic that two people would define a particular image in an identical manner.

3.3.3 Pictorial Database

The IDBMS does not deal with the issue of creating this portion of the system. The research here has focused in identifying management strategies that can work in conjunction with current graphics software capable of running in an IBM Personal Computer or compatible environment. And, since all routines have been implemented in the C programming language, they can easily be made compatible with many other computing environments.

3.3.4 User Interface

The IDBMS has been constructed as a user-friendly package. Menus are consistently displayed to offer the user assistance and/or advice about commands. Error messages are precise and offer the option of correcting the error. There is a provision to add more attributes without reorganizing the entire database. There are routines by which the user can add more parameters to the attribute list to meet his/her requirements for each particular slide/pix without having to change existing parameters, although that is also an option. Also, delete and recovery facilities are incorporated for individual image ID and whole library. Finally, a fully inverted file structure and hashing methods are used to minimize response time.

3.4 Using The IDBMS

To retrieve images based on attributes, the user enters a query specifying the criteria. For example, the user may want to retrieve all images in the database that contain a home, a tree, and a dog. In this example, the query would be as follows, SUBJECTS: home;tree;dog. The IDBMS first searches through the dictionaries to confirm that the words are valid. If not, the user is notified to correct the error.

Then the system searches and creates a list of all the images corresponding to each description. Next, the system compiles a short-list of images satisfying all descriptions. If desired, the number of images fulfilling individual criterion can be displayed. This enables the user to structure his/her criteria in a manner suitable to his/her needs. Finally, the total number of images that meet all of the selection criteria is displayed.

3.5 Uniqueness of the IDBMS

This system is unique in a number of ways. The first is that IDBMS is designed specifically to operate in a microcomputer environment. Cumbersome mainframes are no longer needed to host image databases.

The database allows multiple values per attribute to be specified at the time of storage and retrieval, and supports the mechanism for automatic checking of synonyms.

Finally, it provides the user with the ability to integrate data imagery with textual, numeric, and graphical representations, until now, a crucial missing component in the business world.

Chapter 4

Traditional Expert System Models

4.1 Introduction

As the following material relates the IDBMS to expert systems, it seems appropriate to first describe some of the current structures of expert systems. This section is devoted to providing an understanding of current implementations and uses of expert systems.

4.2 Definition

An expert system is a device by which knowledge and experience in a specific area of interest is accumulated and organized into a computer. By accessing this computer-based knowledge, an individual is able to obtain "expert advice" about that particular area [13]. Usually, the expert system's knowledge is gained over a long period of time. The process is one of constant, incremental growth and improvement of the knowledge base. "Fundamental and important growth of the system is a process that continues all of its useful life [6]."

Although the knowledge is narrow in domain, expert systems have nonetheless become increasingly popular, for they compile relevant information in a practical manner. Also, the small size of each domain allows for its storage on inexpensive computers. Thus, one is readily able to access the knowledge to use as an aid in understanding the specifics of that field. It must be stressed, however, that the goal of an expert system is not to replace the person in the field. Rather, it should only be used as a tool for managing and acquiring knowledge [13, 27].

4.2.1 Engineering Criteria

For the expert system to be effective, it must adhere to a set of simple criteria outlined by Davis in [6]. These "basic commandments" are important in developing a keen understanding of expert systems and the reasoning necessary for determining the success of a system.

The first, and perhaps the most fundamental observation - *In the knowledge lies the power* - suggests that success of an expert system lies in the quantity and quality of the knowledge, as opposed to rules that embody the information; hence, problem-solving performance should be based on the extensive amount of knowledge of the task at hand.

That *the knowledge is often inexact and incomplete* must be noted, for problems faced by expert systems rarely have complete laws or theories governing their solutions.

The knowledge is often ill-specified, because the expert himself does not know the amount of relevant information necessary for creating a perfect system. In fact, he may not even know how much he knows!

The statement *amateurs become experts incrementally* stands to say that knowledge must be acquired slowly.

And finally, *expert systems need to be flexible and transparent*. Since most systems must undergo several revisions, flexibility is essential in contributing to the ability of the system to undergo changes easily. Transparency is also important: how can the system be improved if its methods are unknown?

4.2.2 Architectural Specifications

In addition to the basic criteria, architectural specifications have also become important considerations. One suggestion is to *separate the inference engine and the knowledge base*. This makes the knowledge more understandable, accessible, and more easily identifiable. If the knowledge base is intermixed with the inference engine, changes to correct errors become less clear and thus, flexibility suffers.

Uniformity of representations also lends to a simpler, more transparent system. By reducing the number of required mechanisms to handle the knowledge, systems are less complicated and more easily understood.

To require less work in determining exactly what knowledge is needed to improve system performance, *the inference engine must be kept simple*. This also assists in keeping reasoning direct and to the point. When the inference engine is less complicated, explanations are easy to produce.

Lastly, *exploit redundancy*. By using multiple overlapping sources of data, a more abundant and robust collection of knowledge is obtained thereby reducing errors caused by incomplete and inexact information.

4.3 Knowledge Representation

4.3.1 Rule-Based Systems

Traditionally, knowledge in expert systems is represented as a set of rules based on compiled experience. By collecting results of case studies, a knowledge base is constructed containing information about a specific area of study. The shape and character of the design space are determined by examining many case studies. This leads way to empirical observations which may be used in determining which parts of the space make sense for which kinds of problems. Thus, the knowledge is well-captured as a collection of informal rules of thumb [6].

This knowledge base grows dynamically as more and more experts include their experiences. Consequently, the knowledge base becomes a domain of empirical associations - rules that encode the experience of accomplished diagnosticians [7]. These rules are of the form, IF IN <situation>, THEN <action> IS TAKEN [3]. Naturally, the rules are modified as the system acquires more knowledge. Once the knowledge has been assembled, it can then be readily accessed through the user interface which allows the user to utilize the computer algorithms constructed to retrieve the desired information.

However, the rule-based expert system is nothing more than a data base of pattern-decision pairs which, when queried, simply solves the problem [3]. Furthermore, the use of empirical associations precludes any more substantive form of explanation. In rule-based systems, all that the system knows is that "A and B suggest C." But, the system does not have the capability of answering why? beyond repeating the rule.

What is now needed is a means by which the knowledge may be represented in a manner so as to allow the system to make inferences based on reasoning rather than experience, because rule-based systems are inflexible and focus only on the rules embodying empirical associations. Rule-based systems do not offer any tools for constructing structural descriptions, techniques for using descriptions to guide diagnosis, nor do they provide further insight into the specific problem at hand, other than the rules themselves.

4.3.2 Reasoning from Structure and Function

In answer to the cry for a more flexible and deeper structure, a new wave of expert system was developed based on a different form of knowledge representation. With this, the system gained the power of diagnostic reasoning based on the ideas of

structure, the "anatomy" of the system, and function, the "physiology" of the system. The previous form of the rules was transformed into rules of the form, IF IN <situation>, THEN <action> IS TAKEN, AND <situation> WILL FOLLOW, adding the ability of reasoning from first principles. This new form of rules, if you will, is based on the idea that understanding a domain often corresponds to an ability to deduce consequences of events that may occur in the domain [3].

Hence, the expert system became a "learning system" acquiring knowledge through the "difficult and important work of enumerating and organizing models of causal interaction [13]." Whereby the rule-based system must acquire knowledge on a case-by-case basis, this revolutionary expert system requires knowledge of only structure and behavior. Given these specifications, the system is enhanced with the ability of diagnosis and reasoning as opposed to the theory of test generation most commonly used by rule-based expert systems. The Image Database Management System exemplifies this new and exciting class of expert systems.

Chapter 5

The IDBMS as an Expert System

5.1 Introduction

In its present form, the Image Database Management System cannot be defined as a "classical expert system." It does, however, exhibit traits attributed to expert systems while incorporating many of the ideas used in constructing expert systems. The use of these traits and ideas is what makes the IDBMS an excellent device for accumulating and organizing pertinent information. The structure of the knowledge coupled with the function of the package lead to the efficient storage and retrieval of pictorial information.

The following chapter explains the relationship between the IDBMS and expert systems. It begins by relating the anatomies of the systems concentrating on the four major aspects of expert systems. Then, the IDBMS is evaluated based on the engineering criteria and architectural specifications outlined earlier. Lastly, possible uses of the IDBMS as an expert system are explored.

5.2 Anatomy

5.2.1 The Knowledge Base

In general, most expert systems consist of four modules. The first, is the **Knowledge Base**. This necessary and important module contains the formal representation of the information provided to the system. It deals with the structures used to represent the knowledge provided by the "expert." Because there is no single global structure to represent knowledge in the most effective scheme,

the manner in which the knowledge is presented is crucial. Efficient knowledge representation is the key to the success of the overall expert system [13].

The knowledge base of the IDBMS consists of the images stored in the database and the attributes assigned to them. Each image can be characterized by the concept of **frames**. Originally proposed by Marvin Minsky in [21], this scheme decomposes knowledge into highly modular pieces consisting of concepts and situations, attributes of concepts, and relationships between concepts.

The images of the IDBMS are best described in this manner because they contain knowledge concerning different ideas on many different levels. Each slide has several kinds of information. Whether a stereotyped situation, like a birthday party or a living room, a re-occurring theme, as in Christmas, a provocative idea, as in politics or unemployment, or a simple object, like a cow, the information in each slide is unstructured and abundant. Furthermore, lower levels of a slide, the tires of a car for example, also add to the myriad of data. Hence, by viewing the knowledge as frames, the discrepancies between the different types and levels of data are reduced, thereby creating a more uniform group of representations.

5.2.2 The Inference Engine

The next module of importance is the **Inference Engine**. It is responsible for interpreting the contents of the knowledge base. Sometimes called the "brains" of the system, it is composed of three parts - the context block, the inference mechanism, and the explanation facility [13].

5.2.2.1 Context Block

The context block is the current state of the problem and the solution. In the IDBMS, the problem is to retrieve the desired images from the database. The method of solution is determined by the user in the form of the query. The system

interprets the user's desired method of retrieval and determines the solution - a list of images.

5.2.2.2 Inference Mechanism

The inference mechanism employs the reasoning used in searching the knowledge in order to reach a goal or conclusion. The reasoning power of the IDBMS lies in the use of the attributes contained in the syntactic database.

The important feature of the syntactic database is its use of synonyms. Upon entry, each slide is divided into pixels and given a specific set of attributes describing each frame. However, on retrieval, it is not apparent which attributes were used in storing the images. Thus, the synonyms become an important part of the system. With this feature, recognition of frames can be organized into hierarchies. The system can hypothesize at many levels, from the very general to the very specific: an animal, a dog, a collie, lassie.

The level of complexity depends on how the dictionaries are created. When constructed, each synonym is given a pointer to a base word. The reasoning used in assigning a base word to a synonym determines how "deep" the IDBMS searches. Each level has its own recognition attribute, but the more specific attributes also include the information contained in the more general attributes above them. Thus, if "dog" was a synonym whose base word was "animal", then when specifying "dog" as a criterion for retrieval, not only would the system return those frames tagged with "dog", it would also return the images marked with "animal". Whether or not this is an acceptable solution must be determined by the expert creating the database.

Currently, the base words of the system do not contain pointers to other words, thereby making the line of reasoning simple. However, the IDBMS places no

restrictions on the number of synonyms attached to each base word. This gives the system immense flexibility, but also creates a potential for problems, and words must be chosen effectively in order to deal with noisy, confused, and unanticipated situations.

On the other hand, the use of synonyms adds to the knowledge of the system. Functionally equivalent words can be used freely without concern, and users need not know the attributes used in describing and storing each slide. The structure of the database in conjunction with the function of the IDBMS handle these problem easily and effectively.

5.2.2.3 Explanation Facility

The explanation facility is the final link in the inference engine of the IDBMS. This task is accomplished by allowing the user to view the images retrieved by a single criterion as well as those which satisfy a list of criteria. The line of reasoning is explained by the system when displaying the selected frames.

As before, this module (the inference engine), although closely related and dependent upon the knowledge representation, must be unique and kept separate from the knowledge base [13]. The IDBMS does this precisely, i.e. the image database, the specifications database and the syntactic database are all stored completely independent of each other.

5.2.3 The Knowledge Acquisition Facility

The **Knowledge Acquisition Facility** deals with the generation of new knowledge. This portion of the expert system has been implemented in many ways; however, a common aspect shared by all systems is the time spent in creating it. For optimal utilization of human experts, the encoded knowledge must grow slowly on a continuing basis. Also, the expert system must provide good facilities for improving upon its knowledge base, or it will become obsolete and irrelevant [13].

This section of the IDBMS is exhibited by the various routines used to insert and/or modify existing data. Insertion, deletion, and recovery facilities are incorporated for individual image IDs and libraries, and there are provisions for adding more attributes in the future without reorganizing the entire database. The IDBMS assumes that the user may be interested in adding more parameters to the attributes list to meet his/her requirements for a particular slide/pix and provides mechanisms for use in achieving these goals. Lastly, implementing the syntactic database with two dictionaries enables the user to insert new base words along with any corresponding synonyms.

5.2.4 The User Interface

The fourth module of the expert system completing the anatomy is the user interface. This module permits the user to benefit from the system. This portion of the IDBMS has previously been described in Chapter 3 of this document.

5.3 System Evaluation

The criteria used in evaluating the effectiveness of the IDBMS as an expert system have been outlined in Chapter 4. The following summarizes each criterion and shows how the IDBMS has satisfied each requirement.

The first criterion requires that the expert system have a large quantity and a high quality of knowledge. In implementing the IDBMS, the images used to demonstrate the strategies were acquired from VCN ExecuVision. This software package was selected since it offers the largest number (over 4,000) of prerendered images. Independent reviewers have highly recommended it as "The Cadillac of Presentation Graphics Software" [29] and "What a word processor is to words, VCN ExecuVision is to graphics" [15]. This package uses sophisticated data imagery satisfying the needs of quantity and quality in the knowledge base.

The requirement of acquiring knowledge slowly is fulfilled by the knowledge acquisition facility. The routines in this section allow for incremental gain of knowledge. The application dictionary of the syntactic database also contributes to this process.

The flexibility and transparency of the system is evident from the different aspects of the IDBMS described earlier. The synonyms, acquisition facility, and knowledge representation each contribute to the flexibility of the system. The transparency is exemplified by the explanation facility showing the line of reasoning used in determining solutions to the queries.

The IDBMS also satisfies the architectural specifications. The inference engine is separate from the knowledge base; characterizing the images as frames lends to the transparency of the system by contributing to the uniformity of representations: requiring that base words have no pointers to synonyms keeps the inference engine simple: and finally, redundancy is exploited by allowing the use of multiple attributes to describe and store slides/pixes.

5.4 Uses for the IDBMS

The Image Database Management System is more than a way to store images in a computer. It is packed with knowledge and can be used to prepare meaningful presentations. Though not an expert system, it contains a sufficient amount of knowledge to provide the user with a framework in using the art of imagery. With the aid of this package, users can facilitate the design process of creating an elegant and concise method that effectively communicates the knowledge contained in the computer: thus, offering one possible solution to the graphical presentation problem.

Chapter 6

Discussion

6.1 Introduction

The purpose of this thesis is to explore the possibilities in the emerging field of presentation graphics and to solve the graphical presentation problem of effectively communicating data contained in a computer. A description has been provided explaining the structure and function of the Image Database Management System used in storing and retrieving images in a microcomputer environment. This system is unique in that it allows multiple values per attribute to be specified at the time of storage and retrieval; it supports the mechanism for automatic checking of synonyms; and it provides the user with the ability to integrate data imagery with textual, numeric, and graphical representations. An explanation of current technologies of expert systems has also been given, and these concepts have been applied to the IDBMS showing that it could be used as an aid in extracting an abundance of information, intelligently.

The remainder of this chapter emphasizes specific research contributions, describes the major limitations of this research, speculates on future developments of the system, and finally, contains concluding remarks.

6.2 Contributions

The attributes used in describing the images contained in the database signify the inter-relationship among information that these values represent. These inter-relationships are essential so that the system does not just present the information requested by the user but also determines intelligently the interest domain of the user-query session. Such domain helps in presenting the user with additional information at a very low cost and in some ways leads the user to a set of useful database values.

The Image Database Management System provides this service in the use of the static dictionary. This dictionary has been constructed from commonly used nouns of the English language. In particular, each word has been chosen on the basis of its relationship to graphical images. The ability to visualize the word is a major consideration in entering it into the dictionary. Moreover, common knowledge of the word is also important. It seemed impractical to enter words one would never use and would only make the system unnecessarily larger without enhancement.

Practicality is also a criterion in distinguishing base words from their synonyms. For example, "telephone" has been chosen as the base word with synonym "phone". This choice has been made to exploit the function of the system. If an image were coded with the base word "telephone", then upon retrieval, specifying "phone" or "telephone" would return those entries containing "telephone", a desirable solution. However, had "phone" been chosen as the base word, retrieval based on "phone" would exclude those entries tagged with "telephone". Since "telephone" is a more universal word than "phone", it makes sense that it should be higher in priority.

6.3 Limitations

The major limitation of this system is in its inability to handle the asymmetry of the English language. In its present form, the system is unable to solve the problem of homographs. A homograph is a word that has multiple meanings, for example, a "ball". In many interpretations, this word refers to a round object used as a toy. This is its current implementation, also serving as a base word to particular kinds of balls - a softball. However, a "ball" could also refer to a formal dance. If used in this manner, the user would not be satisfied with the outcome.

Colloquialisms are also a potential for problems. Webster's dictionary defines "semi" as meaning "half in quantity or value." In some circles, though, "semi" is used to refer to a large truck. Possible solutions to these problems are outlined in [27] and could be useful in this system.

An important feature of this system is its ability to manage textual, numeric and graphical information. This is accomplished by treating each type of information as a slide. With the current implementation, however, numbers and text are treated as images, and the system loses the ability of distinguishing between components of the slide. Thus, individual numbers and lines of text cannot be accessed. Ideally, the system should not only provide for the management of data; it should also offer, separate from the database, the option of processing the information. Therefore, the data structures of the information should be sufficiently flexible.

6.4 Future Work

More research is needed in constructing a syntactic database that functions in a logical manner. Since the knowledge of the system can overlap and tangle in interesting ways, how to represent these entanglements and what to do about them are problems that require further thought. The fact that the function of this system is highly user independent warrants the need for a communication medium that reflects the locality of the system. User preferences must be established. An optimal solution would be to implement a standard dictionary, or thesaurus. Such systems are available for computers today, as shown by the use of spelling checkers in some word processing environments, and would standardize the syntactic database.

Also, upon retrieval, the IDBMS currently handles the AND function. Given a list of attributes, it returns only the images which meet all the criteria. Another useful implementation would be to provide a facility to retrieve information based on the logical operator OR. This would return an aggregate of the images meeting each of the criteria individually. Presently, to achieve this result, each attribute must be specified one at a time.

Currently, the IDBMS is a local storage database: that is, it searches for information that is contained within the personal computer. Future implementations could provide for a means of searching through a network of systems. With this, it would be able to access much more information than can be stored on one system. Protocols could be developed to transfer and receive data preserving knowledge representation and allowing equal access to the "minds" of many systems.

The structure and function of the IDBMS lay the foundation for creating such possibilities. With its ability to process different types of information, the IDBMS

provides a uniform representation of the knowledge. This uniformity lends to the ease of manipulating varied amounts of information and makes file transfers across networks less complicated. Discontinuities between different computer architectures remains to be a problem, but with a more structured knowledge base, systems can be tailored more easily.

The lack of a unified format for describing information is also a problem in communicating data over different systems. However, The IDBMS has the potential for solving this problem. Once again, the importance of the synonyms in the syntactic database becomes apparent. If a method were established by which the descriptors used in tagging information could be transparent to the users, lack of uniformity would no longer be a problem. Part of this problem can be solved by standardizing the syntactic database, but more work is still needed to deal with asymmetries in the English language.

Essentially, this system could be used to create a more formal version of an expert system. By combining it with statistical information pertaining to effective human/computer communication, the work here can be expanded to include more specific solutions to more complex problems. Work must be done, though, in creating a query system with the ability to tailor itself to the needs of the user. Networking problems may arise from lengthy strings, noise, and the transfer of large amounts of data, and a dependable means of communication must be established. Another problem is the variance in languages used by people in communicating their ideas. The system must be able to intelligently decipher the terminology and devise adequate solutions. User queries should be decomposed and interpreted effectively so as to minimize retrieval of meaningless data and maximize acquisition of relevant textual, numerical, and graphical information.

Using ideas developed in the fields of Artificial Intelligence, Natural

Language Processing, and Speech Recognition, this system could eventually become an excellent communication device, packed with the ability to understand and relay information across various mediums. Enhanced with more reasoning ability, it could become a powerful learning device.

6.5 Concluding Remarks

The research done here attempts to capture expert knowledge into a personal computer in an effort to manage the data in a manner advantageous to the business environment. In the field of computer graphics, the concept, which designers should consider in determining how they can incorporate their ideas into pictures, is important. and, up until now it has not been possible for the user to exploit the abundance of information contained in a personal computer. Hopefully, the work here will assist in constructing a useful relationship between man and computer needed for more sophisticated and effective communication tools.

Appendix A

Static Dictionary

ABDOMEN	BELLY	STOMACH	
ACTOR	MIME	PERFORMER	PLAYER
ADDITION	INCREMENT	RAISE	
ADMINISTRATOR	EXECUTIVE	MANAGER	OFFICIAL
ADVANCE	ADVANCEMENT	PROGRESS	
ADVERTISEMENT	ADVERTISING	PROPAGANDA	PUBLICITY
ADVICE	ADVISEMENT	COUNSEL	
AGENT	BUYER	DELEGATE	MIDDLEMAN
AIRPLANE	HELICOPTER	PLANE	
ALCOHOL	BOOZE	DRINK	LIQUOR
ALPHABET	ABC		
AMATEUR	APPRENTICE	BEGINNER	ROOKIE
AMBITION	ASPIRATION	EAGERNESS	
ANGER	FURY	MAD	RAGE
ANIMAL	BEAST	CREATURE	
ANIMATION	CARTOON		
ANNEX	EXTENSION		
ANNOUNCEMENT	DECLARATION		
ANNUAL	YEARBOOK		
ANSWER	REPLY	RESPONSE	
ANTENNA			
APE	BABOON	GORILLA	MONKEY

APOLOGY	DEFEND	EXCUSE		
APPENDIX	ADDENDUM	SUPPLEMENT		
APPETIZER	ANTIPASTO			
APPLAUSE	CLAP	OVATION		
APPLICANT	CANDIDATE	NOMINEE		
APPOINTMENT	RESERVATION			
ARC	ARCH	CURVATURE	CURVE	
ARCHITECT	BUILDER			
AREA	REGION	TERRITORY	ZONE	
ARGUMENT	DISAGREEMENT	DISPUTE		
ARITHMETIC	CALCULATION	COMPUTATION	MATH	
ARMORY	ARSENAL			
ASH	ASHES	CINDER	SOOT	
ASSASSIN	GUNMAN	HIT MAN	MURDERER	
ASSISTANT	AID	AIDE	ATTENDANT	HELPER
ATHLETICS	GAMES	SPORTS		
ATTACK	ASSAULT	BATTLE		
ATTORNEY	LAWYER			
AUTHOR	WRITER			
AUTOMOBILE	AUTO	CAR		
BABY	CHILD	INFANT	YOUTH	
BACKPACK	KNAPSACK	PACK		
BAG	POUCH	SACK		
BALL	BASEBALL	BASKETBALL	FOOTBALL	SOFTBALL
BALLOON				

BALLOT	VOTE			
BAND	ORCHESTRA	SYMPHONY		
BANISTER	RAILING			
BANK	TREASURY			
BAR	PUB	SALOON		
BARBER	BEAUTICIAN	COSMETOLOGIST	HAIRCUTTER	HAIRDRESSER
BARTENDER	BARKEEPER	BARMAID		
BEACH	COAST	SHORE		
BEARD	WHISKERS			
BED				
BEDSPREAD	BEDCOVER	QUILT	SHEET	
BEGGAR	BUM	MOOCHER		
BELT	SASH	SUSPENDERS	WAISTBAND	
BET	STAKE	WAGER		
BICYCLE	BIKE	CYCLE		
BIRTHMARK	MOLE			
BLADE	KNIFE	RAZOR		
BLENDER	MIXER			
BOAT	SHIP			
BOMB	DYNAMITE	MISSILE	TORPEDO	
BOOK				
BOTTLE	JAR			
BOX	CARTON			
BOXING	FISTICUFFS	PRIZEFIGHTING		
BOY	LAD	LADDIE	SON	

BRIDGE			
BROACH	BROOCH	CLIP	PIN
BRUISE	CONTUSION		
BUILDING	STRUCTURE		
BULLFIGHTER	MATADOR	TOREADOR	
BURIAL	ENTOMBMENT	INTERMENT	
BUS	COACH		
BUSINESS	COMMERCE	INDUSTRY	TRADE
BUSINESSMAN	DEALER	MERCHANT	
CABINET	CUPBOARDS		
CADAVER	CORPSE		
CALENDAR			
CANDLE			
CANYON	VALLEY		
CAPTION	LEGEND		
CARPET	RUG		
CASSETTE			
CAT	KITTEN		
CAVE	CAVERN		
CEMENT	CONCRETE		
CEMETERY	GRAVEYARD		
CHAIR	COUCH	SEAT	SOFA
CHALKBOARD	BLACKBOARD		
CHANNEL	PIPELINE		
CHARACTER	MARK	SIGN	

CHART	GRAPH	MAP		
CHECK	BILL	INVOICE		
CHESSE				
CHIMNEY	SMOKESTACK			
CHRISTMAS	NATIVITY	YULETIDE		
CHURCH	MONASTARY	STEEPLE		
CIGAR	CIGARETTE			
CLERGYMAN	MINISTER	PREACHER	PRIEST	REVEREND
CLOCK	CHRONOMETER	WATCH		
CLOSET				
CLOTHES	BLOUSE SHIRT	CLOTHING	DRESS	PANTS
CLOUDS				
CLOWN	COMEDIAN	COMIC	FOOL	
CLUB	ALLIANCE	ASSOCIATION	LEAGUE	UNION
COAT	JACKET	SWEATER		
COLLISION	ACCIDENT	WRECK		
COLOR	HUE	PIGMENT	TINT	
COMMENT	REMARK			
COMMUNICATION	DISCUSSION			
COMPANION	MATE	SPOUSE		
COMPUTER	CALCULATOR	MAINFRAME		
CONDIMENT	PEPPER	SALT		
CONFERENCE	CONVENTION	MEETING	SEMINAR	
CONTEST	COMPETITION			
CONTRACT	DEAL	PACT	TREATY	

CONVERSATION	DIALOGUE	TALK	
COPY	CARBON	DITTO	REPRODUCTION
CORPORATION	CARTEL	COMPANY	
COSMETICS	MAKEUP		
COSTUME	DISGUISE	MASK	VEIL
COUPLE	DUO	PAIR	
COW	BULL	CALF	
CRIMINAL	CONVICT	FELON	VILLAIN
CRITIQUE	REVIEW		
CROSSWALK			
CROWD	MOB		
CURE	ANTIDOTE	REMEDY	VACCINE
CURTAIN	DRAPES		
CUSTOMER	CLIENT	PATRON	SHOPPER
DANCE	BALLET		
DANCER	BALLERINA		
DANGER	HAZARD	PERIL	
DEATH	CASUALTY	FATALITY	
DEER	FAWN		
DEVIL	DEMON	FIEND	SATAN
DIPLOMA			
DISEASE	ILLNESS	SICKNESS	
DOCTOR	DENTIST	PHYSICIAN	SURGEON
DOG	CANINE	PUPPY	
DONKEY	BURRO	JACKASS	

DRAWERS	DRESSER	VANITY	
DREAM	FANTASY	NIGHTMARE	
DRIVER	MOTORIST		
DRUGS	COCAINE	MARIJUANA	
EARTH	PLANET	WORLD	
EDUCATION	TEACHING	TRAINING	
ELECTRONICS	CIRCUITS		
ELEVATOR	ESCALATOR		
EMIGRANT	ALIEN	IMMIGRANT	MIGRANT
EMPLOYMENT	JOB	WORK	
ENGINE	MOTOR		
ENJOYMENT	PLEASURE	RECREATION	
ENTERTAINMENT	AMUSEMENT	PERFORMANCE	RECITAL
ENTRANCE	DOOR		
ENVIRONMENT	ATMOSPHERE	SURROUNDINGS	
EQUIPMENT	MACHINERY	MATERIALS	
ESCORT	USHER		
EVENING	SUNDOWN	SUNSET	
EXPERIMENT	TRIAL		
FABRIC	CLOTH		
FACE			
FACT	REALITY		
FACTORY	MILL		
FAILURE	DEFEAT	UNSUCCESSFUL	
FAME	CELEBRITY	NOTORIETY	

FAMILY	FOLKS	PARENTS	RELATIVES	
FARMING	AGRICULTURE	AGRONOMY	CULTIVATION	
FASHION	STYLE	TREND	VOGUE	
FATE	CIRCUMSTANCE	DESTINY		
FATHER	DAD			
FATIGUE	EXHAUSTION	WEARINESS		
FAUCET	HYDRANT	SPIGOT	VALVE	
FEAR	FRIGHT	HORROR	TERROR	
FICTION				
FIELD	PASTURE			
FINANCE	BONDS	STOCKS		
FIRE	BLAZE	BURNING	FLAME	TORCH
FIREPLACE				
FIREWORKS	FIRECRACKER			
FISH				
FLAG	BANNER	PENNANT		
FLASH	TWINKLE			
FLOWER	BLOSSOM			
FOAM	LATHER	SUDS		
FOOD	NOURISHMENT			
FOOT	FEET			
FOREHEAD	BROW			
FOREST	TIMBER	WOODS		
FRAGRANCE	AROMA	PERFUME	SCENT	SMELL
FREEDOM	LIBERTY			

FRIEND	ACQUAINTANCE	ASSOCIATE	COLLEAGUE	PARTNER
FRISBEE				
FROG				
FRONTIER				
FRUIT	APPLE	BANANA	ORANGE	
FRYING PAN	SKILLET			
FUTURE				
GARAGE	CARPORT			
GARBAGE	FILTH TRASH	LITTER	RUBBLE	SEWAGE
GARDEN				
GENIUS	INTELLECT	INTELLIGENT		
GESTURE	SIGNAL			
GHOST	APPARITION	PHANTOM		
GIANT				
GIFT	PRESENT			
GIRL	LADY	WOMAN		
GLASS	CUP	MUG		
GLASSES	SPECTACLES	SUNGLASSES		
GORGE	CHASM	RAVINE		
GRADUATION				
GRASS	LAWN			
GRAVE	COFFIN	TOMB	TOMBSTONE	
GREETING	HELLO	SALUTATION		
GUARD	SENTRY	WATCHMAN		
GUEST	VISITOR			

GUN	PISTOL	RIFLE	
GYMNASIUM	GYM		
HAIR			
HANDKERCHIEF			
HANDWRITING	CALLIGRAPHY	PENMANSHIP	
HANGAR	HANGER		
HARBOR	PORT		
HAT	CAP	HELMET	
HAZE	MIST		
HEAD			
HEADLINE	HEADING		
HEART			
HEAVEN			
HELL	HADES		
HISTORY	CHRONICLE		
HOLE	CAVITY	GAP	
HOMOSEXUAL	FAG	LESBIAN	QUEER
HORSE	PONY		
HOSPITAL	CLINIC	INFIRMARY	
HOTDOG	FRANKFURTER	WIENER	
HOTEL	LODGE	MOTEL	RESORT
HOUSE	HOME	LODGING	SHELTER
HUMAN	PEOPLE	PERSON	
HUMOR	SATIRE		
HUSBAND			

HUT	SHACK			
ICE				
IMPOSTOR	IMPOSTURE	PHONY		
INCOME	REVENUE	SALARY		
INHIBITION	TABOO			
INITIALS	INSIGNIA	MONOGRAM		
INSPECTOR	INVESTIGATOR	SPY		
INSTRUMENT				
INTEGER				
INTELLIGENCE				
INTERIOR				
INVASION				
INVENTION				
INVENTOR	INNOVATOR			
INVENTORY	STOCK	SUPPLY		
JAIL	PRISON			
JUDGE				
JUNGLE				
KEYS	KEYCHAIN			
LANGUAGE	DIALECT	SPEECH	TERMINOLOGY	VOCABULARY
LAW	REGULATION	RULE		
LEADER	CHAIRMAN	DICTATOR	PRESIDENT	
LEAVES	FOLIAGE			
LECTURE	SERMON			
LEGISLATURE	CONGRESS	SENATE		

LIBRARY

LICENSE

PERMIT

LIGHT

FLASHLIGHT

LAMP

LIGHTBULB

LIGHTNING

THUNDER

THUNDERBOLT

LIST

ROSTER

LOVE

AFFECTION

DEVOTION

LOVER

BOYFRIEND

GIRLFRIEND

SWEETHEART

LUCK

LUGGAGE

HANGBAG

SUITCASE

LUXURY

FRILL

MAGIC

SORCERY

WITCHCRAFT

MAGICIAN

WARLOCK

WITCH

MAID

HOUSEKEEPER

SERVANT

MAIL

LETTER

POSTCARD

MAN

GENTLEMAN

MANSION

CASTLE

MANOR

MANUAL

HANDBOOK

INSTRUCTIONS

MARRIAGE

MATRIMONY

WEDDING

WEDLOCK

MAZE

MEAL

BANQUET
SUPPER

BREAKFAST

DINNER

LUNCH

MEAT

BEEF

FLESH

STEAK

MEDICINE

DRUG

MEDICATION

MENU

MERCHANDISE

MESSAGE

MEMO

NOTE

TELEGRAM

MESSENGER	COURIER			
MIDGET	DWARF			
MILITARY	AIR FORCE SERVICE	ARMY	MARINES	NAVY
MIND	BRAIN			
MIRAGE	DELUSION			
MIRROR				
MISER	SCROOGE	TIGHTWAD		
MISSIONARY	EVANGELIST			
MIXTURE	BLEND	MIX		
MODEL	EXAMPLE			
MODERATOR	ARBITRATOR			
MONEY	CASH DOLLARS	CENTS	COINS	CURRENCY
MONUMENT	MEMORIAL	STATUE		
MOON				
MORALS	ETHICS			
MORNING	DAWN	SUNRISE		
MORTICIAN	CORONER	UNDERTAKER		
MOTHER	MOM			
MOTION	MOVEMENT			
MOTORCYCLE	MINIBIKE	MOTORBIKE		
MOUNTAIN	HILL			
MOUSE	RAT			
MOUTH				
MOVIE	FILM	SHOW		

MURDER	KILLING		
MUSCLE			
MUSEUM	GALLERY		
MUSICIAN			
NAME	TITLE		
NAP			
NATURE			
NECKLACE	CHARM	PENDANT	
NEWS	INFORMATION		
NIGHT			
NOODLE	SPAGHETTI		
NOSE	SNOUT		
NOSTALGIA			
NOTEBOOK	FOLDER		
NUMBER	NUMERAL		
NURSE	BABYSITTER		
OBITUARY			
OBSERVATORY	LOOKOUT		
OBSERVER	AUDIENCE	SPECTATOR	
OBSTACLE	OBSTRUCTION		
OCEAN	SEA		
OFFICE			
OINTMENT	BALM	CREAM	SALVE
ORIGINAL	PROTOTYPE		
OUTPUT	YIELD		

OUTSIDE	OUTDOORS			
OWNER	PROPRIETOR			
OYSTER				
PAIN	ACHE			
PAINT				
PAJAMAS	NIGHTGOWN			
PAPER				
PARADE				
PARCEL	PACKAGE			
PARK	PLAYGROUND			
PARTICLE	ATOM	MOLECULE		
PARTY	CELEBRATION	FIESTA		
PASSPORT	VISA			
PASTRY	CAKE	COOKIES	DOUGHNUT	
PAWN				
PEACH				
PEANUT				
PEN	PENCIL			
PENALTY	FINE			
PERCUSSION	DRUMS			
PERIMETER	BORDER	CIRCUMFERENCE		
PERIOD				
PERIODICAL	DIGEST	JOURNAL	MAGAZINE	NEWSPAPER
PERMISSION	CONSENT			
PESTICIDE				

PHONOGRAPH	RECORD PLAYER		
PHOTOGRAPHER	CAMERAMAN		
PIANO	ORGAN		
PICTURE	IMAGE	PHOTOGRAPH	PORTRAIT
PIER	DOCK	WHARF	
PILE	HEAP	STACK	
PILLAR	COLUMN		
PILLOW			
PILOT	AIRMAN	AVIATOR	
PLAN	DESIGN	STRATEGY	
PLANTS	FLOWERS	SHRUBS	TREES
PLATE	BOWL	CHINA	
PLATEAU	MESA		
POEM	POETRY	RHYME	VERSE
POISON	VENOM		
POLICE	COP	OFFICER	POLICEMAN
POLLUTION	SMOG		
POOL	PUDDLE		
POPCORN			
POSTER	AD	BILLBOARD	
POTATO	FRENCH FRIES		
POULTRY	CHICKEN	TURKEY	
POVERTY			
POWDER	DUST		
POWER	ELECTRICITY	ENERGY	

PRAYER	BLESSING	GRACE	
PRECIPITATION	RAIN	SNOW	
PREDICTION	FORECAST	PROPHECY	
PREGNANCY	GESTATION		
PREJUDICE	BIAS		
PRESERVATION	CONSERVATION		
PRICE	COST	EXPENSE	
PRIZE	REWARD	TROPHY	
PRODUCTION	CREATION	FABRICATION	
PROFESSIONAL	EXPERT		
PROFIT	EARNINGS		
PROGRAM	SCHEDULE		
PROJECT			
PROPOSAL	PROPOSITION		
PROSTITUTE	CALL GIRL	HOOKER	
PUPPET			
PURSE	POCKETBOOK		
PUZZLE			
QUESTION	INQUIRY	QUERY	
RABBIT			
RACE	DISCRIMINATION	RACISM	
RACKET			
RADICAL	REVOLUTIONARY		
RADIO	STEREO		
RAILROAD	SUBWAY	TRAIN	TROLLEY

RAINBOW

RAY BEAM SUNBEAM

RECORD

REFEREE UMPIRE

RELIGION BELIEF FAITH

REPORT BULLETIN

REPORTER PRESS

RESUME

RHYTHM BEAT

RING

RIVER BROOK CANAL CREEK STREAM

ROBOT ANDROID

ROOM APARTMENT DORMITORY

ROPE CORD NOOSE

ROTATION ORBIT REVOLUTION

RULER

RUSSIAN COMMUNIST

SAFE LOCKER VAULT

SAILOR MARINER SEAMAN

SALIVA SPIT

SAND DIRT

SANDWICH HAMBURGER

SATELLITE

SAUCE

SAYING PROVERB STATEMENT

SCALE

SCAR

BLEMISH

SCARECROW

SCHOOL

COLLEGE

UNIVERSITY

SCORE

TALLY

SCOUT

SEAL

SEASON

FALL

SPRING

SUMMER

WINTER

SECRETARY

SEED

GRAIN

SELLER

PEDDLER

VENDOR

SENIOR

ELDER

SHADE

SHADOW

SHELVES

BOOKCASE

SHELF

SHINE

LUSTER

SHOES

BOOTS

SNEAKERS

SHOWER

BATHTUB

TUB

SHRINE

SANCTUARY

SIDEWALK

WALKWAY

SILENCE

QUIET

SINK

DRAIN

SKY

SLAVE

SLAVERY

BONDAGE

SLEEP

SLICE	SEGMENT		
SLIDE			
SLIPPERS	SANDALS		
SLOGAN	MOTTO		
SMOKE	FUMES		
SOAP	DETERGENT	SHAMPOO	
SOCK	PANTYHOSE	STOCKING	
SOCKET	PLUG		
SOLDIER	WARRIOR		
SONG	HYMN	MELODY	TUNE
SOUND	NOISE		
SOUP	STEW		
SPA	HOT TUB	JACUZZI	WHIRLPOOL
SPEAR	JAVELIN		
SPINE	BACKBONE	VERTEBRAE	
STADIUM	ARENA	COLISEUM	
STAIN			
STAIRS	STEPS		
STAMP			
STAR			
STATE			
STATION	DEPOT		
STONE	ROCK		
STOPLIGHT			
STORE	MARKET	SHOP	

STORM	HURRICANE	TORNADO	
STORY	FABLE	TALE	
STOVE	OVEN		
STRANGER	FOREIGNER		
STREET	FREEWAY	HIGHWAY	ROAD
STRIPPER	STRIPPING	STRIPTease	
SUBJECT	THEME	TOPIC	
SUBMARINE	SUB		
SUBSTITUTE	REPLACEMENT	SURROGATE	
SUCCESS	ACCOMPLISHMENT	TRIUMPH	VICTORY
SUGAR			
SUICIDE			
SUIT	TUXEDO		
SUN			
SURFBOARD			
SWAMP	MARSH		
TABLE			
TALENT	EXPERTISE	SKILL	
TARGET			
TASK	ASSIGNMENT	CHORE	DUTY
TASTE	FLAVOR		
TAX	LEVY		
TAXI	CAB	TAXICAB	
TEACHER	INSTRUCTOR	PROFESSOR	TUTOR
TEARS	TEARDROPS		

TEETH	TOOTH		
TELEPHONE	PHONE		
TELEVISION	TV		
TERRAIN	TOPOGRAPHY		
TEST	EXAMINATION		
THANKSGIVING			
THEATER	CINEMA	PLAYHOUSE	
THEORY	HYPOTHESIS		
THIEF	STEALER		
TICKET	LABEL	TAG	
TIE			
TIRE	WHEEL		
TOASTER			
TOILET	LAVATORY	OUTHOUSE	
TOKEN	SYMBOL		
TONGUE			
TOOL			
TOP	APEX	PEAK	SUMMIT
TOURIST	TRAVELER		
TOY			
TRACK			
TRADITION	CULTURE	CUSTOM	
TRAFFIC			
TRAIL	PATH	PATHWAY	
TRANSPORTATION	TRANSIT		

TREASURE

TRICK	GAG	JOKE	PRANK	STUNT
TRIP	TOUR	VACATION	VOYAGE	
TRIVIA				
TRUCE	ARMISTICE			
TRUCK	PICKUP	SEMI		
UMBRELLA	PARASOL			
UNIVERSE	SPACE			
UTENSIL	FORK	SPOON		
VANDAL	VANDALISM			
VEGETABLES	PRODUCE			
VEIN	ARTERY			
VETERAN				
VETERINARIAN				
VICTIM				
VIRUS	COLD	WART		
VISOR				
WALK	HIKE	MARCH	SAUNTER	STROLL
WANDERER	ROVER	VAGABOND		
WATER	LIQUID			
WATERFALL	CASCADE	FALLS		
WAVE	SURF			
WAX				
WEALTH	FORTUNE	RICHES	WORTH	
WIND	AIR			

WINDOW

WOOD

BOARD

LUMBER

WORKBENCH

SAWHORSE

WORKSHOP

STUDIO

WREATH

References

- [1] C. M. Brown.
Padl-2 : A Technical Summary.
IEEE Computer Graphics & Applications :69-97, 1982.
- [2] W. A. Carter.
ROMULUS : The Design of a Geometric Modeller.
Pub P-80 CM 01 by Cam-I Inc. 1979.
- [3] B. Chandrasekaran and Sanjay Mittal.
Deep Versus Compiled Knowledge Approaches to Diagnostic Problem Solving.
Int. J. of Man-Machine Studies 19(5):425-436, 1983.
- [4] S. K. Chang (editor).
Special Section on Pictorial Information Systems.
Policy Analysis and Information Systems 1(2), January, 1978.
- [5] W. S. Cleveland, C.S. Horris, and R. McGill.
Experiments on Quantitative Judgement of Graphs and Maps.
The Bell System Technical Journal :1659-1674, July-August, 1983.
- [6] Randall Davis.
Expert Systems: Where are we? and Where do you go from here?
AI Magazine 3(2):3-22, 1982.
- [7] Randall Davis.
Reasoning from First Principles in Electronic Troubleshooting.
Int. J. of Man-Machine Studies 19(5):403-423, 1983.
- [8] Randall Davis.
Diagnostic Reasoning Based on Structure and Function: Paths of Interaction and the Locality Principle.
Artificial Intelligence 24:347-410, 1984.
- [9] Richard O. Duda and John G. Gaschnig.
Knowledge Based Expert Systems Come of Age.
Byte :238-281, September, 1981.
- [10] Richard O. Duda and Edward H. Shortliffe.
Expert Systems Research.
Science 220(4594):261-268, April. 1983.
- [11] A. G. Flutter and P. N. Rolph.
Polysurf : An Interactive System for the Computer-Aided Design and Manufacturing of Components.
CAD 76 Conference London :150-158, 1976.

- [12] J. D. Foley and A. Van Dam.
Fundamentals of Interactive Computer Graphics.
Addison-Wesley, Reading, MA, 1982.
- [13] Amar Gupta and B. E. Prasad (editors).
Expert Systems Volume I: Principles.
IEEE Press, New York, 1987.
- [14] Amar Gupta and B. E. Prasad (editors).
Expert Systems Volume II: Microcomputer Based Systems.
IEEE Press, New York, 1987.
- [15] W. J. Hawkins.
Bits and Bytes.
Popular Science, January, 1984.
- [16] G. N. Hounsfield.
Computerized Transverse Axial Scanning, Part I.
Brit. J. Radiology 46:1016-1022, .
- [17] Gary Kahn, Steve Nowlan, and John McDermott.
A Foundation for Knowledge Acquisition.
IEEE Workshop on Principles of Knowledge-Based Systems:89-96, December,
1984.
- [18] Y. E. Kalay.
Interactive Shape Generation and Spatial Conflict Testing.
18 Design Automation Conference Nashville Tennessee:75-89, 1981.
- [19] R. S. Ledley, et. al.
Computerized Transaxial X-ray Tomography of the Human Body.
Science 186:207-212, 1974.
- [20] Jock D. Mackinlay.
Automatic Design of Graphical Presentations.
PhD thesis, Stanford University, December, 1986.
Report No. STAN-CS-86-1138.
- [21] Marvin Minsky.
A Framework for Representing Knowledge.
MIT AI Memo #306, Massachusetts Institute of Technology, June, 1974.
Cambridge, MA.
- [22] W. M. Newman and R. F. Sproull.
Principles of Interactive Computer Graphics.
McGraw-Hill, New York, 1979.
- [23] B. E. Prasad, Amar Gupta, Hoo-min D. Toong, and S. E. Madnick.
A Microcomputer-Based Image Database Management System.
IEEE Transactions on Industrial Electronics IE-34(1):83-88, February, 1987.

- [24] B. E. Prasad.
Design Specifications for EV-Image Database System.
Technical Report, Massachusetts Institute of Technology, 1987.
- [25] J. A. Reggia, Dana S. Nau, and Pearl Y. Wang.
A New Inference Method for Frame-Based Expert Systems.
Proc. of AAAI-79 :333-337, Washington, August, 1983.
- [26] Mark Stefik.
An Examination of a Frame-Structured Representation System.
Proc. of IJCAI '79 :845-852, August, 1979.
- [27] Glen James Stump.
Computer Aided Image Selection from an Image Database.
Bachelor's thesis, Massachusetts Institute of Technology, January, 1986.
- [28] The Merriam-Webster Thesaurus.
Pocket Books, New York, 1978.
- [29] Hoo-min D. Toong, and Amar Gupta.
A New Direction in Personal Computer Software.
Proceedings of the IEEE 72(3):377-387, March, 1984.
- [30] Frederick A. Wang, Ahmed H. M. El-Sherbini, Stan Fry, Mike Smutek, and Nancy Webb.
The Wang Professional Image Computer: A New Dimension to Personal and Office Computing.
Proceedings of the IEEE 72(3):300-311, March. 1984.
- [31] J. Wilhelms and B. Barsky.
Using Dynamic Analysis to Animate Articulated Bodies such as Humans and Robots.
Proc. Graphics Interface '85 :97-104, Montreal, 1985.
- [32] D. Zeltzer.
Motor Control Techniques for Figure Animation.
IEEE Computer Graphics and Applications 2(9):53, 1982.





STORAGE AND RETRIEVAL OF PICTORIAL INFORMATION IN HETEROGENEOUS COMPUTING SYSTEMS

LOWELL W. KIM

Many large databases contain three different types of information - numerical, textual, and pictorial. While there are widely accepted standards such as ASCII for the storing of numerical and textual data, the area of pictorial information lacks such a widely accepted representation format. In order to share pictorial information in a heterogeneous environment, it is necessary to come up with techniques that work in spite of the lack of a common standard.

There are several different techniques in use for storing pictorial images. Bit mapped image storage mechanisms store the image in terms of individual pixels; vector-based techniques economize on the storage requirements by taking advantage of geometric properties; quadtree approaches split the image into four quadrants, test each quadrant for homogeneity, and subdivide further until homogeneity is achieved; and pyramid-oriented approach involves continued subdivision until the level of individual pixels is reached.

Since pictorial data occupies large amounts of storage space, it is usual practice to use some compression technique. Common techniques are: Statistical Image Data Compression, Transform Image Coding, and Hybrid Image Coding. Many of these techniques use algorithms and codes that disfavor retrieval of pictorial information by unauthorized persons.

Based on the diversity of the methods used for storing information, and the additional complexity caused by the use of compression algorithms, it is very difficult to come up with a methodology that will allow access to pictorial information across heterogeneous computing systems. This difficulty may be overcome by allowing the pictorial information to be decompressed, and even the whole image recreated from its vectors if necessary, in the native environment, and then transmitting the regenerated picture to the computer that requested it.



Chapter 1

Heterogeneous Computing Environments

Large organizations are becoming increasingly dependent upon computerized data. This dependence as well as the sheer size of the organizations have made it necessary for most large organizations, along with many small ones to rely on multiple computer systems to support their operations.

1.1 Digital Computers

The advent of digital computers paved the way for technology to take a giant step forward in almost every discipline, including electrical engineering, the forefather or groundworks of the digital computer itself.

1.1.1 Digital Communication

The development of information provision services, including the transmission of pictorial material, has been nurtured by the close relationship between the digital computer and digital communication systems.

Arguably one of the most radical changes [that developed as a direct result of the development of the digital computer] has taken place in communication, in which the accent is now heavily

placed upon representation of the input information in discrete (sampled and quantised) form, leading to a much greater flexibility in the scope and nature of the operations which can¹ subsequently be carried out on the data.

With the increased processing power of the digital computer, coding and transmission methods can take advantage of the flexible structure of the data.

1.1.2 An Inherent Deficiency

There has been, however, one major obstacle for digital computer image transmitters to overcome - that of limited bandwidth. Although digital communication links offer several advantages over their analogue counterparts, they also require a much higher equivalent bandwidth.

There has been a perennial desire for an increase in bandwidth which would enable more users to benefit from a communication system. Coupled with the taxing demands on the system of transmitting images at a reasonable rate, this desire has become an even more critical and important one. Thus, there has been much interest in picture transmission at as low a bandwidth or bit rate as is conveniently possible. In fact, virtually every attempt

¹
R.J. Clarke, Transform Coding of Images (New York: Academic Press, Inc.), p. 1.

to come up with a successful image-processing procedure has involved some sort of bandwidth requirement reduction scheme.

1.2 Problem Development

Generally, the numerous diverse computer systems an organization employs were chosen and/or designed by the organizations with the desire to fulfil a specific need. These needs have different computing requirements depending on the nature of the task. Some tasks may require fast processing, but minimal memory; whereas, other tasks may require archives and archives of memory, but can be as slow as the first computer ever built.

There are many other issues involved in choosing the appropriate system besides processor speed and memory size including: cost, desired level of reliability and fault-tolerance, availability of relevant hardware and software, type of information being processed, type of processing being performed, etc.

Thus, many large organizations have found themselves in environments in which they have a number of dissimilar and incompatible hardware and software systems in operation. Often, data processing jobs need to process information stored on a number of these dissimilar systems in a similar fashion. This frequent occurrence has not been accounted for in the design of heterogeneous systems.

While each of these systems may still meet the objectives of the original task for which it was chosen, the heterogeneity of the systems presents a major obstacle in situations requiring access and assimilation of information resident on the dissimilar computing systems.

New techniques need to be developed to allow easy, efficient, and intelligent access to information hosted on multiple heterogeneous systems.

1.3 Existing Deficiencies

The problem of inefficient, incomplete, and time-consuming access to information in heterogeneous system environments can be traced, from a technical viewpoint, to functional deficiencies at several different levels as follows:

1. Structured and unstructured applications;
2. Information versus knowledge;
3. Diverse types of information;
4. Semantics;
5. Communications;
6. Granularity; and
7. Security.

2

Amar Gupta and Stuart E. Madnick, An Overview of Knowledge-based Integrated Information Systems Engineering -- Objectives and Directions, p. 2.

Each of these deficiencies complicates the issue of information access a little further.

1.4 Diverse Types of Information

One important issue to pursue further is the existence of diverse types of information in nearly every computing environment. Information can be represented by numbers, text, graphics, pictures, speech, or video. Since each different type of information has its own characteristics, attributes, and applications, each type must be stored and retrieved in a somewhat different manner. The different types of information are stored and referenced in their own individual way.

After considerable research and debate, standards have been established regarding the efficient and effective storage of numerical and textual information. However, there is a noted deficiency in the standardization of procedures for pictorial information storage and retrieval. Thus the topic of my research: the storage and retrieval of pictorial information.

Chapter 2

Heterogeneous Display Environments

The emergence of heterogeneous computing environments in virtually every data-dependent organization has given birth to a renaissance of research into handling information sharing among diverse systems. Among the major obstacles facing the sharing of pictorial information is the uncomformity of the resolution of the different video screens.

Not only do different vendors market displays of different resolution than other vendors, but also each vendor often produces different resolution displays for its different models or hardware configurations. This complication produces a plethora of different display dimensions. Further complicating the issue, more often than not the dimensions of a given display are not proportionate to any other displays.

2.1 Whole Number Proportionate Conversions

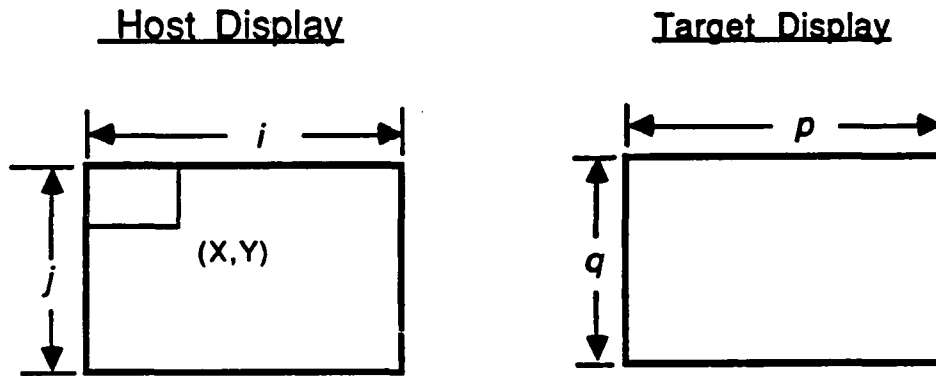
When porting an image from one display environment to another multiply proportionate environment (i.e. having a whole number proportionality constant), pixel representation conversions are simple. Simply "blow-up"

the image to fit the new resolution display. For example, if the host display has resolution exactly half that of the target display in both the horizontal and vertical axes, then simply expand each image pixel to cover a block of two-by-two pixels on the target display. The new image will not be taking advantage of the higher resolution display or the target system, but the image will be truly represented. Smoothing techniques can then be applied to the new image on the target system to fully exploit the dimensions and resolution of the new display.

2.2 Non-exact Conversions

In cases where the two transferring displays do not have matching or multiply proportionate resolution, formulas and approximations must be used. The theory behind these conversions is to proportionately expand or contract the image to cover the entire dimensions of the target display.

Consider Figure 3-1 on the next page. This image conversion device is a bit-by-bit, or pixel-by-pixel mapping of the original image to the target system. It involves approximations (actually just rounding off errors), but still preserves high fidelity quality from the original image. The device was designed to be used by the host system to convert an image into a form usable by a specified target system.



(X,Y) coordinates of current pixel being encoded.

$(X * p/i, Y * q/j)$ encoding formula to locate corresponding coordinates on target display.

All coordinates are rounded off to the nearest integer.

Figure 2-1: Host-To-Target Image Conversion

2.3 An Incomplete Solution

Although these conversion mechanisms perform the calculations and transformation necessary to port images between two different display environments, it is not practical to expect every computing system to contain conversion formulas from its display dimensions to every

other possible dimension. Furthermore, image-borrowing systems would need to report their display dimensions to the transmitting image coding mechanism in order to ensure proper conversion. In short, there would be extensive overhead, both in memory usage and in processing time, involved in transmitting images among systems with heterogeneous display environments using this shortsighted strategy.

Chapter 3

Image Storage Theories

After a somewhat superficial review of available literature concerning image storage theories, I have chosen four major theories to discuss: bit-mapped, vector-based, quadtrees, and pyramid storage methods. Although this certainly is not an exhaustive list of image storage theories, these four theories represent the bulk of the methodologies in use today.

3.1 Bit-Mapped

Bit-mapped image storage mechanisms take each pixel from the display and store it as part of the image representation. This process continues bit by bit for the entire range of the display.

Due to the incredibly large quantities of information needed to store an image by bit-mapping every pixel, image data compression mechanisms have been developed. These techniques, their purpose, their mechanisms, and their usefulness, will be discussed in the next chapter. For now, bit-mapped storage mechanisms take advantage of these data image compression techniques to reduce the amount of information needed to be stored to preserve an image with adequate fidelity.

3.2 Vector-Based

Vector-based image storage mechanisms take advantage of properties of simple geometric figures. For example, a circle is not stored as a number of pixel settings. Instead it is stored as a circle with a given center and a given radius.

This storage method greatly reduces the amount of information needed to be stored in order to adequately represent an image. It is however somewhat limited in the range of images which it can accommodate. Also, coding of the images into geometric representations is a major task in and of itself.

3.3 Quadtrees

Quadrees are trees of degree four (4) whose leaves represent homogeneous blocks of an image. They are formed by recursively subdividing an image into quadrants and analyzing the homogeneity of its subparts.

Given a criterion for deciding that a digital image is uniform or homogeneous (for example, the standard deviation of its gray levels falls below a given threshold), it is possible to recursively subdivide a given image into homogeneous pieces. First check the image for homogeneity. If the image is homogeneous to start with, we are done. If it is not, split the image

into quadrants and test each of them for homogeneity. If a given quadrant is found to be homogeneous, that block of the image is done; if it is found to be not homogeneous, subdivide it into quadrants again, and so on, until all its parts are found to be homogeneous.

The results of the subdivision process can be represented by a tree of degree 4 (a "quadtree"). The root node of the tree represents the entire image, and the children of a node represent its quadrants. Thus the leaf nodes represent blocks (sub...subquadrants) of the image that are homogeneous.³

3.4 Pyramids

The basic image pyramid construction scheme is based on recursive subdivision into quadrants, just like a quadtree construction. The difference is that the subdivisions of the pyramid structure always keep subdividing until they reach the individual pixel level. Thus, the leaves on the bottom layer of the pyramid, the base of the pyramid, represent single pixels. This strategy could be considered a primitive version of the

3

A. Rosenfeld, *Quadtrees and Pyramids: Hierarchical Representation of Images*, p. 2.

⁴
quadtree storage mechanism.

3.5 Analysis

Vector-based procedures are somewhat limited. They are constrained by the fact that not all objects or images are easily represented by simple geometric figures. Even for those images that do consist of well defined geometric shapes, an image creator must clearly distinguish and define these shapes in order for a vector-based image storage mechanism to function correctly. Vector-based storage devices are far superior to bit-mapped devices for certain very simple images. This specializing does not interest us in the formulation of an image storage standard since it is to be used with all types of images.

Further, quadtree storage is a more efficient storage mechanism than pyramid storage simply because redundant subdivisions are eliminated. A quadrant that is found to be "homogeneous" (as defined by each particular application) need not be subdivided again. By avoiding this superfluous work, both processing time and required

⁴
Rosenfeld, op. cit., p. 4

storage space is reduced.⁵

Quadtree storage however falls into a pitfall in dealing with heterogeneous display environments. The number of subdivisions each quadrant of the display contains becomes tedious and complicated to maintain for different possible display environments.

Thus, let us look further into a bit-mapped approach. We need to refine its mechanism in order to develop an image storing standard.

⁵
Dyer, Space Efficiency of Region Representation By
Quadtrees, p. 2.

Chapter 4

Image Data Compression

One critical issue involved with image processing is the compression of the image data in order to reduce bandwidth requirements. In particular, we are interested in the compression of data relative to bit-mapped data storage.

4.1 Statistical Image Data Compression

There are a number of different approaches to image data compression. The vast majority of these methods reduce to three basic types:

1. Predictive or Spatial coding - carried out in the spatial domain,
2. Transform coding - a frequency domain process, and
3. Hybrid coding - a combination of the more attractive features of the first two methods.

All three of these approaches are examples of statistical image data compression mechanisms; i.e. they are concerned with the statistical features of the image (the changes in frequencies, etc.), not the appearance of the image in terms of human perception.

They make no attempt to code image information in what might be described as a meaningful way, i.e., [there is no attempt at] abstracting those features which the human observer might consider important. They simply operate by virtue of the statistical properties of the image or class of
6
images in question.

Each of these image coding mechanisms has its advantages and its disadvantages. One method is more appropriate than the others in different situations. It mainly depends on the error tolerance and speed requirements that are important to the system.

4.2 Evaluation Criteria

In terms of measuring the success of a data compression technique, bandwidth requirements can be stated in terms of bits/element -- in other words, the number of bits required to appropriately and sufficiently represent an element. Obviously, the lower bit/element ratio corresponds to the smaller bandwidth requirement and accordingly to the more successful compression technique.

In addition to bandwidth reduction, processing time, error tolerance, and of course reliability, or accuracy, are also important factors in the evaluation of a data compression technique.

⁶ Clarke, op. cit., p. 2.

4.3 Predictive or Spatial Image Coding

The objective of spatial image coding is to effectively reduce the number of bits necessary to represent a picture and still maintain some fidelity quality relating the pre and post coded images.

4.3.1 Procedure

This procedure is done by correlating adjacent picture elements with each other. This coding method relies on the assumption that the image data source is highly correlated, which implies that picture elements lying in the same neighbourhood will tend to have similar amplitudes.

Thus, we can exploit the value(s) of one or more previously determined elements from the same line, or from a previous line or lines, or frame(s) to form a prediction of the present element. We then subtract our prediction, which we expect to be quite good on the average given the nature of the image in a statistical context, from the actual value of the present element and expect, again on the average, to obtain a quite small value.

The degree of accuracy of the predictions naturally depends on the uniformity or smoothness of the image. The calculated difference between the prediction and the actual value is then coded and transmitted or stored. The

greatly decreased magnitude of the argument being stored is the method through which the predictive image coding scheme reduces its bandwidth requirement.

4.3.2 Evaluation

This method is simple, easy to implement, and with an adaptive system gives good quality images in the 1-2 bit/element range. It is, however, quite sensitive to variations in input data statistics and also to channel errors.

4.4 Transform Image Coding

The major objective in transform image coding is to manipulate the image into an invertible form which can be more easily coded. Usually this entails providing data in a form which is more uncorrelated than the original picture elements. This abstraction enables the bandwidth reduction process to be implemented in the image transform domain with coders of minimal memory capacity.

4.4.1 Procedure

More specifically, mathematical transforms are used to effect a spectral decomposition of the spatial domain input signal. The well known Fourier transform had been used frequently for image coding in the past, however, it has, for a large part, been set aside by other transforms

which produce more efficient code and have the requirement of performing real number manipulations only.

4.4.2 Evaluation

Transform coding is a much more complex image data compression technique than predictive coding. Any advance in high-speed digital hardware greatly affects a transform coder's efficiency. Given an average image (i.e. one that does not have large amounts of intricate spatial detail), an adaptive system will produce good image quality at rates between 0.5 and 1.0 bit/element. Transform coding is less sensitive to errors than predictive coding; however, when an error is encountered which affects system performance, it tends to be somewhat more damaging for transform coders than for their counterparts due to uncertain propagation of the original error.

4.5 Hybrid Image Coding

Hybrid image coding, as one might expect, takes some of the advantages of each of the two previous methods. It utilizes the quickness and ease of predictive coding as well as the effectiveness of transform coding.

As mentioned earlier, transform coding is relatively costly to implement. There is a great deal to gain by performing two-dimensional sampling and coding. However,

AD-A133 854

INTEGRATING IMAGES APPLICATIONS AND COMMUNICATIONS
NETWORKS VOLUME 5(U) MASSACHUSETTS INST OF TECH
CAMBRIDGE A GUPTA ET AL. DEC 87 MIT-KBIISE-5

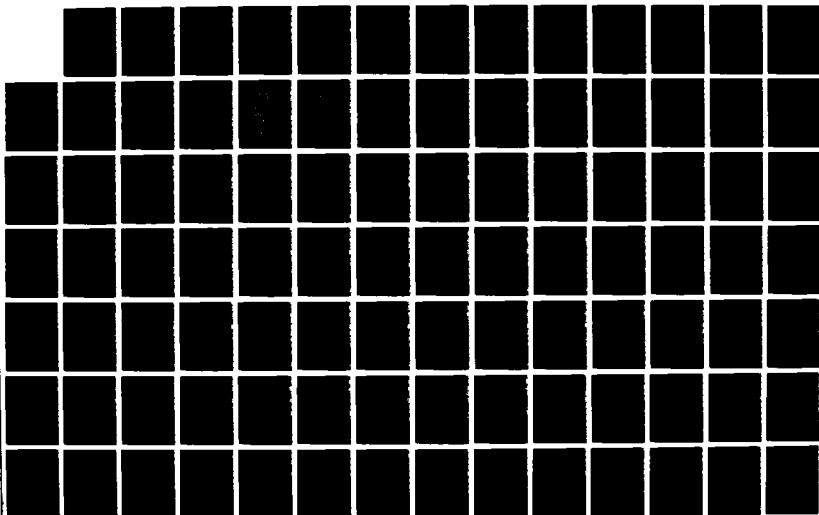
2/4

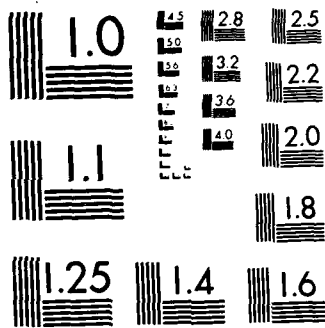
UNCLASSIFIED

DTR537-85-C-00003

F/G 5/2

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

the requirements of implementation of two-dimensional transform coding is quite severe. This drawback led to a predominance of hybrid coding techniques, at least up to present history. As new technological advances occur, implementation costs of transform coding are becoming less and less important.

4.5.1 Procedure

Using one-dimensional transform coding in one axial direction and augmenting that with a predictive coding operation in the other axial direction, hybrid coding schemes are able to take advantage of the correlation existing in both horizontal and vertical directions in the image. Alternatively, hybrid schemes may also perform the predictive step first and then transform the results, but this variation requires a more complex system.⁷

4.5.2 Evaluation

As may be expected, hybrid image data coding approaches have some advantages and some disadvantages of both of its constituent methods: minimum coding rates are not as low as those of pure transform coding but implementation is easier. With adaptive techniques,

⁷ Clarke, op. cit., p. 5.

hybrid approaches achieve rates around 1 bit/element with adequate reconstructed image quality. Hybrid methods yield a more accurate image than do its predecessors due to the two dimensional sampling of the image.⁸

4.6 Anomalies

There are a number of peripheral image data compression schemes. They include an hybrid optical/digital interframe scheme and adaptive transform coders.

4.6.1 Hybrid Optical/Digital Scheme

Optical and digital processes are becoming increasingly directly competitive. Within the hybrid optical/digital data compression technique, there is an example in which optical computations can replace digital computations.

The general structure of the hybrid/optical interframe compression system is broken into two parts as shown in Figure 1. The portions of the system using digital and optical componentry are separated and distinct, as are their functionalities.

The first part, the optical spatial compression

⁸

R.A. Jones, Adaptive Hybrid Picture Coding, p.1.

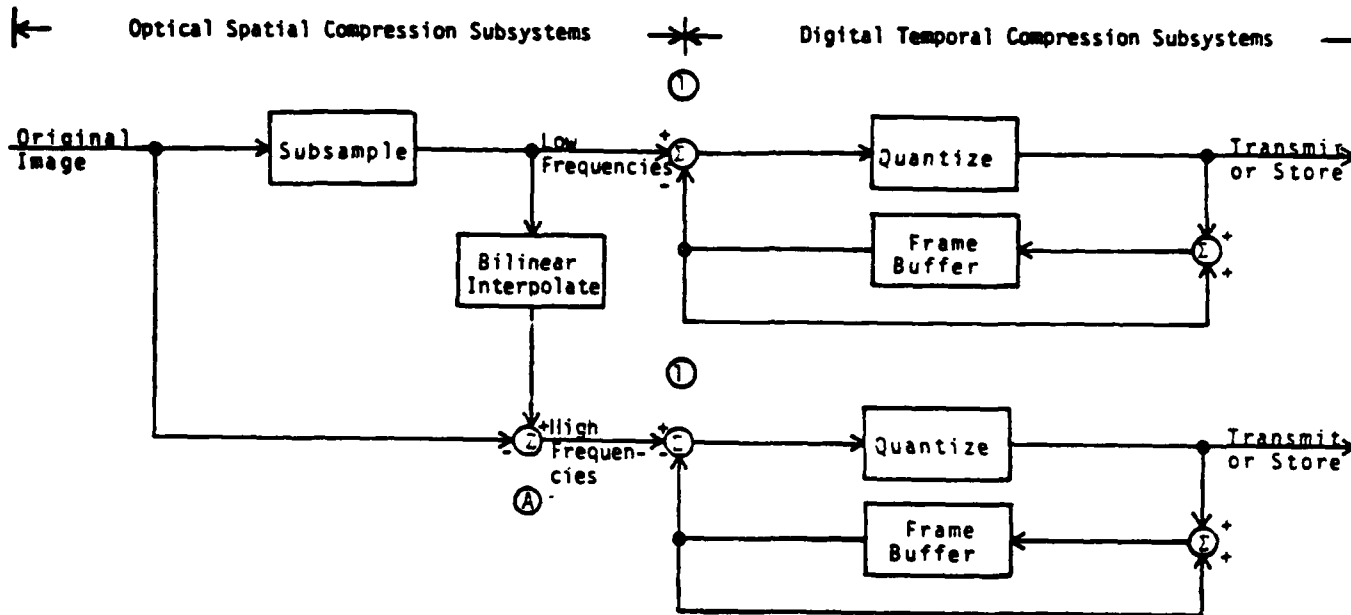


Figure 4-1: Hybrid Optical/Digital Compression Schematic

subsystem, is where the spatial data redundancy is eliminated. The image is subsampled, and the subsamples are used to reconstruct a low-frequency version of the original image using bilinear interpolation of the subsamples. The low-frequency version of the image is then subtracted from the original image. In other words, the original image passes through a high-pass filter.

Next, the image passes through the digital temporal compression subsystem where temporal data redundancy is

eliminated. The quantizer and feedback structure are similar to a conventional DPCM image data compression system only in this method there is parallel instead of serial data flow around the quantization/feedback loop.⁹

This type of image coding is comparable to earlier mentioned ones in performance. However, the optical subsystem and optical computations creates the need for additional devices, algorithms, and interfaces for analysis of the image data. The added subsystem adds to the complexity and cost of the overall system.

4.6.2 Adaptive Transform Image Coders

The motivation behind adaptive image coding procedures is the necessity to model imagery as a nonstationary source. The bandwidth compression algorithm performs a "learning" procedure by which a localized model is developed, which is, however, only applicable to smaller limited regions. In adaptive coding schemes, important parameters of the bandwidth compression algorithm are image dependent.¹⁰

9

B.R. Hunt, Hybrid Optical/Digital Interframe Image Data Compression Scheme, pp. 1-3.

10

Richard B. Cox and Andrew G. Tescher, Channel Rate Equalization Techniques for Adaptive Coders, pp. 1-2.

Chapter 5

Pictorial Data Storage

The compressed image data is now in bit form. That is to say the image is stored as a stream of coded bits. The next step is to standardize the method in which the image bits are or can be coded.

Currently, different computing systems have no way of being able to interpret the stream of image data bits without knowing a retrieval or decoding mechanism. It is not the simple task of storing a value for each pixel of the image and transferring that information to another system. Compression mechanisms must be used to reduce memory and bandwidth requirements, and all compression mechanisms generate image data bits in very different ways. Therefore, either a decoder in some standard form must be sent with the bit stream, or else a standard compression mechanism must be established.

5.1 Universal Decoder

It would be virtually impossible to develop a universal decoder interpreter. There are too many variables involved: screen size and resolution, possible image data transformations, byte size, and the list goes

on and on. Therefore it seems that a universal compression mechanism would be the solution to the standardization issue.

5.2 The Importance of Compression

Compression devices are an essential element of image processing. Simulations show that on the average a total of 3,702,720 bits/document are required to represent an image if no compression is attempted; whereas, after one-dimensional run-length coding, an average of only 445,316 bits/document are required. Furthermore, it was found that after use of one of the variations of the ordering technique compression mechanism, only 264,632 bits/document are required. Thus, the ordering technique reduces the number of coded bits by approximately 93 percent compared to uncompressed data.¹¹ That is 3,438,088 bits/document, on the average, that the transmission lines do not have to worry about. This tremendous decrease in transmitted bits reduces the chance of transmission faults; i.e. dropped or tainted packets due to noise or distortion. Since bandwidth is important, with today's technology anyway, the compression of image data is essential.

11

David Ting and Birendra Prasada, "Digital Processing Techniques for Encoding of Graphics," Proceeding of the IEEE, Vol. 68, No. 7, 1980, p. 757.

Table 5-I: Comparison of Average Document Size Compressed Versus Uncompressed

<u>Compression Type</u>	<u>Average bits/document</u>
not	3,702,720
one-dimensional run-length	445,316
ordering technique	264,632

5.3 A Standard Compression Mechanism

An effective and efficient compression technique needs to be chosen which can be easily implemented on as many different systems as possible. The choice of a standard compression mechanism goes beyond the scope of this thesis. There are many different mechanisms already out there including a number of hybrid strategies combining two or more other mechanisms, some mechanisms that require special hardware, etc. A number of important issues have been brought out that must be considered in the decision, but a thorough analysis of all the compression mechanisms in terms of performance, efficiency, adaptability, cost, and many other factors needs to be conducted before choosing an appropriate mechanism.

Chapter 6

Pictorial Data Retrieval

The issue of pictorial data retrieval mainly concerns the representation and cataloging of images in a pictorial database. The critical questions are:

1. How to index images for a database, and
2. How to locate an image in a database.

6.1 Introduction

While alphanumeric databases are quite common today, pictorial databases are quite rare in comparison. This shortcoming is due to a number of factors including:

1. The large amounts of storage space needed to store even small pictures,
2. The dearth of simple, naive-user-oriented languages and techniques for extracting and manipulating pictorial data, and¹²
3. The relative newness of pictorial information storage technology.

12

R.B. Abhyankar and R.L. Kashyap, "Pictorial Data Description and Retrieval with Relational Languages", p. 1

Clearly, it would be very beneficial for a number of different types of professionals for the field of pictorial databases to be developed further. It would very convenient to provide a high level query language that would enable non-professional programmers to extract information from and manipulate image data without knowing the details of how it is stored. Standardization of picture formats is an essential step towards this goal.

However, constructing an integrated system comprising a conventional database system and an image database system would best fulfil the need.

13

6.2 Picture Indexing

An index to a picture is an auxiliary structure which aids in accessing the information in the picture. The construction and use of indices for pictures is an important problem in picture data management.

A picture index can aid access in either or both of two ways:

1. The index allows the localities of a picture to be ordered for search so that those highly likely to satisfy the query are checked first.

13

R.B. Abhyankar and R.L. Kashyap, "Pictorial Data Description and Retrieval with Relational Languages," p. 1.

2. The index provides the answer to the query directly, so that the original scene is¹⁴ bypassed entirely.

Picture indexing can play a helpful role in image transmission, particularly if the receiver wants specific portions of an image rather than a complete image. Furthermore, by sending indices and/or progressive refinements in lieu of original data or in front of it, receiving processes have a chance to analyze the information being received through the non-costly-to-transmit indices and, if appropriate, stop the transmission to avoid costly mistakes and/or superfluous¹⁵ transmission.

6.3 Name-Value Slots

Name-Value slots are a mechanism for storing image information in image headers. They are self-describing units of information with a name, used for accessing the information, and a value, the actual value of the slot. The underlying meaning of a name-value slot is that it

¹⁴ Steven L. Tanimoto, "Hierarchical Picture Indexing And Description", p. 1.

¹⁵ Steven L. Tanimoto, Hierarchical Picture Indexing and Description, pp. 1-3.

describes a relation between a given name and value. Name-value slots provide a generality and flexibility that is not available with static information storage schemes.

16

16 Peter G. Selfridge, Name-Value slots and the Storage of Image Information, pp. 2-4.

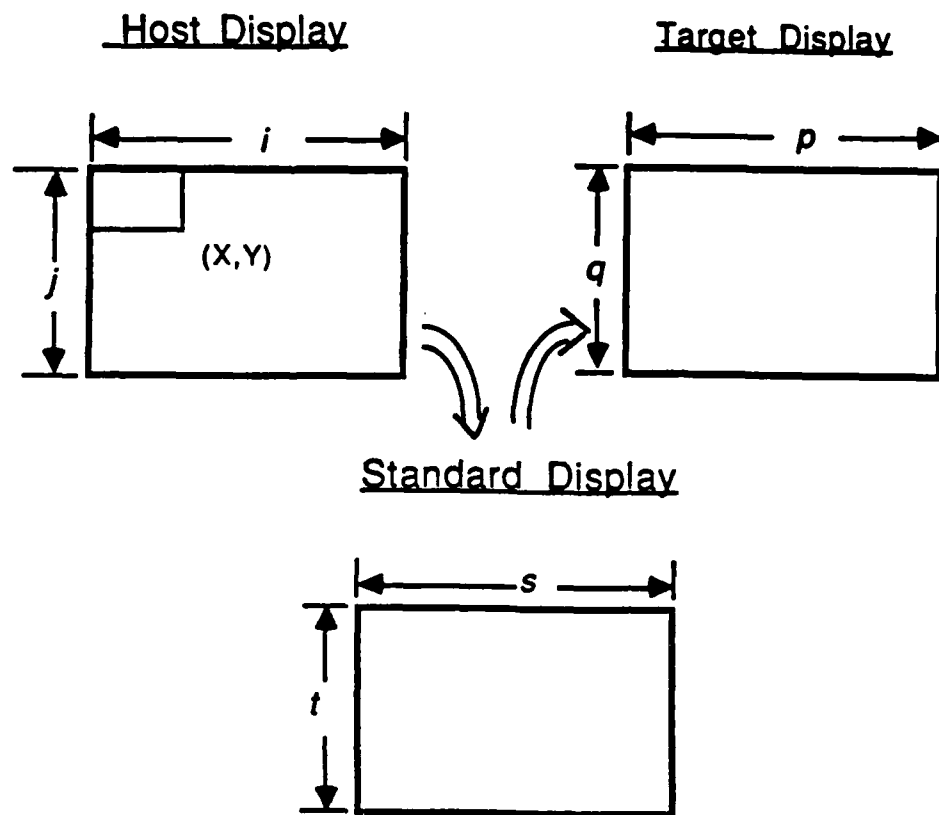
Chapter 7

An Alternative

From the above discussion, it has become apparent that there are a number of important issues that need to be considered when developing a pictorial information storage model. The model must be adaptable to any system, be able to communicate among dissimilar and otherwise incompatible systems, have good post image fidelity, be somewhat efficient, and require minimum bandwidth.

7.1 Basic Architecture

I believe I have come up with a viable format for a solution to this problem. The basic architecture of the suggested mechanism is shown in Figure 7-1. It is similar to the host-to-target conversion mechanism discussed earlier, however, it has an additional step which makes each step modular. This modularity facilitates clean and easy switches among different computing systems.



(X, Y) coordinates of current pixel being encoded.

A $(X * s/i, Y * t/j)$ converted to standard coordinates.

B $(X * s/i * p/s, Y * t/j * q/t)$ target coordinates found using receiver's decoder.

All coordinates rounded to nearest integer.

Figure 7-1: Suggested Image Storing Mechanism

7.2 Still Undetermined Absolutes

The strategy described is a "big-picture" view of the mechanism. A number of the exact specifications have been left to be determined after further, intensive study.

7.2.1 Dimensions of the "Standard" Display

The "standard" display dimensions labelled 's' and 't' are left to be assigned after determining the most opportune display size. That is, it must be ascertained what display size is most common or easiest for a majority of systems to convert to. Also, loss or gain of resolution should be considered in deciding the standard resolution, since every transmitted image will have at best resolution $s \times t$.

7.2.2 Standard Compression Mechanism

The standard compression mechanism still needs to be determined. Issues involved with this decision have been discussed in an earlier chapter. Perhaps it would be best to determine a couple of standards for different categories of image types.

7.3 Procedure

Let us step through the processing of an image following the suggested mechanism:

1. An image starts on the host system with display dimensions $i \times j$.
2. The image is bit-mapped using the given conversion formulas 'A' on each pixel to obtain an image in standard resolution and of dimension $s \times t$.
3. The data is compressed using the standard compression mechanism.
4. The data is transmitted to the target system.
5. The data is decompressed, and the standardized image is bit-mapped using the given formulas 'B' to convert the image to the receiving system's resolution and dimension.
6. (Optional) Smoothing and refining can be performed by the target system to take advantage of higher resolution displays.

7.4 Modularity

The standard-dimension-display concept gives this mechanism its modularity and universality. All systems can convert the pictorial information stored on it to fit the standard form. The coding and transmission of the image data is entirely independent of the target system.

Step 4 is the interface between the host and target systems. After step 4, the target system takes over the processing of the image. Thus, the receiving system simply converts the standard form image into one that fits the resident display dimensions.

7.5 Conclusion

This mechanism is clean, simple, and universal. I believe this mechanism could be used as a standard for storage of any pictorial information in heterogeneous computing environments.

It is limited in its speed. Real-time application can be all but ignored as impossible. The intermediary step is hindered by its universality. In order to be compatible with all systems, the standard form necessarily must be of a very basic level.

REFERENCES

- [Abhyankar 80] Abhyankar, R.B. and R.L. Kashyap.
Pictorial Data Description and Retrieval
with Relational Languages.
Technical Report, Purdue University, 1980.
- [Clarke 85] Clarke, R.J.
Transform Coding of Images.
Academic Press, Inc., 1985.
- [Cox 76] Cox, Richard V. and Andrew G. Tescher.
Channel Rate Equalization Techniques For
Adaptive Transform Coders.
Technical Report, The Aerospace
Corporation, 1976.
- [Dyer 80] Dyer.
Space Efficiency of Region Representation
By Quadrees.
Technical Report, University of Illinois at
Chicago Circle, 1980.
- [Hunt 80] Hunt, B.R. and H. Ito.
Hybrid Optical/Digital Interframe Image
Data Compression Scheme.
Technical Report, University of Arizona,
1980.
- [Ichikawa 80] Ichikawa, T. et al.
Retrieval of Image Features in Terms of
Content-Addressing of Hierarchically
Structured Image Data.
Technical Report, Hiroshima University,
1980.
- [Jones 76] Jones, R.A.
Adaptive Hybrid Picture Coding.
Technical Report, Boeing Aerospace Company,
1976.
- [Lawrence 83] Lawrence, Victor B. et. al.
IEEE Communications Society's Tutorials in
Modern Communications.
Computer Science Press, Inc., 1983.

- [Rosenfeld 83] Rosenfeld, A.
Quadrees and Pyramids: Hierarchical Representation of Images.
Technical Report, University of Maryland,
1983.
- [Selfridge 80] Selfridge, Peter G.
Name-Value Slots and the Storage of Image Information.
Technical Report, University of Rochester,
1980.
- [Stoffel 81] Stoffel, J.C. and J.F. Moreland.
A Survey of Electronic Techniques for Pictorial Reproduction.
IEEE Transactions on Communications , Dec.,
1981.
- [Tanimoto 80] Tanimoto, Steven L.
Hierarchical Picture Indexing and Description.
Technical Report, University of Washington,
1980.
- [Ting 80] Ting, David and Birendra Prasada.
Digital Processing Techniques for Encoding of Graphics.
Proceedings of the IEEE , Nov., 1980.

AN EXPERT SYSTEM FOR ACCESSING AND INTEGRATING DESIGN ANALYSIS KNOWLEDGE

WILLIAM H. B. HABECK

A number of different design tools are currently in use in large manufacturing organizations. Each of these tools addresses a specific design domain. For example, packages such as CADAM and CATIA are used for mechanical design, while ITAM and PHOENICS are oriented towards thermal design. Unfortunately, each of these design environments requires data to be specified in a particular manner and it is virtually impossible to automatically transfer data from one environment to another.

In the long run, the ideal situation would be to create a new and powerful system which incorporates and integrates the functions of all existing design tools. In the short run, however, it is appropriate to think in terms of designing an intelligent front-end that can assist the designer in making the best use of existing packages. Based on information supplied by IBM, and employing ideas and techniques from expert systems, such an intelligent front-end has been developed at MIT. The details of this work are described in this technical report.

The intelligent front-end has been written in C Language. The knowledge base consists of three commonly used design packages and information about the relevance of each package. The inference mechanism uses the weight-based forward chaining method. The user is asked to respond to a series of questions. For each question, there is a set of possible answers covering all cases. ("Don't know" is a valid answer). Each answer has three weights which are used to update the certainty values for the three programs. Certainty values and weights range between zero and one. Questions are chosen based on their expected ability to make maximum changes in the certainty value. These expectations are based on a response history, within the knowledge base, which is updated after every run.

TECHNICAL REPORT #10

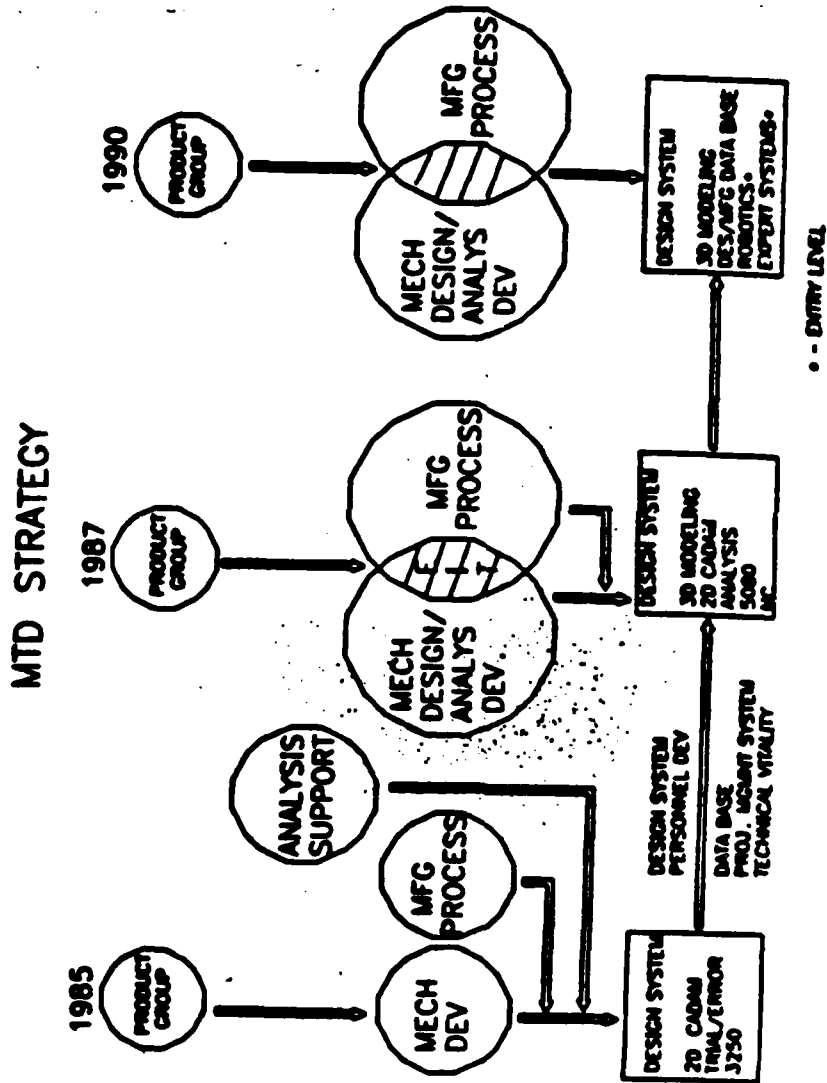
I. Introduction

Background

Although the exciting field of artificial intelligence is decades away from its goal of simulating human thought, it has already spawned some commercial applications in the promising area of expert systems. Expert systems are software packages that make decisions as experts in a narrowly-defined field would. Although they lack common sense, each has knowledge about a certain area that is sufficient to operate on par with experts in that area. As a result, expertise can be widely shared and can survive the expert's retirement. This paper covers a small expert system, for use by IBM engineers in Kingston, New York, that chooses an appropriate design analysis package based on the answers to a set of questions about the design. The order of the questions is based on previous answers to minimize the total number of questions asked.

The designers at IBM in Kingston are mechanical and electrical engineers who determine the designs of many different varieties of electronic and electrical equipment, including processor packages, circuit boards, power supplies, thermocouples, cathode ray tubes, keyboards, disk drives, hard disks, line printers, laser printers, fans, card readers,

Figure 1: IBM Overall Project Description



magnetic tape drives, cartridge drives, enclosures for electronic equipment, and systems made up of several of these types of equipment. Chief goals in design are to minimize the cost of producing the desired component while maintaining high standards of reliability and performance. Items are subject to specifications concerning their ability to withstand certain environmental conditions. Other goals include meeting a weight limit, a durability minimum, an average lifetime, and using standard subcomponents.

The most accurate way to test a component to make sure it meets certain criteria is to build a prototype of it and subject it to tests. This is also the most expensive alternative. On the low cost end of possibilities is making a simple mathematical model of a component and solving certain equations to determine its behavior. Unfortunately, the simplicity needed for a solution to exist might hide some physical reality. Design modeling and analysis computer packages, known as CAD systems for "computer aided design", serve to build a highly accurate model without expensive materials acquisition and assembly. Typically the item is built on the screen using a set of commands roughly analagous to prototype assembly. It can be manipulated to get views of all sides and of inside pieces. Physical parameters and constraints are added for the subcomponents and, for the most

part, the CAD package calculates whatever values the engineer needs. Changes can be analyzed much more easily and tradeoffs can be weighed more accurately. Systems with graphics provide the designer with a sense of realism that used to be found only in prototypes.

The packages used by the IBM designers are named ITAM, CAEDS, and PHOENICS. The three of them are powerful and fairly general-purpose. For most designs encountered, at least one of the programs is adequate for the analysis. ITAM stands for Interactive Thermal Analysis Modeler. It is executed from a graphics preprocessor known as CADAM that allow the user to build a two-dimensional view of an object. The thermodynamics group uses ITAM to find out the temperature at certain subcomponents and the volume of air moved through the various parts of the system.

CAEDS is an acronym for Computer Aided Engineering Design Systems. It consists of a solid modeler that operates in three dimensions, a finite element solver that can generate a mesh from a solid geometry, and a systems analysis module that can simulate total system dynamics or perform a static analysis.

PHOENICS analyzes models to determine heat flow, temperature, and other thermodynamic properties. It does not accept graphical input, but it can send a model to a graphics

package known as GRAFFIC for display. The input language is sufficiently rich so that the lack of interactive graphics is not a serious drawback. PHOENICS is the most flexible package allowing user modification of some of its Fortran routines to accomodate special features and properties.

The problem facing IBM is that its product development cycles are too long for today's marketplace. With mainframe conception to production times as long as ten years and personal computer development times of three years, keeping up with the demand for new technology is problematic. IBM's goal is to bring engineering and manufacturing divisions closer together so that lead times can be reduced. One source of delay has been the need to send designs through an analysis support group to ensure compliance with specifications. There are a limited number of engineers qualified to analyze the various properties of a design, such as the temperature at various components, the air flow around boards, or the pressures certain points are subjected to. If this bottleneck is to be eliminated, the analysis packages must be capable of being used by the original designers. A decision support system integrating computer-aided design knowledge is planned for 1990 deployment.

The design analysis packages available at IBM are difficult to use because of their complexity. This complexity

is due to their broad applicability and considerable functionality. The programs are quite powerful and can be used to determine a variety of physical properties for a wide range of designs.

The analysis packages are so complicated that a designer may have trouble deciding which one is appropriate for a particular design. For instance, ITAM only works with two-dimensional models while PHOENICS can do four-dimensional analysis, i.e. three-dimensional analysis in time. This difference is not obvious from a cursory reading of the relevant documentation and even if it was, it might not be clear whether a two-dimensional approach could suffice.

To allow as much generality as possible, the analysis programs are command-driven with hundreds of commands and parameters to learn. A certain problem can be specified in many different ways. The level of detail can be varied and several physical quantities can be predicted. Although default values are helpful, the designer must actively guide the analysis to get useful results. Designers have to know when approximations can be made and have an idea of what analysis areas are important.

Because the three packages used at IBM have different capabilities and are complicated enough so that these differences are not obvious, it is important to address the

problem of choosing which package to run. Picking an analysis tool that is too powerful can lead to extra effort in specifying the design and can generate extraneous information. On the other hand, picking an inadequate package can lead to wasted effort when the designer finds out after entering the design that further progress is restricted and critical information cannot be determined. As part of the overall system to make the analysis easier, a package selector containing knowledge of the various options is essential.

Objective

The overall task involves making the analysis packages easier for the designers to use by putting an expert system around them. This expert system will interact with the designers in terms that they can understand and reduce the complexity of the analysis by narrowing the options at each stage to those that are relevant. The expert system will take the designer's responses and translate them into the language of the analysis package. In this way, design analysis specialists need to be consulted less often and cease to be a bottleneck in the design process. Designers can try out more different designs and can perform deeper analyses than before.

The expert system should be as easy to use as possible. It should be portable, expandable, and efficient. The user

interface should have a logical design and handle input errors gracefully. A good system will have "abstraction barriers", boundaries where modules on either side are independent of each other's implementation, to facilitate change and enhance understanding. Future modifications to the system should be provided for by using structured methods and carefully dividing the knowledge in the package from the inference methods applied to that knowledge.

Approach

As a first step in developing the expert system for design analysis, an intelligent front end (IFE) was built. The purpose of the IFE is to decide which analysis program should be used for a given design. This paper is mainly concerned with the design and implementation of this IFE. Further pieces of the system, such as PAM, the PHOENICS Assistance Module, are covered in less detail.

The IFE is an expert system using a data-driven ("forward-chaining") method to arrive at its conclusion. As all expert systems do, it consists of an inference engine and a knowledge base. The inference engine is in the form of single executable module containing 8086 code compiled from four source files using Microsoft C. The knowledge base is a single sequential input file containing the system's knowledge

in the form of questions, answers, and numerical weights. Development was done on a Sperry PC using MS-DOS 2.11 and Version 3.00 of the C compiler. The IFE is designed to work on any MS-DOS machine with the standard BIOS (Basic Input/Output System), one floppy disk drive, 64,000 characters of memory, and a color or monochrome screen. By modifying a single definition, one can render the program BIOS-independent, allowing its portability to any workstation, microcomputer, or other system supporting the C language.

The design goals outlined in the objective above have been followed as closely as possible. The program contains over 30 separate functions with a minimum amount of information passing between them. Many values, such as the maximum number of questions, have been parameterized to facilitate improvement. The knowledge file itself has a simple enough format so that it can be enhanced with a standard text editor. The user interface takes advantage of all four arrow keys for selecting answer options from a one-line menu and providing full walkback and walkforward capabilities. A help screen is available at any point. The division of knowledge between the sequential file and the computer program has been carefully thought out. Any additional knowledge, including information about a new design analysis package, can be added almost completely through modification of the input file.

The output consists of a list of acceptable analysis packages for a design, chosen from among PHOENICS, CAEDS, and ITAM. Typically, the list will consist of a single package, although it is possible to have a verdict for two or three packages, or even "none of the above." These are displayed on the screen as well as passed on to the rest of the system in an output file. The output file also retains answers that are helpful in later parts of the overall system to avoid repeating questions.

II. Expert Systems

History Expert systems are a branch of artificial intelligence (AI), which is itself a branch of computer science. Computer science began with the first computers, built in the late 1940's. In 1956, the term "artificial intelligence" was coined and the field was started during a meeting at Dartmouth College between Marvin Minsky, John McCarthy, Nathaniel Rochester, and Claude Shannon. (1) AI is a broad field covering robotics, speech recognition, image analysis, expert systems and the study of human intelligence. The first expert system was devised around 1960 by Allen Newell and Herbert Simon of Carnegie-Mellon University and J.C. Shaw of the Rand Corporation. Known as "Logic Theorist", this program proved various mathematical theorems from Principia Mathematica. In some cases, the proofs were more elegant than those discovered by humans. (2)

In 1965, work was begun on DENDRAL, the first program to use heuristics and the first commercial expert system.

DENDRAL's purpose was to analyze organic compounds using mass

(1) Joel N. Shurkin, "Expert Systems: The Practical Face of Artificial Intelligence," Technology Review, 86 (6): 72-78 (November / December 1983), p.73.

(2) Shurkin, p.73.

spectroscopy and its developers were Edward Feigenbaum, Joshua Lederberg, and Carl Djerassi, all at Stanford. By this time, researchers had virtually given up on finding a small set of strategies that could be applied with brute force to any problem. Heuristics substantially reduced the computation time associated with problem-solving. (3) Instead of trying to recreate the human mind, expert systems developers now have the more practical goal of making computers more productive. AI has become another software technology that can be combined with conventional computer systems. (4)

In 1983, there were 200 U.S. researchers in the expert systems field and 500 worldwide. The program base consisted of 50 systems, with only a half dozen making money. Uses included medicine, business management, computer design and repair, and the search for natural resources. (5) Today there are additional applications and major computer hardware manufacturers have substantial AI projects occurring either onsite or through third parties. While previously, special machines specifically designed to run AI applications were used for expert systems, current conventional chips are fast

(3) Shurkin, p.74.

(4) Dwight B. Davis, "Artificial Intelligence Enters the Mainstream," High Technology, 6 (7): 16-23 (July 1986), p.16

(5) Shurkin, p.73.

enough to reduce some of the advantages associated with using specialized hardware. Much of this specialized hardware is designed to use LISP, a language well-suited for AI due to its symbolic processing and self-modifying capabilities. (6)

Current problems keeping expert systems from proliferating very fast include a lack of computer power, deficiencies in the current programming languages so that they fail to capture the subtleties in a problem, and the slow speed at which knowledge engineering takes place. However, fifth-generation computer projects are underway in Japan and the United States, knowledge representation languages are being researched at Stanford, MIT, and Carnegie-Mellon, and improvements in the knowledge engineering process are being actively pursued. Bruce Buchanan at Stanford is working on "knowledge acquisition", where the expert can interact directly with the computer without using a programmer as an intermediary. (7) Since around July 1985, IBM has had an AI Projects Office. Its major AI products have been expert system development packages for its mainframe computers. Outside the firm, developers are targeting their applications for the IBM RT PC engineering workstation. LISP compilers for many hardware targets have made using special-purpose LISP machines less

(6) Shurkin, p.77.

(7) Shurkin, p.77.

critical. (8) To broaden their markets, AI tool kit vendors have rewritten their expert system development tools for conventional machines. They are now introducing new versions written in C, even though it is not a symbolic processing language. The HP Spectrum series of computers can support different kinds of coprocessors. One future combination might be a general-purpose microprocessor teamed with a LISP processor. (9)

AI researchers disagree on how useful and advanced commercial expert systems are. Marvin Minsky sees them as being shallow and even detrimental in that they draw off talent needed to develop models of common sense. (10) Roger Schank decries the lack of understanding expert systems have about their domains. Anything outside their rules is ignored or handled badly, their ability to learn from experience is quite limited, and innovation and reflection are lacking in these systems. (11) Despite these criticisms, expert system technology seems to be firmly planted in the marketplace. As knowledge engineering becomes more efficient, the number of

(8) Davis, p.17.

(9) Davis, p.18.

(10) Shurkin, p.78.

(11) Roger Schank, "Roger Schank on Expert Systems," Publisher's Weekly, 226 (12): 40-43 (September 21, 1984), p.40

expert systems available should expand quite rapidly.

Theory developed within the field of artificial intelligence.

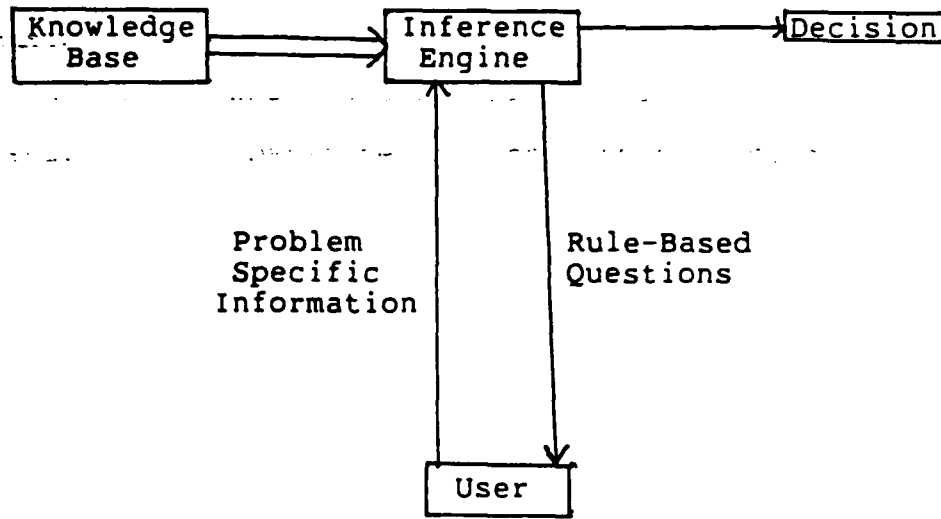
Although expert systems are a fairly new technology, research and experience with commercial applications has built up a body of knowledge about such systems that is widely accepted for the most part. An expert system is defined as "a computer program that mimics a human expert; using the methods and information acquired and developed by a human expert, an expert system can solve problems, make predictions, suggest possible treatments, and offer advice with a degree of accuracy equal to that of its human counterpart." (12) Expert systems perform best in situations characterized by limited possibilities and manageable amounts of information. They are cost-effective whenever there is a great demand for a few experts or when the price of expertise is high. (13)

The basic steps in building an expert system are fairly standard. First, one should find an expert. A single expert is better than several, since the reasoning process needs to be consistent. However, if the experts have different specialties within the area of interest, their combined

(12) Michael Ham, "Playing by the Rules," PC World, 2 (1): 34-41 (January 1984), p.34.

(13) Ham, p.34.

Figure 2: Overview of Expert Systems Theory



knowledge can be quite powerful and overlapping reasoning methods do not create much of a problem. The intuition an expert has, based on experience and knowledge, can be turned into rules. This conversion, known as knowledge engineering, begins quickly but can take years to perfect. Within a week, a programmer can have a rudimentary framework for a specialized field. Working with the expert brings out subtleties and leads to more complicated models of decision making. (14) One key goal of expert system generation is to separate knowledge from the procedures that manipulate it. If the knowledge base is isolated from the programming logic, it can be more readily examined, modified, and maintained. (15) The system should have the capability to grow as new information becomes available, just as an expert can. Adding, deleting, and modifying facts and rules should be straightforward and economical process. (16)

Three types of expert systems have been identified: rule-based, frame-based, and blackboard systems. A rule-based system operates with a knowledge base consisting of facts and rules. Additional facts are elicited from the user for each new problem. The facts cause certain rules to be applied,

(14) Ham, p.36.

(15) Ham, p.36.

(16) Ham, p.38.

generating more facts and eventually a solution. (17) Three types of rule-based systems exist. A "data-driven" or "forward-chaining" strategy, keeps accumulating facts that allow various rules to be activated generating more facts. Typically, more than one rule can be applied at a certain time, so a method for choosing rules deserves careful consideration. Poor rule selection can lead to a system widening the scope of its inquiry without coming any closer to a conclusion. In a "goal-driven" or "backward-chaining" system, a conclusion is chosen and facts are elicited to support that conclusion. If this goal becomes inconsistent with the information supplied, a new conclusion is chosen for examination. The disadvantage of this method is that time can be wasted on erroneous goals that could easily be dispelled if the user could volunteer some additional information. The third type of rule-based system addresses this problem. In this "mixed-mode" strategy, goals are pursued but additional user-supplied information is welcome. (18)

Besides rule-based expert systems, there are frame-based and blackboard structures. In frame-based systems, the knowledge is kept in "frames", miniature databases arranged in a hierarchy, with child frames inheriting attributes from

(17) Ham, p.36.

(18) Ham, p.37.

their parent frames. Frames are used to store rules, calculation procedures, and pointers to other frames. Specialties dealing with object classes find this strategy particularly useful. Blackboard systems act as expert system conferences. The "blackboard" is a database where conclusions are shared between several otherwise independent expert systems. The knowledge sources use each other to cover a wider specialty. (19)

Two major issues come up in the development of expert systems. One is the role of uncertainty. Experts often have to make judgments based on incomplete information, loosely correlated relationships, and exception-riddled rules. Expert systems typically maintain certainty factors for their facts and rules. However, research has shown that ordinary probability is not sufficient enough for manipulating uncertainties due to the high degree of interrelationships within a problem. One promising finding is that precision in certainty values is not a major requirement. One decimal place of accuracy works almost as well as three. (20) Another issue is that of auditing the computer's decision. Systems are much more likely to be used, have their advice followed, and be maintained properly if they provide a method for

(19) Ham, p.37.

(20) Ham, p.37.

justifying decisions through recreation and explanation of the decision-making process. Supporting arguments for a conclusion give the user confidence in what might otherwise be considered a maverick result. Backtracking through the reasoning process is crucial during development. Programmers can determine if the right answer is being given for the wrong reasons. Experts can find places where their knowledge has been misinterpreted. (21)

Accuracy and maintainability are additional goals. Expert system performance should at least match that of the experts in its area of specialization. Although this accuracy may be low, it is superior to that provided by any alternative. Increased performance can be achieved through additions and modifications to the knowledge base. In general, increasing the knowledge base is superior to changing the program logic. Therefore, one should avoid a rigidly-organized collection of rules. (22)

One interesting result gained from keeping the knowledge base flexible is that it can usually be replaced completely by another set of knowledge. The program itself is known as an "inference engine." Inference engines can be and are marketed separately, allowing many applications to be developed simply

(21) Ham, p.38.

(22) Ham, p.38.

by generating the right set of rules and facts. (23)

Critics find expert systems theory lacking. According to MIT's Randall Davis, human experts have seven attributes: "They can solve problems. They can explain the results. They can learn by experience. They can restructure their knowledge. They are able to break rules when necessary. They can determine relevance. And their performance 'degrades gracefully' as they reach the limits of their knowledge." Only the first three characteristics are true for expert systems. (24) Schank believes that current systems are applicable to straightforward applications where all the details can be compiled, but are inadequate in situations requiring learning and insight. He feels that the current knowledge engineering approach is misguided, focusing on rules and recipes rather than the underlying thought processes. (25)

Applications

As mentioned earlier, DENDRAL was the first commercial expert system. Since its development, many more systems have been developed and a few are mentioned consistently in the literature, representing various fields of application and

(23) Shurkin, p.75.

(24) Shurkin, p.75

(25) Schank, p.43.

various development methods and theories. DENDRAL began as a system for determining the chemical structure of a complex compound using mass spectrum analysis. It serves as a good example of what additional knowledge can do to improve a system. Using mathematical rules, the structure of $C(21)H(44)S(2)$ can be one of 44 million possibilities. Adding chemical topology rules reduces this to 15 million and mass spectrum analysis knowledge brings the number of possibilities down to 1.3 million. Upgraded to use nuclear magnetic resonance results, DENDRAL can determine a single structure for the compound. (26) This millionfold improvement was probably achieved by increasing the knowledge base by ten to twenty percent.

MYCIN is a good example of the division between inference engine and knowledge base. MYCIN was developed by Edward Shortliffe, a Harvard premed student, and the medical experts Stanley Cohen and Stanton Axline. (27) Its specialty is diagnosing meningitis and its knowledge base consists of 450 rules and 1000 facts. Facts are all of the form "The <attribute> of <object> is <value> with <certainty factor>." A rule is applied when a fact or set of facts are true with sufficient certainty and its application results in an

(26) Ham, p.38.

(27) Shurkin, p.74.

additional fact. (28) When the knowledge base for meningitis diagnosis was removed, MYCIN's creators were left with EMYCIN, (for Essential MYCIN), an inference engine which IBM later used to diagnose malfunctions in computer disk drives. EMYCIN was also used to build SACON (Structural Analysis Consultant), a program that identifies for structural engineers the best strategy for using a complex computer simulation program. (29) Although MYCIN is famous, it is not used much because of the excessive time it takes to interact with the program. Diagnosing meningitis is usually a time-critical task, so physicians rely on their quicker intuitive resources. (30)

Another diagnostic system is PUFF, which is used daily at the Pacific Medical Center in San Francisco to diagnose lung diseases. PUFF has only fifty rules, but it performs as well in its field of expertise as MYCIN does in meningitis cases with 450 rules. CATS-1 is General Electric's diesel locomotive repair aid which requires 550 rules to be helpful in 50% of repair situations and 1500 rules to be of use in 80% of the possible cases. (31) CATS-1 has been converted to run on a personal computer, an increasingly common occurrence as

(28) Ham, p.36.

(29) Shurkin, p.75

(30) Ham, p.40.

(31) Ham, p.36.

PC's increase in power. In general, the large amount of rules and facts and the size of the source code makes a hard disk essential. (32)

Several systems include automatic data collection to reduce the amount of time required for human interaction and thus increase the number of situations where the expert system's advice can be elicited. ONCOCIN determines therapies for cancer patients by being part of a record-keeping program. As the physician fills out a patient's form at a terminal, ONCOCIN notes the answers and has a suggested therapy filled in at the bottom of the form when the physician reaches it.

(33) A system known as HELP (Health Evaluation through Logical Processing) is even more integrated. Developed by Homer Warner at the University of Utah School of Medicine in Salt Lake City over a period of 18 years, it has been installed in about six hospitals, including the Ainet-Ogden Memorial Hospital in Elmira, New York. HELP maintains complete patient records and has reduced nurses' paperwork by two-thirds. Its advice to physicians is followed 80% of the time. The intensive care unit has four of its beds hooked directly into HELP for automated recording of important

(32) Ham, p.41.

(33) Ham, p.40.

patient variables. (34)

Another medical expert system is Internist, developed by Jack Myers, a specialist in internal medicine at the University of Pittsburgh and Harry Pople, Jr., a computer scientist. Internist has information about 500 diseases in its knowledge base and can diagnose 75% of major medical problems. An improved version, known as Caduceus, incorporates a model of the human body and its functions. In a test, Internist was right 25 out of 43 diagnoses, while the physicians on the cases were right 28 times and clinical experts were correct 35 times. With Internist comparable to experts, Caduceus has a good chance of performing better than the experts. (35)

Expert systems are at work in the computer industry as well as in the health profession. The most widely known is XCON (Expert Configurer), an advanced version of R1, developed jointly by Carnegie-Mellon and Digital Equipment Corporation. XCON decides how to configure VAX computer installations to user specifications. The system saves DEC between \$18 and 20 million per year in manufacturing costs by reducing false orders for unneeded components. Xerox developed a system

(34) Patricia Mandell, "Computers That Humanize Health Care," Ms, 14 (11): 103-104 (May 1986).

(35) Shurkin, p.76.

Figure 3: Examples of Expert Systems

<u>System Name</u>	<u>Purpose</u>
MYCIN	Meningitis diagnosis
EMYCIN	Expert system shell (inference engine)56789012
DENDRAL	Determine structure of organic compounds
SACON	Assist in use of structural analysis program
PUFF	Diagnose lung diseases
CATS-1	Diesel locomotive repair assistance
ONCOCIN	Determines cancer therapies
HELP	Hospital record-keeping, and treatment advice
Internist	Diagnose internal diseases
Caduceus	Add human functions model to Internist
XCON	Configuring minicomputer installations
M.I.	Inference engine and development environment
Trillium	Coordinate development of copier interfaces
ACE	Analysis of telephone cable repair reports
Prospector	Location of mineral deposits
ESE/MVS	Inference engine for MVS computers
KEE	Inference engine and development environment

called Trillium which facilitated efficient collaboration between designers in the development of copier interfaces.

(36) Bell Laboratories has a system known as ACE that analyzes repair reports from telephone cable repairmen. (37)

Other systems include Prospector by SRI International, which contains the knowledge of many different ore specialists, and is used to locate mineral deposits. (38) Teknowledge of Palo Alto has an expert system to decide what actions to take when a well bit gets stuck during oil drilling. With oil rig idle time costing \$100,000 per day, there is no time to fly in drilling experts. Cognitive Systems of New Haven has a system to help insurance agents choose the best policy combination for customers. (39)

Inference engines are being marketed by a number of firms. IBM sells Expert System Environments for MVS and VM machines. Teknowledge has a package called M.I., Intellicorp sells KEE (Knowledge Engineering Environment), and Texas Instruments markets the TI Personal Consultant. (40)

(36) Davis, p.16.

(37) Shurkin, p.77.

(38) Ham, p.36.

(39) Shurkin, p.77.

(40) David Stamps, "Expert Systems, Software Gets Smart -- But Can It Think?," Publisher's Weekly, 226 (12): 36-39 (September 21, 1984), p.38.

In summary, the expert systems commercial field is well-populated, but has plenty of room for additional players. Many systems have been written in traditional artificial intelligence languages sometimes for specialized machines, but packages for microcomputers are becoming popular as well. Personal computers have limitations that make expert system development challenging, but their power is increasing. There is some movement towards using C as a language for writing expert systems, but some applications need the speed of a dedicated LISP machine. However, integration of data processing and knowledge processing seems to be a future goal; coprocessors could provide an efficient implementation of this integration.

III. Computer-Aided Design

History

Computer-aided design, commonly known as CAD, is "a discipline that provides the required know-how in computer hardware and software, in systems analysis and in engineering methodology for specifying, designing, implementing, introducing and using computer-based systems for design purposes." (41) CAD has passed through three distinct phases in its relatively short existence. In the 1960's, high technology companies developed CAD systems for their own use. During the 1970's, companies specializing in turnkey systems sold minicomputer-based packages to users in medium and large firms. In the 1980's, CAD software vendors are targeting superminicomputers or workstations with their packages. (42) Microcomputer-based solutions are also increasing in popularity.

In 1963, one of the first CAD systems was developed by Sutherland. Known as SKETCHPAD, it allowed interactive manipulation of graphical images. In 1964, IBM completed for the General Motors Research Laboratories a system called DAC-1

(41) Jose Encarnacao and Ernst G. Schlectendahl, Computer Aided Design, (Berlin: Springer-Verlag, 1983), p.3.

(42) John Stark, What Every Engineer Should Know About Practical CAD/CAM Applications, (New York: Marcel Dekker, Inc., 1986), p.40

for "Design Augmented by Computer." It was less interactive than SKETCHPAD, but it was probably the first commercial CAD package. Bell Laboratories developed the GRAPHIC 1 remote display system in 1965 for arranging printed-circuit components and wirings, and for composing and editing text. GRAPHIC I introduced the concept of distributing CAD processing power between local interactive workstations and a central host computer. In 1966, IBM developed a system to aid in the design of hybrid integrated-circuit modules used in its System 360. RCA brought out GOLD in 1972 to help it design integrated circuit mask layouts. (43)

The first half of the 1970's was a period where much headway was made on developing CAD theory. Much of the foundation for today's industry was laid during this time. Lockheed demonstrated the cost-effectiveness of computer graphics in 1973 and the second half of the 1970's brought CAD out of the laboratory as it became economically attractive to more and more firms. Today, the CAD market is firmly in place and experiencing rapid growth. (44)

Computer-aided engineering (CAE) combines CAD and computer-aided manufacturing (CAM). Previous to 1981, CAM had followed a different path, beginning independently with the

(43) Encarnacao and Schlectendahl, p.9.

(44) Encarnacao and Schlectendahl, pp.9-10.

first automatically controlled milling machine, which used MIT's Whirlwind computer and led to the Automated Programmed Tool. (45) CAE's revenues in 1984 were \$276 million and are expected to pass the \$2 billion mark in 1989. (46) The total worldwide computer graphics market in 1983 was \$3 billion and growing at 30% per year. (47) Leading the CAE field are three relatively small Silicon Valley companies, Daisy Systems, Mentor Graphics, and Valid, with 79% of the 1984 market. However, many small companies are undercutting the expensive workstations with personal computer solutions that provide almost as much power at more reasonable price. While the major players sell systems for an average of \$45,000 each, PC systems run from \$8500 to \$35,000. (48) Personal computers cannot provide solid modelling yet and their resolution and color selection are poorer than in mainframe and minicomputer application. Finite element analysis is too CPU-intensive for PC's and other advanced commands are unavailable as well. However, most sophisticated CAD functions available only on mainframes are rarely used. A PC CAD system typically needs

(45) Encarnacao and Schlectendahl, p.9.

(46) John Paul Newport, Jr., "How PC's Shook an Industry," Fortune, 112 (6): 105-106 (September 16, 1985), p.105.

(47) Gary R. Bertoline, Fundamentals of CAD, (Albany: Delmar Publishers, Inc., 1985), p.8.

(48) Newport, p.105.

512K of memory, a hard disk, a coprocessor, and a light pen or digitizing tablet. (49)

In December 1984, there were over 15,000 PC's being used for computer-aided drafting and design (CADD) and a growth rate of 63%. The least expensive packages, designed to run on existing PC's, were priced as low as \$1000. Learning times are also low. PC CADD systems typically take less than a month to learn while workstation-based packages require three months of learning. (50) As of May 1985, over 20 CADD systems for personal computers were available, including AutoCAD, VersaCAD, CADPlan, MicroCAD, RoboCAD, and CADD/2D. As personal computers become more powerful, microprocessor-based CAD systems should perform as well as today's workstations. Meanwhile, workstation capabilities will approach those of today's mainframes. (51)

Theory

Computer-aided design in its traditional form consists of interactive graphics for building a model and a series of

(49) Alex Lee, "Engineering Design on Micros," Computers and Electronics, 22 (9): 86-90 (September 1984), p.86.

(50) Eric Teicholz and Dan Smith, "Where are We and Where are We Going on PC's?", Architectural Record, 173 (6): 47-49 (May 1985), p.47.

(51) Teicholz and Smith, p.49.

tools that use the model after it has been made, either for design analysis or for programming the machines and robots that will build a product according to the design. . . . Because computer-aided manufacturing is usually incorporated, many packages are known as CAD/CAM systems. The major benefits provided by a CAD/CAM system are reduced cost and cycle time and improved quality in the design and manufacture of a new product. (52) Because the design phase can determine whether a product gets to market at all, CAD/CAM is doubly important because its increased productivity effect is felt most strongly during design. (53) These systems do have drawbacks, however. They are expensive and may be too difficult for some engineers to use on a regular basis. Virtually all are incapable of designing a part from user specifications alone. Neither can they choose the better of two competing designs. There is no standard CAD/CAM system that is "the best" for all products or all applications. (54) Although many systems try to incorporate as much functionality as possible, a given system is always missing a few vital features. (55)

CAD/CAM software can be classified into six levels. Next

(52) Stark, p.13.

(53) Stark, p.11.

(54) Stark, pp.15-17.

(55) Teicholz and Smith, p.49.

to the hardware is the computer systems software. On top of that is graphics software, followed by the CAD/CAM systems software. Product modelling software is the next level, while applications software and user-developed programs complete the hierarchy. (56) Different systems have different modelling capabilities. Some support three-dimensional modelling while others are limited to two dimensions. Two-dimensional models have the drawback that different views of the same component are not logically connected within the program, so that changes in one view usually must be duplicated in others. Within a three-dimensional framework, there are three types of geometric modelling: wireframe, surface, and solid modelling. Wireframe modelling defines the edges of the figure, surface modelling describes the surface between the edges, and solid models know about the material inside the object, under the surfaces. (57) For most surface and solid models, geometrical characteristics can be determined. These include volumes, surface areas, moments of inertia, lengths, and angles. (58) Mass properties can also be determined from solid models, using user-supplied dimensions and densities. Among these properties are volume, weight, center of gravity, and moments

(56) Stark, p.21.

(57) Stark, p.34.

(58) Stark, p.36.

of inertia. (59)

Solid modelling is typically done in one of two ways. For most models, primitive modelling will work. Intersection, union, or differencing of standard building blocks is repeated to generate the model. Although programs typically come with a dozen primitives, only four are necessary: the plane, cylinder, cone, and sphere. (60) To design a bolt, one could halve a sphere with a plane, attach a cylinder and thread it with a filleting tool made from a cone somehow. For more complex models, such as automobile exhaust manifolds, boundary modelling can be used. Two dimensional outlines of any shape are "raised" to some desired thickness. The Pentagon Building could be modelled by drawing a pentagon with a smaller pentagon inside it and raising this shape to the proper height. The principle at work in boundary modelling is to define the geometry and topology separately. (61) In either kind of solid modelling, graphics functions are available for rotation, translation, scaling, duplication, and cross-sectioning. (62)

(59) Bertoline, p.289.

(60) C.B. Besant and C.W.K. Lui, Computer-Aided Design and Manufacture, (Chichester: Ellis Horwood Limited, 1986), p.158.

(61) Besant and Lui, p.160.

(62) Besant and Lui, p. 162.

Various kinds of analysis can be done on completed models. The application of stress to a particular point or surface can be simulated and the resulting distortion displayed graphically. (63) One way of carrying out stress and other analyses is through finite element analysis, a technique originally used in the aircraft industry. The CAD system carries out the difficult step of mesh generation, the division of the model into small standardized elements. (64)

The user interface to a CAD/CAM system is a very important component. A CAD system should help the designer in every way possible. Not only should functions for specific tasks be readily available, but coordination between designers doing different tasks on the same system or model should be supported. (65) User complaints concerning a CAD system most frequently concern the inadequacy of menu systems, online documentation and error messages. Complex human-computer interfaces that require special expertise can be damaging to an organization by forming a powerful elite composed of those who can use the system. (66) A design bottleneck is created

(63) Bertoline, p.289.

(64) Besant and Lui, p.23.

(65) Vivienne Begg, Developing Expert CAD Systems, (London: Kogan Page, 1984), p.33.

(66) Begg, p.34.

which can be (and has been) used to hold the company to ransom. (67)

To avoid this bottleneck, Begg advocates adding an expert system to a CAD package to make it simpler to use. It could be one of two forms. It could act as a translator and interpose itself between the designer and the CAD system or it could act as a consultant. (68) The translator would work with several design languages, each at a different level of abstraction. Automated design procedures would be provided for use with the higher language levels. (69) Advantages of the consultant model would be the preservation of formalism and conventional algorithms by keeping the heuristics away from the CAD package. (70)

Applications

Among the many areas where CAD/CAM systems can be applied are molecular structure modelling in chemistry, animation, medical research, aircraft flight simulation, integrated circuits and printed circuit board design, and structural design in the aircraft, shipbuilding, and automobile

(67) Begg, p.35.

(68) Begg, p.78.

(69) Begg, p.73.

(70) Begg, p.78.

industries. (71). Drafting packages can make architectural and mechanical drawings as well as block diagrams, electronic schematic drawings, and wiring diagrams for electrical engineering applications. (72) Computer-aided manufacturing systems use CAD output consisting of parts definitions and assembly relations to program numerically-controlled machines and robots. (73)

Specific systems are numerous. Architects can choose from over 126 computer-aided drafting and design (CADD) systems and over 67 architectural engineering packages. (74) HOUSE24 is a system that makes drawings and generates bill estimates for platform frame construction houses in Japan. (75) CADIC is an integrated circuit design package with four modules. MANCAD accepts a manually-entered description of an integrated circuit layout and converts it to an appropriate data structure for use within the system. CADIC1 is an interactive design aid that can be used to manipulate the circuit layout.

(71) Besant and Lui, p.22.

(72) Bertoline, pp.294-298.

(73) Besant and Lui, p.318.

(74) Joan Blatterman, "1985 Guide to Computer Software for Architects and Engineers," Architectural Record, 173 (12): 49-80 (October 1985), pp.62-79.

(75) Joanna Wexler, ed., CAD84: Sixth International Conference and Exhibition on Computers in Design Engineering, (Surrey: Butterworth & Co. Ltd, 1984), p.38.

Figure 4: Examples of Computer-Aided Design Systems

<u>System Name</u>	<u>Purpose</u>
SKETCHPAD	interactive graphics manipulation
DAC-1	drafting for General Motors
GRAPHIC 1	printed-circuit component arrangement
GOLD	integrated circuit mask layout for RCA
HOUSE24	drawings and estimates for Japanese houses
CADIC	integrated circuit design
MENULAY	user interface construction
MAPLE	microcomputer backplane configuration
(other)	molecular modelling
	animation
	medical research
	aircraft flight simulation
	structural design
AutoCAD	computer-aided
VersaCAD	design packages
CADPlan	for microcomputers
MicroCAD	

DRCCAD is a design rule language compiler that converts design rules to a form that can be understood by CADIC2, the online design rule checker that evaluates the integrated circuit for conformance to the supplied rules. (76)

MENULAY, written by Martin Lamb, provides some of the functionality that Begg suggested for a CAD expert system. It constructs user interfaces using input in the form of intuitive gestures, presumably pointing to icons. People who have no computer knowledge can design and improve software interfaces. The end result is a C program that goes on top of the application. MENULAY has been used on its own interface, on a sketch editor, a computer-assisted instruction program, and a musical notation editor. (77)

A system called MAPLE configures the backplanes of microcomputers. Written by J.A. Bowen at the University of Reading, it takes a specification consisting of hardware functionality, software functionality and design constraints, and decides which boards and software packages from its library should be installed to meet the requirements. (78)

Expert systems for CAD include SACON, previously-mentioned, which acts as a structural analysis

(76) Wexler, p.68.

(77) Wexler, pp.8-10.

(78) Wexler, p.547.

consultant. A package known as ELAS works with complex oil-well log analysis programs and goes a step further by imposing itself between the user and the analysis system. (79)

The next chapter describes three CAD packages that are used by IBM to carry out design analysis: PHOENICS, CAEDS, and ITAM. The knowledge base of the intelligent front-end described later in this paper is based entirely on information known about these packages.

(79) Begg, p.77.

IV. Design Packages Under Consideration

PHOENICS

PHOENICS is a design analysis program sold by CHAM Limited that can be used to model fluid flow, heat transfer, and combustion. The input to the system consists of twenty-four groups of commands, dealing with the grid definition, boundary conditions, selection of various parameters, and output format and regions of interest. The default values for most of the parameters are quite reasonable in many cases, so that it is almost never necessary to supply all twenty-four groups. In addition, the system comes with a library of models that can be adapted for a wide variety of situations. The input is all textual in nature. Grids are defined by specifying the cell boundaries and total length for each dimension: x, y, z and time if desired. Coordinate systems can be cartesian, cylindrical, or user-defined curvilinear. Boundary conditions and sources are given in terms of the cells they operate on, which side of each cell is affected, and two values that can be manipulated to represent a fixed-flux or fixed-value situation as well as variable flux. A number of variables can be solved for simultaneously. Solutions are done with an iterative method. Output control consists of specifying the variables and the cells to print. The output can be strictly numerical or a contour plot.

Figure 5: Sample PHOENICS Documentation

- GROUP 2. Time dependence and related parameters.
- GROUP 3. x-direction grid specification.
- GROUP 4. y-direction grid specification.
- GROUP 5. z-direction grid specification.
- GROUP 6. Body-fitting and other grid distortions.
- GROUP 7. Variables (including porosities) named, stored & solved.
- GROUP 8. Terms (in differential equations) and devices.
- GROUP 9. Properties of the medium (or media).
- GROUP 10. Interphase-transfer processes and properties.
- GROUP 11. Initialization of fields of variables, porosities, etc
- GROUP 12. Adjustments to fluxes of convection and diffusion.
- GROUP 13. Boundary and internal conditions, and special sources.
- GROUP 14. Downstream pressure (for free parabolic flow).
- GROUP 15. Termination criteria for sweeps and outer iterations.
- GROUP 16. Termination criteria for inner iterations.
- GROUP 17. Under-relaxation and related devices.
- GROUP 18. Limits on variable values or increments to them.
- GROUP 19. Data communicated by satellite to GROUND
- GROUP 20. Control of preliminary printout.

- GROUP 21. Frequency and extent of field printout.
- GROUP 22. Location of spot-value & frequency of residual printout.
- GROUP 23. Variable-by-variable field printout and plot and/or tabulation of spot-values and residuals.
- GROUP 24. Preparations for continuation runs.

For more information on any individual group n, type GROUP n.
 All integers and reals are defaulted to 0 or 0.0 unless otherwise indicated by <default value> after variable name.
 Defaults of all logicals are indicated thus <default value>.

** GROUP 1----- GROUP 1

- GROUP 1. Run identifiers and other preliminaries.
- * Command TEXT(Any message up to 40 characters ... will cause the message to be printed out by EARTH.
 - * REAL(A,B,C,....) used to declare local user variables, up to 20 allowed.
 - * INTEGER(L,J,K,....) used to declare local user variables, up to 20 allowed;

For help, simply type the variable or command name.

** GROUP 2----- GROUP 2

- GROUP 2. Time-dependence and related parameters.
- STEADY<T>,TFIRST,LSTEP<l>,TLAST<l.0>,FSTEP<l>.
 TFRAC(1-ntfr)<ntfr*1.0>..ntfr<100>
 Note: ntfr is set in MAIN of the SATLIT file.
- * Command for setting "power-law" time intervals is: GRDPWR(T,LSTEP,TLAST,POWER)

Although there is no graphics front-end, the modelling commands are quite flexible and modelling can be done from an engineering drawing or even a sketch. Grids containing rectilinear cells of equal size can be generated by giving one command for each dimension specifying total length and number of divisions. Making the cells smaller in areas of particular interest can be done by using a power law to distribute the boundaries. (The equal size case is actually done using a power law with an exponent of 1.) Cylindrical-polar coordinates are another option which can facilitate the modelling of designs containing rounded components. Cells are created by taking a cylinder with the top being flat and making cuts at constant heights, constant radii, and constant angles. For the most flexibility, a curvilinear grid system can be specified. Cell faces still touch the same adjacent cell faces completely but the cell shape is up to the user. This system is also known as a "body-fitting coordinate system" since each cell can be a miniature of the whole object, causing the outermost cell boundaries to define the outside of the objects.

Boundary conditions and sources, including the physical boundaries of the component are specified by naming a group of cells through a PATCH statement. One named, a patch can be specified in any number of COVAL statements, each of which

determines the behavior of a variable within each cell or through a particular boundary in each cell. Variables associated with the grid dimensions can be used to make boundary conditions independent of how many cells the grid has or what size it is. Complicated obstructions can be simplified by introducing a porosity variable that defaults to 1.0, but can be set to any value from 0 meaning a solid blockage to 1.0.

The default solution method is to take x-y planes and determine for each cell and variable what the value at the center of the cell is using a weighted average of the variable values at each face. This is done iteratively until a certain user-supplied number of iterations is reached or until the corrections made on an iteration fall below a certain user-given value. The program does this a number of times in vertical sweeps from low z to high z. If the model is dynamic, the top level sweep is done in time, from earliest to latest. No repetition is necessary since causality makes it impossible for an event or value to affect an earlier event or value. Other solution methods include the whole-field method which iterates over all the cells rather than just those on a single plane, using more memory but increasing execution speed, and the parabolic method, where the user assures the system that fluid flow is upward and only one pass in the z

direction is needed per time frame.

Output is done through PATCH commands that select the cells to be displayed. The format is typically the numerical values of a specified variable in the center of each cell in the PATCH. Crude contour plots suitable for sending to a line printer are also a feature. There is a package known as GRAFFIC which can take numerical data and draw three-dimensional representations of it. GRAFFIC will also work on PHOENICS input geometry so that it can be checked against reality.

CAEDS

CAEDS is a CAD system licensed to IBM by Structural Dynamics Research Corporation. The acronym stands for "Computer-Aided Engineering Design System" and is quite accurate in that CAEDS is a complete system. By integrating an interactive graphics design function with complete analysis facilities, it speeds up the design process and allows efficient comparison of alternative designs. The graphics system can do two- or three-dimensional modelling on a graphics workstation. Originally confined to mainframes, it was recently introduced on an IBM RT PC workstation. It contains a solid modeller that can do primitive modelling as well as boundary modelling. The solid modeller can regenerate

a mesh for finite element analysis or if the structure can be represented by interconnected beams, a frame analysis system is available. All modules are menu-driven with a feature that allows the user to bypass intermediate menus by supplying several menu choices at once.

Finite element analysis consists of decomposing the model into a finite number of idealized elements interconnected at a finite number of points. A system of equations based on known quantities is solved and values for all elements are determined. The CAEDS finite element solver is known as SUPERB and can determine the static displacements, forces, stress and strains of complex structures in response to concentrated or distributed external forces, thermal expansion, enforced displacements, accelerations and centrifugal loads. It can also determine the natural frequencies and mode shapes of complex structures, and can calculate the modal participation factors and modal coefficients for shock spectrum input. CAEDS provides heat conduction analysis in the steady-state considering the effects of internal heat generation and convective heat flux at element surfaces.

Models can be represented in several coordinate systems, including Cartesian, cylindrical, and spherical. Local Cartesian systems based on a certain node can also be used.

TASK — MODEL PREPARATION
LOAD — NODAL FORCE — DEFAULTS

HELP
 TASK
 MODEL FILE
 DISPLAY OPTIONS
 VIEW OPTIONS
 GROUP

ORIENT
 NODE
 ELEMENT
 COORDINATE SYSTEM
 LOAD
 RESTRAINT
 COUPLES
 KINEMATICS
 MULTI POINT CONSTRAINTS

NODAL FORCE
 ELEMENT PRESSURE
 DIRECTORY
 CURRENT
 DELETE SET
 RENAME
 STATUS

DEFAULTS
 CREATE
 SKETCH
 LABEL
 LIST
 DELETE
 MODIFY

The sub-operation **DEFAULTS** enables the user to force parameters to be used for the next nodal force

the **DEFAULTS** sub-operation offers the following s

FORCE
COLOR
PALETTE
STATUS

Select **FORCE** and the system will prompt

ENTER 6 FORCES

requesting the user to specify the forces to be assigned next nodal forces. Values entered are the forces in the present Coordinate System of the selected nodes.

Select **COLOR** and the system will prompt

ENTER NODAL FORCE COLOR NAME OR NO. (default)

requesting the user to enter the color code to be assigned next nodal forces.

When **PALETTE** is selected from the sub-menu the user sees the following sub-menu:

FIXED_COLORS
ENTER
MODIFY
DELETE
LIST
RENAME
STATUS

The same sub-menu is available within all entity demand operations. It allows the user to create, modify, list and rename user-defined colors.

Enter **FIXED_COLORS** and the system will display of system defined colors with their corresponding color at the top of the graphic display region on super terminals.

Enter **ENTER** to create a user-defined color. The prompt:

ENTER COLOR NAME OR NO. (default)
 (integer input value must be > 16)

Enter the name or number of the color to be created 16 system-defined colors (see **STATUS**). The system accept existing color names or numbers. The system

ENTER % RED, GREEN, BLUE (default percentages)

Enter the desired percentages of red, green and blue system will report:

New color number, new color name

However, the modelling system allows only Cartesian coordinates in three-dimensional modelling and Cartesian or polar for two-dimensional models.

The solid geometric modeller can be interfaced to the CADAM and CATIA design systems, allowing the transfer of designs from those packages to CAEDS. The primitive geometric elements provided include the block, cone, cylinder, hexahedron, sphere, tube, and quadrilateral. Solids can be created in five different ways: a primitive can be dimensioned as an object, objects can be added or subtracted from each other, two-dimensional profiles can be revolved or extruded, a set of cross-sections can be "skinned", or elemental parts can be combined. Extrusion is basically the boundary modelling described in Chapter II. Skinning involves fitting a surface to the cross-sections which have been lined up parallel to each other. For instance, a tetrahedron could be modelled by skinning a set of parallel equilateral triangles.

The model is treated as a single object and only one object at a time can be active in the modeller. This does not preclude bringing two objects into the workspace in order to combine them. For each object, or piece of an object, the user can supply inertial properties, surface area, volume, density, and mass. Of course, CAEDS will attempt to calculate values that are not supplied based on known values. When

doing finite element analysis, isotropic, orthotropic, and temperature dependent material properties can be supplied. Finite elements available for steady state heat conduction analysis include a one-dimensional bar, a two-dimensional flat surface, an axisymmetric solid represented in two dimensions, a curved surface, a thick shell or a solid. Poisson's equation for steady-state temperature distribution is used.

ITAM

ITAM is an analysis program that uses the CADAM graphical design package as its preprocessor. The two-dimensional model built in CADAM is a nodal network which represents a cross-section of some mechanical product that has an airflow. This limits the products that can be analyzed to those with fans. These include personal computers, large disk drives, tape drives, many terminals, and of course, mainframes. Air inlets and outlets are supplied by the user as well as all the obstacles encountered by the air flow as it travels through the enclosure. Temperature sources are also user-supplied as well as the magnitude of the fluid flowrate at the inlets.

As a single cross-section is used for the analysis, care must be taken that as many inlets and outlets are included in the chosen section. If there are fans on the top and bottom of the machine, a cross section in the x-z plane would be

appropriate. Lack of symmetry can complicate matters. An obstacle that occupies only half of the ignored dimension might be modeled as being half as wide with some loss of accuracy in the results. Of course, multiple cross-sections can be done in separate runs. However, the method for combining results may vary from model to model.

ITAM will compute temperature, static pressure, and fluid flowrate at critical locations. Using this information, designers can determine if component temperature tolerances are being exceeded.

Comparisons

The three packages described above vary considerably in complexity, functionality, applicability, and in their user interfaces. ITAM is certainly the easiest to use, while PHOENICS and CAEDS are both fairly complex in different ways.

ITAM is also the least general, confining its analysis capabilities to fluid flow problems where virtually all the fans reside in a single cross-section. PHOENICS has the weakness of not having interactive graphics capabilities although its GRAFFIC interface can at least assure the user that the set of inputs represents a physical reality. CAEDS uses a very powerful modeller capable of working with solids while ITAM has CADAM which is constrained to two dimensions.

Figure 7: Comparison of Design Analysis Packages

PHOENICS	CAEDS	ITAM
cannot enter model graphically	solid modelling graphics preprocessor	run from CADAM graphics package
three-dimensional modelling	three-dimensional modelling	limited to two dimensions
can observe variables in user-defined time frames	dynamic analysis yields natural frequencies	all variables steady state
dynamic heat conduction analysis	steady-state heat conduction only	steady-state heat flow only
requires mainframe to run	requires IBM RT PC workstation	requires IBM PC
fluid-flow, heat transfer, combustion	heat-conduction, structural analysis, geometric values	fluid flow, heat transfer
command-driven (24 groups)	menu-driven (100's of menus)	command-driven (few commands)
analysis done on user-defined grid	finite element analysis	analysis done on nodal network
Cartesian, cylindrical, user-defined coordinate systems	Cartesian, cylindrical, spherical coordinate systems	Cartesian coordinates only
does not require fan or pump	does not require fan or pump	requires fan or pump

PHOENICS gives the user full control over the time domain. The user can specify exactly where in time each frame is. CAEDS has dynamic analysis, but it amounts to interpreting the eigenvalue solutions to its equations. Results are in terms of modal quantities and natural frequencies. ITAM analyzes from a single steady-state time frame. CAEDS heat conduction is limited to a steady-state solution as well.

ITAM and PHOENICS both do fluid flow and heat transfer, while CAEDS does heat conduction. CAEDS is not designed for fluid flow but it may be possible to simulate fluid flow with the right set of parameters. ITAM and PHOENICS are also both command-driven, but PHOENICS has many more commands. CAEDS is menu-driven and presumably easier to use. PHOENICS does group its commands into 24 sections to simplify its user interface somewhat.

Being more sophisticated and requiring more memory to handle three-dimensional analyses through time, PHOENICS is confined to running on a mainframe computer. CAEDS has recently been ported to an IBM RT PC workstation. ITAM is simple enough to run on practically any personal computer that supports CADAM.

PHOENICS's problem domain is a user-defined grid of boxes of up to four dimensions while CAEDS does its analysis on a set of finite elements. ITAM is restricted to a nodal network

where inlets and outlets are source and sink nodes and obstacles are dead ends. ITAM's need to have a fan or a pump in the model for the analysis is absent from the much more general CAEDS and PHOENICS.

These are a few of the major differences between the three packages. Experts with experience using each of these packages can undoubtedly find twice as many other differences in their functionality and appropriate problem domain.

IV. Design of GAPS Program

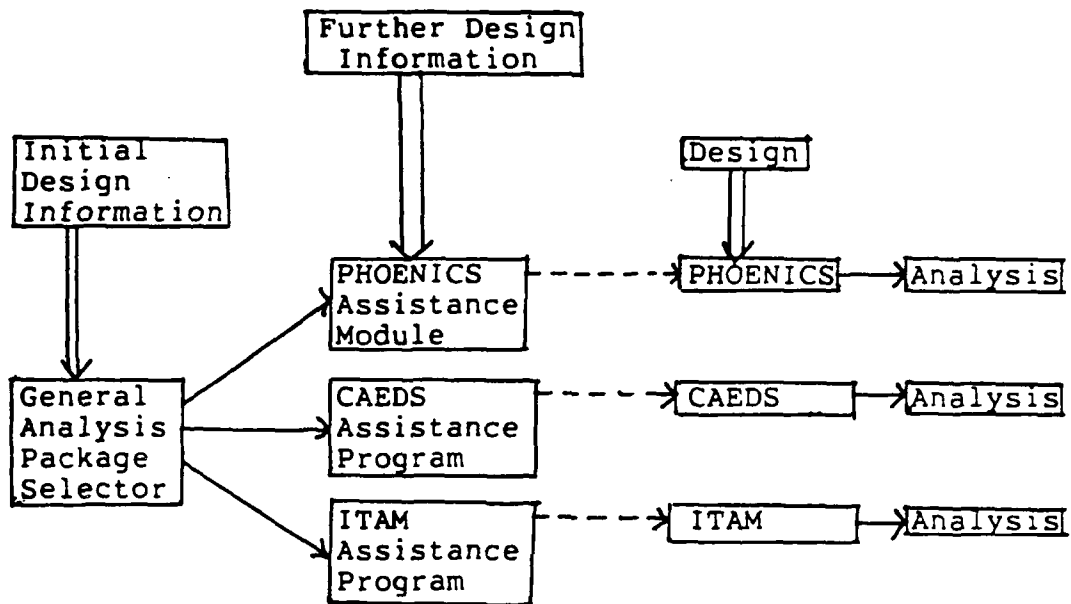
System Overview

The overall system that the intelligent front end will run in will be designed to make the CAD/CAM process as straightforward and fast as possible. Designs will no longer go through a special section devoted to analysis. The engineers who created the product design will perform their own analysis with the aid of an expert decision support system. This system will present the engineer with a more comfortable and intuitive interface for design analysis. Expert systems will presumably be a part of the computer-aided manufacturing side of the product design as well. As a gateway to the design analysis piece of the system will be an intelligent front end, a expert system that chooses a package based on an interactive questioning session. The assistance system for that package will be activated as the user leaves the IFE. Further questions will be asked relating specifically to the use of the chosen product.

Design Considerations

The range of expert systems approaches can be differentiated by the amount of knowledge in the program itself, also known as the inference engine. Knowledge not in

Figure 8: System Overview



the program, the "knowledge base", resides in input files in a fairly generalized format. There are advantages and disadvantages associated with any approach. Generally speaking, the more knowledge is in the program, the faster it will find an answer and the more customized will be the user interface. A finer level of detail is allowed and multiple paradigms can be followed in a system where the knowledge resides in the program. The program appears more intelligent. On the other hand, such a program can be hard to modify when circumstances change or additional knowledge is added. When the knowledge is mostly in a set of input files, changes can often be made without modifying the program at all. The savings in time and money can be substantial, since the knowledge needs less translation, recompilation and program integrity-checking is avoided, and the knowledge engineer does not need to know the computer language being used or the implementation details of the expert system.

The intelligent front end, used to determine which design analysis package to run, was intended to be flexible enough to allow the addition of new information without the program source itself being changed. The kinds of new information covered by this provision include the names of the programs, the distinguishing questions, their menu options, and the importance of each option on the final choice. By modifying

its input file, the IFE can be modified to ask questions about any set of programs. With the exception of the storage allocation parameters and the AI paradigm used to make the decision, all of the knowledge embodied in the IFE is in its input file. The IFE also contains a learning mechanism, whereby it can calculate the expected informational value of a question by observation of the previous responses to that question.

The approach of placing almost all the knowledge outside the program was chosen for three reasons. First, it separated the AI paradigm used from the knowledge the paradigm was being applied to. This abstraction barrier made program development easier by keeping problem-specific details out of the source. Initial testing could be done using a small set of questions and could therefore be more thorough. Second, the approach facilitated the addition and improvement of questions as more knowledge of the analysis packages was attained. This fine tuning is expected to go on over the lifetime of the IFE as new insights are made into the differences between packages. New packages can be considered for addition to the system at lower cost, since the IFE source does not have to be changed to accommodate them. The third reason for the approach was to organize the knowledge in a way that could be understood by those who are not computer programmers. Developing the

program to handle a generic input requires standardization of that input. Instead of having to decipher a convoluted logic structure that somehow manages to accomplish its goal of implementing a certain piece of knowledge, one can clearly see what the complete set of questions is and what importance each menu option for each question has on the choice of analysis package.

Inference Engine

The AI paradigm used in the inference engine, a forward-chaining method, was chosen for its ability to capture the uncertainty inherent in the problem of choosing between a set of programs with overlapping abilities. For each design analysis package, a weight between zero and one is maintained. A zero indicates that the package has been eliminated from consideration while a one indicates that no information has been given that would make the package any less than fully suitable for performing the analysis. Values between zero and one obviously indicate varying levels of uncertainty as to the package's suitability. The weights are updated after each question through multiplication by a set of weights associated with the menu option chosen. The next question is selected based on the expected change in package weights it will have. In other words, at each point, the question with the highest

expected informational value is chosen. The input file consists of the design analysis package names, their minimum weights for consideration, the questions, their menu options and their associated weights. IFE execution stops when there are no more useful questions or when all but one package has been eliminated. For each package in the running, the IFE asks the user whether further assistance is needed. If yes, it chains off to another program specifically designed to ask questions about the package chosen and formulate a strategy for using it.

The intelligent front end was named GAPS, for General Analysis Program Selector. For transportability and maintenance purposes, it was written in Microsoft C with standard MS-DOS BIOS calls for screen control. The number of modules is over thirty since each module was designed for a single purpose whenever possible. A top-down approach was used in the design, meaning that each module had its function analyzed and broken into subcomponents if necessary. Module length is under 100 lines in almost every case. The main program assigns input and output files, and calls three subroutines that read the input file, ask the questions, and write the output files.

Knowledge Base

The knowledge base consists of a standard sequential text file which can be displayed with the DOS "type" command and modified with most word processing programs. The entire knowledge base is read into main memory each time GAPS is run. times. For instance, "Can the fans be intersected with a single plane?" does not make sense if the user has just responded negatively to the question "Is there a fan in the design?" To avoid such a situation, condition records were devised. Each question carries the number of conditions it has in its header. A condition record consists of a question label, an answer, and a flag giving the meaning of a match. Some answers preclude a question while others make questions sensible when they were no times. For instance, "Can the fans be intersected with a single plane?" does not make sense if the user has just responded negatively to the question "Is there a fan in the design?" To avoid such a situation, condition records were devised. Each question carries the number of conditions it has in its header. A condition record consists of a question label, an answer, and a flag giving the meaning of a match. Some answers preclude a question while others make questions sensible when they were not previously so. The question label stays with a question for its lifetime so condition records keep their original meanings. Question insertion and deletion do not disturb conditions for

questions.

Data to pass to the output file has been associated with the answers to some questions. Not only is the text given, but the output file location is too. This is necessary to preserve a certain output file format no matter what the order of the questions is.

Support for Design Decisions

The input file was designed in a hierarchal fashion. Record counts precede the records they are counting to reduce complexity and make the input file more informative to the reader. Analysis package information was placed before the questions because of its smaller record size and the fact that there are fewer packages than questions. All of the information associated with a question is listed before the next question for ease of reading and modification. Identification numbers for questions are used so that additional questions can be inserted without changing condition records on other questions.

The program was designed to consist of many small single-purpose modules for ease of implementation and testing and later modifications. The short module length enhances understanding of the routine and provides more abstraction barriers to hide implementation details from higher levels.

VI. Implementation of GAPS

Format of Knowledge Base

Each line typically consists of one number or one character string. In lines containing character strings, the string is assumed to start at the beginning of the line (column 1) and extends to the next return character, but does not include it. The top two lines contain the number of analysis packages and number of questions. On the third line is a tolerance value for choosing the package. Any package which has a weight within the tolerance value of the winning package's weight is also approved for use. As of this writing, the tolerance has been rather arbitrarily set at 0.1. Following the tolerance are sets of two-line records for each analysis package. The first line contains the name of the package (PHOENICS for example). The second contains a minimum weight that must be maintained for the package to be selected. In the event that no package is above its minimum weight after questioning, the user is notified that no packages are appropriate for evaluating the design. This is a plausible occurrence since the questioning phase requires a weight of zero to eliminate a package from consideration.

After the design analysis program information come all the question records, which are of variable length. The order of questions is significant only to the extent that in the case

of two or more questions tying for highest expected information content, the question appearing first in the input file is asked. Each question has a numerical label that is intended to uniquely identify it throughout the question's lifetime. This label is used by other questions in their condition sections, which will be explained shortly. After this identifying number comes a one-line string containing the question text itself. Following this is the number of conditions. If positive, the question depends on the answer to other questions to decide whether they can be asked.

For each condition, there are three integer values placed on three separate lines. The first is a question identifier, the second identifies a menu option (a one if it is the first option for its question, a two if it is the second option, and so forth), and the third is a flag set to one or zero. If one, the question we are reading can only be asked if the given menu option is selected in reply to the question corresponding to the identifying number. If zero, the question can only be asked if the identified question has not been answered with the given menu option. This includes the case where the question has not been asked at all. The condition feature has been implemented to avoid asking questions that are inappropriate or redundant.

After the condition section is the number of menu options

available to answer the question with. Each option record consists of a line containing the menu option text, followed by a line containing a frequency count and a set of weights. The frequency count is the number of times throughout the history of the input file and its predecessors that this particular option has been chosen in response to this question. The count is set at one to begin with, so no possible options are discounted due to not being chosen before. At the end of each option record is a number indicating whether information should be propagated to the next program in the decision support system if the option is chosen. If the flag is positive, it indicates a spot in the output file (an intermediate file in the context of the whole system) where text should be placed. The text to use occurs on the line after the output index. If the index is 0 or -1, there is no output text line.

Blank lines are inserted between questions to maintain readability. Since the input file is rewritten by the program based only on the information saved when it was read in, comments are not supported.

Information Storage

The knowledge found in the input file is primarily stored in several struct arrays. Structs are variable types in C

where other types can be bound together, each with a name attached. For instance, a struct could contain a string called "city" containing the name of a city and an integer called "pop" containing the population of the city. Each design analysis program has its information stored in a struct containing the name of the program as a string and the minimum value for approval as a floating point number (float). Each question has a struct where it combines its identifying number, text, number of conditions, condition struct array, number of menu options (answers), and menu option struct arrays. Condition structs contain the label, menu option selection, and condition flag. Menu option structs have text, number of times option was selected in the past, weights to multiply with current package weights, the index to place output and the output string needed in this index location should the option be chosen.

Internal information consists of a record of which questions were asked so far in the session and what the menu option selections were. This is used primarily to implement the walkback feature which allows going back as far as one wants to correct an answer. It is also used to update the usage histories on the selected menu options. As a way to speed up condition checking and question selection a bit, a table is maintained with an entry for each question saying

whether it has been previously asked in the current session. To support walkback, a history of the weight values after each question is maintained in a two-dimensional array.

The information listed above is stored only where it is needed. There are no global variables. The input file variables are stored in the main top-level routine. It passes their pointers to the read routine to initialize them from the input file. It then passes only those that are necessary to the "ask" routine. For instance, the minimum values for the design package weights are not passed to "ask". This policy is followed throughout the layers. Each routine gets only the information it needs to accomplish its function. This is known as functional coupling. The advantage is if a variable format changes for some reason, it is easy to see what functions depend on that variable and may need to be changed. It is also an advantage during debugging. If some variable is getting bad values, it is not very hard to track down which modules could be causing this problem.

Structure of Inference Engine

As mentioned earlier, the GAPS inference engine was written in C and consisted of three major routines called from the main program module. The routines performed knowledge base input, interaction with the user, and updated knowledge

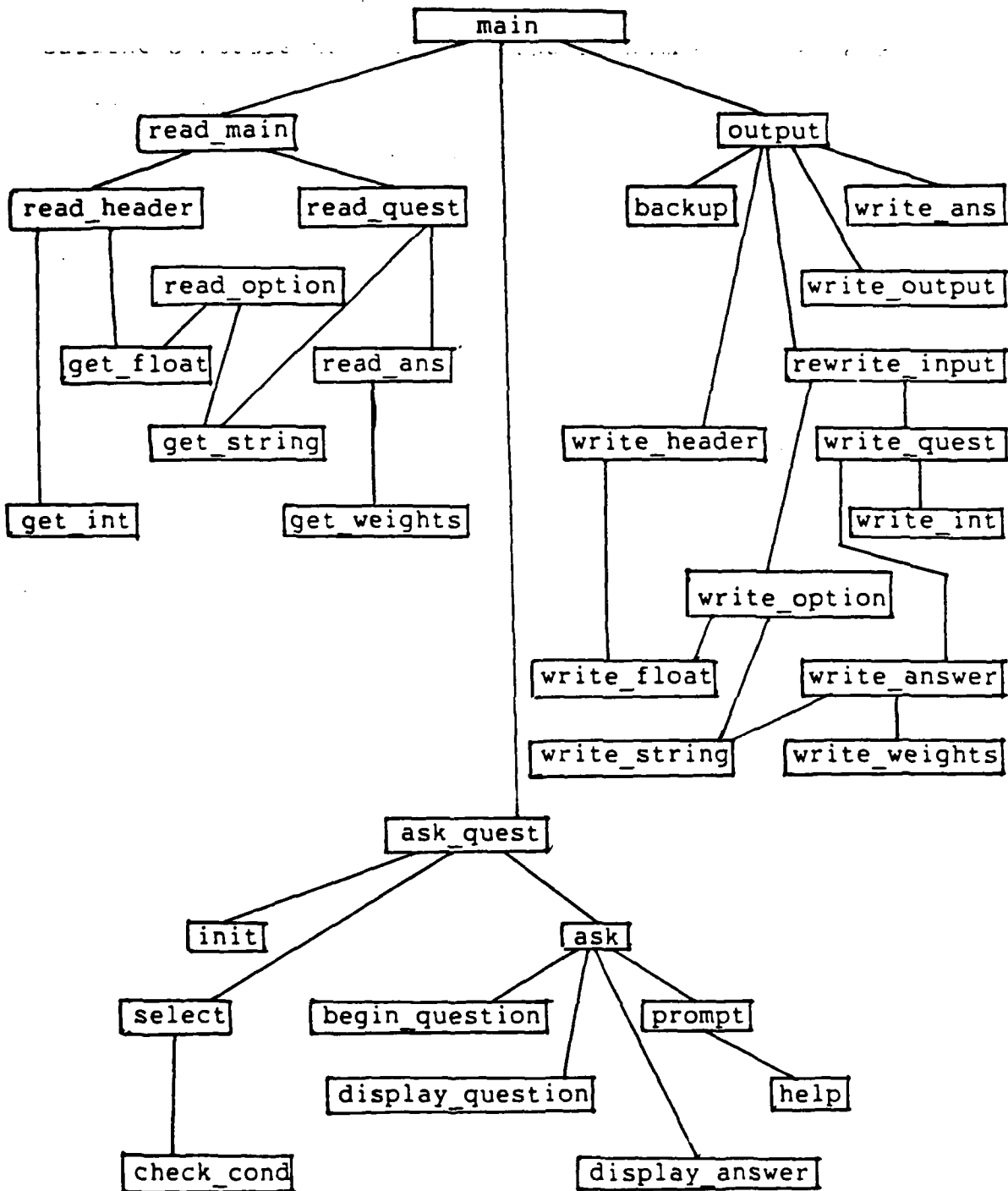
base output, along with screen output of the recommendations.

The input file reading routine carries out its work by calling a module to read the header (number of packages, number of questions, tolerance value), a module to read each analysis package record, and a module to read the question records. This last routine calls a subroutine to get records for each menu option.

The module that asks questions begins by initializing key history variables such as the weights table over time and the used questions list. It then goes through a loop in which it calls a selection subroutine and an asking routine. The selection routine has a subroutine to determine whether conditions have been met before it makes a choice based on informational value. The asking is divided into several parts to make interfacing with the BIOS routines easier. One routine clears the screen, another prints the question, one displays each menu option, and yet another prompts for the selection. The prompt routine has a subroutine for displaying usage information if the user needs help.

Output consists of backing up the input file, rewriting it, putting the acceptable analysis package names in a file as well as displaying them at the terminal, and putting any transferable data related to how a question was answered into the output file. Each of these four operations is done by a

Figure 10: GAPS Inference Engine Module Dependency Diagram



certain module. The backup routine does a straight character to character copy. Rewriting has the same structure as the input routines for the most part. There is a routine to rewrite the header, one to write the package choices, and another to write out question records. This routine uses a subroutine for the answer choices. At the bottom of the hierarchy of module are basic input/output routines that accomplish string moves in a line-oriented fashion, integer reads and writes, and floating point I/O. The output routine asks if assistance is needed for the recommended design analysis program. If yes, GAPS chains to the program determined by the first three letters of the design analysis program. PHOENICS' assistance package resides in PHO_FRONT.EXE.

The program text is organized so that a routine's subroutines always follow it in the listing. All of the modules are in the same source file in order to keep code from getting lost, although this tends to drive up compile time.

C Language Considerations

Writing the intelligent front end in C was a choice guided mostly by a desire to give it some portability. C is becoming somewhat of a standard among nonbusiness programmers and its compactness makes it easy to port from machine to machine. C

makes it easy to do what is efficient for the computer to do (such as using `i++` to increment `i`) and it makes it hard to program what is not efficient. For example, string moves and compares have to be done with functions. Therefore, the executable module runs quite fast, especially one generated by Microsoft's C Compiler.

The compiler itself is rather slow. This seems to be the major drawback. Inserting debugging code or fixing syntax errors adds overhead to the development time. Another problem with syntax errors was their tendency to defeat Microsoft's error recovery scheme. However, this problem is only in the Version 3.00 compiler. The new 4.00 version has much improved recovery from minor errors, such as missing semicolons, commas instead of semicolons, and unbalanced braces.

Using C instead of LISP in an AI context did not seem to have any drawbacks. The AI paradigm used is rather ancient and does not need special languages to be implemented well. In addition, it is strictly mathematical. Other techniques require more symbolic processing ability than C can muster, although as noted in Chapter II, rewriting LISP programs in C has become an acceptable practice in expert systems development.

Overall, C was a good language to develop GAPS in. Its property of being strongly typed aided in tracking down

intermodule problems. Its popularity and portability give the IFE a wide variety of target hosts. The long compile times were compensated for by the quick executable code.

AD-A195 054

INTEGRATING IMAGES APPLICATIONS AND COMMUNICATIONS
NETWORKS VOLUME 5(U) MASSACHUSETTS INST OF TECH
CAMBRIDGE A GUPTA ET AL. DEC 87 MIT-KBIISE-3

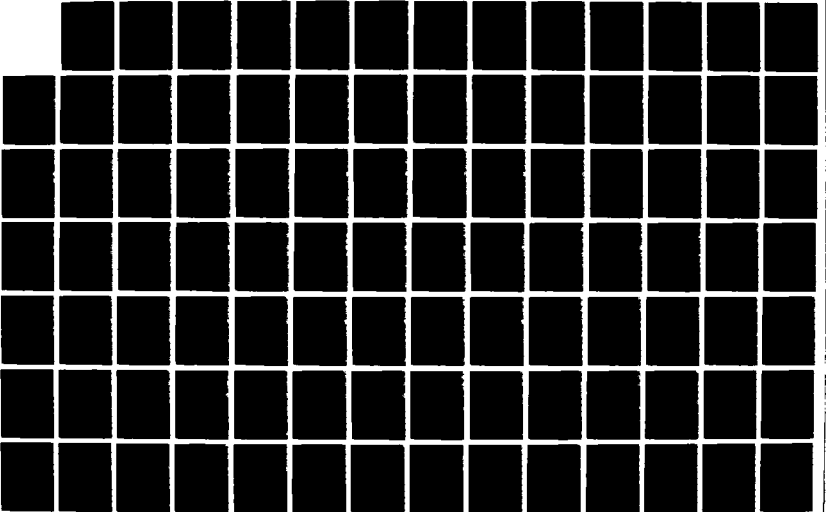
3/4

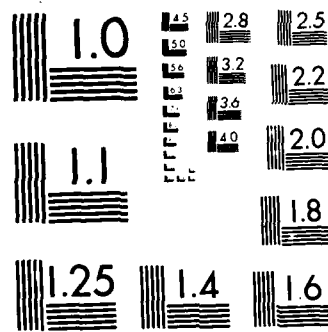
UNCLASSIFIED

DTR537-85-C-00003

F/G 5/2

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

VII. GAPS User Considerations

Instructions for Use

To use the General Analysis Package Selector, the computer or workstation should be on and the MS-DOS operating system or a compatible system should be loaded. Insert the disk containing GAPS into the floppy disk drive slot and select that drive as the default. For instance, if the floppy disk is drive A:, type A: after inserting the disk. Type GAPS to run the program using the input file "GAPS.INP". For a different input file, type GAPS <input file name>. A title page will appear on the screen while the input file is read into memory. After it is completely read, you will be given the option of typing function key 10 for an introductory page or the space bar to start the program. If F10 is selected, press the space bar after reading the introduction to start the program. To suppress the title page and F10 prompt, you may type GAPS /quiet or GAPS /quiet <input file name>.

GAPS will ask a series of questions assuming that you have a particular design in mind. "Don't know" is usually available as a response, so the features do not have to be too rigidly defined. However, the more you know about the design in question, the fewer questions will be needed to determine the appropriate package.

Each question will appear on the top line of the screen,

Figure 11: Sample GAPS Session

A:\>gaps

Welcome to GAPS. General Analysis Package Selector
Version 1.0 5/5/87

1. Does the analysis require three-dimensional modeling?

- a. Yes
- b. No
- c. Don't know

>b

2. Do you need interactive graphics to express your design?

- a. Yes
- b. No
- c. Don't know

>y

3. Does the design contain a fan?

- a. Yes
- b. No
- c. Don't know

>b

Questioning complete. Updating response frequencies...

CAEDS is recommended for the analysis with confidence 0.93.

Do you want assistance in using CAEDS?no

preceded by a number corresponding to how many questions have been previously asked. For instance, the fourth question asked is preceded by a four. Below the question are what should be all possible responses to it. The first response is highlighted to indicate that pressing the RETURN key will select it. Commands available at this point involve only single key presses. To move the highlighting from option to option, use the right and left arrow keys. Attempts to move past the last option using the right arrow will be ignored. Moving as far to the left as possible brings one to the "walkback" option, a feature that will be discussed shortly. Further pressing of the left arrow key will be ignored. To select an option, position the highlighting over it and press the RETURN key. You may also select the "yes" answer, if there is one, by pressing the "y" key. The "n" key works similarly, selecting the "no" answer. The highlighting is not moved to the selected answer in this case. GAPS moves on to the next question. To display a list of all possible responses, enter a "?". After reading the help screen, press the space bar to redisplay the question.

If a wrong response is entered at some point, you may go back and correct it using the walkback feature. Walkback mode can be selected by choosing the "<walkback>" option hidden to the left of the first response option when the question is

asked. It can also be selected by pressing the up-arrow key. The last question answered is displayed along with the highlighted response that was chosen. The other responses are not displayed. At this point, you may choose to have the question asked again by pressing the RETURN key. You may decide the response is acceptable and move back to the current question by pressing the down arrow key. If the mistake occurred on an earlier question, you may continue to move backwards using the up-arrow key. Pressing the down-arrow key will move you ahead a question without changing any answers. When you are at the question whose response you would like to change, press RETURN. The answers to any following questions are eliminated. For example, if you are about to answer question five and you decide to change the response to question two, you can press the up-arrow key three times and press RETURN. Question two is asked again and the previous answers to questions three and four are forgotten. While in walkback mode, you can return to the current question by pressing the "." key. You can also get a list of valid inputs by entering a "?".

At any time during the questioning, you may exit the program by pressing the ESC key. GAPS will ask if you want to quit. If yes, enter a "y" and press RETURN. Otherwise, enter an "n" and return to the question.

When GAPS has exhausted its supply of questions, or when two of the packages have been eliminated from consideration, it will print out which analysis package or packages are suitable for analyzing the design. It will then ask whether assistance is desired for each package. If the response is "yes", GAPS will "chain" to the assistance module for the package in question. It will not return to inquire about assistance for any other module.

Interpreting Output

The average user need only worry about the design analysis program(s) recommended and their associated certainty values. Certainty values can range from .30 to 1.00. A low certainty value (less than 0.75) can often be corrected by examining the design more carefully to change some "don't know" answers to decisive ones. More than one option will be displayed only when the second-place program has a certainty value within .10 of the winning program.

Interpreting the output file is more difficult, but only needs to be done in order to write a program that will be run after GAPS and to avoid most duplicate questions. The output file defaults to being GAPS.OUT, although a different output file can be specified as the second argument on the GAPS call. The first one or two lines contain the recommended CAD system

names followed by their suitability scores. The end of this section is signified by a line containing eight hyphens in a row. Output strings as given in the input file show up one per line. If a certain index has an undefined string because its question was not asked, three hyphens are placed on its line. The last defined string is followed by eight hyphens on the next line to signify the end of the file.

Modifying the Knowledge Base

The knowledge base can be modified in many different ways. One of the design goals for GAPS was to make this modification as straightforward as possible. Various modifications include adding questions, adding answers to existing questions, eliminating existing questions, and changing the weights associated with a certain answer. It is expected that the experts who know the most about the design analysis packages will want to write additional questions or reassess the old questions.

Before changing the knowledge base, one should preserve the current version by copying it. The filename "GAPS.BAK" should not be used as it is used by GAPS to back up the knowledge base while it is being updated. To edit the knowledge base, any standard text editor that produces a file capable of being "TYPED" can be used.

Whenever adding or deleting questions, do not forget to change the question count which is stored on the second line of the file. To delete a question, start eliminating lines at the label, an integer occurring on the line previous to the question text, and stop deleting lines just before the label of the next question. To add a question, move to the end of the file, or into the middle if you so desire, just before some question's label and enter the question record based on the file format given in Chapter VI. The label comes first. Make sure it is unique. Next is the question text. Quotes are definitely not necessary and the question should be no more than 77 characters long. The number of conditions goes on the next line, followed by that many condition records.

Next, the answer count is placed on its own line and followed by that many answer records. Each answer record contains answer text which should not exceed 11 characters followed by a line containing frequency data and weights. Frequency should begin at one for every new answer so that the actual response history will have a significant effect while the initial behavior is not dysfunctional. Weights should be between zero and one and need only have one decimal place of precision. Each corresponds to an option in the order listed at the top of the file. The first weight refers to PHOENICS, the second to CAEDS, and the third to ITAM. Each weight

should be interpreted as the figure to multiply the current certainty of its program by if the answer is chosen. For instance, the "No" answer for "Does the design contain a fan?" has a weight of zero for ITAM since lack of a fan will eliminate ITAM from the running. "Don't know" answers typically receive equal weights below one to indicate less certainty while not favoring any particular program. Do not ignore the complexity of the package when making up a weight. Being able to use ITAM instead of PHOENICS is a plus. If an answer keeps ITAM in the running, it should reduce the appropriateness of PHOENICS. Following the weights is a line containing the output file index and a line containing an output string. The index is typically -1 to indicate "empty" and the string is blank. Be sure to add one to the question count in the file header after adding a question.

VIII. Further Exploration

Using Inference Engine for Other Applications

The GAPS system is specifically used to select a design analysis package. However, this is largely a function of its knowledge base which contains the names of analysis packages as options and questions relating to design. With changes made to the input file, the system could be used to select between any set of software applications. One can imagine creating a knowledge base for choosing among desktop publishing options, or among word processors, or among expert systems shells. With the design analysis questions replaced, the system could be making software choices without any modification of the C source.

The generality of the GAPS inference engine extends beyond the original application of software choices to any recurring choice situations. For instance, it could be used to help a person decide whether to eat out or cook at home. Students could use it to decide which class to spend time on. A shopping decision system would use the inference engine to pick a grocery store or a shopping mall. An automobile dealer could have a system in the showroom to help buyers decide on a model.

Any nontrivial decision situation that occurs often enough

to make the initial knowledge base creation worthwhile might be a candidate. Choosing whether to take public transportation or a taxi seems too straightforward. Either you have to get to your destination in a hurry in which case a taxi is called for or time is not critical. The inference engine is inappropriate for the taxi choice for another reason. There is no time to consult a knowledge base no matter how few questions the system asks. The necessity of having the situation recur implies that a movie-selection system would be inappropriate. The knowledge base would have to change about once a week.

Also needed is some sort of expert advice that could be encapsulated in a knowledge base. For the eating out decision, a food critic and chef could be consulted. For choosing a car, an automotive critic would be able to identify the different criteria that consumers feel are important.

The GAPS inference engine could never become a marketable expert systems shell. The knowledge base format is not as intuitive as that of a rule-based system. The generality is not as broad as M.I. or KEE. It does not have the ability to explain its decision. However, it is interesting to note that a system that was designed from the very beginning to choose between the analysis packages PHOENICS, CAEDS, and ITAM, has the ability to make other choices simply by working with a

different input file.

Modifying the Inference Engine

Even though much can be accomplished without any changes to the C source code, there are cases, especially if the system is being ported to a non-MS-DOS machine, where modifications to the inference engine itself will have to be made. Modifications in the algorithm used may be desired, and this will obviously necessitate program changes.

Because the program is written in C, it is more portable than if it had been written in any other language. However, C is not perfect. Different compilers have varied extensions. Moving between MS-DOS C compilers may necessitate changes in some of the nonstandard string functions as well as the use of different #include files. The Microsoft subroutine for BIOS calls is "int86". In other compilers, it is undoubtedly different. Microsoft's compiler does not allow one to become sloppy with pointers. Therefore, it will not present the conversion problems that movement from a lax compiler to a strict one might.

There is no harm in attempting to compile the GAPS program without carefully examining it. Missing #include files will be noted, as will be missing subroutines during the link phase. Probably all that will need to be changed are the

allocation functions, possibly the "void" type, the BIOS calling procedure, and character-limited string comparison functions. The compilation batch file itself will obviously have to be ignored.

Porting the system to an operating system without BIOS, a category including every operating system except MS-DOS, may be just as simple. First of all, one must change the #define of the symbol PC to be a NO. This will cause parallel logic in the interactive routines to be used. Instead of controlling the cursor and responding to each key press immediately, standard printf and scanf functions are used. While the ease of use and elegance are decreased in this non-screen-driven mode, complete functionality is retained. After this change, the other necessary changes are as above, namely, the #include file names, and the allocation and string functions.

A few modifications to the algorithm used by the GAPS system can be done without changing the knowledge base or the input/output routines. For instance, question selection could be purely sequential simply by changing the "select" module to return a number one larger than the last number it returned. Selection could be frequency-independent by weighting each answer equally. The exit criteria could also be limited to exhausting the questions rather than decisively choosing a

package when it is the only feasible one left, by removing the test. Professor J.L. Schwartz has suggested this solution to the order-dependence problem, in which the order of the questions determines the answer. It would eliminate incorrect recommendations where a package was suggested where none should have been feasible.

An algorithmic change which would necessitate some knowledge base modification would be to use weights greater than or equal to one. This would change the decision criterion from least unsuitable to most suitable. Of course, some way of absolutely eliminating the option with a fatal flaw would have to exist. Perhaps zero weights could be retained.

Several changes to the #include file may have to be made over time. For instance, there are limitations on the number of questions, the number of answers for each question, and the length of answers. Changing any of these values will require recompilation, but the "gapsdef.h" #include file is well-commented enough to make the modification task very straightforward.

Implementing the Overall System

The overall design process as seen from the view of the GAPS system is a unified one. An engineer will design a

mechanical product to the point where it can be analyzed, but before it becomes needlessly dependent on a certain design analysis package, for instance, modelling it with the CAEDS solid modeller. The engineer should then run the GAPS program on a PC or workstation, find the appropriate package and use the assistance program for that package. The design analysis package should be run using the advice given and the results used to improve the design until it is successful in meeting specifications. An additional expert system over the analysis package itself is a definite possibility.

The questions asked by the GAPS intelligent front end should only be directed towards deciding which program to use. Additional program-dependent querying should occur in the assistance programs. While the interface between the IFE and the assistance system has been implemented through the GAPS.OUT data file, further propagation of user-supplied information is problematic. The reason is that while both the IFE and assistance system will be run on the same system (even sequentially in most cases), the design analysis program has a good chance of being based on a different host. Chapter IX examines this information transfer problem and other design issues involved in the implementation of one piece of the assistance system, PAM, the Phoenix Assistance Module.

IX. The PHOENICS Assistance Module

Design Considerations

Unlike the GAPS intelligent front-end, PAM has not been implemented and its design premises have not been subject to scrutiny. The somewhat speculative suggestions which follow are intended to aid the developer in formulating a final workable design and actually implementing this expert system.

The choice between building a C inference engine and using an expert systems shell on the market is not an easy one. Any off-the-shelf product should run on the IBM RT PC and hopefully on whatever other machines IBM design engineers are expected to use. Obviously, the choice of software vendor will have to be consistent with IBM policy.

Before choosing a development system, it will be useful to examine the issue of what sort of assistance needs to be given. No graphical support needs to be provided as PHOENICS input is purely alphanumeric. "Handholding" and "getting started" information should possibly be available as an option, but the engineer should be treated as a design specialist confronted with a complex analysis package containing a suboptimal user interface, features which he or she will never use, and a user manual that too often assumes a heavy grounding in the theory of fluid flow and heat transfer.

One of the more arcane topics which should be clarified by PAM is the theory of solving simultaneous equations iteratively. Expert input should be obtained on when to under-relax and by how much, how many iterations are necessary, and when a parabolic solution (one sweep) can be used. Model-building without graphics may be difficult for some. What sort of shape to make each grid cell, what size cells should go where, how idealized the model can be, and other questions such as which turbulence model should be used, should be addressed as intelligently as possible, using the advice of PHOENICS experts.

The interface to the rest of the system should be carefully thought out. The GAPS output format can be manipulated by modifying strings and indices in its input file. If other design analysis packages are listed in that file as being equally applicable, they might provide a way out if insurmountable problems arise. If PAM output is to a file, its format should be designed for use with the file transfer package between the PAM host (probably an RT) and the PHOENICS host. An alternative is for PAM's output to consist of a hard-copy list of usage suggestions referred to by the engineer while using PHOENICS. Remotely accessing PHOENICS directly from PAM seems too restrictive. The programs should be decoupled in time as they are in space. That is, the

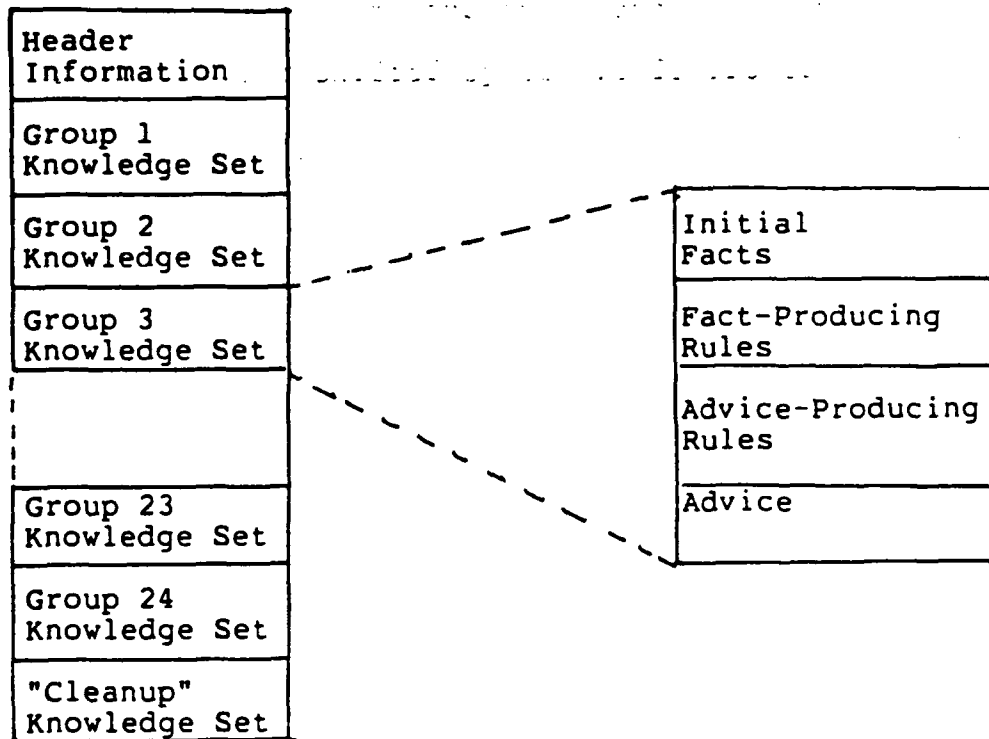
PHOENICS run should occur when its host is up and running under a reasonable load and not necessarily immediately after PAM is run. If a unified system is developed, where the entire design environment is on a single host, decoupling would not occur. Each piece of the system would chain to the next.

Where PAM will differ from GAPS and many other expert systems is that its answer will not consist of a single choice or even a few choices. Whereas GAPS picks a program and MYCIN picks a disease, PAM will produce sets of commands, lists of advice, and a general strategy for approaching PHOENICS. This precludes the GAPS inference engine necessarily and may eliminate other shells as well. One approach to stay within the expert systems paradigm is view the advice process as a series of decisions. The next section discusses a specific suggested design.

PAM Design

The purpose of this proposed design is to make a first pass at the problem of building the PHOENICS assistance system. Being instructive, it will serve to identify stumbling blocks and areas of ambiguity. For clarity, this is written as though PAM has been implemented. However, it should not be construed as a proven approach.

Figure 13: PAM Knowledge Base Layout



For portability and greater flexibility in design, PAM is written in C. Its output consists of a file written in PHOENICS input language suitable for initial input into the package. This will set up any parameters gleaned from the user discussion. In addition, PAM will put out a human-readable file, suitable for printing, explaining what its commands to PHOENICS are and what strategy to follow once command moves from the PAM-built file to the user.

PAM's knowledge base consists of sets of if-then rules loosely based on the 24 groups of commands available in GAPS. Rules consists of a set of preconditions, followed by a series of facts and advice. Facts about the design can be used as preconditions and are used by the input file generator, but advice is placed into the print file and not otherwise used. Therefore, advice can be unstructured and of a more human-readable form. Allowing multiple result facts differs from the MYCIN approach and indicates that a broad answer is needed and not a narrow solution. Facts can be designated as global in which case they would go into a blackboard abstraction and be available for later use.

A typical knowledge set could deal with the determination of a coordinate system and the type of grid within that system. It would know the conditions under which cylindrical coordinates were best and the reasons for making one dimension

more fine-grained than another. A mixed-mode method would be used where the system would alternate periodically between attempting to make forward progress and backtracking from hypothetical solutions, for instance, attempting to show that Cartesian coordinates were most appropriate. The questioning would end for the set when no more rules could be applied. A question limit or derived facts threshold could also be added to prevent the interaction from bogging down in an attempt to derive marginal facts. For better modularity, it would be best to write the PHOENICS input file commands at this point and then go on to the next set.

At the end of the program, a "cleanup" function has one last knowledge set that works with any rules on the "blackboard" and adds advice or commands to the output files. User questions are kept to a minimum, perhaps none if possible.

If possible, the output files are sent to the PHOENICS host and the local printer without user intervention. If requested, a set of instructions for getting into PHOENICS is displayed and/or sent to the printer.

Implementation Suggestions

From the experience of implementing GAPS, it has been found that programming in C on a personal computer is an

agreeable approach. It is helpful to have many different modules, each with a specific purpose, arranged logically in a set of source files. A library should be maintained for PAM so that changes in one file do not require compilation of the entire source. Modules should be tested independently and diagnostics should remain a hidden option as they are in GAPS.

Advice can be abstracted from the rules by having special rules to convert facts to advice only and making all other rules produce only facts. As far as data representation, rules and facts can be stored in structs while advice is kept in strings.

If possible, the cleanup function should act similarly to the other knowledge sets so that a different set of modules do not have to be maintained. For ease in editing, each knowledge set should have a different input file. This will also minimize the wait at the beginning of the program since only a fraction of the total knowledge need be read in at that point.

As noted earlier, these are only suggestions. The proposed design should be carefully examined and refined before implementation is started, especially in the area of knowledge representation. Expert advice should be solicited to resolve ambiguities.

X. Conclusion

Problem Summary

The problem faced by the IBM designers is that the analysis of their designs is a complicated process that only a few people have knowledge of that package initiates another questioning session that decides which commands and parameter values are appropriate and gives whatever additional assistance it can based on the user's desires.

GAPS Summary

The GAPS IFE was written in C and designed so that as much of the knowledge as possible was kept outside the program in a single input file. GAPS concerned itself with the user interface of asking questions and the AI paradigm of forward-chaining but has knowledge of that package initiates another questioning session that decides which commands and parameter values are appropriate and gives whatever additional assistance it can based on the user's desires.

GAPS Summary

The GAPS IFE was written in C and designed so that as much of the knowledge as possible was kept outside the program in a single input file. GAPS concerned itself with the user interface of asking questions and the AI paradigm of

forward-chaining by measuring and maintaining relative uncertainty through a weighting scheme. The scheme utilizes values from 0 to 1 for each of the design analysis programs and after each question is answered, updates the current weights through multiplication by a set of weights associated with the menu option just chosen.

Support for Approach

The expert system design consisting of a knowledge base and an inference engine is a common one. Using C as a development language allows portability and facilitates maintenance due to its growing popularity. The weighting system allows flexibility, represents uncertainty, and is fairly intuitive. The question selection procedure is designed to work towards an answer using the fewest questions, by maximizing the expected absolute change in total weight sums. The input file format is easy for the program to read and write without being too difficult for the user to understand. In addition, it is powerful enough to avoid redundant or inappropriate questions and determine the output file format.

Future Directions

It is expected that design analysis experts will examine

the behavior of GAPS and along with programmers and/or knowledge engineers, refine the system so that it gives better and better results. The implementation of PAM and the other assistance packages will proceed, hopefully using some of the advice in this discussion. Further educational use of the concepts presented in this paper could include finding a better way to represent CAD/CAM usage knowledge, implementing the system using an expert systems shell, or applying the weighting technique to another recurring choice problem.

Bibliography

- Begg, Vivienne. Developing Expert CAD Systems. London: Kogan Page, 1984.
- Bertoline, Gary R. Fundamentals of CAD. Albany: Delmar Publishers, Inc., 1985.
- Besant, C.B. and Lui, C.W.K. Computer-Aiding Design and Manufacture. Chichester: Ellis Horwood Ltd, 1986.
- Brody, Herb. "CAD Meets CAM," High Technology, 7 (5): 12-18 (May 1987)
- Davis, Dwight B. "Artificial Intelligence Enters the Mainstream," High Technology, 6 (7): 16-23 (July 1986)
- Encarnacao, Jose and Schlectendahl, Ernst G. Computer Aided Design. Berlin: Springer-Verlag, 1983.
- Ham, Michael. "Playing by the Rules," PC World, 2 (1): 34-41 (January 1984)
- Lee, Alex. "Engineering Design on Micros," Computers and Electronics, 22 (9): 86-90 (September 1984)
- Mandell, Patricia. "Computers That Humanize Health Care," Ms, 14 (11): 103-104 (May 1986)
- Newport, John Paul Jr. "How PC's Shook an Industry," Fortune, 112 (6): 105-106 (September 16, 1985)
- Schank, Roger. "Roger Schank on Expert Systems," Publisher's Weekly, 226 (12): 40-43 (September 21, 1984)
- Shurkin, Joel N. "Expert Systems: The Practical Face of Artificial Intelligence," Technology Review, 86 (6): 72-76 (November / December 1983)
- Stamps, David. "Expert Systems, Software Gets Smart -- But Can It Think?," Publisher's Weekly, 226 (12): 36-39 (September 21, 1984)
- Stark, John. What Every Engineer Should Know About Practical CAD/CAM Applications. New York: Marcel Dekker, Inc., 1986.

Teicholz, Eric and Smith, Dan. "Where Are We and Where
Are We Going on PC's?", Architectural Record, 173 (6):
47-49 (May 1985)

Wexler, Joanna. CAD84: Sixth International Conference
and Exhibition on Computers in Design Engineering.
Surrey: Butterworth & Co. Ltd, 1984.

Appendix A
GAPS Source Code Listing

Source file name: gaps.c

/* GAPS -- General Analysis Program Selector

by William Habeck

Initial Date: 3/12/87

Last Revised: 5/5/87

Purpose: To decide which analysis package should be used to evaluate a design. As of 5/5/87, knowledge base contained questions for PHOENICS, CAEDS, and ITAM. */

```

#define LINT_ARGS      /* do argument type-checking */
#include <stdio.h>     /* standard input/output */
#include <stdlib.h>    /* standard library */
#include <string.h>    /* string functions */
#include <process.h>   /* program chaining and exit functions */
#include <malloc.h>    /* memory allocation */
#include <dos.h>       /* interface to BIOS */
#include "gapsdef.h"  /* #defines for parameters and symbols for better
                      readability, such as YES and NO */
#include "scrn.h"     /* screen-specific #defines */
#include "scrn.c"     /* screen-handling functions */

int fprintf();
int fscanf();
int fgetc();
int strcmpi();
int isupper();
int islower();
int _filbuf();

struct answerstr { /* structure containing answer information */
    char *ans;          /* the answer string */
    int history;        /* how many times answer has been chosen */
    float weights[MAXOPT]; /* what weights to multiply current option
                           selection indices by */
    int out_index;     /* where in the output to put this answer */
    char *out_string; /* how to display answer in output file */
};

struct condstr { /* structure containing condition information */
    int label; /* question to which condition applies */
    int ansnum; /* answer to which condition applies */
    int flag; /* flag telling which kind of condition:
              ALLOWED = this question is only allowed if
                      answer <ansnum> was given to question <label>
              CANT_ASK = if answer <ansnum> was given to questio
                      <label>, then this question cannot be asked */
};

```

```

struct questionstr { /* structure containing question information */
    int label; /* unique identifier for question */
    char *quest; /* question string */
    int numcond; /* number of preconditions */
    struct condstr cond[MAXCOND]; /* conditions for asking question */
    int ansnum; /* number of answers */
    struct answerstr answer[MAXANS]; /* answers */
};

struct optstr { /* structure containing option information */
    char *opt; /* option string */
    float minval; /* minimum index value to select this option */
};

struct qa { /* structure for question and answer history */
    int index; /* index of question asked */
    int ans; /* answer chosen */
};

int debug = NO; /* debugging flag, turn on with /debug */
int quiet = NO; /* flag to suppress title page, turn on with /quiet or /q

#define DEBUG (*deb_func)
#define OUT debug_file() /* debug output */

int debug_flag=YES;
FILE *file_ptr;

/* purpose of debug_file function: to return a pointer to the file
collecting debugging information */

FILE *debug_file()
{
    char file_name[20];
    if (debug == NO) return(stdout); /* nothing printed anyway, so standard
output is as good as anything */

    if (debug_flag == YES) {
        fprintf(stdout,"File for debug output:");
        fscanf(stdin,"%s",file_name);
        /* strcpy(file_name,"a.b"); */
        if (file_name[0] == '.')
            file_ptr = stdout;
        else
            file_ptr = fopen(file_name,"w");
        debug_flag = NO;
        return(file_ptr);
    }
    else
        return(file_ptr);
}

```

}

```
/* purpose of nodebug function: used instead of fprintf when debugging
is off to suppress output */
```

```
int nodebug(outfile,string)
FILE *outfile;
char *string;
{
return(0);
}
```

```
int (*deb_func)(); /* allows debug function to be chosen at runtime */
```

```
/* purpose of hide_cursor routine: to hide the cursor if debugging is off *
```

```
void hide_cursor() {
if (PC == YES) {
if (debug == NO)
cursor(25,0,0);
else
cursor(23,0,0); }
}
```

```
/* *** program execution begins here *** */
```

```
void main(argc,argv)
int argc;
char *argv[];
{
struct questionstr question[MAXQUEST];
int num_quest; /* number of questions */
int num_opt; /* number of options */
struct optstr option[MAXOPT];
struct condstr cond;
struct answerstr ans;
float tolerance; /* how close to best option index one has to be */
struct qa qa_hist[MAXQUEST]; /* questions asked and answers given in
this session */

float final_weight[MAXOPT];
char infile_name[20];
char outfile_name[20];
int i,j;

FILE *input_file; /* file containing all the knowledge */
FILE *output_file; /* file to send to the next piece of the system */
FILE *fopen();

void title1(void);
void title2(void);
```

```

void read_main(FILE *,float *, int *, struct optstr[MAXOPT], int *,
  struct questionstr[MAXQUEST]);

void ask_quest(int,int,struct questionstr[MAXQUEST],
  struct qa[MAXQUEST],float[MAXOPT]);

void output(FILE *, FILE *, char[], float, int, struct optstr[MAXOPT],
  int, struct questionstr[MAXQUEST], struct qa[MAXQUEST],
  float[MAXOPT]);

/* use the arguments to change input/output files
defaults are gaps.inp, gaps.out.
first, look for /debug and /quiet control arguments */

strcpy(infile_name,"gaps.inp");
if (argc > 1)
  if (strcmp(argv[1],"/debug") == 0)
    debug = YES;
  else if (strcmp(argv[1],"/DEBUG") == 0)
    debug = YES;
  else if (strcmp(argv[1],"/quiet") == 0)
    quiet = YES;
  else if (strcmp(argv[1],"/QUIET") == 0)
    quiet = YES;
  else if (strcmp(argv[1],"/q") == 0)
    quiet = YES;
  else if (strcmp(argv[1],"/Q") == 0)
    quiet = YES;
  else
    strcpy(infile_name,argv[1]);

if (debug == NO)
  deb_func = nodebug;
else
  deb_func = fprintf;

if (argc > 2)
  if (quiet == YES) {
    strcpy(infile_name,argv[2]);
    if (argc > 3)
      strcpy(outfile_name, argv[3]);
    else
      strcpy(outfile_name, "gaps.out"); }
  else
    strcpy(outfile_name, argv[2]);

else
  strcpy(outfile_name, "gaps.out");

input_file = fopen(infile_name,"r");
output_file = fopen(outfile_name,"w");

DEBUG(OUT,"Input file = %s\n",infile_name);

```

```

    DEBUG(OUT,"Output file = %s\n",outfile_name);
/* read the input file */
    if (quiet == NO) titel();
    read_main(input_file,&tolerance,&num_opt,option,&num_quest,question);
    if (quiet == NO) titel2();
    DEBUG(OUT,"tolerance,num_opt,num_quest = %f,%d,%d\n",tolerance,
        num_opt,num_quest);
    if (num_opt>MAXOPT)
        num_opt = MAXOPT; /* should also have an error message here */
/* ask the questions */
    ask_quest(num_opt,num_quest,question,qa_hist,final_weight);
/* print out results */
    output(input_file,output_file,infile_name,tolerance,num_opt,option,
        num_quest, question,qa_hist,final_weight);
    return;
}

/* purpose of titel routine: to print out the title page up to
but not including the F10 prompt */
void titel() {
    if (PC == NO) {
        printf("\n\nWelcome to GAPS, General Analysis Package Selector\n");
        printf("Version 1.0 5/5/87\n"); }
    else {
        clrscr();
        setpage(0);

        linedis("GGGGGG",TITLER,TITLEC,TC,0);
        linedis("G",TITLER+1,TITLEC,TC,0);
        linedis("G",TITLER+2,TITLEC,TC,0);
        linedis("G",TITLER+3,TITLEC,TC,0);
        linedis("GGG",TITLER+3,TITLEC+3,TC,0);
        linedis("G",TITLER+4,TITLEC,TC,0);
        linedis("G",TITLER+4,TITLEC+5,TC,0);
        linedis("G",TITLER+5,TITLEC,TC,0);
        linedis("G",TITLER+5,TITLEC+5,TC,0);
        linedis("GGGGGG",TITLER+6,TITLEC,TC,0);

        linedis("AA",TITLER,TITLEC+10,TC,0);
        linedis("A",TITLER+1,TITLEC+9,TC,0);
        linedis("A",TITLER+1,TITLEC+12,TC,0);
        linedis("A",TITLER+2,TITLEC+8,TC,0);
        linedis("A",TITLER+2,TITLEC+13,TC,0);
        linedis("AAAAA",TITLER+3,TITLEC+8,TC,0);
        linedis("A",TITLER+4,TITLEC+8,TC,0);

```

```

linedis("A",TITLER+4,TITLEC+13,TC,0);
linedis("A",TITLER+5,TITLEC+8,TC,0);
linedis("A",TITLER+5,TITLEC+13,TC,0);
linedis("A",TITLER+6,TITLEC+8,TC,0);
linedis("A",TITLER+6,TITLEC+13,TC,0);

linedis("PPPPPP",TITLER,TITLEC+16,TC,0);
linedis("P",TITLER+1,TITLEC+16,TC,0);
linedis("P",TITLER+1,TITLEC+21,TC,0);
linedis("P",TITLER+2,TITLEC+16,TC,0);
linedis("P",TITLER+2,TITLEC+21,TC,0);
linedis("PPPPPP",TITLER+3,TITLEC+16,TC,0);
linedis("P",TITLER+4,TITLEC+16,TC,0);
linedis("P",TITLER+5,TITLEC+16,TC,0);
linedis("P",TITLER+6,TITLEC+16,TC,0);

linedis("SSSSS",TITLER,TITLEC+25,TC,0);
linedis("S",TITLER+1,TITLEC+24,TC,0);
linedis("S",TITLER+2,TITLEC+24,TC,0);
linedis("SSSSS",TITLER+3,TITLEC+25,TC,0);
linedis("S",TITLER+4,TITLEC+29,TC,0);
linedis("S",TITLER+5,TITLEC+29,TC,0);
linedis("SSSSS",TITLER+6,TITLEC+24,TC,0);

linedis("General",TITLER+9,TITLEC-1,HIGH,0);
linedis("Analysis",TITLER+9,TITLEC+7,HIGH,0);
linedis("Program",TITLER+9,TITLEC+16,HIGH,0);
linedis("Selector",TITLER+9,TITLEC+24,HIGH,0);

linedis("Version 1.0    5/5/87",TITLER+11,TITLEC+5,NORM,0);
hide_cursor();
return;
}

/* purpose of title2 routine: to print out the rest of the title page
and the introductory page if requested */
void title2() {

char ecode;
char ink;

if (PC == YES) {
linedis("Press",TITLER+16,TITLEC-10,HIGH,0);
linedis("SPACEBAR",TITLER+16,TITLEC-4,REVNORM,0);
linedis("to Begin or",TITLER+16,TITLEC+5,HIGH,0);
linedis("F10",TITLER+16,TITLEC+17,REVNORM,0);
linedis("for an Introduction",TITLER+16,TITLEC+21,HIGH,0);
while (1) {
hide_cursor();
ink = inkey(&ecode);
if (ink == ' ') return;
DEBUG(OUT,"in title2, ink was %d, ecode was %d\n",ink,ecode);
if ((ink == 0) && (ecode == 68)) {

```

```

clearscr();
linedis("GAPS is an interactive decision aid for design analysis packages."
TITLER,10,NORM,0);
linedis("GAPS is an interactive decision aid for design analysis packages."
linedis("By asking you a series of questions, it will determine which of th
linedis("general analysis packages is best for your application.",
TITLER+2,6,NORM,0);
linedis("The questions will appear on the top line. Please select the",
TITLER+3,10,NORM,0);
linedis("best answer from among the options given by using the arrow keys",
TITLER+4,6,NORM,0);
linedis("to move to it and then pressing RETURN. Changing a previous answe
TITLER+5,6,NORM,0);
linedis("can be done by using the up-arrow key to walk back.",
TITLER+6,6,NORM,0);
linedis("A list of valid options will be displayed whenever you type a ?",
TITLER+7,10,NORM,0);
linedis("Please press SPACEBAR to begin.",TITLER+9,24,REVNORM,0); }
}
return;
}

```

```

/* purpose of read_main routine: to read the knowledge base completely into
memory */

```

```

void read_main(input_file,tolerance,num_opt,option,num_quest,question)

```

```

FILE *input_file;
float *tolerance;
int *num_opt;
struct optstr option[MAXOPT];
int *num_quest;
struct questionstr question[MAXQUEST];
{
void read_header(FILE *,int *,int *,float *);
struct optstr read_option(FILE *);
struct questionstr read_quest(FILE *,int);
int i;

DEBUG(OUT,"calling read_header");
read_header(input_file,num_opt,num_quest,tolerance);
DEBUG(OUT,"back from read_header,num_opt=%d,num_quest=%d,tolerance=%f\n",
*num_opt,*num_quest,*tolerance);

/* read the package options */

for (i=0;i<*num_opt;i++) {
option[i] = read_option(input_file);
DEBUG(OUT,"option[%d] = %s,%f\n",i,option[i].opt,option[i].minval);}

DEBUG(OUT,"done reading options/");
}

```

```

/* read the questions and answers */
for (i=0;i<*num_quest;i++) {
    DEBUG(OUT," read_quest i = %d /",i);
    question[i] = read_quest(input_file,*num_opt); }

    DEBUG(OUT,"done reading questions\n");

    return;
}

/* purpose of read_header routine: to read the number of options and
questions and the tolerance level */

void read_header(input_file,num_opt,num_quest,tolerance)
FILE *input_file;
int *num_opt;
int *num_quest;
float *tolerance;
{
    int get_int(FILE *);
    float get_float(FILE *);
    *num_opt = get_int(input_file);
    if (*num_opt>MAXOPT)
        *num_opt = MAXOPT; /* should also have an error message here */
    *num_quest = get_int(input_file);
    if (*num_quest>MAXQUEST)
        *num_quest = MAXQUEST;
    *tolerance = get_float(input_file);
    return;
}

/* purpose of the read_option function: to read information on one option */
struct optstr read_option(input_file)
FILE *input_file;
{
    struct optstr opt;
    float get_float(FILE *);
    char *get_string(FILE *);

    opt.opt = get_string(input_file);
    DEBUG(OUT,"opt.opt=%s\n",opt.opt);
    opt.minval = get_float(input_file);
    return(opt);
}

/* purpose of read_quest function: to read information on a single question
struct questionstr read_quest(input_file,num_opt)
FILE *input_file;

```

```

int num_opt;
{
    int i;
    int ans;
    struct questionstr quest;
    struct answerstr read_ans(FILE *,int);
    int get_int(FILE *);
    char *get_string(FILE *);

    quest.label = get_int(input_file);
    quest.quest = get_string(input_file);
    quest.numcond = get_int(input_file);
    DEBUG(OUT,"label,quest,numcond = %d,%s,%d/",quest.label,quest.quest,
        quest.numcond);

    for (i=0;i<quest.numcond;i++) {
        quest.cond[i].label = get_int(input_file);
        quest.cond[i].ansnum = get_int(input_file);
        quest.cond[i].flag = (get_int(input_file) == 0) ? CANT_ASK : ALLOWED;
        DEBUG(OUT,"i=%d,cond[i].label,ansnum,flag=%d,%d,%d\n",i,quest.cond[i].
            label,quest.cond[i].ansnum,quest.cond[i].flag);
    }
    quest.ansnum = get_int(input_file);

    /* read the answers */

    for (i=0;i<quest.ansnum;i++) {
        DEBUG(OUT,"read_ans i=%d/",i);
        quest.answer[i] = read_ans(input_file,num_opt); }

    return(quest);
}

/* purpose of read_ans function: to read information on a single answer */
struct answerstr read_ans(input_file,num_opt)
FILE *input_file;
int num_opt;
{
    struct answerstr ans;
    int freq;
    char *get_string(FILE *);
    void get_weights(FILE *,int,int *,float[MAXOPT]);
    int get_int(FILE *);

    ans.ans = get_string(input_file);
    get_weights(input_file,num_opt,&freq,ans.weights);
    ans.history = freq;
    DEBUG(OUT,"freq=%d ",freq);
    DEBUG(OUT,"ans.weights=%f %f %f\n",ans.weights[0],
        ans.weights[1],ans.weights[2]);
    ans.out_index = get_int(input_file);
}

```

```

if (ans.out_index > 0)
    ans.out_string = get_string(input_file);
else {
    strcpy(ans.out_string, " ");
    ans.out_index = EMPTY;
}
return(ans);
}

/* purpose of get_weights routine: to read the frequency and the weights fo
an answer */

void get_weights(input_file,num_opt,freq,weights)
FILE *input_file;
int num_opt;
int *freq;
float weights[MAXOPT];
{
    int i;

    *freq = get_int(input_file);
    for (i=0;i<num_opt;i++)
        weights[i] = get_float(input_file);
    return;
}

/* purpose of get_int function: to read an integer value from the knowledge
base */
int get_int(input_file)
FILE *input_file;
{
    long position;
    int int_val;
    fscanf(input_file,"%d",&int_val);
    DEBUG(OUT,"coming out of get_int with %d\n",int_val);
    return(int_val);
}

/* purpose of get_float function: to read a floating-point number from
the knowledge base */
float get_float(input_file)
FILE *input_file;
{
    float float_val;
    fscanf(input_file,"%f",&float_val);
    return(float_val);
}

/* purpose of get_string function: to read a string from the knowledge
base */

```

```

char *get_string(input_file)
FILE *input_file;
{
    int c;
    int i,j;
    char string[200];
    char *cptr;

    /* read until not blank, tab, or newline */

    while (1) {
        c = fgetc(input_file) & 0177;
        if ((c != ' ') && (c != '\n') && (c != '\t')) break; }

    string[0] = (char)c;
    i = 1;

    /* read until newline */

    while (1) {
        c = fgetc(input_file);
        if (c == EOF) break;
        c = c & 0177;
        if (c == '\n') break;
        string[i] = (char)c;
        i++; }
    for (i--;string[i] == ' ';i--); /* eliminate trailing blanks */
    i++;
    string[i] = '\0';
    cptr = malloc((unsigned int)(i+1));
    strcpy(cptr,string);
    return(cptr);
}

/* purpose of ask_quest routine: direct the process of asking the engineer
questions */

void ask_quest(num_opt,num_quest,question,qa_hist,final_weight)

int num_opt;
int num_quest;
struct questionstr question[MAXQUEST];
struct qa qa_hist[MAXQUEST];
float final_weight[MAXOPT];
{
    int used_table[MAXQUEST]; /* keeps track of which questions have been asked
float weight_hist[MAXQUEST][MAXOPT]; /* history of what option weights were
int quest_no; /* which question to ask */
int ans_num;
int next;
int last;
int i;

```

```

void init(int,int,int[MAXQUEST],float[MAXQUEST][MAXOPT],struct qa[MAXQUEST]
int select(int,int,struct questionstr[MAXQUEST],int,float[MAXQUEST][MAXOPT]
    int[MAXQUEST],struct qa[MAXQUEST]);
int ask(int,struct questionstr[MAXQUEST],int,struct qa[MAXQUEST],int *);
void update(int,int,int,struct questionstr *,int,int[MAXQUEST],
    float[MAXQUEST][MAXOPT],struct qa[MAXQUEST]);

/* initialize the state arrays */
DEBUG(OUT,"initializing the state arrays\n");
init(num_opt,num_quest,used_table,weight_hist,qa_hist);

/* pick the first question */

DEBUG(OUT,"selecting the first question\n");
quest_no = select(num_opt,num_quest,question,0,weight_hist,used_table,
    qa_hist);

next = 0;
while (quest_no != END) {
    last = next;
    DEBUG(OUT,"asking a question\n");
    next = ask(last,question,quest_no,qa_hist,&ans_num); /* ask question */
    DEBUG(OUT,"back from ask, last=%d,next=%d\n",last,next);
    if (next == last+1) {
        DEBUG(OUT,"updating info\n");

        /* update history data, select another question */
        update(next,num_opt,quest_no,&(question[quest_no]),ans_num,
            used_table,weight_hist,qa_hist);
        quest_no = select(num_opt,num_quest,question,next,weight_hist,
            used_table,qa_hist);
        DEBUG(OUT,"back from select, quest_no = %d\n",quest_no);
    }
    else {
        DEBUG(OUT,"next=%d\n",next);
        for (i=next;i<=last;i++)
            used_table[qa_hist[i].index] = NOT_ASKED;
        quest_no = qa_hist[next].index;
    }
}

/* decision has been made */

for (i=0;i<num_opt;i++)
    final_weight[i] = weight_hist[next][i];

return;
}

/* purpose of init routine: to initialize the historical arrays */
void init(num_opt,num_quest,used_table,weight_hist,qa_hist)

```

```

int num_opt;
int num_quest;
int used_table[MAXQUEST];
float weight_hist[MAXQUEST][MAXOPT];
struct qa qa_hist[MAXQUEST];
{
  int i,j;
  for (i=0; i<num_quest; i++) {
    DEBUG(OUT,"init pass %d\n",i);
    used_table[i] = NOT_ASKED;
    qa_hist[i].index = EMPTY;
    qa_hist[i].ans = EMPTY;
    for (j=0; j<num_opt; j++)
      weight_hist[i][j] = (i==0) ? 1.00 : 0.00 ; /* put 1's to start, else
  }
}

/* purpose of select function: to choose the best question to ask */

int select(num_opt,num_quest,question,time,weight_hist,used_table,qa_hist)
int num_opt;
int num_quest;
struct questionstr question[MAXQUEST];
int time; /* index into the weight_hist, which question we're on */
float weight_hist[MAXQUEST][MAXOPT];
int used_table[MAXQUEST];
struct qa qa_hist[MAXQUEST];
{
  int non_zero; /* used to count number of non-zero weights */
  float diff[MAXOPT]; /* temporary array to accumulate differences */
  float sum_diff; /* um of expected difference in weights */
  float best_diff; /* best expected difference in weight */
  int quest_no; /* number of question with best difference */
  int ok; /* YES if preconditions are met */
  int asked; /* number of times this question has ever been asked */
  float asked_real; /* floating point version of asked */
  struct answerstr answer; /* answer to a question */
  int i,j,k; /* counters */
  int check_cond(int,struct questionstr[MAXQUEST],int,int[MAXQUEST],
                struct qa[MAXQUEST]);

  non_zero = 0;
  DEBUG(OUT,"into select,time=%d\n",time);

  /* check to see if only one package is left */

  for (i=0; i<num_opt; i++)
    if (weight_hist[time][i]>0.01) non_zero++;
  DEBUG(OUT,"non-zero is %d\n",non_zero);
  if (non_zero<2) return(END); /* only one package left, stop questions */
  quest_no = END; /* default */
  best_diff = 0.00;
  for (i=0; i<num_quest; i++)

```

```

if (used_table[i]==NOT_ASKED) {
  DEBUG(OUT,"question %d not asked, checking conditions\n",i);
  ok = check_cond(i,question,num_quest,used_table,qa_hist);
  if (ok==YES) {
    DEBUG(OUT,"made it OK past cond check\n");
    for (j=0;j<num_opt;j++)
      diff[j] = 0.0;

    /* calculate expected weight change */

    asked = 0;
    for (j=0;j<question[i].ansnum;j++)
      asked += question[i].answer[j].history;
    asked_real = asked * 1.0;
    for (j=0;j<question[i].ansnum;j++) {
      answer = question[i].answer[j];
      DEBUG(OUT,"num_opt=%d\n",num_opt);
      for (k=0;k<num_opt;k++)
        diff[k] = diff[k] + (answer.history/asked_real) *
          (1 - answer.weights[k]) * weight_hist[time][k];
    }
    sum_diff = 0;
    for (k=0;k<num_opt;sum_diff += diff[k++]);
    DEBUG(OUT,"i=%d,sum_diff=%f,best_diff=%f\n",i,sum_diff,best_diff)

    /* update highest expected change if necessary */

    if (sum_diff > best_diff) {
      best_diff = sum_diff;
      quest_no = i;
    }
  } /* end of difference checking */
} /* end of checking this question */
return(quest_no);
}

```

/* purpose of check_cond function: to check on the conditions for asking a question -- have they been met? */

```

int check_cond(i,question,num_quest,used_table,qa_hist)
int i;
struct questionstr question[MAXQUEST];
int num_quest;
int used_table[MAXQUEST];
struct qa qa_hist[MAXQUEST];
{
  int ok; /* YES if conditions have been met */
  int label; /* label of the question */
  int index; /* index to get at question */
  int answer_given; /* when question was asked, what the answer was */
  struct condstr cond; /* precondition on a question */
  int j,k; /* counters */
  int found; /* found flag for array searches */

```

```

ok = YES;
for (j=0;(j<question[i].numcond)&&(ok==YES);j++) {
    cond = question[i].cond[j];
    label = cond.label;

/* find which question has this label */
    found = NO;
    for (k=0;(k<num_quest)&&(found==NO);k++)
        if (question[k].label==label) {
            found = YES;
            index = k;
        }
    DEBUG(OUT,"found=%d,index=%d,cond.flag=%d\n",found,index,cond.flag);
    if (found==YES) {
        if (used_table[index]==NOT_ASKED)
            if (cond.flag==CANT_ASK)
                ok = YES;
            else
                ok = NO;
        else { /* it was asked, so check the answer */
            found = NO;
            for (k=0;k<num_quest;k++)
                if (qa_hist[k].index == index) {
                    found = YES;
                    answer_given = qa_hist[k].ans;
                }
            if (found==YES)
                if (cond.flag==ALLOWED) { /* quest. allowed if right ans. */
                    DEBUG(OUT,"allowed -- answer_given=%d,cond.ansnum=%d",
                        answer_given,cond.ansnum);
                    if (cond.ansnum==answer_given)
                        ok = YES;
                    else
                        ok = NO; }
                else { /* question disallowed on answer match */
                    if (cond.ansnum==answer_given)
                        ok = NO;
                    else
                        ok = YES; }
            } /* end of the else 15 lines up */
        } /* end of if FOUND==YES 20 lines up */
    } /* end of checking condition j */
    return(ok);
}

/* bring in second half of file (split due to editor limitations) */
#include "gaps2.c"

```

Source file name: gaps2.c

```

/* purpose of ask function: asks the user the chosen question */

int ask(last,question,quest_no,qa_hist,answer_given)
int last;
struct questionstr question[MAXQUEST];
int quest_no;
struct qa qa_hist[MAXQUEST];
int *answer_given;
{
    struct questionstr quest;
    int pointer;
    int i;
    int exit_flag;
    int walkback;
    int status;
    int enhance;

    void begin_question(int);
    void display_question(char[]);
    void display_answer(int,char[],int,int);
    int prompt(struct questionstr,int *,int *,int);
    void err_mess(char *);
    int quit_yn(void);
    void cursor(int,int,int);
    void dump_qa(struct qa[]);

    dump_qa(qa_hist);
    quest = question[quest_no];
    DEBUG(OUT,"quest_no=%d,quest.quest=%s\n",quest_no,quest.quest);
    pointer = last;
    exit_flag = NO;
    qa_hist[pointer].index = quest_no;

    while (exit_flag == NO) {

        /* prepare screen for next question */
        begin_question(pointer);

        /* send quest.quest to the screen */
        display_question(quest.quest);

        if (pointer == last) {
            walkback = NO; }
        else
            walkback = YES;

        for (i=0;i<quest.ansnum;i++) {

```

```

/* send quest.answer[i].ans to the screen */
if ((walkback == YES) && (qa_hist[pointer].ans == i))
    enhance = YES;
else
    enhance = NO;
display_answer(i,quest.answer[i].ans,enhance,walkback);
}
hide_cursor();

```

```

/* prompt for input */

```

```

status = prompt(quest,&walkback,&pointer,last);
DEBUG(OUT,"status=%d\n",status);
if (status == ASK_AGAIN) {
    DEBUG(OUT,"last=%d,",last);
    DEBUG(OUT,"ptr=%d,qa_hist=%d\n",pointer,qa_hist[pointer].index);
    quest = question[qa_hist[pointer].index];
}
else if (status == EXIT_ASK)
    exit_flag = YES;
else if (status == ERROR)
    err_mess("Illegal input character.");
else if (status == QUIT) {
    if (quit_yn() == 'y') exit(0); }
else if (status == Y) {
    for (i=0;i<quest.ansnum;i++)
        if (strncmp("Yes",quest.answer[i].ans,3) == 0)
            if (status == Y)
                status = i;
            else
                err_mess("Ambiguous\n");
    if (status == Y)
        err_mess("No \"yes\" answers given.");
    else {
        *answer_given = status;
        pointer++;
        exit_flag = YES; }
}
else if (status == N) {
    for (i=0;i<quest.ansnum;i++)
        if (strncmp("No",quest.answer[i].ans,2) == 0)
            if (status == N)
                status = i;
            else
                err_mess("Ambiguous\n");
    if (status == N)
        err_mess("No \"no\" answers given.");
    else {
        *answer_given = status;
        pointer++;
        exit_flag = YES; }
}
else if ((status >= 0) && (status < quest.ansnum)) {

```

```

        *answer_given = status;
        pointer++;
        exit_flag = YES;
    }
    else
        err_mess("Illegal input character");
} /* close the while */
return(pointer);
}

/* purpose of begin_question routine: to print the question number */
void begin_question(quest_no)
int quest_no;
{
    int printf();
    char buff[5];
    if (PC == NO)
        printf("\n\n\n\n\n %2d. ",quest_no+1);
    else {
        /* clear the screen, put quest_no in the right place */;
        clrscr();
        sprintf(buff,"%2d.",quest_no+1);
        linedis(buff,0,0,NORM,0); }
    return;
}

/* purpose of display_question routine: to send the question to the screen
void display_question(quest)
char quest[];
{
    if (PC == NO)
        printf("%s \n\n",quest);
    else
        /* send cursor to upper left, print quest */
        linedis(quest,0,3,HIGH,0);
    return;
}

/* purpose of display_answer routine: to send one answer to the screen */
void display_answer(ans_num,answer,enhance,walkback)
int ans_num;
char answer[];
int enhance;
int walkback;
{
    char c;
    if (PC == NO) {

```

```

c = 'a' + ans_num;
if (enhance == NO)
    printf(" %c. %s\n",c,answer);
else
    printf(" * %c. %s *** \n",c,answer);
return;
}
else
    /* calculate spot in 2nd row to display answer, display it */ ;
    if (enhance == NO) {
        if (walkback == NO)
            if (ans_num == 0)
                linedis(answer,1,3,REVNORM,0);
            else
                linedis(answer,1,ans_num*12+3,NORM,0); }
        else
            linedis(answer,1,ans_num*12+3,REVNORM,0);
return;
}

/* purpose of prompt function: to solicit a response to the question
and return it */

int prompt(quest,walkback,pointer,last)
struct questionstr quest;
int *walkback;
int *pointer;
int last;
{
    int ans_num;
    void help(int,int,int,int);
    char c;
    int i;
    int ans_ptr,scr_ans_ptr;
    char str_in[10];
    int fgetchar(void);
    char ecode;
    char str[MAXANSLEN];

    ans_num = quest.ansnum;

    if (PC == NO) {
        if (*walkback == YES)
            printf("<=>"); /* different prompt symbol in walkback */
        else
            printf(">");

        fscanf(stdin,"%s10",str_in);
        c = str_in[0];
        switch(c) {
            case '<': /* walkback */

```

```

DEBUG(OUT, "*pointer=%d\n", *pointer);
if (--(*pointer) < 0) *pointer = 0;
DEBUG(OUT, "*pointer=%d\n", *pointer);
return(ASK_AGAIN);
case '>': /* walkforward */
if (*walkback == YES) {
(*pointer)++;
return(ASK_AGAIN); }
else
return(ERROR);
case '=': /* change this answer */
if (*walkback == YES) {
*walkback = NO;
return(EXIT_ASK); }
else
return(ERROR);
case '.': /* return to current question */
*pointer = last;
return(ASK_AGAIN);
case 'Y': /* pick the yes answer */
case 'y':
if (*walkback == YES)
return(ERROR);
else
return(Y);
case 'N': /* pick the no answer */
case 'n':
if (*walkback == YES)
return(ERROR);
else
return(N);
case '?': /* get help information */
help(*walkback, *pointer, last, ans_num);
return(ASK_AGAIN);
case 'h':
case 'H':
if (ans_num > 7)
return(8);
else {
help(*walkback, *pointer, last, ans_num);
return(ASK_AGAIN); }
case 'q': /* quit the program */
case 'Q':
return(QUIT);
default: /* hopefully a letter corresponding to an answer */
i = (int)c;
if (i > 64 && i < 91)
return(i - 65);
else if (i > 96 && i < 123)
return(i - 97);
else
return(ERROR);
}
}

```

```

else {
    ans_ptr = 0;
    scr_ans_ptr = 0;
    while (1) {
        hide_cursor();
        c = inkey(&ecode);
        DEBUG(OUT, "c=%d,ecode=%d\n", c, ecode);
        if (c == 0) {
            if (ecode == 72) { /* up-arrow means walkback */
                if (--(*pointer) < 0) *pointer = 0;
                return(ASK_AGAIN); }
            else if (ecode == 75 && (*walkback == NO)) {
/* back-arrow to get to previous option */
                if (--ans_ptr < -1) ans_ptr = -1;
                else if (ans_ptr < 0 && scr_ans_ptr > 0) {
                    blank(24,1,3,0);
                    linedis("<walkback>",1,3,REVNORM,0);
                    linedis((quest.answer)[0].ans,1,15,NORM,0);
                    scr_ans_ptr = 0; }
                else if (ans_ptr < 0 && scr_ans_ptr <= 0) {
                    blank(12,1,3,0);
                    linedis("<walkback>",1,3,REVNORM,0);
                    for (i=0;i<ans_num;i++) {
                        blank(12,1,15+12*i,0);
                        linedis((quest.answer)[i].ans,1,15+12*i,NORM,0); }
                    scr_ans_ptr = 0; }
                else {
                    if (scr_ans_ptr == 0)
                        i = 0; /* copout */
                    else {
                        blank(24,1,scr_ans_ptr*12-9,0);
                        i = ans_ptr;
                        linedis((quest.answer)[i].ans,1,scr_ans_ptr*12-9,REVNORM,0);
                        i++;
                        linedis((quest.answer)[i].ans,1,scr_ans_ptr*12+3,NORM,0);
                        scr_ans_ptr--; }
                }
            }
        else if (ecode == 77 && (*walkback == NO))
/* forward arrow means go right one option */
            if (++ans_ptr >= ans_num)
                ans_ptr = ans_num - 1;
            else {
                i = ans_ptr;
                blank(24,1,scr_ans_ptr*12+3,0);
                if (i == 0) strcpy(str,"<walkback>");
                else strcpy(str,(quest.answer)[i-1].ans);
                linedis(str,1,scr_ans_ptr*12+3,NORM,0);
                linedis(quest.answer[i].ans,1,scr_ans_ptr*12+15,REVNORM,0);
                scr_ans_ptr++; }
        else if (ecode == 80 && (*walkback == YES)) {
/* down-arrow means walkforward */
            (*pointer)++;
            return(ASK_AGAIN); }
    }
}

```

```

    else
        return(ERROR);
    }
else if (c == 27 || c == 'q' || c == 'Q') return(QUIT);
/* ESC means quit */
else if (c == '?' || c == 'H' || c == 'h') { /* get help */
    help(*walkback,*pointer,last,ans_num);
    return(ASK_AGAIN); }
else if (c == 13 && *walkback == NO)
/* RETURN means select the highlighted answer */
if (ans_ptr == -1) {
    if (--(*pointer) < 0) *pointer = 0;
    return(ASK_AGAIN); }
else
    return(ans_ptr);
else if ((c == 'Y' || c == 'y') && *walkback == NO) /* pick yes answer
    return(Y);
else if ((c == 'N' || c == 'n') && *walkback == NO) /* pick no answer
    return(N);
else if (c == '.' && *walkback == YES) { /* return to current quest
    *pointer = last;
    return(ASK_AGAIN); }
else if ((c == '=' || c == 13) && *walkback == YES) {
/* equals sign or RETURN means change the answer in walkback mode *
    *walkback = NO;
    return(EXIT_ASK); }
else
    return(ERROR);
}
}
}

```

```

/* purpose of help routine: to display information about command
possibilities and their meanings */

```

```

void help(walkback,pointer,last,ans_num)
int walkback;
int pointer;
int last;
int ans_num;
{
    char *work80;
    int row;
    char c,ecode;

    if (PC == NO)
        if (walkback == YES) {
            printf("\n\n Valid responses to the <=> prompt are: \n\n");
            printf(" ? to receive this help screen\n");
            if (pointer > 0)
                printf(" < to back up to question %d\n",pointer);
            printf(" > to go forward to question %d\n",pointer+2);
        }
}

```

```

printf(" . to return to question %d\n",last+1);
printf(" = to correct the answer to question %d\n",pointer+1);
if (pointer <= last) {
    printf(" (warning: this clears the answers to ");
    printf("following questions)\n"); } else;
printf(" q to quit, exiting the program\n"); }
else { /* walkback is NO */
printf("\n\n Valid responses to the > prompt are: \n\n");
printf(" ? to receive this help screen\n");
printf(" Y to choose the \"yes\" option\n");
printf(" N to choose the \"no\" option\n");
if (pointer > 0)
    printf(" < to back up to question %d\n",pointer);
printf(" q to quit, exiting the program\n\n");
printf("or a letter between \"a\" and \"%c\" ",
'a' + ans_num - 1);
printf("corresponding to the best answer\n");
} /* end of walkback test */
else {
/* print the same stuff in the help area on the PC */;
work80 = malloc(80);
if (walkback == YES) {
    row = 4;
    linedis("Valid inputs are:",row++,6,NORM,0);
    linedis(" ? to receive this help screen",row++,6,NORM,0)
    if (pointer > 0) {
        sprintf(work80," up-arrow to back up to question %d",pointe
        linedis(work80,row++,6,NORM,0); }
    sprintf(work80," down-arrow to go forward to question %d",pointe
    linedis(work80,row++,6,NORM,0);
    sprintf(work80," . to return to question %d",last+1);
    linedis(work80,row++,6,NORM,0);
    sprintf(work80," RETURN to correct the answer to question %d
    linedis(work80,row++,6,NORM,0);
    if (pointer <= last)
        linedis(" (warning: clears your answers to any fol
    else;
    linedis(" ESC to quit, exiting the program",row++,6,NORM,
else { /* walkback is NO */
    row = 4;
    linedis("Valid inputs are:",row++,6,NORM,0);
    linedis(" ? to receive this help screen",row++,6,NORM,0)
    linedis(" left-arrow move highlighting to previous option",row++
    linedis(" right-arrow move highlighting to next option",row++,6,N
    linedis(" RETURN select highlighted option",row++,6,NORM,0);
    linedis(" Y to choose the \"yes\" option",row++,6,NORM,
    linedis(" N to choose the \"no\" option",row++,6,NORM,0)
    if (pointer > 0) {
        sprintf(work80," up-arrow to back up to question %d",pointe
        linedis(work80,row++,6,NORM,0); }
    linedis(" ESC to quit, exiting the program",row++,6,NORM,
} /* end of walkback test */
linedis("Press SPACEBAR to continue...",row+2,6,REVNORM,0);
hide_cursor();

```

```

    while ((c = inkey(&ecode)) != 32);
} /* end of PC test */
return;
}

/* purpose of err_mess routine: to print an error message on the screen */
void err_mess(string)
char *string;
{
    if (PC == YES) cursor(15,0,0);
    printf("\n %s \n",string);
    return;
}

/* purpose of quit_yn function: to verify that the user really wants to
exit the function */

int quit_yn()
{
    int c;
    char s[10];
    if (PC == YES) cursor(5,0,0);
    printf("\nQuit? (y/n) ");
    fscanf(stdin,"%10s",s);
    c = (int)s[0];
    return(c);
}

/* purpose of update routine: update the history arrays based on the
last response */

void update(next,num_opt,quest_no,question,ans_num,used_table,
weight_hist,qa_hist)
int next;
int num_opt;
int quest_no;
struct questionstr *question;
int ans_num;
int used_table[MAXQUEST];
float weight_hist[MAXQUEST][MAXOPT];
struct qa qa_hist[MAXQUEST];
{
    float temp;
    int i;
    DEBUG(OUT,"entering update, next=%d,ansnum=%d\n",next,ans_num);
    for (i=0;i<num_opt;i++) {
        weight_hist[next][i] = weight_hist[next-1][i] *
            (temp = question->answer[ans_num].weights[i]);
    }
}

```

```

    DEBUG(OUT, "temp=%f\n", temp); }
    DEBUG(OUT, "next-1=%d, quest_no=%d, ans_num=%d\n", next-1, quest_no,
        ans_num);
    dump_qa(qa_hist);
    qa_hist[next-1].index = quest_no;
    qa_hist[next-1].ans = ans_num;
    used_table[quest_no] = ASKED;
    dump_qa(qa_hist);
    return;
}

/* purpose of output routine: to direct all output occuring at the end
   of the program */

void output(input_file, output_file, infile_name, tol, num_opt, option, num_quest
    question, qa_hist, final_weight)

FILE *input_file;
FILE *output_file;
char infile_name[];
float tol;
int num_opt;
struct optstr option[MAXOPT];
int num_quest;
struct questionstr question[MAXQUEST];
struct qa qa_hist[MAXQUEST];
float final_weight[MAXOPT];
{
    int i;
    struct qa qai;
    double toler;
    void backup(FILE *);
    void rewrite_input(FILE *, double, int, struct optstr[MAXOPT], int,
        struct questionstr[MAXQUEST]);

    void write_output(FILE *, float[MAXOPT], int, struct optstr[MAXOPT],
        double);

    void write_ans(FILE *, struct qa[MAXQUEST], struct questionstr[MAXQUEST]);
    int fclose(FILE *);

/* print message saying done with questioning */
    if (PC == NO)
        printf("\n\nQuestioning complete. Updating response frequencies...");
    else {
        clrscr();
        linedis("Questioning complete. Updating response frequencies...",
            5, 0, NORM, 0);
        hide_cursor(); }

/* update the histories */
    for (i=0; (qai=qa_hist[i]).index != EMPTY; i++)
        (question[qai.index].answer[qai.ans].history)++;

```

```

/* backup the input file */
DEBUG(OUT,"calling backup\n");
backup(input_file);

/* reopen it for writing */
input_file = fopen(infile_name,"w");

toler = (double)tol;
/* rewrite the input file */
DEBUG(OUT,"input_file reopened, calling rewrite output\n");
rewrite_input(input_file,toler,num_opt,option,num_quest,question);

/* determine the winner(s) */
DEBUG(OUT,"back from rewrite, calling write_output\n");
write_output(output_file,final_weight,num_opt,option,toler);

/* pass along important answers to avoid redundant questions */
DEBUG(OUT,"back from write_output, calling write_ans");
write_ans(output_file,qa_hist,question);

fclose(output_file);
return;
}

/* purpose of the backup routine: to make a copy of the input file prior
to rewriting it */

void backup(input_file)
FILE *input_file;
{
FILE *backup_file;
int fputc(int,FILE *);
int rewind(FILE *);
int fclose(FILE *);
int c;

backup_file = fopen("gaps.bak","w");
rewind(input_file);
while ((c=fgetc(input_file)) != EOF)
fputc(c,backup_file);
fclose(backup_file);
return;
}

/* purpose of rewrite_input routine: to rewrite the knowledge base */

void rewrite_input(input_file,tolerance,num_opt,option,num_quest,question)
FILE *input_file;
double tolerance;
int num_opt;
struct optstr option[MAXOPT];

```

```
int num_quest;
struct questionstr question[MAXQUEST];
{
    void write_header(FILE *,int,int,double);

    void write_option(FILE *,struct optstr);
    void write_quest(FILE *,int,struct questionstr);
    int i;

    rewind(input_file);
    write_header(input_file,num_opt,num_quest,tolerance);

    for (i=0;i<num_opt;i++)
        write_option(input_file,option[i]);

    for (i=0;i<num_quest;i++)
        write_quest(input_file,num_opt,question[i]);

    return;
}
```

/* purpose of write_header routine: to write the header information */

```
void write_header(input_file,num_opt,num_quest,toler)
FILE *input_file;
int num_opt;
int num_quest;
double toler;
{
    void write_int(FILE *,int);
    void write_float(FILE *,float);

    write_int(input_file,num_opt);
    write_int(input_file,num_quest);
    write_float(input_file,(float)toler);

    return;
}
```

/* purpose of write_option routine: to write the information concerning one option */

```
void write_option(input_file,option)
FILE *input_file;
struct optstr option;
{
    void write_float(FILE *,float);
    void write_string(FILE *,char[]);

    write_string(input_file,option.opt);
    write_float(input_file,option.minval);
}
```

```

    return;
}

/* purpose of write_quest routine: to write the information concerning
   one question */

void write_quest(input_file,num_opt,question)
FILE *input_file;
int num_opt;
struct questionstr question;
{
    int i;

    void write_int(FILE *,int);
    void write_float(FILE *,float);
    void write_string(FILE *,char[]);
    void write_answer(FILE *,int,struct answerstr);

    write_int(input_file,question.label);
    write_string(input_file,question.quest);
    write_int(input_file,question.numcond);

    for (i=0;i<question.numcond;i++) {
        write_int(input_file,question.cond[i].label);
        write_int(input_file,question.cond[i].ansnum);
        write_int(input_file,question.cond[i].flag);
    }

    write_int(input_file,question.ansnum);
    for (i=0;i<question.ansnum;i++)
        write_answer(input_file,num_opt,question.answer[i]);

    return;
}

/* purpose of write_answer routine: to write the information for one
   answer */

void write_answer(input_file,num_opt,answer)
FILE *input_file;
int num_opt;
struct answerstr answer;
{
    void write_string(FILE *,char[]);
    void write_weights(FILE *,int,int,float[MAXOPT]);
    void write_int(FILE *,int);

    write_string(input_file,answer.ans);
    write_weights(input_file,answer.history,num_opt,answer.weights);
    write_int(input_file,answer.out_index);
    write_string(input_file,answer.out_string);
}

```

```
    return;  
}
```

```
/* purpose of write_weights routine: to write the frequency and weights  
   for an answer */
```

```
void write_weights(input_file,freq,num_opt,weights)  
FILE *input_file;  
int freq;  
int num_opt;  
float weights[MAXOPT];  
{  
    int i;  
  
    fprintf(input_file,"%d ",freq);  
    for (i=0;i<num_opt;i++)  
        fprintf(input_file,"%f ",weights[i]);  
    fprintf(input_file," \n");  
  
    return;  
}
```

```
/* purpose of the write_int routine: to write an integer to a file */
```

```
void write_int(input_file,int_val)  
FILE *input_file;  
int int_val;  
{  
    fprintf(input_file,"%d \n",int_val);  
    return;  
}
```

```
/* purpose of write_float routine: to write a floating point number to file
```

```
void write_float(input_file,float_val)  
FILE *input_file;  
float float_val;  
{  
    fprintf(input_file,"%f \n",float_val);  
    return;  
}
```

```
/* purpose of write_string routine: to write a string to a file */
```

```
void write_string(input_file,string)  
FILE *input_file;  
char string[];  
{  
    fprintf(input_file,"%s \n",string);  
    return;  
}
```

}

```
/* purpose of write_output routine: to send the results to the screen
and the output file */
```

```
void write_output(output_file,final_weight,num_opt,option,tolerance)
```

```
FILE *output_file;
```

```
float final_weight[MAXOPT];
```

```
int num_opt;
```

```
struct optstr option[MAXOPT];
```

```
double tolerance;
```

```
{
```

```
float high; /* high value encountered so far */
```

```
int winner; /* holder of the high value */
```

```
int i,j,k;
```

```
float table[MAXOPT];
```

```
int key[MAXOPT];
```

```
int num_ent;
```

```
float tt;
```

```
int tk;
```

```
char prog_name[15];
```

```
char resp[10];
```

```
high = 0.00;
```

```
num_ent = 0;
```

```
winner = EMPTY;
```

```
for (i=0;i<num_opt;i++)
```

```
if (final_weight[i] > option[i].minval) {
```

```
if (final_weight[i] > high) {
```

```
winner = i;
```

```
high = final_weight[i];
```

```
}
```

```
/* put it into the table */
```

```
key[num_ent] = i;
```

```
table[num_ent] = final_weight[i];
```

```
num_ent++; }
```

```
/* bubble sort the table */
```

```
if (num_ent > 1) {
```

```
for (i=0;i<num_ent-1;i++)
```

```
for (j=0;j<=i;j++)
```

```
if (table[j] < table[j+1]) {
```

```
tt = table[j];
```

```
tk = key[j];
```

```
table[j] = table[j+1];
```

```
key[j] = key[j+1];
```

```
table[j+1] = tt;
```

```
key[j+1] = tk; }
```

```

}

if (PC == YES) {
    clrscr();
    cursor(5,0,0); }

if (winner == EMPTY) {
    fprintf(output_file,"None are applicable.\n");
    printf("None of the packages is applicable.\n"); }

for (i=0;i<num_opt;i++)
    if (final_weight[i] > option[i].minval)
        if (final_weight[i] + tolerance > high)
            fprintf(output_file,"%s %f \n",option[i].opt,final_weight[i]);
fprintf(output_file,"----- \n");

fprintf(stdout,"\n\n");
for (j=0;j<num_ent;j++)
    if (table[j] + tolerance > high) {
        fprintf(stdout,"\n%s is recommended for the analysis with ",
            option[key[j]].opt);
        fprintf(stdout,"confidence %4.2f.\n",final_weight[key[j]]); }
fprintf(stdout,"\n");

for (j=0;j<num_ent;j++)
    if (table[j] + tolerance > high) {
        fprintf(stdout,"Do you want assistance in using %s?",
            option[key[j]].opt);
        fscanf(stdin,"%s",resp);
        if (resp[0] == 'y' || resp[0] == 'Y') {
            strncpy(prog_name,option[key[j]].opt,3);
            strcpy(prog_name+3,"_front.exe");
            execl(prog_name,0);
        }
    }

return;
}

/* purpose of the write_ans routine: to write the values associated with
certain answers to the screen */

void write_ans(output_file,qa_hist,question)
FILE *output_file;
struct qa qa_hist[MAXQUEST];
struct questionstr question[MAXQUEST];
{
    struct qa qai;
    int out_table[MAXOUTLIN];
    int aoi;
    int high;
    int i;

```

```

for (i=0;i<MAXOUTLIN;out_table[i++]=EMPTY);

high = 1;
for (i=0;i<MAXQUEST && (qai=qa_hist[i]).ans != EMPTY;i++)
  if ((aoi=question[qai.index].answer[qai.ans].out_index) != EMPTY) {
    out_table[aoi] = i;
    if (aoi > high) high = aoi;
  }

for (i=1;i <= high;i++)
  if (out_table[i] == EMPTY)
    fprintf(output_file,"----- \n");
  else {
    qai = qa_hist[out_table[i]];
    fprintf(output_file,"%s \n",question[qai.index].answer[qai.ans].
      out_string);
  }
  fprintf(output_file,"--- \n");

return;
}

/* purpose of dump_qa routine: used during debugging to examine
the question and answer history array */

void dump_qa(qa_hist)
struct qa qa_hist[];
{
  int i;
  for (i=0;i<5;i++)
    DEBUG(OUT,"i=%d,index=%d,ans=%d\n",i,qa_hist[i].index,qa_hist[i].ans);
  return;
}

```

Source file name: gapsdef.h

```
#define ALLOWED 1      /* question is allowed based on previous response */
#define CANT_ASK 0    /* question cannot be asked because of prev. response
#define MAXOPT 3      /* maximum options, i.e. analysis programs */
#define MAXQUEST 30   /* maximum questions */
#define MAXANS 6      /* maximum choices of answers per question */
#define MAXCOND 5     /* maximum conditions on asking questions */
#define ASKED 1       /* question has already been asked */
#define NOT_ASKED 0   /* question has not already been asked */
#define MAXANSLEN 20  /* maximum answer string length */
#define MAXQUESTLEN 100 /* maximum question string length */
#define MAXOPTLEN 12  /* maximum option string length (PHOENICS) */
```

```
#define MAXOUT 10      /* maximum output string length */
#define EMPTY (-1)    /* spot in array is empty (undefined) */
#define END (-1)      /* no more questions to ask */
#define YES 1
#define NO 0
#define MAXOUTLIN 40  /* maximum index of output line */
#define ERROR (-10)   /* user gave invalid response */
#define ASK_AGAIN (-11) /* ask the question again */
#define EXIT_ASK (-12) /* end walkback mode, clear forward, ask again */
#define QUIT (-99)    /* user wants to exit program */
#define Y (-20)       /* user wants the "Yes" response */
#define N (-21)       /* user wants the "No" response */
#define PC YES        /* set to NO if not IBM PC (MS-DOS BIOS) specific */
#define TITLER 8      /* upper left corner of title */
#define TITLEC 24
#define TC REVNORM    /* title color */
```

Source file name: scrn.c

/* Functions for accessing the screen directly
by B.E. Prasad

Modified by Bill Habeck for Microsoft C. */

#define VIDEO_INT 0x10

```

/*****
 *
 * inkey.c <==> Accepts keystroke from the keyboard and interprets
 * its value(code)
 *
 * <---->
 * <----> ecode (if the is code is extended code, the scan co
 * in register AH is placed in it. otherwise ignore it
 * |___ Register AL value.
 *****/

```

```

*****/
char inkey(ecode)
char *ecode;
{

```

```

    union REGS regs;
    char c;
    regs.h.ah = 0x0f;
    int86 (VIDEO_INT, &regs, &regs); /* get video status */
    regs.h.ah=0;
    int86(0x16, &regs, &regs);
    *ecode=(char)regs.h.ah;
    c=(char)regs.h.al;
    return(c);
}

```

```

}

/*****
*
* clearscr.c <==> Clears the screen
*          <---->
*          <----->
*          |_____ void
*
*****/

void clearscr()
{
    union REGS regs;
    regs.h.ah = 0x0f; /* get video status */
    int86 ( VIDEO_INT, &regs, &regs);
    regs.h.ah = 0; /* set mode & clear screen */
    int86 ( VIDEO_INT, &regs, &regs);
}

void blank(num, row, column, page)
int num, row, column, page;
{
    int i;
    void chardis(int, int, char, int, int);
    for (i=0; i<num; i++)
        chardis(row, column+i, 'a', INVIS, page);
    return;
}

void cursor(row, column, page)
int row, column, page;
{
    union REGS r;
    r.h.ah = 0x02;
    r.h.dh = (unsigned char)row;
    r.h.dl = (unsigned char)column;
    r.h.bh = (unsigned char)page;
    int86(VIDEO_INT, &r, &r);
    return;
}

void chardis(row, column, chr, color, page)
int row, column;
char chr;
int color, page;
{
    union REGS r;

    r.h.ah = 2;
    r.h.dh = (unsigned char)row;
    r.h.dl = (unsigned char)column;

```

```
r.h.bh = (unsigned char)page;
int86(0x10,&r,&r);
r.h.ah = 9;
r.h.bh = (unsigned char)page;
r.h.bl = (unsigned char)color;
r.x.cx = 1;
r.h.al = (unsigned char)chr;
int86(0x10,&r,&r);
return;
}
```

```
void linedis(linestr,row,col,color,page)
char linestr[];
int row,col,color,page;

{
    int j;

    for (j=col;j<col+strlen(linestr);j++)
        chardis(row,j,linestr[j-col],color,page);

    return;
}
```

```
void setmode(mode)
int mode;
{
    union REGS r;
    r.h.ah = 0;
    r.h.al = (unsigned char)mode;
    int86(0x10,&r,&r);
    return;
}
```

```
void setpage(page)
int page;
{
    union REGS r;
    r.h.ah = 5;
    r.h.al = (unsigned char)page;
    int86(0x10,&r,&r);
    return;
}
```

Source file name: scrn.h

```
#define INVIS 0
#define NORM_ 1
#define NORM 3
#define HIGH_ 9
#define HIGH 11
#define REVNORM_ 16
#define REVNORM 18
#define REVHIGH_ 24
#define REVHIGH 31
#define FLASH 128
```

Appendix B
GAPS Input File Listing

Input file name: gaps.inp

3

11

0.100000

PHOENICS

0.300000

CAEDS

0.300000

ITAM

0.300000

10

Does the analysis require three-dimensional modeling?

0

3

Yes

9 1.000000 0.990000 0.000000

-1

No

4 0.950000 0.980000 1.000000

-1

Don't know

2 0.970000 0.970000 0.950000

-1

20

Are dynamic results (as opposed to steady-state) necessary?

1

10

0

1

3

Yes

5 1.000000 0.000000 0.000000

-1

No

3 0.990000 1.000000 1.000000

-1

Don't know

1 0.970000 0.960000 0.960000

-1

30

Does the design contain a fan?

0

3

Yes

3 0.950000 0.900000 1.000000

-1

No
2 1.000000 0.950000 0.000000
-1

Don't know
3 0.970000 0.970000 0.970000
-1

35
Can all the fans be cut with a single plane?

1
30
0
1
4

Yes
3 0.980000 0.950000 1.000000
-1

All but 1
1 0.990000 0.980000 1.000000
-1

No
1 1.000000 1.000000 0.600000
-1

Don't know
1 0.970000 0.970000 0.970000
-1

40
Do you need interactive graphics to express your design?

0
3

Yes
4 0.000000 1.000000 1.000000
1

IG
No
3 1.000000 1.000000 1.000000
-1

Don't know
2 0.970000 0.970000 0.970000
-1

50
Can you use CADAM to express your design?

1
40
0
1
3

Yes

4 1.000000 1.000000 1.000000
-1

No

1 1.000000 1.000000 0.000000
-1

Don't know

1 0.970000 0.970000 0.970000
-1

60

Is significant volume change occurring in the model (such as in a piston)?

0

3

Yes

1 1.000000 0.200000 0.000000
-1

No

5 0.990000 1.000000 1.000000
-1

Don't know

1 0.970000 0.960000 0.960000
-1

70

Is there an air flow consisting almost completely of upward air movement?

0

3

Yes

3 1.000000 0.800000 0.900000
-1

No

3 0.950000 1.000000 1.000000
-1

Don't know

1 0.970000 0.970000 0.970000
-1

200

Do you have access to PHOENICS?

0

3

Yes

3 1.000000 1.000000 1.000000
-1

No

1 0.000000 1.000000 1.000000

-1

Don't know

1 0.800000 1.000000 1.000000

-1

210

Do you have access to CAEDS?

0

3

Yes

3 1.000000 1.000000 1.000000

-1

No

2 1.000000 0.000000 1.000000

-1

Don't know

3 1.000000 0.800000 1.000000

-1

220

Do you have access to ITAM?

0

3

Yes

4 1.000000 1.000000 1.000000

-1

No

1 1.000000 1.000000 0.000000

-1

Don't know

1 1.000000 1.000000 0.800000

-1

DATA COMMUNICATION NETWORKS: A COMPARATIVE EVALUATION

PHILLIP SEUNG-HO YOO

A comparative evaluation of data communication networks in two large organizations highlights a number of critical connectivity issues. These issues have been identified and analyzed through the medium of personal interviews, supplemented by survey questionnaire data gathered specifically for this effort.

The user community in both organizations consists of several different segments, each with its own set of goals and priorities. While bandwidth is important to one segment, security is the most important issue to another segment. The level of technical background of users ranges from novice to very sophisticated. The diversity in the user community has been one of the major constraints in developing an organization-wide network.

Individual departments of both organizations have implemented their own network solutions. Based on the availability of financial resources, these departments have gone ahead in implementing closed and proprietary solutions, which cannot easily be merged together.

Because both organizations have a totally decentralized mechanism for making decisions relating to acquisition of hardware and software, they are populated with an unusually large array of heterogeneous computational facilities. It is very difficult to come up with any technical solution that can cover the full breadth of the problem.

Voice and computer based information have so far been transmitted over separate systems. Also, while computer based information was earlier mainly of numerical and textual types, the advent of pictorial and graphical representations is increasing the communication load. The new generation of networks is being designed to cover all types of information.

The responsibilities of the owning administrative group and the client administrative group are being distinguished. The former has access to data on a read-write basis. In addition, it creates duplicate sets of its database. These duplicate sets are on a read-only basis. Subject to their being authorized access, client groups are reading information from the latter set only. This concept of duality needs further refinement.

Both organizations are currently working on the above issues. The hope is that the final system will indeed be a true distributed system which will improve speed of access, eliminate all duplication, and significantly reduce the amount of paperwork.

1 Research Objective and Methodology

The objective of this research is to explore the issues and problems faced by Information Systems executives in managing the data communications needs of the organization. Particular attention is paid to problems associated with data network connectivity.

For these purposes, the organization is assumed to consist of a number of departments or functional groups with differing needs and agendas that must be serviced by a central Information Systems organization. Close examination of two sophisticated existing environments may illuminate policy alternatives for IS planners.

A data communications network must support the services required by the various users of the system. The design and implementation of a system to satisfy these requirements must take into account certain historical restrictions and constraints inherited from existing systems. The issues that the author has identified are:

- **Heterogeneous hardware** — different departments and subgroups will have made independent decisions regarding their own computing needs that may not conform to the organization's formal or informal standards. A network design must accommodate and integrate these existing systems.
- **Architectural constraints** — characteristics of the organization's buildings may preclude or alter certain network design decisions. Network solutions may be sub-optimal but necessary in view of these constraints.
- **Wide range of user sophistication** — the data network must be accessible and adequate for both the novice and expert user. Education and support become key issues that must be addressed.
- **Wide range of user needs** — various different users will have different needs and requirements. Some will require different services, while others may require strict data security. For yet a different group, cost containment will be their primary concern. All these issues must be satisfied by a successful complete data network solution.

- **Existing networks** — departments and groups within the organization may have found it necessary to implement data networks to satisfy their own needs. These solutions must be integrated into any overall design in order to be successful.
- **Data networks are evolutionary** — the topology of a network is constantly changing as incremental users request and are granted service. It may be exceedingly difficult to adhere strictly to a planned design.
- **Network management is often fragmented** — since the data network must satisfy both department-specific needs as well as central planning issues, management is often shared between the central agent and the departmental groups.

This list is not intended to be exhaustive, however, it represents the core of critical issues that must be addressed by IS planners.

The author has selected the MIT and Harvard University environments for study because they demonstrate all of the above historical issues.

- **They are technologically sophisticated** — both universities have complex and diverse communications needs.
- **They are populated with heterogeneous hardware** — products are acquired both through decentralized purchasing decisions and outright grants. Universities are unable to enforce any hardware standards and must therefore address integration issues.
- **University buildings often predate data network needs** — a number of the buildings both at MIT and Harvard are poorly adapted to accommodate data network wiring .
- **They have both very sophisticated and novice users**
- **There are three distinct classes of users** — students, faculty, and administrative users all have different needs and requirements that complicate the data network planning process.
- **Many departments have implemented their own network solutions** — networks are often provided by together with equipment grants or are implemented to meet departmental needs. Often these solutions are closed solutions, proprietary to the donor vendor and present significant integration problems. Both MIT and Harvard possess numerous links to external networks giving rise to security problems not faced by closed systems.

- **Both their networks have evolved over time**
- **Their networks are managed jointly by central and departmental agents — both universities have a central IS planning office as well as departmental network managers that serve their own groups.**

1.1 The Use of Comparative Evaluation

Comprehensive evaluation requires an ideal standard for objective comparison. Since data networks evolve through incremental additions, the current topology and technology rarely adheres to a notion of an optimal solution. If an organization possessed infinite financial resources and could afford the time to completely replace the system and retrain all the users, then it could alter its network in response to every technological advance. This is hardly a reasonable assumption.

The use of a comparative evaluation eliminates this difficulty. The definition of an objective standard is a task beyond the scope of this work. The comparative evaluation will examine both the current network implementations of both universities as well as their plans for future expansion. Both networks will be judged on their ability to meet the needs of each university. This examination will be made with respect to a number of evaluation criteria.

1.2 Scope of the Evaluation

This evaluation examines data networks as they are used at both institutions. Both universities contain "islands of networks" as well as a more prolific "main" system¹. The author examines both these components since it is likely that the existence of isolated data networks suggest inadequacies and incapacibilities of the main network to service critical needs.

¹ Modem connections for ad hoc communications are not treated as network links.

The author nonetheless takes the perspective of the central IS planner in his task of successfully integrating and supporting the variegated needs, resources, and sophistication of numerous clients.

2 Evaluation Methodology

The data central to the evaluation is obtained through interviews with Information Systems network managers as well as survey questionnaire data. The survey data shed light on the network's effectiveness in servicing the various types of users in the target community. The survey asks respondents to evaluate the network on several different criteria.

2.1 User Community

The target user community consists of faculty, administration, and students. Each group has a slightly different set of needs and preferences. The appraisals of these groups needs not be identical since they may be receiving different levels of service.

Faculty users use the network to support their research and to help coordinate collaborative efforts with colleagues. Students use the network to gain access to the host computers they need for coursework, papers, games, and mail. Administrative users make heavy use of internal databases, presenting unique security issues.

2.2 Evaluation Criteria

A number of factors must be examined in order to adequately evaluate a data network. Some of the factors address the coverage of service delivered to the users. Other issues relate to the manageability and operability of the network design. The following evaluation criteria will be applied:

- Functionality
- Network Reach — Connectivity
- Network Performance and Reliability
- Network Control

- Network Support — Maintainability
- Security
- Planning

2.2.1 Functionality

A data network can provide many functions above and beyond connecting terminals to computers. It maintains information on all the computers and resources it connects so that it can route, store, and translate messages from one computer to another with a minimum of user training.

In a university environment, Harvard University has found the following functions to be critical to the user community, listed in descending priority.

- **Database Access** makes information stored in computers available to users. This includes gateways to outside databases and the network computer resource directory.
- **Resource Sharing** allows for sharing expensive disk drives, printers, plotters, file servers, *etc.*, among a number of personal or small departmental computers with a minimum of special commands or software modifications.
- **Document Interchange** converts revisable word processing documents into a standard form so that they may be communicated on the network and reconverted for use on a different word processor.
- **File Transfer** enables users to move data and/or text documents across the network.
- **Image Communications** provides high-speed communications to meet the special requirements of electronic publishing and graphics/image transmission.
- **Electronic Mail** maintains a university-wide user directory and stores and forwards messages to multiple locations. Electronic mail acts as the envelope and the post office for the delivery of all kinds of electronic communication including revisable WP documents and images. A university electronic mail system should provide

connectivity between other electronic mail systems on and off campus.

- **Terminal-Host Communications** enable users to access host computers using simple interactive terminals over the network.

It should be noted that the various segments of the user community may assign differing levels of importance to these varying services, based on their own needs and requirements.

2.2.2 Network Reach — Connectivity

In addition to providing the functional services required by the users, a data network must reach all clients and resources to which users desire access. Furthermore, users that desire service should be able to gain access. It is not sufficient for a network to simply support file transfer to a select set of users. When protocol incompatibility or the lack of a physical connection prevent this transaction with the desired host computer or file server, the network is not providing adequate connectivity.

Assessing connectivity involves both physical links as well as protocol compatibility. The establishment of a physical connection depends critically on the current wiring to date. If network drops have already been established nearby, then the task is simple and inexpensive. If the building has not yet been wired, then the incremental costs may be considerable (over \$20,000).

The incompatibility of protocols can interfere with connectivity as well. A user may desire communication with another client physically connected to the network, but be unable because his machine doesn't understand the other's protocol.

2.2.3 Network Performance and Reliability

The next important criteria is the raw performance of the network in moving data. Once the user has seen to it that the services he desires have been supported, and that the network has sufficient reach to provide him access to whomever and whatever he wishes; his concern turns to the network's speed and reliability in providing the service.

Performance is normally measured in terms of the data rate (or *bandwidth*) supported by the network, data link, and physical layers. The principal Ethernet standard has a bandwidth of 10 million bits per second (Mbps). However, the actual data throughput performance may be a mere fraction of this. Higher level protocols consume a significant chunk of the available bandwidth in order to effect error detection and correction. One widespread protocol, TCP/IP delivers an effective bandwidth of approximately 1.5 Mbps over a typical Ethernet.

The above measures are still performance measures for ideal conditions. The observed performance is very much a function of the load factor on the network. If several machines are contending for the same network at the exact same time, a great deal of bandwidth will be consumed in resolving the contention.

2.2.4 Network Control

In addition to providing requisite functional services, the data network must have control elements to help it to respond to users' needs. Research at Harvard University has uncovered two principal issues:

- Cost control — the ability to monitor and limit usage of the network, shared resources, and databases accessed by the network.
- Network management — the ability to monitor network performance, to administer network identification and password information, and to perform diagnosis on network components².

Faculty groups would like to be able to control the usage costs of outside research databases. Additionally, administrative users would like to be able to control network usage costs for external networks.

² "Harvard University Long-Range Telecommunications Plan: Needs Assessment Summary," Harvard University Office for Information Technology, Telecommunications Services Division, August 13, 1986, p. III-20.

2.2.5 Network Support — Maintainability

A data network is much more than simply providing a technology. It also includes support functions to ensure that the network is installed properly and that users are trained in its operation. Support issues fall into three primary areas:

- Technical — assistance to users on how to install data center network software, resolve technical problems, and provide data center users with an understanding of how the network functions;
- End-user training — on how to access the network, the steps required to connect to different computers, and how to identify and report problems with the network;
- Maintenance — such as installing new software updates, diagnosing network errors, and installing new facilities³

2.2.6 Security

The data security issue is critically important in a university environment. Students must be denied access to the class work of their classmates. Sensitive administrative information like grades, financial situation, and payroll must be kept secure.

It is interesting that the critical issue is not a general measure of data security, but the perception of security by users requiring it. Users in the Medical area or the Office of the Registrar have more stringent security requirements than do faculty members concerned with student plagiarism.

The security issue is complex, because in order to ensure security for administrative data, the network must be designed to limit student access to machines that contain sensitive information.

2.2.7 Planning

³ *Ibid.*, pp. III-22 and III-23.

This criterion attempts to address the efficacy of central planning in anticipating and addressing users' needs in defining future expansion and redesigning the network. It also attempts to assess the success of the IS planner in effectively coordinating the implementation by gaining the cooperation of the key stakeholders like departmental Information Systems managers.

3 Principal Protocols

In order to fully appreciate some of the connectivity and functionality issues encountered in data networks it is necessary to understand the capabilities and limitations of the menu of principal protocols in use in the networking environment. All the protocols described below are incompatible with one another. Often network managers do not have a choice in adopting network protocols, since they may be dictated by the hardware vendor.

The network protocols refer to the middle three layers of the Reference Model of Open System Interconnection (OSI) developed by the International Standards Organization (ISO)⁴. The OSI defines distinct layers according to defined principles:

- A layer should be created where a different level of abstraction is needed;
- Each layer should perform a well defined function;
- The function of each layer should be chosen with an eye toward defining internationally standardized protocols;
- The layer boundaries should be chosen to minimize the information flow across the interfaces;
- The number of layers should be large enough that distinct functions need not be thrown together in the same layer out of necessity and small enough that the architecture does not become unwieldy⁵.

Protocols that adhere strictly to the OSI layer boundaries will map very closely to one another, facilitating protocol conversion. Furthermore, the simplification of the interface indicates that upper layer protocols may be laid over any of several different lower layer protocols without difficulty.

The ISO OSI reference model defines seven layers:

⁴ Andrew S. Tanenbaum, *Computer Networks* (Englewood Cliffs, NJ: Prentice Hall, 1981), pp. 15-21.

⁵ H. Zimmerman, "OSI Reference Model — The ISO Model of Architecture for Open Systems Interconnection," *IEEE Transactions on Communications*, Vol. COM-28, April 1980, pp. 425-432.

7. **Application layer** — The content of this layer is up to the individual user. When two user programs on different machines communicate, they alone determine the set of allowed messages and the action taken upon receipt of each.
6. **Presentation layer** — performs functions that are requested sufficiently often to warrant finding a general solution for them, rather than letting each user solve the problems. These functions can often be performed by library routines called by the user.
5. **Session layer** — is the user's interface into the network. The user must negotiate with this layer to establish a connection with a process on another machine.
4. **Transport layer** — also known as the host-to-host layer, accepts data from the session layer, splits them up into smaller units (if need be), pass these to the network layer, and ensure that the pieces all arrive correctly at the other end.
3. **Network layer** — sometimes called the communication subnet layer, controls the operation of the subnet. This layer basically accepts messages from the source host, converts them to packets, and sees to it that the packets get directed toward the destination.
2. **Data link layer** — takes a raw transmission facility and transforms it into a line that appears free of transmission errors to the network layer.
1. **Physical layer** — concerned with transmitting raw bits over a communications channel. The design issues here largely deal with mechanical, electrical, and procedural interfacing to the subnet.

The lower three layers are primarily dictated by the transport medium (*e.g.*, Ethernet and X.25 public packet-switch networks). The application layer is of primary interest to sophisticated users beyond the scope of this study. The presentation layer contains general purpose services like remote login, file transfer, and data encryption which are of interest. However, compatibility within the middle three layers will determine the ability of networks to offer uniform presentation layers.

Differing network protocols for the middle layers may be laid on top of the same network layer. DECNET, SNA, XNS, and TCP/IP may all be implemented on an Ethernet

(often simultaneously). This study examines protocol decisions at the middle layers of the ISO reference model.

3.1 TCP/IP

The Department of Defense developed the Transmission Control Protocol/ Internet Protocol (TCP/IP) for its ARPANET. Because ARPANET is a nationwide network of protocols, TCP/IP was designed to be extremely flexible. It can connect with many different kinds of computers and its addressing system can accommodate hosts on many different networks. TCP/IP supports three standard functions: network mail, file transfer, and remote login⁶.

Almost all TCP/IP hosts implement two standard user applications (protocols): FTP, or File Transfer Protocol, and TELNET, a basic remote login protocol. The Simple Mail Transfer Protocol (SMTP) is widely used to distribute electronic mail messages throughout the Internet domain. TCP/IP has become the closest thing to a protocol standard at the transport and presentation layers of the ISO networking standard. The TCP/IP protocols have been implemented on a number of hardware platforms and operating systems. Notably, nearly all UNIX systems use TCP/IP networking protocols.

3.2 DECNET

DECNET protocols are a central component of Digital Equipment Corporation's Digital Network Architecture (DNA). DECNETs are closed systems in that the protocols are proprietary to DEC. DECNET will run over both generations of Ethernet (thin and fat cable) as well as over fiber-optic cable and twisted-pair cables. DEC also supports wide area network (WAN) capabilities to help users link local area networks spanning several

⁶ "Networks at MIT," Information Systems Series Memo IS-10-1, MIT, November 13, 1986, p. 5.

cities or countries throughout the world.

DECNET supports several application functions, including: file transfer between any two devices in the network, electronic mail, and remote login (subject to privilege security restrictions). A facility called finger allows a user to determine who is logged onto the DECNET environment and where.

DEC also provides some related services that are part of the Digital Network Architecture. The Maintenance Operations Protocol (MOP) enables the system manager to download software over the network as well as run diagnostics on remote machines. The Local Area Transfer protocol (LAT) is used to link DEC's terminal servers to DECNET hosts.

An important recent extension is DEC's Local Area VAX Clustering (LAVC) supported over the Ethernet. This service provides for remote booting and remote file serving among DEC VAX machines. The LAVC protocol as well as all the above are registered with ISO though they are proprietary to the Digital Equipment Corporation⁷.

3.3 SNA

Systems Network Architecture (SNA), announced by IBM in 1974, is IBM's strategic communications blueprint from which to define, design, and implement interconnection and resource sharing among communications network products. These specifications provide the set of rules, logical structures, procedures, formats, and protocols that are implemented in various hardware and software products⁸.

SNA is implemented in a variety of IBM hardware and software products, but IBM seems to be favoring the adoption of SNA as an open standard. The company has

⁷ *Networks and Communications Buyer's Guide*, Digital Equipment Corporation, Maynard MA, 1986 October — December, pp. 1.1-1.9.

⁸ Thomas J. Routt, "Distributed SNA: A network architecture gets on track," *Data Communications*, February 1987, p. 116.

significantly extended its architecture to provide the broad range of services required of a fully functional network architecture. IBM supports the interconnection of distinct and separate SNA networks through its SNA Network Interconnection (SNI). It has further recently introduced distributed services to address trends toward general decentralization and the migration of data processing and communications capabilities to desktop workstations.

SNA currently provides support for remote login, mail, and file transfer. Future products will soon support document interchange and distribution services as well.

3.4 PRONET

PRONET was developed by PROTEON to support its networking products. It is the basis for PROTEON's NOVELL operating system that provides users a broad range of services in addition to the basic file transfer, remote login, and electronic mail. NOVELL is a network operating system designed especially for microcomputers that has rapidly become an industry standard for package software developers. PROTEON has widely distributed the specifications for interfacing to NOVELL and most major software developers now offer a NOVELL network version (*e.g.*, DBase, Microsoft Word).

NOVELL offers the ability to share files as well as printers. For file access, NOVELL provides superb security. Multiple users can access the same file and even modify different records within the same file simultaneously. Most workstations in a NOVELL environment have only a floppy disk drive — a single copy of all applications software is maintained on a shared file server.

In addition to its distributed functionality, NOVELL offers fast access and fault tolerance. The user interface is menu-driven for ease of use. NOVELL users are reluctant to relinquish the greater flexibility and functionality merely to gain TCP/IP compatibility.

3.5 NFS

Sun Microsystems developed the Network File System (NFS) to support its broad range of "disked" and diskless workstation products. It provides users with highly transparent file access. A user may be oblivious to the fact that his files do not actually reside on his host or workstation, but may be maintained on a remote file server. NFS functions between different types of operating systems, giving users on NFS hosts access to files on "alien" machines.

Like PROTEON's NOVELL, NFS offers a number of benefits. Being able to move heavily accessed files to central servers can greatly reduce workstation costs by shrinking the remote station's disk capacity. Having a central file server lowers maintenance costs and simplifies the process of making tape backups. SUN has provided additional resources to facilitate shared development among programming teams as well.

3.6 XNS

Xerox Networking Service (XNS) is a set of protocols developed by Xerox for local area networking. The XNS protocols parallel TCP/IP in their functionality. It offers file sharing services lying somewhere between TCP/IP FTP and SUN's NFS. Unlike NFS, XNS users edit a copy of their document, but the link is much tighter than with FTP. With XNS services, users download their files from a central server to their local workstation for editing and development. When finished, the user uploads the modified documents to the central server.

XNS also provides centralized servers for electronic mail, authentication, and printer spooling. Xerox has not been very successful in establishing XNS as a *de facto* standard.

4 Characterization of the MIT Environment

MIT has one of the most highly networked computing environments in the country.

4.1 Network Topology

The MIT data network consists of a campus-wide backbone network linking client sub-networks as well as a number of isolated "islands of networks." Backbone and attached client subnets are referred to as the MIT Campus Network. The Campus Network maintains links to a number of external networks as well. A number of departments like the Medical department, Administrative Systems, and Registrar's office maintain their own local area networks that are not a part of the Campus Network.

4.1.1 Internal Networks

4.1.1.1 Backbone

The backbone is a 10-megabit fiber optic token ring consisting of gateways to each of the sub-net clients. The backbone runs both TCP/IP and CHAOSNET, TCP/IP being the standard for all campus-wide communication. The central IS manager, Telecommunications Systems, supports and maintains the fiber optic transmission medium as well as the gateways. Campus Network service is supplied by installing a gateway (MicroVAX II) in the building being served as well as running cable to the host.

4.1.1.2 Subnets

The sub-networks are almost all Ethernet (10-megabit coaxial) with half-repeater/fiber optic connections to other buildings. Several different protocols are utilized by the various sub-nets (DECNET, TCP/IP, XNS, CHAOSNET). These sub-networks are managed by departmental managers who often hire their own technical staff to support and maintain their LANs.

Since the backbone protocol standard is TCP/IP, any communications between sub-networks must use TCP/IP. This is not a difficulty for subnets using that protocol, but DECNET and XNS LANs have difficulties gaining inter-subnetwork service. These difficulties will be discussed at length in Chapter 7. The onus falls on the departmental manager to purchase the protocol conversion hardware and software in order to enable inter-subnetwork communication.

At MIT, the TCP/IP set of protocols runs on several kinds of hardware and with a variety of operating systems. All Project Athena machines run TCP/IP. Even before the initiation of the campus-wide network, all computers with direct connections to the ARPANET ran TCP/IP as did some computers that accessed ARPANET via these directly connected machines.

4.1.1.3 Isolated Networks

Some sub-nets are isolated from the backbone, both by choice and through connectivity problems. LANs for the Medical Department, the Office of the Registrar, and Administrative Systems are isolated from the Campus Network for security reasons. All three organizations are concerned a compromise of network data security could result in the release of sensitive information. As a result, they maintain their own closed network, resorting to dedicated terminal lines or modems to gain access to administrative timeshare host computers on an as necessary basis.

Administrative Systems and the Medical Department are using Proteon PRONET protocols together with NOVELL operating system. They are both reluctant to go to TCP/IP because it does not support the full functionality they require within their local network environment. NOVELL capabilities in sharing files and printers are very valuable to these users.

The Medical Department has used a central file server to eliminate the cost of a hard disk for each its IBM PC workstations. Workers keep personal files on floppies or on the

central server. A single network version of DBase or Microsoft Word is more cost effective than supplying each user with his own copy. Maintenance and tape backup are streamlined since one copy is less time consuming than 40. Printer spoolers improve the accessibility and affordability of laser printing.

Both would quickly adopt a solution enabling them to use both NOVELL and TCP/IP for inter-network communications tasks (file transfer, electronic mail, and remote login) but do not have the resources or technical expertise to create a solution. The problem could be solved through software, but neither group has the technical staff to complete the development. A hardware solution could be purchased but would run \$20,000-25,000.

Other networks are isolated because of difficulties in wiring that particular building. Still others present severe protocol incompatibility problems that preclude joining the Campus Network without undertaking a significant development project.

4.1.1.4 CHAOSNET

The CHAOSNET is a home-grown MIT product, developed at the Artificial Intelligence Laboratory as a local network for its LISP machines. It outgrew this original form and spread to other research groups around campus. Before the advent of the Campus Network, the CHAOSNET was the largest-scale attempt at a coherent medium for communication between MIT computer facilities.

Though the CHAOSNET protocol is incompatible with TCP/IP, it features a very similar user interface, including both the FTP and TELNET protocols. In spite of this superficial resemblance, you cannot usually transfer files between IP and CHAOS hosts or log in to a host on the other networks via TELNET⁹. Special multi-protocol gateways must be installed to support transfer between specific sub-networks.

⁹ "Networks at MIT," Information Systems Series Memo IS-10-1, MIT, November 13, 1986. p. 5.

4.1.2 External Networks

4.1.2.1 ARPANET

The ARPANET is one of the oldest, largest, and most fully-implemented of the long-distance networks. Established by the Department of Defense, access is limited to organizations and people engaged in federally funded research. This network was recently split in half. The military and defense contractors were separated onto their own secure and reliable network called MILNET. The ARPANET remains more experimental, serving the more general research institutions. A gateway between the two networks lets outsiders send mail to MILNET members. The TCP/IP protocols implemented at MIT were originally developed for the ARPANET¹⁰.

4.1.2.2 CSNET

CSNET is a research network linking computer scientists and engineers at sites throughout the United States, Canada, and Europe. It was developed to provide TCP/IP-type services to computer science institutions that weren't part of the ARPANET, and to make electronic mail exchange possible with ARPANET hosts. Initial funding was furnished by the National Science Foundation with the understanding that eventually the network would become self-sufficient.

Membership in CSNET is open to any organization engaged in research or advanced development in computer science or computer engineering. Members include universities, corporations, government agencies, and non-profit organizations. CSNET users are professors, graduate students, undergraduates, corporate research staff, visiting scientists, government researchers, and other professionals in the field of computer science and electrical engineering¹¹.

¹⁰ *Ibid*, p. 8.

¹¹ *Ibid*, p. 9.

4.1.2.3 BITNET

BITNET connects mainframes at universities and other research institutions worldwide. It is expanding rapidly and now includes about 1500 sites (network nodes). All users at member institutions can access the facilities that BITNET offers. At present these include electronic messages and mail, and the transfer of programs and documents, but not remote login.

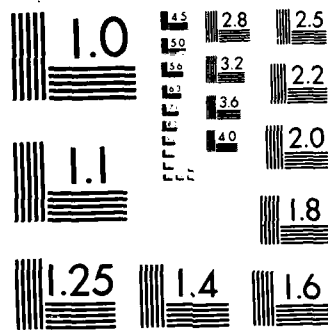
BITNET is inexpensive to use and maintain. Rather than designing its own network software, or using the TCP/IP protocols, BITNET takes advantage of a standard IBM facility called RSCS (Remote Spooling Communications Subsystem) for VM, and JES2 or JES3 for MVS, which are already in place on the network hosts. Because of the network's rampant growth, "JNET" software has also been developed to connect to BITNET from DEC computers running VMS, and "UREP" software to connect UNIX systems. Each member institution contributes its share of the network by leasing a line from a telephone company to link with a nearby network node, and accessing this line through a 9600 bps modem¹².

4.1.2.4 USENET

Just as BITNET uses RSCS, the USENET network uses software that comes as part of UNIX. The name UUCP (UNIX-to-UNIX Copy) can be applied to two different network services.

The original UUCP is a file transfer program. It permits the transfer of files between two UNIX systems, either over hardwired lines or by dialing up. A mail service was subsequently grafted on top of the original UUCP, forming a mail network. UUCP

¹² *Ibid.*, p. 7.



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

mail permits forwarding of mail over several systems, but does not handle the routing or acknowledge errors.

4.1.2.5 Supercomputer networks (JVNC Net)

MIT maintains T1 network links both to the John von Neumann Computing facility in Princeton, NJ as well as Harvard University. This network supports National Science Foundation work in supercomputing.

4.1.2.6 Centrex

MIT's voice network needs are currently served by an IA ESS Centrex system located in New England Telephone's Central Office on Ware Street in Cambridge. In addition to basic telephone service, Centrex supports low speed dial-up communications up to 4800 bits per second (bps) using modems. This capability is currently being used for time-sharing computer access, asynchronous file transfers and access to external networks and remote database. MIT is in the process of acquiring a 5 ESS voice/data PBX to replace its Centrex system. The ramifications will be examined in Chapter 7.

4.2 Telecommunications Systems

Telecommunication Systems is the central administrator of both the MIT Campus Network backbone and telecommunications services. It operates and services the campus-wide phone services. It also operates and maintains the Proteon token ring backbone. Any client wishing to join the Campus Network must negotiate with Telecommunications Systems to acquire service.

Telecommunications Systems handles all installation and maintenance of gateways to the Campus Network backbone. Consulting services are available on a fee basis.

4.3 Project Athena

Project Athena is a five-year program to explore new, innovative uses of computing in the MIT curriculum. Major computer manufacturers have developed high-performance graphics affordable workstations that may significantly impact undergraduate education. The MIT faculty was concerned that too little was being done to integrate the new computational technology into the undergraduate educational experience. Project Athena arose from this concern.

Project Athena's workstation clusters are scattered throughout the Campus Network. Project Athena staff play a major role in defining and influencing planning for the entire network¹³. Athena is significantly advancing the state of the art in distributed computing.

¹³ Steven R. Lerman, "Questions and Answers About Project Athena," MIT Project Athena, Revision C, November 1986, p. 2.

5 Characterization of the Harvard Environment

Harvard University trails MIT in terms of communications sophistication. There is no real central backbone interconnecting the various sub-networks that populate the campus.

5.1 Network Topology

At present, Harvard University does not have in place a campus-wide network. There are a variety of data network resources put into place to meet the needs of the faculties and departments at Harvard.

5.1.1 Internal Networks

5.1.1.1 Centrex

Like MIT, Harvard University's voice network needs are currently served by an IA ESS Centrex system. Harvard also uses Centrex to support 4800 baud dial-up communications through modems. This capability is currently being used for time-sharing computer access, asynchronous file transfers and access to external networks and remote databases¹⁴.

5.1.1.2 OIT Network

The Computing and Information Utilities Division (CIU) of the Office of Information Technology (OIT) operates a network to provide users throughout the university access to its centralized computing facilities at 1730 Cambridge Street.

¹⁴ "Harvard University Long Range Telecommunications Plan: Resource Summary," Harvard University Office for Information Technology, Telecommunications Services Division, August 13, 1986, p. III-1.

Network services include low speed dial-up services via Centrex facilities, dedicated links for support of IBM 3270 Bisync devices at 9600 bps, and specialized networks such as the Harvard On-Line Library Information System (HOLLIS). CIU also provides protocol conversion service to allow asynchronous terminals and other devices to access applications designed for the IBM 3270 environment¹⁵.

5.1.1.3 Harvard Business School Network

The Harvard Business School operates a network linking approximately 1800 users to its mainframe systems located in Baker Library. Terminals and PCs running terminal emulation programs access either of the Business School's two mainframes — a DEC 1091 and an IBM 4381 — via an IDX 3000 Data Switch. All devices are directly connected to the data switch using multiplexers distributed throughout the campus and operate asynchronously at 9600 bps¹⁶.

5.1.1.4 FASNET

The Faculty of Arts and Sciences Network (FASNET) is a broadband, coaxial cable network serving a number of faculty and administration buildings as well as the computing resources at 1730 Cambridge Street (OIT), the Science Center, and the Aiken Computational Laboratory.

The primary network service provided on FASNET is Sytek's LocalNet 20, an asynchronous terminal-to-host application operating at 9600 bps. Several hundred ports provide connections among terminals, PCs, and 30 host computers throughout the served area.

Another network service implemented on FASNET is the IBM PCNet. PCNet is a 2 Mbps local area network (LAN) designed to connect IBM PCs for communications and

¹⁵ *Ibid.*

¹⁶ *Ibid.*, p. III-2.

resource sharing (printers, disks, file, *etc.*). Several IBM PCs in administration buildings are currently connected via such a LAN¹⁷.

5.1.1.5 Ethernets

Ethernet is a high speed LAN designed to support the exchange of data among devices within a limited geographical area. It is based on coaxial cable technology and is primarily configured to operate at a rate of 10 Mbps. At Harvard, Ethernets are primarily employed in computer-intensive environments such as computer rooms and research laboratory areas.

In early 1986, the FAS implemented an Ethernet using fiber optic cable. This 10 Mbps network interconnects computer networks (including other Ethernets) in over a dozen buildings providing high speed file sharing and image transfer capabilities. It is also the primary network providing access to the external supercomputer network via a DEC VAX 11/750 located in the Aiken Computational Laboratory. Since it was designed as a transparent transport facility, the FAS Fiber Optic Ethernet supports a variety of network protocols such as DECNET, TCP/IP, and XNS¹⁸.

5.1.2 External Networks

- ARPANET
- CSNET
- BITNET
- USENET
- Supercomputer Networks

¹⁷ *Ibid.*

¹⁸ *Ibid.*

5.2 The Office of Information Technology

The Office of Information Technology (OIT) is in the process of implementing a plan to create a university-wide communications network incorporating voice, data, and imaging. OIT is charged with the operation of telecommunication services for the university. OIT has historically operated and maintained only large mainframe computing facilities and provided phone line access to their machines.

6 Comparative Evaluation on Criteria

In evaluating the networks it is necessary to examine the impact on all three segments of the user community — the faculty, the administration, and the students. The evaluation will be based on personal interviews with key network planning personnel supplemented by questionnaire data.

The perspective throughout this analysis will be the examination of how protocol standardization (or non-standardization) contributes to the service delivery failure. The critical issue is that of network connectivity.

Data for the evaluation was obtained through personal interviews and survey questionnaires. The author also draws conclusions from data obtained by Harvard University pursuant to designing their University Network. Twenty-five questionnaire responses from MIT users form the complement to the Harvard data. The principal network managers for both universities were interviewed, in addition to lead users in all three segments of both environments.

6.1 General User Characteristics

Before considering the several evaluation criteria with regard to the three segments of the user community, it is worthwhile to make some general remarks about the use characteristics of these segments.

6.1.1 Administration

Administrative users utilize data processing to manage resources and facilitate processing of paperwork. They are typically heavy computer users, averaging 5 to 6 hours a day on a computer or terminal. Their first priority is database access. The questionnaire data gathered did not qualify administrative users' database needs as either on-line or batch. Some administrators may be satisfied with infrequent batch report generation. Most

administrators agreed, however, that access to central administration databases was needed to provide more timely information, eliminate duplication of data entry, and reduce the amount of paper that flows between offices. Three classes of databases were identified:

- **Financial:** budgets, Accounts Payable/Accounts Receivable, purchasing
- **Human resources:** payroll, personnel, appointments
- **Physical:** facilities management

In addition to database access, electronic mail, file transfer, and resource sharing are also high priorities. As a result of their heavy use, administrative staff are experienced and, once trained, they are quite capable in their work. They tend, however, to be relatively unsophisticated from a technical standpoint. Administrative users use data communications primarily within their physical local office and often have technical or consulting support within their office.

6.1.2 Faculty

Faculty members make use of computer resources in order to advance their research using library database access for on-line cataloguing and on-line circulation (OCLC). Databases accessed may reside both within the university and outside it as well. Electronic mail can keep a faculty member in touch with colleagues at other universities or research institutions. The ability to transfer and to exchange editable word processing documents between collaborators and publishers also becomes a priority. For electronic publishing, it is important to be able to circulate image as well as text.

Faculty members are infrequent computer users. A simple, easy to use (and remember) interface is very attractive to them. They generally require more consulting and troubleshooting assistance than administrative users. Their communications are largely

geographically centered around their department or school. Local area networks are often able to meet all the needs of faculty users. Often the department or school maintains some consulting resources of their own to meet their members needs.

6.1.3 Students

Students use computing facilities primarily in support of their coursework as well as for word processing for papers and thesis. Since students are expected to complete their own work, document interchange is not an important service. Resource sharing (printers, file servers, *etc.*) and file transfer are more important. Students need to be able to gain access to timesharing hosts from remote workstations or terminals as well as use electronic mail to exchange information with classmates and course administration.

Student users are scattered throughout the campus. Some are very sophisticated users that demand advanced functionality and are able to educate themselves quickly regarding difficult and complicated interfaces. Others are computer novices that have questions about nearly everything. The diversity of the population results in varying usage from light (less than 1 hour/day) to heavy (5 to 6 hours/day).

6.2 Functionality

The several user service requirements will here be considered one at a time. When segments of the user community provide differing evaluations, the distinction is noted.

6.2.1 Database Access

Database access is most important to administrative users. Offices like the Registrar, Budget, and University Administration depend on timely access to internal databases to perform their jobs. Security issues, however, prevent most of these offices from joining the mainstream network.

At MIT, databases are maintained on dedicated administration machines on closed local area networks. They cannot be accessed from the Campus Network. As a result, databases are maintained independently. Users requiring information from another system must gain access through IBM 3270 terminal access on an as needed basis.

MIT's administrative users gave the data network slightly favorable marks on its support of access to central databases. The principal complaints pertain to the awkwardness of transfer methods forced by security difficulties. The Medical Department has adopted the method of having 2400 foot magnetic tapes created and physically transferred to gain access to the data it needs.

Student and personnel information is updated on a daily basis by the registrar and personnel offices. Without on-line access, the process of creating the tape, transporting it, and extracting the information can take two to three days. The process is time consuming and Knott updates her data only as often as she has to (approximately twice a month). Since this is neither timely nor cost efficient, Alison Knott, Manager for Information Systems for the Medical Department is desperately seeking an alternative¹⁹.

Harvard has approached the problem similarly, dedicating IBM hosts to maintaining the administrative databases. Access from remote hosts is on an ad hoc basis. On the FAS Ethernet, database access is well-supported by NFS. However, the FAS Ethernet is populated with non-NFS hosts that cannot take advantage of the Network File System.

Some faculty users rely on internal and external databases for research information. In both institutions, this service receives slightly favorable ratings. Local area networks facilitate access to data internal to the department or research area, like MIT's Plasma Fusion Laboratory and Harvard's Aiken Computational Laboratory.

6.2.2 Resource Sharing

¹⁹ Personal interview with Alison Knott, Manager of Information Systems, Medical Department on April 29, 1987.

This service is generally served quite adequately by both the Harvard and MIT data networks. For all classes of users, resource sharing is primarily at the local area network level. Since each LAN is configured to serve the local users, it is pretty effective in delivering the necessary service.

MIT's administrative users gave the highest marks; it was evident that the most attention was paid to resource sharing at the departmental level. The Medical Department and Administrative Systems have invested heavily in their local environments (NOVELL, print servers, file servers) and have achieved effective support of their local groups.

The Medical Department elected to install its own computing and network facilities in support of its users. The use of the NOVELL network operating system created the opportunity to provide for each user a low cost workstation with free access to word processing (Microsoft Word), database management (dBase), central file and data, and printers. A user is not constrained by the failure of his single workstation; he can simply continue work on another and access the same resources.

Faculty members were quite variable, depending on the resources and sophistication of the department. A professor at the Sloan School finds himself isolated without any shared resources whereas a physicist in the Center for Space Research gives highest marks to his ability to share resources like printers and database servers. Students gave lower but satisfied marks, but may have been unaware of the underlying resource sharing of Project Athena.

6.2.3 Document Interchange

This service is required by administration and faculty users. The ability to exchange modifiable word processing documents greatly facilitates research and work for both sets of users. This service is possible to a great extent due to a standardization in word processing packages. In environments where a single package dominates, document

interchange is easy. Other environments are populated by a wide variety of word processing packages which exacerbate the problem.

Within MIT's Administrative Systems, incompatibility between DECMate WPS format and the various IBM PC word processing package formats (Word Perfect, Multimate, *etc.*) create difficulty in exchanging documents. Users have to resort to exchanging plain text documents to circumvent the difficulty. While this does result in the elimination of repetitive re-typing, a true standard for document interchange would be a decided win.

Attempts have been made to establish a standard. The Microcomputer Center's consultants encourage users to acquire Microsoft Word, since it provides excellent functionality and supports document exchange between the IBM PC and Apple Macintosh versions. Nonetheless users are reluctant to learn a new editor and word processing package once they have invested considerable time and money in another package. Moreover, users that do not avail themselves of the advice of consultants will not receive guidance and will make their own decisions. No formal standard is in place.

The appropriate agency to effect a document interchange standard is Telecommunications Systems. This office should officially endorse a standard document format (like Microsoft Word or Document Interchange Format) and offer an internetwork document transfer utility to user of the Campus Network. This would create an incentive for users to adhere to the standard and would greatly improve network service to users.

6.2.4 File Transfer

File transfer within a LAN is well-supported. Transfer across the network depends critically on protocol compatibility.

In the MIT environment, any file transfer across the Campus Network must use TCP/IP. Networks running TCP/IP have no difficulty with this. LANs that have selected other network protocols however are left stranded without backbone support. These

isolated networks have installed dedicated lines or modem connections for ad hoc file transfer capability. The Medical Department has resorted to transferring data by physically transporting magnetic tapes, because the bandwidth of terminal-to-host transfer is insufficient for their needs.

The Harvard environment consists of a number of disjoint islands of networking. Transfers between networks are often impossible. Even within the FAS Ethernet, TCP/IP and DECNET hosts are unable to transfer files even though they are physically connected.

All classes of users view file transfer as a priority. Since most transfers are transacted within a local network, the service level seen by users is mostly adequate to their needs. Only users desiring transfers from protocol-incompatible external networks observe difficulties.

6.2.5 Image Communications

The absence of image document standards is a great obstacle to effective image communications. The ability to exchange image documents was universally rated quite poor. No general format has reached anywhere near the stature of an effective standard even within local environments. Macintosh PICT resources are commonly exchanged but only among Macintosh users. With more faculty and student users demanding the capabilities of electronic publishing, the need for image communications is viewed to be a significant growth area in the next five years.

6.2.6 Electronic Mail

Unlike some of the above services which are adequate when supported only at the local level, electronic mail is greatly desired across the entire university network. This poses a great challenge in protocol compatibility and conversion. All classes of users use electronic mail to broadcast organizational and coordinating information.

At MIT, Telecommunication Systems attempts to arrive at solutions that will solve the compatibility problems for each non-TCP/IP subnetwork. In a number of cases, a host is identified that can act as a mail gateway for its sub-network. This host will convert mail messages received via TCP/IP and will distribute messages to users/hosts on its own sub-network.

Given the responses, users that are a part of the Campus Network are very pleased with their ability to receive and transmit electronic mail. Isolated networks are also satisfied with electronic mail support.

The system manager at Harvard's Aiken Computational Laboratory has attempted to resolve electronic mail problems by distributing a standard set of mail protocols for all Harvard hosts. This standardization effort has met with some success and has greatly facilitated the reception and transmission of electronic mail.

6.2.7 Terminal-Host Communications

Terminal-to-Host communications is still a critical service required by all classes of users. Part of this importance stems from connectivity problems described above. Administrative users both at MIT and Harvard rely on remote login access to timeshare hosts for database access as well as processing needs.

Again protocol compatibility determines the domain of hosts to which a user can gain access. At MIT, Project Athena uses TCP/IP, giving student users access to nearly any coursework-related host on the Campus Network. Dedicated lines and local area networks serve faculty and administrative users. The dial-in services provided by Centrex are also used to support low speed interactive login sessions. Service is generally adequate since any acute need has been met by the implementation of an *ad hoc* solution.

Harvard also relies quite heavily on Centrex to support administrative users. The FASNET supports remote login service for faculty and college administration. Harvard is

still very much constrained by the absence of a university-wide network facility that would connect the various "islands" of networking.

6.3 Network Reach — Connectivity

Two factors interfere with the network's ability to reach all users desiring data communications service: difficulties in establishing a physical connection and protocol compatibility problems.

6.3.1 Physical Connection

In both the Harvard and MIT environments, users that desire access are denied it for physical connection issues. Harvard's lack of a university-wide spine makes it impossible for users across the river at Harvard Business School to gain high speed access to the rest of the data networks. The only links now supported are through dedicated phone lines connected to the OIT administrative hosts.

At MIT, the physical connection issue is slightly different. There exist users in buildings that make it nearly impossible to wire for Campus Network access. In addition, the incremental wiring costs for the first user in a building preclude some users from gaining access. The costs for adding a building to the Campus Network run on the order of \$50,000. These must be assumed by the first user desiring service. The second subscriber may be assessed charges as low as \$5,000 once the building is already on the Campus Network.

In buildings not yet wired for the Campus Network, departments not willing to spend \$50K play a waiting game for someone else to "take the plunge." If no group has the willingness and the resources to do so, then all groups will be permanently isolated from the Campus Network.

Jeff Schiller, Networking Director for Telecommunications Systems, describes two examples where coalitions of departments have banded together to share the initial costs of wiring the building. This helps spread the initial wiring burden among a number of departments and helps alleviate the problem. It is necessary to have several groups all simultaneously desiring and prepared to "be networked."

It should be noted that in the two example Schiller cited, three departments were involved and worked together on an ongoing basis. The decision to share networking costs was a natural extension of an existing spirit of cooperation.

In buildings occupied by many more small and unrelated departments, cooperation may be more difficult to achieve. If the preparation and network requirements of the groups varies greatly, then the "novice" users have an incentive to withhold their support and save money, in the hopes that the remaining groups will install it anyway (due to their greater motivation).

Still, there may be a role for Telecommunications Systems to play in arranging and facilitating these coalitions. It would improve the physical connectivity and improve the reach of the Campus Network.

6.3.2 Protocol Incompatibility

Harvard has not yet faced some of the more difficult issues in protocol incompatibility. The fiber optic FAS Ethernet avoids protocol difficulties by using fiber star bridges that transparently join Ethernets, creating one logical network. TCP/IP, NFS, DECNET, and XNS hosts all communicate over the same Ethernet without interfering with one another (except in degrading performance, see below). Hosts running incompatible protocols still have difficulty talking to one another. Nonetheless, users desiring access can be easily connected to the network to communicate with other hosts running the same protocols.

Because MIT has adopted a protocol standard for Campus Network backbone communications, compatibility does become an issue. Alison Knott, manager of Information Systems for the Medical Department, would like to join the Campus Network but does not want to sacrifice the functionality of NOVELL just to obtain TCP/IP compatibility. It would be possible to invest in a PRONET to TCP/IP gateway to rectify the problem, but the investment (about \$25,000) is beyond the department's resources.

The problems MIT is facing and that Harvard will soon face point up the acute need for inter-networking protocol standards to alleviate the obstacles to an integrated university-wide network.

6.4 Network Performance and Reliability

Network performance is generally very good in both environments. At MIT, only 10% of the available bandwidth of the campus backbone is used even at instantaneous peak load. Bandwidth is not a problem. Faculty and administrative users report satisfaction with network performance. Some students perceive poor response for remote login, but interviews reveal that the true source is a bottleneck at the timeshare host. Reliability is generally good, though somewhat sensitive to power surges and drops. Telecommunications Systems has taken steps to correct this problem by placing both the network bootstrap host and the Kerberos authentication server on uninterruptable power supplies.

At Harvard, the HBS network is adequate to its own needs. The FAS Ethernet presents a different problem. FAS Ethernet is now quite large with a number of diskless SUN workstations accessing filesystems through NFS, Xerox workstations, and DEC VAXs. This creates a problem since Ethernet is a broadcast medium. Performance falls off rapidly as the Ethernet approaches saturation. Hosts are being inundated with broadcast traffic. Network performance suffers as a result.

A Ethernet LAN bridge helps partition a single logical Ethernet by only allowing packets addressed to the far side of the bridge. Broadcast traffic of course must go through. The FAS Ethernet may have to be split into distinct networks to alleviate the problem.

6.5 Network Control

Cost is a major source of concern. Users regard the installation and operating costs as prohibitively high. Three years ago, MIT Medical Department spent over \$300,000 for timeshare support through dedicated lines and for operations. The manager had no choice but to opt for their own dedicated local area network. The department's operating costs are now \$178,000, less than half what her costs would be today. She has in place now computing resources that deliver an order of magnitude better performance and functionality at a fraction the cost. Now, initial wiring (\$20,000), gateway (\$25,000) and monthly operating costs total over a quarter of her operating budget. The price of connectivity is currently beyond her means, especially in light of her discomfort with the security of the Campus Network (see below).

At Harvard, the opposite situation prevails. OIT charges departments for their usage of systems they manage. The principal data network segment, however, is managed by the Faculty of Arts and Sciences and the managers have not established a charge back system. Users are not assuming their share of the costs.

Network management is difficult in both environments, due to the proliferation of Ethernet. Ethernet is a passive medium and is difficult to partition. With some Ethernet transceivers (like 3Com), it is necessary to interrupt service in order to change the topology. Adding a 3Com transceiver requires breaking the cable and plugging each of two ends on either side of the transceiver. Invasive or "vampire" transceivers connect by penetrating the insulation of the coaxial cable to make contact.

Invasive taps promote flexibility, since the cable need not be cut into many different lengths in anticipation of expansion. They can also damage the cable. In-line transceivers (3Com) make more solid connections but interrupt the cable for installation. The best solution is the use of a multi-port Ethernet transceiver (like DEC's DELNI). This is a product that can connect up to eight hosts from a single interruption in the cable. Traffic monitoring is very simple on Ethernet and can be conducted from any node of the network. Diagnosis of coaxial Ethernets is a fairly simple task.

Fiber-optic cable is very difficult to work with, due to its fragility. Diagnosis of fiber problems require special equipment that Harvard and MIT have only recently acquired. Prior to their acquisition, both FAS and Telecommunications Systems managers were plagued by debugging headaches.

6.6 Network Support — Maintainability

Without exception, this is the area where all users in both environments desired the most improvement. MIT has more resources in place to support the user community. Telecommunications Systems maintains consulting support for administrative and faculty users on a fee basis. All users (including students) can receive systems advice from the Microcomputer Store on the selection and installation of microcomputer systems. Project Athena maintains a staff of over 40 consultants to answer student questions free of charge. Consultants staff high activity workstation clusters during peak periods as well as maintaining a user "hot line" for on-line inquiries.

Harvard's OIT attempts to meet the needs of faculty and administration users, but comes short of user expectations. The Faculty of Arts and Sciences maintains user support consultants (terminal watchers) at the Science Center to answer student questions. The Technology Products Center serves the same role as MIT's Microcomputer Store for Harvard University.

Users all would like better information dissemination regarding network news as well as future plans. Sub-network managers are left to their own devices to plan and troubleshoot their LANs. They desire greater technical support and end-user training.

6.7 Security

Network data security is a critical issue for administrative users. Sensitive information must be protected from even the most determined efforts of mischievous students. Security directly impacts on the network reach issue in that even though a physical connection can be established and even if a protocol standard (or protocol conversion) is created, administrative users will not open their networks unless their security requirements are satisfied.

This issue is particularly difficult on broadcast medium like the Ethernet. Every host on the Ethernet "sees" each packet, that is each bundle of information transmitted on the Ethernet cable can be disassembled and read by every host. This problem can be solved in part by not sending sensitive information as plain text, employing a data encryption scheme. But this solution requires a sophisticated system of "key" distribution for encoding and decoding encrypted packets.

MIT's Project Athena has made significant progress in this area through the development of the Kerberos authentication server. Kerberos distributes "tickets" to authenticate communications connections for authorized users. Kerberos verification is necessary to establish user-to-host interaction as well as host-to-host and host-to-server sessions.

Kerberos authentication operates for all Internet (TCP/IP) communications. Any client (user or host) requesting a connection triggers a request of the Kerberos authentication server to obtain tickets to verify access to the desired service. Without appropriate Kerberos tickets the target host will deny access. Project Athena hopes to

distribute freely the Kerberos authentication scheme to all MIT network managers as soon it is satisfied with the system.

Harvard trails MIT in the security issue. Password protection is in effect on all timeshare systems, of course, but OIT has yet to attempt to address the problem of data network security.

6.8 Planning

Network planning is more an organizational problem than a technical one. The decentralized nature of the acquisition and installation of systems exacerbates the situation. Sub-network managers would like assistance and information regarding future expansion and technology but are unable to obtain it from central planning. In order for network planning to be effective, it is critically important that the IS planner involve the sub-network managers and users in the process. As the results have shown, the effectiveness of the data network depends not only on decisions made by the central office but also on decisions made by other stakeholders as well.

MIT's Telecommunication Systems is largely reacting to requests. The network topology evolves as requests incrementally add nodes to the network. Telecommunications Systems is woefully understaffed to meet any planning needs. The director of networking for Telecommunications Systems splits his time between Project Athena and the Campus Network, which fully occupies his time. Staff members at the Laboratory for Computer Science have undertaken independent surveys of users' needs in order to develop a strategic plan for voice, data, and video networks. Telecommunications Systems has not formally endorsed the project and results are still pending.

Harvard's OIT is currently undertaking a major effort to install a university-wide network. They have faced the reality of organizational difficulties and have created a steering committee of the major stakeholders in order to improve the quality of the network

design as well as to facilitate its implementation. OIT has hired an outside consultant to assist in the design process of an integrated voice, data, and video network. The design process consists of six phases:

1. Research Plan — defines the data requirements, data collection methodology, and data analysis methodology for the network design
2. Needs Assessment — user needs are identified using information gathered from 120 interviews and 300 questionnaires
3. Resource Summary — describes the existing network facilities for voice, data, and video
4. Network Architecture Recommendation — identifies the optimal network design, considering traffic analysis, functional requirements, and existing resources
5. Request for Proposal — describes the specific implementation in sufficient detail for vendors to bid on the project
6. Implementation

OIT has circulated their Request for Proposal and is reviewing vendor proposals. The research findings and network architecture recommendation have been incorporated into the analysis in the next chapter.

Harvard's methodology for defining the design as well as managing the organizational issues is quite commendable and may serve as an example for future IS planners.

6.9 Evaluation Summary

The following table summarizes the results of the author's research.

	MIT			Harvard		
	Students	Faculty	Admin	Students	Faculty	Admin
Functionality	+	o	+	o	o	-
Network reach	+	o	o	+	o	-
Network reliability & performance	-	o	+	o	o	o
Network control	o	o	-	o	o	o
Network support	+	-	-	o	-	-
Security	o	o	-	o	o	-
Planning	+	-	-	o	+	+

+ Superior
o Average
- Inferior

7 Network Backbone — The Choice of a Protocol Standard

Given the diversity of the individual networks and user requirements in the university environment, what is the optimal approach to linking them all together? Which protocol architecture will provide the most interoperable internetwork environment? Examining the answers planners have found for Harvard and MIT may help answer these questions.

MIT's Campus Network consists of a central backbone linking the client sub-networks around the Institute. Harvard's planned University-wide network has adopted an identical architecture. TCP/IP is the protocol standard for MIT's backbone communications. Harvard's Request for Proposal recommends that TCP/IP be the protocol implemented over their High Speed Data Network backbone. In view of the manifold problems with protocol incompatibility, the selection of a protocol standard is a critically important decision for network planning.

7.1 ISO Internetworking — Implications for the Backbone

Internetworking is communications among an interconnected set of networks. An interoperable internetwork is one that provides services to heterogeneous hosts on different subnetworks. The International Standards Organization's goal is to provide protocol standards that will support a homogeneous set of services across heterogeneous hosts and subnetworks.

Network designers have investigated and implemented a number of interconnection strategies that attempt to facilitate communications among computers and terminals connected to different networks. The selection of an optimal strategy depends on the characteristics of the networks to be connected. One critical characteristic regards the nature of the interactions between network clients — either connection-based or connectionless.

A network is connection-based if interactions are primarily point-to-point with some duration. A session is initiated by an application establishing a connection with a remote application. Once established, the applications can freely exchange data. When complete, the connection is released. This paradigm is also referred to as a virtual circuit.

The classic example is the voice telephone network, which is operated by human users who establish connections (call), transfer data (talk), and release connections (hang up). In a connection-based network, applications (like bulk file transfer or remote login) establish connections, transfer data and when completed, release the connection.

A connectionless network, on the other hand, does not establish or maintain any relationship between individual data transfers. All of the addressing and other information needed to convey data from source to destination is included explicitly in each data unit. Broadcast communications, periodic data sampling, and other request/response applications (such as directory and identification services) in which a single request is followed by a single response, benefit from connectionless interaction²⁰. Network designers also refer to this as a datagram paradigm.

Piscitello finds two fundamental strategies to be most applicable to internetworking in an OSI network:

- Hop-by-hop enhancement
- Internetwork protocol (which Piscitello refers to as connection-less internetworking)

The determination of the preferred strategy depends on the characteristics of the subnetworks that are to be connected.

7.1.1 Hop-By-Hop Enhancement

This strategy is preferred if the networks to be connected:

²⁰ David M. Piscitello *et al.*, "Internetworking in an OSI environment," *Data Communications*, May 1986, pp. 120-121.

- Offer predominantly connection-oriented services
- Exist where close cooperation among the network administrators can be achieved and enforced
- Exist where the extent to which the individual network services differ is limited

With this approach, connection-oriented internetworking may be achieved by relaying the services of one network directly onto corresponding services of other networks. An underlying assumption of this network interconnection is that it is easier to solve when the services that the subnetworks offer are the same than when they are different.

All subnetworks that are to be interconnected must provide exactly the OSI Network Layer service. Any subnetwork that does not provide this service must be enhanced or modified to do so. Relays are used to passively map the connection establishment, data transfer, and connection release utilities of one subnetwork onto another whenever network connection cross subnetwork boundaries.

Consider a host that desires a connection with a host residing on another subnetwork. If the "calling" host's subnetwork already supports the OSI Network Layer service, then his packets are relayed through a gateway mapping the requested service to the adjacent subnetwork's Network Layer service. If the "receiving" host resides on the adjacent subnetwork, then the trip takes only one hop. [See Figure 1]

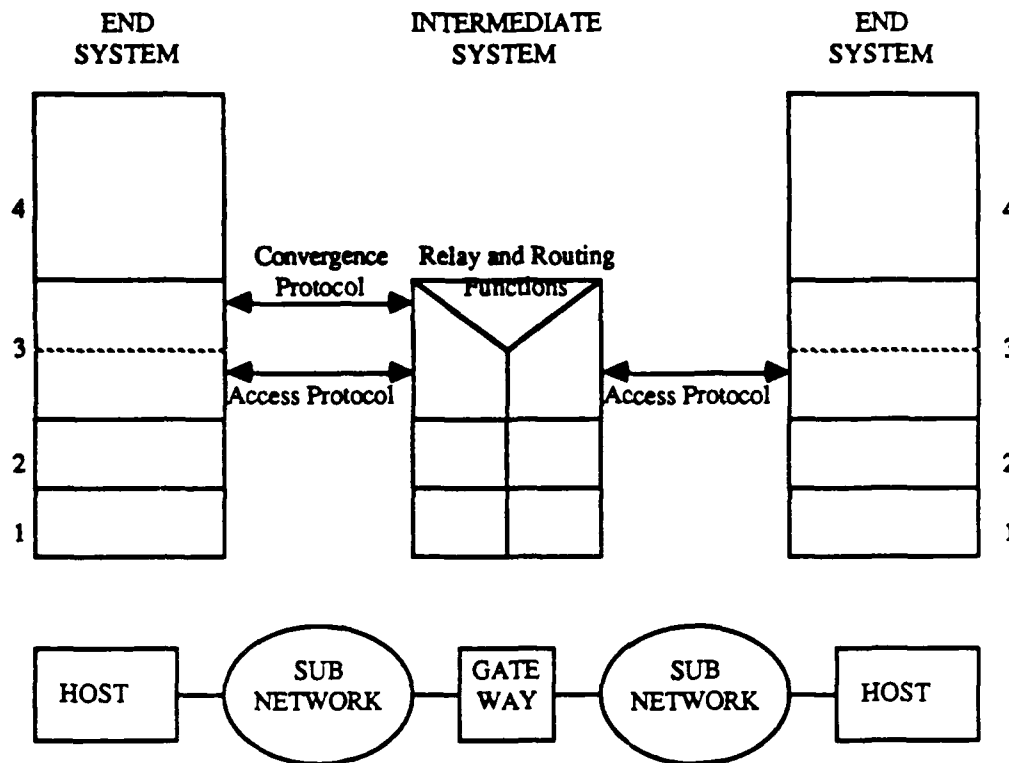


Figure 1: Hop-by-Hop Enhancement

Source: Piscitello *et al*, *Data Communications*, May 1986, p. 122.

Now if the "calling" host's subnetwork does not provide an OSI Network Layer (perhaps only a subset), then that subnetwork protocol must be enhanced to provide the interfaces and functionality required of a network service. That is the function of a convergence protocol, shown in Figure 1 above. This hop has to be enhanced in order to support internetwork communications.

In this strategy, gateways perform a mapping of the service offered by one network onto another. In general, the gateways do not add services. Rather, they perform the relaying and switching functions necessary to bind the individual subnetworks into a unified or global network. A consequence of this approach is that either all of the subnetworks must inherently provide equivalent services or each must be enhanced to some common level of service.

The enhancement of subnetworks up to OSI network service may be accomplished either by direct modification of the subnetwork protocol or through the use of a subnetwork dependent convergence protocol (SNDTCP). An SNDTCP operates on top of a subnet access protocol to provide the elements of the OSI network service that are missing from the access protocol²¹.

7.1.2 Internet Protocol

This strategy is preferred if the networks to be connected:

- Offer predominantly connectionless services or a mix of connectionless and connection-based service
- Exist where network administrators are largely autonomous
- Exist where the extent to which the individual network services differ cannot be predicted or controlled

It differs from the hop-by-hop approach in that instead of creating a pairwise protocol map for each gateway, a single explicit standard Internet Protocol (henceforth ISO IP) is used for all end-end communications.

Since the ISO IP is a Network Layer service, it performs the addressing and routing functions necessary for end-to-end communications. Because this protocol set adheres to the ISO OSI model, the protocol will function regardless of what the underlying data link layer is. The ISO IP could be layered over Ethernet, IEEE 802.5 Token Ring, X.25 Public Data Networks, or even twisted pair. ISO IP makes minimal assumptions about the services available — only those specified in the interface between the network and data link layers.

²¹ *Ibid.*

Using this approach, a host wanting to broadcast a message over the internetwork simply uses the ISO IP to provide network service and takes care of routing its message over the internetwork.

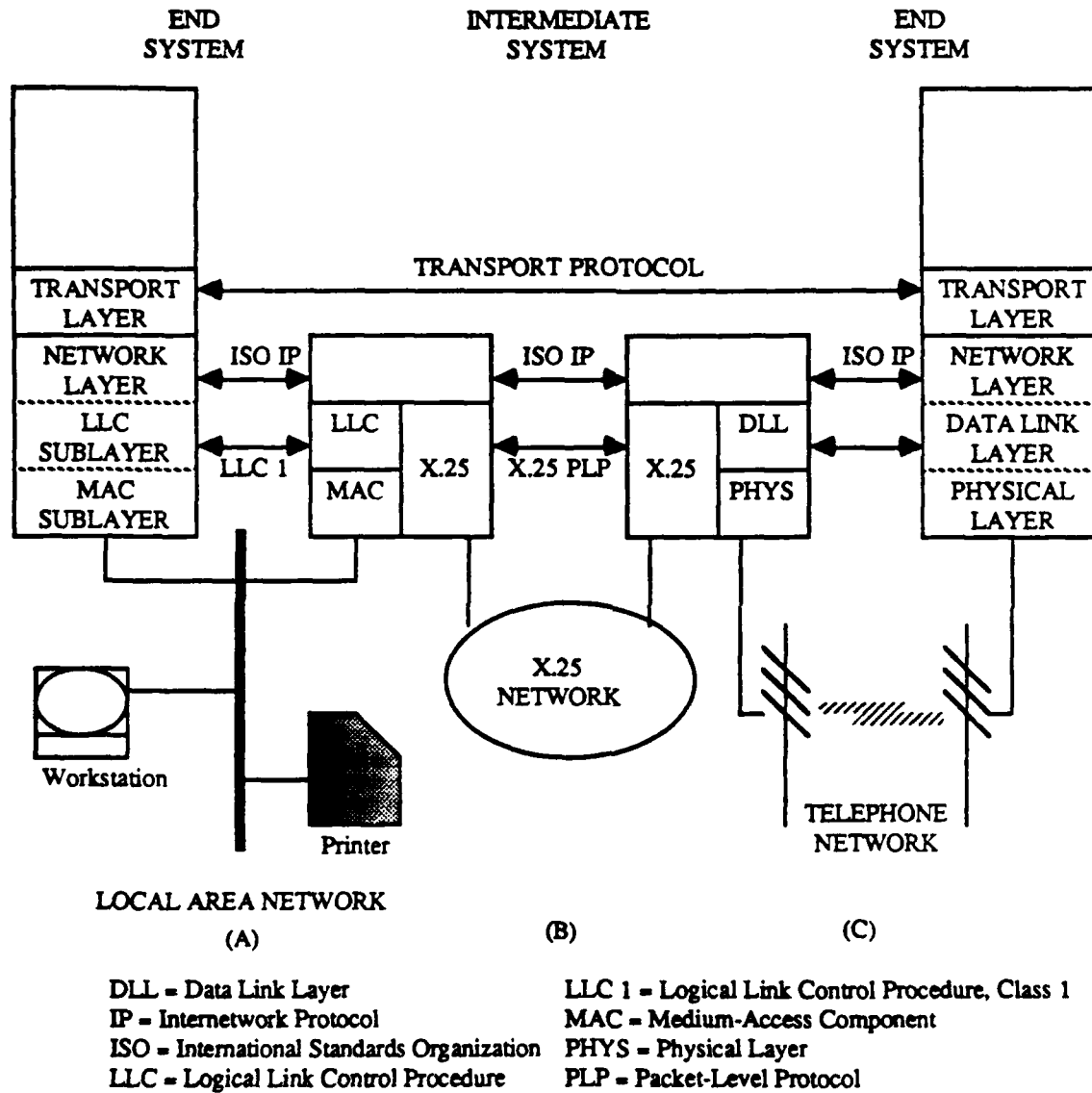


Figure 2: Internetworking Protocol

Source: Piscitello et al, *Data Communications*, May 1986, p. 125.

Since the ISO IP is connectionless, Internetworking Protocol Data Units (IPDU) form the basic packet of information. In order to create a virtual circuit, support from the Transport Layer is necessary. The Transport Layer protocol would take care of guaranteeing arrival, sequencing the IPDUs, so that they might be interpreted as a continuous flow of information.

The important point here is that even if a "connection" has been established at the Transport Layer, the IPDUs conveying information might be routed independently. The transport layer service assembles and sorts the IPDUs to present a continuous connection-based data stream to the end hosts.

The underlying subnetworks should provide only a data transmission service. No subnetwork enhancement is necessary; an ISO IP can be operated directly over the Data Link Layer.

It is important to note that neither of these approaches interferes with subnetwork-specific operations. In the hop-by-hop approach, network service local to the subnetwork is conducted business as usual. For a DECNET, Transport Protocol (TP) messages that leave the local net are mapped into an equivalent Network Layer protocol for the target network. TP messages that stay local are unchanged.

ISO IP is just a complement to the subnetwork-specific network service (if it exists). DECNET might continue to offer TP support in addition to adding an ISO IP service. That way internetwork applications would use ISO IP, local ones could continue to use TP.

7.2 Network Requirements

Before evaluating the strategies elected by the two universities, the author first verifies that the two shared the same goals and technical requirements.

7.2.1 Harvard University

Harvard University has circulated a Request for Proposal for their University Network. They have specified a network architecture that calls for a High Speed Data Network (HSDN) backbone connecting the several access subnetworks around the university.

Harvard's Request for Proposal details the requirements for the High Speed Data Network²². The HSDN will become the primary information transport for the University, linking major schools, departments, building clusters, and individual buildings.

Of primary importance to Harvard is the technical adherence of the network and gateways to the principles set forth in the ISO recommendation for Open System Interconnection (OSI) as well as those of the IEEE 802 committees and proceedings as adopted to date. Harvard must be well positioned to move forward with implementations of systems based on the ISO/OSI reference model when they become available²³.

Functional support. Primary applications to be served fall into the following broad categories: message transfer and/or electronic mail (X.400); bulk file transfer to and from shared file servers and host resources; remote host log-in ; distributed data base; and high volume image data transfer.

Technical Requirements. The target medium will be fiber optic cable at a minimum data rate of 10 Mbps. Gateways to the HSDN must include the hardware and software necessary to interface the HSDN with existing data networks. The gateway devices must be capable of isolating local traffic from the backbone network and providing routing information to local network users. The HSDN must connect the following internal Harvard campus networks:

²² "Request for Proposal for an Integrated Telecommunications Network for Harvard University," Harvard University Office for Information Technology, Telecommunications Services Division, January 1987, Version 2, pp. 46-48.

²³ *Ibid.*, pp. 46-47.

- Ethernet (TCP/IP, DECNET, XNS, LAT)
- Star LAN 802.3
- Token Ring IEEE 802.5
- PBX — ISDN
- Broadband (Sytek — LocalNet 20, IBM PC Net, Ethernet)
- IBM Bisync and SDLC SNA
- Appletalk
- IDX 3000 Data PBX (T1)

As well as the following external network gateways:

- ARPANET (TCP/IP)
- BITNET (BSC)
- UUCP
- Public Packet Networks (X.25)
- Supercomputer Net (T1)

The Harvard University environment is characterized by primarily connection-oriented transactions as evidenced by the list of functions above. Furthermore, subnetwork administration is very autonomous. Subnetworks are managed by different departments and offices as well as by different schools possessing near complete independence. Services vary significantly from subnetwork to subnetwork.

7.2.2 MIT

MIT's network requirements are nearly identical to Harvard's. MIT's Campus Network backbone is a 10 Mbps PROTEON token ring linking the building Ethernets scattered across the Institute. The functions to be supported are the same. Both are

implemented over high speed fiber optic cable. MIT's internal access requirements are not as demanding:

- Ethernet (TCP/IP, DECNET, XNS, LAT)
- Star LAN 802.3
- Token Ring IEEE 802.5
- PBX — ISDN
- Broadband (Ethernet)
- IBM Bisync
- Appletalk

The external network gateway requirements are identical. MIT's communications are dominated by connection-oriented applications, though the services supported on the access subnetworks are more alike than at Harvard. Again, network administration is highly decentralized with subnetwork managers having total control over their own resources.

7.3 Selection of TCP/IP

A summary of the network requirements below in Figure 3 clearly indicates that the two universities had nearly identical goals and technical requirements. Both universities standardized the backbone protocol by selecting TCP/IP.

The reasons that TCP/IP was selected are the same for both universities. Since MIT Campus Network has evolved much more than Harvard's the fact that there existed a significant installed base of TCP/IP hosts weighed more heavily in MIT's decision. Furthermore, at the time there existed no practical alternative that offered the same interoperability and flexibility in hardware support. TCP/IP is available on the largest number of vendors' equipment. Finally, the importance of the Defense Data Network

(ARPANET and MILNET) to MIT's research work and communication allowed no other decision.

	HARVARD	MIT
Type of Service	Connection-based	Connection-based
Functional Requirements	Database access Resource sharing Document/File transfer Image communications Electronic mail	Document/File Transfer Electronic Mail Database Access Resource Sharing
Subnetworks	Ethernet, Token Ring PBX, Broadband, IBM Bisync	Ethernet, Token Ring PBX, Broadband, IBM
Bisync	Appletalk, SNA	Appletalk
Administration	Autonomous	Autonomous
Backbone Medium	Fiber optic cable	Fiber optic cable
Protocol Selected	TCP/IP	TCP/IP
Reasons	Availability Interoperability Pre-installed base Importance of DDN	Pre-installed base Importance of DDN Interoperability Availability
Implementation	Multiple network layer protocols	Single network layer protocol
Implication	Requires convergence at gateway	Requires conversion at host
Effective Strategy	Hop-by-Hop Enhancement	Internetworking Protocol

Figure 3: Summary of Internetworking Strategy

TCP/IP is the protocol recommended in Harvard's RFP. "Because of its emergence as a *de facto* standard in educational and research networking, the TCP/IP suite of protocols is preferred for the HSDN.²⁴" TCP/IP offers the greatest interoperability of

²⁴ RFP for Harvard University, p. 47.

any existing protocol suite on the market. For a university with a number of different vendors, this issue is of paramount importance.

The TCP/IP protocols are not without their drawbacks as well. They do not conform to the ISO OSI reference model. The Internet IMP-IMP (Interface Message Processor) occupies both the data link and the network layers and the Source to destination IMP protocol overlaps the network and transport layers²⁵.

Since there does not exist a distinct OSI network layer, the migration path from TCP/IP to ISO IP may require significant modification of user internetwork applications²⁶.

Furthermore, the basic utilities supported with the Internet Protocols provide a subset of the capabilities supported by access subnetwork-specific protocols. FTP, TELNET, and SMTP do supply most of the functionality required in both environments, but in convergence mapping they represent bottlenecks. An example of this is described in Section 7.4.1 below.

7.3.1 Protocol Implementation

The two universities diverge on their implementation of the protocol. Harvard has elected to minimize the impact on existing networks by permitting the access subnetworks to continue to use the same network layer protocols for internetwork communication. This places the burden of standardization on the gateway hosts. They are collectively responsible for converting the various subnetwork to the backbone standard, TCP/IP. Each gateway is in essence performing a protocol convergence function.

The MIT environment maintains a single internetwork network layer protocol. This is partially an artifact of the early entrenchment of TCP/IP in the computing environment. It is necessary for network hosts to convert to TCP/IP to become full partners to the

²⁵ Tanenbaum, p. 22.

²⁶ *Ibid*, pp. 226-231.

internetwork domain.

Non-TCP/IP hosts that do not convert can gain access through the acquisition of gateways. The difference between the gateways proposed by Harvard and those employed at MIT is that the function of the Harvard gateways are largely transparent to the user. On a Harvard DECNET host, a user could still use the familiar DEC Disk Access Protocol to retrieve files from TCP/IP hosts (See the DECNET section below). The gateway handles the mapping between applications. A user on an MIT DECNET host would currently have to have an account on the gateway machine in order to gain access. Furthermore he would have to learn to use TCP/IP's File Transfer Protocol to accomplish his ends.

7.3.2 Resultant Internetworking Strategies

The divergent implementation of the protocol standard has effectively chosen differing internetworking strategies for each university. Harvard's gateway convergence implementation makes their approach an extension of the hop-by-hop enhancement. All internetwork communication consists of exactly two hops — once onto the backbone and once off it into the target access subnet. This architecture greatly reduces the number of SNDCPs that must be implemented. In the original, each distinct subnetwork-to-subnetwork link required a SNDCP.

With 15 different subnetwork access protocols, 105 SNDCPs would have been required (15 choose 2). With a backbone, each access protocol must be converged only to the protocol standard for the backbone, requiring only 15 SNDCPs.

MIT has arrived at the Internetworking Protocol through simple standardization on TCP/IP at the host level. As remarked before, this was not likely a deliberate decision that anticipated future ISO work in internetworking. The level of TCP/IP support is more of a historical and evolutionary effect. The author would like, for the moment, postpone consideration of non-TCP/IP hosts under this schema to the following section and turn to the trade-offs between the two effective strategies.

Piscitello compares the hop-by-hop and ISO IP approaches and discovers some important advantages to the ISO IP strategy. The ISO IP should be used where LANs are involved in internetworking. Benefits are derived from resource optimization, throughput enhancement (through the use of load-splitting techniques) and redundancy and resiliency (the ability to adapt to redundancy)²⁷.

Resource optimization. Using the hop-by-hop approach, resources (such as buffers, a connection-state-information base and CPU) must be reserved at both end systems as well as at the gateways for the duration of the connection. The gateways must have ample capacity to maintain a number of connections even if no traffic is passed. Clearly, if connections remain idle for long periods of time, valuable network resources are wasted.

In contrast, if the ISO IP is used, the sending end system may free resources as soon as the data unit's transmission is completed. Any communicating pair of Network service users that has long periods of inactivity imposes no overhead. Therefore, the ability of the gateway to process requests from any other communicating pair remains unaffected. This typically results in highly efficient use of resources²⁸.

Throughput enhancement. In many internetworking scenarios, the ability to route IP data units independently is particularly useful. Data exchanged between hosts attached to one subnetwork can be routed to hosts on a different (remote) subnetwork without the constraint that all data must be routed down the same path. Using multiple paths to transmit data to the same destination typically improves throughput and reduces response time²⁹. Figure 4 illustrates load splitting.

²⁷ Piscitello, p. 130.

²⁸ *Ibid.*, pp. 130-133.

²⁹ *Ibid.*, p. 133.

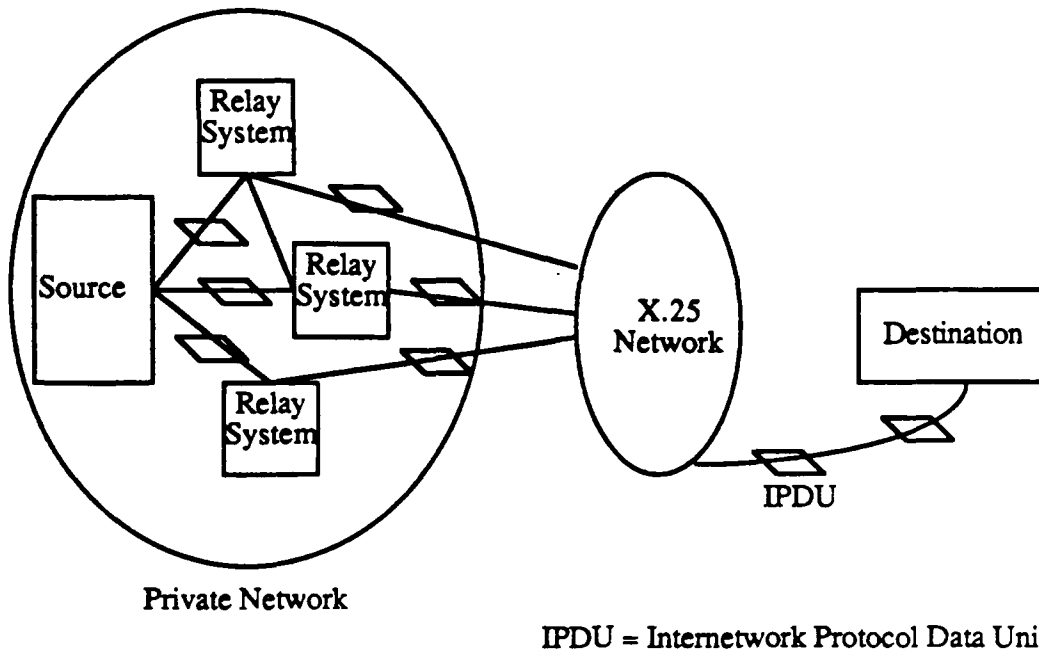


Figure 4: Load Splitting
Source: Piscitello, *Data Communications*, May 1986, p. 135.

On the surface, the Harvard and MIT data network implementations appear identical. Both use the same medium, subnetworks, protocols, and both use gateways. Nonetheless it has been shown that Harvard's resultant internetworking strategy is inferior to the MIT ISO IP approach.

7.4 TCP/IP — Implications for the Subnetworks

7.4.1 DECNET

Due to the availability of public domain TCP/IP support as well as recent product introductions, the DECNET manager has no concern over the selection of TCP/IP as the backbone standard.

Carnegie Mellon University has implemented TCP/IP for VAX/VMS systems, which it makes available at essentially no cost (only tape medium, documentation, and

shipping costs). The protocol support is entirely software based and therefore requires no additional hardware. It provides TCP/IP capabilities and utilities to complement those already provided by DECNET. Users can use FTP to access files on Internet hosts and then switch to use Data Access Protocol (DAP) to access information on DECNET hosts.

There is a performance penalty, however. TCP/IP consumes an order of magnitude more resources (CPU and I/O bandwidth) than DECNET for analogous functions. Peter Roden, Manager of VMS Systems for Harvard's Science Center, believes that the poor performance stems from CMU's implementation rather than from anything inherent to the task. If this is true, than more efficient implementations may be obtained that do not impose this performance premium. The lack of a competitive offering suggests that system managers do not view the penalty enough to warrant laying out real money for a better product.

Digital Equipment Corporation has recently introduced a DECNET-Internet Gateway³⁰. This product provides bidirectional access to system resources and utilities between DECNET and Internet resources based on a network applications mapping. It provides for file access and transfer, remote virtual terminal access, and mail exchange according to the following mapping:

<u>Application</u>	<u>DECNET Protocol</u>	<u>Internet Protocol</u>
File Transfer	DAP (Data Access Protocol)	FTP (File Transfer Protocol)
Remote Terminal	CTERM (Command Terminal)	TELNET
Mail	MAIL-11	SMTP (Simple Mail Transfer Protocol)

The gateway gives DECNET users access to Internet nodes as well as giving Internet hosts access to DECNET services. Unlike previous products, this one does not require users accounts on the gateway node nor special software on systems that use its

³⁰ Karen L. Gillin and Peter N. Harbo, "The DECnet-Internet Gateway," Networks and Communication Software Engineering Group, DEC, Littleton, MA, February 12, 1987, p. 1.

services. A user may access nodes on the alternate network using the network applications with which he is familiar as though the node were resident on the same network. Specific knowledge of the foreign network applications or syntax is not required. This is a decided improvement over the dual TCP/IP and DECNET implementation approach described above.

Since the DECNET and Internet services are not exactly symmetric, DECNET users may experience some slight variation. The DECNET utilities are supersets of the Internet utilities. FTP provides only a subset of the functionality the DAP provides. In many cases there is no corresponding FTP message for a DAP message³¹.

FTP specifies 3-digit return codes to specify the success or failure of the requested action, with 100 different possible values. DAP has literally hundreds of error codes defined for basically any error that could be received on a DEC system. Although it is easy to map a DAP error code to an FTP return code, the converse is not true³².

Due to the heterogeneous nature of the systems that may use the TELNET protocol, few assumptions are made about the remote systems and their capabilities. Therefore, TELNET keeps minimal information at the client end about the terminal at the server end. The CTERM protocol, on the other hand, keeps extensive information about the server process at the client end, enabling it to take better advantage of graphics workstation capabilities³³.

7.4.2 XNS

The implications for XNS subnetworks are much more severe. Harvard has two clusters of Xerox workstations, one in the Vanserg building for the Classics Department and the other in Aiken Computational Laboratory. The Vanserg cluster consists of Xerox

³¹ *Ibid.*, p. 5.

³² *Ibid.*, pp. 8-9.

³³ *Ibid.*, p. 6.

Stars used to examine Greek texts in their original form, Xerox printer servers and a file server of 400 Megabytes of Greek literature. The Aiken cluster maintains some Stars as well as a central file server for the Xerox machines. The Classics Department also maintains accounts on a host at Harvard's Science Center Computing facility.

In the current implementation, all these buildings are a part of the FAS Ethernet. On this network, XNS, TCP/IP, DECNET, and LAT protocols all coexist without interfering with each other. All three buildings (Vanserg, Aiken, and the Science Center) are served by the Ethernet, so XNS operates transparently to the users.

In the proposed network architecture, the Ethernet would be split into discrete segments each serving individual buildings or clusters. Although there exists UNIX software that permit UNIX hosts to access XNS file servers, the unavailability of true XNS to TCP/IP gateways presents a difficult problem for the Classics Department.

One possibility might be to add TCP/IP support to all XNS hosts. The difficulty here is that Xerox's systems are proprietary and do not include source licenses. Any modifications would have to be implemented by Xerox, and may not be available on a timely basis, if at all.

The alternatives presently being considered are to install a separate XNS network link clustering the three buildings. The cost of this installation would have to be borne by the Classics Department and the Division of Applied Sciences and may be prohibitive. The other possibility under deliberation is for the Classics Department to acquire its own central file server to eliminate its dependence on Aiken's resources. The last alternative is to simply abandon the Xerox systems in favor of more compatible Sun or DEC products that would integrate more effectively in the University network plan.

The optimal solution depends greatly on the progress of ISO OSI in adoption by computer and communications vendors. If all vendors were to deliver ISO IP upgrades tomorrow, the Classics Department's problem would be solved. Segmentation of the Ethernet would pose no difficulty. If adoption of ISO IP is slowed, then the optimal

solution would depend on the hardware prices the department would be able to obtain from the vendors for competing products.

MIT's Sloan School of Management possesses a number of Xerox workstations on its Ethernet. Since all the hosts reside on the same network, Sloan does not face Harvard's acute connectivity difficulties. Protocol incompatibility problems exist, but some have been worked around. Telecommunications Systems has installed XNS support on a UNIX host which serves as the electronic mail distribution point for the Sloan XNS clients.

7.4.3 PRONET

As described previously, PRONET users are reluctant to sacrifice the greater functionality of the NOVELL network operating system to gain TCP/IP compatibility. MIT's Medical Department has three principal concerns about joining the Campus Network. The biggest is the question of the security of the Campus Network. The others are the high cost of gaining connection (\$50,000 for installation and gateway, plus operating charges) and the loss of functionality.

A gateway solution would allow the subnetwork to continue to use NOVELL without any interference from the Campus Network. The difficulties that NOVELL subnetwork managers identify are not difficulties arising from the selection of TCP/IP as a backbone standard protocol so much as they stem from broader security issues and cost constraints.

There are no NOVELL users at Harvard.

7.4.4 SNA

There are no SNA installations at MIT. Harvard, however, has SNA running on its IBM hosts at the OIT Computing Center. TCP/IP convergence is a difficult proposition but is being neatly avoided by an approach adopted by OIT. The SNA subnetwork will treat the backbone network simply as a data delivery system.

Bob Carroll, Director of the OIT Computing Center described the approach. The backbone gateways serving the SNA access networks will simply envelope the SNA packets inside the backbone protocol. At the receiving end, the TCP/IP envelopes will be removed, and the SNA packets will continue on the target SNA access net.

In this system, it does not matter what the backbone standard protocol is. Since no attempt is made to converge protocols, no incompatibility is encountered. This is, however, not a solution that offers any interoperability among SNA and non-SNA hosts. Unless IBM offers ISO IP support and Harvard migrates to an ISO IP approach, no real internetworking will be achieved.

A possible alternative solution would be the incorporation of SNA gateways to ISO IP products. For example, DECNET is moving quickly to an ISO OSI model to be reached the next implementation, Phase V. DEC offers a SNA/DECNET gateway product that could remedy the connectivity problem.

Harvard has committed in principle to moving toward ISO OSI (see Section 7.2.1). IBM supports the standardization of protocols, but is lobbying to have its SNA be that standard³⁴. Nonetheless, ISO leaders are confident of the convergence toward protocol standardization on the OSI reference model among the major manufacturers³⁵.

7.4.5 NFS

The selection of TCP/IP is the best news that NFS managers could possibly receive. NFS operates over TCP/IP. The standardization of TCP/IP guarantees the maximum interoperability of NFS. NFS users will have no difficulty in mounting file systems across the backbone and even into external Internet domains.

³⁴ Anura Guruge, *SNA: Theory and Practice — A comprehensive guide to IBM's Systems Network Architecture* (Exeter, Devonshire, England: A. Wheaton & Company Limited, 1984), pp. 383-386.

³⁵ Richard des Jardins, "Towards the Information Society: World Cooperation on Open Systems Standardization," *Computer Network Usage: Recent Experiences*, L. Csaba, K. Tamay, and T.Szentivanyi eds., (New York: North-Holland, 1986), pp. 15-17.

8 Which Way to ISO Internetworking?

Consider the position today of an Information Systems planner deliberating the optimal strategy for internetworking. He has a network environment consisting of heterogeneous local area networks, some perhaps separated by significant distances. He understands the significance and potential of ISO OSI protocol standardization, but has internetwork needs now. The following attempts to focus this decision.

8.1 Protocol Selection Criteria

There are a number of important issues influencing the selection of a protocol suite for internetwork communications. The author here examines the evaluation of Network Layer and Transport Layer protocols for internetworking. They fall into four major categories.

- Functionality
- Availability
- Interoperability
- Performance
- Cost

A protocol's functionality is an important issue in its evaluation. For research organizations with sophisticated users, a wide range of transport layer support is important. The sophisticated user will want access to both highly reliable virtual circuit and faster datagram support. Less sophisticated users value easy-to-use presentation and applications level utilities that support their basic networking needs (e.g., file transfer, remote login, electronic mail).

There are multiple facets of availability. An important issue is time, particularly when the manager is considering ISO OSI protocols. The length of product introduction

delays is uncertain and hard to predict. Another facet is the protocol's availability from various vendors. "Can I get protocol X support for my DEC VAX as well as my IBM 4341?" is the question being asked. This question is inextricably linked to the question of implementation examined above in contrasting Harvard and MIT's approaches.

Interoperability also consists of a number of sub-issues. The interface between this protocol suite and the protocols used on the subnetworks is one issue. Are gateways currently offered by computer and communications products manufacturers to facilitate the implementation of the planned internetwork architecture? How dependent is the network layer on the underlying data link support? Will the network layer provide support over Ethernet, token ring, as well as X.25 public data networks?

Performance measures have more to do with routing efficiency than with data lossage. A network and transport layer protocol suite if correctly implemented will provide users with the functionality required. The data link layer choices are the source of much of data lossage. If a network layer relies on static routing, then it lacks the flexibility to find alternate paths when a particular link is disrupted. Poor routing algorithms can result in excessive looping and therefore performance loss.

The cost issue includes the expense of acquiring protocol support as well as a measure of the hardware investment necessary to implement it.

8.2 Network Environment Characterization

As observed above the assessment of protocol selection depends greatly on the target environment. The network planner must answer a number of questions before he can begin to assess the relative merits of one internetwork approach over another.

- What kind of users do I have? Are their needs sophisticated or do they mainly need basic general utilities?
- What type of functionality must I support? What services must the internetwork provide?

- What are the various types of subnetworks that will need access to the internetwork? What are the data link layers involved? Which subnetwork-specific protocols are currently being used?
- What hardware is in use? How heterogeneous is the computing environment? Does one manufacturer's equipment dominate?
- How decentralized is network administration?
- What development and technical resources do I have available? Will I have to buy off-the-shelf products or can I develop some missing links myself? Do I have the technical support resources in-house or will I have to rely on a vendor?

These questions must be answered to generate a context for the evaluation.

8.3 Author's Evaluation

The author will exercise the evaluation approach by indulging in the evaluation of some alternative protocol suites from the above developed MIT/Harvard network context. The results are summarized below. Digital Equipment Corporation's DECNET has been included as an intermediate step between the static position of staying with TCP/IP and waiting for OSI. DEC is attempting to make its Digital Network Architecture (DNA) Phase V OSI compatible. Adoption of DECNET would provide some internetwork services immediately with a high likelihood of successful migration to OSI.

TCP/IP's biggest advantage is that it already has a tremendous installed base. The ARPANET and MIT's Project Athena are two important examples. Furthermore, TCP/IP's availability from the largest number of different vendors has made it a *de facto* standard. The biggest short coming is the uncertain path of migration to an OSI standard.

DECNET is available now and offers a better migration path to OSI. Its proprietary nature, however, makes it very limiting as an alternative for the heterogeneous MIT and Harvard environments.

	TCP/IP	OSI	DECNET
Functionality			
Sophisticated user	o	+	+
Novice user	o	+	+
Availability			
Time	+	--	+
Multiple vendor	+	++	--
Uncertainty	+	--	+
Interoperability			
Interface to subnetworks	+	++	o
Flexibility over data links	o	++	+
Performance			
Routing efficiency	+	+	o
Cost			
Installation	++	?	-
Operating costs	o	?	-
Migration to OSI	--	++	+

++ Excellent
 + Satisfactory
 o Satisfactory
 -
 -- Poor

Figure 5: Author's Evaluation for Harvard/MIT Environment

OSI IP will offer the greatest interoperability of any of the alternatives. This assumes that manufacturers will in fact eventually conform to the OSI standard. The author confesses a certain enthusiasm for OSI and a degree of optimism that this will indeed happen. The sole difficulty is that OSI IP is not available. Moreover, it is difficult to predict when the major manufacturers will all offer OSI products. Leadership by one vendor will provide some momentum, but unless all major vendors follow suit complete interoperability will never be achieved.

For the Harvard/MIT context, the author would select OSI IP. Both universities are able to take a long-term view for planning. The uncertainty regarding the timing of vendor product convergence on the OSI standard is important. MIT is in a better position to wait,

but Harvard has much more to gain by waiting. It will certainly be possible for Harvard to migrate to OSI IP when it becomes available, but the cost may not be expensive.

9 Plans for the Future

Both universities are looking into their future communications needs and are attempting to acquire facilities that will meet or exceed these requirements well into the 1990s. The uncertainty regarding the timetable of the arrival of ISO OSI protocols makes current network design decisions difficult.

9.1 Harvard's University Network

Harvard's Request for Proposal details an integrated voice, data, and video network. The architecture calls for an integrated voice and data PBX as well as parallel backbones to support data and video communications. It is difficult to evaluate their future plans since proposals are still being formalized by the bidding vendors.

Harvard's OIT has done a remarkable job of gaining the cooperation and support of the various schools and departments around the campus. Harvard's schools are known to value their independence, and OIT's efforts are a real accomplishment.

Examination of the short-range implementation of the University Network indicates that it falls far short of an interoperable internetworking environment. The views voiced by the OIT Computing Center indicate that not all access subnetworks will be full partners in sharing services. Harvard must be prepared to move quickly to an ISO IP implementation to achieve a truly interoperable internetwork.

9.2 MIT Campus Network

Much more can be said about the state and course of the MIT Campus Network. The network is in a state of transition with the addition of a major new telecommunications system. Now is a time of opportunity.

9.2.1 5 ESS Voice/Data PBX

MIT's Telecommunications Systems is in the process of acquiring a 5 ESS integrated voice/data PBX to replace their current CENTREX system. Immediate plans call for installation of voice capability only, however, wiring will be completed for four wire pairs to support voice transmission as well as four additional pairs to support Local Area Network access when that service is added. Fiber optic links will join all the switches.

Dennis Baron, a manager for Telecommunications Systems, believes that the PBX will be used primarily to replace dedicated lines for administration users. He sees the 5 ESS installation as encouraging a reorganization or possible replacement of the backbone. Baron would like to relocate the Campus Network gateways to concentrate them in the switch node locations. Since the switch nodes are linked by fiber optic cable, these links could provide some valuable redundancy.

Thus, the 5 ESS creates the opportunity for a profound change in MIT's Campus Network. When ISO IP implementations become available, MIT should migrate to such a connectionless internetwork environment. The 5 ESS links will offer a tremendous amount of redundancy that an ISO IP scheme could utilize to greatly increase the throughput of the network. Individual packets of a single session could be routed independently to eliminate bottlenecks and improve performance.

The most significant aspect of the PBX system is its reach. Everyone has a telephone. The ISO IP approach of treating the subnetworks simply as data pipelines without regard for their speed, bandwidth, and reliability gives it the flexibility to incorporate the voice/data switch into the internetwork environment. The PBX trunks and lines will provide a tremendous amount of redundancy that would improve the overall robustness of the Campus Network.

9.2.2 Security

Data security is a high priority for administrative users in a university environment. It must be addressed if administrative networks are to become full-fledged clients to a

university-wide system. It is interesting to note that the key issue here is "perceived" security rather than any objective measure.

MIT's Kerberos authentication scheme helps protect against unauthorized users from gaining access to privileged or sensitive information but does not solve the problem of intruders intercepting the data at the network level. Some sort of Data Encryption Standard (DES) must be implemented to safeguard the content of the data. Maintaining network transmission at the lowest power levels can help discover attempts at tampering with the physical transmission medium.

9.2.3 Planning as a Problem in Organizational Behavior

Even if the the IS planner develops a network design that solves the protocol compatibility problem as well as addressing the security issue, he must attend to the organizational issue of co-opting possible opposition among the key stakeholders. Because the decision-making process about the acquisition of computing and communications hardware is decentralized, it is necessary for the IS planner to gain the cooperation of the departmental decision makers or at least to mollify them.

The method that Harvard has adopted in defining its university-wide network is worth careful consideration as a model. From the outset, users were informed what the goals were and how they would be achieved. Reports summarizing progress and findings were published as soon as possible so that stakeholders could observe and participate in the process. A steering committee was created with every school and administrative group represented, giving formal recognition of their opinions.

Since the process has not yet advanced into the implementation phase, it would be premature to pass judgement on Harvard's methodology. However, it is certainly the case that the major stakeholders are satisfied that their views have been heard. Furthermore, the steering committee representatives serve as champions of the network plan within their own organizations.

MIT is attempting a project that incorporates some of the same attention to stakeholder positions. Presently, administrative users that require access to central administration data maintained on another network were forced to negotiate access on an *ad hoc* basis with contacts in the organization responsible for the database. Administrative Systems is striving to develop a distributed database system that would greatly facilitate access as well as eliminate repetitive data entry.

In order to minimize the risk perceived by the client administrative groups, Administrative Systems is employing a phased implementation.

- Creation of read-only duplicate databases by the owning administrative group. Security issues are circumvented by permitting access only through dial-up. The owning group may isolate itself (and the integrity of its system) simply by disabling its modems. The control over the link satisfies the users' perception of security. Faculty and administrative groups can gain access to sections of the database pertaining to them (subject to authorization) on a read-only basis.
- User ability to modify low-level information like address and phone number. Owning organization allows write privileges for segments of their database. The duplicate database will be eliminated by each owning organization.
- Elimination of modem links in favor of Campus Network links. This will improve data rates for transfer and access by taking advantage of the superior bandwidth available over the Campus Network.

It is hoped that the system will eventually evolve into a true distributed system that would improve speed of access and eliminate all duplication and reduce paperwork. Tom Shea, of Administrative Systems, ultimately hopes that MIT can go to digital admissions applications, greatly reducing the paperwork burden for the Institute.

9.3 Summation

The author sets great store in the promise of the International Standards Organization's efforts to create internetworking standards within its Open Systems

Interconnection reference model. The cooperation and coordination of the major computer and communications vendors in adhering to these standards is the key to addressing the connectivity problems within any communications environment.

Specific to the university environment, the issues of data security and managing the planning process need direct attention. Without the support and cooperation of the autonomous subnetwork managers, the goal of a fully interoperable internetworking environment will be difficult to achieve.

APPENDIX — SAMPLE MIT QUESTIONNAIRE

1. Name: _____
Title/Year: _____
2. Please whether you are: (circle one)
FACULTY
ADMINISTRATION
STUDENT
3. Department or School _____
4. Location
Office/Dorm _____
Extension _____
5. What is your best estimate of the average total number of hours a day you use a computer or terminal? (Please circle one.)
0-1 hour
1-2 hours
2-4 hours
4-6 hours
More than 6 hours
6. On the average, what percent of the time you currently spend communicating with other computers do you communicate with the following?
 - a. Local computer (located in your department) including shared disks and printers _____ %
 - b. Computer within your local subnetwork _____ %
 - c. One of the major computer centers outside your local subnet but within MIT _____ %
 - d. Networks outside MIT (e.g., ARPANET, BITNET, USENET, CSNET) _____ %

7. In general, which types of communications with MIT computers and outside networks do you utilize? Check each box that applies.

**Facility
Transfer**

**Interactive File
Terminal**

- a. Local computer or file server (in your department)
- b. Computer within your local subnet
- c. Major computer center within MIT but outside your local subnet
- d. Networks outside MIT

8. Please read the following items and indicate whether the item is important to you. (Circle the letter corresponding to your response.)

A means NOT AT ALL IMPORTANT
 B means SOMEWHAT IMPORTANT
 C means IMPORTANT
 D means VERY IMPORTANT
 E means EXTREMELY IMPORTANT

Circle your rating

Access to databases outside MIT for research	A	B	C	D	E
Access to databases inside MIT for departmental or administrative information.....	A	B	C	D	E
Ability to access different networks within MIT	A	B	C	D	E
Interchange revisable word processing documents	A	B	C	D	E
Electronic mail	A	B	C	D	E
File transfer	A	B	C	D	E
Share resources like printers, file servers, etc.	A	B	C	D	E
Remote login.....	A	B	C	D	E
Ability to communicate text and image documents.....	A	B	C	D	E

What other services do you feel are important?

SERVICE #1 _____

SERVICE #2 _____

SERVICE #3 _____

9. Please rate MIT's current data network on its ability to fulfill your needs with regard to the following services:

	LOW	Circle your rating				HIGH
Access to databases outside MIT for research	1	2	3	4	5	
Access to databases inside MIT for departmental or administrative information.....	1	2	3	4	5	
Ability to access different networks within MIT	1	2	3	4	5	
Interchange revisable word processing documents	1	2	3	4	5	
Electronic mail	1	2	3	4	5	
File transfer	1	2	3	4	5	
Share resources like printers, file servers, etc.	1	2	3	4	5	
Remote login.....	1	2	3	4	5	
Ability to communicate text and image documents.....	1	2	3	4	5	

10. Please read the following standards and rate the MIT data network by circling the appropriate number.

	LOW	Circle your rating				HIGH
Ability to reach the entire user community	1	2	3	4	5	
Performance (speed and response).....	1	2	3	4	5	
Reliability.....	1	2	3	4	5	
Operating costs (your own)	1	2	3	4	5	
Planning efficiency.....	1	2	3	4	5	
 Overall satisfaction with the MIT data network	 1	 2	 3	 4	 5	

11. If there is a service that was not described and is very important to you, please describe that service below.

12. Which protocols are used on your subnetwork?

13. Please describe any specific problems that you have had with the MIT data network, specifically protocol incompatibilities.

14. Please describe any especially effective aspect(s) of the MIT data network.

BIBLIOGRAPHY

- Bazewicz, Mieczyslaw. "The Design, Implementation and Information Services in the Interuniversity Network /MSK/ — Results and Experiences." *Computer Network Usage: Recent Experiences*. L. Csaba, K. Tarnay, and T. Szentivanyi, eds. New York: North-Holland, 1986.
- Bazewicz, Mieczyslaw and Frank Stuchlik. "Computer Networks in the Academic Environment." *Computer Network Usage: Recent Experiences*. L. Csaba, K. Tarnay, and T. Szentivanyi, eds. New York: North-Holland, 1986.
- Chorafas, Dimitris N. *The Handbook of Data Communications and Computer Networks*. Princeton, NJ: Petrocelli Books, 1985.
- Denning, Peter J. "Security of Data in Networks." *American Scientist*, Vol. 75, pp. 12-14.
- des Jardins, Richard. "Towards the Information Society: World Cooperation on Open Systems Standardization." *Computer Network Usage: Recent Experiences*. L. Csaba, K. Tarnay, and T. Szentivanyi, eds. New York: North-Holland, 1986.
- Gillin, Karen L. and Peter N. Harbo. "The DECNet-Internet Gateway." Network and Communication Software Engineering Group, Digital Equipment Corporation. February 12, 1987.
- Guruge, Anura. *SNA: Theory and Practice — A comprehensive guide to IBM's Systems Network Architecture*. Exeter, Devonshire, England: A. Wheaton & Company Limited, 1984.
- "Harvard University Telecommunications Long Range Plan: Needs Assessment Summary." Harvard University Office for Information Technology, Telecommunications Services Division, August 13, 1986.
- "Harvard University Telecommunications Long Range Plan: Network Architecture Recommendation." Harvard University Office for Information Technology, Telecommunications Services Division, January 1987.
- "Harvard University Telecommunications Long Range Plan: Research Plan." Harvard University Office for Information Technology, Telecommunications Services Division, April 7, 1986.

"Harvard University Telecommunications Long Range Plan: Resource Summary." Harvard University Office for Information Technology, Telecommunications Services Division, August 18, 1986.

"ISDN — Betting on the Future." *Datapro Reports on Telecommunications*. Delran, NJ: Datapro Research Corporation, 1986.

Lerman, Steven R. "Questions and Answers About Project Athena." MIT Project Athena, Revision C, November 1986.

Lidinsky, William P. "Insights Into the Implementation and Application of Heterogeneous Local Area Networks." Proceedings of the 7th Annual Data Communications Symposium (*ACM SIGCOMM Computer Communication Review*, Vol. 11, No. 4, pp. 52-61).

Merims, Linda and K. L. Lieden. "Essential Athena." MIT Project Athena, Revision D, January 16, 1987.

Multi-Vendor DC Networks — Planning, Perspectives and Strategies. Wellesley, MA: Q.E.D. Information Sciences, Inc., 1982.

Networks and Communications Buyer's Guide. Digital Equipment Corporation, 1986 October — December.

"Networks at MIT." Information Systems Series Memo IS-10-1, MIT, November 13, 1986.

Piscitello, David M., Alan J. Weissberger, Scott A. Stein, and A. Lyman Chapin. "Internetworking in an OSI environment." *Data Communications*, May 1986, pp. 118-136.

"Request For Proposal for an Integrated Telecommunications Network for Harvard University." Harvard University Office for Information Technology, Telecommunications Services Division, Version 2, January 1987.

Rosner, Roland. "Networked Computer Services for Universities." *Computer Network Usage: Recent Experiences*. L. Csaba, K. Tamay, and T. Szentivanyi, eds. New York: North-Holland, 1986.

Routt, Thomas J. "Distributed SNA: A network architecture gets on track." *Data Communications*, February 1987.

Sherman, Richard H. and Melvin G. Gable. "Considerations in interconnecting diverse local nets." *Cases in Network Design*. Bracker, William E. and Ray Sarch, eds. New York: McGraw-Hill, Inc., 1985.

Slone, John P. "Assessing LAN Technologies." *Auerbach Data Communications Management*. Auerbach Publishers Inc., 1986.

Spratt, E.B., D.J. Caul, and S.E. Binns. "The Management of Services on Computer Networks." *Computer Network Usage: Recent Experiences*. L. Csaba, K. Tarnay, and T. Szentivanyi, eds. New York: North-Holland, 1986.

Tanenbaum, Andrew S. *Computer Networks*. Englewood Cliffs, NJ: Prentice Hall, Inc., 1981.

"Telecommunications Wiring Guide." Harvard University Telecommunications Services Division, May 22, 1986.

Treese, Win and Michael Johnson *et al.* "Project Athena Technical Plan: The Athena Workstation." MIT Project Athena, Revision 1.2, November 26, 1986.

Weissberger, Alan J. and Jay E. Israel. "What the New Internetworking Standards Provide." *Data Communications*, February 1987, pp. 141-146+.

Zimmerman, H. "OSI Reference Model — The ISO Model of Architecture for Open Systems Interconnection." *IEEE Transactions on Communications*, Vol. COM-28, April 1980, pp. 425-432.

END

DATE

FILMED

9-88

DTIC