EORGIA INSTITUTE OF TECHNOLOGY PROJECT A	DMINISTRATI	OFFICE OF CO ON DATA SHEET	NTRACT ADMINISTRATION
Project No. <u>G-36-663</u> Project Director: <u>Dr. W. E. Underwood</u>	d A	X ORIGINAL DATE	
Sponsor: MERADCOM; Procurement & Pro	oduction Dire	ctorate; Ft. Belvoi	r, VA 22060
Type Agreement: Delivery Order 001	l under Contr	act DAAK70-79-D-008	7
Award Period: From <u>4/15/82</u> To Sponsor Amount: <u>\$73,982</u> Cost Sharing: <u>none</u>	1/31/83	(Performance)	(Reports) Contracted through GTRI/S¥X
Title: Analysis of Existing Decision	n Support Sys	tems to Provide Cur	rent Baseline
Applications, Implementation Metho	odologics, Pr	oblem Areas, Hwe, a	nd Comm. Cmplexities
ADMINISTRATIVE DATA	OCA Contact	William F. Brown	
() Sponsor Technical Contact:		2) Sponsor Admin/Contra T. A. Bryant	ictual Matters:
	1999 A.	ONR RR	
	1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 -	Ga. Tech	
		Atlanta, GA 30332	
Defense Priority Rating: DO-S1		Security Classification:	Unclassified
RESTRICTIONS See Attached Gov't Supp Travel: Foreign travel must have prior approval	elemental Informat – Contact OCA i	ion Sheet for Additional R n each case. Domestic tra-	equirements. vel requires sponsor
Equipment: Title vests with <u>None propo</u>	sed.		
COMMENTS-	Contraction and	Activity of the	
	1965		293031-123 20
	a state of the second s	522	APR 1982
	And a start	53246	RECEIVED Research Reports
		E	00000000000000000000000000000000000000
COPIES TO:		and the second	
Administrative Coordinator Research Property Management Accounting Procurement/EES Supply Services ECRM OCA 4781	Security Services Coordinator (CC ervices (DCA)	<ul> <li>EES Public R</li> <li>Computer Inp Project File Other</li> </ul>	elations (2) but

EORGIA INSTITUTE OF TECHNOLOGY	OFFICE OF CONTRACT ADMINISTRATION	
SPONSORED PROJECT TERMI	NATION/CLOSEOUT SHEET	The second
	May 9, 1984	
	Date	-
Project No. <u>G-36-663</u>	School/KADK ICS	-
ncludes Subproject No.(s)		-
Dr. W. F. Uriderwood		53
Project Director(s)	GTRI / ŠXX	
SponsorUNR_RR, Georgia lech		-
Title "Analysis of existing Decision Support S	ystems to Provide Current Baseline Applic	a-
tions, Implementation Methodologies, Pro	blem Areas, Hardware and Communication	10
Complexities"		
1/21/92	(9. (	
Trective Completion Date: 1/31/03	(Performance) 1/31/65 (Report	5)
	and the second sec	
Grant/Contract Closeout Actions Remaining:		
News		
Final Invoice or Final Fiscal Report	(Final Invoice Submitted)	
Closing Documents	A STREET	-
Einel Depart of Inventions		
Govt. Property Inventory & Related C	Certificate	
		1
Classified Material Certificate	and the second	
Other	and the second second second second	1.1
Continues Project No.	Continued by Project No.	
		16.1
COPIES TO:		
Project Director	Library	
Research Administrative Network	GTRI	
Research Property Management	Research Communications (2)	
Accounting Procurement/EES Supply Services	Other	
Research Security Services		450
Reports Coordinator (OCA)		
		-
Legal Services		

N.

22.2017.04

1 Marco

4.1

To: Mike Evans, COTR, AIRMICS From: Bill Underwood, Project Director Date: June 7, 1982 Subject: Monthly R&D Status Report

This memo summarizes activities for the period April 15, 1982 to May 31, 1982 conducted in connection with the DSS Baseline project.

G-36-663

Staff working on this project during this period were W.E. Underwood (1/2 time), Mariann Ogelvee, Graduate Asst. (1/3 time), and Larry Riegel, Grad. Asst. (1/3 time). While Phil Siegmann, Assoc. Prof. was not billed to this project during this period he also made contributions.

The primary activities scheduled for this period were development of a project task schedule, a literature review, design of a survey instrument, and identification of organizations for study. The task schedule was presented to you May 11 and verbally approved.

An automated literature search was conducted to support the task of identifying organizations that have DSS's and to obtain an uptodate survey of the DSS R&D literature. We searched the NTIS Government Report Announcements for the period 1975-April 1982 (41 citations), Dissertation Abstracts (17 citations), Science abstracts/INSP for the period 1977-April, 1982 (100 citations), the ABI/INFORM database for the period 1971-April, 1982 (105 citations).

Much of this period was spent 'in designing a survey instrument for the study and in developing a plan for data collection, analysis and evaluation. The survey instrument has two components, a set of questions for decision- makers who use decision support systems and a set of questions for

DSS developers or vendors. The questionaire for decisionmakers includes questions that elicit the capabilities of the dss and problems in dss usage from the decision-makers point of view. The questionaire for the developer/vendors include questions that provide information as to the developmental strategies and the hardware, software, telecommunication, documentation, dialog, modelling, and database characteristics. Almost all of the information in the survey will be obtained from personal interviews. The results of the analysis of the survey will be primarily descriptive as our sampling technique is to obtain a representative sample of dss applications, developmental methodologies and characteristics. We will report the detail of the survey instrument at the first inprocess review.

Approximately 250 users and developers of DSSs have been identified including 30 in the Atlanta area. We are developing management and technical contacts in some of these organizations and will be recommending in late June a representative sample for study.

We are attending DSS-82 in June and will be interviewing several users and developers of DSSs in San Francisco and Palo Alto.

# GIT-ICS-84/01

February, 1984

Analysis of Existing Decision Support Systems to Provide a Current Baseline of Applications, Implementation Methodologies, Problem Areas, Hardware, and Communication Complexities

> William E. Underwood Philip J. Siegmann John Gehl

This study was performed for the U. S. Army Institute for Research in Management Information and Computer Sciences (AIRMICS) under contract no. DAAK70-79-D-0087/0011 to the School of Information and Computer Science, Georgia Tech. Findings and opinions expressed herein are those of the authors and should not be construed as official Department of Army positions, policies, or decisions, unless so designated by other documentation.

### ACKNOWLEDGEMENTS

We would first like to express our gratitude to the administrators, line managers, and staff managers of the organizations who participated in this study for sharing with us their experiences with DSS. We would also like to thank our Graduate Assistants, Marianne Ogilvee and Larry Riegel, for their assistance with the data collection. Finally, we are most appreciative of the support of AIRMICS, and in particular, of Michael Evans, Al Curry and Jim Gantt for their patience and their critical review during the process of the study.

# TABLE OF CONTENTS

# ABSTRACT ACKNOWLEDGEMENTS

].	INTRODUCTION	1
2.	SURVEY OF DSS USERS AND DEVELOPERS	3
	2.1 Survey Methodology	3
	2.2 Selection of DSS Users and Developers	3
	2.3 DSS Descriptions	5
	2.4 Comparision of Surveyed DSS	22
	2.4.1 Hardware, Operating Systems and Development Tools	25
	2.4.2 Dialog and Display	25
	2.4.3 Modeling	30
	2.4.4 Data Management	39
	2.4.5 Communications	41
	2.4.6 Technical and Management Problems	42
	2.4.7 DSS Development Strategies	43
	2.4.8 Comparative Evaluation	50
3.	DEVELOPMENT SIRAIEGY FOR A PROIOTYPE DSS IN THE STAMMIS ENVIRONMENT	53
	3.1 The STAMMIS Environment	53
	3.2 Prototype DSS Development Strategy	54
	3.3 Additional Recommendations	67
4.	SUMMARY	74
REF	ERENCES	75
APP	PENDIX A	80

PAGE

# LIST OF FIGURES

FIGU	JRE	PAGE
1.	Applications of the Surveyed DSS	6
2.	The EPIC Teleterminal	7
3.	Interface of F&M to General Ledger	16
4.	Time-Distance Display from Train Dispatcher DSS	18
5.	Dialog Manager of EXPRESS	20
6.	MYCIN Production Rules	23
7.	MYCIN Dialog	24
8.	Hardware, Operating Systems and Development Tools	26
9.	Dialog, Reporting and Graphics Characteristics	27
10.	DSS/F's Menu-Oriented Dialog	29
11.	Modeling Characteristics	31
12.	Three Dimensional Matrix Model	34
13.	VisiCalc's Screen-Oriented Editor	36
14.	DSS/F's Line-Oriented Editor	37
15.	DataBase Management Characteristics	40
16.	DSS Development Strategies	45
17.	Development Strategies for Surveyed DSS	48
18.	Sequence of Processes in Adaptive Design	57
19.	Sample Infoscope Screens	62
20.	Guidelines for Prototype DSS Development	68
21.	Xerox Star Display	70

#### 1. INTRODUCTION

#### 1.1 Background

Scott Morton (1971) pointed out that management information systems are used by managers for help in making routine decisions or in solving so-called "structured" problems but that such systems are inadequate for ad hoc, novel, decision-making situations or so-called "semistructured" or "unstructured" problems. Keen and Scott Morton (1978) coined the phrase "Decision Support System" (DSS) to refer to a system that provides "computer-based support for management decision makers who are dealing with semistructured problems". A number of businesses have developed decision support systems, and researchers in industry and academia have suggested approaches to DSS development.

A Standard Army Multicommand Management Information System (STAMMIS) is a computer-based management information reporting system. The U. S. Army Computer Systems Command is responsible for the development and maintenance of the STAMMIS systems.

The Army Institute for Research in Management Information and Computer Sciences (AIRMICS) has initiated a research program to determine the feasibility of DSS in the STAMMIS environment--that is to say, in the environment in which Army managers make semistructured financial, logistic and personnel decisions based on the types of information in STAMMIS files and reports.

A previous AIRMICS research project [Callahan (1979)] presented case studies of four DSS: IBM's Geodata Analysis and Display System, First National Bank of Chicago's Executive Information System, Gould Corporation's WINS, and RCA's Industrial Relations Information System. Among the conclusions of that study were the need for a more detailed survey to determine the current DSS technological baseline.

Another important DSS research and development issue is appropriate approach for developing a decision support the that system. Some developers of DSS maintain the software life cycle (Requirements Analysis, traditional System Design, Detailed Design, Code and Test, Integration Test, Installation, and Maintenance), with detailed documentation and review after each phase, is inappropriate for developing DSS. Among the reasons cited is the difficulty a manager or systems analyst faces in formulating the requirements of a system to support semistructured or decision making. unstructured These researchers and developers recommend an approach that they refer to as adaptive [Keen (1980)], evolutionary [Courbon et al. (1980)], iterative [Sprague and Carlson (1982)], that involves or developing a prototype, not for the purpose of determining requirements, but to be used by the manager and iteratively enhanced by the system developer. Other DSS developers

[Thierauf (1982)] maintain that the software life cycle methodology is appropriate.

This question is of central importance to the U. S. Army. Research sponsored by AIRMICS on software life cycle management has indicated that detailed functional requirements and phased implementation are important aspects of software management. The apparent ad hoc nature of developing a prototype and then enhancing this prototype seems to fly in the face of software life cycle research results that indicate the importance of detailed functional requirements. How should the Army develop DSS--via adaptive design or via the software life cycle methodology? This study addresses this question by surveying how successful DSS have been developed in the commercial world.

#### 1.2 Purpose

The major tasks of this study were:

First to select a group of DSS user organizations for study.

Second, to interview managers, DSS users, and DSS developers to collect system descriptions and information on the DSS development methodologies used. The data collected was to include: system objectives; management's evaluation of the degree to which the system met their objectives; management's recommendations for modification and expansion of the system; significant technical problems; hardware, software and communication characteristics; and development tools.

Third, to analyze the collected data, perform a comparative evaluation of the system characteristics, development tools, and development methodologies as they impact the quality of the end system and ease of implementation.

Finally, to recommend a development methodology for an experimental DSS in the STAMMIS environment.

1.3 Scope

The methodology used in this study is described in Section 2.1. The DSS user and developer organizations selected for study are identified in Section 2.2. The DSS surveyed are described in section 2.3. In section 2.4 the system characteristics, development tools and development methodologies are comparatively evaluated with regard to their impact on quality of end system and ease of implementation. In Section 3 the STAMMIS environment is described, a DSS development methodology is recommended and additional DSS research and development recommendations are presented.

#### 2. SURVEY OF DSS USERS AND DEVELOPERS

## 2.1 Study Methodology

The first step was to identify organizations that use DSS and select some of these for study. A DSS was considered to be an interactive computer-based system that was used to support management decisions involving "semistructured" or "unstructured" problems. The details of the selection procedure are presented in section 2.2.

The information was obtained from five sources. First, information from decision makers who were using the DSS directly or via intermediaries. Second, information from staff or staff analysts who were using the DSS. Third, information from developers of specific DSS. Fourth, information from developers of DSS tools. Last, information from user manuals and DSS technical documents.

It was decided to collect the information via interviews and analysis of DSS documentation rather than a mailed questionnaire because the detailed information would be difficult to elicit with a questionnaire. Furthermore the source of the information was managers, management staff, and technical development staff who were busy and might hesitate to fill in a questionnaire, but might give us 15-30 minutes of their time for an interview.

Interview checklists were developed that elicited information from decision makers and staff as to the objectives of the system, their evaluation of the degree to which the DSS had met these objectives, the types of problem solving and decisions supported, benefits, training, ease of use, and recommended modifications or extensions to the DSS. The interview checklist for the specific DSS developer and DSS tool developers included questions as to development strategy, hardware, software, technical problems, communications, documentation, modeling and data management characteristics (see Appendix A). The information obtained is summarized in section 2.3.

The next step was to analyze the data. A comparative evaluation was made of system characteristics, development tools, and development methodologies as they impacted the quality of the end system and ease of implementation. This analysis is presented in section 2.4.

2.2 Selection of DSS Users and Developers.

An automated literature search was conducted to support the task of identifying and selecting organizations that use DSS and to obtain an up-to-date survey of the DSS R&D literature. Bibliographic data bases searched included the NTIS Government Reports Announcements for the period 1975-April 1982 (41 citations), Dissertation Abstracts (17 citations), Science Abstracts/INSP for the period 1977-April 1982 (100 citations), and the ABI/INFORM database for the period 1971-April 1982 (105 citations).

Additional user and developer organizations were obtained by attending the Second International Conference on Decision Support Systems. A number of vendors were contacted to obtain technical and user documentation, and to obtain client lists. Finally, 20 large companies in the Atlanta area were contacted to determine whether they used DSS.

number of DSS surveys and case studies have been A published. Carlson (1977) contains case studies of American Airlines' DSS called An Analytic Information System (AAIMS), First National Bank of Chicago's Executive Information System (EIS), Purdue's Generalized Planning System (GPLAN), RCA's Industrial Relations Information System (IRIS), Managerial Analysis for Profit Planning (MAPP), and General Motor's Relational Generalized Information System (REGIS). Keen and Scott Morton (1978) use seven case studies to illustrate the concepts of a DSS. These cases include a DSS for the laundry equipment division of a major corporation, Great Eastern Bank's Portfolio Management System (PMS), a DSS for financial planning (Projector), IBM's Geodata Analysis and Display System (GADS), a market budgeting DSS (EKANDAID) built using EXPRESS, Ztrux's Capacity Information System (CIS), and Donovan and Madnick's Generalized Management Information System (GMIS). Alter (1979) contains eight detailed case studies, namely, Connoisseur Foods, Great Eastern Bank (PMS), Gotaas-Larson Shipping Corp., Equitable Life (CAUSE), Interactive Market Systems, The Great Northern Bank, The Cost of Living Council, and American Airlines (AAIMS). Callahan (1979) contains detailed case descriptions of four DSS: IBM (GADS), First National Bank of Chicago (EIS), Gould Corp. (WINS), and RCA (IRIS).

Fifteen user organizations were selected that represent a range of organization types. Nine of the companies selected are in the Atlanta area. The others are headquartered in Washington, Illinois, California, the District of Columbia, and New Jersey. There are three manufacturers (Lockheed-Georgia, The Coca-Cola Company, and Boeing), one bank (Citizens and Southern Bank), one transportation company (Norfolk Southern), one electric utility (Georgia Power), one agribusiness (Gold Kist), one service firm (Rollins/Orkin), one oil exploration, drilling, engineering and construction firm (Sante Fe International), one telephone utility (Southern Bell), one government organization (White House Office of Planning and Evaluation), one medical organization (Stanford Medical Center), one software Science America), one conglomerate (Management vendor (Northwest Industries), and one communications research organization (Bell Laboratories).

Characteristics of the twenty DSS in these fifteen DSS user organizations are presented in the next section. The DSS in these organizations support a wide range of decision support functions or applications (see Fig. 1). Eight of these organizations developed their own DSS or contracted for DSS development. Eight organizations used DSS tools available from vendors. Six of the systems available from vendors (EXPRESS, EIS, SYSTEM-W, EMYCIN, EMPIRE, IFPS (plus options)) could be characterized as DSS kernels or meta-DSS for building specific DSS. Two of the DSS user organizations sell their software -- Boeing Computer Services (EIS) and Management Science America (F&M).

#### 2.3 DSS Descriptions

# Bell Laboratories (EPIC)

Several DSS are under development by the Management Information and Systems Division of Bell Labs: the Executive Planning Information and Communications (EPIC) System, the Financial Analysis and Control System (FACTS), and the Administrative Management Information Network (ADMIN).

The design of the EPIC system began in mid 1980 in response to a request from Eell Labs executives for an office automation system that would make their work easier and themselves more effective. Existing office automation systems were examined and it was determined that they were indadequate to meet the needs of the executives. A prototype was designed and then used by executives at Eell Labs, their secretaries and support staff. The director of the Management Information and Systems Division iteracted with the users of the prototype to obtain their reactions and suggestions for improvement. Executives interviewed felt that EPIC met its objective in making communication easier and providing them easy access to information they routinely needed.

The EPIC system runs on any of several minicomputers under version 4.2 of the UNIX operating system. EPIC teleterminals are connected through a local data communication network and a remote network of telephone lines and multiplexers. The teleterminal includes a telephone handset and speakerphone, a keyboard and an 8" diagonal video display surrounded by 25 push-buttons (See Fig. 2). The EPIC software changes the labels alongside these pushbuttons, which are known as "soft keys". The names of EPIC services appear adjacent to a button and the user activates the service by pressing the button next to the name of the service.

One of the design requirements for the EPIC teleterminal was that it be an office fixture that fit into the decor of an executives office. The design has certainly met this requirement.

electronic mail "to do" list Executive Support personal file management scheduling executive decision making "to do" list executive calendar access to employee records for executive's area access to external news, econonomic and corporate data daily proforma income statements weekly snapshots of Corporate Reports performance indicators (graphic displays) access to airline schedules graphical display of production line status ad hoc query Corporate Planning strategic planning acquisition analysis corporate model ad bog model corporate model ad hoc modeling and reporting Financial Planning spread sheets cash vs. lease vs. loan for financing expansion rate of return analysis for computer acquisition profit and loss analysis operating budget models consolidation of divisional budgets capital expenditure analysis integration of tax forecasts Sales Management risk analysis sales forecasting sales budget planning market analysis market analysis Operational Management route planning for trains manufacturing process planning monitoring expense levels unit volume and users budget control monitoring expense levels, unit volume, and revenues the second s Project Management Portfolio Management Loan and Deposit Analysis (Bank Management) Contract Negotiations (Monte Carlo Risk Analysis) Portfolio Management Purchasing Models Space Planning Medical Diagnosis and Therapy Recommendations Knowledge-based Instruction the definition of a first

Figure 1. Applications of the Surveyed DSS



Figure 2. The EPIC Teleterminal (printed with permission of Bell Laboratories)

namesto discontrata andre and

S MEAS - JSJA INCOMONIAN LINE 30 US SHO N A LEBOR

Sus erections fistoricate a local de

VE ALON BELOCHTERED DA I TOR - PYEA

The EPIC system provides four types of service: enhanced calling, electronic messaging, information and news retrieval, and calendar keeping. The enhanced calling services consist of a personal directory, company directory assistance, automatic dialing and a most recent calls list. When a user selects a listing from a directory or list of most recent calls and pushes a button, the number is automatically dialed.

The electronic messaging system allows the user to easily create messages by following the instructions displayed on the teleterminal screen. Once the message is created the user can check it for spelling, change it, specify a return receipt, specify a distribution list, and when completed send it simply by pushing a button.

Upon receiving an electronic message, the EPIC user is alerted by a brief tone and status light. The message can be read, printed, discarded or replied to with either another electronic message or a telephone call. To reply with a call, the user simply presses a button and EPIC automatically dials the telephone number given in the message. Since secretaries usually receive an executive's telephone calls, the EPIC system allows them to send "memos of calls" to the executive.

The information and news retrieval service provides the capability to build and maintain personal information files, to easily access corporate information files and readily gather information from external news sources. It also provides access to airline schedules to support travel planning.

EPIC also provides a calendar management service. Again the user simply accepts prompts from EPIC to construct or modify calendars. When the executive turns on his terminal, his daily calendar is displayed.

These services provide decision support to executives by streamlining the exchange of information among decision makers and providing easy access to the latest corporate and external business information.

EPIC users also have access to information produced by the ADMIN and FACTS systems. ADMIN provides a weekly snapshot of the corporate data base. Under development are personnel and space management systems, ad hoc query and reporting for space planning, and personnel tracking. FACTS is used for financial analysis and planning. It uses the ADABASE DBMS to organize corporate data and will eventually have a modeling language. Both systems are menu driven and have color graphics capabilities. Since ADMIN and FACTS are still under initial development, management evaluations were not available.

#### The Boeing Company (Executive Information Services)

The Estimation Information System was developed by Boeing during the early 1970's to track aerospace project requirements. Projects in which EIS was used include AWACS and the Boeing 747. The name of the system was changed to Executive Information Services (EIS). Boeing Computer Services maintains EIS and sells the software. According to their marketing group there are more than 2000 users of EIS in more than 400 organizations. Managers interviewed used EIS for budgeting, financial planning, manpower planning, forecasting, rate analysis, cost scheduling, performance tracking, and ad hoc query. The project management systems used are Program Estimation and Tracking (PET) and Project Software and Development Inc.'s PROJECT-2.

Meta-DSS for building specific DSS. EIS is a It consists of a centralized Interface Manager which interprets the EIS command language to interface dialog, modeling, database, and foreign language components. The dialog component consists of device drivers and an I/O manager to interface to crts, 327X terminals, and graphics terminals. The modeling components consist of direct command interfaces to a modeling language , report writer and graphics functions. It is possible to call routines from the financial and statistical library directly or from the modeling language. The database components consist of EZDAIA, a menu driven database builder; XCEL, a program for extracting data from external sources; and the EIS n-dimensional database. There are also interfaces to database management systems such as FOCUS, INQUIRE, and RAMIS-II. Finally, there are interfaces to "foreign" languages such as, APL, TELEGRAPH, and SAS. The command HS links the user back to the host Interfaces to foreign languages and system commands. database systems are via the CALL command. Most of the commands can be abbreviated to two characters, e.g. HS, CS and XS. EIS has a HELP function, an online training manual and a C.A.I. tutorial on EIS.

At Boeing the system operates on the IBM 3081 under VM/CMS. Graphics terminals include the IBM 3279, Tektronics 4021, Intelligent Systems Intercolor, Ramtek 6212 and the Hewlett-Packard 8-color flatbed plotter.

# The Citizens and Southern National Bank (Branch Bank Budget Management System, Mortgage Portfolio Pricing Model)

Citizens and Southern is the largest banking institution in Georgia. To support branch bank manager's control of expenses, the corporate planning group developed a Budget Management System. The system is written in FORESIGHT [United Computing Systems (1981)], a financial planning language, and operates on an IBM 3033 mainframe. Data required for the system is extracted from the general ledger sytem via a data extraction routine written by the data processing group. It is possible to ask "what if" questions using this system, but branch mangers do not know FORESIGHT's financial planning language well enough to use it for this purpose. When managers need additional analyses that require a knowledge of FORESIGHT, they ask the staff in the comptroller's office to perform these analyses.

To address these problems and attain greater flexibility, it is planned to obtain Management Science America's F&M, a financial planning system, which interfaces to the MSA General Ledger System that C&S Bank currently uses [see discussion of F&M under Management Science America this section]. in It is also planned to link IBM PC's to F&M using 3278 terminal emulation and IBM's System Network Architecture (SNA). Access will be via leased lines to provide additional system security.

Corporate Planning also used VisiCalc [Visicorp (1982)], on an IBM PC to build a Mortgage Portfolio Pricing Model for the Mortgage Lending Division of the bank. The VisiSeries is a product of VisiCorp, San Jose, California. It consists of VisiCalc Advanced Version (an electronic spreadsheet), VisiTrend/Plot (a trend-analysis and graphics program), VisiTerm (a communication program), Visifile (a program), and VisiSchedule (a project file-management management program). VisiCalc, short for Visible Calculator, was developed in 1978, and was the first of the electronic spreadsheets for microcomputers. The original VisiCalc was a self-contained program, which means that when data from the spreadsheet file was needed for a report or graphical representation, the user had to manually rekey the data. Conversely, when data from other files was needed for VisiCalc, it had to be reentered. The VisiSeries, however, uses a data file format called Data Interchange Format (DIF) that allows programs in the VisiSeries to exchange data without modification or reentry.

Corporate Planning uses a variety of tools for ad hoc modeling and reporting. SAS, a statistical analysis package [SAS Institute (1982)], is used for loan and deposit analysis. APL/Data Interface is used for on-line query and trend analysis. FORESIGHT and VisiCalc are used for ad hoc product reports.

# The Coca-Cola Company (Peel the Onion, GPGS, Field Office Financial Planning)

The Coca Cola Company markets soft drinks, coffee, tea, and citrus products, and recently acquired Columbia Pictures. Coca-Cola had sales of 6.25 billion dollars in 1982. The DSS efforts studied are in the corporate offices and the largest division, Coca-Cola USA. In the corporate office there is a Management Sciences Group and Client Support Group reporting to the Director of Corporate Information Systems who reports to the Chief Financial Officer of the firm. In Coca-Cola USA there is a Decision Support Systems Group in the Management Information Systems Department.

The USA division originated the search for financial planning tools in 1977. A number of financial modeling systems were benchmarked against a budget application. The Interactive Financial Planning System (IFPS) [Execucom (1983a)] was selected. SAS was selected for statistical analysis and FOCUS [Information Builders (1982)] for data management. The Mainframe is an IBM 3081 with TSC. In the following paragraphs the interface of various DSS functions through the operating system commmand language and the IFPS command language is described.

At the operating system level the command "IFPS" will invoke the Financial Planning System, the command "FOCUS" will invoke the database management system and the command "SAS" will invoke the statistical analysis system. Two other commands at the operating system level, DATASPAN and SENTRY, will invoke optional systems available from Execucom. DATASPAN is a file transformation program that allows the user to extract data from existing corporate or external files. It creates an IFPS data file from a sequential character file. SENTRY is used to create a set of prompts and validation checks for data entry to IFPS data files [Execucom (1981)].

Typing the command IFPS at the operating system level puts the user at the executive command level of IFPS. At this level five types of commands can be given: subsystem commands, file manipulation commands, execution commands, an optimization command, and link commands.

The subsystem commands are: MODEL (enter the modeling language subsystem), CONSOLIDATE (consolidate multiple models or one model using multiple data files, and enter the modeling language subsystem), DATAFILE (enter a subsystem for creating and editing datafiles), REPORT (enter a subsystem for editing and creating reports), FORMULA (enter a subsystem for creating, editing and testing formulas), STRUCTURE (enter a subsystem for creating and editing structures), SUBTASK (enter a subsystem for creating or editing subtasks), and CMDFILE (enter a subsystem for creating and editing command files). In the model subsystem there is an optional graphics function invoked by the command GRAPH.

The file manipulation commands allow the user to command such tasks as listing, copying or merging files. The execute commands allow the user to execute a command file, execute a FORTRAN program or quit IFPS. The optimization command is "OPTIMUM". It is an IFPS option that allows optimization of linear or nonlinear IFPS models [Execucom (1983b)].

The link commands are the commands for interfacing IFPS to other DSS tools. IFPS contains simple predefined

statistical functions. For more complex statistical analyses there are interfaces to SAS and SPSS. For example, the comand LINK SAS will allow the creation of an IFPS data file from a SAS data set. From SAS the user can also link to an IFPS datafile. There are also interfaces from IFPS to the database management systems INFO and FOCUS. For example, the command LINK FOCUS will access FOCUS data and create IFPS data files

LINK TS is an interface to Trend-Spotter (a Computer Pictures product) to have IFPS data files displayed on a color graphics terminal or printer. Trend-Spotter also supports the production of color transparencies and 35 mm slides [Execucom (1982b)]. Finally, there is an interface via the LINK command to McCormack and Dodge's General Ledger System GPLUS.

Applications of IFPS at Coke include Peel the Onion, a profit and loss model, operating budget models, a present value model to evaluate cash versus lease versus loan alternatives for financing bottler plant expansion, a rate of return model used for making decisions about purchasing versus leasing computer equipment, and marketing models [Mingledorff (1981)]. IFPS is also being used to ask "what if" questions, perform goal seeking tasks, and to do sensitivity analyses.

TkS-80 microcomputers and DSS/F are being used in some of the field offices for financial planning. Micro-Decision Support System/Finance (DSS/F) is a product of Ferox Micro Systems, Arlington, Virginia. DSS/F is also distributed by Addison-Wesley. Other tools from Ferox that enhance development of DSS on microcomputers are LogOn (for data communications), Consultants Edge (for building menus), GraphPower (plotter graphics for the HP7470A plotter), and the Data Machine (TDA) (data management software).

#### Georgia Power (Executive Information System)

Georgia Power is an electric utility company owned by Southern Services, a holding company that also owns Gulf Power, Mississippi Power and Alabama Power. The Southern Company had revenues of 4.9 billion dollars in 1982. At Georgia Power there are three projects of relevance to this study: the Information Support Center, the PC laboratory and the Executive Information System.

The Information Support Center provides analysts with capability for ad hoc reporting, query and graphics. There are 10 users in 5 departments. Applications include equal employment opportunity and employee promotion, deposit analysis, and labor negotiations.

The hardware consists of an IBM 4331 with IBM 3270 terminals to an Amdahl at Southern Company's Data Center.

The operating systems are MVS on the Amdahl and VM/CMS on the IBM 4331. The software is IBM's VS/APL, APL/DI, and A Department Reporting System (ADRS) [IBM (1979, 1980)]. The original proponent of the Information Support Center was the former General Manager of Management Information Support.

The PC Lab is a laboratory with IBM PC and Apple computers to train staff in the use of microcomputers and software such as Visicalc.

An Executive Information System is being implemented. In the system planning stage executives were asked what information they needed in their office. The five major responses were: 1) a proforma income statement, 2) an automated "to do" list, 3) performance indicators, 4) employee records for their area, and 5) an executive calendar. The proforma income statement (updated daily) is being implemented using EMPIRE [Applied Data Research (1982)]. The automated "to do" list is being implemented in ADKS. The Performance Indicator Tracking System (PITS) is being implemented in ADRS. PITS uses bar graphs to show projected versus actuals. The employee query capability is be implemented in APL/DI and ADRI business graphics. being The Executive Calendar will be provided by IBM's Professional Office System (PROFS). The Executive Information System will also provide online access to scenarios from the corporate model.

Georgia Power also uses a number of large models that are run in batch mode to which management has online, but not interactive, access. Included are a corporate model, a model for mix of generating plants, and a rate case model.

#### Gold Kist, Inc. (Peanut Purchasing Model)

Gold Kist is a farmers cooperative. There are four operating groups -- Poultry, Agriservices, Agricommodities (grain), and Agriproducts (pecans, peanuts).

Initially, timesharing was used to provide financial and modeling capabilities. Three years ago the decision was made to purchase a financial planning language to support preparation of budgets, spread sheets and proforma income statements. The criteria used to select a financial planning language were user friendliness, a nonprocedural modeling language and optimization capability. Real Decisions report on Financial Modeling Languages was used as a source of information for the evaluation and IFPS was selected [Real Decisions (1982)].

One of the DSS applications was a peanut purchasing model used by buyers. The model includes variable and fixed costs at buying stations and shelling plants. The goal of the model is to determine how much can be paid for peanuts given the objective of maximizing revenue. This model gives the buyer a "justification" for his offers. He can say to the farmer "this is why I can only pay you this much". Another benefit is accurate cost estimates to help in protecting the margin on large purchases.

Gold Kist purchased IFPS from Execucom two and one/half years ago. They also use Execucom's OPTIMUM and SAS. Their hardware is an IBM 4300 under CMS. For graphics they use Picturepak on an 8 pen color HP plotter.

Other IFPS applications at Gold Kist include contract negotiations using Monte Carlo analysis to determine risk of agreeing to contractual terms, engineering design (heat transfer models), and budget preparation by farm managers in the field.

### Lockheed-Georgia (GENPLAN, MIDS)

Lockheed-Georgia is an aircraft manufacturer whose aircraft include the Cl30 Hercules, the Cl41, the C5A and C5B. Two DSS were studied -- the Management Information and Decision System (MIDS) and the manufacturing planning system GENPLAN.

MIDS is an online color graphics management reporting system used by top executives at Lockheed-Georgia. MIDS was constructed to meet the needs of the CEO for online access to corporate information. The hardware is a DEC PDP11/34 and Intelligent Systems intercolor terminals. There is a menu interface to the functions available. Other features are executive mail and calendar facilities. Access to reports and other displays can be accomplished by the user with only four keystrokes. Displays include plots, barcharts, and graphical display of production line status. A major benefit cited by management is their immediate access information. to the same up-to-date company Planned enhancements include automatic data extraction from the corporate database, financial planning and ad hoc query capabilities.

GENPLAN [Waterbury (1981)] is a knowledge-based system that supports optimal planning for manufacturing detailed parts, subassemblies and major assemblies for aerospace products. It is used by manufacturing engineers, manufacturing planners, time standard engineers and production engineers. A manufacturing planner uses the system by describing the detailed part, subassembly or assembly that is to be manufactured. GENPLAN uses its knowledge base to generate the fabrication or assembly operations that are required to produce the part or subassembly. The GENPLAN knowledge base includes information on manufacturing operations, tools, manufacturing standards, time standards, cost centers and load centers.

#### Management Science America (Forecasting and Modeling)

Management Science America (MSA) is the world's largest independent software company with 1982 revenues of over \$100 million. Their Forecasting and Modeling system (F&M) is integrated with their General Ledger system. The General Ledger system also interfaces to accounts payable, inventory control, accounts receivable, personnel and payroll systems (see Fig. 3). F&M operates on the IBM 3300 under TSO/OS, ICCF/DOS and CMS/VM. Interactive access is supported by the Communication Interface Control System (CICS).

consists of a modeling language, reporting F&M capabilities, data files and color graphics. MSA uses F&M for sales forecasting, budgeting and financial analysis. Models are developed by Financial Analysts and used by Product Managers. The development of F&M was funded by inviting the General Ledger clients to be co-developers of F&M. There 17 codevelopers initially including Qantas Airways, Bank of Ohio and Logarithmic. Users of the General Ledger who had agreed to be co-developers were asked what features they would like to see in a financial forecasting and modeling system. These features were used to develop system specifications which were distributed to the codevelopers for feedback. The co-developers participated in design walkthroughs, testing, user training and reading drafts of the documentation. When the first verstion was released to co-developers, feedback was used to enhance the F&M was first installed in 1979 and is in its system. fourth release. User feedback indicated demand for enhanced budgeting capabilities which resulted in EASYPLAN. The Sales group at MSA also indicated the need for color graphics. At the last user meeting (INTERACT) there were approximately 1000 participants.

Peachtree Software, a subsidiary of MSA, markets a collection of office productivity software packages for microcomputers called Executive-PeachPak. This package includes: Business Graphics, PeachCalc (an electronic spreadsheet), Peachtext (word processing and spelling checker), Peachlink (linking microcomputers to mainframes), and a mail list manager.

# Norfolk Southern Corporation (Train Dispatcher, Pattern Recognition System)

The Norfolk Southern Corporation is the nations fifth largest railway system. Norfolk Southern is the holding company for the Southern Railway and the Norfolk and Western Railway which merged in June, 1982.

A computer-aided train dispatching DSS was developed by the Operations Research Department [Sauder and Westerman(1983)]. The DSS is a minicomputer-based information system with an online capability for optimal route



Figure 3. Interface of F & M to General Ledger

planning (see Fig. 4). It was developed to support dispatchers in Southern Railway's Alabama Division. Since the system was put into operation in September 1980, train delay has been 15% lower, reflecting an annual cost savings of over \$300,000. The dispatching support system is now being expanded to all other Southern Railway dispatching operations.

- The system was developed via the following steps:
- (1) A simulator was built to simulate field operations.
- (2) Various ways of inputting information to the system were defined. (There was no requirements or design document).
- (3) A prototype was constructed with static displays.
- (4) A train dispatcher was shown the displays and asked if this was what she wanted. Her suggested modifications were incorporated into the displays. This process was repeated several times.
- (5) Parts of the information system were developed.
- (6) The displays remained the same but the internal structure changed considerably.

The Systems Group has developed a marketing DSS called the Pattern Recognition System. A Nova Eclipse is used to extract data from a consolidated revenue history file. The extracted data is stored on a floppy disk and input to an Intelligent Systems microcomputer. A DSS on the microcomputer produces scatter plots from the data profiles. Colors are used to differentiate commodity types. The system has the capability to retrace the previous steps in the analysis, and to stop and restart the analysis.

Initially, the users knew that they wanted scatter plots but could not give much more detail. Development progressed by showing prototypes to users and asking for their suggestions. The prototype went through ten versions. This approach was used because the developers past experience indicated that customers did not readily understand specifications but readily understood exhibits. This prototype has been replaced by SAS and SASGRAPH on a mainframe computer.

## Northwest Industries (Executive Data Base System)

Northwest Industries, with a gross income of about \$3 billion in 1982, is a conglomerate of nine operating companies. The Executive Data Base System was conceived in 1976 and initially consisted of a prototype system that provided access to about 70 standard reports and limited forecasting, calculation and statistical analysis functions. EXPRESS was interfaced to this basic system in 1977 to add modeling and statistical analysis capabilities. The system is used by top management, including the Chief Executive Officer, financial analysts and corporate planners. The data bases include historical data on each of the operating



Figure 4. Time-Distance Display from Train Dispatching DSS. A time distance graph displaying train movement through a five siding network in a four and one-half hour time frame. Eastbound trains move diagonally from left to right meeting westbound trains where lines intersect. divisions, economic time series and external data bases such as Standard and Poor's Compustat.

EXPRESS is essentially a DSS Kernel or Meta-DSS--that is, a collection of DSS tools and a dialogue manager that can be used for rapidly building specific decision support systems. Sprague and Carlson (1982) call a meta-DSS a DSS Generator. EXPRESS uses the VM/CMS operating system on IBM equipment. This operating system provides a virtual memory capability.

The EXPRESS dialogue manager is illustrated in Fig. 5. A command interpreter interprets commands from the user by using a dictionary. The dictionary contains entries for command names, data representation names, subscripts, etc. Associated with each entry are expected parameters, interactive menu displays, and sequences of commands to be executed based on which alternative in the menu the user selects. All information from the user must pass through the command interpreter. Commands are requests for execution of a routine in the system library, modeling system, graphics system, statistics library or report genertor. The command interpreter uses the file system of the operating system to access these routines. The routines can display information directly on the terminal but all information passed to them interactively must be interpreted by the command interpreter.

### Rollins (Financial Planning, Orkin Market Analysis)

Rollins consists of five divisions: Orkin Exterminating Company, Media (outdoor advertising and radio), Rollins Protection Services, Patterson Oil and Gas, and Rollins Lawn Care. Rollin's revenues in 1982 were over \$500 million. Rollins uses IFPS on CYBERNET and FOCUS and EXPRESS on TYM-SHARE and an HP 7220 graphics terminal.

There are about 75 users of IFPS. IFPS is used for acquisition analysis, budgeting and consolidation of corporate plans, divisional operations control, project management and financial planning. Acquisition analysis includes determination of rate of return and payback period on investment. "What if" questions are asked such as "If one division grows by a 5% rate rather than 10%, what is the financial impact on Rollins?" Such factors as expense levels, unit volume, pricing and revenues are monitored to determine impact on divisional operations. IFPS is also used to prepare sales reports, to analyze staffing levels versus sales volume, and to analyze capital expenditures.

The Orkin Division of Rollins uses FOCUS and EXPRESS for market analysis and sales budgeting. FOCUS is used to capture data to put into EXPRESS. Data files used include a 5 year sales history for 300 branch locations and two services, and purchasing power of counties. Data is aggregated



# Figure 5. Dialog Manager of EXPRESS

by counties, designated marketing areas, districts, regions and advertising categories.

### Sante Fe International (Budget Planning)

Santa Fe International is an oil and gas exploration, drilling, engineering and construction company with offices in twenty-one countries. The corporate planning group uses System-W [Comshare (1982)] for consolidation of divisional operating forecasts, integration of tax forecasts, and ad hoc financial analysis. They initially developed their applications on a time-sharing system before purchasing the product. They are currently developing a five-year strategic planning model.

System-W originally operated on the Xerox SIGMA 9 under the Commander-II operating system. It is now available on IBM mainframes under VM/CMS and OS/MVS. System-W is written in PASCAL.

#### Southern Bell (Corporate Planning)

Southern Bell is a telephone utility with subscribers in Georgia, Florida, and North and South Carolina. AT&T made IFPS available to its operating companies in 1979. At Southern Bell IFPS is used by corporate planning on an Amdahl 470V8 under VM/CMS.

Applications include spreadsheets, a study to determine costs of providing extended area service, zero-based budgeting, Monte Carlo studies of market share, and various ad hoc projects.

As compared with the use of the services of a data processing support groups or the use of a programming language such as FORTRAN, COBOL or BASIC, the following benefits of IFPS were cited: greater flexibility in producing spreadsheets and financial models; faster turn around in preparing budgets; less overhead for modifications; capabilities for "goal seeking", "what if" analyses and consolidations.

# Stanford University Medical Center, Medical Computer Science

MYCIN is a special purpose decision support system developed by Shortliffe (1976). EMYCIN [van Melle (1980)], a generalized MYCIN, is a system for building expert systems that is written in InterLisp and operates on DEC hardware under the TENEX or TOPS20 operating systems. The MYCIN knowledge base was recreated using EMYCIN. EMYCIN is used primarily by medical and computer science students but has had some nonmedical applications. MYCIN provides physicians with expert advise in diagnosing bacterial infections. The expert advise is represented as production rules in a knowledge base (see Fig 6). During a dialogue with a physician, MYCIN askes the physician for clinical, laboratory, and historical data. Then MYCIN "reasons" using the knowledge base as to the organisms most likely to be causing the infection. A sample dialog is shown in Fig 7.

Current projects in the Medical Computer Science Department are NEOMYCIN [Clancey and Letsinger (1979)] and GUIDON [Clancey (1979)]. NEOMYCIN is a project to restructure the MYCIN knowledge base to explicitly separate diagnostic strategy rules from specific medical knowledge. GUIDON is a knowledge-based computer assisted instruction program that uses the NEOMYCIN knowledge base.

# White House, Office of Planning and Evaluation (Presidential Scheduling)

This office is responsible for strategic planning at the white House. Strategic plans involve decisions on foreign policy, domestic policy and national security. The objective is to give a perspective on these various policy areas.

Several databases and models have been developed to support scheduling the President's time. This is accomplished by preparing forecasts, analyses, and decision papers that assess positions on issues confronting the President. Recommendations are made to the President's scheduler. One of the models that is used is a modified Petri Net model relating events taking place in the Congress, the bureaucracy, and on the domestic and international scene. Databases that are used include the OCA tracking system (which tracks issues as they move through the bureaucracy), cabinet minutes, presidential calendar history, and the New York Times database.

The programming languages FORTRAN 77, PL/1, BASIC and PASCAL are used to develop models. SPSS is used for statistical analysis, Datatrieve for ad hoc reporting, keyword-in-context and word processing for text processing. The hardware used is a DEC VAX 11/780 'time sharing system with VT-125 terminals and an HP 8 color plotter.

Other White House users of modeling, ad hoc query and data analysis for decision support include the Council of Economic Advisors, the Office of Science and Technology, and the National Security Council, but staff in these areas were not interviewed.

2.4 Comparison of Surveyed DSS

	Rule	27	1000 C
619:001	If	1)	the gram stain of the organism is gram negative, and
1		2)	the morphology of the organism is rod, and
		3)	the aerobicity of the organism is anaerobic,
	Then:		There is suggestive evidence (.7) that

entrand to entrans to balance and

### Figure 6. MYCIN Production Rule

N ON

1.7

清出

#### PATIENT-248

Patient's name: (first-last)

\*\* C.R.

2) Sex

\*\* MALE

3) Age:

\*\* 52

4) Have you been able to obtain positive microbiological information about a possible infection of C.R.?

**\*\* YES** 

#### INFECTION-1

5) What is the infection)

\*\* ENDARTERITIS

The first significant organism from this blood culture (CULTURE-1) will be referred to as:

#### ORGANISM-1

- 9) Enter the identity of ORGANISM-1:
- **\*\*** UNKNOWN
- 10) Is ORGANISM-1 a rod or coccus (etc):
- \*\* ROD
- 11) The gram stain of ORGANISM-1:
- **\*\*** GRAMNEG
- 12) Have there been POSITIVE cultures yielding organisms about which you will NOT be seeking advice?

\*\* YES

INFECTION-1 is ENDARTERITIS with BACTEREMIA

<Item 1> E. COLI [ORGANISM-1]

- <1tem 2> SALMONELLA (species unknown) [ORGANISM-1]
- <Item 3> KLEBSIELLA-PNEUMONIAE [ORGANISM-1]
- <Item 4> PSEUDOMONAS-AERUGINOSA [ORGANISM-1]

<Item 5> ENTEROBACTER [ORGANISM-1]

<Item 6> PROTEUS-NON-MIRABILIS [ORGANISM-1]

[Rec 1] My preferred therapy recommendation is as follows:

In order to cover for Items <1 3 4 5 6>:
 Give: GENTAMICIN
 Dose: 128 mg (1.7 mg/kg) q8h IV (or IM) for 10 days
 Comments: Modify dose in renal failure

In order to cover for Item <2>: Give: CHLORAMPHENICOL Dose: 563 mg (7.5 mg/kg) g6h for 14 days Comments: Monitor patient's white count Do you wish to see the next choice therapy?

\*\* NO

Figure 7. MYCIN Dialog

In this section the system characteristics, development tools and development methodologies of the surveyed DSS are compared with regard to their impact on the quality of the end system and the ease of implementation.

#### 2.4.1 Hardware, Operating Systems, and Development Tools

The types of computer hardware and operating systems on which the DSS operate is shown in Fig. 8. Also shown are the tools used for developing the DSS.

Financial modeling or planning languages are the primary "programming" language tool. In some cases the financial planning language is used by financial analysts or corporate planners and the results are given to management. In this case the financial planning language is a major component of the DSS. In other cases the financial planning language is used to build a DSS used by managers. In this latter case the language is not used by the managers themselves, so the language is not a component of the DSS, but a tool used to build the DSS.

#### 2.4.2 Dialog and Display

Two aspects of the DSS-user interface are relevant: the dialog style and the display of information (see Fig. 9). The types of dialog style observed included: command language, command files, menu, menu with function keys, menu with soft keys, question-answer, fill in the blanks, and combinations of these.

A uniform dialog style contributed to ease of use--for example, an integrated command language that allows the user to access all of the functions without having to learn different languages. The DSS products from Ferox Micro Systems can be accessed through a menu control structure (see Fig. 10).

The command language type of dialog was the least desirable for the infrequent user. The easiest dialog styles for the infrequent user were the menu, menu with function keys, menu with soft keys, fill in the blanks or question-answer. The more frequent user often expressed a preference for the command language and command file style of dialog.

By the display of information is meant the reporting and graphics capabilities of the DSS. The reporting function was supported by default reports, a report writer, and word processing. Default reports contributed to ease of use in that the user could view information without having to be concerned with the task of formating the information. When customized reports were desired one could then use a report writer. If it was desired to combine reports or graphs with text, this could be accomplished with a word processor.

DSS NAME	COMPUTER	OPERATING SYSTEM	DEVELOPMENT TOOLS
EPIC	DEC PDP 11/70	UNIX	Programmers Workbench, C
EXECUTIVE INFOR- MATION SERVICES	IBM 3081 IBM 3033	VM/CMS OS/MVS	FORTRAN, ASSEMBLY
BRANCH BANK MANAGEMT	IBM 3033	VM/CMS	FORESIGHT
MORTGAGE PORTFOLIO	IBM PC	PC DOS	VisiCalc
PEEL THE ONION, GPGS	IBM 3081	TSO VECTORIO	IFPS, SAS, FOCUS
FIELD OFF. FIN. PLAN.	TRS-80	TRS DOS	DSS/F
EXECUTIVE INFOR- MATION SYSTEM	IBE4331	VM/CMS	VS/APL, APL/D1, ADRS, EMPIRE, PROFS
PEANUT PURCHAS.	IBM 4300	CM/CMS	IFPS
GENPLAN	IBM 3081	VM/CMS	FORTRAN, COBOL
MIDS	DEC PDP 11/34	RSTS	BASIC PLUS
F & M	IBM 3300	TSO/OS, VM/CMS	CICS, COBOL
TRAIN DISPATCHER	DATA GEN. 130	AOS	FORTRAN
PATTERN RECOG- NITION SYSTEM	Intelligent Systems Intercolor	Intelligent Systems OS	MICROSOFT BASIC
EXECUTIVE DATABASE	IBM	VM/CMS	EXPRESS
ROLLINS FIN. PLAN.	CYBERNET		IFPS
ORKIN MARKET ANAL.	TYMSHARE		EXPRESS
SANTE FE BUDGET	XEROX SIGMA 9	COMMANDER II	System-W
SOUTHERN BELL	Amdahl 470VB	VM/CMS	IFPS
MYCIN	DEC 20	TENEX, TOPS20	Interlisp
PRESIDENTIAL SCHEDULING	DEC VAX 11/780	VMS	SPSS, Datatrieve KWIC, Inquirer PASCAL, BASIC

end reader and the same of the base and

Figure 8.

HARDWARE, OPERATING SYSTEMS AND DEVELOPMENT TOOLS

DSS NAME	DIALOG STYLE	REPORTING	GRAPHICS DEVICE	GRAPHICS REPRESENTATION
EPIC	menu with soft keys	default reports	none	none
EXECUTIVE	command	report	IBM 3279	line graphs
INFORMATION	language,	writer	HP 8 color	histograms
and a second second	menu		plotter	pic charts
BRANCH BANK	command	default	printer	bar charts
MANAGEMENT	language, menu	reports, report writ <b>er</b>	graphics	line graphs
MORTGAGE	single	print	printer	bar charts
PORTFOLIO	letter	spreadsheet	graphics	pie charts
	commands			line charts
PEEL THE	command	default	printer	bar charts
ONION, GPGS	language	reports, report writer	graphics	line, pie, bubble, scatter, ridge
FIELD OFFICE	menu	guick &	printer	bar charts,
FINANCIAL	command	dirty	graphics	pie charts
PLANNING	file	report writer		líne graphs
EXECUTIVE	command	report	printer	line & bar
INFO SYS	language	writer	graphics	graphs
PEANUT	command	default	CRT	bar charts
PURCHASING	language	reports,	graphics	line, pie,
	Server a service fragment of	report		bubble
	Co. red. 1	writer		scatter, ridge
GENPLAN	menu	default	none	none
C. P. C.	command	reports,		
	language	(fabrication	diters table lines	
	fill in	& assembly		
	the blank	operations)		
MIDS	menu	default	Intelligent	histograms
	color	reports	Systems	line graphs
			Intercolor	alant lavout

Figure 9. 1

Dialog,

Reporting and Graphics Characteristics

DSS NAME	DIALOG STYLE	REPORTING	GRAPHICS DEVICE	GRAPHICS REPRESENTATION
F & M	command language	default reports	IBM 3279	none
TRAIN DISPATCHER	menu function keys, fill in the bla	default reports ink	color graphics plotter	time/distance display
PATERN RECOGNITION SYSTEM	menu	none	Intelligent Systems Intercolor	scatter plots bar charts
EXECUTIVE DATABASE SYSTEM	command letter language	display model report writer	IBM 3279	line graphs, histograms, pie charts
ROLLINS FINANCIAL PLANNING	command language	default reports, report writer	printer graphics	bar charts line, pie, bubble, scatter, ridge
DRKIN MARKET ANALYSIS	command	display model report writer	HP 7220 graphics	line graphs pie charts histograms
SANTE FE BUDGET PLANNING	command language, menu	report writer	IBM 3279 color CRT	bar charts, pie charts, line graphs
SOUTHERN BELL CORPORATE PLANNING	command language	default reports, report writer	printer graphics	bar charts line, pie, bubble scatter, ridge
MYCIN	question/ answer	none	none	none
PRESIDENTIAL SCHEDULING	menu command language	word processor datatrieve	VT125 graphic HP 8 color plotter	histograms line graphs

Figure 9. Dialog, Reporting and Graphics Characteristics (cont)
and our graphics werentle frage experisive, Color cat and \*\*\*\*\*\*\*\*\*\* + + + tort is (1) +------MAIN MENU------+-+ + 1. CREATE/CHANGE MODEL FILES + 2. SOLVE MODEL + + 3. VIEW/PRINT RESULTS + 4. ASK "WHAT IF?" 5. + PERFORM "GOAL SEEKING" ANALYSIS + 12 La 10 + 2 CONSOLIDATE/MANIPULATE MATRIX DATA 6. + + 7. CREATE GRAPHICS + + 8. CHANGE MATRIX DIMENSIONS + 9. MANAGE FILES + + + 1900 SELECT (1 ..... 9 OR O TO LEAVE MENUS): + + + + + \*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\*\*\* + + + + -----CREATE/CHANGE MODEL FILES----+-+ + + 1. CREATE/CHANGE LOGIC FILE + + 2. CREATE/CHANGE DATA FILE + + 9199 3. CREATE/CHANGE REPORT SPEC FILE + CREATE A DATA INPUT WORKSHEET + 4. + + 5. CREATE/CHANGE JOB FILE CREATE/CHANGE GEN FILE + 6. + + + 5 CREATE/CHANGE GRAPHIC SPEC FILE 7. + 8. + PRINT FILE + COMPILE LOGIC FILE + 9. + + + + SELECT (1..... 9 DR 0 TO QUIT): + + + + \*\*\*\*\*\*\*\*\*\* Ness houses a sport see and a single industry and interior barres in and

in by tone "beau of the sector sector back of the beau to beau to

printereg aphies. Sht. grabhica, pic spinter, "doa'n or graphies states of picture. "tipker of spinter.

Figure 10.

ON SHOTSY DO

DSS/F's Menu-oriented Dialog

der iston storen und an here filen hole.

The types of graphics devices that were used included printer graphics, crt graphics, pen plotter, color crt graphics, and color printer or plotter. Printer graphics and crt graphics were the least expensive. Color crt and color printer or plotter were better for presentations but more expensive.

Graphics representations included: line graphs, bar charts, pie charts, scatter diagrams, tables, line graphs, time distance displays, and floor layouts. Color graphics software that provided a capability for customized graphs added greatly to the attractiveness of the system to managers.

With the slide show feature available with DSS/F a color monitor can be used for presentations to small groups. A large screen projector TV can be used for larger groups, and a color graphics printer or plotter for hardcopy.

Output of text or graphics hardcopy can "tie up" a single-processor microcomputer. Many of the microcomputers will have a processor devoted to spooling output. If the microcomputer does not have a processor dedicated to spooling, this feature can be added to any micro by the addition of a "smart buffer" [Data Match (1982)].

## 2.4.3 Modeling

One the most important functional capabilities of the the observed DSS was the provision of a powerful but easy to use modeling language. These modeling languages can be characterized by the types of data objects in the model, the types of relations and operations in the model, the functions provided for use in the modeling language, the types of modeling language statements, and in terms of the control strategies available in the modeling system (see Fig. 11).

The most common data object for financial models was two to nine dimensional arrays with scalar or text values. Dimensions represent logical classifications of data. For example, in Fig. 12 is shown a 3-dimensional model from EIS. One dimension consists of blocks, another of cost elements, and a third of items. Additionally, their may be family names, e.g. actuals versus planned, in a dimension. There may also be logical relationships among data items. For example, sales in dollars = sales volume \* unit price.

The relations and operations used in these models were arithmetic, logical and relational. The functions supplied were mathematical and financial. The statements had arithmetic, logical and relational expressions.

If the model was procedural, it resembled a programming language in that the statements were sequential with decision statements such as if-then-else and iterative statments such as while-do. If the model was nonprocedural it

DSS NAME	DATA OBJECTS	RELATIONS OPERATIONS FUNCTIONS	STATEMENTS	CONTROL STRATEGIES
EPIC	none	none	none	none
EXECUTIVE INFORMATION SERVICES	2 to 5 dim. arrays scalers	arithmetic logical mathematical financial	control equations	solution of simultaneous equations
BRANCH BANK MANAGEMENT	two dim. matrix integers or reals	arithmetic logical accumulate average round	line, column period arithmetic	what if consolidation
MORTGAGE PORTFOLIO	two dim. matrix	arithmetic	screen oriented editor row column reference	what if anal.
PEEL THE ONION, GPGS	two dim. matrix	arithmetic mathematical financial	equations	sim. eqn solver goal seeking sensitivity analysis
			onia Aa	what if anal. Monte Carlo risk anal. forecasting
FIELD OFFICE FIN. PLAN.	two dim. matrix	arithmetic	line oriented editor row column reference	consolidation
EXECUTIVE INFORMATION SYSTEM	two dim. matrix scalers	arithmetic relational mathematical financial	assignment control	consolidation sensitivity what if goal seeking
	inter i			

1

Figure 11. Modeling Characteristics

199 27

DSS NAME	DATA OBJECTS	RELATIONS OPERATIONS FUNCTIONS	STATEMENTS	PROGRAMMED CONTROL STRATEGIES
PEANUT PURCHASING	two dim. matrix	arithmetic mathematical financial	equations	simul. eqn solver goal seeking sensitivity analysis what if anal. Monte Carlo risk anal. forecasting
5ENPLAN	attributes, fabrication operation description	arithmetic control S	coded rules	backward chaining
MIDS	none	none	none	none
F & M	two dim matrix scaleers	arithmetic relational financial	equations assignment control	risk anal. goal seeking what if anal. forecasting
TRAIN DISPATCHER	distances intersectin paths	connectivity g		branch & bound
PATTERN RECOG. SYS.	none	none	none	none
EXECUTIVE DATABASE SYSTEM	2 to 9 dim. arrays	arithmetic relational financial mathematical	equations control	goal seeking simultaneous equations
ROLLINS FINANCIAL PLANNING	two dim. matrix	arithmetic mathematical financial	equations	simul. eqn solver goal seeking sensitivity analysis what if anal. Monte Carlo risk anal.

Figure 11. Modeling Characteristics (cont)

forecasting

DSS NAME	DATA OBJECTS	RELATIONS OPERATIONS FUNCTIONS	STATEMENTS	PROGRAMMED CONTROL STRATEGIES
ORKIN MARKET ANALYSIS	2 to 9 dím. matrix	arithmetic relational mathematical financial	equations control	goal seeking simultaneous equations
SANTE FE BUDGET PLANNING	multi dimensional array	arithmetic mathematical financial relational	equations	simul. eqn. solver consolidation what if anal. forecasting
SOUTHERN BELL CORP. PLAN.	two dim. matrix	arithmetic mathematical financial	equations	simult. equat. solver goal seeking sensitivity analysis what if anal. Monte Carlo risk anal. forecasting
MYCIN	attribute object value certainty factors	logical relational arithmetic maping conclusion	production rules	backward chaining
PRESIDENTIAL SCHEDULING	tables weights networks	precedence arithmetic logical relational	assignment control	General Inquirer KWIC K nearest neighbor pattern classification

Figure 11. Modeling Characteristics (cont)



## COST ELEMENTS

TOTAL MKT(UNITS) MARKET SHARE(PTS) SALES VOLUME(UNITS) UNIT PRICE SALES(DOLLARS) RAU MATERIALS PACKAGING MATERIALS LABOR COST DISTRIBUTION COST VARIABLE COMMISSIONS TOTAL VARIABLE COST CONTRIB. MARGIN DEPRECIATION DIVISIONAL FIXED COST CORT. ALLOCATIONS TOTAL FIXED COST NET INCOME(AT)

ITENS

Three Dimensional Matrix Model Figure 12. (Courtesy, Boeing Computer Services)

Maria I.

consisted of a set of equations that might be recursive, simultaneous, linear, non-linear or logical.

A screen-oriented editor, such as used in VisiCalc, allows the model to be built by addressing individual cells displayed on the screen by lettered columns and numbered rows (see Fig. 13). VisiCalc also allows cells to be addressed by positioning the cursor at those locations.

The command line above the spreadsheet in Fig. 13 indicates that the value in column D, row 25 is defined to be .43029 times the value in column D, row 39. Since the total value for that column was defined to be sum of the elements in each row, the other values in the column will be adjusted automatically. The display in Fig. 13 is only a portion (a window) of the entire spreadsheet.

line-oriented editor requires the model to be built A separate from seeing the rows and columns displayed on the screen. In Fig. 14 the specification for an oil rig model built using the DSS/F line-oriented editor is shown. The editor assigns line numbers that are used for specifying changes. Line 4 is followed by the model row number 2. The row numbers are used to specify calculations. For example, line 8 shows a formula whose interpretation is IDLE WEEKS = 52 - LEASE WEEKS - MAINTENANCE WEEKS. There is no formula for LEASE WEEKS and MAINTENANCE WEEKS. The values of these variables are supplied later through a data file. DSS/F has several functions that allow one to express relations involving prior time periods. One of these is LAG used in line 15. DSS/F also has several financial functions. The formula in line 23 uses one of these DEPR, for calculating depreciation.

Other types of data objects that were observed in the DSS included tables, text, lists, maps, attribute-object-value triples, context trees, certainty factors, networks of activities, events and durations, locations and paths.

The relations between and operations on these data objects included: arithmetic, relational and logical operations on the arrays, tables, lists and scalars; precedence and subactivity relations of the networks of activities and events; logical, relational and arithmetic relations on the attribute-object-value triples; and expand or combine regions on the maps. Other types of statements used in the models included: assignment and control statements, productions, equations and networks.

Mathematical functions included absolute value, maximum, minimum, log, exponential, rounding, average, accumulate. Financial functions included depreciation, amortization, present value, cost/benefit ratio, discounted cash flow, internal rate of return.

A number of the DSS supported formulation of a model

						E to a la la la la la la	13 3 1
	A	B	С	D	E	F	G
9	TYPE			MEDICAL	TOTAL	COST/	
10	OF	CPT	RVS	ENCOUN	RVS	RVS	COST/
11	VISIT	CODE	VALUE	DATA	UNITS	UNIT	ENCOUN
12							
23							
24	Brief	90040	15.00	2750	41250	1.16	17.43
25	Limited	90050	20.00	5000	99999	1.16	23.24
26	Intermed	90060	25.00	2000	50001	1.16	29.05
270	Comoreben	900B0	35.00	600	21002	1.16	40.67
28	Sempt anan						
29		1122-102					
130	Minia Sev	00030	5 00	150	750	1 14	5.81
1 31		70030	5.00	130	130		5.01
1 24							
1 34						nores of	
1 33	INPALLENT						
34							
35	Initial	90215	45.00	110	4952	1.16	52.28
36	Follow-up	90250	23.00	400	9200	1.16	26.72
37							
38		1					
39	TOTAL		NA	11620	244845	NA	NA

ebre te finite and the first strands strands and and and and and a set of the set of the

in Lements, Theory Treats and Links

section automatically. The display in

D25 /FI (V) .43029\*D39

10 pl 12 1 .... 28

# Figure 13. VisiCalc's Screen-oriented Editor

	the second second second	takan fotarobh add, afloudde,
is an	atticute of the	and the state of the state of the state of
<	- COMMAND: EDIT	
FIL	E: DIL LOG	and used show the a lot way is an
NEW	IFILE? YES	
INF	UT	Man they bellet in the set of all
1:	C DIL RIG INVESTMENT ANA	ALYSIS
2:	C	
3:	C 1. REVENUES	form stranger indente-and bread
4:	2 'DEADWEIGHT-TONS'	
5:	3 'LEASE (\$/TON/MONTH)'	
6:	4 'LEASE WEEKS'	entrances in the tet as somethe
7:	5 'MAINTENANCE WEEKS'	
8:	6 'IDLE WEEKS'	= 52.0 -4 -5
9:	7 'LEASE REVENUE'	= 2 * (3/30.0) * 4
10:	C another area	
11:	C 2. COSTS	
12:	8 'INFLATION RATE	
13:	10 'BROKERAGE RATE'	
14:	11 'BROKERAGE COST'	= 7 \$ 10
15:	12 'DRILLING COST RATE'	= 12 LAG 1 * 8 FOR 2 TO 5
16:	13 'DRILLING COSTS'	$= 12 \pm 4(0.5 \pm 12 \pm 5)$
17:	14 'REPAIR COST RATE'	= 14 LAG 1 ¥ 8 FOR 2 TO 5
18:	15 'REPAIR COSTS'	= 14 * 5
19:	16 'INSURANCE COSTS'	= 16 LAG 1 * 8 FOR 2 TO 5
20:	17 'ADMIN COSTS'	= 17 LAG 1 * 8 FOR 2 TO 5
21:	20 'COST OF OIL RIG'	
22:	21 'DEPR DATA'	
23:	22 'DEPRECIATION'	= 20 DEPR 21
24:	25 'TOTAL EXPENSES'	= 11 + 13 + 15 + 16 + 17 + 22
25:	如此的人们,最近11月的的原因,不是在一些发展了这个。	

pare frontes strates of france and conservation of the second and and conservation of the second and the states of the second of the second of the

to the volution want by the set of

L'Art le ters de Mandel Carlander Le traite Ma singegen d'is appelyers of clovers the second space was another the second the

second received a lossed apedeater losting

Landnos Sematiporgeng 5 to 101-11/11

Figure 14.

territ of

1 HORIDAN

DSS/F's Line-Oriented Editor

stonants is bloosts is and all blook ACTING THE TRANSPORT FERDING

ters that that y that stated at 1000

via a statistical analysis package such as SAS or SPSS. These packages provide a variety of functions such as time series analysis, multiple regression, and cross tabulation. The typical approach was to use an existing package rather than to develop a new statistical package.

A distinctive aspect of the observed DSS was the inclusion of a preprogrammed control strategy for problem solving. First a model of a problem is formulated. This amounts to knowledge of properties and relationships of objects in some problem domain. Then a problem solving control strategy searches for one or more solutions. This supports the decision maker or management analyst by substantially reducing the amount of effort required to find solutions and makes possible the consideration of alternatives that might not have been explored otherwise.

The preprogrammed control strategies observed included solution of simultaneous equations, "what if" analysis, sensitivity analysis, goal seeking, Monte Carlo simulation, forecasting, branch-and-bound, and backward chaining.

A "what if" capability allows the user to temporarily change any data or assumptions in a model and resolve the Sensitivity analysis allows the user to investigate model. the response of a variable or a set of variables to a series of stepped changes in the values of another model variable. A goal seeking capability allows a goal or performance level to be specified and a variable selected to adjust in order to achieve this goal. Other model variables are automatically adjusted to achieve this goal. Monte Carlo simulation supports risk analysis by allowing the incorporation of uncertainty in the model via randomly generated variable values and collections of statistics on multiple computed solutions. Forecasting involves projection of values of a variable at a future time based on curve fitting or regression techniques. Branch-and-bound is a control strategy for finding an optimal solution to a problem. Backward chaining is a technique for reasoning backward from hypothesis to facts in an attempt to confirm an hypothesis.

Most decision makers did not use the modeling language feature directly. Rather it was used by assistants to the decision makers or by staff analysts.

Two of the systems could be characterized as knowledgebased systems: GENPLAN and MYCIN. Knowledge-based systems contain rules or models that capture relationships, properties, and logical relations among objects. The knowledge is used for such purposes as planning or problem diagnosis. A knowledge-based system can provide management support similar to that of an apprentice, assistant or colleague. These systems "behave" as though they "understand" the user's requests and "reason" about the problem to develop plans or diagnose causes of problems.

## 2.4.4 Data Management Characteristics.

The data management capabilities of the surveyed DSS were examined as to the underlying data model, the types of data management operations available, whether there was a data dictionary or a query language, the types of protection, and data extraction capabilities (see Fig 15).

A data model is a data structure and operations on this data structure. A data model is used for representing data items, records, files, relations among files, and for manipulating these data objects. The data models used in the observed DSS included text, list, record, relational, hierarchical, and network.

By a full range of data base management operations is meant the existence of operations for describing, creating, and deleting records and files, relating records and files, joining and dividing records and files and searching for records.

A data dictionary is used to maintain a list of files and data elements in the data base and includes such information as the name of the data element, synonyms, type of data (e.g. integer, real, alphabetic), range of values and location in the data base. It is an essential element of an easy to use, powerful, query language.

The most common method of preventing unauthorized access to a DSS is via a dedicated computer and the use of user ID's and passwords. When security was important and there was a need for remote access to the computer system, dedicated leased lines were used. The only "horror" story encounted concerned a disgruntled employee who destroyed some files. This could have been prevented by changing the ID's and passwords regularly.

Complications arise when there is a need for sharing data models and DSS system programs among organizational units. Most of the DSS surveyed had data management capabilities which provided control over access capabilities and level of access. Control of access capabilities included the capability to read, update and write records. Access level could be controlled at the file, record, field or range of values levels. The best approach here seemed to be the inclusion in the DSS of a data base management system with extensive protection features. If the data management system provides the capability to automatically edit data on input, this contributes to maintenance of data integrity.

One of the most interesting information protection schemes used was that in EPIC (ADMIN FACTS). An executive or manager could see only information in files of divisions or departments reporting to him. Furthermore, each manager had file areas reserved strictly for private data that no other manager could see. The details of this scheme were

DSS NAME	DATA , MODEL ,	OPERATIONS	QUERY LANG.	DATA EXTRACT.
EPIC ADMIN FACTS	hierarchical or network	full range	yes	special program
EXECUTIVE INFO. SERV.	n dim. matrix network	full range	yes	yes
BRANCH BANK	record	subset	no	special program
MORTGAGE PORTFOLIO	two dim. matrix	subset	по	no
PEEL THE ONION	record	subset	no	yes (option)
FIELD OFFICE FINANCIAL	two dim. matrix	subset	no	yes
EXEC INFO SYS	record	subset	yes	no to the same
PEANUT PUR.	record	subset	no	yes (option)
GENPLAN	record	subset	по	no
MIDS	none	none	no	no
F & M	record	subset	no	only from General Ledger
TRAIN DIS.	hierarchical	no deletion	yes	no
PATTERN RECOG.	record	subset	no	special program
EXEC DATABASE	network	full range	yes	yes
ROLLINS FINAN.	record	subset	no	yes (option)
ORKIN MARKET	network	full range	по	no a chaoda istab
SANTE FE	relational	full range	yes	yes
SO. BELL	record	subset	no	yes (option)
MYCIN	list	full range	yes	no
PRESIDENTIAL SCHEDULING	record & text	subset matching	yes	yes

and a set to dealers to a set of the

Figure 15. DataBase Management Characteristics

not available, as they were considered proprietary. It seems however, to be similar to the scheme first implemented on MULTICS.

If the objective of the DSS included data analysis and modeling, the capability for extracting data from external files substantially increased the utility of the system. The simplest method of data extraction was to provide the capability to read external reports and tabular data representations as ASCII text files.

The most common method of transferring data was to have a special purpose program written by the data processing department that extracted data from computer files and constructed a tape or diskfile that was then either reloaded for use by the users on their DSS or that was physically transferred to another computer on which the DSS operated. The reasons given for this procedure included the maintenance of data security, the lack of computer to computer links, or the lack of a general file extraction routine.

## 2.4.5 Communications

Communication between a user terminal and a DSS on a mainframe computer was supported in a variety of ways including asynchronous or synchronous, direct, private or leased lines, and software drivers for various terminals. Microcomputer interface to mainframes was usually via an asychronous RS-232c protocol, or emulation of a common terminal such as the IBM 3270 or DEC VT-125.

Users of a DSS on a time-sharing network or users of a data service dialed a local number to connect to a multiplexor that was connected to a central computer in another city via a private line. None of these various communication characteristics seemed to impact the quality of the DSS, other than the use of direct or private lines to support data or system security.

Another type of communication was the electronic mail facility available on several DSS operating on the same mainframe. This facility was extended in the BELL EPIC system by modifying a teleterminal (telephone and terminal) that allowed autodialing of other EPIC users and transfer of electronic messages to other teleterminals connected to an EPIC system that might be on another mainframe.

One of the major technical problems in DSS development was the communication among computers of different manufacturers. This will be discussed along with other technical problems in the following section.

## 2.4.6 Technical and Management Problems.

The most significant technical problems concerned communication among computers, data integrity and security, and development methodology.

The sharing of data and programs among various mainframes and microcomputers was also cited as a problem by managers. They expressed a desire to use the output of other systems in the organization but were frustrated by the lack of cooperation by other organizational units in sharing data and the technical difficulty of communication between systems.

This communication was typically needed to acquire data from one mainframe for use by a DSS on another. While there are national standards for interprocessor communications, various protocols continue to be used. The protocols are poorly documented or proprietary which further complicates the development of a communications interface.

A partial soluction to this problem is to use a communications software package such as CROSSTALK [MICROSTUFF(1982)] that is available on numerous micro and main frame computers. Any two computers with CROSSTALK can communicate with each other and files can be transferred between them.

Another partial solution is to build software interfaces for transforming file formats for one DSS component into file formats required by another.

An issue raised by many of the managers and their staff was the difficulty of obtaining accurate and timely data from other parts of the organization. Another problem mentioned was the need to maintain the security of data moved from a mainframe to other mainframes or microcomputers.

A suggestion of several managers was that since corporate information was an important corporate resource, there should be a Director of Information Resources whose responsibilities included those of data base administration and the coordination of computer information policies. This same individual might be responsible for creating "owners" of data by providing access capabilities. The "owners" would then be responsible for data integrity, accuracy and currency. This position/office might also be responsible for seeing that the protection features of the system -hardware, software and communication facilities -- were adequate.

A number of development problems were repeatedly related by the staff responsible for DSS development. Managers did not understand the capabilities of computers well enough to state their needs. Thus, it was difficult for the analyst to formulate those needs as functional requirements. The only approach that seemed to work was to show the manager a prototype or vendored system that he could react to and then modify that system.

Managers on the other hand often complained that they could not communicate with the system analysts because of the analyst's lack of management knowledge. Additionally, management noted that when the data processing department had done work for then, it took too long to get results, cost too much, and the systems developed were too difficult to use.

One of the conclusions that can be drawn from this survey is that specific DSS are being developed by corporate planning staffs or comptroller's offices rather than by the data processing departments. These management support staffs equipped with development tools such as financial planning languages, database management systems, statistical analysis packages and DSS kernels, are supplanting the traditional data processing department functions in building support systems for management.

## 2.4.7 DSS Development Strategies.

What strategies have been used for developing successful decision support systems? If there are several possible strategies, which one is most appropriate for developing a successful prototype decision support system in the STAMMIS environment? The first question will be addressed in this section, the second in Section 3.

First some terminology will be introduced and a perspective will be developed on system development strategies that will aid in classifying the development strategies encountered during the survey.

A system development strategy is a high level plan for solving a system development problem. A system development problem can be represented as a triple: (1) an initial state consisting of a description of user needs, (2) a goal state or criterion which is a computer system and (3) a set of system development operations (techniques or methods). The system development problem is reduced to a sequence of subproblems, such that if each of the subproblems can be solved in sequence, then the entire system development problem is solved. This sequence of subproblems is the system development strategy.

These subproblems must be reduced to simpler subproblems that are eventually primitive problems that are solved. Research sponsored by AIRMICS has pointed to the importance of determining a system life cycle methodology consisting of system development and maintenance methodologies. A number of system development strategies have arisen from experience in developing systems. They differ on the basis of such factors as the structure of the problem, the size of the problem, the resources that can be brought to bear on the problem, and other constraints.

## 2.4.7.1 Alter's Categorization of Development Strategies

Following Alter (1981) four system development strategies outlined in Fig. 16 will now be considered.

The phased development strategy with various refinements is the most widely used and studied of the methodologies. It has been used for developing automated systems to replace manual operational procedures, to develop new systems (ballistic missile defense systems) and to develop Management Information Systems (MIS). According to Alter (and in accordance with what is widely held) the benefits of this strategy are management control of the development effort, maintainability of the developed system, and less risk due to turnover.

The disadvantages are large documentation costs, receiving no benefits until the project is completed, possible incorrect requirements statement due to lack of feedback during a lengthy developmental cycle, and risk of technical problems due to lengthy delay before technical feasibility is proved.

There are two variations of the phased development strategy that address the cost and risks.

- 1A. phased development strategy with prototyping
- 1B. phased development strategy with phased implementation.

The first variety is used to design and verify requirements and thus reduce the risks. The second is used when the system is a large system and either there are not enough resources to implement the entire system and/or the benefits of part of the system are desired before the estimated time for complete implementation.

The <u>turnkey development</u> <u>strategy</u>, as Alter describes it, begins with identifying a generic problem for which software from a vendor already exists. Systems analysts and software engineers would view this as a version of the phased development strategy where one does complete requirements analysis and then checks to see if there was existing software that meets all or part of the requirements or that might be modified to meet the requirements. This strategy was not encountered in the current study, however, exactly as Alter described it. Instead of a requirements phase conducted by a systems analyst, a management area specialist identifies a problem for which a package supplied by a vendor already exists. This "end run" around the data

#### Phased Development Strategy 1.

Complete each step below in sequence with complete documentation and user signoff after each phase.

- Problem Definition and Planning. a.
- Requirements Definition (analysis of existing b. system) to determine what is needed to solve the problem.

THE THE RE ITELS HIT IN. ILEY STRAND

- Design System and modules с. and the action of
- d. Code and test modules
- e. Integration and acceptance testing
- f. Installation (release, fielding)
- Turnkey Development Strategy 2.
  - Identify a generic problem for which a computerized a. exists which has been developed solution elsewhere.
  - Install the turnkey system or a modified version of b. it.
  - c. use it

TTALLING AND AND

- Identify its shortcomings d.
- Change the computer system to fit the organization e. or vice versa.

Incremental Development Strategy. 3.

- Start with existing computer system a.
- Identify its shortcomings b.
- c. Create whatever is needed to improve decision making.
- d. Use the revised system, reviewing periodically to identify its shortcomings.

Evolutionary (Adaptive or Iterative) Development 4. Strategy

Do each of the following steps rapidly in sequence and then repeat as many times as necessary.

TALES BE MILLION IN TOOLS SEAMONDER TO TOOLGUE

- Partial requirements formulation a.
  - Design partial system and modules. b.
  - code and test modules. с.
- d. Use of partial system
- e. Obtain feedback from users as to
- need for modification
  other needs and problems

Figure 16. DSS Development Strategies

and all advances of a sub-states device bar sub-state to the sub-states of sub-states of sub-states of sub-states

processing staff or MIS staff is explained as lack of confidence in those staffs to be responsive to management needs. The data processing department is often viewed as a group of programmers who do not understand management. The management area analyst is often correct.

An advantage of the turnkey development strategy is that it often results in early benefits to management, including experience with DSS. A disadvantage is that the costs are sometimes excessive as there is a pattern of using the software on a timesharing service and avoiding the use of corporate computer resources altogether. Other problems that can arise are the use of a wide variety of DSS products that cannot be fully supported by the inhouse MIS or data processing staff. A variety of vendored DSS tools can make it difficult to extract data from the corporate data base and to share data among users of different systems.

The incremental development strategy can also be viewed as a version of the maintenance phase of the software life cycle in which a phased development strategy was used. The basic idea is to start with the existing computer system (usually an MIS) and identify additional management needs for decision support. The advantage of this approach is that incremental changes may facilitate user learning. The two disadvantages that Alter identifies for this approach. are that too many incremental changes may degrade the system design and documentation, and incremental changes may not solve the problem if new concepts and approaches are needed. Both of these risks would be minimized by a maintenance methodology and by the recognition that a large system that used and adapts to needs may need to be restructured or is redeveloped.

The evolutionary development strategy under various names, including adaptive [Keen (1980)] and iterative [Sprague and Carlson (1982)], is widely espoused by DSS researchers as the strategy for developing DSS. The essential idea is to formulate some of the needs of the user, rapidly build a prototype to meet these needs, have the user use it, modify the prototype to meet the needs and repeat this process till the prototype stabilizes.

The advantages of this approach are that the methodology applies when there are difficulties in writing detailed functional requirements. Also, the users get a better idea of what computer-based support can do for them. Further, the risk of producing a large system that does not support management problem solving is minimized.

A disadvantage of this approach is that users may find it difficult to adapt to rapid change. Also, if the prototype development is not focused on significant managerial problems needing support, the result can be a system that is not very useful or of limited applicability. One might think that because detailed documentation of programs and management review were not required as in the phased development approach, that there was inadequate management control and potential for excessive development costs. This does not seem to be the case. Adaptive Design as reported in the literature and as encountered in this study is not the same as ad hoc system development. The investment in adaptive design seems to be comparable to the investment in the requirements definition phase of phased development with the advantages mentioned above.

## 2.4.7.2 Development Strategies of Surveyed DSS

The types of strategies used for developing the DSS included in the current study are shown in Fig. 17. Of the systems developed five were in our judgement turnkey systems, three were developed using phased development, one DSS was developed using an incremental strategy, two used phased development to create a prototype, and nine involved evolutionary or adaptive development of a prototype. These results would seem to confirm Alter's conclusion (Alter (1979)) that "there is not one path to DSS Nirvana".

Five of the DSS developed adaptively involved the use by corporate or financial planning staffs, of turnkey DSS tools to develop specific DSS for other organizational units (Branch Bank Management, Mortgage Portfolio, Peel the Onion, Field Office Financial Planning and Peanut Purchasing Model). Four of the systems judged to be turnkey systems--Rollins Financial Planning, Orkin Market Analysis, Sante Fe Budget Planning, Southern Bell Corporate Planning--involved the use by a corporate planner, market analyst and financial analysts of a financial planning language or meta-DSS to develop ad hoc models. In the first case the staff was acting as developers. In the second case they were acting as DSS users.

The finding that phased development is used to develop successful DSS contradicts Keen's contention that DSS imply an adaptive design strategy. For this reason some of the cases that Keen states support his conclusion that an adaptive design strategy is required for DSS will be reviewed.

## 2.4.7.3 Review of Five Other DSS Case Studies

Keen (1980) lists 17 DSS case studies that he states were developed using an adaptive design strategy. Six of these case studies appeared in Keen and Scott Morton (1978). Four of these case studies contained information concerning development strategy: Portfolio Management System (PMS), Capacity Information System (CIS), Brandaid, and Generalized Management Information System (GMIS). Another of the 17 case studies, ISSPA, appeared in Keen and Gambino (1983).

PMS is a DSS for investment managers. It is a graphics-oriented system with a variety of models and a

# EPIC

# EIS

Branch Bank Management Mortgage Portfolio Peel the Onion, GPGS Field Office Fin. Planning Peanut Furchasing MIDS

GENPLAN

F&M

Train Dispatcher DSS Pattern Recogn. System Executive DataBase Sys. Rollins Financial Planning Orkin Market Analysis Sante Fe Budget Planning Southern Bell Corp Planning MYCIN

Presidential Scheduling

Phased Development of Prototype Phased Development Adaptive Adaptive Adaptive Adaptive Executive Information System Turnkey, Phased Development Adaptive Adaptive Phased Development Incremental Phased Development of Prototype Adaptive dev. of Prototype Prototype, Turnkey Turnkey Turnkey Turnkey Turnkey Adaptive Dev. of Prototype Adaptive Development

Developed an Unite and

#### Comparison of Development Strategies of Figure 17. Surveyed DSS

anantes set dama in the store set buse test that the

bus is shown to writer a city matere build for an la cut

large database. A prototype was developed by Gerrity (1971) as an academic experiment and initially used by the trust department of Big Western Bank's trust division. An implementation of PMS based on the prototype took 3 years to complete. An implementation of PMS at Great Eastern Bank consisted of a feasibililty study, design and phased implementation. Both systems evolved but it is stated that the evolution was slow.

The Capacity Information System (CIS) is another graphics-oriented DSS used by product planners in a truck manufacturing company. The following are quotes from the Keen and Scott Morton description of the study.

The design team emphasized human engineering issues from the start. They specified the output from the system well before the routines that generated it were discussed. This had several additional ever advantages; it provided a clear criterion for assessing and peripheral equipment and gave top hardware management a fairly precise description of what they were being asked to approve. Users had a clear idea of what the system would look like, well before the development effort got underway. CIS was thus designed mainly in terms of the software interface between the user and the system.

The design of CIS emphasizes ease of use and fast response time; these requirements were major constraints on the specification of analytic models and data bases for the system.

CIS took almost 2 years to implement. The mandate to develop a proposal was given in July 1973. In January 1974, the design team made a formal presentation of the intended outputs of CIS to top management. Over the next year, the system designer and two programmers worked full time on CIS; substantial time was spent on developing criteria for selecting hardware, software, and peripherals. They also wrote 40,000 lines of FORTRAN code. The system was made operational in January 1975 but another 6 months were needed to make modifications and adjustments. These 6 months were highly frustrating to the designers; all the "creative" work had been completed, but many wrinkles remained to be ironed out.

Brandaid is a DSS for marketing managers. It is a marketing model developed using EXPRESS. The discussion stresses the adaptiveness of the model and the model's evolution.

The Generalized Management Information System was developed by Donovan and Madnick (1976) at MIT. GMIS was designed to support ad hoc decisions. These are situations in which data needs and functions cannot be predefined. GMIS was designed to provide an environment in which a variety of modeling components (TROLL, ISP, DYNAMO, APL/EPLAN, APL, FORTRAN, and PL/1) need to share data stored in multiple data bases (SEQUEL). CMIS uses IBM's Virtual Machine operating system (VM) to facilitate the interfacing of the different modeling components. The approach used was to build a component interface module rather than to modify the individual modeling components. The use of component interfaces facilitated flexibility and system evolution.

Keen and Gambino developed the Interactive Support System for Policy Analysts (ISSFA) using the adaptive design strategy. The development began as a research project but had as an eventual objective the implementation of a commercial product. Micro-DSS/Analysis distributed by Addison-Wesley is the resulting product. The DSS was developed by constructing a prototype that implemented some basic features needed by the user. The system was used and modified per user requests. Finally the system was restructured to turn it into a product.

The conclusions that we reach from an analysis of these five case studies are as follows: A prototype version of PMS was developed as an academic experiment, but the implementations in organizations were via phased development. CIS was constructed using phased development. Brandaid is a model constructed using EXPRESS. GMIS is a set of DSS tools interfaced using VM to facilitate DSS evolution. Evolution referred to in the Brandaid case refers to the evolution of models, not of the dialog style, analytical commands or data management components. The adaptive design strategy was used for ISSPA. PMS, ISSPA and GMIS seem to be clear cases of the adaptive design strategy. Since Keen considers Brandaid to be a case of adaptive design, he must be considering model modification to be an aspect of adaptive design.

In summary, all of the four strategies--phased development, incremental, turnkey and evolutionary (adaptive) design--have been used to develop successful DSS. The essential idea of adaptive design--rapid, repetitive prototyping--is useful (1) when there is difficulty formulating DSS requirements that may be due to the semistructured or unstructured nature of the management problems, (2) management doesn't want to see requirements but wants a system that will meet some of their requirements "here and now", (3) management wants a system that is easy for them to use, and (4) financial resources are not available for a large DSS.

## 2.4.8 Summary of Comparative Evaluation

All of the surveyed systems were used in the solution of problems that required interaction between the computer and executives, managers, or management staff. With few exceptions the systems were used to construct, modify and use models. The models often involved incomplete or uncertain knowledge. The models were used for such functions as analysis, planning, forecasting, scheduling and diagnosis. In other words, the systems were used to support management decision-making regarding what have been called semi-structured or unstructured problems.

One of the most powerful features of these systems was a nonprocedural modeling language combined with a preprogrammed control strategy for solving the problem represented by the model. Examples of these automatic problem solving strategies were simultaneous equation solvers, what if analysis, sensitivity analysis, Monte Carlo risk analysis and branch-and-bound.

Another conclusion one can draw from the survey data is that different types of support are needed for different management roles in the organization. Executives and line managers do not usually use the model building, statistical analysis and comprehensive database management features of the system. These functions are used by management staff. Executives and line managers will use the models constructed by staff, use the graphics and representations, compare planned projects and data, and use rudimentary file management features. In other words, managers use these systems to support their functions. Detailed analysis, formulation of models, and data management are not management functions, but the functions of management staff.

The most successful DSS for support of executives and top level management were EPIC, MIDS, and the Executive Data Base System. The quality of these systems can be attributed to beginning with the needs of the executives rather than the needs of financial analysts, corporate planners or other organizational units. The systems are designed to be easy to learn and easy to use. The dialog design includes such features as menu with soft keys or four key access to any report. The systems have a simple design, were developed using prototypes, complexities were put off until later, and the systems were developed by individuals with a management and technical background.

A major factor in the successful development of DSS in an organization was the recognition by top level management of the need for computer-based management support above and beyond that provided by MIS and management science or operations research models. CEOs at Northwest Industries, Lockheed-Georgia and Bell Laboratories specifically requested this type of support.

The most successful DSS for support of operational managers were GENPLAN and the Train Dispatcher DSS. The success of GENPLAN can be attributed to its knowledge-based approach. The success of the Train Dispatcher DSS is due to the prototyping methodology, the inclusion of a modified branch and bound optimization technique, and the excellent user interface. Cost benefits played a major role in the justification of these two DSS.

The most widely used DSS in an organization was boeing's EIS. EIS is a meta-DSS with interfaces to a braod range of DSS tools -- modeling languages, database management systems, graphics packages, statistical packages, project management tools and external databases. It's use is promoted in the organization and its development is supported by a subsidiary, Boeing Computer Services, which markets EIS.

DSS tools such as financial planning languages, statistical analysis packages, data management systems, graphics packages, and data entry and data extraction routines served two DSS functions. First, they were used by management staff to support their roles in the organization -- financial planning, corporate planning, market analysis, forecasting, ad hoc modeling and reporting. This was usually the purpose for which the tools were purchased or developed and the staff interviewed reported that the tools met these objectives and further, were used by the staff to build specific DSS for managers, which is the second DSS function. These tools significantly contributed to the ease of implementation of specific DSS by making it possible for management staff rather than data processing staff to develop specific DSS.

Some of the most successful DSS were those built from meta-DSS, that is, from an integrated set of DSS tools. Because of cost many organizations had not acquired all of DSS options that were available. The increased the availability of lower cost DSS tools, such as DSS/F, DSS/A, VisiCalc, Infoscope and Lotus 1-2-3, on microcomputers, along with the increasing number of vendors who are making subsets of their mainframe DSS tools available on microcom-puters, may substantially alter the picture. The adaptive, (evolutionary or iterative) strategy has distinct advantages over the phased-development strategy for the development of computer-based systems to support management decision making. These advantages include: (1) rapid delivery of a working prototype that provides solutions to management problems, (2) addressing management reticence to use a computer-based system by developing in managers the perception that the DSS environment is friendly and that they can be successful in using DSS and (3) addressing the difficulty of formulating requirements for semi-structured or unstructured management problems.

## 3. A DEVELOPMENT STRATEGY FOR A PROTOTYPE DSS IN THE STAMMIS ENVIRONMENT

## 3.1 The STAMMIS Environment

A STAMMIS (Standard of Army Multicommand Management Information System) is one a large number of computer-based information systems for financial, logistics and personnel The U. Army Computer Systems Command management. S. (USACSC) is responsible for the design, development, test, installation, programming and systems support of the STAM-MIS. The three largest STAMMIS are STANFINS (Standard Financial System), SAILS (Standard Army Integrated Logistics System), and SIDPERS (Standard Installation/Division Personnel System). Other STAMMIS include IFS (Integrated Eacilities System), STARFIARS (Standard Financial Inventory Reporting System), STARCIPS (Civilian Personnel Pay System), SCIPMIS (Standard Civilian Personnel MIS, VTAADS (Vertical The Army Authorization Documents System), MARDIS (Modernized Army Research and Development Information System), DS4 (Direct Support Unit Standard Supply System), and IABS (Installation Automated Budgeting System).

The current hardware environment consists of IBM 360s at the bases and a variety of computer hardware (IBM, Univac, and CDC) at the Department of Army level. Each base has a Data Processing Installation (DPI) and each installation has a Management Information Systems Officer (MISO). The STAMMIS programs are run in batch mode.

VIABLE (Vertical Installation Automatic Baseline) is a Department of Army project to replace the IBM 360 computers in the Base Operating Systems (BASOPS) environment. VIABLE will eventually consist of a network of Amdahl 470s at five Regional Data Centers (RDCs) managed by Electronic Data System software on the regional computers will Systems. include: OS/MVS operating system, COBOL, FORTRAN and ASSEM-BLER programming language processors, SAC (Security Access Control), CICS/VS for telecommunications, Data Reporter for information retrieval and an Applied Data Research DBMS. At each base there will be a Data Processing Center (DPC) consisting of an IBM 4331 networked to one of the five regional data centers. The system software on the IBM 4331 will include: DOS/VSE operating system, COBOL language processor, and ACF/NCP2, CNS2, and EP4 network software.

The STAMMIS are being converted from BASOPS IEM 360s to the VIABLE system. In the VIABLE environment the STAMMIS will be interactive. The interactivity will include at least online data entry and file inquiry.

At the present time it is not planned to make the Department of Army level computers a part of the VIABLE system. DPCs will be able to communicate only with their RDC and not with other DPCs on the VIABLE network. For the purposes of this study, the STAMMIS environment will be considered to be the VIABLE network, the information available in the STAMMIS reports, the individuals at the bases who use this information, and the types of management decisions made at bases using this information. In the next section a strategy for developing a prototype DSS in the STAMMIS environment will be recommended.

There are a number of other issues that arose during interviews with STAMMIS users that are beyond the scope of this study but need to be studied. The IBM mainframe conputers in the current Base Operating Systems environment are already being used for decision support. At one base we visited we were told that there were approximately 100 word processors and microcomputers from 20 different vendors. Some of these are being used for decision support functions. This is "well and good," but who will support and coordinate systems software, the interface to mainframe databases or networking of workstations? Can the security of classified information on these systems be insured? Can accidental deletion of essential information be prevented? Who is responsible for ownership of this information? Who will coordinate the development of DSS in the Army? The Department using it, the Management Information Systems Office (MISC), or the Computer Systems Command? Whose budget should fund the development and operation of the DSS? Due to different chains of command the various groups who are developing and using these systems are not talking to each other. Who will coordinate DSS activities and DSS policies? Keen (1982) offers some suggestions for addressing this last See also the suggestions of some of the managers problem. interviewed during this study which are summarized in Section 2.4.6.

## 3.2 Prototype DSS Development Strategy

One of the objectives of this study was to recommend a methodology for the development of a prototype decision support system in the STAMMIS environment. Major development problems that have been identified from this survey of commercial decision support systems and review of case histories of decision support system development are:

Managers have difficulty in saying what their decision support needs are.

Systems analysts have difficulty in translating these needs into functional requirements.

Managers and their staff have very little time to devote to system definition and review of requirements, design and implementation.

Managers, in general, will not use a system that is not easy to learn and easy to use.

Managment is skeptical of the ability of data processing departments to respond to their needs.

Financial resources are usually not available to develop a decision support system via the traditional software life cycle methodology.

These problems have led a number of DSS researchers and developers to suggest and use a development strategy that has been called the adaptive design strategy [Keen and Gambino (1983)], the evolutionary, or the iterative design strategy. There is not unanimity among DSS developers in recommending this approach to development of decision support systems. Alter (1981) found four strategies used successfully in developing DSS (see section 2.4.7.1). Thierauf (1982) suggests the traditional software life cycle methodology as the appropriate method for DSS development. The twenty DSS surveyed in this study were developed using all four strategies that Alter found plus variations.

It is not clear that the adaptive design strategy and the software life cycle methodology need be disparate. The adaptive design strategy is essentially rapid prototyping [ACM SIGSOFT Symposium (1982)] followed by iterative refinement [Basili and Turner (1975)]. Rapid prototyping is a technique that is used in the software life cycle methodology when there is difficulty defining functional requirements. The difference is that in the adaptive design strategy the functional requirements are not produced after the initial use of the prototype but after continued use of the prototype and iterative enhancement. The iterative enhancement contributes to determining feasibility, user acceptance and better definition of requirements. Nevertheless, the cases examined in section 2.4.7.2 of this report indicate that after prototype stabilization the development process reverts to the traditional software life cycle methodology. Furthermore, contrary to Keen's and others' contention that adaptive design does not produce system requirements, nor is it a feasibility study, the actual case studies belie this contention.

It is recommended that the adaptive design strategy be used for developing a DSS prototype in the STAMMIS environment, because analysis of the DSS surveyed in this study and an examination of previous case studies seems to indicate that adaptive design addresses the problems listed at the beginning of this section better than the software life cycle methodology without rapid prototyping and iterative enhancement.

There are four phases in the recommended approach.

- 1. Study the problem domain.
- 2. Rapidly build/introduce the initial prototype.
- Iterative use and modification of the prototype.

## 4. Development of the prototype as a DSS product.

The second and third phases are the ones most often associated with adaptive design, Phases 1 and 4 appear in interviews with developers and are implicit in the DSS case studies reviewed in section 2.4.7.2. Furthermore, Phase 1 should be the first phase of any software development project. The sequence of processes involved in adaptive design are illustrated in Fig. 18.

The adaptive design strategy will be presented as a set of guidelines (heuristics or rules of thumb) for development of a decision support system that an experienced systems analyst/designer/programmer could use in approaching the development of a DSS for the first time. The guidelines have been formulated from an analysis of interviews with developers of decision support systems and from a review of the practical experiences of DSS developers reported in the literature. The guidelines are particularly influenced by the work of Keen and Gambino (1983). The guidelines will be presented with a discussion and then enumerated at the end of this section.

Rule 1.1: Study and understand the decision processes of the individual decision maker and of his/her organizational context.

The objective of phase 1 is to understand the decisionmaking context. What the analyst needs is some perspective for understanding individual and organizational problem solving. In other words, the analyst needs a model of decision processes which can be used to describe the decision processes of a specific individual or organization.

The construction of this model is the area where the least concrete advice can be given. Data flow diagrams, structured analysis, and data dictionaries are three techniques that are often used to define functional requirements of data processing systems. To the best of our knowledge, this model, which might be called the Data Flow model, has not been used successfully for defining requirements of interactive systems or management decision-making (problem solving) processes. While some DSS researchers [Sprague and Carlson (1982) p. 15] claim that functional requirements of DSS cannot be specified via the traditional requirements techniques, this has yet to be conclusively demonstrated. The best advice that can be given at the present time is that the analyst needs to be acquainted with the various models developed to explain observed decision making behavior in individuals and organizations.

The decision model most often cited in the DSS literature [Scott Morton (1971), Garrity (1971)] is the Intelligence, Design, Choice (IDC) model [Simon (1960, 1977)]. The IDC model identifies three distict phases of problem solving:







Figure 18. Sequence of Processes in Adaptive Design

ten tonyanda atsautura aspio

Intelligence Recognition of a problem.

Design Formulation of the problem and search for alternative solutions.

Choice Decision (selecting an alternative), implementation and management control.

Simon also distinguished programmed decisions from nonprogrammed decisions. A programmed decision is repetitive and routine. A definite procedure has been created for handling them to the extent that they do not have to be worked out each time they occur. A non-programmed decision is novel or unstructured and consequential. It has not arisen before, or its precise nature is elusive and complex. It is so important that it requires custom tailored treatment.

The human problem solving model [Newell and Simon (1972)] seems to provide a useful perspective in understanding management decision making. A problem search graph is a model of the alternatives that an individual explores in trying to solve a problem. Problem search graphs are developed from an analysis of protocols (verbal reports of individuals involved in problem solving). Of particular interest are what Newell (1969) calls ill-structured problems, that is, tasks that little is known about. Newell has found that when people know little about a problem, they use weak search methods such as generate and test, matching, breadth-first and depth-first search and hill climbing. As the problem space is explored, and more is learned about the problem, the problem is often reformulated and strong search methods (explicit procedures) are developed for its solution.

Managers and their staff, often encounter novel, unstructured but consequential problems. They do not know enough about the problem to write down a solution, so they have to search for one and in doing so they acquire additional knowledge that allows them to better structure the problem.

The implications of this observation are that a decision support system should support this type of managerial problem solving. To do so, it must support problem formulation, weak search methods, problem reformulation, and the expression of explicit solution procedures.

The IDC and human problem solving models are both cognitive models of decision making. There are other models of decision making that emphasize the organizational aspects of decision making. These models include: the organizational behavior model [Cyert and March (1963)], the interaction model [Mintzberg (1973)], the communciation model [Farace, Monge, and Russell (1977)], and the garbage can model [March and Olsen (1976)]. A systems analyst should have an acquaintance with these paradigms of individual cognitive problem solving and organizational decision making.

In summary, the systems analyst attempting to understand the decision-making context should collect data on current decision making using techniques such as interviews, observations, protocols, questionnaires, and historical records. He should be seeking answers to questions such as:

What are the specific problems that the manager must deal with?

Which of these problems are structured, which are unstructured?

How does the manager currently go about solving the problem?

How does the manager conceptualize the problem? (tables, graphs, etc.)

The answers to these questions will help the analyst understand the decision-making context from the perspective of the IDC and Problem Search Space models of problem solving.

The systems analyst can then apply one of the following rules to decide whether to use the software life cycle methodology or adaptive design.

Rule 1.2: If the study phase reveals that:

the problem domain consists of structured problems,

functional requirements can be formulated by the systems analyst,

feasibility is easily demonstrated,

funds are available for the development of a product,

then use the traditional software life cycle methodology.

Rule 1.3: If the DSS study phase reveals that:

the problem domain consists of unstructured or semistructured problems,

the systems analyst finds it difficult to translate user needs into functional requirements,

managers are not familiar with what DSS can do for them,

management requires that feasibility be demonstrated, and

the funds available are not sufficient for development of a DSS product,

then use an adaptive design strategy to develop and refine a DSS prototype.

The second phase involves rapidly building an initial prototype decision support system. The qualifications of the DSS prototype developers are critical.

Rule 2.1: Select DSS developers with management training who can relate well to managers and professional staff, and a productive programmer who is experienced in the development of online systems.

The development is accomplished by a two person team, a project manager/analyst and a designer/programmer, working closely with a potential DSS user. The analyst has to be able to understand the decision-making context and to communicate well with a manager and his staff. This implies a Bachelors degree, or an MBA, or equivalent experience. It important that the analyst and designer/programmer be is viewed more as facilitator than as an analyst and a program-Since the prototype is to be built rapidly, the mer. designer/programmer must be more productive than the average data processing programmer. Furthermore, since the design of the man-machine interface is critical, the designer/programmmer should have experience with on-line systems.

Rule 2.2 [Keen and Gambino (1983)]

- Design the Dialog first.
  - (a) Define the data representation.
  - (b) Define DSS commands that correspond to the verbs that the manager uses to describe operations on the data representation.

In discussions with the potential user of the system the designer first focuses on representation of the data that the manger will use in the system and second sketches a user system dialog. The manager should be asked to suggest the verbs that he would use for describing how to manipulate the data representation and the system designer should suggest a dialogue type. Commands should be designed to correspond to user verbs. For a detailed discussion of dialog design techniques for LSS see Sprague and Carlson (1982) [chapter 4].

Rule 2.3: Use prompts, menus, and online help to facilitate user learning.

These features are exemplified by Infoscope, a dialog and data management tool developed by Userview (1982). Command options are shown in logical groups at the bottom of the screen (see Fig. 19(a)). One can get help on a topic and see it in the context of the situation when help is selected (see Fig 19 (b)).

Infoscope also provides a virtual screen capability. VisiCalc has a virtual spreadsheet capability. Infoscope allows multiple displays in the virtual screen memory. The box in the lower right of the screen indicates the portion of the virtual screen that is being displayed.

Rule 2.4: Provide the user the option of using the DSS at various levels of expertise.

Allow the user to choose as an option a menu, a command or an abbreviated command style of dialog. This feature is also exemplified in Infoscope. In Fig. 19(c) is shown the prompt to select a menu or type a command name.

Rule 2.5: [Keen and Gambino (1983)] Select a subset of the commands for implementation in the initial prototype.

The principle of phased implementation often used in the software life cycle methodology is also used in adaptive design. For example, even though fifty command types might be identified by the designer, only a subset of these should be implemented in the initial prototype. This is because some commands will be of higher priority to the user, some will be more easily implemented than others and it may be difficult to design the dialog for some commands. It is important to get a prototype operating to which the user can then react.

Rule 2.6 To achieve a system design that is easily modified, use principles of module cohesion and module coupling to achieve module independence.

The system designer now defines a system architecture. Since the next phase will involve rapid iterative modification of the system, the system must be easily modified. This implies that the modules should be relatively independent. More specifically the modules should be loosely coupled (the modules interconnections should be primarily via input/output parameters) and the modules should be highly cohesive (each should perform a single function) (see Yourdon and Constantine (1979)).

## Rule 2.7 Keep the data management capabilities simple

Most of the surveyed DSS did not require the full capabilities of a database management system. When a generalized DBMS was available it often wasn't used due to its complexity. If a database management system is used as

-	Dateboo	Datebook-1 (21 from file DATEBOOK)		lisc. food
	Focus	DATE after 4 May	83	
3 A		TIME TYPE NO	TE	
Pers-14(94	from file PE	RS)	A CONTRACT	
Pocus:	DEPARTMENT	'production" or "shipp	ing''	
FIRSTNAME	LASTHAME	STATE DEPARTMENT	TITLE	SALA
Renee	DiMeara	GA Production	Machinist	16,2
Maggie	Horton	lA Shipping	Supervisor	14,3
Anna	Brady	WV Shipping	Supervisor	13,0
Renec Maggie Julie Anna	DiMeara Horton Williams Brady	GA Production IA Shipping FL Shipping WV Shipping	Machinist Supervisor Clerk Supervisor	16 14 13 14

	395 1 Apr	83 65	5.52 Food	
If you see a word corresponding to	910 12 Hay	Da	tebook-1 (	21 F
bar until the arrow points to that word and press the Enter key. The		F	ocus:	DATE
backspace key [<-] will let you back up if you need to. You can also just	Pers-1 (9	4 from f	ile PERS)	TI
you prefer.	Focus:	DEPAR	INENT "pro	duct
If you don't see the action you	FIRSTNAME	LASTNA	ME S	TATE
want, press the slash key [/] until your choice appears, or press [Esc] to stop using the Selection feature.	Renee Maggie Julie Anna	DiMear Horton Willian Brady	a G I ms F	A Pr A Si L Si U Si
Change which records a scope disp SELECT: Enter Focus Look Make Sort ? 1	lays Help		la	

d

Figure 17. Sample Infoscope Screens (Courtesy The Microstuf Co.)

b

	becks-1 (32 from file CHECKS)					
Focus; everything						
# DA	TE AMT	TYPE	TO-WHOM	DESCR		
395 1 Apr	83 65.52	Food	Food Wagon	Misc. foodst		
396 2 Apr	83 898.88	Mortgage	Riveria Savings Co.	April paymen		
397 3 Apr	83 56.20 83 43.85	Utilities	Riveria Power Co.	March electr		
397 3 Apr 398 3 Apr	83 52.20 83 43.85	Utilities	Riveria Power Co.	March gas March elect		

the state of the second st

. Fred here and

Des Sent Sinders Sinde

aved 1 PV 1A 184

N. S. Statis

Set Tortepol a for another for

the state of the state of the state

gy and is which the out

and the state of the state of the

and the lights

the bear of the

and the second s

Figure 19. Sample Infoscope Screen (cont) (Courtesy The Microstuf Co.)

a DSS tool it probably should be one based on the record, matrix or relational model due to their relative simplicity. The database capabilities most often needed were capabilities to manually create, modify and delete data files, to obtain external tabular data, associate data with variables in models, to construct a new file from data in several other files, to sort data, to associate data files with models, and to save the results of models in data files.

Rule 2.8 [Keen and Gambino (1983)] Use off-the-shelf library routines whenever possible.

If one can use existing libraries of statistical function and graphics routines and an existing financial planning language, this will significantly speed up the prototyping. Modifications to these routines will consist primarily of developing an interactive interface to the user and interfacing the various existing components.

Rule 2.9 A DSS software tools environment is also needed for rapidly building the prototype and subsequent iterative modification.

Rapid construction of a prototype is also supported by software development tools such as a relational data base system with a high level query language, a dialog management program, communications software, and an interpretive programming language such as BASIC, LISP or APL.

Prototyping can be facilitated by interfacing previously developed components via component interfaces rather than modifying the individual components. For example, develop a module that interfaces modeling and database components; develop a module that interfaces dialog and database, etc. Donovan and Jacoby (1977) discuss the use of an operating system with virtual memory management as a tool for building component interfaces. See Sprague and Carlson (1982, Chapter 10) for a discussion of interfacing dialog, database and modeling components of a DSS.

Rule 2.10 Deliver the initial prototype rapidly and inexpensively.

To demonstrate feasibility, to show the potential users what can be done, to solicit their reactions and better understand their needs, the prototype should be developed in a short time frame. The time frame for phase 1, development of the initial prototype should be on the order of 4 to 8 weeks.

Phase three of adaptive design consists of user training, use of the system, user reaction to the system, programmer support of the user, iterative extensions of the prototype and modifications to the user's manual. The user is learning how the computer can support his decisions, and
the programmer is learning what is required to support the manager's decisions. Belady and Lehman (1979) have observed that any computer-based system that is used evolves and they have formulated several laws of software evolution. The iterative enhancement phase of adaptive design seems to be a way of speeding up this evolution.

Rule 3.1 [Keen and Gambino (1983)] Select a user who knows the task domain well and is highly motivated to use computer technology to support management.

Because adaptive design is so dependent on user involvement and the quality of user reactions, the selection of the user takes on added importance.

Rule 3.2 Develop in the user a sense that the DSS environment is friendly and that the user can do things successfully.

One of the major obstacles to managerial usage of DSS is that DSS have been difficult to learn to use. The prototype DSS has been designed in phase two to be easy to learn and easy to use. The data representation and operations will be familiar to the user. Managers who are using a terminal for the first time may perceive the environment as hostile in which case a period is needed for familiarization.

Psychologists have identified three stages in learning that are relevant to learning to use a DSS. First, the is standing back, passing judgements and learner not They perceive actively involved in learning. the environment as hostile, are intimidated by it, and feel it demanding of them something they don't have. The is individuals stay in this state until such time as they perceive the material as familiar, at which time they enter the second state. In this state they see the problem as possible, but they're unable to do it. They experience a feeling of inadequacy and high frustration until they begin to develop a sense of emerging competency. In the third stage themselves to have been successful. feel The they is friendly, they can do things environment in the environment and they perceive that they have learned something.

Rule 3.3 [Keen and Gambino (1983)] Support first, extend later.

A primary objective of this phase is to adapt the prototype to the user's reactions as he uses the system. Thus, development of additional features that were included in the initial design of the prototype, but not yet implemented, should be secondary to support of existing functions.

Rule 3.4 Construct an automatic monitor of DSS use.

Additional data on the use of the system should be collected via a decision support system monitor. For example, the monitor might log the commands that the user issues in interacting with the DSS, system prompt, system responses, and error messages. This data can be analyzed to determine areas where improvements might be made in the user interface.

Rule 3.5 A hierarchically structured operating system and a database management system with extensive protection features will contribute to data and software integrity and security.

If there are multiple DSS users sharing data and/or software, use a hierarchically structured operating system such as provided by VM/370 to enhance software integrity and security. [See Donovan and Madnick (1975, 1976)]. The essential idea is to provide multiple levels of protection with different locking mechanisms and combinations.

The database management system should provide control over types of access (read, update, write) and level of access (file, record, field, and range of values). The capability to automatically edit data on input will also contribute to maintenance of data integrity.

Strictly speaking, this concludes the recommended strategy for developing a prototype decision support system in the STAMMIS environment. The next phase would be the conversion of the prototype DSS to a DSS product. It is inevitable that due to the rapid design, construction and modification of the prototype, there will be inefficiencies in the prototype. What one has produced is a program that supports some managers in solving some problems. A number of features of a robust system will not have been handled, namely--error handling, documentation and testing of lin-kages. It is likely that the system will need to be The development would revert to the traditional redesigned. software life cycle methodology. Furthermore, as keen and Gambino (1983) observe, and in conformance with the Conservation Law of Belady and Lehman, while adaptive design speeds up the development of a prototype, conversion of this system product requires substantial prototype to a additional time.

In conclusion, we are not recommending the only possible strategy for developing a DSS prototype, but merely a strategy that has been demonstrated to work in producing various successful DSS prototypes. It addresses some of the problems encountered by major a systems analyst/designer/programmer who is asked to develop a system to support managerial problem solving, that do not seem to by the traditional software life cycle be addressed methodology. The development of a prototype DSS in the STAMMIS environment should be viewed as a research and development project. The project should consist of a series

of experiments to refine a prototype (or model). The result will be a better understanding of the requirements of a DSS in the STAMMIS environment. The suggested guidelines for development of a prototype DSS in the STAMMIS environment are summarized in Fig. 20.

#### 3.3 Additional Recommendations

1. AIRMICS should develop a DSS for research management.

Management of research is a domain that has unstructured and semistructured problems. There is a need for support in formulating these problems, in generating alternative approaches to their solutions, in implementing these approaches through allocation of research funds, in monitoring and evaluating the results of research, and in reformulating the research problems.

A conversation with an AIRMICS research manager indicates the potential advantages and feasibility of this recommendation. He indicated that decision support programs were developed for him by a member of his staff to support project planning and project monitoring. These programs were developed using a development strategy that could be best characterized as the adaptive design strategy. Futhermore, the programs were judged by the manager as very useful in supporting his research management decisions.

2. AIRMICS should acquire additional DSS devices and software tools.

This report presents a DSS technological baseline. Since, however, AIRMICS is responsible for DSS research and development in the Army, AIRMICS staff needs to have more detailed knowledge of the functional capabilities of the various hardware devices and software systems that have been used or could be used to develop decision support systems. Computer terminal features that need to be investigated include: touch screens, light pens, soft keys, cursor movement by mouse or touch pad, color graphics terminals, and bit map graphics.

Office automation technology that models the office desk, such as the XEROX 8010 Star Professional Workstation [Smith et al. (1982)] or the Apple LISA [Williams (1983), Wagner and Holt (1983)], could be investigated as a means for providing managers with DSS representations similar to those that they currently use. The Star System is a "user friendly" "electronic desktop". lt uses electronic counterparts of objects in the office, e.g. paper, folders, file cabinets, in-basket, out-basket, mail boxes and cal-A desktop as it appears on the Star screen is culators. shown in Fig. 21. An open window displaying a document is shown at the left of the screen. Commonly used icons (letRule 1.1 Choose some perspective (model(s)) from which to understand decision processes in an organization.

Rule 1.2 If the study phase reveals that:

the problem domain consists of structured problems,

functional requirements can be formulated by the systems analyst,

feasibility is easily demonstrated, and

funds are available for the development of a product;

then use the traditional software life cycle methodology.

Rule 1.3 If the DSS study phase reveals that:

the problem domain consists of unstructured or semi-structured problems, and

the systems analyst finds it difficult to translate user needs into functional requirements,

managers are not familiar with what DSS can do for them,

management requires that feasibility be demonstrated, and

funds available are not sufficient for development of a DSS product,

then use an adaptive design strategy to develop and refine a DSS prototype.

Rule 2.1 Select DSS developers with management training who can relate well to managers and professional staff, and a productive programmer who is experienced in the development of online systems.

Rule 2.2 Design the dialog first.
(a) Define the data representation.
(b) Define DSS commands that correspond to the verbs that the manager uses to describe operations on the data representation.

Rule 2.3 Use prompts, menus, and online help to facilitate user learning.

Figure 20. Guidelines for Prototype DSS Development.

Rule 2.4 Provide the user the option of using the DSS at various levels of expertise.

- Rule 2.5 Select a subset of the commands for implementation in the initial prototype.
- Rule 2.6 To achieve a system design that is easily modified, use principles of module cohesion and module coupling to achieve module independence.
- Rule 2.7 Keep the data management capabilities simple.
- Rule 2.8 Use off-the-shelf library routines whenever possible.
- Rule 2.9 A DSS software tools environment is also needed for rapidly building the prototype and subsequent iterative modification.
- Rule 2.10 Deliver the initial prototype rapidly and inexpensively.
- Rule 3.1 Select a user who knows the task domain well and is highly motivated to use computer technology to support management.
- Rule 3.2 Develop in the user a perception that the DSS environment is friendly and that he can do things successfully in that environment.
- Rule 3.3 Support first, extend later.
- Rule 3.4 Construct an automatic monitor of DSS use.
- Rule 3.5 A hierarchically structured operating system and a database management system with extensive protection features will contribute to software and data integrity and security.

Figure 20. Guidelines for Prototype DSS Development. (cont.)



## Figure 21. Xerox Star Display

ters, memos and blank paper) appear at the upper right of the screen.

The teleterminal that is used as an executive work station in the Bell Labs EPIC system might be examined as a prototype Army "executive work station".

Various single and multiuser microcomputers could be investigated as DSS workstations. Networking of microcomputers via Ethernet could be investigated as an environment distributed decision for support. Microcomputers of particular interest are those based on the 16-bit INTEL 8086 and 32-bit Motorola 68000 microprocessors. Multiuser operating systems on these microcomputers such as XENIX, UNIX III, PICK and OASIS should also be investigated as environments for building DSS. The PICK operating system includes an online query language based on a data dictionary that can be used for ad hoc query capability. Industry prognosticators are predicting that the UNIX operating system will dominate the multiuser microcomputer industry during this decade.

Commercially available financial planning systems and meta-DSS such as EIS, EXPRESS, EMPIRE, IFPS (plus options) and System-W that operate on mainframes should be examined in detail. Decision support tools for microcomputers such as DSS/F, DSS/A, DSS/Dynamo, CONTEXT/MBA [Context Management Systems (1982)], and Micro/W should be examined for their utility as DSS tools for Army applications. Relational data base systems such as ORACLE, INFORMIX, IDOL, Database Plus, Ingress and Distributed Ingress, should be examined for their flexibility in managing DSS user files and providing guery capabilities.

Emerging technologies such as natural language interfaces to data base management systems as exemplified by such systems as INTELLECT should be examined [Artificial Inteluser Corp (1982)]. INTELLECT allows the ligence to retrieve, sort, link, and compare data in a database using English questions or commands of the sort, "Show me the name, address and age of all administrators" and "Has anyone sold over \$2,000,000 this year". INTELLECT detects and ambiguities and misspellings corrects and asks for clarifications. Its primary advantages over formal database query languages seems to be the ease with which it can be learned and used by the occasional user.

3. AIRMICS should develop a DSS that supports the collection and analysis of Army management decision data.

One of the weakest areas in DSS research and development is collection of data on decisions and evaluation of this data. To evaluate Army DSS and to enhance Army decision making the DSS might log (1) the types of problems encountered, (2) the decisions that have been made and (3) the results of implementing these decisions.

approach to recording management problems and An decisions that has been developed by Dr. Larry Weed, a physician, might be investigated. A number of years ago Dr. weed shocked his colleagues by stating at professional meetings that "We don't know what we are doing." What he meant was that physicians did not have adequate records in a form that would allow them to evaluate their past decisions in order to improve on them in the future. To address this situation, Dr. Weed developed the Problem Oriented Medical Record which has since been adopted nationally by many health organizations and is taught in several medical schools as a framework for recording medical observations and decisions. He also constructed an interactive system the Problem Oriented Medical Information System called (PROMIS) which is used by both the patient and the physician to record patient symptoms, problems, therapies and followups [Schultz (1980)].

4. AIRMICS should investigate decision support for software project management using expert systems technology.

Artificial Intelligence research has produced some technologies that could be investigated as a framework for decision support systems. The technology of particular interest is that called knowledge-based systems or expert systems. This technology involves the representation and organization of knowledge in particular domains. The knowledge is used for automatic reasoning and problem solving. While these technologies could be developed for a DSS in the STAMMIS environment, it is recommended that they first be applied to decision suport for software systems development, an area of long-standing interest to AIRMICS and the Army.

Much of our knowledge of system development involves heuristics or rules of thumb. A key to success in the development of software systems is the experience of the expert analyst. Much of this knowledge is not written down in the standard software methodology that a software development oranization establishes, nor does it appear in textbooks. When it is documented, it is often not wellorganized, nor is it in a form that could be used to test that knowledge against new cases to determine its adequacy.

Underwood (1981a) began an investigation using EMYCIN to organize the knowledge that an expert software project manager, systems analyst, or systems designer used in planning the development of the system, diagnosing project problems from symptoms, and generating alternative approaches to the solutions of those problems. The set of rules and guidelines in Section 3.2 of this report is indicative of some of the types of rules that would be incorporated in such an expert system for system development.

A set of rules for systems development is an example of

# REFERENCES

- Artificial Intelligence Corp. (1982), INTELLECT Query System: The LEX Utility, Waltham, Mass.
- Association for Computing Machinery (1982), ACM SIGSOFT Second Software Engineering Symposium: Workshop on Rapid Prototyping. Columbia Maryland.
- Addison-Wesley (1981a), User Reference Manual for Micro-DSS/Finance. Reading, Mass.
- Addison-Wesley (1981b), User Reference Manual for Micro-DSS/Analysis, Reading Mass.
- Alter, Steven (1979), Decision Support Systems: Current Practice and Future Challenges, Addison-Wesley, Reading, Mass.
- Applied Data Research (1982), The EMPIRE User Reference Manual, P12A-UR-10, Application Products Division, Princeton, N.J.
- Basili, V. and A.J. Turner (1975) "Iterative Enhancement: A Practical Technique for Software Development", IEEE Trans. of Soft. Eng., Vol. SE-1, No. 4, December.
- Belady, L. A. and M. M. Lehman (1977), "The Characteristics of Large Systems", RC 6785 (#28969), IBM.
- Bennett, J. L. (1983), Building Decision Support Systems, Addison-Wesley, Reading, Mass.
- Bonczek, R.H., C.W. Holsapple and A.B. Whinston (1981), Foundations of Decision Support Systems, Academic Press, New York.
- Boeing Computer Services (1982), EIS Users Guide, Doc. No. 10221-004, Seattle, Wash.
- Boeing Computer Services (1983), EIS Language Reference Manual, 10221-008-R2, Seattle, Wash.
- Callahan, Leslie G., Jr. (1979), "Brief Survey of Operational Decision Support Systems", AIRMICS, Georgia Institute of Technology, Atlanta, Georgia.

Carlson, Eric D. (ed.) (1977), Database, Vol. 8, No. 3.

- Clancey, W.J. (1979) "Transfer of rule-based expertise through a tutorial dialogue," Rep. No. STAN-CS-769, Computer Science Department, Stanford University (Doctoral Dissertation).
- Clancey, W.J. and R. Letsinger (1981), "NEOMYCIN: Reconfiguring an expert system for application to teaching,"

IJCAI-7 pp. 829-836.

- Comshare, Inc. (1982), System-W Model Specification, Ann Arbor, Michigan.
- Context Management Systems (1982), The Context MBA, Torrance, California.
- Courbon, J.C., J. Grajew, and J. Tolovi (1980), "Design and Implementation of Interactive Decision Support Systems: An Evolutive Approach," Unpublished Working Paper.
- Cyert, K. M. and J. G. March (1963), A Behavioral Theory of the Firm, Prentice-Hall, Englewood Cliffs, N.J.

Data Match, (1982) Smart Buffer, Atlanta, Georgia.

- Donovan, J. J. and H. P. Jacoby (1977), "Virtual Machine Communication for the Implementation of Decision Support Systems," IEEE Transactions on Software Engineering, Vol. SE-3, No. 5, Sept. 1977, pp. 333-342.
- Donovan, J.J, and S.E. Madnick (1975) "Hierarchical Approaches to Computer System Integrity," IBM Systems Journal, Vol. 14, No. 2, pp 188-202.
- Donovan, J.J. and S.E. Madnick (1976), "Virtual Machine Advantages in Security, Integrity, and Decision Support Systems," IBM Systems Journal, Vol. 15 No. 3, pp 270-278.

Execucon (1981), IFPS SENTRY User's Manual, Austin, Texas.

- Execucom (1982a), IFPS DATASPAN User's Manual, kelease 2.0, Austin, Texas.
- Execucom (1982b), IFPS/GRAPHICS Preliminary User's Manual, Release 1.0, Austin, Texas.
- Execucom (1983a), IFPS User's Manual, Release 9.0, Austin Texas.
- Execucom (1983b), IFPS/OPTIMUM User's Manual, Release 3.0, Austin Texas.
- Farace, R. V., Monge, P. R., and H. M. Russell (1977) Communicating and Organizing, Addison-Wesley, Reading, Mass.
- Gerrity, T.P., Jr. (1971) "Design of Man-Machine Decision System: An Application to Portfolio Management," Sloan Management Review, 12(2), pp. 59-75.

IBM (1979) ADRS User's Guide, SH20-2165, White Plains, N.Y.

- IBM (1980), APL Data Interface Program Description/Operators Manual SH20/1954, White Plains, N.Y.
- Information Builders Inc. (1982), FOCUS User's Manual, New York, N.Y.
- Keen, Peter G. W. (1980), "Adaptive Design for Decision Support Systems," Database, Vol. 12, No. 1&2, Fall, pp. 15-25.
- Keen, Peter G. W. and Michael S. Scott Morton (1978), Decision Support Systems: An Organizational Perspective, Addison-Wesley.
- Keen, P.G.W. (1982), "A Policy Statement for Managing Micro Computers" Micro-Mainframe, Cambridge, Mass.
- Keen, P.G.W. and T.J. Gambino (1983), "Building a Decision Support System: The Mythical Man-Month Revisited," in Building Decision Support Systems, J.L. Bennett (ed.), Addison-Wesley, Reading, Mass.
- Management Decision Systems, Inc. (1981), EXPRESS Introductory Manual, Waltham, Mass.
- Management Science America, Inc., (1982), Forecasting and Modeling System: System Documentation, Release 4, Atlanta, Georgia.
- Management Science America, Inc., (1982) Forecasting and Modeling System: Application Guide, Release 4, Atlanta, Georgia.
- March J.G. and J.P. Olsen (1976), Ambiguity and Choice in Organizations, Bergen: Universitetsforlaget.
- Mason, R.E.A. and T. T. Carey (1983), "Prototyping of Interactive Information Systems," CACM, May.
- Microstuf (1982), Crosstalk: Smart Terminal and File Transfer Program User's Manual, Microstuf, Inc. Atlanta, Georgia.
- Mingledorff, Julia M. (1981), "DSS Adds Life to Coca-Cola," Transactions of the First International Conference on Decision Support Systems, Execucom Systems Corp, Austin, Texas.
- Mintzberg, H. (1973), The Nature of Managerial Work, Harper and Row, New York.
- Newell, A. (1969) "Heuristic Programming: Ill-structured Problems," Progress in Operations Research, Vol. III, Julius Aronofsky (ed.), John Wiley, N.Y., pp. 362-414.

Newell, A. and H. A. Simon (1971), Human Problem Solving,

Prentice-Hall, Englewood Cliffs, N.J.

Real Decisions (1982), Financial Modeling Decisions, Stanford, Conn.

SAS Institute (1982), SAS Users Guide, Cary, N. C.

- Sauder, R. L. and W. M. Westerman (1983), "Computer Aided Train Dispatching: Decision Support Thru Optimization," Norfolk Southern Corporation, Atlanta, Georgia.
- Schultz, J.k. (1980), "PROMIS: Problem-Oriented Medical Information System," What can be Automated? E.W. Arden (ed.), MIT Press, Cambridge, pp. 880-915.
- Scott Morton, M.S. (1971), Management Decision Systems, Graduate School of Business Administration, Harvard University, Boston, Mass.
- Shortliffe, E.H. (1976), Computer-based Medical Consultations: MYCIN. New York: American Elsevier.
- Simon, H.A. (1960), The New Science of Management Decision. Harper and Bros., New York
- Simon, H.A. (1977) The New Science of Management Decision. 2nd Ed., Prentice Hall, Englewood Cliffs, N.J.
- Smith, D.C., C. Irby, R. Kimball and B. Verplank, (1982), "Designing the Star User Interface," Byte, April.
- Sprague, Kalph and Eric C. Carlson (1982), Building Effective Decision Support Systems, Prentice-Hall, Englewood Cliffs, N. J.
- SPSS, Inc. (1982), SPSS User's Guide, Northfield, Minn.
- Thierauf, Robert J. (1982), Decision Support Systems for Effective Planning and Control, Prentice-Hall, Englewood Cliffs, N.J.
- van Melle, W. (1980), "A domain independent system that aids in constructing consultation programs," Rep. No. STAN-CS-80-820, Computer Science Department. Stanford University (Doctoral Dissertation).
- Underwood, W. E. and J. P. Summerville (1981a), "PROJ-CON: A Prototype Project Management Consultant," Proceedings of the International Conference of Cybernetics and Society.
- Underwood, W. E. (1981b), "Preliminary Specification of an Organization Description Language" AIRMICS, Georgia Institute of Technology, Atlanta, Georgia.

United Computing Systems, Inc. (1981), FORESIGHT Reference Manual, Version 6.2, Kansas City, Missouri.

and J. Garbers (1982), Infoscope, Userview Corp. Microstuf, Inc., Atlanta, Georgia.

Visicorp (1980), VisiCalc Guide, San Jose, California.

- Wagner, R. and J. Holt, (1983), "LISA-Close Up and Personal," Softtalk, Vol.4, September.
- Waterbury (1981), "Computer-Aided Process Planning," Assembly Engineering Magazine.

Williams, G. (1983) "The LISA Computer System," Byte, Vol. 8, No. 2.

Yourdon, E. and L. L. Constantine (1979), Structured Design, Prentice-Hall, Englewood Cliffs, N.J.

saids it is a start of the start of a start of the start and the start and

Test market of seal activation index to found test autores of charge series average in the series

the internet officered anob.

Viene 2.1 affino: Mavig Bi gginilation

real desirage and a state a state of a second compared to a first leave

Sur Thy standard dependence of the standard of the

the second set of the state of the state will be the

1.1.2432.4

Literation of alaylane, Landsalande ogen 1 691 of the second of the ideals in superior shall be

development of acquisition of the s

company of the second addition of the post

Sho adarang halamada anounted that Jade

bat operation average la participation and a second

table to the second second second

When the state of the state of the state of the

1128368

System Chi Lacior (State)

Storre upiet. part

MARINE COM

rolts plate in the sea blubs

## APPENDIX A

## INTERVIEW CHECKLIST

Organization Name Location Type of Organization Interviewee(s) Names Title Department DSS Role (manager, user, developer) Name of DSS What specific tasks does the DSS support? How often does the interviewee use it? many other users of this DSS are there in the How organization? what were the original objectives of the DSS? To what degree has the system met the objectives? Are there any recommendations that the manager or other of the DSS would make for modification and/or users expansion of the DSS? was cost/benefit analysis a part of the justification of the development or acquisition of the DSS? What are the benefits of the DSS? Can you describe a critical instance indicating the utility of the DSS? What type of training is given to the DSS user? Can we see a demonstration of the system? Can we have some sample ouputs of the DSS?

#### System Characteristics

what computer does the DSS operate on? what operating sytem is used? what types of terminals are used? what type of graphics devices does the system use? what types of graphics representations can be produced? What types of communication links are there between user terminals and the computer? Are there intercomputer links? Describe the user interface? What are the reporting capabilities of the system? what are the characteristics of the modeling or financial planning language? If there are models, what are some of the major variables in the models? Does the sytem have statistical analysis capabilities? what type of data management capabilities does the sytem have? What type of data does the interviewee use or have access to? What type of data model does it use? Does it have a full range of DBMS operations? what type of security features does the DSS have?

### Development Methodology

When was the development started? (or when was it purchased?) when was it first installed? How was the DSS developed? Lid you develop a new system or extend a transaction-based system or MIS? Was the DSS (or DSS tools) purchased from a vendor? Why did you use this development approach? Was there detailed documentation and review after each phase? Was there a feasibility study? was there a complete or partial formulation of requirments? Was a requirements document prepared? was a prototype developed? If so, how many prototype versions were there? Was there a system design document? were program specifications written? was there a system acceptance test? What development tools did you use? How is the DSS maintained? How many versions have been released? Are there any significant technical problems? Can we see the the system documentation or will you discuss the system architecture with us? will they give, loan or let us purchase user and/or reference manuals for the system? What are the names of the manuals?