DEVELOPMENT OF DECISION SUPPORT SYSTEMS (DSS) SOME PRACTICAL CONSIDERATIONS

By

V.B. Kaujalgi

October 1996

Please address all correspondence to:

Prof. V.B. Kaujalgi
Professor
Indian Institute of Management
Bannerghatta Road
Bangalore 560 076
India

Fax: (080) 6644050

DEVELOPMENT OF DECISION SUPPORT SYSTEMS (DSS)

- SOME PRACTICAL CONSIDERATIONS

---

1. Introduction
2. DSS Structure and I-P-O structure
3. Menu Structure v/s Structure Chart of Modules
4. Practical DSS Features
5. Conclusion

References

PREFACE

Use of computers have been traditionally in Management Information Systems (MIS) and Data Processing (DP). Recent advancements in networking, different needs of the users and improved techniques of software development have led to usage of computers in different ways. The most important new applications of computers are in the field of Decision Support Systems (DSS) and Office Automation Systems (OAS).

Advancements in micro computers which can be effectively networked have helped to a great extent, popularity of DSS and OAS.

In this paper, attempts are made to describe some of the important characteristics of DSS software. Based on the author's experience in developing a few real life DSS, distribution of the code in various modules of DSS have been described. Such an analysis should help the DSS developer in planning proper strategy for software development for DSS.

I would like to thank faculty colleagues of mine and programmers who were associated with various DSS projects as part of a research project at this Institute funded by DOE and UNDP from 1984 to 1989.

Bangalore
10 October 1996

Dr. V.B. KAUJALGI
IIM, Bangalore.

DEVELOPMENT OF
DECISION SUPPORT SYSTEMS (DSS)
- SOME PRACTICAL GUIDELINES FOR DEVELOPMENT

1. Introduction :

Use of computers has been changing during the last four decades. The advancements in hardware and software have influenced to a great extent new use of the computers. This is particularly true for the use of computers in business applications for relevant information systems.

Presently the computers can be used for :

* MIS
* Office Automation
* DSS, ES, EIS, etc.

Hence, the traditional application of computers for MIS at different levels like operational, tactical and strategic levels are shown in Fig.1. It has been modified to a three-dimensional structure with the above three distinct applications as the third dimension. This structure is shown in Fig 2.

The basic concept of DSS is dependent upon the solution of semi-structured problems as shown in Fig 3.

Decision Support Systems (DSS) have been defined by Keen and Mortan [1] as :

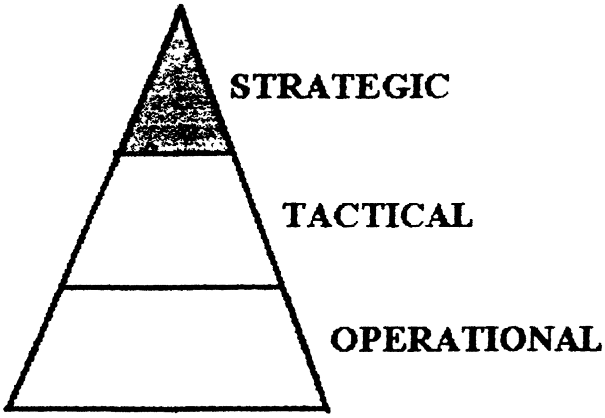

FIG 1: 3- LEVELS of MIS

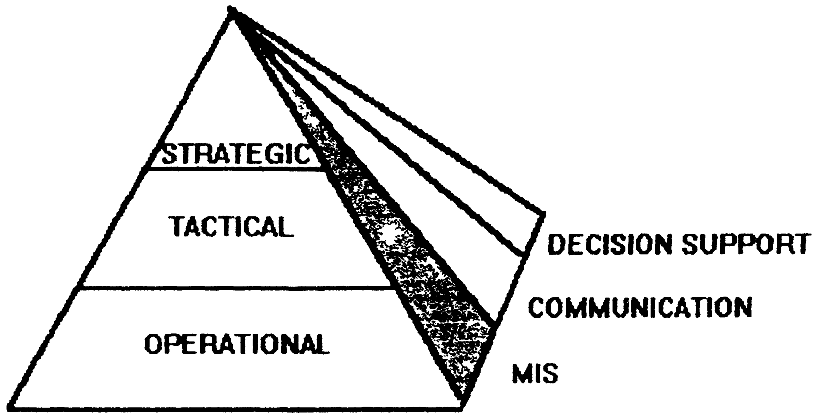

FIG 2 : 3-D VIEW of COMPUTER APPLICATIONS

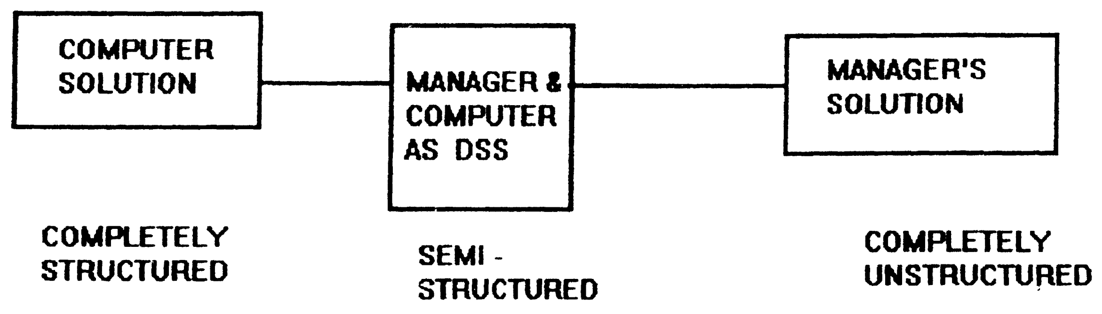

FIG 3 : SEMI-STRUCTURED PROBLEMS

The present revolution of microcomputers and advancements of Local Area Networks (LAN), Wide Area Network (WAN) have had major impact on the structure of DSS. Varied and intelligent software packages have helped in the development of DSS software packages.

This paper describes DSS structure and how it influences software structure in Section 2. Section 3 shows the relationship between menu structure of a typical DSS and Structure Chart of this DSS software package. Distribution of code among different modules of DSS are described in Section 4 based on a number of DSS for varied management applications. In the last section, relevant conclusion and future research directions are discussed.

2. DSS and Software Structure :

According to Sprague & Carlson [2], Decision Support System (DSS) consists of three modules as shown in Fig 4. These modules are :

* Model Management Module (MMM)
* Data Management Module (DMM)
* Interface Management Module (IMM)

This logical structure is not directly suitable for the development of software.

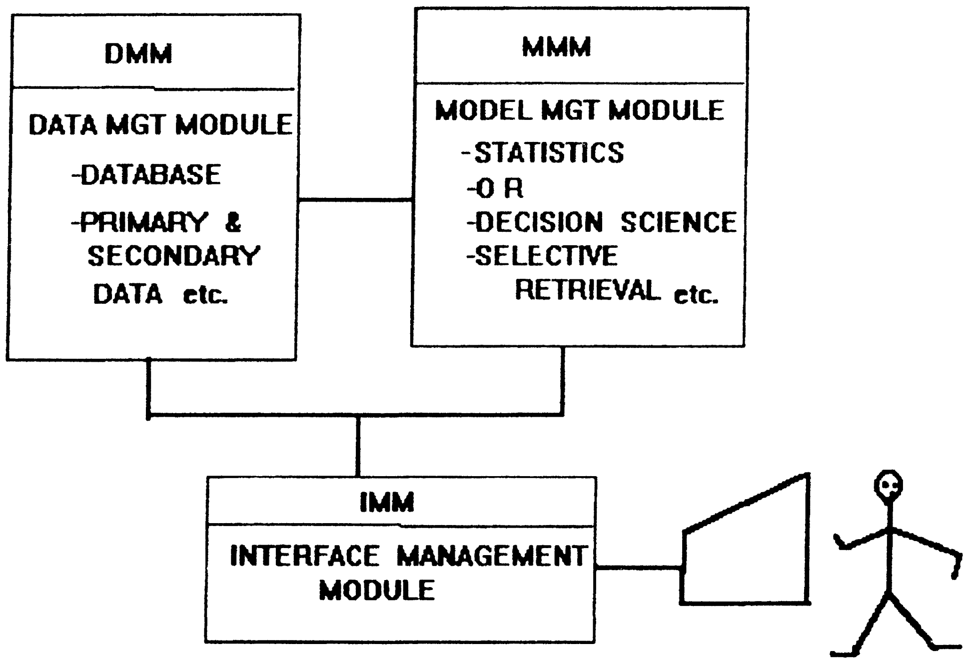

FIG 4 : STRUCTURE of DSS

Most of the DSS software packages depend upon the interaction with users. Processing of data from DMM and solution of models embedded in MMM and display of results in meaningful way so that the user can easily understand the results. He can further interact with DSS. Such a requirement will force a software package having Input (I), Process (P) and Output (O) structure. This structure can be called as I-P-O structure. IBM's Structured Systems Analysis and Design called "Study Organization Plan (SOP)" [3] concentrates on similar I-P-O method of development of the software. Yourdon's Structured Methodology based on Data Flow Diagram (DFD) [4] is also based on I-P-O Structure.

Structured Methodology concentrates on Data Flow Diagram (DFD) during analysis stage. The next step in this methodology is concerned with the Design. There is no specific algorithmic approach for moving from Analysis to Design. Only the guidelines like "Transform Analysis" and "Transaction Analysis" are proposed in this method. Transaction Analysis is based on identification of a Transaction and then taking different actions depending upon the business requirements. Transform Analysis considers the input, processing and output aspects of a transaction. Hence, it may be called as I-P-O structure.

DSS can be easily designed based on this concept of I-P-O. A typical general structure of a software package based on I-P-O structure is shown in Fig 5. As shown in Fig 5, "DSS Program" the main module controls the operation of DSS

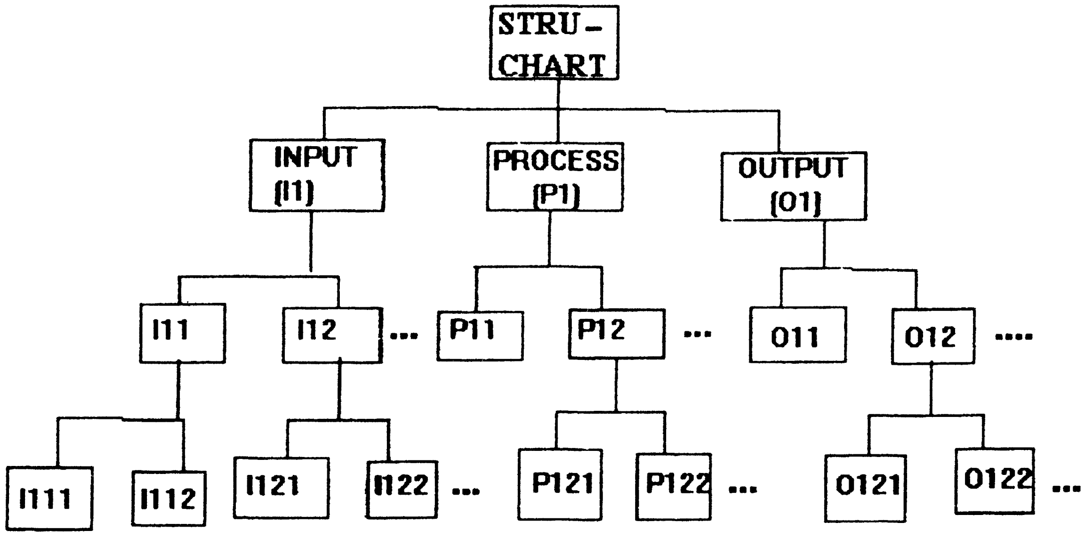

FIG 5 : STRUCTURE CHART of DSS

package. I1, P1 and O1 are the respective modules for Input, Process and Output. Sufficient lower level modules like I11, I12, ..., P11, P12, ... O11, O12, ..., I111, I112, ..., P111, P112, ..., O111, O112, ... etc are designed. Yourdon's Structured Method also uses "Transaction Analysis" for software development. In such a case, based on the data which may be input by the user is used to invoke different software modules. Such characteristics exist in the top modules of the structure chart. This strategy may correspond to the selection of a specific option in DSS. Hence, DSS can be easily built using I-P-O structure of Transform Analysis. Menu Selection can be designed using the concepts of Transaction Analysis.

3. Menu Structure of DSS and Structured Chart :

One of the important characteristics of DSS is its interactive mode of operation. There should be high level of communication between DSS and the user. Generally interaction between users and DSS can be achieved by :

- Command Entry
- Menu Selection
- Icon Selection

Commands demand that the user is conversant with the syntax of the commands. If error occurs, then he has to refer to the manual or go to HELP option of the menu. This style of interaction is the easiest for programming but not attractive from the user' point of view.

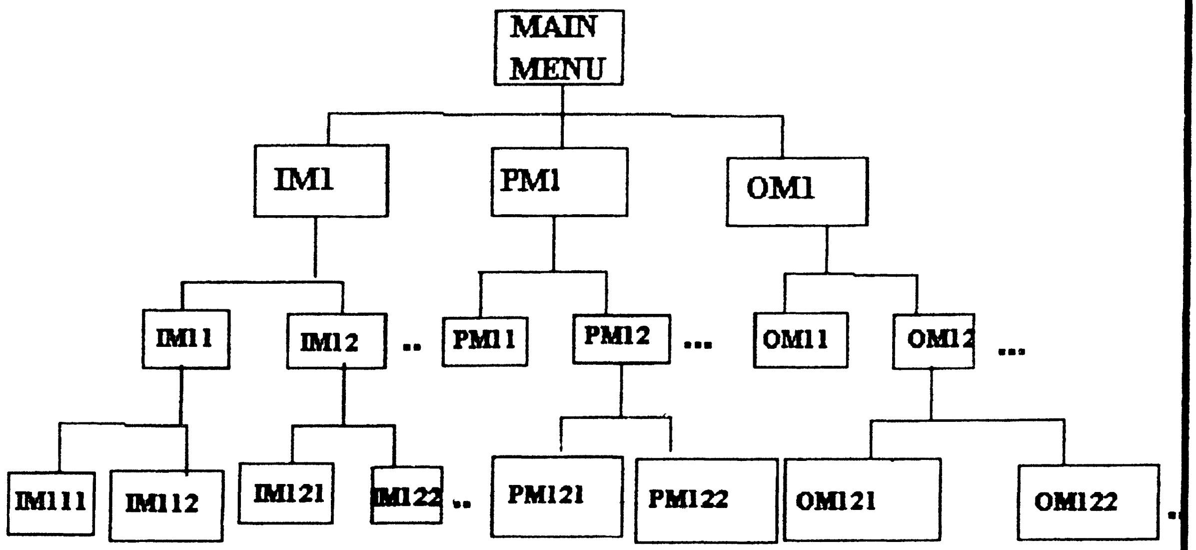

FIG 6 : MENU STRUCTURE of DSS

Menu selection is relatively easy for the user. Such a DSS design will require extra burden on the coding for DSS development. Wrong selection by a user has to be trapped and asked him to correct. Cursor may also move across the selection items of a menu.

In WINDOWS environment, icons have become very popular. Such packages are very attractive for the users. The size of the code of such DSS packages is maximum among the three options of development. Such interaction can be achieved by making DSS very user friendly. Menu driven packages have high degree of acceptability by the users. A typical structure chart as shown in Fig 5 matches with the DSS structure based on I-P-O as shown in Fig 6. Main menu of DSS package corresponds to the "DSS Program" the main module. Menu structures IM1, PM1, OM1, corresponding to I1, P1, O1 in Fig 5 are created. At the lower level (IM11, IM12), (IM21, IM22), (PM11, PM12), (PM21, PM22), (OM11, OM12), (OM21, OM22), (IM111, IM112), (IM121, IM122), (PM111, PM112), (PM211, PM212), (OM111, OM112), (OM121, OM122) etc are designed, depending upon the DSS design. Such a design is good for the development of the DSS software and also it is easy for the users when they have the Menu Structure as a "road-map: for the DSS Structure. Such a software development strategy is easy to follow in the development of a DSS. Hence, there is one-one mapping between the menu chart and structure chart of a DSS package.

4. Distribution of Code in DSS :

A common perception of DSS is that most of the code is. associated with MMM. It is beyond doubt, that the models are important in DSS and they are critical process modules. It is not necessary that such critical modules also have to be the largest modules in terms of lines of code. In fact, it is the main aim of this section to show that most of the code is concerned with Input and Output modules and not concerned with processing modules.

User interaction is most of the time with Input and Output modules. Input modules may accept input after proper display, check for correctness, etc. If it is not proper, then reasonable explanation is given to the user and request him to try again. Error corrections can be very elaborate depending upon the flexibility DSS. Similarly Output may give various options of outputs like screen, print, plot, etc. Depending upon the models, there may be different types of outputs. Based on the results, the user may like to go back to one of the previous menu. Such activities take considerable portion of coding in Input and Output modules.

Generally most of the DSS packages have models from various disciplines like Statistics, Operations Research, Decision Sciences, Economics, Econometrics, Finance, etc. These models are part of MMM (Model Management Modules) of a typical DSS package. Depending upon the design strategy of a DSS package, it is possible to use standard packages to solve the standard models of MMM. In fact, it is always advisable to consider the feasibility of such an option rather than trying to code the model's algorithm. Such a strategy of using standard packages, leads to a DSS package in which most of the code is concerned with Input and Output aspects of DSS package. There will be minimum code to interface with the standard package for providing input data and capturing the output of the package after it processes the input data and gets the solution.

4.1 DSS from CAM Project - Important Characteristics

Let us consider Table 1 concerned with ten DSS Projects developed in Computer Aided Management (CAM) Project [5] a UNDP-DOE funded project at Indian Institute of Management, Bangalore. The present list of ten DSS packages were part of a large number of DSS packages developed as part of this project with the help of various types of users from the industry. All the ten DSS packages are mentioned in the References [6-15]. Interested readers can refer to the corresponding reports for further details of specific DSS packages.

4.2 DSS data from CAM-Project :

The Table 1, gives the details of lines of codes corresponding to Input, Process and Output portion of the various DSS packages. Total of all the three portions of the DSS package is also indicated. It may be noted that the packages are developed using various languages and packages.

Table 1 : Important I-P-O Characteristics of Ten DSS Software Packages

Sl. No.	DSS (Ver No.)	H/W	Language	Lines of Code				Remarks
				Input Lines	Control+ Process Lines	Output Lines	Total Lines	
1	ISPAP (Ver 3.3)	VAX 11/750	FORTRAN	1380	1170	630	3180	GKS for Graphics
2	GOODS (Ver 9.2)	DOS	FORTRAN	4000	2210	480	6690	Hyper Lindo package for LP Solution
3	CHESS (Ver 12.2)	DOS	BASIC Dbase III+	2600	2000	650	5250	BASIC & dBASE III+ (No standard package used.
4	DSS-FM (Ver 15.1)	VAX 11/750	FORTRAN	1250	1380	2640	5270	No standard package used.
5	DSS-SED (Ver 16.1)	VAX 11/750	COBOL BASIC	390	150	7410	7950	No standard package used.
6	(RDBD) (Ver 17.1)	DOS	BASIC, PASCAL, C	3000	3380	3360	9740	No standard package used.
7	MILK (Ver 23.1)	DOS	BASIC	550	200	100	850	Hyper Lindo package for LP solution
8	BUSES (Ver 24.1)	DOS	BASIC	730	2400	600	3730	No standard package used.
9	VAHANA (Ver 26.1)	DOS	BASIC	150	Nil	690	840	GAMS for LP solution.
10	OPTFEED (Ver 28.1)	DOS	BASIC	1420	390	4620	6430	Hyper Lindo package for LP solution

4.3 Distribution of Code in I-P-O Structure :

Based on the Table 1 data, Fig. 7 shows the distribution of codes in I, P, O categories for the DSS packages studied. Processing code is less compared to the Input and Output code. The maximum processing code is less than 4000 lines.

Usually output code is higher compared to Input code. In the present study, Input is lesser than 4000 lines of code, while output code has gone upto 8000 lines of code.

- Usually effective DSS can be designed around 8000 - 10000 lines of code as the 1st version of the DSS package. Acceptability from users is important for the 1st version before elaborate designs are made for the next version.

- Processing code ONLY includes the code necessary for interfacing the standard packages, in case they are to be accessed for the solution of the models of DSS packages.

4.4 Distribution of I-P-O Lines :

The overall distribution of I-P-O codes are :

INPUT	15,530	(31.1%)
PROCESS	13,200	(26.,4%)
OUTPUT	21,180	(42.4%)

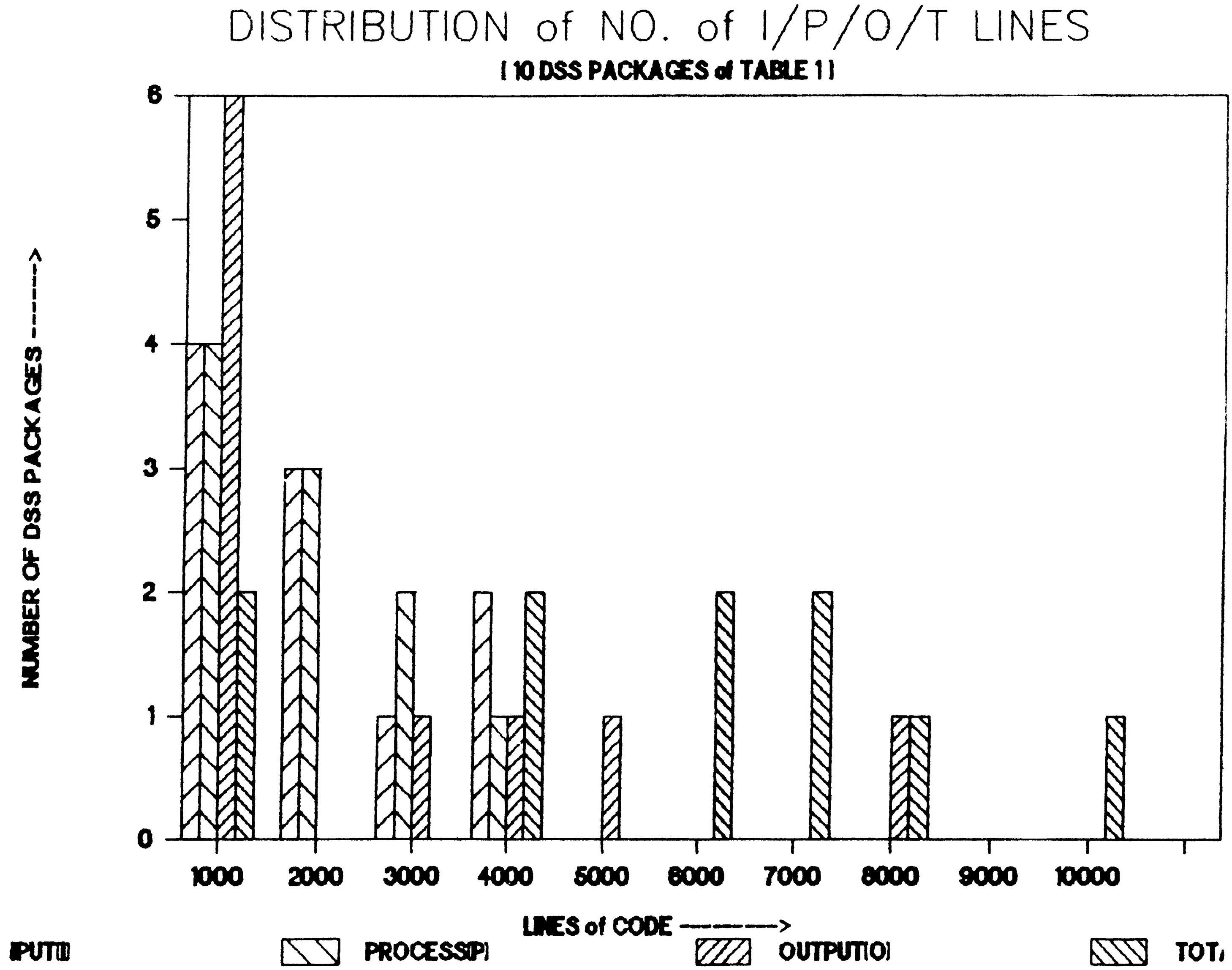

FIG 7 : DISTRIBUTION OF CODE IN I-P-O STRUCTURE

From the Fig 8, one can see that PROCESSING code is the least. Input and Output code adds upto 36,730 (73.5%).

This is the general structure of a typical DSS package in which Input and Output should form the bulk of the code. In fact one can hypothesize that "A software package should not have more than 30% code in Processing with standard package for solution of a model embedded".

4.5 Distribution of Languages for DSS Packages :

It may be recalled that all the DSS packages mentioned in this survey, were developed during 1984-1988 period. As shown in Fig 9, BASIC is used for 53.8% of packages, followed by FORTRAN (23.1). DSS packages were designed so that they are cost effective. It could have been possible to use 4GL tools and other non-procedural facilities.

The developer of a DSS package has better control on the design of the DSS package. It is unlikely that the code will be modified frequently. Hence the user is not expected to change the code. Hence, choice of a 3GL is not out of place, even in the present scenario of available software development facilities.

5 Conclusion :

Decision Support Systems (DSS) will play important role in the future use of computing facility for the middle and top management of any organization.

DISTRIBUTION of NO. of I/P/O LINES

(10 DSS PACKAGES of TABLE 1)

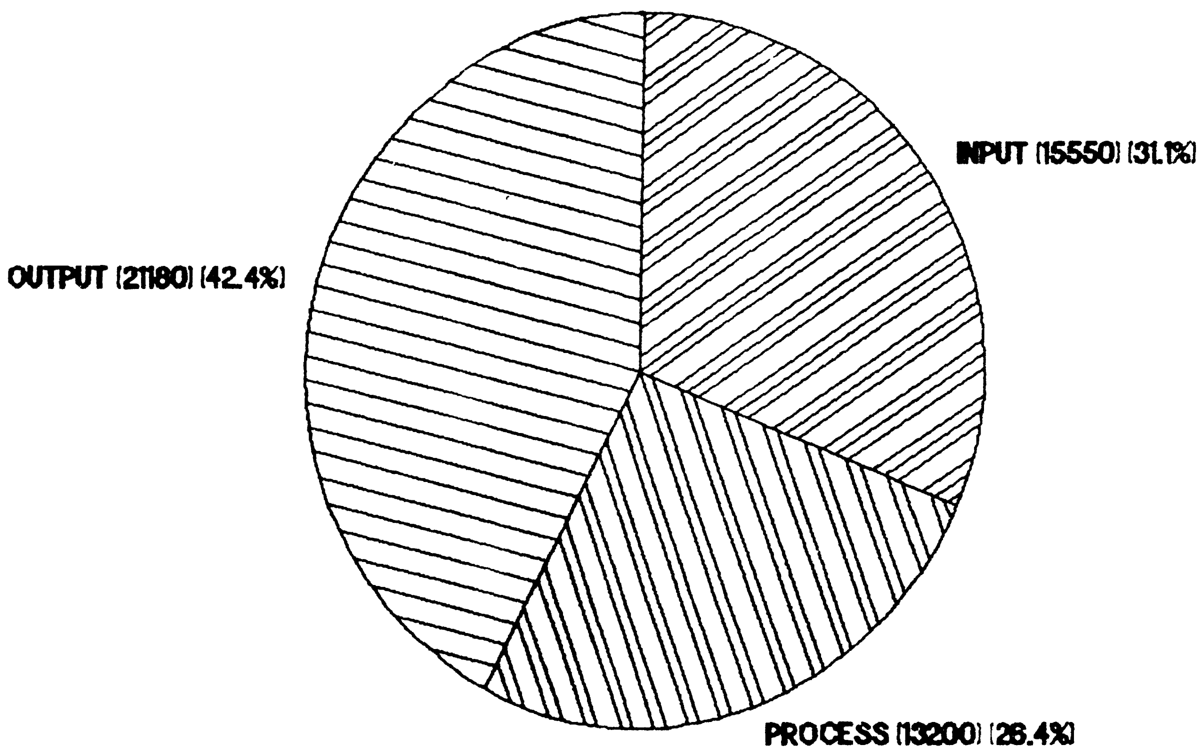

FIG 8 : I-P-O CODE DISTRIBUTION FOR ALL DSS PACKAGES

DSS V/S LANGUAGES

(10 DSS FROM TABLE 1)

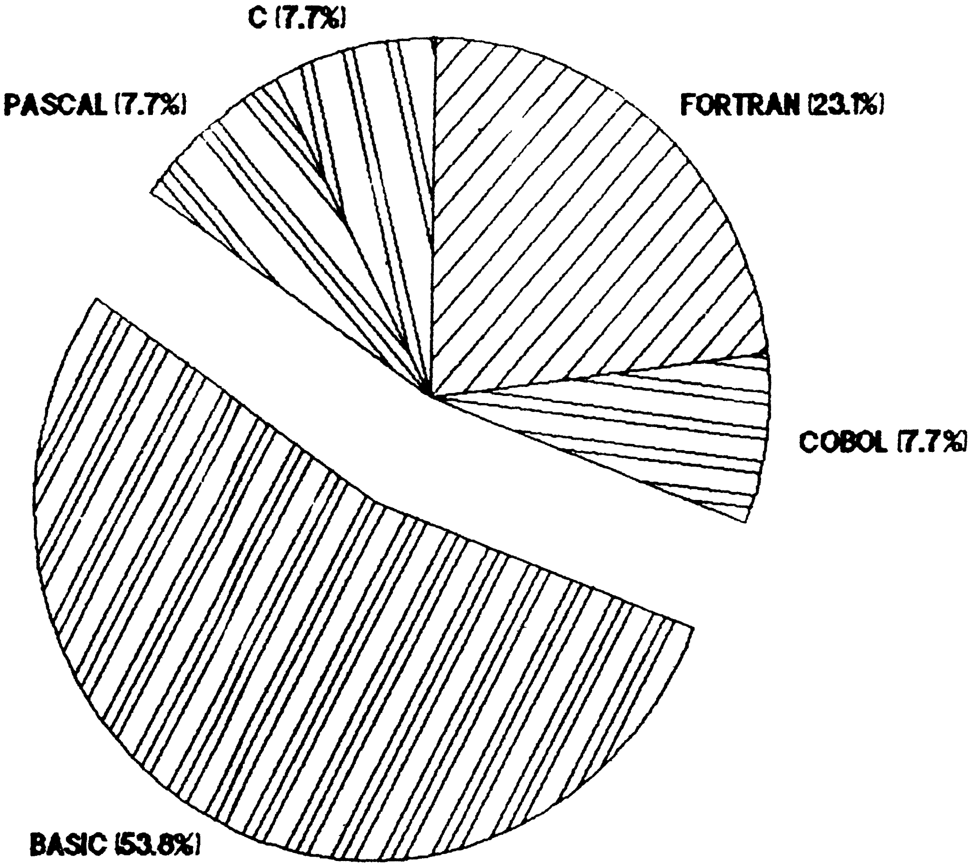

FIG 9 : DISTRIBUTION OF 3GL USED IN THE DSS PACKAGES

- DSS should always have at least 70% of the total code corresponding to Input and Output.

- Processing code should not be more than 30% of the total code when a standard package for the solution of a model of the DSS package.

- It is always advisable to use standard package available off-the-shelf to reduce the code corresponding to process. This helps the developer of a DSS Package to concentrate on Input and Output portion of the DSS package which is very crucial to the success of a DSS package.

- No harm in developing DSS packages using 3GL.

- Acceptability of a DSS package by the users is important for further improvement.

- Spiral Modelling [16] advocated by Bohem is very useful in the development of a DSS package rather than the traditional water-fall-Model.

- DSS packages have close relationship to the recent concept of Data Warehousing (D/w) [17]. Hence there will be a number of interesting facilities for middle and top management by synergetic concepts of DSS and D/W.

References

[1] Keen, P., Morton, M.C : Decision Support Systems ; An Organizational Perspective, Addison-Wesley, 1982.

[2] Sprague, R., Carlson E. : Building Effective Decision Support Systems, Prentice Hall, 1982.

[3] ___________ : Business Study Plan Guide (GE 20-05273) 3rd Ed, IBM Corp, July 1981.

[4] Yourdon E : Modern Structured Analysis, Prentice-Hall International INc, 1989.

[5] Kaujalgi V.B : "Centre for Computer Aided Management - An Overview", iimb Management Review, Vol 1, No.1, July 1986.

[6] Kaujalgi V.B, Rao A.K, Seetha : "GOODS General Optimization of Distribution System" CAM Report No.9, 1989.

[7] S. Jagadish, Siresha : "CHESS - Corporate Health Evaluation Software System, CAM Report No.12, 1989.

[8] Kasi V : "DSS-FM - Decision Support System for Farm Management, CAM Report No.15, 1989.

[9] Kaujalgi V.B, Sridhar, Padmaja : "DSS-SED (Decision Support System for Bombay Stock Exchange), CAM Report No.16, 1989.

[10] Kaujalgi V.B, Chandrasekhar : "tRDBD - Tools for Relational Data Base Design", CAM Report No.17, 1989.

[11] Apte, P.G, Vidhya H.V : "TEX-PLAN (Demand Forecasting & Capacity Planning), CAM Report No.22, 1989.

[12] Nagadevara V : MILK (MilK Industry L.P. Kit), CAM Report No.23, 1989.

[13] Rao, M.R, Kaujalgi V.B., Prasad M.S : "BUSES - BUS Utilization System Evaluation Software", CAM Report No. 24, 1989.

[14] Rao, A.K, Kaujalgi V.B, May Abraham : "VAHANA - Vehicle Scheduling & Headway Analysis for Urban Road Transport Systems, CAM Report No.26, 1989.

[15] Gopalan M.R, Prasad M.S : OPT-FEED - Optimal Mix of Feeds Production, CAM Report No.28, 1989.

[16] Boehm B.W. : "A Spiral Model of Software Development and Enhancement", IEEE Computer 21, 5 (1988) p. 61-72.

[17] Inmon W.H. : Building the Data Warehouse, 2nd Ed. John Wiley & Sons Inc., 1996.
