NOT FOR QUOTATION
WITHOUT PERMISSION
OF THE AUTHOR

MANAGERIAL DECISION SUPPORT SYSTEMS: AN INTRODUCTION

Ronald M. Lee

August 1982
WP-82-72

Working Papers are interim reports on work of the International Institute for Applied Systems Analysis and have received only limited review. Views or opinions expressed herein do not necessarily represent those of the Institute or of its National Member Organizations.

INTERNATIONAL INSTITUTE FOR APPLIED SYSTEMS ANALYSIS
A-2361 Laxenburg, Austria

MANAGERIAL DECISION SUPPORT SYSTEMS: AN INTRODUCTION*

Ronald M. Lee

A term that has gained considerable popularity in recent years is "Decision Support Systems" (DSS). This refers essentially to a new perspective on the use of computers in private and public organizations, one that focuses primarily on the decision making problems of a manager.

The concept of a DSS was originally motivated by a reaction to an earlier concept of a "Management Information System" (MIS), popular in the 1960's and early 70's. Articles about MIS tended to be visionary portraits of a manager surrounded by gleaming technology, running the firm with buttons and levers like the captain of a rocket ship. Much of this was based on predicted advances in computer technology.

As it turned out, computer technology did advance, in many cases more quickly than the predictions. Where progress was lacking, however, was in our ability to apply this technology to managerial problems. By the end of the 1970's, the function of a manager was still little affected by the new technology.

This is not to say that computers haven't affected business organizations as a whole, however. In the U.S. for instance, it is now common for companies large and small to do their payroll processing by computer. Other common applications are various types of bookkeeping functions, sales order processing and recording of inventory information.

---

* Translated version appeared 25 June, 1982 in a special "Informatica" section of the Portuguese newspaper, O JORNAL.

The types of computer applications that have been most successful in organizations are those where there already existed well-defined procedures for doing this activity manually.

ILL-STRUCTURED PROBLEMS

Most managerial problems, by contrast, tend to be "ill-structured". Ask a manager what he or she does and you will likely get an evasive answer like "I oversee production" or "I direct marketing efforts". If you try to get more specific, the replies may be in terms of activitiess like reading reports, issuing orders, signing contracts, inspecting the plant, telephone calls, committee meetings, etc. Still, none of these answers helps much in finding out what types of information processing is done in the manager's decision activity.

The problem is that a manager's activities are almost by definition irregular and difficult to systematize. Those activities that do become systematic are generally passed on to a subordinate, leaving the manager in the role of an "exception handler", dealing only with the unusual cases that aren't covered by a regular procedure.

SUPPORTING MANAGEMENT ACTIVITIES

It is observations such as these that led to the invention of the term "Decision Support Systems". The emphasis is on the second word, support. The view is that managerial activity is much too ill-structured, diverse and dynamic to try to encompasss in a single comprehensive computer system. A more humble, but more realistic goal is rather to support or aid the manager in selected sub-tasks of this process.

Adopting this perspective leads to a much different attitude towards system development of DSS's than for conventional data processing applications.

EFFECTIVE VERSUS EFFICIENT SYSTEMS

An important criterion in data processing is efficiency. The application is justified if it reduces the information handling costs of the organization and/or performs the function at greater speed or with greater accuracy than is possible manually.

The over-riding criterion for a Decision Support System, by contrast, is effectiveness. The goal is to make better -- not cheaper or faster -- decisions. Efficiency concentrates on doing things right. Effectiveness focuses on doing the right things. In data processing, the system's function is generally obvious (e.g. payroll, sales order processing), and the problem is to do this better. In decision support systems, what the system is to do is itself problematic, and the principal focus of effort. While the efficiency of a data processing application is fairly easily measured in terms of cost, it is not so easy to measure improvements in decision making. On the other hand, these improvements are, potentially, of far greater importance to the organization.

NOT COMPREHENSIVE

Another common criterion for data processing applications is comprehensiveness. A payroll program, for instance, needs to include rules for all the possible types of overtime, sick leave, vacations, tax and insurance computations, etc. as well as checking and cross-checking for errors or inconsistencies in the input data.

Again, in contrast, decision support applications don't attempt to encompass a given managerial activity, but only those parts where the computer can provide useful assistance.

EVOLUTIONARY DEVELOPMENT

A third difference between data processing and decision support applications is in the process of systems development.

Data processing applications are developed in a fashion somewhat like building a factory. They both begin with a general specification of the function to be performed. Next a detailed analysis and design is done, which specifies the layout and processsing steps to be performed. Then comes the construction stage when the programs are written. This is followed by a testing period where the system is checked out for accuracy and efficiency. Finally, it is put into production. Implicit in this approach is the assumption that once the system is operational, there should be little need for later revision. (It often happens, of course, that later revisions are necessary. But because of the way the system was built, even small revisions tend to be complicated, time consuming, and costly.)

The approach for developing a decision support system is quite the opposite. There is no overall specification of the system's function, nor are there distinct design and implementation phases. Beginning with the view that managerial needs are largely ill-structured, and vary considerably from one day to the next, the emphasis is not on carefully planned design, but rather on ad hoc functionality -- i.e. the ability to write "quick and dirty" programs that may be used only a single time.

Certain of these ad hoc programs may be found useful for repeated uses. What often happens in these cases is that the manager/user, after experimenting with the initial version of the program, gets many more ideas for enhancements and extensions. Once these are provided, the manager recognizes further extensions, and so on. A DSS thus develops by an iterative, evolutionary process, rather than by single-shot analysis-design-implementation.

Part of the need for this evolutionary, adaptive approach is the fact that managerial decision problems change rapidly with changing conditions inside and outside of the organization.

Another important aspect is that the system, to succeed, must integrate smoothly into the manager's ongoing activities. This typically involves a learning process on the part of the manager as he or she becomes familiar with the technology and how it can be used within his or her particular role. This is a key point: the value of the DSS relies heavily on how well it can be tailored and customized into the manager's work activities. The choice of what the system is to do, the type of interface, etc. lies with the manager, not with what the systems analyst thinks the manager should have.

ROLES INVOLVED

This leads to consideration of the roles of the people involved in developing and using a decision support system. So far, we have discussed the manager as the principal user. Conceptually, we may separate this into two aspects: the person with the decision problem, and the person who actually operates the computer. This second role is sometimes called a "chauffeur"; i.e. just as in driving a car, the manager may have someone else "drive" the system or he or she may drive it personally.

The other role is that of the system developer. This may in some cases also be handled by the manager personally, but more often is a different person. In this case, the manager and system builder need to be in close communication as the system develops. For these purposes, it is important for the system builder to have a background capable of emphasizing with the manager's problems. Indeed, this is often more important than sophisticated computer expertise.

COMPUTER LANGUAGES FOR DSS

This leads us to the issue of the computer language used to develop a decision support system. As noted above, an important aspect is to be able to write programs quickly and to modify them easily. Computer efficiency is secondary (especially considering that computation costs continue to drop substantially from year to year). For these reasons, the computer language used for building a DSS need to be "high level", i.e. with commands using a vocabulary close to the way the problem is originally stated.

So far, decision support systems that make use of a time-sharing computer have been written mainly using the language called APL. This language is somewhat difficult to learn, especially due to its mathematically oriented notation. On the other hand, one can code quite complicated programs in a small number of lines. More recently, micro-computers have taken on importance as a basis for decision support systems. In large companies these have the advantage that because of their low cost, a department can buy one independently, without having to coordinate with the central data processsing department.

Likewise, managers in small companies can often afford to buy one for decision support purposes, even though the company may not be able to afford the bigger types of computers needed for data processing. The language most common on these micro-computers is BASIC. While not as high-level as APL, it is on the other hand quite easy to learn. Indeed, much of the success in the micro-computer market (which in the U.S. is over a million machines) is due to the fact that BASIC is simple enough that people can learn to use it on their own.

FUTURE DEVELOPMENTS

In the future, we are likely to see the development of additional languages for building decision support systems. These will no doubt be even more directly oriented towards the conceptual vocabulary of managers (rather than to computer programmers). A promising area of development (coincidentally the subject of the auther's research), draws on developments from the field of Artificial Intelligence for developing computer languages which can 'understand' a manager's problems directly, in much the way they would be stated to another manager or staff assistant.