DECISION SUPPORT SYSTEMS: TOOLS, EXPECTATIONS AND REALITIES

INFORMATICA 3/87

Janez Grad and Milton A. Jenkins*
Ekonomska fakulteta Borisa Kidriča, Ljubljana, Yugoslavia
* School of Business, Indiana University, Bloomington, Indiana, USA

UDK 519.86

In this paper we describe and analyze Decision Support Systems (DSS) as a class of Information Systems. We first state a short definition of DSS, explain its relation to Management Information System, and the objectives of DSS. Afterwards, our major discussions are devoted to the following issues respectively: (i) The iterative design for DSS (prototyping approach) as part of the broader aspect of DSS building, where we consider cost and benefit impacts and the possible problems and bottle-necks of this approach. (ii) The current computer and information technology that supports DSS, e.g.: on-line computer systems with the necessary software support for interactive use, software packages for computer graphics, statistics, operations research, financial planning, etc., quick hit DSS programs, fourth generation languages, database management systems and so on. The advantages, disadvantages, and limitations in practice are discussed. (iii) The current status and application of DSS for marketing analysis, sales forecasting, financial planning, transportation, human resources management, use of graphics in decision making, etc. (iv) The future trends in DSS. We give a broad overview of the expected development and use of DSS, in particular the further impact of technology and the necessary changes in organizations and decision processes.

Some methodologies and program packages that were listed above were developed in the research programs carried out at the Graduate School of Business, Indiana University, Bloomington, Indiana, USA, and Ekonomska fakulteta Borisa Kidriča, Ljubljana, Yugoslavia.

1. INTRODUCTION

In this paper we describe and analyze Decision Support Systems (DSS) as a class of Information Systems (IS) that support decision-making activities of managers and others who are involved in decision-making procedures and processes, the technology of DSS and its application, and the trends and future directions of DSS and supporting technologies.

We first discuss the relations between Electronic Data Processing (EDP), Management Information Systems (MIS) and DSS. Different views about this issue are briefly stated and explained, taking into consideration the personalities of various authors. The practitioners and the theoreticians frequently view the problems differently. We then concentrate our attention on the variety of different technologies and their applications in DSS. We detail more the most important technologies in DSS, emphasizing those on which some reasonable amount of research has been conducted within the high schools the authors belong to. For instance, data base management, analytical methods, computer graphics, spreadsheets, prototyping and fourth-generation languages (4GLs). In conclusion we summarize the general views and believes on the future trends and developments in DSS and DSS technology. Short-term trends up to 1990, and long-term trends, after 1990, are briefly discussed. We believe these forthcoming technological and organizational changes in DSS will have a significant impact on future high school curriculum and pedagogy in the MIS area.

2. DECISION SUPPORT SYSTEMS VERSUS MANAGEMENT INFORMATION SYSTEMS AND THE OBJECTIVES OF DECISION SUPPORT SYSTEMS

Within the steady advancement of computer-based IS in organizations a new stage has been reached where the term DSS has been introduced. Different explanations of DSS have been provided. Some view DSS as another step in the natural evolutionary advancement of information technology and its use in the organizational context, following EDP and MIS. Others view DSS as an important subject of MIS or just a type of system that has been developed and used for several years already but has only recently been uniquely defined. Some claim that the term has been introduced merely to attract people but it does not define anything particular new in the field of computer-based IS. We discuss briefly these issues in the following lines, state some definitions on DSS and in this way define the subject which we present and analyze in this paper.

Many vague, either restrictive or very broad, definitions of DSS were put forward in the 1970s. They didn't help to clarify which of the above stated explanations were appropriate and which were not. The first definition of DSS as an interactive computer-based system that helps decision makers utilize data and models to solve unstructured problems was later extended to all the systems that contribute to decision making. Some examples of complex systems were also examined. More promising than the definitional approach or the example approach was the "characteristics" approach of DSS [2, 14] that associated with DSS the following characteristics:

- they are aimed at the less well structured, underspecified problems that upper-level managers face
- they combine the use of models of analytic techniques with traditional data access and retrieval functions
- they focus on features that make them easy to use by noncomputer people in an interactive mode
- they emphasize flexibility and adaptability to accommodate changes in the environment and the decision-making approach of the user

In short, DSS should support managers in their decision-making activities. The ideas and research resulted in the form of programmed packages for building DSS, first used on mainframes and later on personal computers.

Based on the previous development and research, Sprague and Carlson [21] gave the following definition of DSS:

"Computer-based systems that help decision makers confront ill-structured problems through direct interaction with data and analysis models".

According to them a good DSS should have the following three capabilities:

- it should be easy to use and should support the interaction with nontechnical users
- it should have access to a wide variety of data, and
- it should provide analysis and modeling in a variety of ways

In order to make a clear distinction between the terms MIS and DSS we must go a bit further in our analysis. There are two general views on what MIS and DSS should be. The first view, the theoretical view, is the view of academicians who are concerned about and who develop theoretical backgrounds of DSS. The second, so called connotational, view is the practitioners' view, who make conclusions and definitions on the basis of their experiences in creating and using some particular MIS and DSS. It is largely the practitioners who support the view that MIS is an advancement of EDP and that DSS is a further advancement of MIS.

Connotational view on EDP-MIS-DSS differentiates the three terms on the following basic characteristics:

- EDP emphasizes optimum data processing with the outputs aimed and used at the operational level
- MIS emphasizes integrated information acquisition environment based on DBMS with the outputs aimed at the tactical level
- DSS emphasizes user decision-making real-time information acquisition environment aimed at top managers and executive decision makers

The view is partially supported by case studies, but it is nevertheless inappropriate as far as the future development of DSS is concerned. Decision-making is not the exclusive domain of the top level management. Decision-making must be distributed across all three functional levels.

The theoretical view proceeds from and is based on the following objectives of the IS function:

- improving performance in order to get the right information to the right person at the right time
- users involved in IS are knowledge workers, such as managers, professionals, and other employees, who also are responsible for further development of information technology
- the paradigm of IS is a goal-seeking organization

This view places DSS among other major technology subsystems of IS which are interacting with each other and other application systems and which are supporting users on all vertical levels of management, not only at the very top level. Other major technology subsystems are (i) the system to support communication needs and (ii) the structured reporting system. This standpoint is defended by a general merging of information technology, operations research, statistics and management science approaches in the form of interactive modeling [21]. DSS is accepted as evolutionary advancement in the systems dimension of a three-dimensional IS model which evolved from the two-dimensional model of MIS. DSS requires new strategies in design of the information systems technology and its interactive usage, from those used in MIS.

Hackathorn and Keen [10] interpret DSS in term of the number of people that participate in the decision-making process as

- independent decision-making where a decision-maker makes decisions. This approach requires personal support
- sequential interdependent decision-making where a decision-maker makes part of a decision, which is then passed on to someone else. The approach requires organizational support
- pooled interdependent decision-making where several decision-makers negotiate and interact in order to make a decision. This requires group support

Experience shows that new approaches and technologies in IS usually promise more than they deliver. Frequently the real contribution is something quite different from what was expected at the very beginning. MIS and office automation diverted from their promises [22]and became well-structured reporting systems (instead of electronic nervous system for organizations) and word processing on personal computers (instead of a paperless office). Nowadays users still believe in the promise of DSS as defined above.

3. DSS CONCEPT, TECHNOLOGY OF DSS, AND APPLICATIONS OF DSS

Traditional data processing systems (such as payroll, inventory, airline and similar reservation systems) since the early 1960s have typically taken the form of predefined reports. Infomation is produced by either aggregating or disaggregating data within the system. The information is static and rigid and any question that had not been included in the process would demand a difficult and time-consuming procedure, including new programs and changes to the data structures. Traditional EDP and MIS are not "user-friendly" (require programmers), their technology, which the programmers use, is rigid, hard to change and demands a lot of programmer time in order to produce results. Questions to the system must be predefined and not some unusual requests from the high-level decision makers.

DSS commonly copes with problems that are not structured, i.e., no procedures for their solution are known. Such problems-decisions, are for example: planning the amount of organizational expenditure, deciding whether to introduce a new product or program, such as a new airline, or a new technology procedure. Decision-making involves multiple criteria and results in a number of trade-offs which are analyzed and modified iteratively by the decision-maker possessing great experience [7]. It requires an interactive computer system with all the necessary support software and a considerable amount of data which the decision-maker uses while exploring the problem. The system must support the decision-maker by suggesting solutions and the possible consequences of accepting a particular solution. Within the subsequent steps of the iterative process the decision-maker may back-track, modify, refine and introduce more sophistication into the solution several times until, he finds a satisfactory final solution. Thus, in this way, a DSS becomes a tool for building a model or creating a solution of the future state of the business, based upon sets of assumptions and relationships supplied by managers and other users.

DSSs need subsystems with data and algorithms that a decision-maker can use. These subsystems can sometimes act independently but they can also be inte rated. Examples of such application are: (1) to retrieve a simple data item while processing an order for a particular commodity, (2) to generate a report of all the firm's foreign customers in the last year, (3) to use mathematical programming algorithm in order to perform allocation planning, (4) to perform a formal statistical analysis to find the correlation between different variables, (5) the use of models, where an expert DSS system has been created, based on the past many years' experiences and decision-making procedures of some expert. The present decision-maker gets help by using this model when he has to make a decision.

The DSS approach is a user-friendly approach. Users acquire information from a DSS without the help from a programmer. They do this by using a query language and/or a request generator. Instead of procedural languages, like COBOL, FORTRAN, etc., DSSs use the fourth-generation languages. In 4GLs one statement is equivalent of many statements in a procedural language. These languages also use system prompts and help commands in order to make them easier to use and understand.

In the following paragraphs we discuss the technology necessary for implementing the DSS concept. We restrict ourselves mainly to the computer software and techniques used in problem-solving.

One of the most important methodologies is a DATA BASE MANAGEMENT. The DSS user (manager) frequently retrieves data items from a DB randomly, produces reports from a DBMS or creates and manipulates more complex logical data structures for a system which allows him to produce models involving data of certain properties. Present DBMS are still part of the software, although in the future they will probably become part of hardware or firmware. They are also available on microcomputers and personal computers, not only on large and powerful mainframes. Software products with imbeded DBMs such as FOCUS, RAMIS II, NOMAD2, EXPRESS, GADS, EIS, REGIS, GMIS, etc. are available on the market. They are expensive to buy (but prices are hopefully declining) and they require large amounts of computing resources [7, 21, 22]. Therefore, initially many of these systems were being used in independent computing companies or in computers manufacturers Efficient use of this technology is based largely on the adequate supporting documentation, such as:

- data flow diagrams, which present a graphic model of processing, the storage of data and the movement of data
- data dictionary, which contains the terms and their definitons
- process descriptions, where each process bubble must be described in sufficient detail [18]

When building the DB management part of DSS we must choose one or more basic data structures, i.e., a method of representing and retrieving data in a computer. In addition to the four well-known data models used in MIS -- the record model (flat file), the hierarchic model, the network model, and the relational model, one more model, the rule model, is being used in DSS environment [21]. This model is common in artificial intelligence systems and in so-called "knowledge-based" DSS. It specifies production rules and enables making inferences based on the data. The rule model describes data by a set of rules, i.e., a set of data definitions. The choice of model should not be based on the representation of the data, but on the operations and integrity constraints.

Among the ANALYTICAL METHODS that DSS needs for analysis and modeling are statistical procedures, data projection or simulation and optimizing models. Statistical packages, such as SPSS and IDA are being used in many universities and firms all around the world. The Interactive Financial Planning System (IFPS) is an example of the financial planning modelling languages for data projection and simulation. It can be used on VAX and some other minicomputers, and mainframes. IFPS has been taught and used in the Graduate School of Business, Indiana University, Bloomington, Indiana, USA as an efficient programming tool for several years, and it has been widely adopted in the USA, in both industry and universities. Its impetus came from the desire to model risk in a way which could easily be understood by executives. IFPS has a self-contained non-procedural language which is easy to understand and use [8]. The logic and output of IFPS resemble those of spreadsheet packages. A number of third-party computer software packages compatible with IFPS are available; For instance: SENTRY for data entry in a form compatible with IFPS; DATASPAN for converting data bases in arbitrary format to a form usable by IFPS; GRAPHICS for presenting IFPS results on color graphics displays; and OPTIMUM for finding optimal solutions of IFPS models by linear, nonlinear, or integer programming. Optimizing models are usually based upon mathematical programming algorithms. A group of researchers at Ekonomska fakulteta Borisa Kidriča, Ljubljana, Yugoslavia has pursued this field of research for many years, developed several program procedures in Operations Research (linear and dynamic programming) and published numerous research papers [4, 5, 19]. Part of this software development was supported by Intertrade, Ljubljana, the IBM representative in Yugoslavia, the Yugoslav computer manufacturers firm Iskra-Delta, Ljubljana and the software house Iskra-CAOP, Ljubljana.

INTERACTIVE COMPUTER CAPABILITIES must be available in DSS environment. The following approaches of DSS computer implementation are now possible:

- DSS software on a large-scale general computer which users access from terminals in an interactive mode
- DSS software on a dedicated minicomputer, where users are involved in different DSS applications simultaneously through their terminals. Here the special DSS software does not slow down other jobs being run on the mainframe. A disadvantage of this approach is that the required central data must often be transmitted to the minicomputer through magnetic tape, etc. which makes the system slow
- the use of personal computers to host DSS of smaller size. This approach is popular because of easy accessibility for the user. Serious disadvantages are far less powerful software than is available on mainframes and minicomputers and relatively small secondary storage capacity. Also the abstraction and loading of central data on microcomputers are even more awkward than in the case of the minicomputers

Some DSS (now being labeled Executive Information Systems (EIS)) help decision-makers by employing built-in EXPERT SYSTEMS -- expert's decision-making procedures. To build such a system, the expert's decision-making process is studied and a computer program is written that behaves similarly to the expert. Expert systems have been used in medical diagnosis, oil exploration, computer chip design, and in various business applications (auditing, making commercial loans, financial planning, etc.).

An especially important technology for business problem-solving and decision-making is COMPUTER GRAPHICS. It helps managers to acquire visual representations of data, relationships and summaries for information activities are not based on predefined processes or procedures. Graphics allows the users to view or search the data in new and creative ways in the context of their particular problems or goals. Several graphics forms can be generated by computers, such as texts, time series charts, bar charts, motion graphics, scatter diagrams, maps, hierarchy charts, sequence charts, etc. They can be used in DSS in many different ways such as: reports, presentations, management tracking of performance, analysis, planning and scheduling, command and control, for design, engineering and prodiction drawings. Graphics can also be used in computer-aided design, computer-aided manufacturing, teleconferencing and videotex systems. The benefits of computer graphics over manual graphics are in costs and time. Formats, scales and colors can be tested in order to obtain the best comprehension of the information. Objections to computer graphics are that high resolution graphics are still very expensive; sometimes the graphs are of low quality, and require skilled and experienced users who can produce good graphics. Some of these objections will become irrelevant with the new computer graphics products.

The Operations and Systems Management department at the Graduate School of Business, Indiana University, Bloomington, Indiana has made in the last decade extensive research on the effects of different forms of computer graphics presentations, their complexity and color of presentation on the human decision-maker, see for example [9]. This line of research is a part of a broader program of research, called PRIMIS - Program of Research for Investigating MIS, which focuses on the user-system interface of DSS. A theory of graphics information presentation has been formalized: Performance with a given information presentation is a function of question difficulty, information complexity, the form of presentation and color.

Another widely accepted DSS technology are SPREADSHEETS. They are self-documenting systems with explanatory internal documentation and prompts that help the user to proceed his dialog with the computer from one step of the problem solving procedure to another. Their main advantages are that the user gets the data in a table form on video screen and the relationships between data series in a form of report. The user may test the impact of some particular data item or group of data items and/or relations among them on the model's output. He can do this by ineractively entering different values for data items, temporarily changing the algorithm and analyzing the computed results. This "what if" capability is present in many spreadsheet packages on the market. These spreadsheet packages are aimed at problem-solving and model-development in fields of financial planning (amortization, depreciation, lease-versus-buy, discounted cash flows and net present value), real-estate investments (finansing alternatives,impact on taxes, payoffs, cash flows), business record-keeping and accounting, budgeting and statistics. Despite their popularity, these spreadsheets have many inherent weaknesses, such as the difficulty of specifying all data requirements a priori, data-model dependence, limitation of the relations that represent the model and model's complexity by the spreadsheet's table format, user's session cannot be recorded and little flexibility in report writing features. It is estimated that 20 to 30 percent of the users will become dissatisfied with spreadsheets and will ask for more powerful tools [22]. Spreadsheets appear to be most useful for smaller problems. The possible solution to the existing variety of many different spreadsheet systems which the users have to learn, would be integrated packages that will combine spreadsheets, word processing, data management, graphics, data communications, and other resources. SYMPHONY is a (not too successful) example of this trend in the software market. SYMPHONY expands the capabilities of LOTUS 1, 2, 3.

QUICK HIT DSS [16] is a term that explains a special procedure used more and more in DSS. It stands for a rather simple DSS prototype which the decision-maker creates and processes before he decides whether to build a full DSS or not. Three types of quick hit DSS include:

- reporting DSS
- short analysis programs
- DSS generators

Reporting DSS is a very frequently used form of decision support which includes simple data manipulations (selecting, summarizing, and listing data from files, some other arithmetic operations on these data, presentation of trends and variances, by means of computer graphics) in order to meet some information needs of decision-maker.

Short Analysis Programs are used for analyzing data. They need small amount of data and are usually written by decision-makers themselves in BASIC or some other high-level programming language. Functions that help the decision-maker to make decisions include projection of costs, income, and profits; allocation of fixed costs among products; project management; graphing of some activity output figures, etc.... Examples of short analysis programs are studied in [1, 6, 16].

Decision Support System Generators are products which include languages, interfaces, and other facilities that help to frame up specific DSS. DSS generator can be used to build more specific DSS within a class of applications. In recent years users are generally interested in DSS generators and fourth-generation languages and not much in the other two types of quick hit DSS.

These quick hit approaches are not appropriate in areas such as forecasting or allocation of resources where a deep understanding of sophisticated methods, techniques and the whole application area is necessary and the final models cannot be replaced by some approximations of them in order to make some starting decision [3].

PROTOTYPING has been defined in many different ways: as a philosophy, a methodology and a procedure. Within each of these classes of definition are further differences. For example, as a methodology for the development of information systems two definitions are very evident: the "rapid prototyping" approach from computer science and the "prototyping methodology (PM)" from MIS. The PM [12] is most appropriate in discussing DSS and is broad enough to encompass the various more limited definitions found in the DSS literature. Under the PM the process for building an operational prototype is described and this operational prototype may be used in various ways, e.g., stand alone, with life-scale methodologies, as pilots or prototypes and as "throwaway" programs. Senn describes prototyping as one of the seven activities within the system development life cycle[20]: preliminary investigation, determination of requirements, development of prototype system, design of system, development of software, systems testing, and implementation. This approach is used when we cannot define all the features of the system in advance, due to the lack of experience or information, or when we face high-cost and high-risk situations. In such cases an inexpensive small-scale version of the software is prepared in order to provide some preliminary information about the environment in which the system is going to work. The prototype is a simple working system that captures the essence of the real system it represents, and it may be refined and redone several times within the iterative process, in order to find the optimum solution for the defined problem.

Prototyping approach needs a software that enables a quick and simple building of a working system. Conventional programming languages and methods, or more adequate software products, like program generators can be used for this purpose. The emphasis is on trying out ideas and providing assumptions about requirement, not on system efficiency or completeness [20]. According to [12], the ideal software prototyping environment has four components:

- a 4GL or other development tool to allow quick creation of the prototype
- well-managed data resources for easy access to corporate data
- a user who has a problem, who has considered the idea of using the new tool, who knows his or her functional area well, and who seeks assistance from data processing
- a prototype builder -- an information systems professional who is versed in using the various development tools and understands the organizations's data resources

The same author also suggests an ideal team size for prototyping -- one user and one builder. Larger teams impose more uncertainty into the problem definition and solving procedure, need more time for coordination, etc... Proper answers to the key output questions represent an important issue of prototyping. These questions are:

- who will receive the output -- what is its planned use
- how much detail is needed -- when and how often is the output needed, and
- by what method

There are three main uses for software prototypes [22]:

- to clarify user requirements. Most users cannot fully describe their current requirements and their future needs. By building, using, and changing a prototype, users can make further decisions about the system they want
- to verify the feasibility of design. A prototype can show the end users of a designed system how the new system would operate, its efficiency and its costs. The end users may find the system inadequate and stop or change it
- to create a final system. Part of the prototype may become part of the production version. Especially, when the system is expected to change very often, it is usefully to use 4GL also for the production version. This makes future use easier

The problems related with prototyping are (i) the need for sophisticated software tools and (ii) that prototyping brings forward only the physical aspects of a system, based on user demands of physical requirements and a quick realization of the needed system. This does not support logical modeling of requirements and solutions. System specifications based on prototypes may inherit inefficiencies and errors of the original system. For this reason, prototyping is most effective when used to enhance, rather than replace, the established analysis process in computer information system development[18].

According to [13] the commercial FOURTH-GENERATION LANGUAGES (4GLs) have provided a significant contribution towards making the concept of prototyping practicable as a methodology for system design and development. They have also helped to create a new information processing environment, referred to as "End-user Computing", despite their primary objective -- to speed up development and maintenance by professional programmers. It is claimed that the productivity in applications development when using 4GLs is 5 to 10 times over that when using third-generation languages, particularly COBOL[17].

The early development of 4GLs started towards the end of the 1960s when time-sharing and DBMS were developed. As the result, three most well-known general-purpose 4GLs software products have become available: RAMIS, FOCUS, and NOMAD. Further, a variety of other user-friendly language interfaces, with elements of non-procedurality, that could be used with the existing DBMS, or previosly developed 4GLs, have been developed (electronic spreadsheets - LOTUS, business modeling - IFPS, business graphics, statistics, etc.). Another significant event, related to the 4GLs evolution was the development of IBM's DB2, which is a member of a family of relational DBMS products from IBM, all supporting a common relational language called SQL (Structured Query Language). IBM announced its entry into the 4GL market in mid 1986, Cross System Product (CSP).

The features of 4GLs that comprise the functionality that is included in fourth-generation software tools are several [22], like: DBMS, data dictionary (DD), non-procedural language, interactive query facilities, report generator, selection and sorting, screen formatter, word processor or text editor, graphics, data analysis and modeling tools, programming interface, software development library, backup and recovery, links to other DBMS, records and file maintenance, etc. The heart of a 4GL is a DBMS, which can manipulate formatted data records, as well as unformatted text and graphics data. Just as important as the DBMS is the DD, for storing the data definitions used by the 4GL. In contrast to 3GLs, 4GLs employ an English-like syntax, and are eventually non-procedural in nature -- they allow statements to occur in the logical order that a user would think, rather than imposing a sequence required by the computer. Besides an underlying command language, many 4GLs provide a variety of interfaces that help end-users in using them. As far as the results (output) are concerned, some 4GLs generate only single programs (code generators), and produce an intermediate step code in 3GL, usually in COBOL, while others generate complete integrated applications (application generators), and do not produce any 3GL intermediate step code (FOCUS, RAMIS II, NOMAD2,...).

The functions performed by 4GLs vary greatly from product to product. Some 4GL products have highly focused but limited functionality, oriented towards specific applications, like decision-support/modeling tools such as IFPS; graphics generators such as Tell-A-Graf; query and report-generating tools such as DATATRIEVE, and INTELLECT. Some 4GLs are more powerful and comprehensive in terms of their functional capabilities, and represent more nearly integrated software systems rather than "programming languages" as the term is commonly understood [13].

First thoughts, that the emergence of 4GLs means the demise of COBOL have been revised when 4GL-related software (analyzers, generators, and programmer workbenches) began emerged. It is now believed that COBOL will continue to be the dominant language of business into the twenty-first century. The 4GL market has not matured yet. The current effort towards 5GL hardware impose a question whether 4GL software will mature at all or simply blend into 5GL software.

4. THE FUTURE TRENDS IN DSS AND DSS TECHNOLOGY

The short-term trends, from 1986 to 1990, will be an extension of today happenings, with a substantial increase in personal and organizational use of computer technology, as high as 70 to 90 percent increase in computer processing power per year. In less than five years, for instance, there are 8 million users of personal computers with an increase by more than 30 percent annually. But even more important than the above stated advancements are the trends in the change in the application of information technology to the point where users no longer face technical intermediaries between technology and its application. The following major trends can be expected [22]:

- personal computer-based DSS will continue to grow, with spreadsheets and other creativity supporting packages taking more and more functions in analysis and decision making
- growth in distributed DSS, with close linkages between mainframe DSS languages and generators and the PC-based facilities
- group DSS approach, supported by local area networks and group communications services, like electronic mail, will become much more common
- DSS products will incorporate products (tools and techniques) of artificial intelligence, instead of the statistical and management science models of the past. "Intelligent DSS" will assimilate expert systems, knowledge representation, natural language query, voice and pattern recognition, etc..., and will be able to "suggest, learn, and understand" tasks and problems

More user friendliness is expected from the computer technology, such as dialog support hardware (light pens, touch screens), high-resolution graphics, speech recognition and synthesis, menus, windows, etc... It has been proven, for example, that for data manipulation the users strongly preferred voice over keying, because they could continually look at the screen while they dictated operations. It is also believed that expert systems, as part of DSS, will be used more than they are now. Many experts still argue on what is and is not an expert system. For this reason, some authors [22] prefer to speak only about practical "expert-like" systems. These are systems, that capture logic of the application problem by means of a small number or even hundreds of IF...THEN...rules, and can be programmed in any high-level language, such as COBOL, FORTRAN, APL, BASIC, LISP and PROLOG or be expressed in a decision table form. What really matters is that they must help users in making better decisions.

Future trends will show no major changes in basic hardware technologies, though speeds and capacities will be improving steadily at about 10 to 20 percent per year rate, physical rate will be diminishing, while the product life-cycle will remain at about three years. In some areas of applications, for example in transactions processing systems, the trends to replace procedural languages with more powerful tools will be very slow.

In the 1990s the personal computer is going to become a management support facility (MSF). It will be widely used in organizations all around the world because of its technological attributes and low cost. Many decision-makers will have MSF both in the office and at home. MSF will have memory sizes of 10-15 megabytes, and secondary disk storage of up to 250 megabytes. Very user-friendly 4GLs, which integrate computing, modeling, data management, and text processing will be available on all devices. More interesting functions of the technology will be related to the decision-support function: the mainframe databases that are necessary in an organization's transactions processing system will be created and combined with large sub-databases at MSF. Certain systems will be developed to work with their local databases in a problem-finding mode which will help the manager in decision-making. One mode of this type of operation is the use of expert systems in order to locate and solve problems. These problems can be categorized as: dealing with a crises, evaluating the overall effect of a change, balancing the use of resources, decisions that must be made on resource replacment or acquisition, and trying to forecast the future. Todays largest expert systems involve thousands of logical rules and thousands of objects to which the rules apply. The goals for the 1990s are the expert systems with tens of thousands of inference rules and up to 100 million objects [22].

Technological and organizational changes will also cause changes in the process of management. Three possible manager groups are foreseen:

- information systems managers, who are responsible for the creation, maintenance, and development of the over-all information systems and its resources;
- user managers, who use the centralized information resources, and create, develop and use their personal and functional area information systems;
- senior managers (executive management) who pursue the information systems and resource allocation policies.

The implications drawn from the application of DSS across all forms of enterprise has great importance to higher education. Universities must produce information literate as well as computer literate graduates who can function in the environment of the modern organization. There is, therefore, a need to provide a general course in information systems that would be taken by all students regardless of their major discipline. We need to produce intelligent users of IS.

The implications for universities offering a major in Informatics are even greater. Here is a constant need to modify existing curriculum to reflect the changes in technology from both a hardware and software perspective. In addition new courses will be required to train information majors in the application of technologies. We believe that in many instances this may best be operationalized by a cooperative effort between academics and practitioners. Where information can be shared, ideas refined and students educated all at the same time.

REFERENCES

1. Alonso, J.R.F. SIMPLE: BASIC Programs for Business Applications, Prentice-Hall, Inc., Englewood Cliffs, N.J. 07632, 1971.
2. Alter, S. "A Taxonomy of Decision Support Systems", Sloan Management Review, Vol. 19, No. 1, Fall 1977, pp. 39-56.
3. Alter, S. DSS-81 Transactions, Execucom Systems Corp., Austin, Tex. 78766, 1981 and an Interview with Steven Alter, Counsilium Associates, Inc., No. 4, Palo Alto Square, Suite 270, Palo Alto, Calif. 94304.
4. Barle, J., Grad, J., Rupnik, V., and A. Vadnal. Računalniško orientirane matematične metode - VI. IMFM, Ljubljana, Oddelek za matematiko, Raziskovalna naloga, št. 85, 1981.
5. Barle, J., Grad, J., and V. Rupnik. Računalniško orientirane matematične metode - X. IMFM, Ljubljana, Oddelek za matematiko, Raziskovalna naloga, št. 199, 1985.
6. Bonini, C.P. Computer Models for Decision Analysis, Scientific Press, Palo Alto, Calif. 94301, 1980.
7. Dickson, G.W. and J.C. Wetherbe. The Management of Information Systems, McGraw-Hill Book Company, New York,..., Hamburg,..., London...., Toronto, 1985.
8. Gray, P. Student Guide to IFPS, McGraw-Hill Book Company, New York,..., Hamburg,..., London,..., Toronto, 1983.
9. Gremillion, L.L. and A.M. Jenkins. The Effects of Color Enhanced Information Presentations, Division of Research, School of Business, Indiana University, Bloomington, Indiana, USA 47405, Discussion Paper No. 173, 1981.
10. Hackathorn, R.D. and P.G.W. Keen. "Organizational Strategies for Personal Computing in Decision Support Systems," MIS Quarterly, September 1981, pp. 21-27.
11. Jenkins, A.M. MIS Design Variables and Decision Making Performance: A Simulation Experiment, UMI Research Press, Ann Arbor, MI, USA, 1983.
12. Jenkins, A.M. Prototyping: A Methodology for the Design and Development of Application Systems, Division of Research, School of Business, Indiana University, Bloomington, Indiana, USA 47405, Discussion Paper No. 227, 1983.
13. Jenkins, A.M.and B. Bordoloi. The Evaluation and Status of Fourth Generation Languages: A Tutorial, IRMIS Working Paper No. W611, Operations and Systems Management, Graduate School of Business, Indiana University, Bloomington, Indiana, USA 47405, 1986.
14. Keen, P.G.W. and M.S. Scott Morton. Decision Support Systems: An Organizational Perspective, Addison-Wesley Publishing Company, Inc., Reading, Mass, 1978.
15. Keen, P.G.W. "Adaptive Design for DSS," Data Base, Vol. 12, Nos. 1-2, Fall 1980, pp. 15-25.
16. Lucas, H.C., Jr. Implementation: The Key to Successful Information Systems, Columbia University Press, New York, N.Y. 10025, 1981.
17. Martin, J. Fourth Generation Languages: Volume 1, Prentice-Hall, Englewood Cliffs, N.J., 1985.
18. Powers, M.J., Adam, D.R., and H.D. Mills. Computer Information Systems Development: Analysis and Design, South-Western Publishing Company, Cincinnati,..., Palo Alto, 1984.
19. Rupnik, V., Grad, J., and G. Resinovič. Ekonomika informacijskih sistemov, Ekonomska fakulteta Borisa Kidriča, FCI, Raziskovalna naloga po pogodbi z Iskro, ZORIN TOZD CAOP, Ljubljana, 1985.
20. Senn, J.A. Analysis and Design of Information Systems, McGraw-Hill Book Company, New York,..., Hamburg,..., London,..., Toronto, 1984.
21. Sprague, R.H., Jr. and E.D. Carlson. Building Effective Decision Support Systems, Prentice-Hall, Inc., Englewood Cliffs, N.J. 07632, 1982.
22. Sprague, R.H., Jr. and B.C. McNurlin. Information Systems Management in Practice, Prentice-Hall, Inc., Englewood Cliffs, N.J. 07632, 1986.