International Institute for
Applied Systems Analysis
IIASA www.iiasa.ac.at

Implementation Essays on Decision Support Systems

Janecek, J.

IIASA Working Paper

WP-80-147

October 1980

IMPLEMENTATION ESSAYS ON DECISION SUPPORT SYSTEMS

Jan Janeček

October 1980
WP-80-147

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

Author's address:

Jan Janeček
Czech University of Technology
Department of Computers
Karlovo namesti 13
12135 PRAHA 2
CZECHOSLOVAKIA

PREFACE

The "Task Force Meeting on Decision Support Systems (DSS)" held at IIASA in June 1980 has stimulated some new thinking in this area of research in the MMT group (Management and Technology Research Area). The discussion pointed out the important role that DSS can play in assisting decision makers. DSS should be seen as a complicated socio-technical system for solving relevant problems in the wider social context.

This paper is oriented towards technical aspects of DSS, but human factors have been taken into account as well. It contains some points of view of implementation; it reviews some basic functions which are to be performed by DSS and techniques which can simplify DSS design. A possible implementation structure based on computer network theory is presented, and in addition, some of the problems involved are discussed.

The author Jan Janecek participated in the IIASA Young Scientists Summer Program 1980. He was attached to the MMT Research Area for three months. This report is one of the results of his work during that period.

ACKNOWLEDGEMENTS

I would like to express my thanks to all who helped me in preparing this paper. The discussions with Gőran Fick improved my notions about DSS, they improved the balance of particular aspects of DSS and cleared up a good deal of confusion.

Alec Lee has taught me to treat the subject more realistically and Alain Barbarie to see the human aspects. Bořivoj Melichar's comments have contributed to the better "pedagogic structure" of the text.

Finally, my thanks go to Kyrstyn Sinclair for her editing and to Mioko Yamada for her secretarial help.

CONTENTS

IMPLEMENTATION ESSAYS ON DECISION SUPPORT SYSTEMS

Jan Janeček

1. INTRODUCTION

Decision making forms an important part of human activity in the social context. Decision making is a goal oriented process; the subject, on the basis of his understanding of the laws and the structure of the objective world surrounding him and on the basis of his knowledge of the goal he wants to achieve, chooses methods and means of exerting his influence on this objective world.

The complexity of the process depends directly on the complexity of the objective world in which the subject makes a decision and on the subject's ability to see and to analyse mutual relations and limits in the objective world. Defining effectiveness of decision making as a function that takes into account the level of goal approaching and the amount of activity exerted, we can see the inverse dependence between effectiveness and the complexity of the objective world (supposing there is some constant level of understanding and the analytic ability of the subject). In order to improve the effectiveness of decision making (or, at least to keep it at the same level when complexities of the objective world increase) we must improve the corresponding abilities of the subject.

Until now the only way to improve the quality of the subject's abilities has been to divide the decision making process into several parts (best achieved by using hierarchical structure). The hierarchical structure, with more than one level of decision making, has enabled organised groups to become the subjects of decisions.

Present computer technology offers us new possibilities of quantitative improvement in that it widens human understanding and analytical ability, as well as further qualitative improvement by enabling automation of some decision making process levels. The purpose of this paper is, primarily, to point out the first impact mentioned above - the possibility of using a computer as a tool to widen human abilities. Such a usage is closely related to our notion of the Decision Support System (and there are no serious theoretical problems in implementing it). The use of computer technology for decision automation could be called a Decision Making System but requires many important results to be obtained in computer science as well as in social sciences - psychology, sociology, etc..

Methodology

Management control is shown in this paper as a hierarchical process with a layered structure of control. The method of "onion skins" used separates particular layers of the system by precisely defined interfaces. The approach permits small parts of the whole system to be treated separately and thus more precisely. The method is useful for the implementation of already designed software as well; it permits the building up and testing of small parts of the system separately, which improves the reliability of the whole system.

The method has been used in areas of operating systems, data bases and computer networks and has proved its advantages in many practical cases.

2. CALL FOR A DECISION SUPPORT SYSTEM (DSS)

The dynamism of the economy increases. Traditional methods of management no longer match the environmental complexity and the speed of change. Methods that information theory and management science offer exceed the capabilities of man. The only means which can help us to come to grips with the situation is the exploitation of computer capabilities.

Computers have been used in economics from the beginning of their existence, but only in separate areas. They have supported information systems, they have been used as a tool for modelling, as a tool for solving optimisation problems. But we need more nowadays. We want to give managers a consistent set of tools that involves information systems, data processing, simulation, optimisation, etc. ([Keen 79], [Wagner 80]). Such a system should also be used by them as a personal tool, always at their disposal and always ready to answer their questions in an interactive way.

A decision support system must be understandable and its management has to be simple enough for non-specialists in computers and for decision makers if it is to be useful. Four characteristics of DSS should be emphasized ([Blanning 79]):

- A DSS must facilitate the interaction between computers and decision makers. It is necessary to bring a computer closer to a decision maker, to put its capacities at the personal disposal of a manager.
- A DSS must assist managers in making unstructured or partially structured decisions and give them the possibility of asking a wide variety of questions in some nonprocedural language.
- A DSS must be comprehensive in terms of the decision process supported or the function performed.
- A DSS must be useful. It must help the decision makers, not burden them with other problems.

To summarise: the aims of DSS should be to assist decision makers, support their judgements and thus improve the overall effectiveness of managers' control.

2.1 Characteristics of management

To show how technological means and methods can be used in DSS design is the target of our work. But first, we must describe albeit briefly the subject we want to improve : decision making.

The task of a decision maker can be defined as the solution of nonstructured or semistructured problems, (by structure we understand here the ability to describe the solving process explicitly, for example by an algorithm.) Moreover, these problems must be solved in an environment that is noted for:

- complexity. Economic systems consist of large numbers of components which are interconnected in a very complex structure. The impossibility of doing experiments makes analysis of component behaviour more difficult.
- the dynamism of the structure.
- indeterminism and uncertainty. Component behaviour is not deterministic in many cases. It can be described by stochastic processes and can be characterised by probability laws, but very often we are dealing with uncertainty; with the inability to specify stochastic behaviour at all.

Three principal types of activity can be distinguished in management - operational control, tactical planning and strategic planning. They differ in their properties and in their position in the managers' control hierarchy ([Keen 78], [Adam 62], [Ayal 78], [Gessford 80])

Strategic planning (long term planning, management strategy) involves the establishment of new goals and new priorities in organisation. It deals with a long time horizon. Methods of strategic planning must deal with a high degree of uncertainty, but the overall complexity should be reduced. Dynamics need not be involved in strategic planning, so we can see it as a static problem.

Tactical planning (short term planning, management tactics) deals with the methods that can be used to achieve the goals previously established. It must involve the state of reality, so its complexity is very large. Dealing with reality brings dynamics and indeterminism to the parameters and to the structure, which must be taken into account.

The operational control (management control) is oriented to those parts of a manager's activity which can be described precisely and automatised by computer programmes. The complexity of reality is reduced by means of a quick answer in a dynamically changing environment.

2.2 *Structure of management control*

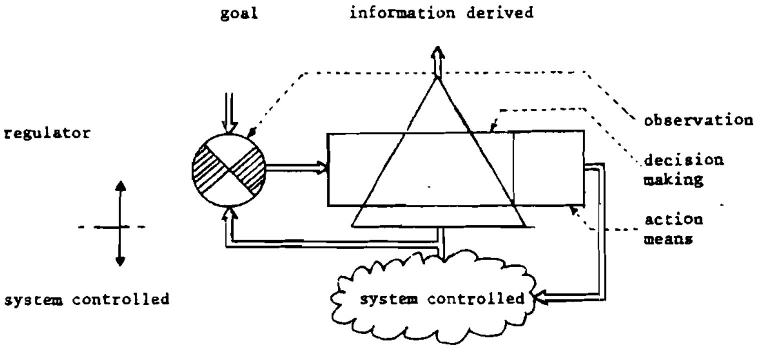

Figure 1.

Looking at the management of an economic system from the control theory point-of-view we can describe every level of activity by the information feedback loop (Fig. 1) Behaviour (or state only) of the system observed is compared with the behaviour (state) required. The decision is made on the basis of the difference between them and on the basis of behaviour observed as well, and the action is applied to the system itself. The goal of the "regulator" is to change the behaviour of the system to the one required. Control function is the primary function of management on each level, but is not the only one. The aggregation of information about the system controlled and transformation into the form and language more convenient for man or higher level control mechanisms are important functions too.

Using this formulation we can see the management system as a hierarchical structure like the one illustrated in Fig.2 (some problems being simplified).

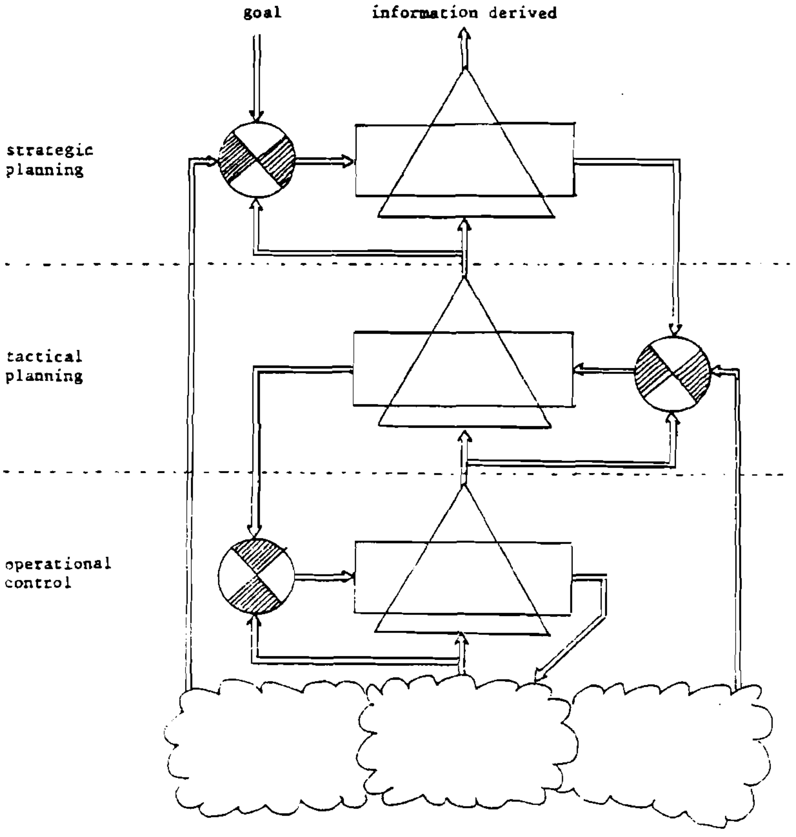

Figure 2.

The higher levels in this structure provide target adjustment for lower levels; the lower levels serve as a tool for implementing actions. Strategic planning, then, sets goals for tactical planning, tactical planning sets parameters for operational control. Conversely, operational control provides actions for tactical planning and at the same time the tactical planning is used to achieve goals set by strategic planning. The aim and the advantages of making such a structurisation for computer network design has been discussed widely in [ISO 79] or [Piatkowski 80]. Only the difference in observation strategy is worth mentioning here.

By structuring management in this way we can demonstrate some properties of management activities. The higher the level of control, the greater the part of the controlled system environment taken into account, so the problems of complexity, risk and uncertainty increase. System stability requirements imply response time requirements. The lower the level of control, the shorter the response time of the regulator must be.

The structuring of management control used here is not the only possibility. A more complex architecture can be used with more levels or more parallel subsystems.

The economic system can be shown as a feedback loop independently of the means used for control purposes. Man could be involved in the loop as well as machine at each level. The use of man as a regulator at an arbitrary level is possible because of his adaptability. But, considering the increasing complexity and dynamics of the system controlled and its environment and the increasing requirements for precision and response time (besides human factors), we must substitute particular functions by technique. This process can be described at separate levels of architecture as successive substitution of manpower in the observation, action and finally also the regulation (or decision making) function. The possibility of such a substitution depends on our ability to structure a particular function and to describe it in terms of the technique used. Nowadays, the boundary between technical means and manpower lies somewhere between tactical planning and operational control, but many functions (observation, action) in tactical planning and strategic planning can be provided by machine as well.

We can submit a definition of DSS based on the facts shown above which will be used in the rest of the paper:

Df: The DSS is a complex of technical means (hardware and software) which will improve effectiveness, precision and speed of all functions which must be provided by managers and supporting executives on the tactical and strategic planning level.

The above definition leaves the decision itself to man, but permits the structuring of DSS into several relatively distinguishable functions which can be discussed separately:

- information function,
- modelling, evaluating and optimising functions, and
- communication function (which, without going into details here, serves to increase effectiveness of man-man communication in management. Man-machine communication is involved in relating part of this paper to cover general principles, but we have to realise that in reality it is a part of the information, modelling, evaluating and optimising functions).

The particular functions will be treated informally in the following three chapters. There are three reasons for this: to survey the range of DSS technical tools, to show that some techniques can improve the quality of DSS or simplify DSS design and implementation, and to build up the criteria for describing the existing DSS systems (Appendix A).

Problems of the decision making process itself are not presented or solved here. The questions concerning:

- formulation of the decision problem (formulation of the problem in some unambiguous language), and
- general problem solving (as defined in artificial intelligence)

must be solved at the next, qualitatively higher level.

3. INFORMATION BASE OF DSS

The similarities between the observation and the aggregation functions of DSS and the basic functions of the information system are evident and some methods of information system design can be used in DSS design as well. Because the solution proposed is oriented towards exploiting existing data bases, the problem of accessing data bases using different data models will be emphasised.

3.1 Data base structures and access methods

Present data base systems have different internal architectures (leaving aside the differences in hardware), data models and access languages.

Let us suppose that the data base architecture corresponds to Fig.3 ([Manola 81]).

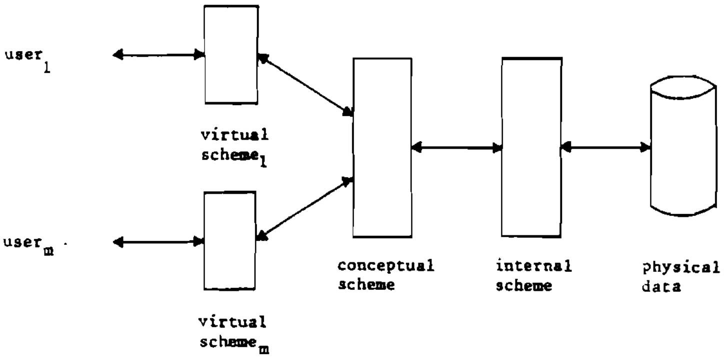

Figure 3.

The data base user is separated from the physical data by several layers of the Data Base Management System (DBMS), by its data schemes and mapping mechanisms. The kernel of DBMS is a layer of architecture that describes the semantics of data stored - the conceptual scheme. The next, virtual scheme (or external scheme) lies between the conceptual scheme and the user. The virtual scheme represents the user's view of the data base, it serves as a window that permits the user to see simply his part of the conceptual scheme and conceals the overall complexity of the data base from him. The mapping mechanism of the virtual scheme permits the user to see his data in the same way (defined by the virtual scheme) independently of conceptual scheme changes; it ensures the logical independence of data. Mappings between the conceptual scheme and physical data transform structures of the conceptual scheme into structures of the internal scheme and structures of the internal scheme into physical data. The two-step mechanism permits the changing of the conceptual scheme as well as ensuring the physical independence of DBMS from hardware used.

The existing DBMS's differ from the described architecture by their absence of a virtual scheme (external scheme, subscheme) and/or by their absence of a layer between the kernel scheme and physical data (in the case where the internal scheme serves as the kernel [Manola 81]).

Several data models in DBMS architectures can be distinguished - the hierarchical, network and relational. The use of different data models on different levels of DBMS architecture is possible; the network model ([Senko 73], [SDD&T 79]) can be used on the internal scheme level, while the conceptual scheme is based on the relational model ([Codd 70], [Chen 76]).

Several classes of DBMS tools, data definition languages (DDL) and data manipulation languages (DML), relate to several models of data. Thus the hierarchical, network and relational languages can be distinguished.

A wide range of languages exists for a wide range of users. Data descriptions similar to the ones in programming languages, and simple directives, are used by application programmers in a programming language frame, for development and maintenance of DBMS. Regular users use parametric and query languages for retrievals and the effectiveness of mechanisms involved is very important. Languages for the casual user have to be simple to use and master.

3.2 Distributed data base architecture

The nature of information and its use leads to the construction of distributed data base systems. The principal reasons for this are:

- The more detailed the data, the more local (in space) is their applicability and the greater their volume. By keeping data in their places of origin and/or use, we shorten the access time and we lower the communication costs and/or memory demands.
- A distributed data base (respecting the previous argument) is more reliable on the whole. Breakdown of one site of the distributed data base has largely local consequences. Important data can be stored in more sites in a parallel way (on account of a complicated synchronisation that ensures consistency [Gardarin 80]), important sites can be doubled (which is cheaper than doubling the whole data base).
- A distributed system can be extended gradually without large, expensive and time-consuming changes. The adaptability of a distributed data base for a different environment and organisation structure is also important.
- The architecture of a distributed data base permits the involvement of existing data bases without changing them.

Constructing a distributed data base involves technological problems. The number of layers in DBMS architecture rises as a result of having to involve existing data bases. The architecture of a distributed data base management system (DDBMS) can be illustrated by Fig.4.

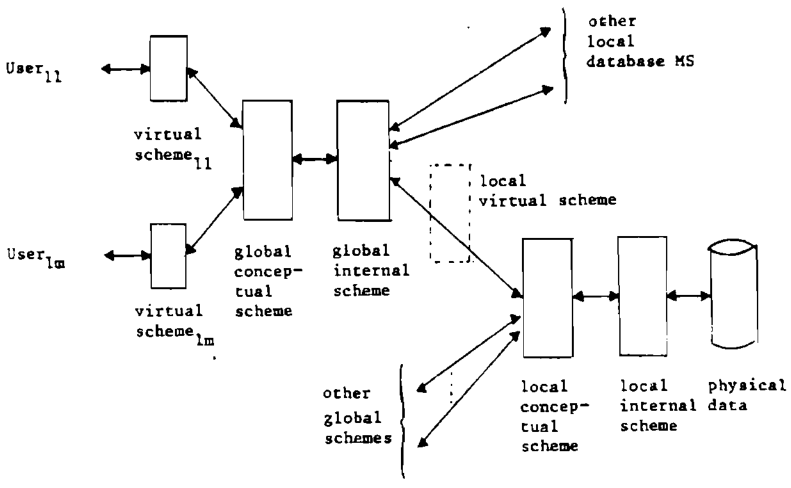

Figure 4.

A global conceptual schema which contains a semantic model of a distributed data base is formed over the set of local conceptual schemes (or local virtual schemes). A global virtual scheme can be added to ensure users' independent view of the global conceptual scheme. The global internal scheme describe the structure of DDBMS (data location, mapping mechanisms between global and local structures). The increase in the number of layers leads to a lowering of DDBMS efficiency as well as to an increasing of system complexity.

Another problem of DDBMS is synchronisation of updates. Local synchronising mechanisms are involved in each local DBMS but another global synchronising mechanism must be added which essentially lowers efficiency and increases the service information flow. In designing DDBMS architecture we aim to limit large and frequent updates to local data bases and to limit global updates to maintenance of DDBMS global catalogues. This goal can be achieved if the structure of DDBMS matches the structure of the organisation which uses it.

3.3 Data aggregation techniques

The user of information systems demands more than mere access to separate data. Frequently he requires a:

- collection of data that fulfils complicated conditions,
- statistical evaluations of large data files (starting from simple characteristics of data files through correlation methods to data filtering),

- nonstandard user-defined data transformations, and
- pattern recognition (the associative searching of data, the recognising of typical situations encoded in data).

These functions result in a smaller data flow from DBMS to user, the user obtains essential informations in the form that he requires.

The functions described can be performed by programmes independently of DBMS or can be included in DBMS itself. The implementation of these functions on user site (on his personal computer, intelligent terminal) improves user independence of DBMS abilities but increases data flow between user and DBMS. The implementation of aggregation functions in DBMS limits user abilities but decreases data flow.

Many different languages can be used to describe user demands for data aggregation. These can be divided into several classes:

- procedural programming languages (user writes the programme performing the function desired; languages such as COBOL, FORTRAN or APL can be used)
- parameter languages (user sets parameters of programme which has been prepared by application programmer)
- descriptive (nonprocedural) programming languages (user describes external behaviour of programme, for example as a set of conditions that must be fulfilled by the data).

The aim is to widen the applicability of descriptive methods that are close to the user's notion of information. Two directions are apparent in this area. The first is to widen their application on nonstandard transformation, the second to bring the language used closer to the natural one and to take advantage of computer graphics.

4. MODELLING BASE OF DSS

Modelling facilities form an essential part of DSS. Very often the DSS are characterised as only the balanced and consistent system of the data base facilities and models.

We will distinguish between two terms in the following text: between model and evaluation technique. The term "model" will be used for the formal description of reality; the mathematical structures such as graphs, labelled graphs, oriented graphs, relational structures or sets of equations are suitable. The model described by the set of equations will be discussed; every structural model is supposed to be transformable into some set of equations (for example, Conway's diagrams can be transformed to grammar of formal language theory). The models which form the basis for decision making procedures (the decision trees and networks) are discussed widely in the classical literature ([Cornell 80], [Gessford 80]. We will discuss largely the models of reality that are able to express dynamic behaviour (based on the concept of [Forrester 73], [Goodman 74]).

The evaluation techniques serve for the evaluation of the model or real world. Such techniques as decision tree evaluation, linear programming, etc. will be treated separately from the models evaluated.

A management system has been described as a multi-level feedback system. The behaviour of its components is generally exponential (exponential, logistic or allometric growth or decrease, periodic behaviour), so the whole system behaviour is of an exponential type. Internal parameters determine whether the system is stable or unstable with respect to the control. It may be very difficult, from the decision makers' point of view, to estimate the response of a system to his decision (or action derived). The possibility of testing the behaviour of a system controlled and consequences of the decision applied are very advantageous in many cases. The decision giving the best response in test conditions can be elected (or some optimisation method can be used instead of this "try and see" method - if it can be used at all) and applied to the real system. A good example of such a system is IFPS [Wagner 80].

The application just described can be characterised as the principal function of the model in DSS. But the learning effect of the "learning by doing" sort cannot be underestimated, neither can the fact that model construction widens our knowledge about reality.

Now we will deal with model construction methods. The methods described are applicable if the structure of the system is understood. Nevertheless they are also useful in the cases where only the external behaviour of the system is known and where relevant internal principles rest hidden to us. Even then, we can design a model that approximates real system external behaviour and we can use this model for reality extrapolation. (An example of this kind of model is the time delay model used in many simulation languages.)

4.1 Dynamic simulation models

Suppose we could describe the behaviour of the system controlled. Seeing the system as static in time, we could use a system of non-differential algebraic equations to describe its equilibrium state. The system of non-differential algebraic equations cannot express dynamic behaviour, so the system of differential or difference equations (DES) must be used.

The complexity and non-linearity of DES means that, in many practical cases, it cannot be solved analytically. We have another way, the only other one we can use, which unfortunately lacks the expressiveness of analytical solution - to create a model of reality, another real system described by the same set of equations. Many methods can be used to build up dynamic models. The methods using computer technology are:

- A structured model on analogue or hybrid computer or on differential digital analyser (DDA). Its advantage is its high speed of simulation run due to parallelism. Its unavoidable disadvantage is that we need special hardware structure for each model.
- A programme for a universal numeric computer which integrates the equation system in a serial way.

The second method is more useful for model construction in management control, where the speed is not as important and adaptability is preferred. But it is necessary to notice here that the difference between these two methods can be pointed out explicitly only because there is a gap in our software technology between DDA and multiprocessor computers which would be filled in the future.

Many languages can be used which range from application-oriented (GPSS,Simscript, DYNAMO, IFPS) through languages or software packages for DES integration (CSMP,GASP) to universal languages in which DES integration can be coded. The universal languages themselves range from simple ones (Algol 60, Pascal) to more powerful ones in which the user-oriented simulation language can be constructed (Simula 67, CLU, ADA).

Attention should be paid to the adaptability and openness of the model, to the possibility of changing it easily and connecting it to the environment represented by the data base.

4.2 Stochastic models

The behaviour of the system controlled is not always deterministic. Some systems can be described as stochastic, such as systems with elements operated by probability laws. The control of stochastic systems is known as control (or decision making) under risk; the system response has a stochastic character. To find the optimal control we must analyse the response of the system. Analytical methods cannot be used in complicated cases; simulation, then,is the only method of solution.

The stochastic variables are replaced by random number generators during the simulation run. The constants of the deterministic model are substituted by random numbers corresponding to the relevant probability law. One run of the model gives one result but we want to obtain the probability distribution of the result. The Monte Carlo method can be used to approximate the result, the large number of runs allows us to determine the character of the response.

Random number generators are elements contained in most simulation languages and some of them can be used for the Monte Carlo simulation. It is necessary to understand the limits of some languages. For example DYNAMO involves random number generators but cannot be used for the statistical evaluation of results. Most simulation languages implement not only random number generators but also the use of a time axis is possible, so the queueing systems can be modelled straightforwardly.

The Monte Carlo modelling method and statistical analysis of results demands greater computer abilities, but their result gives us more detailed information about stochastic system behaviour than we could obtain by running a deterministic model on the median values.

4.3 Fuzzy models

The construction methods of deterministic or stochastic models are based on the objective evaluation of the system modelled, on the proposition that its structure and parameters can be set objectively. In many cases man's subjectivity is involved in the system (a classical example is the physician who constructs the model of his patient's state of health) and, at the least, parameters depend on human judgement, on man's "shady" strategy of perception. We have to deal with subjectivity of judgemental data which cannot be transformed into statistical form.

In these cases a mathematical apparatus can be used which operates directly with "fuzzy" data - fuzzy sets theory and fuzzy logic ([Adamo 79], [Baldwin 79], [Dubois 79], [Negoita 79], [Willaeys 79]). The value of the variable v, the range of which is a set A, is substituted by the membership function $\mu(v)$ defined over set A which describes the likelihood that the variable v has the value a for each element $a \in A$.

Application of fuzzy sets theory brings computing complexity to evaluation and simulation but the mathematical tools used preserve the nature of human judgement and of uncertainty.

4.4 The use of models in simulation games

Application possibilities of simulation models in managerial control are broader than for those of the analysis of quantitative properties of reality. Including a simulation model in the interactive simulation (by designing interfaces between user(s) and model) and in the simulation game (designing game rules and the role(s) of player(s) in the game) we obtain a very powerful tool (Fig.5 [Klabbers 80]).

A simulation game can be used simply as a learning aid (to teach the student about the operation of a complex social system; to teach the manager about the control of his particular part of the organisation) and this is a very important function for DSS ([Bell 75]).

Nevertheless, another aspect of simulation games must be pointed out. Simulation games based on dynamic models of reality are useful in studying individual human behaviour, learning processes and man's abilities to cope with complicated situations. The dynamics of models representing a dynamic environment give us the possibility to observe the dynamic properties of these phenomena ([Schecher 71]).

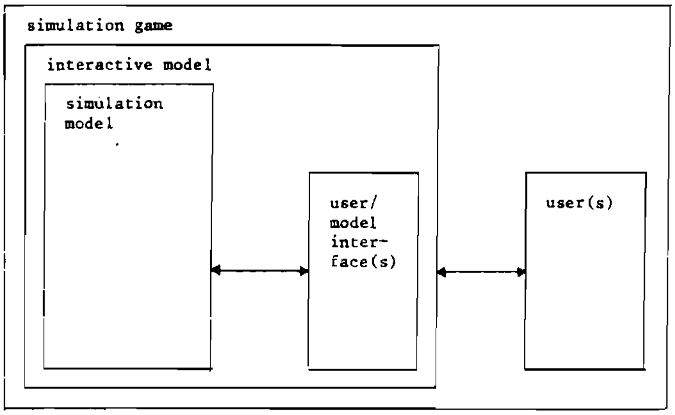

Figure 5.

4.5 *Evaluation and optimisation techniques*

Nowadays, classical methods of managerial control form a basis for decision making and must be taken into account in designing any general system for decision support. Interactive programmes and programme packages exist for ([Cornell 80], [Gessford 80]):

- statistical analysis (multivariate analysis, regression analysis, cluster analysis,...),
- decision tree evaluation,
- payoff matrix evaluation,
- network analysis (PERT, CPM),
- linear programming (the allocation problem, the transport problem), and
- mathematical programming.

The relevant programmes for particular techniques use different programming languages and/or interfaces and are hardly incorporated into consistent systems.

5. COMMUNICATION BASE OF DSS

The management system consists of human organisation as well as technical means. Information about the real world can be derived by objective measuring or is necessarily based on human observation and judgement. Information obtained is further used by technical means as well as by man in the decision process. The decision making itself, understood as the human activity, is supported by technical tools. The accepted decision must be coded into the form acceptable for the lower level of control architecture.

All these activities imply a need for interfaces between man and the technique used, as well as a need for using a technique to improve interpersonal communication.

5.1 Man-machine communication

The man-machine communication problem arises from the difference between the human notion of information and the technical one. Questions arising in this area are:

- how to code unprecise, fuzzy human notions about the real world state into numerical data; how to translate human descriptions of the function desired into the computer programme,
- how to interpret large data descriptions, the results of programmes, to be easily understood by the user; how to generalise the results of separate programme runs, and
- how to design the dialogue between user and machine to be acceptable even for casual users.

The issues form separate areas of research and were studied especially in the data base theory and are closely related to artificial intelligence issues. Some results were obtained that can be generalised and used here as well as in other areas of man-computer interaction (operating systems, programming languages, etc.). For a survey of methods and references see [Melichar 79], [Manna 77].

This recent approach is based primarily on the exchange of text information, but some results have been obtained in computer graphics and speech.

5.2 Computerised interpersonal communication

Research in this area is largely devoted to technical means which can improve the effectiveness of interpersonal communication. Services such as text processing, distribution, storage and retrieval, and text dispatching are being developed and they are combined with telecommunication facilities such as teletext and electronic mail ([Erlandsen 79]).

The issues form part of an office automation concept which is very close to the concept of DSS e.g . the use of teleconferencing tools for the Delphi technique implementation, the use of electronic mail in the playing group or organisation games, etc..

6. DSS AS A "MAN-KNOWLEDGE" INTERFACE

After twenty years of existence, the information systems, in spite of their performance, have remained distant from their users in offices and they have changed the style of managers' work only slightly. Development in personal computing raised the notion that small computers could bring change to classical methods of managerial control; that they could surmount the limits of human mental abilities, the limits of human memory, precision and speed of evaluation. Development in microcomputer technology allows us to place the business computer on the manager's desk as his personal tool. So, the general opinion is that it would be more convenient for the manager to use it and, as a result, would improve the effectiveness of his own work ([Keen 79]).

Unfortunately, the hardware possibilities themselves do not necessarily mean the manager will benefit, or that his work will improve. They only mean that the hardware technology will not limit the areas of application for some time to come.

Is the situation the same in the software area? The answer is necessarily "no" for several years. The development of software tools which form the interface between man and machine is much slower.

And do we know precisely, in every case, what we want to achieve, how we want to change decision making? Do we know enough about the principles of human observation, decision and evaluation to say precisely how technology can help the manager in his task? To be objective we have to answer "no" and we have to add that the solving of these questions will need a lot of time and a lot of research work in the social sciences.

At the same time we must see that every step of development in decision making theory is conditioned by the improvement of technology used in practice (hardware and software) and the developments in technology used must take into account the results of decision making theory. The only means of development is to go through this loop and to change the goals (from the point-of-view of technology) or presumptions about the technical possibilities (from the point-of-view of management control theory).

Our next goal will be to start, in this loop, by finding out what can be done in the software area, respecting the present level of hardware and software technology and the subjective notion of managers' needs. Our goal will be to design the structure of DSS, making it flexible and open enough to changes in the future, a structure which will be able to include small management computers in managers' offices.

6.1 Data base mechanisms of DSS

Flexible data access is the key problem of DSS. DSS implementation must take into account several types of data bases which can be connected to DSS during its period of evolution (e.g. during its life time [Keen 80]). The transformation problem for data access languages can be solved by adding a new level of software which would translate the user's data scheme into the data schemes of particular data bases. The internal structure of this layer includes the picture of the topology and the picture of the semantics of the whole data base system as well as the user's data schemes. The translating mechanisms must translate the user's access language (nonprocedural, learning features, graphical tools) into the internal languages of DSS and, further, into the languages of particular connected data bases ([Adiba 78], [Biller 79], [Bonczek 79], [Cardenas 80], [Mylopoulos 76]).

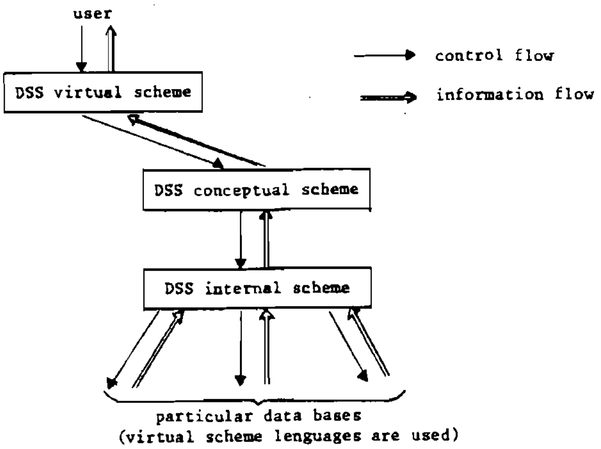

Figure 6.

The DSS conceptual scheme describes the model of the world which is significant for decision making, and is transformed by the virtual scheme into the model of a particular user - decision maker. The internal scheme contains information about DSS topology and semantics.

The problem that must be solved is not the structure (although several simplifications can be applied to lower the overhead of the layered architecture). The problems lie in the description and construction of interfaces in some nonprocedural way, and the methodology to be used to construct consistent access mechanisms and translating functions.

6.2 Modelling abilities of DSS

As has been stated, the problem of the integration of data bases into DSS involves the problem of interfaces. Most contemporary systems for management control are based on the simple communication scheme which can be described by the following oriented graphs of control and data flow (Fig. 7).

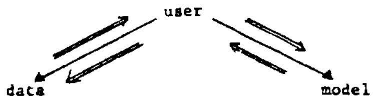

Figure 7.

The function of the user as an interface between data and model lowers the efficiency of the user's work and burdens him unnecessarily with details of the model. The model's direct access to the data is the way to improve the situation, but a model-data interface must be designed for this purpose.

The general scheme of communication which can be used was presented, for example, in [Bonczek 80] (besides the simple classification scheme of DSS languages).

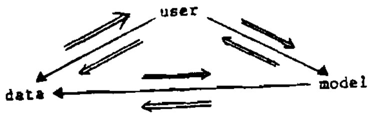

Figure 8.

The oriented graph of control respects the natural flow of control. It allows the user and running models to directly control data access. The oriented graph of data flow is somewhat redundant and can be reduced. Several possibilities exist (a primitive one is presented in Fig.7) for reducing the communication scheme. One of them, respecting the uniform view of the real and the model world is presented in Fig.9.

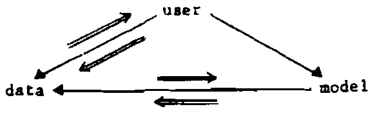

Figure 9.

The results of model runs are obtainable in the data base; the user sees the model world in the same way as the real world (through the data base language).

The other possibility based on the concept of an active data base [Manola 81] is to reduce the data flow graph and to change slightly the structure of the control graph (Fig. 10)

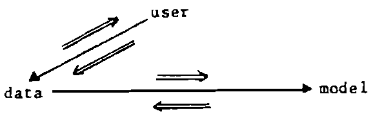

Figure 10.

The advantages of the scheme can be simply derived from that of the active data base, the necessity of this communication scheme arises from intermodel integration.

The model-data interface can use the language of the DSS conceptual scheme (Fig. 11)

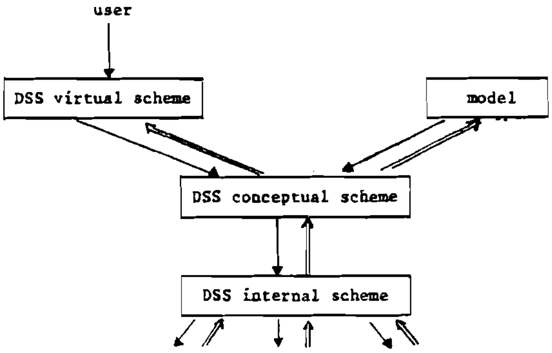

Figure 11.

The problem of integrating the model which uses another data access language, can be solved in the same way as the problem of the data bases integration. The translating mechanism can be placed between the model and the conceptual scheme (the same, in effect, as the translating mechanism between the DSS conceptual scheme and data bases Fig.6)). Continuing with data bases and models integration, the translating mechanism can be involved in the internal scheme (especially noteworthy for remote models).

The content of the DSS data base describes the real world situation. The models need some description of their world too, but that of the real world cannot be used. The copy of a real world description wastes the data base space, so a mechanism must be included in the internal scheme to take care of model and real world data sharing (Fig.12). Information contained in the personal part of DSS data base [Sagalowicz 80] is different from information in the real world data base due to the running of models.

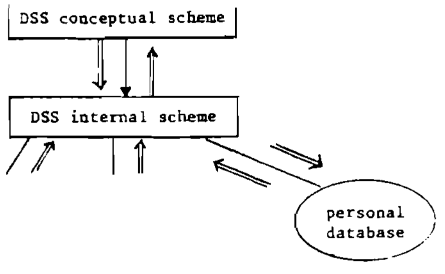

Figure 12.

A personal data base, containing different data, has wider use than that of simply describing the model world; it serves as a data base for personal information, as a mail box or a teleconferencing medium.

6.3 *Logical structure of DSS*

Respecting the previous facts and the possibility of using a personal data base for man-man communication, we can describe the overall structure of a flexible DSS by Fig.13.

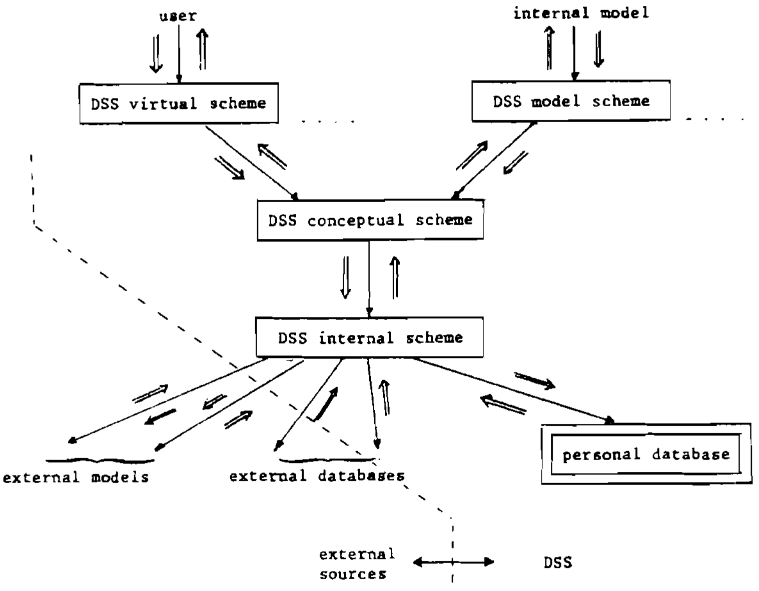

Figure 13.

The scheme represents a logical structure of the system. The physical implementation can use the wide range of means from a single shared processor with back end data storage and terminal network to the distributed system. The possible physical structures that reduce the data flow are presented (Fig.14). The first uses a single processor performing DSS functions, the second uses distributed architecture based on the central DSS function processor and star network of microcomputers performing virtual scheme access functions.

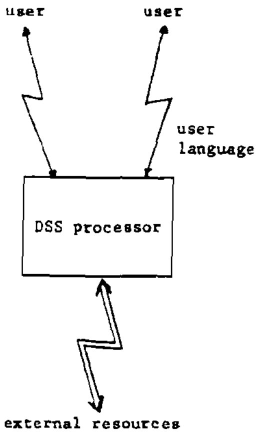

Figure 14a

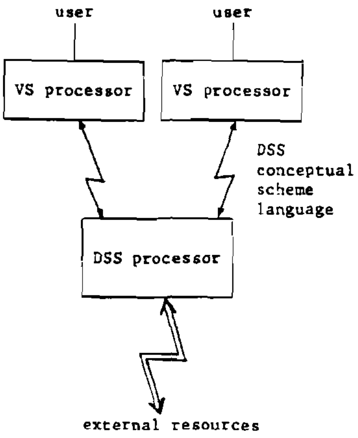

Figure 14b

7. CONCLUSION

This paper has presented the technological point-of-view on DSS architecture, the functions to be performed and methods of implementation.

Summarising the functions mentioned here, the DSS would:

- Ensure access to a number of data bases. These data bases may differ in data schemes, in data access methods and, of course, in the data description and data manipulation languages used. The homogeneous structure (e.g the uniform schemes, access methods and languages) can hardly be taken into consideration for DSS applied to tactical and strategic decision making, where succesive development and openness are presumed. The possible solution would be to include in DSS tools that help the user in constructing translators between his own data scheme and the data scheme of the accessed data base and translators between the languages of these data bases.

  Data bases can also contain libraries of models or submodels of DSS besides the real world description.

- Ensure ways of aggregating large data files into values that condense information for models or for decision makers directly. The language of DSS has to assist the decision maker in constructing his own means of aggregation.

- Be equipped with DSS language which allows user to develop his own models, to describe them in some nonprocedural form (for example as a set of equations, structural scheme, etc.) and to run these models in an interactive way. This approach to modelling helps the decision maker to understand the behaviour of real systems and teaches him to control the real system properly, then to make optimal decisions. Such usage is of great importance, especially for tactical planning purposes where the goal is "well known" and where some dynamic optimisation cannot be used (because of nonlinearity that cannot be removed, or stochastic behaviour, or uncertainty).

- Be equipped with evaluation and optimisation means, and/or by DSS language that helps the user to create them (preferably in some nonprocedural manner). The use of evaluation tools is important for all types of decision making. Optimisation tools (mostly based on linear programming techniques) are necessary for strategic decision making where optimal goals are to be established.

- Include a local (personal) data base for the storage of personal data and containing a "state of the world" for models (supposing the decision maker's access to data possessed by the model will be close to the access to real world data). The data scheme of a local data base can be very similar to the data scheme the decision maker uses, so a relatively simple translation mechanism can be used. This data base can also be used as a private library of models or submodels of DSS.

- Include special data base functions devoted to office work automation (text editing and preparation, mailing and tele-conferencing services).

The possible hierarchical architecture of DSS has been described as well as its physical implementation by a distributed computer system which includes small computer systems. The intention expressed in this paper can be aptly summed up in the words of [Earnest 81]:

> The most important role of small system in DSS is to provide a communication link to vast information and computational resources - to act as a kind of "porthole on the (complicated) world".

The goal of this paper has been to describe the structure of DSS implementation and its complexity. The hierarchy proposed shows the way to a flexible support system, opened to continual change, but a lot of work must be done in the future to transform this idea into reality. Several key areas of work should be mentioned:

- The development of methods of knowledge representation and corresponding languages, and the design of a representation structure (in the form of a data model) for a specific area of management.
- The development of dialogue languages and their implementation methods, the design of a dialogue language for the specific area of management respecting implementation possibilities.
- The development of methods for a construction of the transformation tools of DSS, the design of corresponding languages which would be used by a facilitator during a specific DSS life time.

APPENDIX A

The concept of DSS presented in this paper is that of a computer-based, very flexible tool that can help managers (but not only them) in everyday work. Several approaches have been used for dealing with DSS problem. Below are some examples:

AIDS - An Interactive Decision System

The system for rationalisation of the judgement process has been described in [Newsted 76]. AIDS is very narrowly oriented. It is designed to be used in the construction and evaluation of decision networks in an interactive way. We have to make sure that the system is very simple in its internal structure. All information (facts, their dependencies and probabilities) must be included by the user who practically inserts parameters of the underlying interactive programme. The very good dialogue designed on this basis is worth mentioning, as is the review of many practical applications.

OPM - On-line Portfolio Management System

The system for portfolio managers' support is the typical example reported in the classical DSS literature ([Keen 78], [Alter 80]); a system which proved its usefulness in practice. It performs data access and contains several functions for data aggregation (reports, tables, histograms, interdependencies) as well as a slight possibility to observe the results of the accepted decision (function CREATE). The system differs only slightly from information systems; its data access flexibility and modelling possibilities are somewhat limited.

Brandaid - Marketing Brand Management System

Brandaid [Keen 78] helps in the development of an annual marketing plan. The system is model-oriented and uses historical data for extrapolation. The internal structure of Brandaid reflects the structure of the market and the user can change parameters and submodels. Modelling features help the manager by showing him the results of his decision in the dynamic environment. The system has no direct access to real world data.

IFPS – Interactive Financial Planning System

This is the combination of an information system, modelling tools and evaluation methods [Wagner 80]. Deterministic or simple stochastic models can be described in a non-procedural language. The Monte Carlo method for evaluation of stochastic models and linear programming means are at the user's disposal.

MYCIN

MYCIN [Winston 77] is an example of classical problem-solving system from the artificial intelligence (AI) area. MYCIN has been designed to advise physicians as to the best treatment for a specific bacterial blood infection. Natural language understanding forms part of the system as does the semantic network describing dialogue structure and containing information about treatment prescription rules. The AI systems are to be seen as development tools for problem-solving itself, but their results can be widely used in DSS design.

REFERENCES

Adam Ch.W.: Application of Digital Computers. In: Computer Handbook (ed. Huskey H.D., Korn G.A.) McGraw-Hill 1962

Adamo J.M.: Toward Introduction of Fuzzy Concepts. In: Models and Decision Making in National Economies (ed. Janssen J.M.L., Pau L.F., Straszak A.), North Holland 1979

Adiba M., Portal D.: A Cooperation System for Heterogeneous Data Base Management System. Inf. Systems 3, 3(1978)

Alter S.L.: Decision Support Systems: Current Practice and Continuing Challenges. Addison-Wesley 1980

Ayal I., Zif J.: R&D Marketing: A Management Simulation. Simulation & Games 9, 4(1978)

Baldwin J.F.: Fuzzy logic and Its Application to Fuzzy Reasoning. In: Advances in Fuzzy Set Theory and Applications (ed. Gupta M.M., Regade R.K., Yager R.R.). North-Holland 1979

Baldwin J.F., Pilsworth B.W.: Fuzy Truth Definition of Possibility Measure for Decision Classification. J. of Man-Machine Studies 11, 4(1979)

Bell D.C.: Simulation Games: Three Research Paradigms. Simulation & Games 6, 3(1975)

Biller H., Neuhold E.J.: Semantics of Data Models. Inf. Systems 3, 1(1978)

Biller H.: On the Equivalences of Data Base Schemas - A Semantic Approach to Data Translation. Inf. Systems 4, 1(1979)

Blanning R.W.: The Functions of a Decision Support System. Information & Management 2, 3(1979)

Bonczek R.H., Whinston A.B.: The Integration of Network Data Base Management and Problem Resolution. Inf. Systems 4, 2(1979)

Bonczek R.H., Holsapple C.W., Whinston A.B.: The Evolving Roles of Models in Decision Support Systems. Decision Sciences 11, 2(1980)

Burch J.: Information Systems: Theory and Practice. Wiley 1979

Cardenas A.F., Pirahesh M.H.: Data Base Communication in a Heterogeneous Data Base Management System Network. Inf. Systems 5, 1(1980)

Chen P.: The Entity Relationship Model: Toward a Unified View of Data. ACM Trans. on DBS 1, 1(1976)

Codd E.F.: A Relational Model of Data for Large Shared Data Banks. CACM 13, 6(1970)

Cornell A.: The Decision-maker's Handbook. Prentice Hall 1980

Dubois D., Prade H.: Outline of Fuzzy Set Theory. In: Advances in Fuzzy Set Theory and Applications (ed. Gupta M.M., Regade R.K., Yager R.R.). North-Holland 1979

Earnest L.: A Look Back at the Office of Future. In: Decision Support Systems: Issues and Challenges. Proceedings of a IIASA Meeting (ed. Fick G., Sprague R.H.Jr.). Pergamon Press 1981

Erlandsen E.V., Wedde T., Kommedal E.: The TELIDOK System. In: Teleinformatics '79, IFIP, North-Holland 1979

Fabrycky W.J., Thuesen G.J.: Economic Decision Analysis. Prentice Hall 1980

Forrester J.W.: Industrial Dynamics. Cambridge, MIT Press 1973

Gardarin G.: Integrity, Consistency, Reliability in Distributed Database Management Systems. In: Distributed Data Bases (ed. Delobel C., Litwin W.). INRIA, North-Holland 1980

Gessford J.E.: Modern Information Systems Designed for Decision Support. Addison-Wesley 1980

Goodman M.R.: Study Notes in System Dynamics. Wright-Alen Press 1974

ISO/TC 79/SC 16 N 227 Reference Model of Open System Interconnection. ISO 1979

Keen P.G.W.: Decision Support Systems: An Organizational Perspective. Addison-Wesley 1978

Keen P.G.W., Wagner G.R.: DSS: An Executive Mind-Support System. Datamation 25, 11(1979)

Keen P.G.W.: Decision Support System: A Research Perspective. CISR 54, Sloan WP 1117-80, MIT, Cambridge 1980

Keen P.G.W.: Editorial Plan for Decision Support System Human System Management 1, 1(1980)

Klabbers J., Hoefnagels K., Truin G.J.: Development of an Interactive Simulation/Game: A Casse Study of DENTIST. Simulation & Games 11, 1(1980)

Manna Z., Waldinger R.: Studies in Automatic Programming Logic. North-Holland 1977

Manola F.A.: Database Technology in Decision Support Systems: An Overview. In: Decision Support Systems: Issues an Challenges. Proceedings of IIASA Meeting (ed Fick G., Sprague R.H.Jr.). Pergamon Press 1981

Melichar B.: Nonprocedural communication between user and application software. WP-79-115 IIASA 1979

Mylopoulos J.: Torus: A Step Towards Bringing the Gap between Data Bases and the Casual User. Inf. Systems 2, 2(1976)

Negoita C.V: On Fuzzy Programming in Planning. In: Models and Decisions Making in National Economics. North-Holland 1979

Piatkowski T.F.: The ISO - ANSI Open System Reference Model - A Proposal for System Approach. Computer networks 4, 3(1980)

Sagalowicz D.: Using Personal Data Bases for Decision Support. In: Decision Support Systems: Issues and Challenges. Proceedings of IIASA Meeting (ed. Fick G., Sprague R.H.Jr.). Pergamon Press 1981

Schechter M.: On the Use of Computer Simulation for Research. Simulation & Games 2, (1971) Scott Morton M.S.: Management Decision Systems. Harward University 1971

The SDD&T Task Group of the CODASYL System Committee: Stored Data Description and Data Translation: A Model and Language. Inf. Systems 2, 3(1979)

Senko M.E., Altman E.B., Astrahan M.M., Fehder P.L.: Data Structures and Accessing in Data Base Systems. IBM System J. 12, 1(1973)

Simon H.A.: The New Science of Management Decision Making. Prentice-Hall 1977

Wagner G.R.: Optimizing Decision Support Systems. Datamation 26, 5(1980)

Willayes D., Malvache N.: Contribution of the Fuzzy Sets Theory to Man-Machine System. In: Advances in Fuzzy Set Theory and Applications (ed. Gupta M.M., Regade R.K., Yager R.R.). North-Holland 1979

Winston P.H.: Artificial Intelligence. Addison-Wesley 1977