Centrum voor Wiskunde en Informatica
Centre for Mathematics and Computer Science

J.M. Anthonisse, J.K. Lenstra, M.W.P. Savelsbergh

Behind the screen: DSS from an OR point of view

Department of Operations Research and System Theory   Report OS-R8805   February

The Centre for Mathematics and Computer Science is a research institute of the Stichting Mathematisch Centrum, which was founded on February 11, 1946, as a nonprofit institution aiming at the promotion of mathematics, computer science, and their applications. It is sponsored by the Dutch Government through the Netherlands Organization for the Advancement of Pure Research (Z.W.O.).

Behind the Screen:

DSS from an OR Point of View

J.M. Anthonisse
Centre for Mathematics and Computer Science, Amsterdam

J.K. Lenstra, M.W.P. Savelsbergh
Centre for Mathematics and Computer Science, Amsterdam
Erasmus University, Rotterdam

Interactive planning systems are a relatively new phenomenon in the practice of operations research. We review the role of quantitative models and methods and the need for man-machine interaction in practical planning situations. We discuss a number of desirable functional requirements of an interactive planning system, and elaborate on the concepts of interaction and its realization in the form of a user interface.

1980 Mathematics Subject Classification: 90B99, 68U30. 69H42
Key Words & Phrases: decision support system, interactive planning system, operations research, planning, model, man-machine interaction, user interface.
Note: This paper will appear in *Decision Support Systems*.

Introduction

This paper deals with *decision support systems* from an operations research perspective. We will concentrate on systems that are designed to support decision making in practical planning situations through man-machine interaction. Hence, we will often use the more specific term *interactive planning systems*. In Keen's terminology [Keen, 1986], they would probably be named *extended* decision support systems, as the use of quantitative techniques is as vital as the role of human insight.

For us, DSS represents a novel approach towards the practice of operations research, which has been made possible by advances in information technology. While the mathematics of operations research is a *normative* occupation which intends to develop a theory of models and algorithms, practical operations research is an *empirical* activity in which formal tools are applied to actual problem situations in a heuristic fashion. This is in particular true for DSS. The DSS community has its philosophers and its architects. As designers of a number of specific interactive planning systems, we for once venture to present our views on the area in general.

We will use the term *interactive planning system* (IPS) to indicate a system that provides support with planning activities by the integration of human perception and mechanical algorithmics in an interactive environment. The purpose of the system is to improve the quality of decision making in terms of effectivity and efficiency. In what follows, we first review the process of planning, the role of models, and the need for interaction. We next specify a number of desirable functional requirements of an IPS. We then elaborate on the concept of man-machine interaction, and discuss its realization in the form of a graphical user interface. We finally comment on the types of solution strategies involved.

Report OS-R8805
Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

Planning

Depending upon the tasks of a unit and its level within an organization, it will have different sets of short-term and long-term goals. Depending upon its size, an organization will have more or less formalized planning procedures to define these goals in the best interest of the organization as a whole and to translate these into a plan for the activities of each unit. Planning is a never ending activity. A plan is usually a revised and extended version of a previous plan. The final stage of a planning process is the decision to adopt a certain plan. In the preceding stages, many plans may have been generated, evaluated, compared and rejected. It is a challenging task to develop and implement systems that support this process.

Before starting our discussion of interactive planning systems, we must consider the characteristics of the user we have in mind and the nature of the problems he has to solve. We assume that the user is a trained professional, knowledgeable about his subject area but not necessarily familiar with the techniques of operations research and computer science. The planning situation he is facing is complex in at least two respects. First, the objectives and constraints are numerous and difficult to quantify. That is, it is impossible to construct a model that precisely captures the real-life situation. Secondly, the process required to achieve an acceptable plan cannot be completely specified in advance. Even after the plan has been developed, it may be difficult to say which of the steps taken were directly relevant to the construction of the final plan.

Each generation of users is confronted with a variety of approaches that claim to facilitate their task, each with its own acromym. Before DSS and IPS, we had - and we still have - MIS and OR.

The aim of *management information systems* is to improve the quality of information in terms of accuracy and timeliness. The emphasis is on *registration* of data in the broad sense of the word: their collection, storage, retrieval, and presentation.

The aim of *operations research* is to improve the quality of decisions. The emphasis is on *planning* on the basis of models of decision situations and algorithms that evaluate tentative decisions and generate reasonable decisions.

Models

A model is an abstract description of a decision situation which relates possible decisions to their quality. In a model, decisions and their quality are specified in terms of variables and relations between them. It is illuminating to distinguish two classes of models.

In the first class, the model is designed to *evaluate* decisions. Thus, a tentative decision is input and its quality is output. Simulation models are examples of this approach. Such models are usually defined as computer programs. The user fully governs the search for a good decision, and several decisions may be tried before one is adopted.

In the second class, the model is designed to *generate* decisions. Thus, a desired quality is input and a decision is output. Linear programming is the prime example of this approach. Quality is here a multidimensional notion, stipulating feasibility on a number of dimensions and optimality on another one. In case the linear programming paradigm does not suffice and one of its many extensions - integer, nonlinear or stochastic programming - is called upon, optimization may be too time consuming and approximation algorithms are used.

With evaluative models, different kinds of 'what if' questions can be answered. First, the situation is fixed and the consequences of different decisions are studied. Secondly, the decision is fixed and its consequences in different situations are studied. With generative models, decisions for a variety of decision situations and quality requirements can be obtained and analyzed.

Decision making vs. decision support

The prototypical OR approach is oriented towards *decision making*: 'Give me the problem, then I will give the optimal solution.' This simplistic attitude does not match the complexity of many planning situations. If a single model is chosen to represent such a situation, its solution - mathematically correct or not - may be unusable in practice. This is because no model, no matter how elaborate, can ever be a perfect representation of reality.

It is often prudent to use a variety of models. Each of these is a picture of the actual situation, but different aspects are emphasized or ignored.

Moreover, it is not always known a priori what constitutes a good decision, because the decision maker does not fully specify his tolerances and priorities.

Quantitative techniques cannot substitute the human decision maker, but the reverse of this statement is also true. Instead of lamenting the limitations of either, one should profit from combining the strong points of both: the insight and experience of the planner, and the power and precision of the algorithms. This is what IPS is all about. An IPS aims at *decision support* rather than decision making. It focuses on helping users prepare decisions.

This point of view has some consequences for the realization of an IPS. At the algorithmic side, it must be equipped with evaluative as well as generative models to enable the planner to produce and judge alternative decisions. As to the interaction, it must be able to manipulate massive amounts of data in real time. It is in this sense that IPS merges OR and MIS.

In accordance with the above, we define the following *design goals* for the development of an IPS:
(1) combine the use of operations research models and methods with advanced data access and retrieval functions;
(2) focus on features which make the system easy to use, such as interactivity, computer graphics, and error prevention;
(3) strive for flexibility and adaptability in order to accommodate changes in the decision situation, the interactive environment, and the planning approach.

Functional requirements

These design goals lead in turn to a number of functional requirements for an IPS.

1. *Functional flexibility*. On the one hand, the system should enable the planner to define and modify a plan. It is then used as an *automatic scratch pad*, which supports the traditional manual planning in a modern way. It provides facilities for the storage, retrieval and display of data of problem situations and decisions; in this respect, it resembles an MIS. It is also able to evaluate the quality of a given plan. The system acts as an *assistant* to the planner.

On the other hand, the system should be able to construct a complete plan and to modify an existing plan by itself. It has now the role of an *automatic pilot*. In addition to the registrative and evaluative facilities, it provides the means to generate a plan of a given quality. The system acts as an *advisor* to the planner.

The roles of assistant and advisor are the extremes of a broad spectrum and there is much inbetween. When the user constructs a plan by hand, he may do so on the basis of suggestions provided by the system at various points. When he completes a plan in this way, he may ask the system for possible improvements. Alternatively, he may construct a partial plan and leave it to the system to complete it; the result can then serve as the starting point for manual modifications. The number of possibilities is virtually unlimited, and it depends on the entire context which style of planning is employed most frequently. Even if the system does not go beyond the role of assistant, it is already a useful tool for planners. It is always the user who is in charge, even if the system functions as advisor.

2. *Ease of use*. If the system is easy to use or 'user friendly', the planner can concentrate on solving the problem at hand. This is a hard job under any circumstances, and a system perceived as difficult to operate may go unused even though of potential value. Features that contribute to ease of use are the following:

(a) *Simplicity*. Features that are not simple to understand will not be used. It is often difficult for the software engineer to detect troublesome aspects of his design. These aspects do become apparent, however, when the functional description is written. They can be avoided by completing this document before implementing the system or at least by having feedback from specification to implementation. Anything that is difficult to explain will almost certainly be difficult to use.

(b) *Consistency*. A consistent system is one that behaves in a generally predictable manner. Function names and calling sequences, graphical representations and colors, all these should follow simple and similar patterns without exceptions. The user is then able to build a conceptual model of how the system reacts; in new situations, he can apply his knowledge with a good chance that it will work. Again, inconsistencies often show up when the functional description is written.

(c) *Completeness and conciseness.* The system must contain a complete and concise set of functions that allow the user to handle his problem effectively. There should be no irritating omissions or redundancies. The strength of the system lies in the coherence of the functions, not in their number.

3. *Robustness.* Users are capable of an extraordinary misuse of the system, either through misunderstanding or for enjoyment. The system should accept such treatment with a minimum of complaint. When the user does something unexpected, the system reports the error in the most helpful manner possible. Only in extreme circumstances errors cause termination of execution, as this generally results in the loss of valuable information.

Interaction

Until now, we have discussed the issue of man-machine interaction in fairly broad terms. We will be more specific in this section.

In the last decade we have witnessed extraordinary advances in information technology, which have resulted in enormous increases in processing power and graphics capabilities. There is now an alternative to batch processing and centralized operations. Due to the practicality to perform intricate computations in real time and to display data and results in an informative way, it is a feasible idea to involve humans throughout the planning process.

Interaction is possible, but why is it desirable? The brief answer is that planning problems tend to be both *hard* and *soft*.

Most practical planning problems are, in any reasonable abstraction, *NP*-hard. This implies that these problem types are probably inherently intractable in a well defined sense [Garey & Johnson 1979]. For practical purposes, it indicates that the solution of realistic problem instances to optimality may require an inordinate computational effort. We have to resort to approximation algorithms, that deliver acceptable solutions within an acceptable amount of time. It is just one step further to embed such algorithms in a heuristic setting. The solution is then found by means of a trial-and-error procedure, in which man and machine divide the tasks in accordance with their respective capabilities. In *interactive optimization* [Fisher 1986], the user controls the solution process by setting initial parameters, selecting algorithms, and adjusting solutions. Jones [1987] introduces the term *grey box* for this type of optimization: the traditional single black box is replaced by a network of black boxes with user intervention required whenever one of them completes execution. In this way, the human planner guides the computer towards promising parts of the solution space.

Another aspect of real-life problem solving is that the notions of *feasibility* and *optimality* are not as precise as in mathematics. Most planning problems contain subjective elements that are difficult to quantify. Feasibility requirements may be soft rather than strict, and tradeoffs between optimality criteria are often not explicitly known but carried implicitly in the value judgement of the decision maker. Interaction is one way of coping with this aspect [Fisher 1986]. While the planner constructs (or modifies, or extends) a plan, he may override constraints; the system should warn him as soon as violation occurs, but it is the planner who determines feasibility. Similarly, the planner decides about the comparative evaluation of the objectives. He has full control and responsibility.

As a consequence, interaction adds to effectivity, efficiency, and acceptability. First, the cooperation between man and machine leads to better solutions. The machine cannot be beaten in solving well-defined detailed problems. The human planner is superior in guiding the overall solution process, in recognizing global patterns, and in observing all kinds of *ad hoc* constraints. Secondly, these better solutions are obtained faster, because interaction allows for flexibility in manipulating data and in selecting alternatives. Finally, an interactive system is more readily accepted. The human planner is not replaced but gets a versatile tool.

User interface

Now that we have indicated *why* interaction is a desirable feature of the planning process, we discuss in general terms *how* interaction has to take place.

The user interface is the part of the system that provides the means for communication between man and machine. It essentially consists of two languages [Bennett 1983]. The first one is the *presentation* language, which is employed by the machine and understood by the user; it expresses what the user sees or senses as context for interaction. The second one is the *action* language, employed by the user and understood by the machine; it expresses what the user can do in order to change the context in a way which will help him to meet his goals. By 'language' we mean the collection of patterns of signs and symbols which one participant in the interaction (man or machine) is allowed to use in presenting information to the other participant.

An IPS should be able to present problem instances and solutions in a meaningful way, i.e., one that permits a quick assessment and analysis of the data being presented. The principal usefulness of computer graphics is the possibility to provide different, and perhaps more insightful, representations of the same data. The use of *iconic* as well as *representational* graphics is clearly relevant here. Iconic graphics display part of the real world, such as a road network or a facility layout, while representational graphics display data summaries, such as bar charts and pie charts. Noteworthy research in this area is Tufte's [1983] work on the visual display of quantitative information and Jones' [1988] attempts to develop novel representations of machine schedules. The benefit of computer graphics to decision making, however, is still a topic of debate. DeSanctis [1984] summarizes the literature on this subject and arrives at several propositions based on persistent trends.

The effect of a graphical user interface can even be stronger if *color* graphics are used. Colors provide an easy way to distinguish between various objects. One should color with taste, however; an excessive use of colors may confuse the picture.

As to the action language, we have already mentioned ease of use as a major functional requirement of an IPS. Simplicity, consistency, completeness and conciseness are, of course, worthy goals in the design of any computerized system. For an IPS they are especially important, and the action language is the prime feature of the system that will reveal whether these goals have been achieved.

All in all, an IPS should provide an interface which the user can interpret easily and control effectively. The design of the user interface is a principal component of the overall design process. Many guidelines have been proposed for this purpose; the recent book by Shneiderman [1987] reviews the subject area. At the risk of repeating ourselves, we emphasize the two central issues: focus on a limited number of well-chosen *representations* and operations on them, and provide *uniformity of structure* so that the user can take the interface for granted as he concentrates on the problem he is solving.

Solution strategy

A few words are in order about the types of solution strategies that are appropriate in the context of IPS. The user of an IPS usually follows some sort of divide and conquer scheme in arriving at an acceptable plan. This scheme often involves a certain decomposition of the problem, based on an aggregation of detailed information or on a selection of attractive alternatives. In current practice, one sees that the user tends to adhere to a single fixed scheme, which has been acquired by habit or is imposed by the system.

A next generation of IPS might provide more flexibility. We are thinking of an IPS that suggests a solution strategy on the basis of characteristics of the particular problem at hand. In terms of the previous discussion, the system itself is involved in determining the structure of the grey box. Such a system should be able to manipulate problem types and solution methods rather than just problem instances and solutions.

Summary and conclusion

To summarize our views, we find that interaction can play a vital role in complex planning situations by integrating human insight and formal models. Many planning problems are too hard and at the same time too soft to be amenable to solution by purely algorithmic techniques. A variety of evaluative and generative models, meaningful representations of problem instances and solutions, and a uniform set of actions to manipulate all these are the main constituents of an IPS which realizes functional flexibility, ease of use and robustness. For what we, with an understandable bias, view as an illustrative implementation of these ideas, we refer the reader to Anthonisse, Lenstra and Savelsbergh

[1987] and Savelsbergh [1988].

By way of conclusion, let us briefly contrast traditional OR with this concept of an IPS. We have already mentioned that, on the practical level, decision making is replaced by decision support. On the technical level, the algorithms are no longer as prominent as they were. The most visible part of an IPS is the user interface. Its only purpose, however, is to create the opportunity to manipulate information in a convenient way. Whether information and manipulation make sense depends on the context, which consists of the practical planning situation on the one hand and the formal models and methods on the other. One might say that the role of information technology pertains to the form, while practice and its abstractions provide the substance. For the OR researcher, an IPS is like the wooden horse of Troy: it enables him to disguise his weapons in an attractive fashion and to bring them closer to practice.

References

J.M. Anthonisse, J.K. Lenstra, M.W.P. Savelsbergh (1987). *Functional Description of CAR, an Interactive System for Computer Aided Routing*, Report OS-R8716, Centre for Mathematics and Computer Science, Amsterdam.

J.L. Bennett (1983). Analysis and design of the user interface for decision support systems. J.L. Bennett (ed.). *Building Decision Support Systems*, Addison-Wesley, Reading, MA, 41-64.

G. DeSanctis (1984). Computer graphics as decision aids: directions for research. *Decision Sci. 15*, 463-487.

M.L. Fisher (1986). Interactive optimization. *Ann. Oper. Res. 4*, 541-556.

M.R. Garey, D.S. Johnson (1979). *Computers and Intractability: a Guide to the Theory of NP-Completeness*, Freeman, San Fransisco.

C.V. Jones (1987). User interfaces. Unpublished manuscript; E.G. Coffman, Jr., J.K. Lenstra, A.H.G. Rinnooy Kan (eds.). *Handbook in Operations Research and Management Science, Volume 3: Computation*, North-Holland, Amsterdam, to appear.

C.V. Jones (1988). The 3-dimensional Gantt chart. *Oper. Res.*, to appear.

P.G.W. Keen (1986). Decision support systems: the next decade. E.R. McLean, H.G. Sol (eds.). *Decision Support Systems: a Decade in Perspective*, North-Holland, Amsterdam, 221-237.

M.W.P. Savelsbergh (1988). *Computer Aided Routing*, Doctoral dissertation, Centre for Mathematics and Computer Science, Amsterdam.

B. Shneiderman (1987). *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, Addison-Wesley, Reading, MA.

E.R. Tufte (1983). *The Visual Display of Quantitative Information*, Graphics Press, Cheshire, CT.