Organizational experiences with multicriteria decision support systems: problems and issues

Jean-Charles Pomerol & Patrick Brézillon
LAFORIA-IBP, Case 169
University of Paris 6
4, place Jussieu, 75252 Paris Cedex 05, France
{pomerol, brezil,}@laforia.ibp.fr

Abstract

We review some experiences with multicriteria decision support systems (MCDSSs). We show that MCDSSs have been able to deal only with the 'choice' phase of decision rather than with tasks involved in developing criteria and alternatives. This strongly limits the use of MCDSSs within organizations. Designers should focus their efforts on alternatives. This weakness of MCDSSs in dealing with alternatives can be allievated by emphasizing the cooperation between the decision maker and the system. We stress the interest of modeling cooperation, explanation and context to improve MCDSSs acceptance. Beyond MCDSSs some organizational lessons extend to other DSSs.

Key words: Decision support systems, Multicriteria Decision, Cooperation, Explanation, Context.

1. Introduction

As organizational complexity increases, it becomes very unlikely that a comprehensive model of a given decision situation can be developed. This implies that the model is completed by the interaction between the decision maker and the system [1]. We thus enter the realm of decision support systems (DSSs). As most human decisions are multi-attribute, multicriteria DSSs (MCDSSs) should be widely used in organizations. Surprisingly, this it is not the case and at the end of the eighties we find only a few MCDSS implementations [2].

Actually, as observed earlier for various systems, including knowledge-based systems [3,4], MCDSSs fail to be used because they do not:

1) contain enough information to be realistic;
2) cope sufficiently wisely with uncertainty;
3) pay enough attention to contextual knowledge;
4) capture user preferences (of course, multi-attribute preferences); and
5) have sufficient organizational support.

As regards MCDSSs, the three first points refer to the quantity of knowledge contained within the alternatives (including the consequences of the alternative for the future). We will show that it is often difficult to design systems that cope with realistic "expanded" alternatives. By "expanded" we mean alternatives that contain sufficient information to really capture the essence of the decision makers' problem.

Many theoretical models exist for the decision maker's preferences [5,6], but it is still very difficult to model the choice and especially the evaluation of complex multi-attribute scenarios. We think that, during the evaluation phases, cooperation between the decision maker and the system is mandatory [1]. Consequently developers should focus, at this stage, on the interactivity rather than on the sophistication of the choice model.

The question of the organizational support is discussed, in more or less similar terms, in the literature from Management Information Systems (MIS) to Executive Information System (EIS) via DSSs. We will merely stress some particular points related to MCDSSs.

The paper is organized as follows. In Section 2, we stress that three of the four decision phases are generally poorly supported by MCDSSs especially as concerns the alternatives. In Section 3, we introduce briefly the notions of cooperation, explanation and context before surveying, in Section 4, the main organizational lessons learnt from MCDSS experiences.

2. Supporting the whole decision process in multicriteria DSSs

According to Simon's decision phases [7], it should be possible to imagine MCDSSs adapted to support the decision maker during each of the four phases of the decision process, namely, intelligence, design, choice and review. However, many designers have only addressed the choice phase, because, as we will show it is easier to deal with than the other issues, due to difficulties pointed out...
in the introduction and hereafter. The same difficulties are present in DSS development and are often obstacles for the organizational acceptance of the systems.

2.1. Intelligence

"Intelligence" refers to the information process, essential in any decision support system because, as Simon said, information constrains decision. Information is collected and stored in the database and the database management system in any DSS [8]. The functions related to the databases (recording, retrieval, structuration, etc.) are not specific to MCDSSs. Consequently, we will not devote a special subsection to this theme. We refer the reader to the general literature on DSSs or Executive Information Systems (EISs) and more recently, data mining. However, let us add that we are not aware of a single MCDSS noticeable from this point of view. This is a fundamental weakness which unfortunately, as we will see, prevents MCDSSs from addressing the fourth phase (review). The intelligence phase is also where the problem is identified, classified and structured. This problem emergence phase, in the sense of Checkland [9] or Landry et al. [10], is generally not addressed by DSS or by MCDSS designers, except in some very rare attempts (e.g., see [11]). Insufficient database(s) is a basic deterrent to the use of DSSs.

About the design phase, there are two specific targets for the MCDSS designers: the alternatives and the criteria. Let us begin with the criteria.

2.2. Criteria

The first task consists in defining a family of criteria. As explained in Bouyssou [12], (see also [13] and [5]), it is often difficult to define criteria. For example, how can we measure the damage caused to the people living near an airport [12] or the robustness of a timetable in case of an incident [14]? As far as we know, the process of specifying the criteria remains a human one, mainly relying on the discussion between people concerned with the problem. The second question is that of the consistency of a family of criteria [13]. Although it seems possible to bring some support to the DM for this particular task, the notion is probably too recent to have been already taken into account in MCDSSs. It is worth noticing that in group decision making, the definition of a criterion is generally conflictual. In this context, the building of criteria may be reinterpreted as a negotiation process [5]. One of the advantages, from an organizational point of view, of regarding discussion about criteria as a negotiation is that it provides a structure and guidelines for discussion, separating people from the issues [15].

Once everybody agrees on a family of criteria, assuming that the alternatives are known, it remains to complete the decision matrix, i.e. to evaluate each alternative according to the criteria. This evaluation theoretically depends on a posterior aggregation procedure, but this fact is generally ignored by designers so that the assessment is generally independent of the aggregation procedure. The system can support a direct assessment method, showing (graphically) to the decision maker, the position of the various alternatives or transforming a pairwise comparison into a numerical (normalized) scale as, for example, in AHP [16].

In the framework of multiattribute utility, the utilities of a given alternative, as regards each attribute, are jointly cardinal. They have consequently to be jointly evaluated (e.g., see [5]). In this case, due to the difficulty of either verifying probabilistic independance or jointly evaluating the alternatives (by solvability or by the mid-preference point method), the support of a MCDSS should be very useful.

Another aspect of the problem is the distinction between facts (objectively described alternatives) and criteria "decision" variables. One may suppose that the transition from the facts to the evaluation of the alternatives follows formalized rules, in this case an expert system can be used to make the bridge between the facts and the criteria, each criterion being defined by a rule base [17].

The distinction between objectives and alternatives raises another point. Keeney [18] has emphasized that the decision maker should define his objectives before seeking for the alternatives that permit to attain his objectives. In our framework, we can think about objectives as aspiration levels defined for each criterion or alternately as very general (strategic) goals. For example, let us think about the general goal, for a producer, of improving the quality of his products. To attain his goal the producer has to define both the criteria (which indicate how to measure the consumers’ satisfaction) and the alternatives (which describe the different variants of the product he is able to produce). It is certain, although not generally acknowledged, that the decision makers reason both forward from the diagnosed situation and backward from the goals [19].

We cannot leave the topic of criteria without discussing the assessment of weights or more generally importance
The choice among many possible variants of a future event is oversimplified. This unfortunately is the other words, the behavior of the alternative, as regards the real alternative actually is a scenario generated by the original alternative (the different variants of the product in our example) [23]. In [14], the term of fully expanded alternative (FEA) is coined to describe a realistic alternative developed from a simple original alternative.

(2) The choice among many possible variants of a future law is another example [8]. It is clear that a multiattribute evaluation of such variants of a law depends on the reaction of several actors (companies, unions, stock holders, and so on). It follows that, to be realistic, it is necessary to "expand" the different variants of the law foreseeing, the consequences of each variant which in turn depends on the actors considered. It is absolutely impossible to develop such complex scenarios without computer support and without using numerous models (including game theory and psychology).

(3) The robustness of the railway timetable in case of disruption will be our third example. Briefly, a railway network being given, the problem is to compare two timetables as regards the resistance to disruption. This resistance to incidents is evaluated from the point of view of the travellers [14]. An original alternative is a couple (railway network, timetable), a fully expanded alternative describes the behavior of the timetable when disruptions occur. This behavior depending, among other things, on the dispatchers' decisions, we have developed a MCDSS to simulate these decisions. This MCDSS produces the timetables resulting from the disruptions (FEA).

It is clear that to transform original alternatives into fully expanded alternatives, many types of modeling can be called upon, for instance game theory, expert systems, negotiation models and so on. Accordingly, the MCDSSs become multi-structural, in the sense that they are based on many types of models [8]. Most of the specific MCDSSs (specific denotes a system seeking to attain a given real organizational objective) are in fact multi-structural. Without doubt, the main effort in introducing MCDSSs in organizations is in the design of fully expanded alternatives using the resources of various models in multi-structural MCDSSs. (The reader is referred to [24] for a survey of 'intelligent' MCDSSs.) In an operational setting, the lack of realism of the modeled alternatives is one of the main weaknesses of DSSs. The weakness is usually in the future components of the alternatives. In other words, the behavior of the alternative, as regards the future events, is oversimplified. Unfortunately, it is the component that is most important in any decision.

2.4. Choice

We believe that when the decision at stake really matters, the decision maker will want to make the decision himself. Thus, as we have said in the introduction, at this stage, the designers' problem is to focus on human-computer interaction in order to support the decision maker not to replace him. Up to now, too many MCDSSs have been designed to produce, via an
aggregation procedure, a choice. Even, if the statement appears debatable to many people (among them, two of three referees of the paper), we claim that the researchers' efforts to produce aggregation models are sufficient (already more than fifty methods exist [5]), and that it would be more valuable to think about criteria, fully expanded alternatives and scenarios.

This does not mean that a MCDSS cannot be useful during the decision phase for instance, in deleting inefficient decisions, pointing out good compromise solutions or facilitating the discussion in a group decision setting. But, we think that the main decision maker's requirement are : 1) a real support for managing the criteria and the alternatives during the look-ahead phase ; 2) an interactive device facilitating the choice via an exploration procedure such as PRIAM [25]. Our guess is that the first point is much more urgent than the second and that neither involves choice models (aggregation).

Finally, let us mention that during the choice phase the decision maker often wishes to go back to the previous phases either because he wants more information or because he wishes to introduce new alternatives (e.g. mixed or compromise alternatives) or criteria. The decision process is by no means linear and this requires great flexibility in the system.

2.5. Review phase and organizational learning

Although it is not clearly stated in Simon [7], we think that one of the main functions of "review" is learning, and we believe that the best support that could be provided to organizations would be for learning. In many cases, we have observed that decision is treated as a one shot game whereas most decisions are more or less repetitive. Human memory has some known biases and, for that reason, cannot accurately analysis decisions ex post. Thus, it is illusory to think we can achieve learning in complex decisions without the help of a computer and a DSS. However, very little seems to have been done in this domain up to now.

There are many possibilities related to learning, review and ex post analysis. First, in some sense, a decision maker can learn the effect of the assignment he has given to the weights. Similary, in outranking methods, the decision maker can learn to modify concordance and discordance factors [26, 27].

In interactive procedures based on aspiration levels, whose prototype are given by DIDASS in continuous programming [28] and PRIAM in a discrete setting [25], the decision maker certainly learns to behave with conflicting goals and he gains facility with tradeoffs. Another idea is to aid the decision maker to recognize certain patterns of decision. Case-based reasoning is a way to approach this question [29]. As a part of the sensitivity analysis, urging a decision maker to enlarge his heuristic search by showing new generated alternatives is very rarely implemented. This is a kind of learning about new compromises different from those generally envisioned in the literature.

Different from learning, another issue is the question of robustness. Some packages are designed to allow a kind of learning about the effect of weights or other factors. However, this kind of learning by means of sensitivity analysis ("what if") is probably not the most important capability for decision makers. They generally don't worry about weights to the extent that they ignore theoretical problems associated with the weights. (Of course, decision makers are wrong in neglecting difficulties raised by additivity, but so it is.) Conversely, when a solution is recommended by a MCDSS, they are interested in knowing whether this solution will resist environmental modifications. It is even more important that, very often, when people involved in a group decision making have reached agreement on a solution they are, on the one hand, generally reluctant to change but, on the other hand, they would like to know the robustness of the solution to environmental changes. Let us call this notion the robustness of the solution (we have previously given an example of robustness of the timetables).

As a whole, the above question is very difficult because environmental changes include changes in alternatives and consequently in the fully expanded alternatives (new alternatives, new evaluation) and on the importance of the criteria. But we believe that generally changes related to the alternatives are more consequential than changes in criteria so that, robustness relative to alternative modifications is at the center of the robustness question. One goal is to design a system in which any change at any level in an alternative is promptly propagated to the corresponding fully expanded alternative. Very little research has been done on this topic.

This question of robustness introduces our last point about reviewing. To learn about decision making it is most important to record the exact conditions prevailing when a decision was made. The main items are : what were our ideas about the possible alternatives (why ?), what was our evaluation of the alternatives, what were our probabilities about future events, our expectations, our perception of the environment ? It is
clear that the previous questions cannot be answered without an appropriate MCDSS which supports the decision maker and records data describing the context of a decision. Designing such a MCDSS is the price to pay for learning and for making serious comparisons to discover the possible flaws which have endangered past decisions. It is also the only way to accumulate reliable data to shed some light on the future decisions. To our knowledge, some EISs have marginally attacked some of these questions but no MCDSS has. This kind of learning seems still to be an empty field has open to MCDSSs.

3. Cooperation, explanation and context

3.1. Cooperation

Cooperative systems are based on a successful combination of human skills and computing power in carrying out a task which cannot be done by the human or the computer alone [30]. The goal is to construct systems that augment and amplify human intelligence in problem solving, decision making and information management rather than just replacing it. Such systems must be active agents working in real-time and in synchronization with users.

Various elements are evoked to specify cooperation modes: decomposition and dynamic allocation of tasks, communication, coordination, collaboration, negotiation, conflict management, initiative, control, dealing with alternative solutions, etc. Any combination of these elements leads to a cooperation model, generally depending on the nature of the application.

Moreover, a partial result at a stage of problem solving may imply, for instance, a change of the reasoning process, implying a new course of cooperation with the system. Such a sequence of partial resolution and successive decision maker interventions leads to a progressive elicitation of the decision maker's preferences (interactive aggregation procedure) and to a choice between various aggregation procedures adapted to the problem, followed by an interactive refinement of the solution satisfying (or implying a change of) the constraints managed by the decision maker.

Although not often acknowledged, two aspects of cooperation are important for high communication quality, namely, the generation of explanations and the modeling of cooperation context. The machine and the user may have to explain things to each other. The role of an explanation is to persuade the other that a partial solution is correct, not just to provide a final justification of an already determined completed solution [31]. During the interaction, the transfer of information between the human and the system has a highly contextual nature [32]. Representing and using context leads to optimizing the communicative acts (i.e. reducing the needs of such acts) and minimizing ambiguities in the discourse.

3.2. Explanation

Explanation aims to 1) modify the knowledge of the other agent (bringing new pieces of knowledge or revising existing ones); 2) to clarify the context and make explicit the part of the knowledge that is shared; 3) to convey new information pieces unknown by the explane; 4) to reach an agreement; 5) to coordinate the agents' activities; 6) to remember our own implicit assumptions, forcing us to trace our own reasoning process and suggesting alternative ways of thinking about a problem.

As an example, in multicriteria interactive procedures, it is very important that the system explains to the DM what is gained on one criteria may be lost on another. For instance, the decision maker has to learn the notions of tradeoff between criteria and to learn that there is not a unique efficient point and that moreover the efficient points are not equivalent.

In a cooperation process, each step of the decision process must be accepted by each agent. If a misunderstanding occurs, then each agent needs to negotiate his point of view, explain and justify. For this, each agent is led to provide the other with additional information that is not necessarily useful in the decision process, but that will be afterwards shared and accepted by both agents. For instance, it may be necessary to point out some changes in the environment implying changes in the alternatives, to recall some relationships between alternatives, and to remember previous choices leading to a failure. Thus, there is a transfer of knowledge between the system and the users. Explanations convey contextual information that are missing in the explane's context [31]. When two agents disagree, each one may offer explanations to try and get the other to change his beliefs, so that an agreement may be achieved. In such case, explanations aim to adjust both agents' contexts to reach compatible interpretations. To facilitate the interaction, the contextual knowledge must be shared.

On the one hand, studies of human-human dialogues show that explanations represent a part of their discussions, most of them being uttered spontaneously without questions [31]. On the other hand, up to now, human-machine interaction has not been, as far as we know, considered as a process of explanation exchange. In fact the explanation of explanations should be understood from the system to the user, and from the user to the system. The former has been largely studied in early knowledge-based systems. The latter is something new. This supposes that the system may learn from its experiences with users in order to be wiser later. Thus, it ought
incrementally acquire knowledge. This implies a revision of the design and the development of MCDSSs as, indeed, of more cooperative DSSs. This is yet a challenge, even for knowledge-based systems.

3.3. Context

The cooperation context contains items like the dialogue memory, the task at hand, the spatio-temporal situation (place, moment, elements, ...), and the psycho-social situation (user model, roles and social status, ...). It is mostly important to capture the context in which events occur and data are obtained because any change in the context may entail some changes in the direction taken by the decision making process. (See the discussion about different types of contexts, in [33].)

Cooperation, explanation and context intervene at various levels of the decision making process: the choice of the criteria and their possible weighting, the choice among various aggregation procedures, the design of the alternatives and their variants, the recognition of certain patterns of multicriteria decision, the progressive elicitation of the DM's preferences, and a synthetic anticipation of the behavior of people implicated in the process.

Though, it is still a difficult question, the aggregation process must result from cooperation between the decision maker and the system. Explanations are useful to facilitate the interactive building of an aggregation. We have already mentioned that alternatives are either scenarios or expanded alternatives. At this level, explanations are very useful because it is very difficult to build a realistic scenario and to follow the many interactions between the decision makers' actions and the responses of the environment. Dealing with expectations (probabilities or whatever else) is often very counter-intuitive and a rich dialogue is necessary to avoid the natural lack of decision maker's confidence facing 'not understandable' computations. The explicit consideration of expanded alternatives is a first step to increase contextual knowledge.

Structuring the knowledge base by means of contextual data is necessary in order to avoid intractable search during the decision making, focusing on wrong features of a situation and of acting inappropriately. The context of the alternatives and the environmental data are particularly important in the decision making process. Events occurring in nature may affect the elaboration of the alternatives, especially fully expanded alternatives and the pertinence of the criteria. In some sense, in the decision theory framework, the states of nature contain part of this contextual information. Making this context as explicit as possible would improve the whole process.

Contextual data are crucial for a designer of alternatives at two levels. First, possible actions depends on expectations which are generally not completely stated in the model. They depend also on strength of preference. Some actions may seem impossible because people have implicit expectations about reactions of other agents or of nature. This context should be modeled and shared with the system. Otherwise, the decision maker will reject the system as too naive.

Second, the context has an influence on expectations. Whatever method is used to model the future (probabilities, possibilities, etc.), the evaluation of the future is very subjective and context-sensitive. According to the knowledge and the mood of the subject, the evaluation may be very different. If a system could capture a part of this information about the context of the forecasts, a richer dialogue would be possible between the decision maker and the system. Introducing more contextual information in the system may result in explaining and correcting biases that are very common at this step.

4. Organizational lessons

We do not consider in this section the issues related to DSS introduction within organizations treated in a well known literature going back to Keen and Scott Morton [34] updated in Ware [35] for knowledge-based systems. We would like to draw some more specific conclusion from our MCDSS experience.

The first difficulty arises with data. A realistic system should contain a huge amount of data. It is implausible to require that the decision maker enter these data into the computer. This raises the question of automatic gathering, updating and adding data. These problems have not been solved except within small areas. We are (re)discovering with the Web data mining and fashions that there is at root of any popular system an efficient database system. Many DSSs emphasized their model components forgetting that the models without the data are worthless. EISs adopting the converse philosophy are, not surprisingly, more used.

The second point, is that, as shown by problems with alternatives, we need better modeling of the future. What really bothers decision makers is the development of an alternative's future. There is not simple alternatives, but rather scenarios or what we have called "fully expanded alternatives". We urgently need better tools and perhaps models for supporting decision makers understanding of future events, competition, reactions, etc...

The third lesson, is that it is almost impossible to capture the preferences of a decision maker, as shown by the quasi-impossibility of a consistent aggregation of
multiple criteria. One ought to admit that the only way to approach decision makers’ preferences is to design interactive DSSs performing a progressive elicitation of preferences. Especially when the decision at stake is important, it is absolutely non-sensical to try to design a DSS to make the decision: support the decision maker, don't replace him. This is known for a long time [34] but is often forgotten. What is less obvious is now to design a system in which the decision maker understands and accepts a progressive elicitation of his preferences. We think that an explanation-based module is a good way to tackle the problem.

Finally, perhaps the principal weakness of our systems, from an organizational point of view, is the lack of review and learning procedures. We cannot separate these two issues which encompass the main difficulties of DSS design. In fact, in order to understand whether a decision was a good decision, you have to remember the contextual data prevalent at the decision time, which supposes a good explicitation of the context, you have to consider the prior probabilities and the events you were aware of at the time of the decision and you need to understand what your preferences were and why. In order to cope with the three previous issues raised by the design of DSSs that learn, we have to think about the main difficulties in the development of specific DSSs.

We summarize in Table 1 our views about the present state of the art in MCDSSs. We are convinced that by providing a framework for sorting out the difficulties involved in criteria, alternatives, evaluation and weighting, the MCDSS setting raises some issues that are more or less identical for any DSS.

<table>
<thead>
<tr>
<th>Poorly supported</th>
<th>Partially supported</th>
<th>Well supported</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intelligence</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Information and data management</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Problem identification and structuring</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Design</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alternatives and scenarios</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Criteria and weighting</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Evaluation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Choice</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interactive choice</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Automatic choice</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Discussion negotiation</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Aggregation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Review</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Recording</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Learning</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

Table 1 MCDSS problems and achievements

5. Conclusion

Whereas MCDSSs form only a subset of DSSs it is interesting to observe that the multi-attribute dimension points to some key issues of DSS design. MCDSS developers have generally designed multicriteria shells containing only general multicriteria knowledge (analogous to expert system shells). Unfortunately, these systems cannot resolve real problems per se and provide no help on the crucial point of coping with realistic alternatives. The future of MCDSSs lies in multi-structural systems able to aid the decision makers in building what we have called, fully expanded alternatives. More complexity and flexibility are required in the processing of alternatives which, we should never forget, are all possible solutions of the decision problem. And it is the solutions, not the criteria, which the decision maker is interested in. For the same reason, the notion of robustness becomes a prominent key to enabling MCDSSs to deal with the concerns of flesh and blood decision makers. These objectives will not be met without introducing contextual information and learning capabilities into the systems. Moreover one cannot forget that the preference owner remains the decision maker which implies that interactivity and progressive elicitation...
of the preferences via dialogues phases and explanations is still a crucial issue.

6. References


