



IFAC/IFORS/IIASA/TIMS

The International Federation of Automatic Control
The International Federation of Operational Research Societies
The International Institute for Applied Systems Analysis
The Institute of Management Sciences

SUPPORT SYSTEMS FOR DECISION AND NEGOTIATION PROCESSES

Preprints of the IFAC/IFORS/IIASA/TIMS Workshop

Warsaw, Poland

June 24-26, 1992

Editors:

Roman Kulikowski

Zbigniew Nahorski

Jan W. Owsiniński

Andrzej Straszak

Systems Research Institute
Polish Academy of Sciences
Warsaw, Poland

VOLUME 1:

Names of first authors: A-K

DECISIONMAKER: LESSONS FROM THE MARKETPLACE

Niall M. Fraser
Department of Management Sciences
University of Waterloo
Waterloo, Ontario,
N2L 3G1 Canada

Abstract: There is a need for negotiation support systems developers to have feedback from system users in order to improve the design of future systems. In this paper, experience gained from working with the DecisionMaker program for negotiation support is reported. Eight "lessons" are presented, and reinforced by reports about specific cases.

Keywords: negotiations, decision support systems, negotiation support systems, conflict analysis, marketing.

1. Introduction

The use of formal methodologies to support decision and negotiation processes is a relatively new idea. Although theoretical work is being done in increasing amounts, little practical experience has been reported. In effect, the support system development process is an open loop system, with no feedback from users to affirm or refute the claims of system developers. The purpose of this paper is to close that loop, in an informal and anecdotal fashion, to the extent permitted by experience with a single computer program.

DecisionMaker: the Conflict Analysis Program (Waterloo Engineering Software, 1991) is a decision support tool for use in situations where there are multiple participants, including negotiations as well as more competitive relationships. *DecisionMaker* has been distributed commercially for about five years. During this period users have been observed in order to guide development of the system. The process was casual; the results presented in this paper are a set of guidelines supported by examples and anecdotes. However, the ideas should be applicable to many decision and negotiation support systems.

The title of this paper is intended to have two meanings. The first meaning is the obvious one; presented are lessons about user needs in decision and negotiation support systems as learned through the marketing of the *DecisionMaker* computer program. The second meaning relates to the location where so many negotiations take place. The kind of haggling that goes on in a village produce market is not really much different than the negotiations that occur anywhere in society. In addition to paying attention to feedback from users, it is important that system developers observe how people do behave, as well as how they should behave.

The advice garnered from the marketing experience with *DecisionMaker* is presented here as a set of key statements, or lessons. Each is supported by reference to particular experiences, often in the form of an anecdote. A brief description of the *DecisionMaker* program is necessary before the lessons are presented.

2. *DecisionMaker*: The Conflict Analysis Program

DecisionMaker is an implementation of the *conflict analysis* methodology introduced by Fraser and Hipel (1978, 1984). It helps to organize information about a complex problem involving more than one participant so that better decisions can be made. It runs on personal computers under the Windows presentation manager.

The conflict analysis methodology requires as input a formal model of the problem. The model has three components: decision makers, options and preferences. The decision makers are the parties involved, such as individuals, companies or countries. Options are actions that each decision maker can take or not take, such as cut prices or withdraw from negotiations. Preferences refer to the importance and desirability for each player of various possible outcomes of the negotiations.

Such a model is quite easy to elicit from a person knowledgeable about a particular negotiation. It corresponds well to how humans think about complex strategic interactions. It is noteworthy that no weights or other quantitative measures of preference are required. The approach is ordinal in nature; that is, it requires only the ranking of preference rather than some numerical measure.

From the structure of the model, the analysis procedure identifies logical relationships that may not be obvious to the user. This is the heart of the approach: an algorithm that mimics the analytic process a human uses to assess all the interactions among the decision makers. Nothing is added or subtracted; the known information is simply presented in a different way.

One of the main features of the output of the procedure is information about stability of possible outcomes. An outcome in this context is some combination of options taken by the decision makers, rather than a distribution of costs and benefits. An outcome can be unstable for a decision maker in the sense that should it occur, the decision maker has good reason to change the options under its control to bring about another outcome. Outcomes that are stable for all decision makers are particularly interesting, because if they come about they would likely persist. Such "equilibrium" outcomes indicate the actions that would eventually be taken by the decision makers.

All of the calculations necessary for the conflict analysis procedure can be done with pencil and paper, but of course it is more convenient to use a computer program. *DecisionMaker* provides a convenient and quick way to authoritatively assess a conflict model.

3. Lessons

Lesson 1: The fact that a system is useful doesn't mean that it is meaningful.

A large manufacturing company, A, had excess production equipment. Rather than scrapping or selling the equipment, they loaned it to another firm, B, under the arrangement that the second firm would return a percentage of the profit earned from the equipment to the first company. The second company was successful, but an argument arose about the fair royalties to be paid.

Three senior managers from company A attended a session with a consultant who was to use *DecisionMaker* in the planning of the negotiations with company B. However, during the discussions that would lead to the development of the conflict model, the managers discovered that they did not agree on many aspects of the problem. They perceived the

options of the decision makers differently, and they perceived the preferences of the company differently. Through the process of trying to develop the model, they realized that the problem was really that they were not communicating among themselves properly. Once they agreed on what the problem was, the solution to it was clear to them without further work.

The simple existence of the methodology was sufficient to meet the clients needs. The attempt to use conflict analysis focused the three managers on the problem, and forced them to communicate about it. The fact that they found the process useful provided no support for the particular methodological approach at all.

Lesson 2: You can alienate half of your market, or all of it.

Based on experience with DecisionMaker, clients and users tend to fall into one of two categories: those who focus on the answer, and those who focus on the process. People who focus on the answer don't care to much how the answer is arrived at.- they simply want to get over the hurdle provided by the problem. They want to have an answer that can be justified in some manner, but have better things to do than understand all of the details of the process. Many managers fall into this category of user, but not all.

On the other hand some users want to know all the details about how a particular result is determined. Given an answer without the exact reasoning behind it, these users will not trust it. Analysts and technicians tend to be this type of user, but not all.

The problem with designing a decision and negotiations system is that the "managers" and the "analysts" cannot be simultaneously satisfied. If the system is transparent so that the details of the process are apparent, the "managers" will be impatient and want a clear answer immediately. If the results are presented through an invisible mechanism, the "analysts" will reject it as unfounded. A system that gives partial information will be rejected by both groups.

There are strong feelings among users about other issues too. Whatever the issue is, a significant part of the potential users for the system will be alienated no matter what is done. The only way to deal with this problem is to recognise a market niche, and design for it. Aim

at either the "manager" or the "analyst" and ensure that half (loosely speaking) of the market is satisfied, rather than trying to meet the needs of both simultaneously, and pleasing neither.

Lesson 3: Details often don't matter.

Decision makers tend to be an impatient lot. They often have a great deal of information in their heads about a particular problem, and expect a consultant or computer program to read their minds somehow, rather than taking the time to properly communicate their knowledge.

As it turns out, this is often not a problem. It seems that frequently the underlying structure of an issue leads to particular results independently of the details. Of course this may not be true in all cases, but it has been a good rule of thumb with the DecisionMaker program. In dealing with a client it is useful to suggest, at an appropriate opportunity, that the procedure be continued even though all the details have not been yet specified. It can be expressed as an instructional or experimental completion of the analysis, with the understanding that the details of the model can be completed later. Almost always such a return to complete the details is forgotten about. See Lesson 7 for a related idea having to do with how users will often make models more complicated than necessary.

Lesson 4: There is no such thing as 'true' preferences.

Any decision or negotiation support system must elicit from the client or user a measure of the desirability of one thing over another. This can often be the most challenging part of the process, because human preferences are sometimes so hard to pin down. People will be inconsistent. They can exhibit paradoxical behaviour, and they can unfortunately express their preferences in ways that are particularly difficult to model mathematically.

A conventional view of preferences seems to be that a human has, somewhere deep inside, preferences that are mathematically convenient, rational, transitive and consistent. However, man's imperfect nature sometimes makes it difficult for him or her to be aware of or to communicate these preferences. Under this view, it is the analyst's or support system's role to assist the decision maker to recognise these "true" preferences.

An alternative view is that a human does not know his preferences until prompted, and then can only focus on one aspect at a time. Moreover, his preferences are subject to positive feedback - once expressed, they are reinforced so that they become more true.

Experience with DecisionMaker suggests that this second view is more accurate. The preference notation for the program (called *preference trees* [Fraser, 1991]) has been designed to take advantage of the approach by focusing on an ordinal, lexicographic scheme.

A manufacturer in Canada was recently faced with two difficult problems. A free trade agreement between Canada and the United States had just been completed, which would open his Canadian markets up to competition from the US where labor costs are cheaper. Also, the unionized workers at his plant were threatening strong action at upcoming contract negotiations.

Several sessions with DecisionMaker reveal that severe labor troubles could only be avoided if the company was likely to move its production to the US in response. However, the company president was adamant about retaining the Canadian facility.

It was pointed out to him that if in fact the Union believed that he was willing to close the Canadian plant, the labor problems would be avoided. In other words, if he was able to deceive the Union about his true preferences, strife-free production in Canada could be continued. However, he rejected this as being too difficult to pull off.

What he did instead was to in fact change his preferences. The close look at his two problems through the DecisionMaker exercise made him realize that if extreme demands were put forth by the Union, it probably would be better to transfer production south. Of course this then achieved the desirable situation for inducing beneficial behaviour by the Union, and the company ended up staying in Canada.

Lesson 5: Satisfying the user is more important than satisfying the developer.

Company ABC had a marketing contract with inventor D. Inventor D wasn't satisfied with ABC's performance and thought that company EFG would do better. There were also many other complications to the three party relationship.

Company ABC approached a consultant for help. Using the DecisionMaker program, the consultant was able to show that ABC really had two choices. One was to renegotiate with D for their mutual benefit. The other was to end up in court with everybody suing everybody. Further, it was clear that ABC had the capability to determine which of these outcomes ensued, and the renegotiation outcome was clearly preferred by ABC (as well as by D and EFG) over the court battle.

The consultant was pleased to have made the problem so clear to the client, and expected a protracted but successful negotiation among the three parties. However, he was shocked to find that his client almost immediately proceeded with legal action against both D and EFG. On inquiry, the consultant found that the client had made a token request to D for renegotiation, and had been turned down. ABC then felt that if the only other alternative was the courts, then they should be proactive rather than reactive. Moreover, ABC was extremely pleased with the help given by the consultant. In their view, they were able to make a timely and clear decision based on the process they had undergone.

It was apparent to the consultant that ABC had not been forthright about its preferences. It actually preferred to take the issue to the courts rather than negotiate, perhaps because it did not like the loss of face associated with renegotiating with D. Again, ABC was not trying mask this feeling, but its preferences were simply complex enough that they were not completely revealed in the modelling process.

The key though was that ABC was able to make a definite decision based on the modelling process, and further was able to credit the formal methodology for allowing the decision to be made. The consultant, and ultimately the support system designer, would be wiser to recognise the contribution of the support system in such cases rather than viewing the process as a failure because the ostensibly correct outcome was not achieved. What is important is that client the both used and was satisfied with the process. Note that this is different from the situation in Lesson 1, in that here the client actually did make use of the results, rather than simply solve his own problem independently of the methodology

Lesson 6: Things that aren't important sometimes are.

The DecisionMaker program permits the modelling and analysis of complex problems involving multiple participants. Under normal circumstances, it is assumed that each of the decision makers in the model has complete information. An alternative is to recognise that in some circumstances a decision maker may have some sort of misperception. He could simply not have the correct view of some aspect of the problem, or he could have been intentionally misled by one of his opponents. Such situations have been called *hypergames* (Bennett, 1977) and the methodology for dealing with them has been well developed (Wang et al., 1988).

It turns out that the benefits of modelling situations as hypergames are not worth the disbenefits. A great deal of work is sometimes required to develop a hypergame model, and often the logic of who knows what is difficult to deal with. In practice, instead of using hypergames one tends to use simpler models and focus on a narrower aspect of the problem.

However, with DecisionMaker, in general once users are appraised of the concepts of hypergames, they always want to be able to analyze hypergames using the computer. They insist on the software being designed to handle hypergames. They may never use the capability, and the program designer knows that it is not an important capability, but the users demand it.

A similar principle applies to the "user interface". The fashion currently is to have applications programs follow the Apple MacIntosh presentation style. DecisionMaker runs under Windows 3, for example, which is the MacIntosh style presentation manager for DOS computers. Obviously not all computer programs fit into this type of user interface well. However, people don't want to learn a new way of dealing with the computer, even if it is better for that particular application.

Lesson 7: Things are often both more complex, and simpler, than they appear.

DecisionMaker concentrates on problems where there are more than one participant. The mathematics of game theory, which deals with such problems, recognises that the smallest

model possible involves two player, each with a choice between two courses of action. Such a model is called an 2x2 game.

One would think that, as the smallest possible model, the 2x2 game would be straightforward to deal with. The contrary is true however. The relationship between the players in a 2x2 game can be extremely complicated. A single example of a 2x2 game called "Prisoners' Dilemma" has had probably thousands of articles written about it, including at least two books (Rapoport and Chammah, 1965; Axelrod, 1984). So, particularly when multiple participants are involved, things can be more complex than they seem.

For an associated reason, things can be simpler than they seem too. Because an apparently simple problem (such as one that could be modelled as a Prisoners' Dilemma) can be mind-boggling, people will tend to make it more complicated than it really is, for example by adding extraneous players or actions. This is because they know the problem is complicated since they have tried to deal with it and have not been able to. However, if they express it as a model involving (say) two players each with two strategies, it just looks too simple. Not that clients or users are aware of this process - its is a consequence of trying to communicate the inherent complexity of the problem without properly recognising the source of the complexity.

The result is that models tend to appear larger and more complicated than they really are. This is especially true when a computer program is being used because the client expects the computer to be able to handle complicated things.

There are two ways of dealing with this problem. One is to guide and educate the client or user as much as possible. The personalities of some clients can limit this approach, however. Another method is to design the decision or negotiation support system in such a manner that the extraneous information is culled as a natural part of the modelling process. This second approach is implemented in DecisionMaker at several point in the program.

Lesson 8: Hostility by users is present, but can be overcome.

Two mining companies, A and B were merging. Each company previously got their explosives from separate suppliers. The Client supplied company A and the Competitor

supplied company B. The new president of the merged firm, AB, was previously the president of company B. The Client wanted to obtain firm AB's business. The options for the Client included lowering prices, a product demonstration and special delivery terms. The options for AB included accepting either supplier or splitting their orders, and requiring various forms of special treatment. The Competitor could also lower prices and offer special terms.

A manager in the Client company had been studying this problem for a month, and was very sceptical about the use of DecisionMaker to help him. His hostility was very apparent, and he begrudged the time spent with the consultant. However, an analysis revealed that there was a stable outcome that the Client manager had overlooked which was quite beneficial. It was a very dramatic moment because the manager literally hit himself on the head and rushed out of the room to implement the solution immediately when he realized the practicality of this insight.

Some people do resent the idea that a computer program can improve on their skills. For quantitative problems, there is a recognition that the computer is an appropriate tool, but for broad decision making and negotiation, there is a resistance to accepting it. This is especially true when such a tool is being used to support a dedicated problem solver, rather than an occasional one.

However, a formal approach like conflict analysis can, in many circumstances, do better than an unsupported human. In cases like this example, it will not overlook important consequences of a situation that can be missed by a human because of the innate complexity of a strategic relationship.

An analogy can be made with the dissemination of linear programming (and other methods of operations research) in business. Linear programming is a method for deciding, for example, what quantities of different products should be manufactured. Beyond 40 years ago, such decisions were made by experienced managers, many of whom were unable to accept computer decision making. However, such decisions are now made by technicians using a computer. The change was an evolution as companies who used linear programming simply made better decisions than companies who did not, and were thus, in the long run, more productive and profitable.

Not all forms of hostility can be solved as quickly as in the case of the mining companies. However, if decision or negotiation support tools do offer value, and help better decisions to be made, they will eventually be accepted. This applies to both the field in general, and any particular support system that is developed.

4. Conclusions

As a new field, decision and negotiation support systems have many hurdles to overcome before they are widely accepted. This is a natural process that will take time and will have many successes and failures. It is important to realize that the process has only begun - the systems developed now are only the foundations on which the widely accepted systems of the future will be based.

In order to foster this development, this paper has presented some ideas based on experiences with using and distributing the DecisionMaker computer program. In summary, the lessons are:

Lesson 1: The fact that a system is useful doesn't mean that it is meaningful.

Lesson 2: You can alienate half of your market, or all of it.

Lesson 3: Details often don't matter.

Lesson 4: There is no such thing as 'true' preferences.

Lesson 5: Satisfying the user is more important than satisfying the developer.

Lesson 6: Things that aren't important sometimes are.

Lesson 7: Things are often both more complex, and simpler, than they appear.

Lesson 8: Hostility by users is present, but can be overcome.

These lessons are not intended to be presented as immutable rules, but more as points to consider.

5. References

Bennett, P.G. (1977). Towards a theory of hypergames, *OMEGA* 5, no. 6, 749-751.

Axelrod, R. (1984) *The Evolution of Cooperation*, Basic Books, New York.

Fraser, N.M. (1991a) Ordinal preference representations, unpublished paper, Department of Management Sciences, University of Waterloo, Waterloo, Ontario Canada.

Fraser, N.M. (1991b) *The Power of the Game*, unpublished manuscript, Department of Management Sciences, University of Waterloo, Waterloo, Ontario, Canada.

Fraser, N.M. and Hipel, K.W. (1979). Solving complex conflicts, *IEEE Transactions on Systems, Man and Cybernetics* 9, 805-815.

Fraser, N.M. and Hipel, K.W. (1984). *Conflict Analysis: Models and Resolutions*. North-Holland, New York.

Rapoport, A. and Chammah, A.M. (1965). *Prisoners' Dilemma*, University of Michigan Press. Ann Arbor.

Wang, M., Hipel, K.W. and Fraser, N.M. (1988). Modelling misperceptions in games. *Behavioral Science* 33, no. 3. 207-223.

Waterloo Engineering Software (1991) *DecisionMaker: the Conflict Analysis Program*. 22 Dupont St. E, Waterloo, Ontario, Canada N2J 2G9. (519) 885-1211.

IBS Konf. Nr.

42070

tbl. podre

I