Group Decision and Negotiation, 3:351–372 (1994)

INTERACT: Developing Software for Interactive Decisions*

PETER BENNETT
ANDREW TAIT
KIERAN MACDONAGH
Department of Management Science, Strathclyde University, 26 Richmond Street, Glasgow G1 1XH, Scotland

Abstract

This paper describes the development and use of a piece of software, INTERACT, for analyzing situations under the control of several interested parties. After briefly discussing forms of analysis used, it outlines the design philosophy of the software itself. Particular attention is paid to its intended use in a decision support role. The functions typically performed in the course of analysis are illustrated using a worked example. Finally, avenues for further development are outlined.

Key words: decision support, game theory, hypergames, conflict

1. Introduction

This paper is concerned with the analysis of situations involving several separate parties, each with a stake in what happens and some capacity to affect it. Furthermore, their aims and interests differ: while usually not in total opposition, they are trying to bring about different outcomes. Such situations provide particular challenges. Because no actor has complete control of events, each must try to take the others' possible actions into account. For any two actors, it may well be that the best course of action for each depends on what the other decides to do. Each may thus try both to anticipate and to influence the other's choices, while knowing that similar attempts may be in train "on the other side of the hill." Their *decisions* thus interact with each other. The various actors may be *taking* decisions separately: though they may come together sometimes to communicate or negotiate, in other cases they may never even meet. But they will find themselves in an outcome determined by the choices made by all.

Such situations—involving "interactive" decisions by separate parties, rather than individual or collective choices—bring forth possibilities for mutual threats, deceit, bluff, and counter-bluff. But these are still only one side of the story. Because conflict is almost never absolute, joint gains can be made by cooperating.

*An earlier version of this paper was read at the Conference on Decision Making under Conditions of Conflict, University of Waterloo, Ontario, August/Sept 1992

So interactive decisions are also to do with cooperation and "collaborative advantage" (Huxham and Macdonald 1992), and with promises as well as threats. Party for this reason, we largely avoid the term "conflict analysis" here, to avoid the impression of being concerned only with outright hostilities such as wars or strikes. Where aims partially diverge, conflict and cooperation become inseparable (Lax and Sebenius 1986).

This article describes the development of some software, called INTERACT, for modeling and analyzing such situations. As elaborated below, INTERACT is designed for users with some knowledge of conflict analysis, and can be used as an aid to research, as a teaching medium, or for decision support. After outlining the ideas on which the models in INTERACT are based, the software itself is illustrated using a worked example. This is followed by a brief discussion of the potential benefits of analysis, and finally of further developments currently being pursued.

2. Forms of analysis

The models used in INTERACT have their basis in game theory and developments from it, including megagame and hypergame analysis. The latter are introduced briefly below, while game theory as such is assumed to need no introduction here. Given its unhelpful connotations, we have found it best to avoid "game" terminology in practice. Nevertheless, the basic building blocks of analysis will be familiar enough:

- the relevant **actors** (or "players" or "parties") engaged in some specified issue, who may be individual people, or groups or organizations,
- the actions or policies open to each actor with respect to the issue, modeled either as **strategies**, or more simply as binary **options**,
- the **scenarios** or **outcomes** that can result from possible choices by the actors,
- the **preferences** each actor has for the possible scenarios, reflecting that actor's aims, interests, likes and dislikes.

Game-based models can be used in various ways. The original aim, still pursued by many, was to explore the existence of "optimal" choices for each side, a task which usually requires preferences to be quantified in terms of utilities. However, an alternative view to which we subscribe is that models are more useful as ways of exploring and understanding the structure of situations, complete with whatever paradoxes and dilemmas they may contain. Even very simple models (using no more than ordinal preferences) can highlight quite subtle points about the interaction of decisions. For example, the well-known game of "chicken" shows up genuine dilemmas of rationality. If each side chooses similarly, both choices turn out to be "wrong." A frequent tactic is to try to "win" by appearing totally unreasonable, or even unable to back down, but if both sides try this, disaster looms. In essence, the problem is that of making credible a threat that implies willingness to act against one's own preferences: "to die rather than give in." Genuine emotions are likely to be generated by the pressure of trying to make such threats believable, intermixed with more cold-blooded attempts to bluff. Similarly, agreeing to cooperate in "Prisoners' Dilemma" implies a promise to forego the advantage gained by exploiting one's partner.

Many real conflicts, from the interpersonal to the international, have elements akin to such games. This is not to suggest that there is nothing else to them. Modeling in terms of actors, options, and preferences will at best provide a partial understanding: for example, a concentration on preferences may distract attention from actors' needs (Burton 1987). Accepting that this is the case, it is still appropriate to develop the actor/option/preference framework so as to allow richer models to be built. In particular, one may typically wish to allow for the following points:

- Many specific options are often available to each side, that may or may not be independent of each other. There is thus a need to manage **combinatorial complexity**.
- Actors may have **differing perceptions** of the issue in which they are engaged. They may thus see different scenarios as possible, or misperceive one another's preferences, or even disagree as to who the relevant actors are.
- Actors are frequently engaged in a complex mess of **interlinked issues**, both between and across organisations.
- The **dynamics** of the situation are often important. Even within a fixed "game", the logic of the situation can change the sequence in which moves are made matters. More generally, the relevant players, their preferences, available options, and perceptions change over time.

A previous paper (Bennett, 1991) discussed two possible ways forward. One is to extend the scope and variety of formal models: the other, more radical, is to develop a "knowledge base" of substantive theory to guide modeling, as in an Expert System. INTERACT is a contribution to the former approach, but provides one path toward the latter.

Existing formal methods go some way toward meeting the complexities just noted. Combinatorial complexity can be addressed using the *Analysis of Options* method (Howard 1987, 1989; Radford 1980). The actions open to each side are modeled in terms of simple binary options, and the model represented in the form of a tableau, showing scenarios as combinations of options taken up or not. Models can be built up and analyzed in a series of easy steps: starting from a particular scenario, one examines whether any participant can benefit by moving away, then—if so—what sanctions others have against such a move. In this way, a set of conclusions about the stability of different scenarios is built up, a process illustrated below.

The recognition that issues may be conceptualized quite differently forms the basis of *Hypergame Analysis* (Bennett 1977; Bennett et al. 1989). Rather than supposing that the same "game" is seen by all, a different game is defined for each actor, representing that particular view of the situation. In principle, these games need have nothing in common.

Though the formal analysis of linked (and multi-level) issues is complex, they can be represented using graphical notations, such as those of hypergame *Preliminary Problem Structuring* (PPS) (Bennett et al. 1989) and *Linked Decision Situations* (Radford 1980). Such problem-structuring is not incorporated directly into formal analysis, but forms a general backdrop for analysis of specific issues.

Some dynamic factors can be allowed for using *multi-stage games* (e.g., Thomas 1987). These are a series of linked games in which the outcome reached at each stage influences which game is encountered next. Thus, one outcome of negotiations may lead into a strike, and so on. More specific suggestions about likely dynamics are provided by the theory of emotions and preference change proposed by Howard (1989).

3. Aims of the INTERACT package

The overall design of INTERACT was influenced by two main factors: an overall philosophy of modeling, and the uses and users envisaged for the software.

One attraction of having a computer package is the ability to build up and analyze models more complex than could easily be done with pen and paper. However, the aim was not to pursue complexity for its own sake. Though it may look more "realistic," there is absolutely no guarantee that a more complex model will be more useful. The more one tries to include, the more data-hungry the model becomes, and the more difficult to test. Overambitious modeling can overwhelm rather than help understanding. Rather than seeking to work with very complex models, our philosophy in developing INTERACT has therefore been *to develop software to support a flexible methodology within which alternative models can be rapidly built up, explored, and modified.* The models in use at any given point remain fairly simple, but one should be able to try adding different sorts of complexity—e.g. extra options, or differences in perception—in a flexible way. One should also be able to move at will between different stages of the modeling process, and add or modify material to any point.

The aim was also for a package that could be used in various ways—e.g., in working "for oneself" on a personal decision problem, as a teaching device, or as part of a research project—but particularly as a tool for *decision support.* That is, one should be able to use it as a medium for working *with* clients actually engaged in the issue, building, and analyzing models in real time (rather than "going away and doing some analysis"). One might well also be dealing with a client *group,* with members holding differing views about the issue in hand. Such differences have to be managed (for example, by trying out alternative models), while retaining the explicit focus of the analysis on managing the client's interactions with external parties. A final requirement was that the user need not be an expert in computing, though he or she should certainly be familiar with the ideas underlying the analysis.

Taken together, these requirements are quite demanding.[1] In summary, our need was for a package that would:

- be easy to learn and operate, given no more than very basic computer literacy. (Similarly, intermittent users should not have to re-learn the system each time.);
- allow one to display and work on any part of the model at any time, so as to make connections between different aspects of the problem, and to modify earlier work;
- provide on-screen information that would be easy for both analysts and their clients to understand. Similarly, it must be easy to explain operations performed on the model, and the causality between action and effect. Otherwise clients will lose "ownership" of the analysis and feel mystified and alienated.
- minimize the possibilities for user error. Error messages, even for trivial mistakes, are embarrassing during a decision-support session. There should be no need to remember a multiplicity of nonobvious commands.
- be portable, and require no unusual hardware.

As to the specific modeling methods used within INTERACT, our general approach has been to adapt and combine existing representations, though avoiding the use of "game" terminology. This is admittedly a conservative tactic, but has the advantage that the analyst familiar with existing methods can more easily use and appreciate the software. In the longer term, use of more advanced computer graphics should open up other possibilities.

INTERACT starts by using the Preliminary Problem Structuring notation, hitherto confined to pencil and paper, as a medium in which to build up an overall picture of the relevant issues. For formal modeling of a specific issue, it uses the tableau representation. This is the most flexible of those currently available, and is also used in existing software such as CONAN (Howard 1986, 1989) and DecisionMaker (Fraser and Hipel 1984, 1988). Since tableaux can appear quite daunting however, particular care is needed to maintain user-friendliness. Strategic Maps provide an easily understood way of displaying the results of analysis. Rather than leaving these to be drawn on paper, INTERACT has the facility to produce them automatically on-screen. Throughout, emphasis is placed on the importance of differing perceptions: every tableau and map expresses a view of the situation attributed to a specified actor.

Of the four forms of complexity listed above, the software thus helps one to manage three: combinatorial complexity, differing perceptions and interlinked issues. There is no facility as yet for building multi-stage models, though use of strategic maps allows one to address the dynamics of the situation to some extent.

4. Designing INTERACT

INTERACT is implemented in a windowed environment—specifically, "Hyperwindows" (Elder 1992)—as these have proven advantages in ease of use (Card 1984). Within this approach, design of the user interface is clearly of paramount importance. To allow one to enter information and perform analyses with minimal effort, a *direct manipulation* style was adopted. The user is presented with a picture of the objects within the program, which can be operated on directly, usually via a mouse. Similarly, the program receives commands via buttons displayed on-screen. This form of interface commonly has three main advantages. Firstly, direct representation facilitates learning (Shneiderman 1983), as one can see the model and manipulate it in a natural manner. Secondly, the display provides a constant guide to the facilities available and a picture of the current model. This should benefit both new and occasional users, drawing on the superiority of recognition over recall memory (Eysenck 1988). Finally, it should also be relatively easy for clients in a facilitation exercise to see and understand the progress of analysis. It is sometimes claimed that a direct representation interface allows syntactic errors to be eliminated altogether, as only operations which make sense to the system are allowed. Though this is an exaggeration, the system can adopt a "do nothing" response to syntactic errors, for example by ignoring any attempt to carry out an undefined operation. This has no dire consequences, and it should be obvious that nothing has happened. The display remains as a cue to suggest other ways of proceeding, while the user is spared patronising error messages to punish failure.

Ease of use was further sought by maintaining consistency between appearance, function and operation. For example, buttons with INTERACT windows look and work just like the Hyperwindows function buttons. Moving an object on the screen is always done in exactly the same way. Colors are used consistently, for example in signifying buttons and objects currently selected. Use of modes within the system has been minimized—for example, by avoiding the need for a "delete" mode—to lessen the risk of inexpert users becoming trapped in some unfamiliar mode (Smith et al. 1982). (Many readers will have experience of trying to escape the grip of a supposedly user-friendly word-processing package.) Given the virtual impossibility of abolishing all modes, three exist within INTERACT, their dangers minimized by clear signposting. Finally, all operations are either easily reversible or, failing that, have to be confirmed by the user. Though the latter can appear clumsy, the number of irreversible operations is very small. Users are thus encouraged to experiment—and hence learn—with less fear of making irretrievable mistakes.

5. Structuring and analysis: A worked example

Rather than trying to recreate a decision-support exercise, we illustrate the modeling process using a conflict reported in the public domain and probably familiar in outline to most readers: that of the dispute between Russia and the Ukraine over the future of the Black Sea Fleet. It should be stressed that this is only intended to illustrate the modeling process. We make no strong claims for the specific models, which are based only on newspaper accounts.

5.1 Problem structuring

The Preliminary Problem Structuring (PPS) notation shows relevant actors as labeled boxes, joined by lines repesenting interactions over specified issues. INTERACT presents the user with a window in which to develop such diagrams. A button on the screen is used to add up to ten "actor boxes," which can be positioned anywhere within the window. Another button adds an issue between any of the actors entered so far, selected via a subsidiary window. Up to nine issues can be added, represented by different styles and colors of line. The PPS diagram of figure 1 shows some issues around the Black Sea fleet dispute. The issue of the fleet itself involves Russia, the Ukraine, the Commonwealth (CIS) High Command, and Forces (serving) in the Ukraine. Some linked issues are also shown: that between relevant republics on the future of nuclear weapons ("Strat Nukes"),

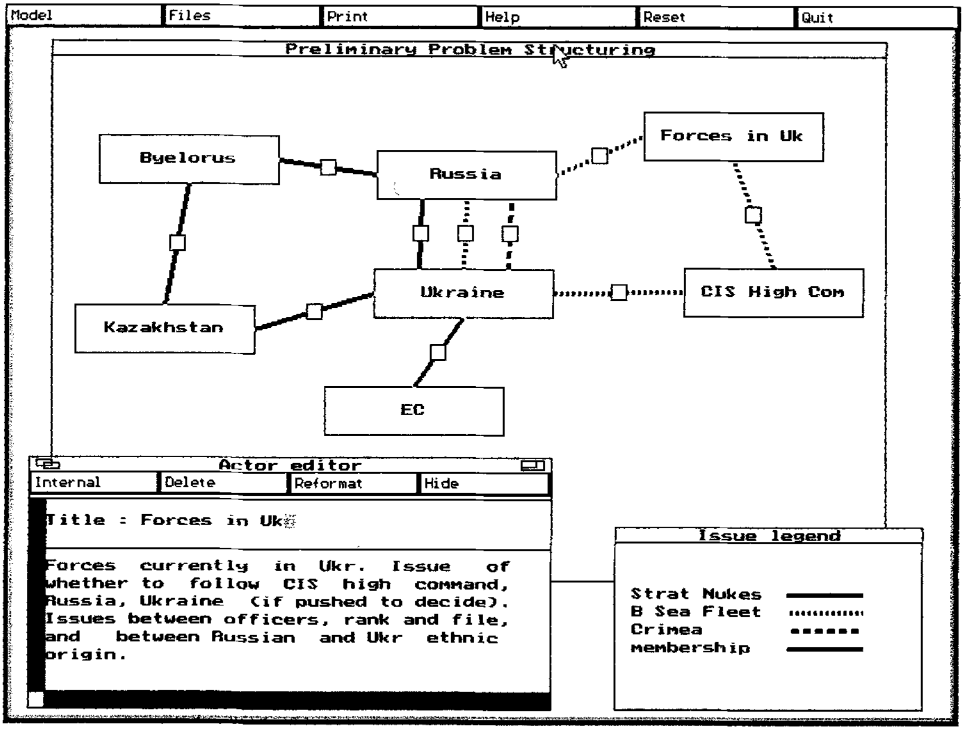

Figure 1. Illustrative PPS window, with actor details editor and issue legend.

the dispute between Russia and the Ukraine over the territory of the Crimea, and between the Ukraine and the EC (over Ukraine's hope of eventual membership). In building up this picture, we are implying that the issues are linked, in the sense that each actor's policy toward one issue is likely to be affected by its involvement in the others. Clearly one could add more; the model is not intended to be exhaustive.

The PPS window acts as an "interactive flip chart" on which actors and issues can be added, changed, or removed, or the picture redrawn, at any time.[2] A "details editor," within a subsidiary window, allows one to (re)name any actor or issue, and add relevant commentary in ordinary text, for example explanation of why an actor is important, notes about other issues affecting likely aims, etc. The small window to the left of figure 1 shows commentary on the actor "Forces in Ukraine." Similar details editors can be activated for other elements of the model (options, scenarios, infeasibilities). They perform a role analogous to the use of "Post-its" to annotate a model drawn up on flip-chart, but can store much larger amounts of information. Use of the details editors allows one *to build up and maintain a structured database about the problem,* which can be accessed and modified at any time. Those attached to actors additionally allow internal PPS diagrams to be built up, showing issues being played out by sub-actors within the original one. This process can be repeated, for example, to represent issues between national governments, between political groupings within each of them, and so on.

5.2 Modeling a specific issue

To proceed further, one must decide on an issue to analyze, perhaps an issue felt to be most important, or most urgent, or simply most interesting. (One can return to the PPS diagram later and choose another issue, but this will be a separate analysis.) Similarly, we need to choose an actor from whose perspective the issue is to be looked at: the software allows one to build up analyses of each. Having highlighted a "current" issue and actor, a model can be built up by introducing binary options for each actor. Figure 2 shows a set of actors and options for the Black Sea Fleet issue, looked at from the point of view of "Ukraine." A new option for any actor is introduced simply by pressing the appropriate button, and a details editor used to (re)name it, add comments, etc. For example, some elaboration of the Russian option "move first" is shown here.

The software invites one to enter options for those actors previously specified for that issue. However, one can add, ignore, or delete actors, for example, so as to model cases in which there is disagreement as to who the relevant actors are. It is usually not difficult to come up with a substantial list of options. In line with the philosophy of managing rather than maximizing complexity, however, not all the options or actors listed at this stage need to be taken forward into the analysis. One can focus analysis more or less narrowly, and—most importantly—change

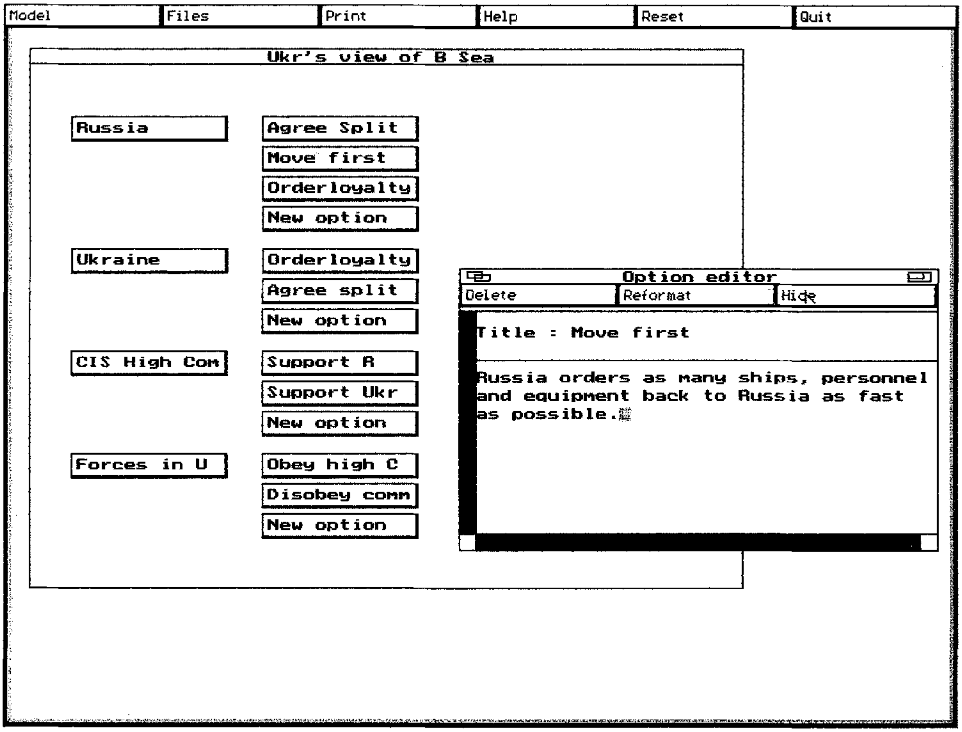

Figure 2. Some options for the "Black Sea Fleet" issue, with details editor for one.

the focus later on. It generally helps to start with a simple model of not more than 8–10 options. The constraint is not what the software can handle (up to 20 options can be entered), but what the user can usefully manipulate. Once some progress had been made, one can come back and extend the model. This helps to keep the complexity of the model within bounds, and avoids having to cope with too much new material at once.

5.3 Developing the analysis

When the user chooses to move on to analysis of the specified options, INTERACT will bring up an *Analysis Tableau:* the main "working sheet" on which to develop the formal model. Figure 3 shows such a tableau. As can be seen, not all the options from the previous list have been included. The tableau is used to add specific scenarios, assumptions about the compatibility of options, and preferences attributed to the actors, all within the current perception of the issue. These can be attended to in any order, but are described in sequence here.

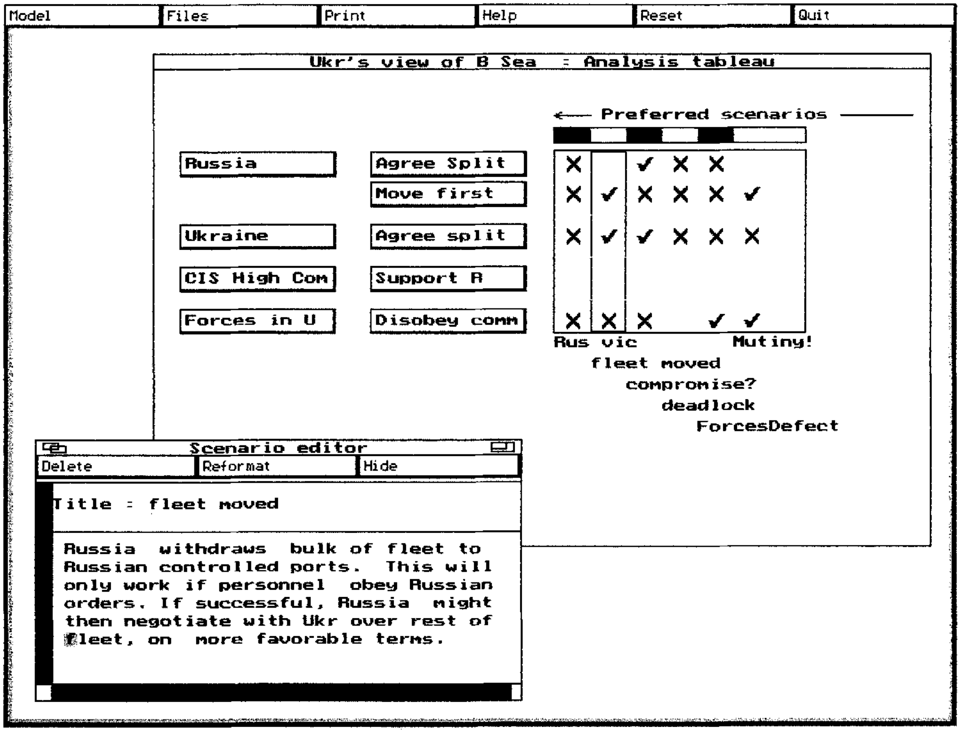

Figure 3. "Black Sea fleet" Analysis Tableau, with six scenarios entered (in preference order for Russia).

Individual scenarios are added by a placing a tick, cross, or blank against each option, to represent options taken up, not taken up, or unspecified. Each column in the tableau thus represents a scenario. For example, the second column represents the scenario "fleet moved," in which Russia decides to move the fleet away even though Ukraine is prepared to split it, and this command is not disobeyed by the Forces. Some commentary on this scenario is shown in the details editor. In the tableau itself, note that the row associated with the High Command's choice has been left blank. Again, following the principle of starting simple, we are first modeling the interaction between the other actors, temporarily ignoring the High Command's choices. So far, six scenarios have been added to the tableau: more can be added at any time, up to a maximum of 100. It will normally be worth considering at least the status quo, solutions currently advocated by each actor (their "positions" on the issue), conflict point(s) reached if the parties insist on their positions, and possible compromises. Adding scenarios one at a time can be time-consuming, but this approach means that the model is built up step-by-step and is more likely to be understood. Alternatively, the software will list all feasible scenarios, and the user can choose to add them all (up to the 100 maximum) to the working list. Unless the model is small, this is not recommended: adding too much at once makes it too easy to lose track of what the scenarios actually mean. However, the listing facility provides a useful running count of how many feasible scenarios there are. It can also be used, once preferences have been assigned, to search for scenarios at the extremes of a player's preferences, or close to one already in the Analysis Tableau.

Some options will usually be judged to be mutually incompatible, rendering certain scenarios *infeasible*. Such judgements can be entered in two ways. First, options for a given actor may be defined as direct alternatives, in the sense that one and only one must be chosen. (The software will show such alternatives bracketed together.) Second, more varied conditions can be entered by entering infeasible combinations via a subsidiary tableau. In both cases, if the user subsequently tries to add an infeasible scenario to the Analysis Tableau, a message will flash up, explaining which assumption has been violated. It is then up to the user to decide how to respond, by altering the scenario or the infeasibility, or even by ignoring the inconsistency in assumptions. The general principle is that the software should draw attention to possible inconsistencies without dictating what the user must do about them.

Preferences for each actor can also be input in two ways. A rough overall ranking can be entered simply by classing options (for any actor), as "desirable" or "undesirable" for the current actor. For example, one might specify that Russia always prefers scenarios in which the "Forces" do not disobey commands. The software can then display the scenarios in order of preference (most preferred to the left), calculated by comparing the number of desirable and undesirable options implements in each scenario. This method is a useful rough guide, but should be used with care: preferences seldom attach unconditionally to single options. Secondly, any individual scenario can be "picked up" with the mouse and moved to a new position. It is usually most effective to use the two methods in combination. Equality of preference is allowed: the colored bar above the tableau allows one to bracket scenarios together, the color only changing when the preference ranking changes. The scenarios in figure 3 are arranged in an order that might plausibly be attributed to Russia: preferences for each other actor would be entered in turn. Any ordering can be altered at any stage of the analysis.

5.4 Outputs of analysis: stability and improvements

As already mentioned, the INTERACT software can perform analyses to check the consistency of the data input, and to list all feasible scenarios generated within the current model. However the main forms of analysis are based around the idea of improvements and sanctions. These terms have the same technical meaning as in the "standard" form of metagame analysis (Howard 1987). Let us consider a particular scenario S1, and actor A. As the name implies, an *improvement* for A from S1 is a move to another feasible scenario (S2, say) which A prefers to S1, and can reach solely by changing his or her own choice of options. In other words, this is a possible "unilateral" move, in which everyone else's choices remain fixed. If such an improvement exists, however, another actor (B, say) may have one or more *sanctions* available. A sanction is a move which B could implement starting from S2—i.e. a possible response to A's projected move—that would make A worse off than in the original scenario S1. Such a sanction thus represents a deterrent against A moving from S1. If there is no such sanction by any other player, A's improvement is said to be *guaranteed*. That is, A may be better off by moving, and can certainly be no worse off, however the other actors respond. One criterion (actually, the weakest or most general criterion) for a scenario to be stable is that no player should have a guaranteed improvement from it.

So far, we have discussed only moves by single actors (as in "non-cooperative" game theory). However, the concepts just outlined can easily be generalized to consider *coalitions* of actors. A coalition, in this context, is simply a set of actors who are able to commit themselves to some joint choice of options. An improvement for a coalition, or "joint improvement," is one that can be implemented by the actors in the coalition, and which leads to an outcome they all prefer. (The case is analogous for a sanction.) Thus extended, the analysis can consider not only individual actors' choices, but also the possible benefits of getting together. As always, conflict and collaboration can coexist: also, collaboration may be at others' expense.

When commanded to do so, INTERACT analyzes the current Analysis Tableau, using the logic just outlined. It is important to remember that this "logic" refers *to a specified actor's perception* of improvements, sanctions, etc. It will list scenarios (a) from which there are no guaranteed improvements for any single actor, and (b) from which there are none for any actor or coalition. In technical terms, these represent respectively the set of *meta-equilibria* and the *core* of the perceived "game" defined by the current tableau.

In addition (and, in practice, often more significantly), INTERACT can also detail the specific improvements from any given scenario, together with any associated sanctions. Sanctions are additionally classified as *willing* or *unwilling,* according to whether the actor(s) implementing them would be acting in accordance with their own preferences or against. The significance of this distinction is discussed later on. For the present, we note that there are problems in establishing the *credibility* of unwilling threats and promises: why should another player believe that one will actually carry them out?

Finally, the software can display the analysis just described in the form of a *Strategic Map*. The map shows scenarios as ellipses, with improvements and sanctions as (plain and dotted) arrows: guaranteed improvements are shown as thicker arrows. The arrows are labeled according to the actor(s) controlling the move: others who would also be advantaged or disadvantaged are shown in brackets. The software draws a local map, showing only those improvements coming directly from the current scenario. Starting from any of the scenarios shown, however, we can bring up *its* "local map" in turn. The logic of the analysis can thus be displayed step-by-step.

Figure 4 shows some typical outputs of analysis, given plausible (though debatable) assumptions about each actor's preferences. First, there is analysis around the "deadlock" scenario (the status quo when the model was built). Two improvements from this are described both in words and in map form:

> First, Forces in the Ukraine (For) could—given plausible assumptions about their preferences—move to a more highly-preferred scenario by "defecting": simply making it clear that they will not necessarily obey orders. This would be to Ukraine's benefit. Russia has a sanction against such a move, but this would involve giving a direct order for the fleet to sail, thus precipitating a mutiny. While this latter scenario is not wanted by the Forces, it is not wanted

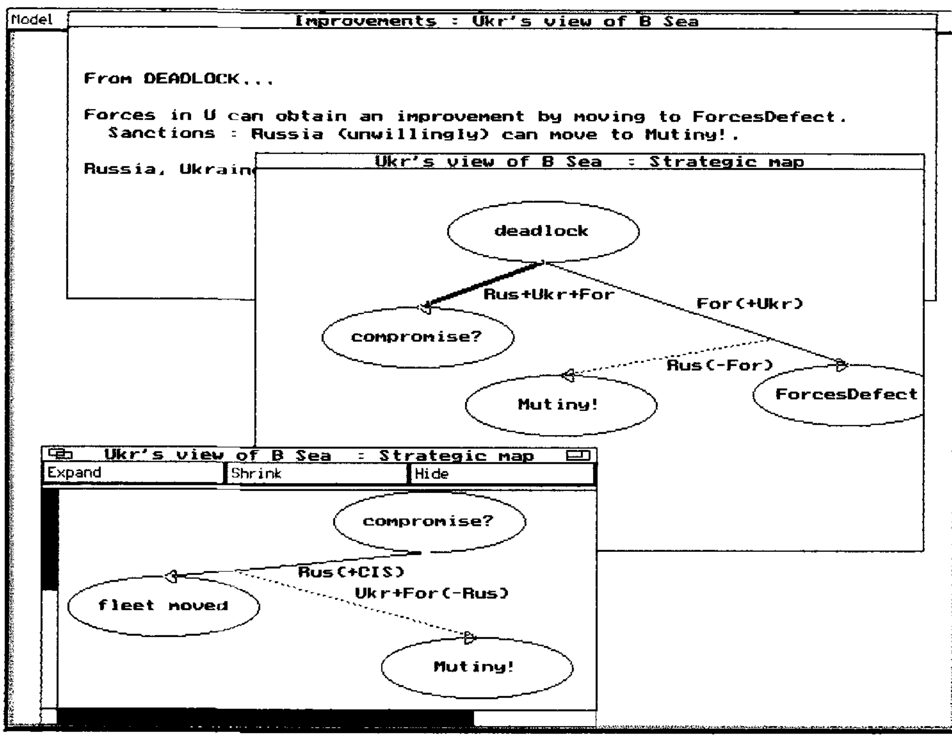

Figure 4. Analysis of improvements from "Deadlock" scenario in "Black Sea Fleet" model, in text and map form. The lower window extends the analysis starting from "compromise." *Note*: Solid arrows denote improvements, dotted arrows sanctions. Thick arrows are guaranteed improvements. The arrows are labeled with the actors controlling the improvements and sanctions: others who would benefit or lose by the move are shown in brackets.

by Russia, either. In other words Russia, in trying to deter defection, can only threaten to use the "unwilling" sanction of turning defection into outright mutiny—a dangerous policy!

Second, however, all sides would gain by moving from "deadlock" to "compromise." No sanctions are so far apparent, so what are the problems with going there?

Some clues to the second question are provided by looking at further improvements from the compromise scenario. As shown in the lower map, Russia could improve by reneging and trying to move the fleet. However, the Forces in the Ukraine have a sanction against this: they can disobey orders, so leading again into the "mutiny" scenario. Once again, this may well be an unwilling sanction: its credibility—given the divisions within the forces noted at the PPS stage—may well be a crucial factor in how the dispute is resolved.

Having made some progress with this simple model, we can complicate it a little by "activating" CIS High Command. Figure 5 shows a tableau made by expanding the previous one. All the previous scenarios now specify also that the High Com-

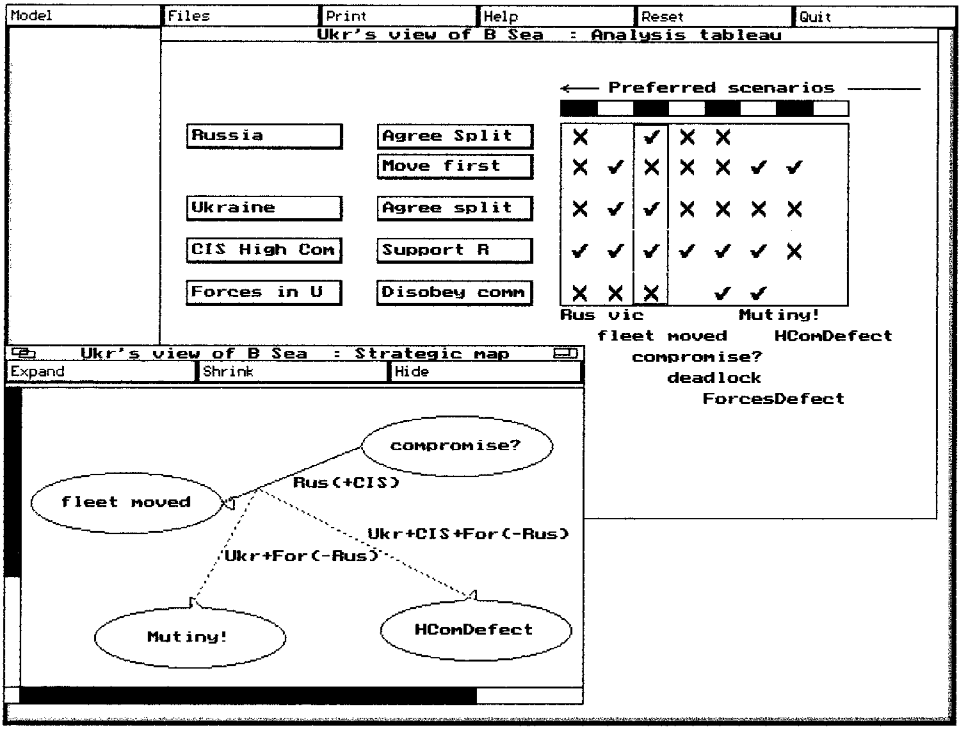

Figure 5. Further model of "Black Sea Fleet" issue, taking account of some "CIS High Command" choices.

mand continues to support Russia in trying to impose discipline from the centre. However a new scenario, "HComDefect," introduces the possibility that the High Command might stop its support by declining to issue orders that would provoke a mutiny. Rather than risk mutiny, the High Command might prefer to settle for a comparatively orderly loss of central control. Incorporating this assumption, the new strategic map shows this as another possible sanction (For Forces, High Command, and Ukraine jointly) against a Russia reneging on a compromise agreement.

5.5 The HELP system

Use of a windowed environment has allowed us to provide an on-line HELP facility using Hypertext (Conklin 1987). Its purpose is not to obviate the need for a (brief) printed manual, but rather to help one find relevant items of information quickly. To this end, the system is *context-sensitive:* the information shown relates to those tasks that the user is likely to be carrying out. The division of facilities within INTERACT into distinct tasks performed via separate windows provides a natural structure for achieving this. Whenever the user requests help, information relating to the active window is displayed, information which should be closely related to the task in hand.

The system currently contains around 7,000 words, divided into about forty discrete units (Hypertext "nodes"). Key words in each section of text, for example, "issues," form links in the network: these are shown in highlighted "hot text," via which the user can move through the network to get further information. Clicking the mouse on the word, in whatever section it appears, will call up more detailed information, in this case about "issues." That window in turn will have further connections within it, and so on. Figure 6 shows two typical messages, called up by asking more about *issues* and *Preliminary Problem Structuring* (PPS). Most links in the network result from attempting to model the user's probable requirements. However, some related topics do not occur together in the normal sequence of events within an analysis. Where detected, these are linked to provide extra cross-referencing. The end result can be seen as a (huge) set of sequential mini-tutorial guides. The user may start with a very general query, such as asking about the term "analyze." The first section encountered will provide broad information about analysis, and one can select key words from this screen to find out more about specific topics. A personalized guide results that reflects the user's choices of inquiry topics and hence the gaps in that particular user's prior knowledge.

Hypertext systems exploit users' own powers of association to retrieve relevant information. Paradoxically, this is sometimes cited as a weakness (Fiderio 1988) on the grounds that users unfamiliar with the subject matter, or with weak associative powers in general, will have difficulty making the appropriate connections. However, because INTERACT is aimed at users with some knowledge of conflict

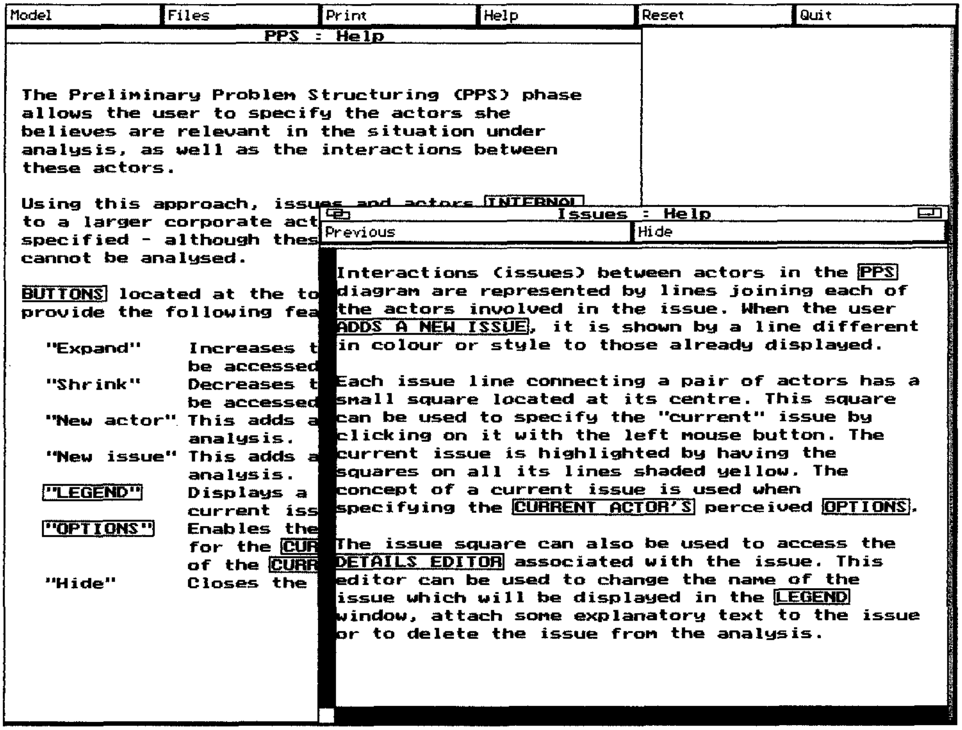

Figure 6. Typical "Help" messages.

analysis, and resembles familiar pen and paper methods, this should not be a problem here. (We also suspect that users attracted to this form of analysis will tend to have fairly strong associative powers.) Another criticism is that users can become disoriented by a large network, as compared with the ordering of material provided by an ordinary document. One way around this is to provide a "graphical browser": small on-screen maps showing where one is in the network. However, the current system is probably too small to need one.

6. On the potential benefits of analysis

Our illustrative analysis, though based only on second-hand reportage, does seem to capture some of the dilemmas faced by various actors, and perhaps provides insight. The ease with which models can be built and tried out makes it not altogether far-fetched to imagine this sort of analysis being carried out on-line with some of the decision makers involved, or at least their close advisors. Here, we have produced an "as if" model of one party's perception. In reality, involvement with one party will give the analyst relatively privileged access to that side's aims and beliefs, but it will be particularly important to try out alternative assumptions about other actors' views. Carrying out analysis for one side is not the only possible form of active involvement: for example analysis might be produced to assist a mediator, or another party affected by the course of the conflict (for example, a company deciding whether to invest in the region).

But what, in general, might one hope to achieve by analysis for real? First, it is important not to overstate potential benefits. Analysis will *not* yield "right answers" in the form of prescriptive solutions. This is not just because of the simplifications inherent in modeling. Interactive decisions often *have* no right answer: no amount of analysis will find one for "chicken." Rather, one can aim to clarify issues and trace the possible consequences of different policies. Hence it may well suggest better ways of managing the interaction.

As with other decision-support methods, potential benefits attach both to the process and the products of modeling. The *process* can help important issues to be surfaced and resolved. The chance to contribute to analysis can itself increase ownership of the results and commitment to agreed actions. Ideally, the end *product* of analysis is an agreed "package" of immediate actions and communications, contingent responses to other actors' moves, and proposals for finding out more about uncertain options, aims and perceptions. We have in mind a formulation similar to the "commitment package" used in Strategic Choice (Friend and Hickling 1987), though the underlying forms of analysis differ. The package is informed by the analysis, but does not flow from it in a mechanical way: interpretation of the results is all-important.

Formally, analysis will tell us about the stability of scenarios in a model, and about patterns of threats, promises, and possible commitments. By taking into account joint improvements (and sanctions) it may help to identify potentially advantageous agreements. But the formal results are only the "bare bones" of an argument; more specific conclusions depend on contextual knowledge of the real situation. For example, identifying unwilling threats (or promises) often provides important clues about actors' behavior. How can one try to make credible an undertaking to act against one's own preferences? Once this question has been prompted by the model, it is not difficult to think of some *generic* answers. For example, credibility may be sought by invoking longer term goals ("we must establish a reputation for trustworthiness") or other linked issues ("if we give way on this, we'll end up giving way on that"). Howard's theory of emotions and changes in preferences, referred to briefly earlier, is also important. He suggests that emotions will tend to act so as to resolve the paradoxes of rational choice: the need to make unwilling threats credible creates negative emotions toward the "other side," while unwilling promises generate positive emotions. In each case, emotional pressure will tend to produce temporarily "irrational" (i.e., counter-to-preference) behavior and/or longer term preference change. All these ideas serve to put flesh on the bare bones of analysis; however, exactly which are going to be most relevant, and in what ways, will be a matter for judgement (usually, the client's). Thus, clients sometimes find that analysis helps them to gain a clearer overall appreciation of the issues; armed with this, they are then well able to think through the more detailed implications for themselves.

7. Current and future developments

7.1 Technical improvements to INTERACT

The existing version of the software—essentially an advanced prototype—can be improved in several ways. Some improvements should allow the software to carry out its current functions more efficiently. At present, data can generally be input and altered with no perceived delay, and the results of most analyses are available within one or two seconds. However, two exceptions to this are stability analysis, if involving more than about 15 scenarios, and listing all feasible scenarios in models with more than about 10 options (potentially, 2^{10} scenarios). In both cases, Fraser and Hipel (1984, 1988) have devised efficient algorithms for carrying out the required calculations. As the definition of stability is different from that employed here, the former algorithm would not transfer directly. However, that employed to calculate the full set of feasible scenarios could be applied and would improve the speed of this operation dramatically.

Another technical limitation of INTERACT as it stands is that display code for some of the windows is tied in with the code for calculating the information to be displayed. This reduces the need for complex data structures in which to record the results of computations, and automatically ensures that all data being used is up to date. However, moving and re-sizing windows in Hyperwindows requires all the windows to be redrawn, so those which have slow display routines severely slow down the overall operation. This could be avoided by separating the computing and display codes, and using a data structure to link the two routines. Windows could then be re-displayed rapidly using the pre-computed result in the data structure. The display-calculate dichotomy has been implemented for the strategic maps, but is also needed in the stability analysis and in the display of all feasible scenarios. Only the fact that these facilities are used infrequently in analysis makes this omission non-critical.

Useful additional features would include efficient ways of dealing with *conditional* options, and the ability to display all scenarios *reachable* by each actor or coalition, from a given scenario. In general, further work will probably concentrate on extending the users of the map, rather than tableau notation, and may well also introduce game tree representations.

One development already prototyped is a *multi-user* version of the software, run over a newtork: this fits in with work elsewhere on "Distributed Negotiation Systems" (Biro et al. 1992). All users share a single model, but each can control only one actor's options. They can thus "play out" the interaction, at the same time communicating by E-mail. This is intended for research and training purposes. For example, personnel managers could explore an industrial relations dispute, interacting both with a model and with each other.

7.2 Toward knowledge-based analysis?

The present HELP facility mainly provides information on the concepts and logic of analysis. But the same approach could be used to structure and use a database of *substantive* knowledge: background information on the case under examination (structured by the formal model), general hints and suggestions on analysis, and perhaps relevant "rules of thumb" about the development and resolution of conflicts. Like the present HELP facility, the knowledge base could be context sensitive, providing information relevant to the current state of analysis. For example, if the model points to the existence of an "unwilling sanction" the user could call up some comments (such as those made earlier in this paper) on tactics often used to make such threats credible. In a similar vein, one might be reminded about typical effects of crises on decision makers' preferences (Milburn 1972; Nicholson 1984). Instead of attempting to use complex AI algorithms to decide which sections of the database were relevant at any given point, the Hypertext methodology would exploit the associative powers of the user *and* the structure of the basic model to select relevant information. This is in contrast to a more ambitious automated reasoning system which would attempt to perform some of the reasoning *for* the analyst. Here, the aim would be for the software to prompt appropriate questions, rather than to provide "expert" answers (Bennett 1991). A still more sophisticated system could allow an analyst to generate and modify his or her own Hypertext database, all of which could be accessed within the overall package.

7.3 Enriching the modeling process

The methods of analysis discussed here are inherently data-hungry. Exploration of "other" actors' possible perspectives, especially, can be taxing. As the models themselves cannot explicitly represent gaps in information, one has to be prepared to make definite, albeit provisional, assumptions, and make quite bold simplifications, in order to proceed. This necessity contrasts with other decision-support methods such as SODA (as regards complexity) and Strategic Choice (as regards uncertainty) (Rosenhead 1989, Chs. 2, 3 and 6, 7, respectively). It can cause frustration in a decision-support setting. Clients may worry that the "richness" of their problem descriptions is getting lost (though judicious use of the details editors can help). One line of development might thus be to combine this form of modeling with one able to represent clients' reasoning about different aspects of the model, such as *why* a certain outcome is desirable, or a scenario infeasible.

This might be done by allowing one to move, via the details editors, into the medium of Graphics-COPE (Eden and Ackermann 1992). Combining "games and maps" in this way has already been tied with pen-and-paper models. In general, the prospects for combining models and methods look promising (Eden and Radford 1990).

7.4 Dynamic modeling

The models described so far represent only "snapshots" in the development of an interaction. However, some elements of dynamic analysis can be inferred from the analysis of strategic maps, and the analysis of emotions provides important clues as to how preferences may change. To go further, one could say that analysis of a tableau covers only one episode in an unfolding "drama" (Howard, Bennett, et al. 1993). The parties' attempts to act rationally at each stage, and the dilemmas they thereby encounter, provide a key to understanding how both they, as characters, and the plot develop over time. This opens up the challenge of analyzing a tree of possible episodes in the drama, each developing from the last, and represented by a model of its own interactions. Collaborative work to develop such "analytical drama theory" is currently underway.

7.5 Final comments

For the present, the software now available greatly enhances the practicality of using the models described here for decision support, as distinct from off-line or historical case studies. Models can be built, modified, and stored much faster than when relying on pencil and paper. One can stop anywhere in the modeling process and add or change information, while holding onto the existing model. New input causes all parts of the working model to be updated, and one can move at will from one part of a model to another. One can immediately carry out analyses, such as the listing of feasible scenarios, that could otherwise take hours even for relatively small models. INTERACT is thus a useful addition to the analyst's armoury, and is also already being used for teaching purposes at several universities worldwide. The current version is available from the authors, and we hope that further development will be influenced by a growing user base.

Notes

1. However, it is important to note that INTERACT is *not* designed as a standalone system for decision makers to use without a "chauffeur" to operate the package. Nor is it intended to be used straight away by someone with no prior understanding of the methods of analysis, though it can certainly be used to teach them.

2. Ordinary flip chart, however, retains some advantages. Though successive versions of the model can be saved, using the software does not provide such an immediate and automatic "history" of how the models have developed. In addition, the present version of INTERACT forces the PPS notation into a more strictly hierarchical structure of "actors within actors" than do pen-and-paper methods. In practice, therefore, it may well be worthwhile to combine the use of both computer and flip chart—as is common in other forms of decision support.

References

Bennett, P.G. (1977). "Toward a Theory of Hypergames," *Omega* 5, 749–751.

Bennett, P.G. (1980). "Hypergames: Developing a Theory of Conflict," *Futures* 12, 489–507.

Bennett, P.G. (1988). "Hypergames as an Aid to Mediation?" In C. Mitchell and K. Webb (eds.), *New Approaches to International Mediation.* Greenwood Press, Westport, CT.

Bennett, P.G., S.A. Cropper, and C.S. Huxham. (1989). "Modelling Interactive Decisions: The Hypergame Focus." In J. Rosenhead (ed.), *Rational Analysis for a Problematic World.* New York: Wiley.

Bennett, P.G. (1991). "Modelling Complex Conflicts: Formalism or Expertise?" *Review of International Studies* 17, 349–364.

Biro, M., et al. (1992). "The Design of DINE: A Distributed Negotiation Support Shell." In T. Jelassi, M.R. Klein, and W.M. Mayon-White (eds.), *Decision Support Systems: Experiences and Expectations.* Amsterdam: North-Holland.

Burton, J. (1987). *Resolving Deep-Rooted Conflicts.* Latham, NY: University Press of America.

Card, S., M. Pavel, and S. Farrell. (1984). "Window-Based Computer Dialogues." In *Human-Computer Interaction—Interact '84.* Amsterdam: North-Holland, pp. 239–243.

Conklin, J. (1987). "Hypertext: A Survey and Introduction," *IEEE Computer* 20(9).

Eden, C., and F. Ackermann. (1992). "Strategy Development and Implementation—The Role of a Group Decision Support System." In S. Kinney, B. Bostrom, and R. Watson (eds.), *Computer Augmented Teamwork: A Guided Tour.* New York: Van Nostrand & Reinhold.

Eden, C., and K.J. Radford. (1990). *Tackling Strategic Problems: The Role of Group Decision Support.* London: Sage.

Elder, M. (1992). Working paper, Department of Management Science, Strathclyde University.

Eysenck, M. (1988). *A Handbook of Cognitive Psychology.* Lawrence Erlbuam Associates.

Fiderio, J. (1988). "A Grand Vision," *BYTE* 13(10).

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

Fraser, N.M., and K.W. Hipel. (1988). "DecisionMaker: The Conflict Analysis Program." Distributed by Waterloo Engineering Software, Ontario, Canada.

Friend, J., and A. Hickling. (1987). *Planning Under Pressure: The Strategic Choice Approach.* Oxford: Pergamon Press.

Howard, N. (1971). *Paradoxes of Rationality.* Boston: M.I.T. Press.

Howard, N. (1987). "The Present and Future of Metagame Analysis," *European Journal of Operations Research* 32, 1–25.

Howard, N. (1989). "The CONAN Play." In J. Rosenhead (ed.), *Rational Analysis for a Problematic World.* New York: Wiley.

Howard, N. et al. (1993). "Manifesto for a Theory of Drama and Irrational Choice," *Journal of the Operational Research Society* 44, 99–103.

Huxham, C.S., and D. Macdonald. (1992). "Introducing Collaborative Advantage," *Management Decision* 30(3), 50–56.

Lax, D., and J.K. Sebenius. (1986). *The Manager as Negotiator.* New York: Free Press.

Milburn, T.W. (1972). "The Management of Crises." In C. Hermann (ed.), *International Crises*. New York: Free Press.
Nicholson, M. (1984). "Aborted Wars and 'Prosaic' Utility Functions," University of Manchester.
Radford, K.J. (1980). *Strategic Planning, An Analytical Approach*. Reston, VA: Reston Publishing Co.
Rosenhead, J. (ed.). (1989). *Rational Analysis for a Problematic World*. New York: Wiley.
Schelling, T.C. (1960). *The Strategy of Conflict*. Cambridge, MA: Harvard University Press.
Schrodt, P.A. (1987). "Pattern Matching, Set Prediction and Foreign Policy Analysis." In S.J. Cimbala (ed.), *Artificial Intelligence and National Security*. Lexington.
Shneiderman, B. (1983). "Direct Manipulation: A Step Beyond Programming Languages," *IEEE Computer* (August), 57–69.
Thomas, L.C. (1987). "Using Game Theory and its Extensions to Model Conflict." In *Analysing Conflict and its Resolution*. Oxford: Oxford University Press.