INTERACT: DEVELOPING SOFTWARE FOR INTERACTIVE DECISIONS

Peter Bennett and Andrew Tait

Department of Management Science
University of Strathclyde
26 Richmond Street
Glasgow
G1 1XH
UNITED KINGDOM

Paper prepared for 34th Annual Convention, International Studies Association
23-27th March 1993, Acapulco, Mexico

Abstract

This paper describes the development and use of a piece of software, "INTERACT", for analysing situations under the control of several interested parties, with particular relevance to International Relations. After briefly discussing forms of analysis used, the paper outlines the design philosophy of the software itself. Various modes of use are considered, including those of "decision support" - in which analysis is done "live" with decision-makers or their advisors - teaching, and research. The functions typically performed in the course of analysis are illustrated by means of a worked example. Finally, avenues of current work are outlined, in particular the development of a "multi-user" version for training and experimental studies, and the prospective development of a knowledge base to underpin the analysis.

An earlier version of this paper was presented to a Conference on Decision-Making Under Conditions of Conflict, University of Waterloo, Ontario, Canada, August-Sept 1992.

March 1993

INTRODUCTION

Most paradigms of International Relations and Political Science stress the significance of situations involving several separate parties, each with a stake in what happens and some capacity to affect it. Such situations provide particular challenges, both to analysts and decision-makers. Because no actor has complete control of events, each must try to take the others' possible actions into account. It may well be that, for any actor, the most appropriate action depends on what others may decide 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* (rather than just the parties) thus interact with each other. Even if they are *taking* decisions separately, they will find themselves in an outcome determined by the choices made by all.

Such situations, typically involving **partial** conflict rather than "straightforward" antagonism, bring forth possibilities not only for mutual threats, deceit, bluff and counter-bluff, but also for cooperation and "collaborative advantage" (Huxham & Macdonald, 1992). Though the term "conflict analysis" id frequently used to cover work of the sort to be described here, this can give the erroneous impression that one is concerned only with outright hostilities. Rather, one should recognise that where aims partially diverge, conflict and cooperation become inseparable (Lex & Sebenius, 1986). These preliminary comments are fairly commonplace, whether the "actors" being studied are the (supposedly) unitary nations posited by the classical IR realist, the wider set of parties seen by adherents to a "world society" view, the factions within each national polity, and so on. Nevertheless, methods for analysing interactive decisions remain in need of further development. This paper describes what are hoped to be some steps forward.

FORMS OF ANALYSIS

The models used here have their basis in game theory, and developments from it including metagame and hypergame analysis (c.f. Rosenhead, Ed, 1989). Given the unhelpful connotations of the word "game", however - e.g. triviality; winners and losers - this terminology is best avoided in practice. The basic building blocks of analysis are:

- the relevant **actors** (who may be individual people, or groups or organisations), engaged in some specified issue,
- the possible **options** (actions or policies) open to each, with respect to the issue,
- the **scenarios** (or outcomes) that can result from possible combinations of choices by the actors,
- the **preferences** each actor has for the possible scenarios, reflecting that actor's aims, interests, likes and dislikes.

Even very simple models can illustrate some quite subtle points about the interaction of decisions. For example, the well-known game of Chicken shows up genuine dilemmas of rationality. (If the other side uses the same reasoning as oneself, both choices always 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 an *unwilling* threat, one that means acting against one's own preferences. Genuine emotions are likely to be generated by the attempts to make such threats credible, intermixed with more cold-blooded attempts to bluff. Similarly, agreeing to cooperate in "Prisoners' Dilemma" involves an unwilling promise.

Many real situations, from the interpersonal to the international, have elements akin to such games. But this is not to suggest that there is nothing else to them. Modelling in terms of actors, options and preferences will at best provide a partial understanding (for example, actors' needs may be neglected (Burton, 1987)). Even if the basic framework is accepted, 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 sequence in which moves are made matters. More generally, the relevant players, and 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 modelling, as in an Expert System The work reported here concentrates on the former approach, but provides a possible link into the latter.

Some formal methods already exist which some way toward meeting the complexities just noted. Combinatorial complexity is addressed using the *Analysis of Options* approach. The actions open to each side are modelled 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 analysed 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.

The recognition that issues may be conceptualised quite differently forms the basis of *Hypergame Analysis*. 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 is provided the theory of emotions within *"soft game theory"* (Howard, 1989). This 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 emotion (and hence irrationality and/or preference change), while unwilling promises generate positive emotions.

AIMS OF THE PACKAGE

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

One attraction of having a computer package is the ability to build up and analyse models more complex than could easily be done with pen and paper. However, our 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 modelling 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 an opportunistic way. One should also be able to move at will between different sorts of model, and add or modify material at 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, building and analysing models in "real time" (rather than "going away and doing some analysis"). It was envisaged that in most cases, the client would be one of the parties to the situation being analysed. This means that the analyst has relatively privileged access to that side's aims and beliefs, but makes it still more important to try out alternative assumptions about other actors' views. One might well also be dealing with a client group, with differing views about the issue in hand. Such differences would have to be managed, while retaining the explicit focus of the analysis on managing the client's interactions with external parties. Finally, the analyst should not be presumed to be an expert in computing, though he or she should certainly be familiar with the ideas underlying the analysis. 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". (Since it was unlikely to be used on an everyday basis, 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.

- minimise the possibilities for user error. Error messages, even for trivial mistakes, are embarrassing during a facilitation session. There should be no need to remember a multiplicity of non-obvious commands.

- be portable, and require no unusual hardware.

- avoid the use of "game" terminology throughout.

As to the specific modelling methods used within INTERACT, our general approach has been to adapt and combine existing representations. 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.)

---

[1] However, it is important to note some uses not intended. INTERACT is not designed as a stand-alone system for use by decision-making teams 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. The use discussed here also contrasts with the possible use of analysis to support mediation (Bennett, 1988) or of software to support "distributed" negotiation (Biro et al, 1992).

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 modelling 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 here 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 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. Use of strategic maps allows one to address the dynamics of the situation in a limited way, but multi-stage analysis is not currently covered. Developments in this direction are however in train.

DESIGNING "INTERACT"

To help meet the above aims, INTERACT is implemented in a windowed environment - specifically, "Hyperwindows" (Elder, 1992): these have well-proven advantages in ease of use (Card, 1984). Within this approach, specific design of the user interface is clearly also of paramount importance. To allow one to enter information and perform analyses with minimal effort, a *direct manipulation* style was adopted. This presents the user with a (necessarily idealized) picture of the objects within the program. These can be operated on directly, usually by "pointing" with a mouse. Similarly, the program receives commands via buttons displayed on-screen. Three main advantages were anticipated. Firstly, direct representation makes learning easier (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, given 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: if (say) the user attempts to carry out an undefined operation, the software will just ignore it. This has no dire consequences, and it should be obvious that nothing has happened. The displayed objects and buttons remain as cues to suggest other ways of achieving the intended aim, while the user is spared patronising error messages to punish failure.

Within this overall approach, ease of use was sought by maintaining "consistency" - things that look similar on the screen perform similar functions, and are operated on in similar ways. For example, buttons within windows look and work exactly like the Hyperwindows function buttons. Moving an object on the screen is always done in exactly the same way. Colours are used consistently, for example in signifying buttons and objects currently selected. Use of modes within the system has been minimised - 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-processor.) Given the virtual impossibility of abolishing all modes, three exist within INTERACT, their dangers minimised 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.

STRUCTURING AND ANALYSIS: A WORKED EXAMPLE

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

Problem Structuring

The Preliminary Problem Structuring (PPS) notation shows relevant actors as labelled boxes, joined by lines representing 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 (re)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 colours 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"), 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 involvement in the others. Clearly one could add more: the model is not intended to be exhaustive.

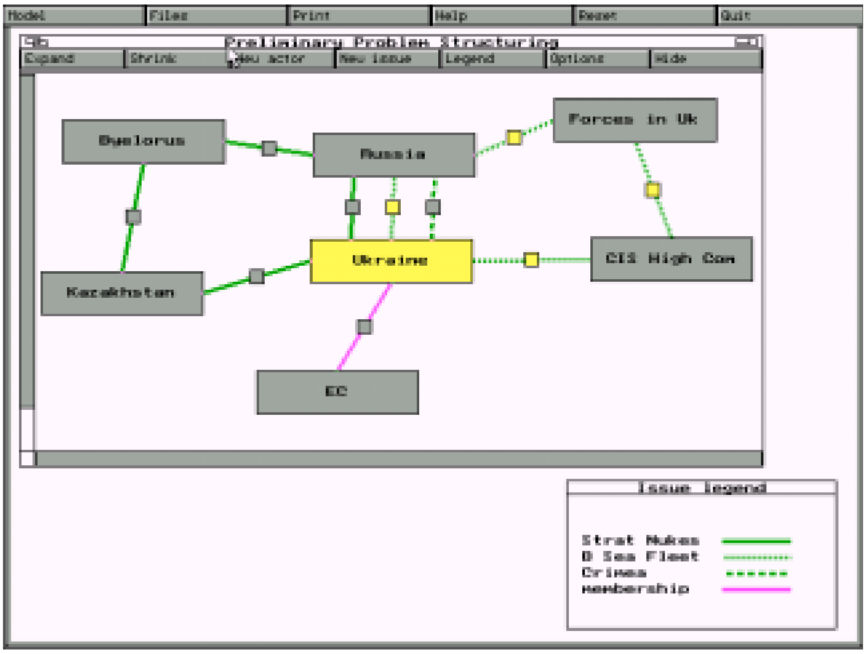

Figure 1: Preliminary Problem Structuring diagram

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" allows one to attach further information about any actor or issue. Using a subsidiary window, the item can be (re)named, and commentary added in ordinary text. For example, relevant comments about an actor can include explanations of why that actor is important, notes about other relevant issues, comments about likely aims, and so on. The small window to the left of Figure 1 shows commentary on the actor "Forces in Ukraine". Similar details

[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

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 is important in allowing one to build up and maintain a structured database about the problem, which can be accessed and modified at any time. The details editors attached to actors additionally allow "internal" PPS diagrams to be built up. This allows one to structure issues being played out by (sub-)actors within it. This process can be repeated, for example to represent issues between firms, between departments within them, and so on down to the level of individuals.

Modelling a Specific Issue

To proceed further, one must decide on an issue to analyse. This may be one felt to be - for example - 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 option "move out" is shown here.

well be worthwhile to combine the use of both computer and flip chart - as is common in other forms of decision support.

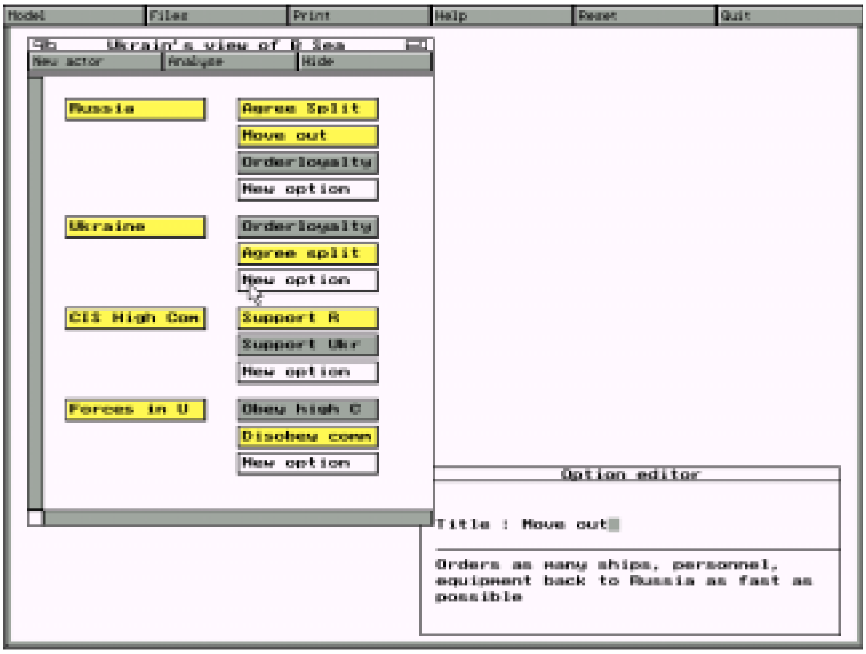

Figure 2: Actors and options

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 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 maximising complexity, however, not all the options or actors listed at this stage need be taken forward into the analysis. One can focus analysis more or less narrowly and - most importantly - change the focus later on. It generally helps to start with a simple model - of not more than about eight or ten options. The constraint is not what the software can handle, 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.

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.

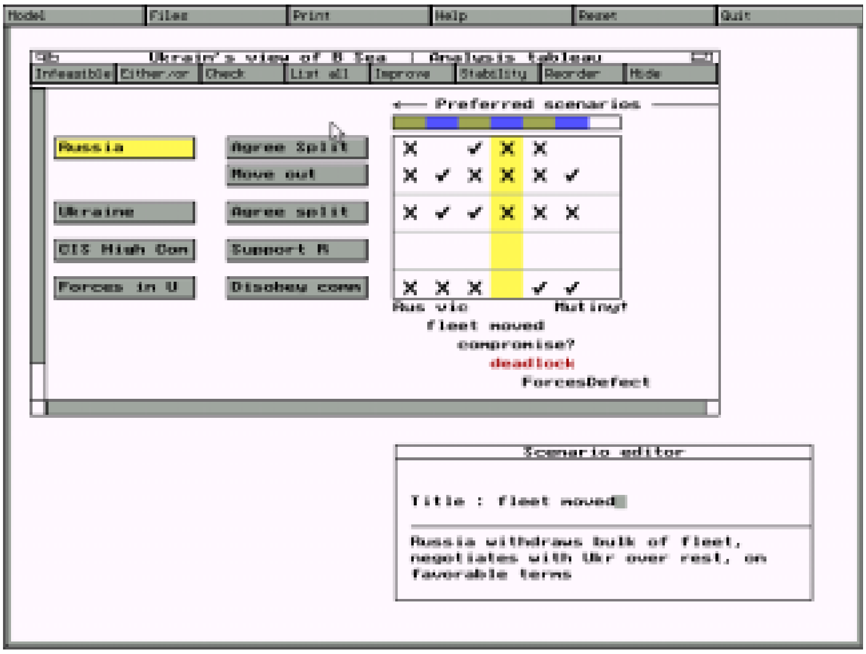

Figure 3: Analysis tableau

Individual scenarios are added by 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 instruction is not disobeyed. Some commentary on this is shown in the details editor. In the tableau itself, note that the row associated with the High Command's choice has been left entirely blank. Again following the principle of "starting simple", we are first modelling the interaction between the other players, temporarily ignoring the High Command's choices. So far the tableau has six labelled scenarios: more can be added at any time. 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 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 to the working list. Unless the model is small, this is not recommended as a model-building method. However, the 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 some scenarios *infeasible*. Such judgements can be entered in two ways. Firstly, 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. Secondly, more varied conditions can be entered by entering infeasible combinations via a subsidiary tableau. 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 (e.g. alter the scenario, or the infeasibility) in general, the software draws attention to possible inconsistencies without dictating what the user must do.

Preferences 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. The software can then display the scenarios in order of preference (most preferred always to the left), calculated by comparing the number of desirable and undesirable options activated 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 coloured bar above the tableau allows one to bracket scenarios together.

Outputs of Analysis: stability and improvements

INTERACT can analyse the current Analysis Tableau to find stable scenarios. It will list scenarios from which there are no guaranteed improvements (a) for any single actor and (b) for any actor or coalition. A coalition, in this context, is simply a set of actors who are able to commit themselves to some joint choice of options. The analysis thus looks not only at individual actors' choices, but at the possible benefits of getting together. As always, conflict and collaboration can coexist: collaboration may be at others' expense. At a more detailed level, the software will find and list all improvements from any scenario, and any sanctions against them. As well as improvements and sanctions amongst scenarios within the Analysis Tableau, the software shows "suggested improvements" if there is the possibility of implementing a generally-preferred option. 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 software also displays the analysis around any scenario 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 labelled 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, and so on. The logic of the analysis can thus be displayed step-by-step.

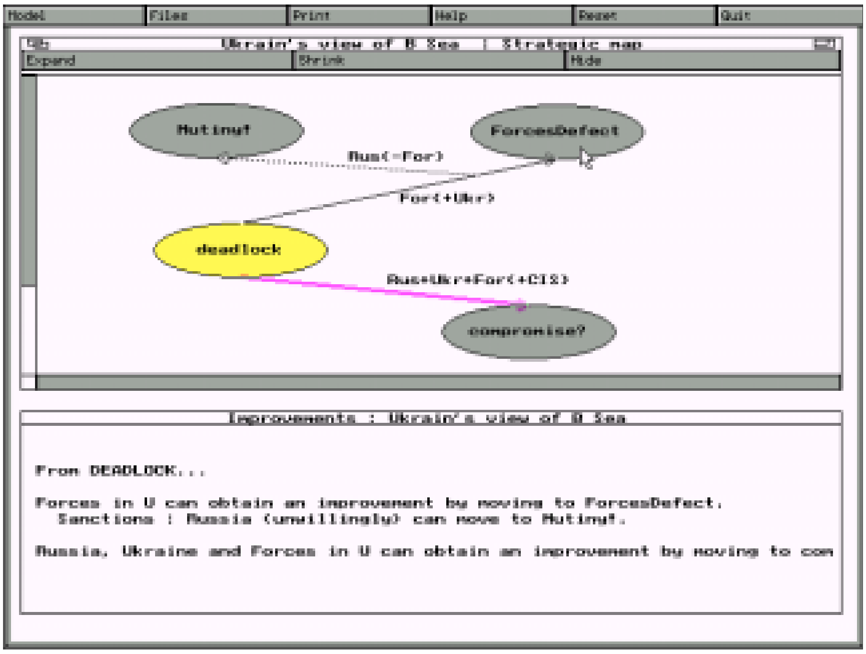

Figure 4: Strategic map

Figure 4 shows some typical outputs of analysis, given plausible (but 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.

> Firstly, 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 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!
>
> Secondly 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.

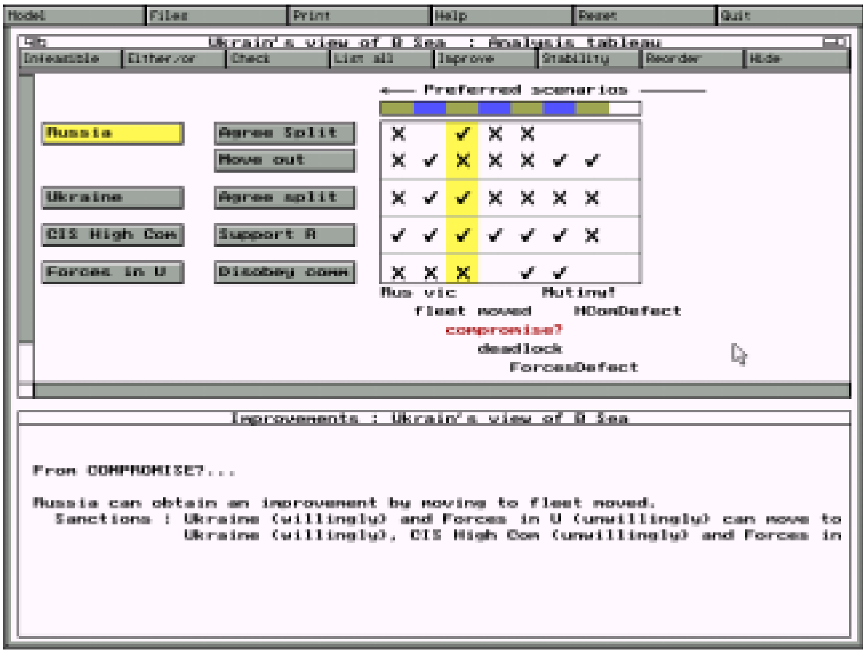

Figure 5: Expanded analysis tableau

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 Command continues to support Russia in trying to impose discipline from the centre. However a new scenario, "H Comm Defect", 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 will show this as another possible sanction (for Forces, High Command and Ukraine jointly) against a Russia reneging on a compromise agreement.

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.

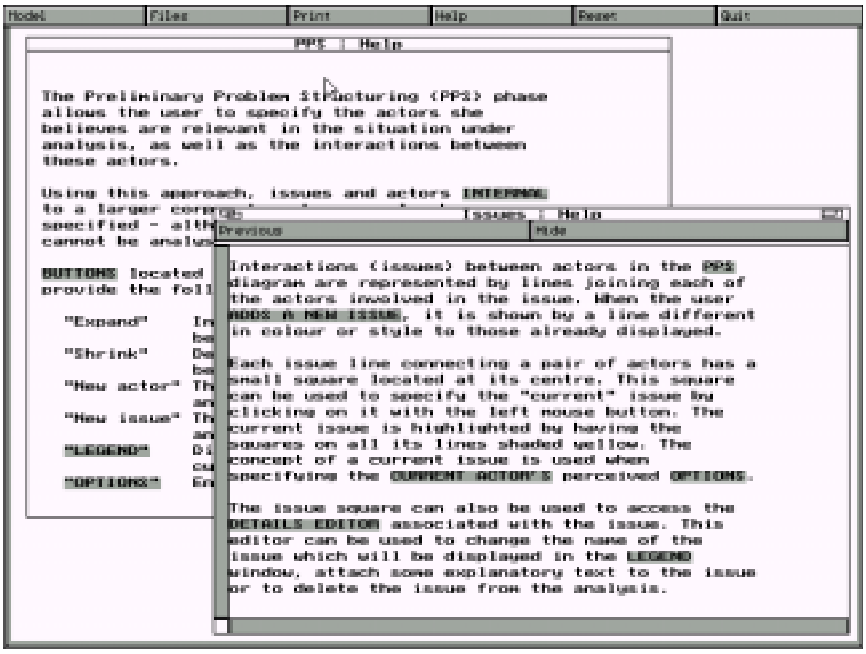

Figure 6: Help system

The system currently contains around 7,000 words, divided into about forty discrete units. Key words in each message ("Issues", say) form nodes 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 message it appears, will call up more detailed information 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 **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 - for example by asking about the term "analyse". The first message encountered will provide broad information about analysis, and one can select key words from this screen to find out more about more specific topics. A personalised guide results, reflecting the user's choices of topics to ask about.

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 analysis, and resembles familiar pen and paper methods, this should not be a problem here. Another criticism is that users can become disorientated 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. The current system is probably too small to need one, though the knowledge base discussed below would do so.

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). We have produced an "as if" model of one party's perception: were one to have the Ukrainian leadership as client, one crucial task would be to combine a model of their view with different assumptions about how other actors (e.g. Russia) might see the situation.

But what, in general, might one hope to achieve by analysis "for real"? Firstly, 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 modelling. 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 modelling. 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. 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". For example, identifying unwilling threats (or promises) often provides important clues about actors' behaviour. 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"). Drama Theory takes the further step of treating unwilling treats and promises as the key to understanding emotion and preference change (Howard et al, Howard, 1993). Analysis of the form undertaken within INTERACT can provide an understanding of a particular episode, while also providing clues to the probable overall development of the drama. All these ideas serve to put flesh on the bare bones of analysis.

CURRENT AND FUTURE DEVELOPMENTS

Technical improvements to INTERACT

The existing version of the software - essentially an advanced prototype - can be improved in several ways. Some should allow the software to carry out its current functions more efficiently. 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 uses of the map, rather than tableau notation, and may well also introduce game tree representations.

Multi-user INTERACT

One development already prototyped is a *multi-user* version of the software, run over a network: 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 international or industrial relations dispute, interacting both with a model and with each other. Trial runs with the prototype version are now in train.

Toward knowledge-based analysis?

The present HELP facility mainly provides information on the "mechanics" 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. 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, *op cit*). A more sophisticated system still could allow an analyst to generate and modify his or her own Hypertext database - all of which could be accessed within the overall package.

Enriching the modelling 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 can cause worry that the "richness" of the problem is getting lost - though judicious use of the details editors can help. One line of development might thus be to combine this form of modelling with one able to represent different forms of reasoning, for example, as to *why* a certain outcome is desirable, or a scenario infeasible. For example, details editors might lead one into the medium of Graphics-COPE (Eden and Ackermann, 1982): combining "games and maps" in this way has already been tried with pen-and-paper models. In general, the prospects for combining models and methods look promising (Eden and Radford, Eds, 1990)

Dynamic Modelling

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". 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 analysing 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 under way.

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. Programs such as INTERACT allow one to build, modify and store alternative models much faster. One can stop anywhere in the modelling 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 facilitator's armoury. But though we have taken the opportunity to illustrate its use, it is only one tool among several available. All have their own advantages, and all will doubtless continue to be improved on.

References

P.G. Bennett (1977): "Toward a Theory of Hypergames" Omega 5 749-751

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

P.G. Bennett (1988): "Hypergames as an Aid to Mediation?" in C Mitchell and K Webb (Eds), New Approaches to International Mediation, Greenwood Press

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

P.G. Bennett (1991): "Modelling complex conflicts: Formalism or expertise?" Rev Int Studies, 17 349-364

M. Biro, E. Bodroghy, A. Bor, E. Knuth and L. Kovacs (1992), The Design of DINE: a Distributed Negotiation Support Shell, in T. Jelassi, M.R. Klein, W.M. Mayon-White (Eds), Decision Support Systems: Experiences and Expectations, North-Holland, Amsterdam

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

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

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

C. Eden and F. Ackermann (1992), Strategy Development and Implementation - the role of a Group Decision Support System, to appear in S. Kinney, B. Bostrom and R. Watson (eds) Computer Augmented Teamwork: a Guided Tour, Van Nostrand & Reinhold, N.Y.

C. Eden and K.J. Radford (1990), Tackling Strategic Problems: the role of group decision support, Sage, London

M. Elder (1992): Working Paper, Department of Management Science, Strathclyde University (forthcoming)

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

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

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

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

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

J.C. Harsanyi (1968): "Games with Incomplete Information Played by 'Bayesian' Players "Management Science (A) 159, 320 & 486

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

N. Howard (1986): "The Usefulness of Metagame Analysis" J.Opl Res.Soc., 37, 430-432

N. Howard (1987): "The Present and Future of Metagame Analysis" Euro.Jnl of O.R. 32, 1-25

N. Howard (1989): "The CONAN Play", in Rational Analysis for a Problematic World, Ed. J. Rosenhead (op cit)

N. Howard (1993): "Dramatic Transformations of the Core" this conference

N. Howard, P. Bennett, J. Bryant, M. Bradley (1992): "Manifesto for a Theory of Drama and Irrational Choice" Nigel Howard Systems / Strathclyde University, July 1992, J.Opl Res Soc. 44 99-103

C.S. Huxham, D Macdonald (1992), Introducing Collaborative Advantage, Management Decision 30(3), 50 -56

D. Lex, J.K. Sebenius (1986), The Manager as Negotiator Free Press, NY

T.W. Milburn (1972): "The Management of Crises" in International Crises, Ed. C. Hermann, Free Press

M. Nicholson (1984): "Aborted Wars and 'Prosaic' Utility Functions", University of Manchester

K.J. Radford (1980): Strategic Planning, an Analytical Approach, Reston, Va.

J. Rosenhead (Ed) (1989): Rational Analysis for a Problematic World, Wiley, 1989

T.C. Schelling (1960): The Strategy of Conflict, Harvard Univ. Press

P.A. Schrodt (1987): "Pattern Matching, Set Prediction and Foreign Policy Analysis", in S.J. Cimbala (Ed) Artificial Intelligence and National Security Lexington, 1987,

B. Shneiderman (1983): "Direct manipulation: a step beyond Programming Languages" IEEE Computer August 1983 pp 57-69

L.C. Thomas (1987): "Using Game Theory and its Extensions to Model Conflict", in Analysing Conflict and its Resolution O.U.P., 1987