# A Decision Support System for Conflict Resolution

by

**Xiaoyong (John) Peng**

A thesis
presented to the University of Waterloo
in fulfilment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Systems Design Engineering

Waterloo, Ontario, Canada, 1999

0-612-38262-1

# Abstract

The major objective of this thesis is to develop a comprehensive decision support system (DSS) for systematically studying real-world strategic conflicts. The main components include the design and implementation of flexible interfaces for formulation and interpretation; construction of novel algorithms for modeling and analyzing small, medium and large-scale conflicts; development of new theoretical ideas in conflict resolution such as an innovative approach to coalition analysis; and incorporation of these new concepts into the DSS. The DSS is based upon existing and new research developments for the Graph Model for Conflict Resolution, and is referred to as GMCR II.

The option form is improved and extensively utilized to represent a conflict in terms of decision makers, options and preferences. Specially designed data structures and corresponding algorithms are implemented for generating possible states, removing infeasible states, coalescing indistinguishable states, and modeling allowable state transitions. For small disputes, a graph-based approach to modeling a conflict is suggested as an input format. Algorithms based on different approaches to facilitate preference elicitation are designed and implemented. Moreover, the DSS permits the choices of different decision makers in a model to be analyzed using different behavior patterns or solution concepts. A range of useful follow-up analyses allows many "what if" questions to be investigated and thereby provides an enhanced understanding of the conflict under study, which should in turn lead to better decisions.

# Acknowledgements

I wish to express my greatest gratitude to my supervisors, Professor Keith W. Hipel and Professor D. Marc Kilgour. Without their valuable guidance, encouragement and support, this research would not have been possible. Special thanks are given to Dr. Liping Fang for his helpful support and advice.

It is my great honor to have Professor Andrew P. Sage of George Mason University as my external examiner. Sincere appreciation is extended to the other members of my examination committee – Professor Mohamed Kamel, Professor Christian Dufournaud, Professor Mitali De and Professor Jim Radford. They have given me valuable suggestions and helped reviewing this thesis.

I am grateful to many faculty and staff members, and fellow students in the Department of Systems Design Engineering, for their companionship and for providing a stimulating intellectual environment.

Finally and most importantly, I would like to thank my family - my wife, my parents and my little daughter - for putting up with me during this especially long and trying period, when I had little to offer in return.

# Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

## 1.1 Strategic Conflicts

A *conflict* is a clash of interests, values, actions, or directions [17]. The word conflict is applicable from the instant that the clash occurs. Conflict in various forms can be found all around us, in international relations, in politics. in economic competition, and in relationships between individuals. Though the most obvious types of "conflicts" may be thought to be wars, revolutions, riots, strikes, and hostilities of other sorts. However, conflict in a more general sense is a natural consequence of one's power to take actions that affect others [7]. Even when we say that there is a potential conflict, we are implying that there is already a conflict of interest or goals, although an explicit clash may not have occurred yet.

Decision situations involving conflicts are known as "interactive" decisions, which involve several parties, each with a stake in the outcome, and some power to affect it. These parties' aims and interests usually differ: though they may not be in total opposition, they are trying to bring about different outcomes. The crucial point is that no one actor has complete control of events. Each must try to take into account – and if possible, influence – the others' possible actions [8]. Their decisions thus "interact" with each other. In fact, Sage [79] points out that in these situations, "nature is replaced by not necessarily hostile opponents", and "people's reactions to each other's actions are of great importance".

More specifically, ***strategic conflicts*** refer to decision problems involving two or more *decision makers* (DMs) [54]. Each DM must select among two or more *courses of action* and the final *outcome* depends on the choices of all DMs. A conflict exists in the sense that DMs have different *preferences* over the possible final outcomes, i.e. their preference rankings are not identical.

Strategic conflicts are so pervasive that they can be found in virtually every area of human interaction. As pointed out by De Bono [17, page 168], "Conflict resolution is probably the most important area for the future of mankind and the continued existence of the world."

## 1.2 Conflict Resolution Methodologies

Formal modeling approaches are required to study the wide variety of conflict situations that can arise in the real world. An important role for mathematics is to provide systematic structures within which arguments can be framed and followed to their conclusion [7]. It is then easier to ensure that assumptions are clearly stated, to examine the resulting models rigorously, and to explore the consequences of different hypotheses. Non-Cooperative Game Theory [87], was the first methodology that aimed to analyze choices, and explain outcomes, in strategic conflicts.

Historically, the idea of game models in term of players, strategies, and outcomes was a great contribution to the modeling of conflict. A "game" is simply a model of a situation with a certain structure of relevant participants ("players"), each with various courses of action available ("moves", making up "strategies"), and with preferences over the possible outcomes [7]. The focus on decision makers (DMs) and their possible courses of action is sometimes characterized as a "rational actor" approach, though "rationality" may sometimes be defined very broadly [7]. Many game-theoretic techniques have not been designed specifically to reflect what happens in the real world, but rather to be mathematically tractable. Classical game theory, as presented by Von Neumann and Morgenstern [87] and Nash [69] is often a difficult and inconvenient way to model many situations that could arise in practice. Often there is insufficient information available for calibrating a game-theoretic model. For many actual problems, there is usually little or no quantitative information, and probabilities are often subjective and "intrinsically unmeasurable" [13]. In practice, one of the most difficult aspects of assessing a conflict is simply organizing the available non-quantitative information. Moreover, to model a wider variety of the strategic behavior that arises in complex conflicts in the real world, the concept of rationality needs to be refined. The need for easy-to-apply tools applicable to complex conflicts and the desire to model a wider variety of strategic behavior, have stimulated the emergence of what this thesis refers to as *conflict resolution* methodologies - a group of methodologies that are distinct from the more traditional approaches of game theory.

Conflict resolution methodologies take into account the possible reactions of a particular DM to the other DMs' known strategies. Moreover, they are absolutely ordinal, which makes conflict models easier to specify and avoids utilities, mixed strategies, and other aspects of game theory that can be difficult to apply and to communicate [50]. These methodologies have two major functions. First, when a conflict is modeled, the available information pertaining to the dispute is put into proper perspective and the problem is systematically structured. Second, the conflict model is analyzed to predict possible solutions to the dispute. Based on the results of a conflict study, a DM can select a realistic course of action that would be most beneficial for his or her purposes. Among this group of methodologies are Howard's *Metagame Analysis* [41], Fraser and Hipel's *Conflict Analysis* [25, 26], Brams' *Theory of Moves* [13] and Fang, Hipel and Kilgour's *Graph Model for Conflict Resolution* [19, 20, 56]. As Kilgour has pointed out [50], "What all of these endeavors have in common is game-theoretic roots—all are essentially game theory variants that have been designed to yield better decision advice or more compelling structural insights."

### 1.2.1 *Metagame Analysis*

Historically, a fresh approach to conflict resolution began with Howard's pioneering development of metagame analysis [41]. Metagame analysis introduces two stability concepts, *general metarationality* and *symmetric metarationality*, which take into account the possible moves and countermoves of the players. An outcome is general metarational for a player if for every unilateral improvement (UI) available to that player, the opponent or opponents are able to respond such that the initial player ends up at a less preferred outcome, which suggests that it might be better for the initial player to remain with the original outcome. An outcome is symmetric metarational for a player if for each of his or her UI(s), the opponent or opponents have responses by which they can guarantee that regardless of the initial player's counter-response, the resulting outcome is less preferred by the initial player to the original one.

Besides defining possible human behavior in conflict situations in an innovative fashion, Howard [41] provided a flexible notation, called ***tabular form***, which formed the basis for the option form, a convenient way to structure and record the main elements of a conflict model. The equilibria resulting from symmetric metarationality and general metarationality are subject to credibility assessment. Metagame analysis does not include an algorithmic method of credibility determination; it is up to the analyst to distinguish whether or not predicted outcomes are based on credible sanctions. Howard [45] has recently used the metaphor of drama to extend metagame analysis to include an understanding of the role of irrationality and emotions in strategic conflicts.

### 1.2.2 *Conflict Analysis*

Improvements and extensions to metagame analysis were provided by Fraser and Hipel [25. 26]. They developed a "conflict analysis" method by adapting traditional metagame analysis to the study of practical problems. Postulating that the credibility of a sanction is related to the preferences of the players, they proposed a convenient method and a straightforward algorithm for the analysis of a conflict. The new solution concepts of ***sequential stability*** and ***simultaneous stability*** were thus developed. Moreover, Fraser and Hipel [25, 24] recommended analyzing every outcome from every DM's point of view for stability.

*Option form* is an improved version of Howard's tabular form used in conflict analysis. With successful applications to real-world conflicts, including large-scale ones, many valuable techniques independent of the stability concept have been introduced under the framework of conflict analysis, including

- *Outcome removal*, to eliminate infeasible outcomes.

- The *Preference tree* [27] method, which ranks the states for a given DM based upon lexicographic preferences over the options.
- *Coalition analysis* [63] [66] [40] in conflicts with more than two DMs, which predicts the coalitions that are most likely to form by determining the overall preferences of a possible coalition, in order to ascertain the strategic implications of coalition formation.
- *Sensitivity analyses*, to assess the relative validity of the results of a conflict analysis even if the information being used is uncertain [26].
- *Hypergame analysis* to analyze conflicts in which there are misperceptions by one or more of the DMs [6, 88].

### 1.2.3 *Theory of Moves*

A recent development, related in spirit to other conflict resolution methodologies, is Brams's Theory of Moves [13]. In TOM (Theory of Moves), the basic equilibrium concept is nonmyopic equilibrium (NME), assuming that most real-life DMs are not so myopic— especially when they are making important decisions – as to consider only the immediate effects of an action, without taking into account possible responses of other players, as well as themselves.

The Theory of Moves comprises several methods for analyzing formal models that differ in several crucial ways from games, although they have much in common. Conflicts are seen as moving from state to state according to unilateral actions by the players—the initial state matters, and the relative powers of the players may be used to continue, or stop, the process.

### 1.2.4 *Graph Model for Conflict Resolution*

The recently developed graph form of the conflict model extends and refines many of these methodologies to describe more accurately the behavior of participants in a strategic conflict [19, 20, 56]. The graph form takes states, rather than individual decisions, as the basic units for describing a conflict. Possible types of social behavior in a conflict are then represented by appropriate solution concepts. The graph model for conflict resolution constitutes a significant reformulation and extension of other existing approaches to the systematic study of strategic conflicts. It has distinct advantages over the other systems, as it possesses a solid theoretical foundation, inherent flexibility, and a comprehensive approach to formally describing strategic behavior. The theoretical background of the Graph Model for Conflict Resolution, and its advantages over other modeling approaches will be described in Chapter 2.

## 1.3 Existing Decision Support Systems for Conflict Resolution

According to Sage [79], "in very general terms, a decision support system (DSS) is a system that supports technological and managerial decision making by assisting in the organization of knowledge about ill-structured, semi-structured issues". Decision support systems (DSSs) have been developed for modeling decision situations involving more than one DM. Close to the context of this thesis are those designed for employment in negotiations, which are also commonly referred to as negotiation support systems. Kilgour *et al.* [55], Thiessen and Loucks [85], and Jelassi and Foroughi [47] provide overviews and comparisons of existing negotiation support systems. Papers describing the theory and application of existing negotiation support systems include contributions by Angus [2], Anson and Jelassi [3], Gauvin *et al.* [30], Jarke *et al.* [46], Nagel and Mills [67], Nunamaker [70], Singh *et al.* [82], and Winter [89]. Radford *et al.* [75] provide an overview of DSSs more specific to the context of conflict resolution.

Conflict resolution methodologies also require implementation algorithms to facilitate their use in practical applications. To permit convenient and expeditious use by practitioners, a given methodology and its associated algorithms should be computerized. In this way, the decision technique is transformed into a realizable decision technology [75].

Software packages have been carried out based on Howard's "metagame analysis" (CONAN [44] and INTERACT [9]), Fraser and Hipel's "conflict analysis" (DecisionMaker [28], SPANNS [65]) and other principles (DSA [64]). GMCR I was the only one based on the Graph Model for Conflict Resolution. It was developed by Fang, Hipel and Kilgour and is included on diskette with their book *Interactive Decision Making: The Graph Model for Conflict Resolution* [20]. All of these software packages are intended to be utilized for aiding decision making under conditions of strategic conflict, which is ill-structured in nature. Therefore, they abide by, less or more, the "most general terms" of DSSs as mentioned in the beginning of this section. A description of these existing DSSs is given as follows.

### 1.3.1 *CONAN*

CONAN (cooperation-or-conflict analysis) [42, 43] is a DSS based on the metagame analysis of Howard [41]. In CONAN, the option form of metagame analysis is employed to interactively model and analyze conflicts. Though the solution concepts of GMR and SMR are used, CONAN does not go immediately to the final equilibrium results by exhaustive stability analysis. After first defining the DMs and options, the user is required to specify a scenario for each analysis. Infeasible moves are specified by the user using "consequence judgement". CONAN keeps track of the preferences of each DM, and the user is required to provide a series of "judgements" during the analysis process. This places a heavy load on user interaction and limits CONAN's ability to analyze large scale conflicts, though more flexibility would be one of its benefits. An off-line graph analysis named "strategic map", though not implemented on computer, is suggested as an important component of CONAN [42].

CONAN was developed in a DOS platform for IBM-compatible PCs. A simple plain text menu is provided as user interface. The keyboard is its only input device. Not being very user-friendly, CONAN identifies its user as "a CONAN expert", working with a client team analyzing a problem.

More recently, the development of another two packages, "Immerse Soap" and "STUDIO" [15] were also announced by Nigel Howard Systems as pilot projects. Immerse Soap was seen by the author more like a illustration package than a software package – no computing component is included. STUDIO was intended to be a Windows based package that would incorporate CONAN, InterAct, and DecisionMaker (the latter two are described below), as well as also drama theory ideas [15]. However, no further information about these two projects is available.

### 1.3.2 *INTERACT*

INTERACT is a DSS that can be used to model and analyze conflicts [8, 9]. The overall objective of the system is to support a flexible modeling methodology. The underlying decision making model is metagame theory [41]. This DSS uses the option form and graphical displays to analyze situations under the control of several interested participants. INTERACT can perform stability analyses for feasible scenarios.

INTERACT is similar to CONAN in terms of their functions, but INTERACT has a graphical user interface, and the "strategic map" has been graphically implemented on-line. INTERACT allows the user to display and work on any part of the model at any time, with opportunities to add commentary on actions, options, scenarios, etc. INTERACT enters preferences in two different ways: one is based upon the number of the options "desirable" or "undesirable" (specified by the user); the other is a purely manual movement of individual scenarios in the tableau.

Running in a "Hyperwindow" environment, INTERACT shows the relevant elements of the model on the screen, together with the available commands. This allows one to manipulate the model easily, while providing a constant visual guide to what is happening. According to the authors [8], MS-Windows was considered as the first alternative platform for INTERACT. It was only because "Hyperwindows" was an ongoing project of their colleagues – who were willing to provide the full source code – that they changed their minds.

INTERACT uses graphical notation to represent "linked issue" or Bennett's Preliminary Problem Structure (PPS), trying to show some recognition of hypergame analysis. But such problem structuring is not incorporated into formal analyses, which still takes account of one issue at a time.

### 1.3.3 *DecisionMaker*

DecisionMaker: The Conflict Analysis Program [29] is a DSS that permits a user to immediately carry out extensive conflict studies. More specifically, it employs the option form and solution concepts of sequential stability for modeling, analyzing and interpreting both small and large conflicts. DecisionMaker handles the removal of logically or preferentially infeasible states for one DM (Type 1 and Type 2 as defined in Chapter 2 of [26]), where the involved options are under the control of a single DM. For a given DM, relative preference can be entered using simple preference statements about the options. Assuming transitivity, an algorithm in the DSS converts the preference statements to a preference ordering of the states.

The specifically designed algorithm and data structure based on the preference tree technique allow DecisionMaker to handle a large number of states efficiently. However, the trade off for using this special data structure includes, feasible states cannot be explicitly listed; state ranking can not be shown which makes the preference specification a blackbox to the user; and some unnecessary restrictions have to be applied to the option statements used for state removal and preference tree construction; etc. DecisionMaker produces sequential stability results that may provide guidance for DMs. A simple status quo analysis is implemented as output interpretation. Another interpretation is finding common options for all the equilibria. Misperception is allowed as a partial implementation of the hypergame method [88].

DecisionMaker is written in the C language for use under the Microsoft Windows operating system on IBM-compatible microcomputers. It has a graphical user interface and is considered to be one of the most successfully designed applications of DSSs for conflict management.

### 1.3.4 *SPANNS*

SPANNS is a proposed DSS for strategic and tactical negotiation support [65]. Besides a rule based system for the tactical component, the strategic support part is based on a conflict analysis model that is planned as an extension of DecisionMaker. Some new features, like hypergame analysis, option-based coalition formalization, and a dynamic factor by a virtual "environment player", were proposed to be integrated into the system. This system was claimed to be under development a few years ago [65], but no operational version is currently available.

### 1.3.5 *DSA*

DSA (Decision Systems Analysis) is a DSS that allows the modeling and solving of a range of games (conflicts of interest) [64]. The underlying methodology is based on a kind of oriented graph ("digraph") instead of trees. Games played on digraph have the flexibility to include cycles that allow the modeling of delaying tactics. Since this DSS uses cardinal payoffs and deals with sophisticated repeated and stochastic games, it might seem to be of little interest here. Nonetheless, it deals with the approach of "Theory of Moves" on $2 \times 2$ games, and orthogonal drawing of directed graphs showing analyses of small size conflicts. DSA is currently intended for research purposes only, and is restricted to small games ($2 \times 2$ and $2 \times 3$).

DSA is implemented on an MS-DOS platform with the keyboard as its sole input device. The author claimed that he was trying to make the next release of DSA available in a MS-Windows environment [64].

### 1.3.6 *GMCR I*

GMCR I was developed by Fang, Hipel and Kilgour and included with the textbook *Interactive Decision Making: Graph Model for Conflict Resolution* [20]. The analysis component of the Graph Model for Conflict Resolution has been well implemented in a powerful engine written in C language. As the first DSS based on the graph model, this DSS reflects major recent achievements in the field of conflict management and can take advantage of the strength of the graph model (refer to Section 2.4). One of the attractive features of GMCR I is its ability to provide stability results for a wide range of solution concepts (refer to Table 2.1). It has been successfully used to analyze a variety of real-world conflicts.

GMCR I emphasizes analysis, and does not support user-system interaction. It requires an ASCII input file containing an available model in a required format, and produces a plain-text output file that can be printed out to show stability results of each state for each DM under a variety of solution concepts. It is basically an analysis program without a modeling component and an interactive interpretation facility. The maximum number of states in a model that GMCR I can analyze is 200 for two-player conflict models, and only 100 for multi-player models.

### 1.3.7 *Summary*

The following conclusions are drawn from the above review:

- DSSs have been at least partially developed for almost all of the conflict resolution methodologies introduced in Section 1.2. The desirability for software tools and decision support systems in this area is obvious.

- Most of the systems with a modeling component use option form, which means option form is well received by this community. Except for the preference tree, no systematic approach for preference elicitation has been presented. More flexible modeling techniques are to be developed.

- Although the listed packages fit into the most general framework of DSS based on what they are intended or claimed to be capable of achieving, most of them do not have comprehensive features for supporting decision making, especially in identifying and implementing preferred alternatives. DecisionMaker seems to be the only operational exception. Still there is a need for improvement on the interpretation side.

- Comprehensive systems should be developed such that different techniques and rationality concepts can be integrated into all the stages of modeling, analysis and interpretation.

- Some of the DSSs (e.g. CONAN and INTERACT) treat analysis and interpretation as the same procedure and have encountered difficulty in dealing with large scale real-world conflicts. The pursuit for performance in analysis in DecisionMaker brings unnecessary restrictions on the modeling. A loose coupling among the components of modeling, analysis and interpretation should be promoted.

- A user-friendly graphical interface is greatly needed to allow non-sophisticated users to utilize the system.

- All the existing and operational systems are PC-based, which is probably because the users of DSSs for conflict resolution are so heterogeneous in their domains of applications that no platform other than a PC is readily common accessible.

- Existing systems, except DecisionMaker, have limitations in handling larger conflict models.
- GMCR I is the only system which implements a wide variety of solution concepts, but also the only system that does not have interactive modeling and interpretation facilities. A real interactive system is needed to enable practitioners to actually appreciate its analysis power and the inherent flexibility of its underlying methodology.

## 1.4 Outline of Thesis

Because strategic is so prevalent within and among organizations both nationally and internationally, the demand for decision support systems to assist decision makers faced with interactive decision problems is increasing.

The objective of this thesis is to provide the next generation of a comprehensive decision support system GMCR II, for systematically studying strategic conflicts, based on the Graph Model for Conflict Resolution paradigm. It is intended to exploit the inherent advantages of its underlying graph model to a great degree, and to constitute a significant improvement over existing DSSs for conflict resolution, in almost all aspects of strategic conflict formulation, analysis and interpretation, and hence become an important tool to assist decision makers or decision analysts in the management of strategic uncertainty.

The rest of this thesis is organized as follows. In the next chapter, the Graph Model for Conflict Resolution, the underlying methodology of GMCR II, is outlined mathematically. Other conflict representation models are also introduced, and the advantages of graph model over other models are explained.

The third chapter gives an overview of the decision support system GMCR II and its development. Subsequently, the design and implementation of this DSS are presented according to its three major components: *Formulation* (Chapters 4), *Analysis* (Chapter 5), and *Output Presentation and Interpretation* (Chapters 6). Related theoretical developments and discussions, which help to make the system more effective in supporting real-world interactive decision making, are also included within appropriate contexts in these three chapters.

The last chapter summarizes the original contributions of this thesis, and suggests directions for future research and development.

# Chapter 2

# Graph Model for Conflict Resolution

The *Graph Model for Conflict Resolution* is a comprehensive procedure for systematically studying real-world disputes [20]. A graph model describes the main characteristics of a strategic conflict in terms of the following key components: decision makers, states, state transitions, and preferences. After developing the graph model, one can use it as a basic structure to extensively analyze the possible strategic interactions among the DMs, in order to identify the possible compromise resolutions or equilibria. A broad range of stability definitions has been defined within the graph model paradigm, allowing it to represent diverse human decision making characteristics, and hence become a truly comprehensive approach to interactive decision making. The output from the stability analysis, as well as related sensitivity analyses can be used, for example, to support decisions made by specific DMs.

The next two sections outline some of the key ideas behind modeling and stability analysis, respectively, in the framework of the graph model for conflict resolution [20]. Subsequently, an illustrative case is presented. Finally, the graph model is compared with other types of conflict models.

## 2.1 Modeling

The Graph Model for Conflict Resolution represents a conflict as moving from state to state (the vertices of a graph) via transitions (the arcs of the graph) controlled by the DMs. A graph model for a conflict consists of a directed graph and a payoff function for each DM taking part in the dispute. Let $\mathbf{N} = \{1, 2, ..., n\}$ denote the set of DMs and $\mathbf{U} = \{u_1, u_2, ..., u_\mu\}$ the set of *states* or possible scenarios of the conflict. A collection of finite *directed graphs* $\{D_i = (\mathbf{U}, \mathbf{A}_i), i \in \mathbf{N}\}$, can be used to model the course of the conflict. The vertices of each graph are the possible states of the conflict and therefore the vertex set, $\mathbf{U}$, is common to all graphs. If DM $i$ can unilaterally move (in one step) from state $u$ to state $u'$, there is an arc with orientation from $u$ to $u'$ in $\mathbf{A}_i$. For each DM $i \in \mathbf{N}$, a *payoff function* $P_i : \mathbf{U} \to \mathbf{R}$, where $\mathbf{R}$ is the set of real numbers, is defined on the set of states. A payoff function measures the worth of states to a DM. It is assumed that values of the payoff function represent only the DM's ordinal rankings of the states, as described in more detail below.

DM $i$'s graph can be represented by $i$'s *reachability matrix*, $\mathbf{R}_i$, which displays the unilateral moves available to DM $i$ from each state. For $i \in \mathbf{N}, \mathbf{R}_i$ is the $\mu \times \mu$

matrix defined by

$$R_i(u,u') = \begin{cases} 1 & \text{if DM } i \text{ can move (in one step)} \\ & \text{from state } u \text{ to state } u' \\ 0 & \text{otherwise} \end{cases}$$

where $u \neq u'$, and by convention

$$R_i(u,u) = 0.$$

A more economical expression of DM $i$'s decision possibilities is his or her *reachable list*. For $i \in \mathbf{N}$, DM $i$'s reachable list for state $u \in \mathbf{U}$ is the set $\mathbf{S}_i(u)$ of all states to which DM $i$ can move (in one step) from state $u$. Therefore,

$$\mathbf{S}_i(u) = \{u' \in \mathbf{U} : R_i(u,u') = 1\}.$$

The payoff function for DM $i$, $P_i$, measures how preferred a state is for $i$. Thus, if $u, u' \in \mathbf{U}$, then $P_i(u) \geq P_i(u')$ iff $i$ prefers $u$ to $u'$, or is indifferent between $u$ and $u'$. When this inequality is strict for all pairs of distinct states for every DM, the conflict is called strict ordinal. Beyond the ordinal information of preference or indifference, nothing can be inferred from the value of $P_i$. For instance, $P_i(u) > P_i(u')$ indicates that $i$ prefers $u$ to $u'$, but the value of $P_i(u) - P_i(u')$ gives no meaningful information about the strength of this preference. For convenience, positive integers are used as the values of $P_i(\cdot)$.

A unilateral improvement from a particular state for a specific DM is any preferred state to which the DM can unilaterally move. To represent *unilateral improvements* (UMs), each DM's reachability matrix can be used to define a matrix

$\mathbf{R}_i^+$. according to

$$R_i^+(u,u') = \begin{cases} 1 & \text{if } R_i(u,u') = 1 \text{ and } P_i(u') > P_i(u) \\ 0 & \text{otherwise} \end{cases}$$

Similarly, DM $i$'s reachable list, $\mathbf{S}_i(u)$, can be replaced by $\mathbf{S}_i^+(u)$, defined by

$$\mathbf{S}_i^+(u) = \{u' \in \mathbf{S}_i(u) : \; R_i^+(u,u') = 1\}.$$

Thus. $\mathbf{S}_i^+(u)$ is called the ***unilateral improvement list*** of DM $i$ from state $k$.

## 2.2 Stability Analysis

The stability analysis of a conflict is carried out by determining the stability of each state for every DM. A state is *stable* for a DM iff that DM has no incentive to deviate from it unilaterally, under a particular behavior model, usually referred to as a stability definition or solution concept. A state is an *equilibrium* or possible resolution under a particular solution concept iff all DMs find it stable under that stability definition.

### 2.2.1 *Overview of Solution Concepts*

In stability analysis, if a DM is able to move away from the state being examined, then what is required is a precise mathematical description of how the value of such a departure is to be measured. A *stability type* or *solution concept* is such a description and is therefore a sociological model of behavior in a strategic conflict. A variety of solution concepts needs to be defined to allow many possible patterns of conflict

Table 2.1: Solution Concepts and Human Behavior

| Solution Concepts | References | Foresight | Disimprovements | Knowledge of Preferences | Strategic Risk |
|---|---|---|---|---|---|
| Nash stability (R) | [68, 69] | Low | Never | Own | Ignores risk |
| General metarationality (GMR) | [41] | Medium | By opponent(s) |  | Avoids risk; conservative |
| Symmetric metarationality (SMR) |  |  |  |  |  |
| Sequential stability (SEQ) | [25, 26] |  | Never | All | Takes some risk; satisfices |
| Limited-move stability ($L_h$, $h > 1$) | [49, 56, 91] | Variable | Strategic |  | Accepts risk; strategizes |
| Nonmyopic stability (NM) | [14, 48, 49, 56] | High |  |  |  |

behavior to be modeled, in order to reflect a wide variety of strategic decision styles, from cautious and conservative to prognosticative and manipulative. In their book, Fang *et al.* [20, Ch.3] define and mathematically compare [20, Ch.5], the graph model solution concepts listed in Table 2.1. Additionally, they demonstrate how graph models can be equivalently expressed using *extensive games*, which are much more complicated and hence not as well suited for practical applications [20, Ch. 4], but do connect the graph model to classical game theory.

The solution concepts provided in Table 2.1 are developed for application to conflicts with two or more than two DMs. The first two columns give the names of the solution concepts, their associated acronyms, and the corresponding original references. The last four columns furnish characterizations of the solution concepts in a qualitative sense, according to the four criteria of foresight, disimprovements, knowledge of preferences and strategic risk. *Foresight* refers to the extent of a DM's ability to think about possible moves that could take place in the future. If the DM has high or long foresight, he or she can imagine many moves and countermoves into the future when evaluating the consequences of an initial move on his or her part. Notice, for instance, that in Nash stability foresight is low, whereas it is very high for non-myopic stability.

The *disimprovements* criterion in the third column refers to a DM's willingness to move to a worse state. A (temporary) move to a less preferred state, in order to reach a more preferred state eventually, is a strategic disimprovement. Disimprovements by opponents are moves by the other DMs to put themselves in worse positions in order to block unilateral improvements by the given DM.

The *knowledge of preferences* column refers to the preference information used in a stability analysis. For example, in a stability analysis under R, GMR or SMR. the preferences of other DMs are not used, although their abilities to move to other states are taken into account. These solution concepts can be quite useful in situations where a decision maker is uncertain about the preferences of his or her competitors. As pointed out in the *strategic risk* column in Table I, a DM who follows GMR or SMR is risk averse and conservative, and hence avoids strategic risk. When a DM follows Nash stability and a state is stable for him or her, he or she has no available unilateral improvements and hence ignores strategic risk. Because the SEQ solution concept has medium foresight, it allows no disimprovements for strategic purposes; preferences of all the decision makers involved are taken into account in the stability calculations. A DM who thinks according to SEQ accepts some strategic risk in searching for "satisficing" [81] solutions, since he or she assumes that any improvement may be selected - decision makers do not necessary acheive the greatest possible improvement. Since limited move and non-myopic stabilities permit strategic disimprovements that will ultimately allow a DM to end up at a more favorable state, these stability types include strategic risk. Under limited move stability, the horizon, $h$, refers to the length of the sequence of moves that a DM can envision beginning at the state being studied for stability. In fact

$L_1$ is equivalent to R and non-myopic stability (NM) is the limit of $L_h$ stability when $h$ approaches infinity.

### 2.2.2 *Unilateral Moves*

For a conflict involving more than two DMs, it is useful to define movements involving more than one DM. Let $\mathbf{H} \subseteq \mathbf{N}$ be any non-empty subset of the DMs, and let $\mathbf{S_H}(u)$ denote the set of all states that can result from any sequence of unilateral moves, by some or all of the DMs in $\mathbf{H}$, starting at state $u$. In this sequence, the same DM may move more than once, but not twice consecutively. If $u' \in \mathbf{S_H}(u)$, let $\Omega_{\mathbf{H}u}(u')$ denote the set of all last players in legal sequences from $u$ to $u'$.

**Definition 2.1** *Let $u \in \mathbf{U}$ and $\mathbf{H} \subseteq \mathbf{N}, \mathbf{H} \neq \emptyset$. A* **unilateral move by H** *from $u$ is a member of $\mathbf{S_H}(u) \subseteq \mathbf{U}$, defined inductively by*

1. *if $j \in \mathbf{H}$ and $u' \in \mathbf{S}_j(u)$, then $u' \in \mathbf{S_H}(u)$,*
2. *if $u' \in \mathbf{S_H}(u), j \in \mathbf{H}$, and $u'' \in \mathbf{S}_j(u')$, then*
   (a) *if $|\Omega_{\mathbf{H}u}(u')| = 1$ and $j \notin \Omega_{\mathbf{H}u}(u')$, then $u'' \in \mathbf{S_H}(u)$ and $j \in \Omega_{\mathbf{H}u}(u'')$,*
   (b) *if $|\Omega_{\mathbf{H}u}(u')| > 1$, then $u'' \in \mathbf{S_H}(u)$ and $j \in \Omega_{\mathbf{H}u}(u'')$.*

If a DM's graph is *transitive*, then for any two consecutive moves be the DM, there is always an equivalent single move available. We have

**Definition 2.2** *Let $u \in \mathbf{U}$ and $\mathbf{H} \subseteq \mathbf{N}, \mathbf{H} \neq \emptyset$. If all DMs' graphs are transitive, a unilateral move by $\mathbf{H}$ from $k$ is a member of $\mathbf{S_H}(u) \subseteq \mathbf{U}$, defined inductively by*

1. *if $j \in \mathbf{H}$ and $u' \in \mathbf{S}_j(u)$, then $u' \in \mathbf{S_H}(u)$,*

2. *if* $u' \in \mathbf{S_H}(u), j \in \mathbf{H}$, *and* $u'' \in \mathbf{S}_j(u')$, *then* $u'' \in \mathbf{S_H}(u)$.

By replacing $\mathbf{S_H}(u), \mathbf{S}_j(u)$, and $\mathbf{S}_j(u')$ by $\mathbf{S_H^+}(u), \mathbf{S}_j^+(u)$, and $\mathbf{S}_j^+(u')$, respectively, in the above definitions, one obtains the definition of a *unilateral improvement (UI)* by **H** when all DMs' graphs are non-transitive or transitive. Refer to [20, Section 3.4] for detailed definitions.

$\mathbf{S_H}(u)$ and $\mathbf{S_H^+}(u)$ can be thought of as **H**'s reachable list and unilateral improvement list, respectively. In particular, the sets $\mathbf{S}_{\mathbf{N}-i}(u)$ and $\mathbf{S}^+_{\mathbf{N}-i}(u)$ represent the possible states of "response sequences" of $i$'s opponents against a move by $i$ to $u$.

### 2.2.3 *Illustrative Definitions of Solution Concepts*

To provide an appreciation of how the graph model can represent the possible strategic interactions among DMs in a strategic conflict, the definitions of the first four solution concepts listed in the left column of Table 2.1 are given next.

**Nash Stability**: *Let* $i \in \mathbf{N}$. *A state* $u \in \mathbf{U}$ *is* **Nash Stable (R)** *for DM i iff* $\mathbf{S}_i^+(u) = \emptyset$.

Under Nash stability, DM $i$ expects that the other DMs will stay at any state $i$ moves to, and consequently that any state that $i$ moves to will be the final state. The initial state $u$ is therefore stable iff $i$ cannot move from $u$ to any state $i$ prefers.

**General Metarationality**: *For* $i \in \mathbf{N}$, *a state* $u \in \mathbf{U}$ *is* **general metarational** **(GMR)** *for DM i iff for every* $u' \in \mathbf{S}_i^+(u)$ *there is at least one state* $u_x \in \mathbf{S}_{\mathbf{N}-i}(u')$ *with* $P_i(u_x) \leq P_i(u)$.

Under general metarationality, DM $i$ expects that the other DMs ($\mathbf{N} - i$) will respond to hurt $i$, if it is possible for them to do so, in any sequence of unilateral moves. DM $i$ anticipates that the conflict will end after $\mathbf{N} - i$ has responded. Additionally, DM $i$'s opponents are assumed to ignore their own payoffs when making their sanctioning moves.

**Symmetric Metarationality**: *For $i \in \mathbf{N}$, a state $u \in \mathbf{U}$ is* **symmetric metarational (SMR)** *for DM $i$ iff for all $u' \in \mathbf{S}_i^+(u)$, there exists $u_x \in \mathbf{S}_{\mathbf{N}-i}(u')$, such that $P_i(u_x) \leq P_i(u)$ and $P_i(u_y) \leq P_i(u)$ for all $u_y \in \mathbf{S}_i(u_x)$.*

The SMR solution concept postulates that DM $i$ expects that he or she will have a chance to counterrespond ($u_y$) to the other DMs' response ($u_x$) to $i$'s original move ($u'$). DM $i$ anticipates that the conflict will end after this counterresponse.

**Sequential Stability**: *For $i \in \mathbf{N}$, a state $u \in \mathbf{U}$ is* **sequentially stable (SEQ)** *for DM $i$ iff for every $u' \in \mathbf{S}_i^+(u)$ there is at least one state $u_x \in \mathbf{S}_{\mathbf{N}-i}^+(u')$ with $P_i(u_x) \leq P_i(u)$.*

The difference between the GMR and SEQ is the requirement of SEQ that any sanction be credible, in the sense that it is a unilateral improvement (or a sequence of unilateral improvements) for the sanctioning DMs.

In the Graph Model of Conflict Resolution, the last two solution concepts listed in Table 2.1, **Limited-move Stability** and **Nonmyopic Statbility**, are defined using the method of ***anticipation*** introduced by Kilgour [49]. It is assumed that a rational player "will choose the alternative which yields the preferred anticipated state." In limited-move stability of horizon $h$, DMs are supposed to be able to foresee a sequence of (maximum possible) length $h$. Nonmyopic stability endows the DMs with sufficient foresight to envision the outcomes of arbitrarily long move-countermove sequences. Detailed definitions for theses two solution concepts and illustrative examples for all the above solution concepts can be found in [20, Ch. 3].

## 2.3 Illustrative Case Study

The Graph Model for Conflict Resolution is now illustrated using a model of a strategic conflict that arose after the discovery of environmental contamination in Elmira, Ontario, Canada. The background of this conflict will now be outlined, and analyses based on the graph model will be described. This conflict was previously studied by Kilgour *et al.* [54]. For additional details about the conflict and the base model, and for references to original sources, see [32]. This case will be used to illustrate the design and implementation of GMCR II in the later chapters. By so doing, the emphasis is to demonstrate how modeling and analysis are conducted in the decision support system, rather than to make any strong claims for this particular model.

Elmira, a town of about 7,500 residents, is located in an agricultural region of southwestern Ontario, roughly equally distant (averaging 75km) from three of the Great Lakes. Municipal water supply is drawn from an underground aquifer. In late 1989, the Ontario Ministry of the Environment [**MoE**] discovered that the aquifer was contaminated by a carcinogen, N-nitroso demethylamine (NDMA). Suspicion fell on the Elmira pesticide and rubber products plant of Uniroyal Chemical Ltd. [**Uniroyal**], which had a history of environmental problems, and was associated with NDMA-producing processes.

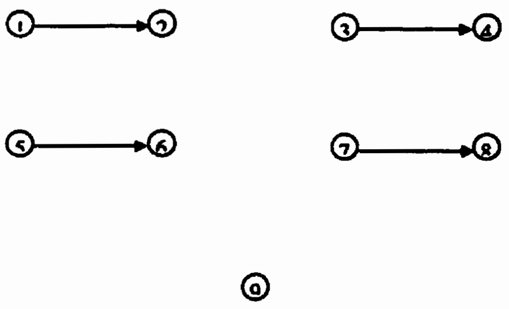
MoE issued a Control Order under the *Environmental Protection Act of Ontario*, requiring that Uniroyal implement a long term collection and treatment system, undertake studies to assess the need for a cleanup, and carry out any necessary cleanup under Ministry supervision. Uniroyal immediately exercised its right to appeal. Meanwhile, various interest groups formed and attempted to influence the process through lobbying and other means. Of particular note was the role of the Regional Municipality of Waterloo and the Township of Woolwich [**Local Government**], which took common positions in the dispute and, encouraged by the Ministry, hired independent consultants and obtained extensive legal advice at substantial cost.

Negotiations involving MoE, Uniroyal, and Local Government began in mid-1991. MoE's objective was to carry out its mandate as efficiently as possible; Uniroyal wanted the Control Order modified or rescinded; Local Government wanted to protect its citizens and its industrial base.

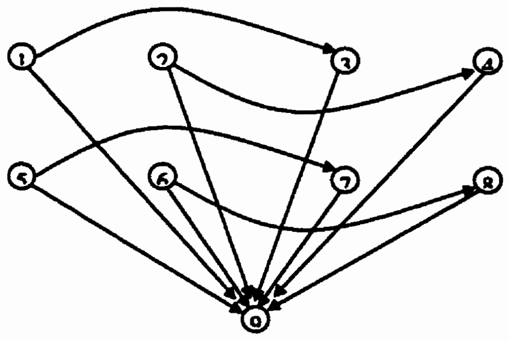
A graph model for these negotiations and the underlying conflict is shown in Figures 2.1 (a), (b), and (c). Figure 2.1 (d) provides an integrated graph for all the three DMs.

To assign meaningful definitions to the states merely represented by numbered nodes in the above graph model (Figure 2.1), an option form of this conflict is used. Explanations and discussions about option form will be given in next section. The three DMs and their possible choices are shown in Table 2.2, and the definitions of the nine feasible states appear in Table 2.3. Note that if Uniroyal abandons its Elmira plant, MoE's and Local Government's choices are irrelevant.

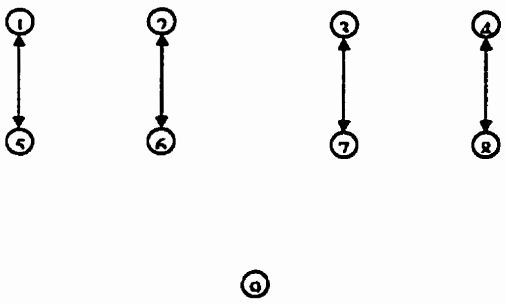
The reachable lists and preferences for each DM (also called the "*analytical representation*" of the graph model, as opposed to the "*graphical representation*" in Figure 2.1) are given in Table 2.5. In mid-1991, the Status Quo was state 1, a



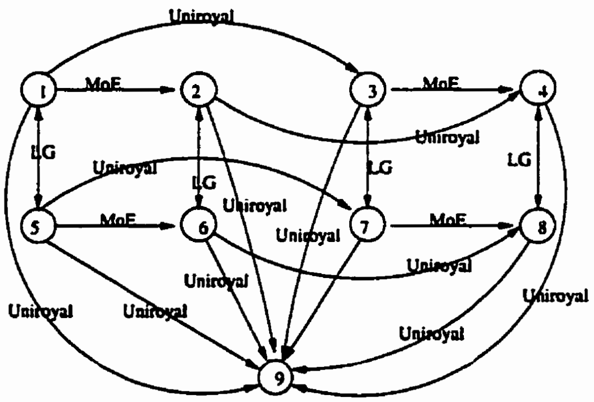
(a) $P_1 = (4\ 3\ \ 8\ 7\ 5\ 2\ 9\ 6\ 1)$



(b) $P_2 = (9\ 2\ \ 4\ 8\ 6\ 1\ 3\ 7\ 5)$



(c) $P_3 = (6\ 2\ \ 8\ 3\ 7\ 4\ 9\ 5\ 1)$



(d) **Integrated graph**

Figure 2.1: Elmira Conflict Model in Graph Form: (a) DM 1 (MoE); (b) DM 2 (Uniroyal); (c) DM 3 (Local Government); (d) Integrated Graph for the 3 DMs

Table 2.2: DMs and Options of the Elmira Conflict Model

| DMs and Options | Interpretation |
|---|---|
| 1. **MoE** | *Ontario Ministry of Environment* |
| (1) Modify | Modify the Control Order to make it more acceptable to Uniroyal |
| 2. **Uniroyal** | *Uniroyal Chemical Limited* |
| (2) Delay | Lengthen the appeal process |
| (3) Accept | Accept the current Control Order |
| (4) Abandon | Abandon Elmira operation |
| 3. **Local Government** | Regional Municipality of Waterloo and Township of Woolwich |
| (5) Insist | Insist that the original Control Order should be applied |

Table 2.3: Feasible States of the Elmira Conflict Model

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1. **MoE** | | | | | | | | | |
| (1) *Modify* | N | Y | N | Y | N | Y | N | Y | – |
| 2. **Uniroyal** | | | | | | | | | |
| (2) *Delay* | Y | Y | N | N | Y | Y | N | N | – |
| (3) *Accept* | N | N | Y | Y | N | N | Y | Y | – |
| (4) *Abandon* | N | N | N | N | N | N | N | N | Y |
| 3. **Local Government** | | | | | | | | | |
| (5) *Insist* | N | N | N | N | Y | Y | Y | Y | – |
| State Number | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |

Table 2.4: Preference Ranking for the Elmira Conflict Model

| | *most preferred* $\longrightarrow$ *least preferred* | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| MoE | 7 | 3 | 4 | 8 | 5 | 1 | 2 | 6 | 9 |
| Uniroyal | 1 | 4 | 8 | 5 | 9 | 3 | 7 | 2 | 6 |
| Local Government | 7 | 3 | 5 | 1 | 8 | 6 | 4 | 2 | 9 |

bargaining impasse with Local Government supporting Uniroyal.

Table 2.5: Reachable Lists ($S_i$) and Payoffs ($P_i$) of the Elmira Conflict Model

| State | MoE | | Uniroyal | | Local Government | |
|---|---|---|---|---|---|---|
| u | $S_1(u)$ | $P_1(u)$ | $S_2(u)$ | $P_2(u)$ | $S_3(u)$ | $P_3(u)$ |
| 1 | 2 | 4 | 3,9 | 9 | 5 | 6 |
| 2 | | 3 | 4,9 | 2 | 6 | 2 |
| 3 | 4 | 8 | 9 | 4 | 7 | 8 |
| 4 | | 7 | 9 | 8 | 8 | 3 |
| 5 | 6 | 5 | 7,9 | 6 | 1 | 7 |
| 6 | | 2 | 8,9 | 1 | 2 | 4 |
| 7 | 8 | 9 | 9 | 3 | 3 | 9 |
| 8 | | 6 | 9 | 7 | 4 | 5 |
| 9 | | 1 | | 5 | | 1 |

The analysis of the conflict using Graph Model for Conflict Resolution finds the equilibria shown in Table 2.6. States 1 and 4 are weak equilibria. and the conflict is unlikely to remain at either for long. The stronger, longer term equilibria occur at state 5. state 8 and state 9.

Historically, Local Government shifted quickly to support the original Control Order, resulting in state 5 for a protracted interval of time. Then MoE and Uniroyal dramatically agreed on a modified version of the original Control Order, thus moving to the equilibrium at state 8. This agreement caught Local Government by surprise; it protested vigorously, but was forced to reach a separate arrangement with Uniroyal. Other relevant background of the Elmira Conflict will be provided

Table 2.6: Equilibria of the Elmira Conflict Model

| State | Equilibrium under |
|---|---|
| 1 | GMR, SMR |
| 4 | GMR, SMR |
| 5 | R, GMR, SMR, SEQ, $L_1 - L_{10}$, NM |
| 8 | R, GMR, SMR, SEQ, $L_1 - L_{10}$, NM |
| 9 | R, GMR, SMR, SEQ, $L_1 - L_{10}$,NM |

*Assume that we only consider limited move stability up to level 10.*

in Sections 5.2 and 5.4.1 when the coalition analysis and sensitivity analyses of this conflict are carried out. Even though GMCR II is capable to handle large or even huge real-world conflict model, this 9-state conflict model is still chosen for as the main illustrative case throughout this thesis. This is because this conflict model is suitable to illustrate a wide range of design features in GMCR II. Another advantage is that information about a conflict model of this size is easier to be presented in same screen.

## 2.4 Comparison with Other Conflict Representation Models

An *abstract game model* conceptualizes a strategic conflict using a formal mathematical structure. Any abstract model attempts to capture the key aspects of a conflict, there by making it easier to understand. A game or conflict model is thus an approximation of reality that systematically structures what are considered to be the most important components of the situation. The conflict model may reinforce the modelers understanding of the conflict. It can also act as a "bookkeeping technique" to help keep track of what is happening [20].

Following are a brief description of representations for strategic conflicts, other than the graph model, that have been developed in the literature.

## Normal Form

The *normal form* of abstract game model was first defined by von Neumann and Morgenstern [86] and is commonly used for describing a conflict in which there are only two players. The basic building blocks for the normal form are the strategies (complete plans of action) for each player, which combine to form states. For games with two DMs, the normal form is often written as a matrix, the rows representing the first player's strategies, and the columns the second player's strategies. Hence, each cell in the matrix represents a state. The actual outcome at the state is not considered important; the normal form displays only preference information, in the form of Neumann-Morgenstern utilities. The normal form is also commonly referred to as *matrix form*. For games with more than two DMs, the normal form is inconvenient, even though theoretically it can be constructed by adding more spatial dimensions.

## Option Form

The format for the conflict model displayed in Table 2.3 is called the *option form* or sometimes the *binary form*, and was originally proposed by Howard [41]. It is used extensively in Metagame Analysis [41] and Conflict Analysis [25, 26] for encoding a conflict model. A game in option form is simply a list of each player's *options* or available courses of action, along with a rule for specifying the payoffs or preferences for each player over the states. To specify a state, the status of every option must be indicated. A "Y" placed beside an option means that the option is taken up by the player controlling it, whereas an "N" indicates that the option is rejected. Any combination of Y's and N's opposite all the options of a given player represents a complete *strategy* for that player. After each player chooses a strategy, the result is a state or outcome. A state thus appears as vector of Y's and N's; each component corresponds to an option in the conflict. While essentially similar to normal form in terms of the kinds of information that can be represented, an option form can conveniently handle games with any finite number of players. Moreover, the option form takes options as its basic building block, and thus is more compact than game models which take strategies (combinations of options) or states (combinations of strategies) as their basic building blocks. For these reasons, option form is capable of representing more complex models in a compact and easily understandable fashion.

### Extensive Form

The *extensive form* is another abstract game model, presented by von Neumann and Morgenstern [86] and refined by Kuhn [62]. In the extensive form, a tree structure is utilized to describe the players' order of choice, and the information available to each player at each choice. Each node in the tree corresponds to an occasion at which one player must act. Each branch from the node represents a possible choice by the player. The extensive form is remarkably flexible, and can depict the flow or evolution of a game and the availability of information.

While it is a powerful tool, the extensive form is not well designed for use in practical applications. The analyst must obtain far too much detailed information about the conflict in order to construct the extensive form for even modestly complex conflict. Another reason why the extensive form is cumbersome to employ in practice is that it requires fixed timing and sequence information on how the game evolves, which may not be available in real-world large-scale interactions. Furthermore, many actual decision problems do not possess definite endpoints that extensive form requires.

In summary, the graph and option forms of the abstract game model work well for models of any size, whereas the normal form is best suited only for models with two DMs, and the extensive form can only be used with fairly simple models.

While more efficient than the extensive form, the graph form is also significantly more flexible than option form and normal form in its capability to model state transitions. It is believed that the information contained in an option or normal form model can be easily transferred into analytical representation of a graph model, but not vice versa. Listed below are some situations for which special forms of state transition information are difficult, if not impossible, to express, or easily ignored, except in graph form.

**Irreversible moves:** Sometimes a DM in a conflict model can cause a conflict to go from state $u$ to $u'$ by a unilateral move, but cannot make the transition back from $q$ to $k$. Irreversible moves exist widely in the real world. In the graph form shown in Figure 2.1, many moves are irreversible, and thus represented by uni-directional arcs. For instance, Uniroyal cannot easily return once it abandons its Elmira operation.

**Common moves:** Sometimes two or more DMs can independently make unilateral moves that cause the model to change from a departure state to exactly the same destination state. Though not very "common", common moves do exist in the real world, *e.g.* Fang *et al.* give an example of common moves in a simplified model of a superpower nuclear confrontation [20, Ch.2].

**Forcing moves:** When a DM makes a certain move, one or more other DMs may have no choice but to make a forced response; thus the original DM achieves a new state "unilaterally". The concept of forcing move was first identified in the proposal of this thesis. An example of a forcing move appears in a model in [35], where one DM (government) forces another DM (industry) to stop a project by denying the license. Though not identified by the authors, another example can also be found in [33].

**Intransitive Moves:** Option form and normal form always assume the transitivity of moves: for any sequence of moves in which one DM moves twice consecutively, there is always an equivalent single move available. Some useful sequential information may be lost by using this assumption. The graph model does not necessary take this assumption, and hence can be applied to more general situations.

The stability analysis for a conflict model utilizes only states, state transitions, and the DMs' preferences over the states. In other words, stability analysis takes place at the state level. Consequently, the graph form of conflict model, with states as its basic building blocks, is especially suitable for use at the stability analysis stage. However, despite its flexibility on state transitions mentioned above, the graph form also has limitations.

First, a state in graph form is identified only by number and displayed as a numbered node, which would seem meaningless to a practitioner. In [20], this problem is solved by employing option form to define the states. Secondly, even though it is much more efficient than extensive form, the graph form does require considerable effort to construct a graph model by directly specifying the states, state transitions and preferences for a medium or large conflict model. The decision support system GMCR II presented in this thesis solves this problem by taking advantage of the high efficiency of the option form in the modeling stage. The option form is utilized to automatically generate the information on feasible states, allowable state transitions, and relative preferences of each DM. This information actually forms the analytical representation of the graph form, which is then used at the analysis stage under the paradigm of the Graph Model for Conflict Resolution. In this way, the option form does not replace, but rather enriches the graph model. Obviously, to successfully fulfill its role in a decision support system, the option form needs to overcome the drawbacks pointed out above in state transitions representation. As will be seen later, one of the main contributions of this thesis is the improvement of option form to enable it to capture key components of a conflict, including state transitions, and to do so in harmony with the graph model. In this thesis, up to Chapter 3, "option form" refers to the option form as described by Howard [41], and Fraser and Hipel [26], which was popularly used before this research. Starting with Chapter 4, however, this term refers to the "improved" option form.

# Chapter 3

# Overview of the Decision Support System GMCR II

## 3.1 Applicability Contexts for GMCR II

There are many situations in which a DSS for conflict resolution, such as GMCR II. can be useful. They include:

1. *A decision maker analyzes a strategic conflict in which he or she is a participant.*

2. *A consultant advises a decision maker.*

   In the above two cases, strategic interactions following the focal participant's actions can be analyzed, and the consequence of certain strategies estimated, in order to improve the participant's position.

3. *An interested third party analyzes a dispute in which he or she is not a decision maker.*

An analyst can utilize GMCR II by using various evolution of a conflict and to estimate, say, what the preferences must have been to produce the observed outcome. The analyst can also study how the structure of the conflict influenced behavior, thereby identifying better ways to structure a future conflict.

4. *A facilitator uses a mediation tool to coordinate information among the actual decision makers, and assess possible compromises.*

   A mediator can utilize GMCR II as communication and analysis tool to estimate possible outcomes by using various preference rankings, without revealing (or knowing) which one correctly describes the participants. This might identify options that are detrimental, irrelevant, or beneficial to all parties.

   An especially useful setting is a meeting of a subset of the actual decision makers. Based on their common assumptions about others, the decision problem can be simulated and potential agreement (collective improvement) can be made among participants.

5. *An analyst conducts simulation studies in which interested participants play the roles of decision makers involved in a conflict.*

6. *It has been difficult to study large scale conflict problems due to the lack of suitable computerized systems. The use of GMCR II as a research tool will fill the gap.*

As can be seen from the above, an intended user of GMCR II:

- can come from any discipline that deals with conflict;
- is not necessarily a sophisticated computer user;

- is not necessarily a frequent user of the system;
- is not necessarily a professional conflict specialist, and therefore may have only basic knowledge of conflict resolution and the graph model.

Moreover, decision environments of real-world conflicts are usually at most partially structured and hence only partly computable, so the user's judgement is a very important part of the decision process. Therefore, user interaction and user centered design are of extreme importance for this DSS. As well, the system must be portable and require no unusual hardware and software support so as to be accessible to a wide variety of intended users.

The versatile decision environments and user profile decide that GMCR II has to be an adaptive system [83], and continuous improvement to the system is desirable. GMCR II is expected to effectively assist users in all phases of modeling, analysis, and interpretation of strategic conflicts.

## 3.2 The GMCR II Framework

The structure of GMCR II is depicted in Figure 3.1. The system comprises a modeling subsystem, an analysis engine, an output interpretation subsystem. The modeling subsystem receives user input via the user interface, processes the input and automatically generates an analytical graph model which can be accepted by the analysis engine. Modeling information, such as DMs and options, feasible state list, reachable lists and preference rankings can be also conveyed to the user interface, making the modeling itself a genuine interactive process. The analysis engine thoroughly analyze the stability of every applicable types on every state for every DM. Stability results are then stored in an efficient and easy-to-retrieve bit-wise structure which is then maintained by the output presentation and interpretation system. The output interpretation system coordinates the display of every aspect of the stability result based on the user's requests via the user interface. Requests for additional analyses can also be directed to analysis engine; the relevant output is then presented to the user via the output interpretation subsystem. A loose coupling, as a good software engineering practice, is maintained among the subsystems.

Comparing the GMCR II structure with the typical DDM (dialog, data and modeling) paradigm of DSS as suggested by Sprague and Carlson [83] and Sage [79], it seems that the DBMS (data base management system) [79], which would be the most important part of a traditional DSS, is missing. The author believes that it is because the special nature of the DSSs for conflict resolution including GMCR II. The versatile decision environments, as outlined in last section, determine that GMCR II cannot be designed as a *specific DSS* [83, 79]. The three technology levels of DSS, according to Sprague and Carlson [83], are "specific DSS", "DSS generator" and "DSS tools". GMCR II is a special form of DSS that can be best described as one between a DSS generator and a specific DSS. It does not have particular domain of application, and does not possesses a pre-existing data-base. It is only when it is applied to a particular strategic conflict, and popularized with the information obtained from the case, that it actually serves as a specific DSS. In that case, the original information and the rich amount of additional information resulting from the extensive analyses, can be envisioned as a data-base, and the output presentation and interpretation subsystem a management tool of the data.

GMCR II has been developed in the Microsoft Windows environment, first in 16-bit Windows, then in 32-bit Windows. Borland C and Windows API [72] were the initial development tools. The recent versions were designed using the object-orient approach and implemented in Microsoft Visual C++ utilizing Microsoft Founda-

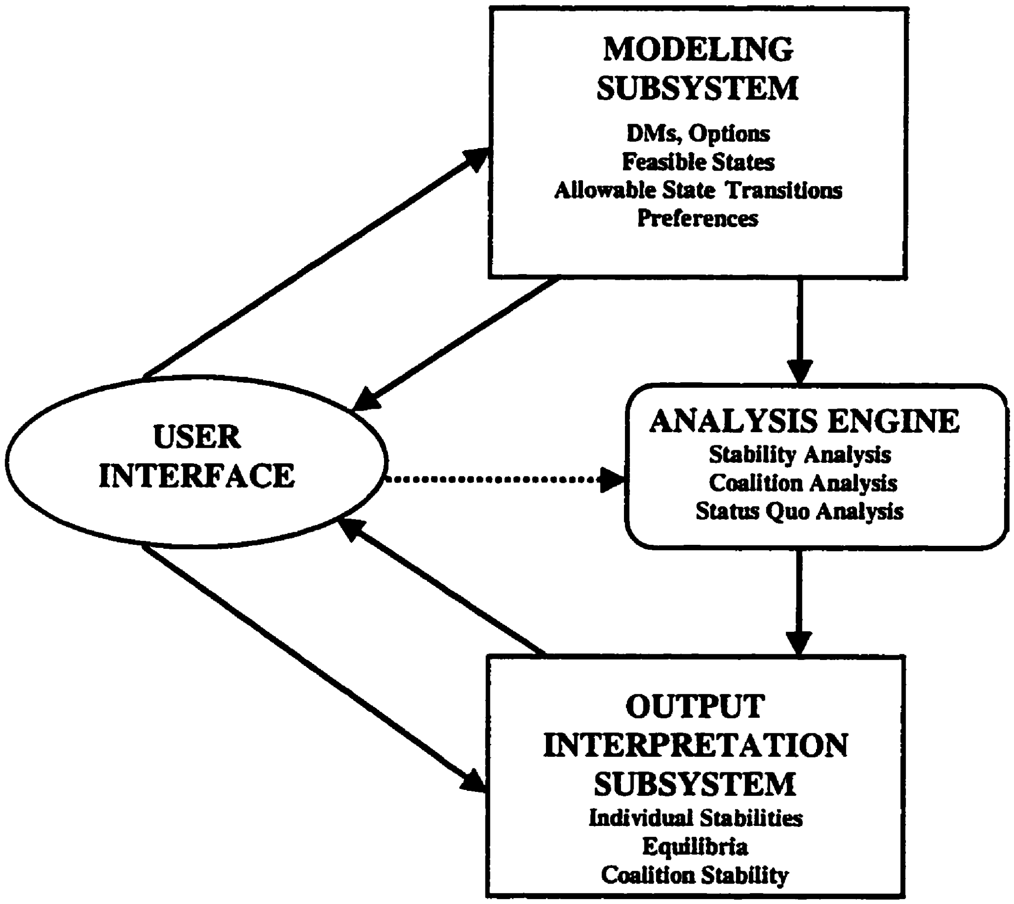


Figure 3.1: GMCR II Structure

tion Classes (MFC). The object-oriented design of GMCR II adopts the popular "Model-View-Controller" architecture, or MVC, which was initially developed by the SmallTalk developer community [90]. MVC suggests that a typical architecture will have three main components: a group of classes and objects that model the underlying application itself; a group of classes or objects that provide a human-interface view of those model-related classes; and a group of classes and objects that control, or synchronize, the behavior of others. Reflecting the context of MFC, the programming model separates the data from the display of the data, and from most user interaction with the data. GMCR II distinguishes two different types of user interfaces. Dialog boxes are used for input that can change the model itself. Property pages, which form a property sheet occupying the client area of the document window, are used strictly for display; they cannot modify information about the model. The advantage of the separation of data, display, and data interface is versatility in viewing the data. This feature is exactly what GMCR II, especially its output presentation and interpretation, needs.

## 3.3 User Interface of GMCR II

Figure 3.2 shows the main frame window of GMCR II. As can be seen, the view, or the client area of this Windows application, is occupied by a property sheet [73], which consists of eight different property pages. The first page, upon the opening of a particular document, will guide the user through the relevant sequences of the operation of this system. The pages labeled "Decision Makers and Options", "Feasible States", "Allowable Transitions", and "State Ranking" enable the user to view the modeling information resulting from his or her own input specifications. This makes the modeling procedure interactive. The last three pages are applicable after the stability analysis of the model. They enable the user to view every aspect of the output information in a variety of helpful formats, from which desirable follow-up analysis can be requested to generate structural insights and decision advices about the conflict model.

These property pages form the display part of the user interface. The use input in these pages only controls how or what aspects of the information is to be displayed; there is no way to alter the established conflict model itself. The input information about the model is elicited through a series of dialog boxes that can be invoked via the menu. Detailed features about the major input dialog box will be described in relevant sections of the next three chapters.

## 3.4 Modeling Subsystem of GMCR II

The modeling sub-system formulates the strategic conflict based on the user's input. In the current version of GMCR II, an improved option form is used to represent the conflict model. An alternative graph-based modeling approach, is proposed at Chapter 4 with preliminary design ideas. The modeling procedure of GMCR II consists of three major steps:

- the generation of feasible states,
- the calculation of allowable state transitions, and
- the elicitation of preference information.

The information gathered through this three steps forms the three major components of an "analytical graph model", which is used as the input to the analysis engine.

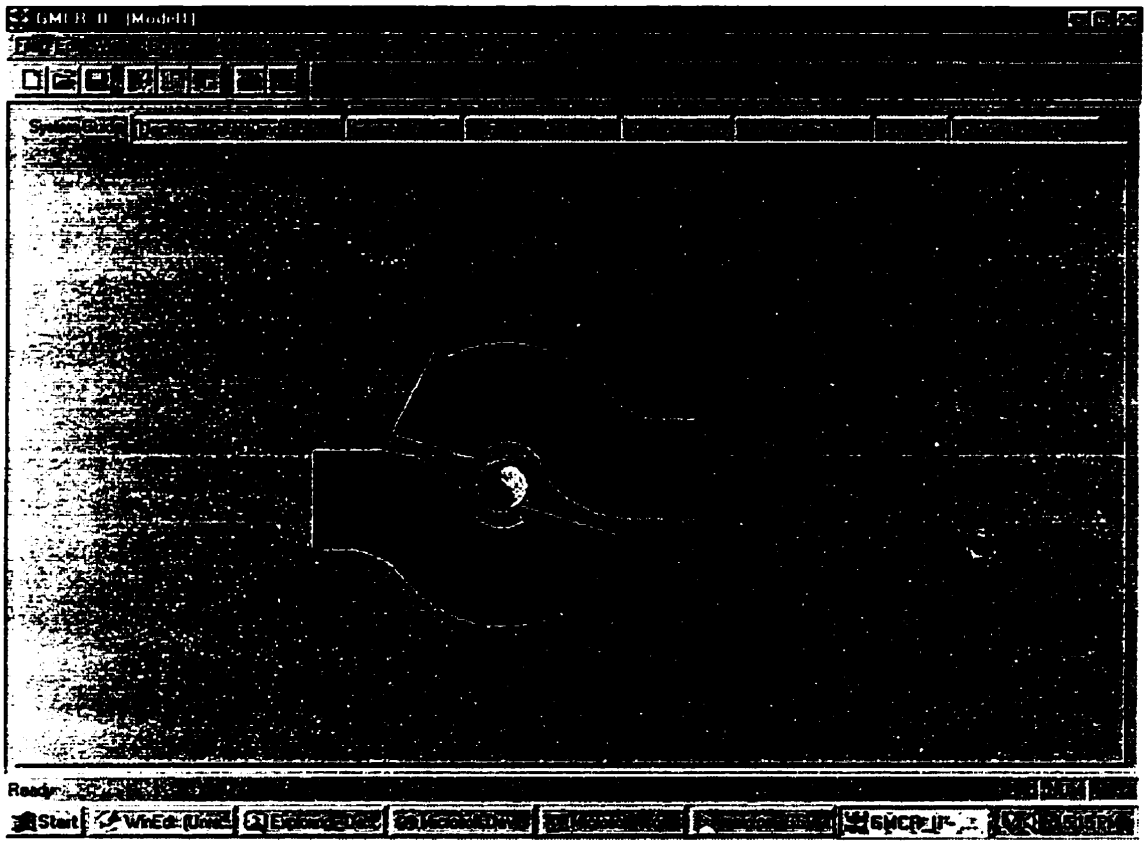


Figure 3.2: The GMCR II Frame Window

## 3.5 Analysis Engine of GMCR II

The analysis engine performs a thorough stability analysis on the conflict model. The output is the stability result on every state, for every DM, under every solution concept as listed in Table 2.1. This large amount of data facilitates the user's various needs in the interpretation stage.

Requests for various forms of follow-up analyses usually arise when or after the user examines different aspects of the stability output. A wide range of possible follow-up analyses, including coalition analysis, status quo analysis, sensitivity analysis, hypergame analysis, and dynamic analysis, are discussed in Chapter 5.

It is believed that in a DSS, the consideration of efficiency is usually secondary to effectiveness [79]. This is an important guideline to the development of DSSs including GMCR II. However, due to the possible combinatorial complexity that could result from the option form representation, and the high frequency of user-system interactions (the effect of which can be affected by any perceivable delay), efficiency, in the case of GMCR II, is itself an important factor of effectiveness. Therefore, a considerable amount of effort was devoted to the improvement of efficiency for various algorithms.

## 3.6 Output Presentation and Interpretation Subsystem of GMCR II

Conflict resolution techniques can mainly be described as descriptive techniques in that they describes a variety of possible compromise resolutions (equilibria) as well as the various social interactions that can cause these equilibria to take place.

GMCR II is intended to be a prescriptive tool [79] that can also advice a decision maker how to optimize his choice of strategies in order to reach his or her most preferred equilibrium within the social constraints of the conflict. The output presentation and interpretation subsystem is the major component that enables GMCR II to have normative [79] function.

In the output presentation and interpretation subsystem, various aspects of the output data are presented to the user via carefully designed user-interface. The innovative output presentations gave the user the idea what optimal equilibrium would be. Vigorous follow-up analyses, especially the status quo analysis facility, provide significant assistance for the user to utilize his/her judgement to rule out infeasible predictions and work out strategic plan for the implementation of the chosen objective.

## 3.7 Validation and Testing of GMCR II

An iterative design approach [83] was adopted in the development of GMCR II. This approach is similar to "prototyping" in that many versions of the system were developed, but it differs in that each version, including the initial one, was real, live, and usable, not just a pilot test. Considerable effort was devoted to validation and testing during this iterative design and development process. The main components of this effort are outlined below.

- GMCR II was employed to model and analyze a range of real-world applications, many of which are documented in a variety of publications, including
  - Elmira groundwater contamination dispute [22, 58, 57, 61], which is used as the main illustrative case throughout this thesis;

  - Flathead river resource development conflict [35, 37];
  - Prijedor refugee return conflict in Bosnia, including analysis of the implications of several hypotheses about the objectives of one DM [60];
  - Softwood lumber dispute between Canada and U.S.A. (phases I [38] and II [39]);
  - Garrison Diversion Unit dispute [36, 21, 71];
  - Cuban missile crisis [59];
  - Trade-in-Service conflict [40];

  In these applications, results were compared with previous analyses wherever feasible. The results produced by GMCR II were always found to be correct.

- Functional and structural tests were conducted on many abstract games, including some classical games for which well-verified stability results are available, such as

  - Chicken (both original [76] and modified [20] versions);
  - Prisoner's Dilemma [77];
  - Other 2 X 2 games (*e.g.* numbers 52 and 70) in Rapoport and Guyer's listing [76].

- Various versions of GMCR II were used by students in at least 60 group projects for the course ***SD533 Conflict Analysis***, and for six undergraduate workshops, in the Department of Systems Design Engineering, University of Waterloo, during 1996-1998. The student feedback was very helpful, and contributed substantially to the features and usability of GMCR II.

- Demonstrations of GMCR II were provided to potential users including management consultants, national defence and peace-keeping personnel, and conflict resolution researchers and practitioners. These demonstrations were often followed by valuable discussions, and sometimes problems for analysis were suggested by the audience. Evaluations were generally favorable and encouraging.

- To test the system's performance on large-scale models, GMCR II was applied to the largest documented conflict model – the Trade-in-Service Conflict [34]. This model has six DMs, 20 options, and 184,320 feasible and distinguishable states. It was analyzed using DecisionMaker for the solution concepts Nash and SEQ only [34, 40]. On a personal computer with Pentium II 266 mHz processor and 96 mb RAM, GMCR II took less than 30 seconds to generate the list of feasible and distinguishable states, about 10 minutes to calculate the preferences of all 6 DMs, and on the order of 70 minutes to obtain Nash and SEQ stability results. GMCR II's results were identical to those calculated by DecisionMaker, and it is believed that GMCR II's speed and other aspects of its performance are superior. Consequently, there is every reason to believe that GMCR II's performance will be suitable for consulting, even when very large scale conflict models are required.

# Chapter 4

# Formulation of Strategic Conflicts

The modeling stage is the problem-structuring phase of a conflict study. In many applications, significant insights are already gained at the modeling stage, before an analysis is even executed.

GMCR II employs the option form as the primary tool to input strategic conflict models. The option form is especially useful when the analyst can focus on the specific courses of action, or options, that are available to each DM. Because states are represented as combinations of options, the number of states is exponentially increasing in the number of options, making the option form very efficient, especially for large models.

The analytical representation of a graph model has four components, namely the list of DMs, the list of feasible states, allowable state transitions for each DM, and the ordinal preferences of each DM over the feasible states. In this chapter, the use of option form in GMCR II to specify the necessary information for the last three modeling components is shown. The option form is formally defined, and equipped with efficient techniques to carry out appropriate tasks and operations.

Improvements are introduced to overcome drawbacks on state transitions outlined in Section 2.4. Moreover, three approaches are designed for ordinal preference elicitation and representation. Since the modeling information gathered in option form constitutes an analytical representation of a graph model and determines the state-based input to the stability analysis, the option-based formulation of conflict does not limit, but rather facilitates the application of the Graph Model for Conflict Resolution to a strategic conflict. A graph-based model (in other words, a model built directly upon a graphical representation), would also fully exploit the flexibility of the graph model, and would be suitable for smaller models. But due to the lack of an appropriate graph-drawing development tool in the Windows platform, no graph-based modeling has been implemented in the current version of GMCR II. Nonetheless, some design ideas for graph-based modeling are presented in the last section of this chapter.

## 4.1 Option Form

The set of *decision makers* in a strategic conflict can be denoted by

$$\mathbf{N} = \{1, 2, \ldots, i, \ldots, n\},$$

where $n = |\mathbf{N}| \geq 2$. Let the set of *options* of decision maker $i \in \mathbf{N}$ be

$$\mathbf{O}_i = \{o_1^i, o_2^i, \ldots, o_{m_i}^i\}.$$

Then the set of all options in the conflict model is

$$\mathbf{O} = \bigcup_{i=1}^{n} \mathbf{O}_i$$

Here the index $i$ indicates which decision maker controls an option. Under some circumstances, this index can be suppressed, and the available option set can be denoted as

$$\mathbf{O} = \{o_1, o_2, \ldots, o_m\},$$

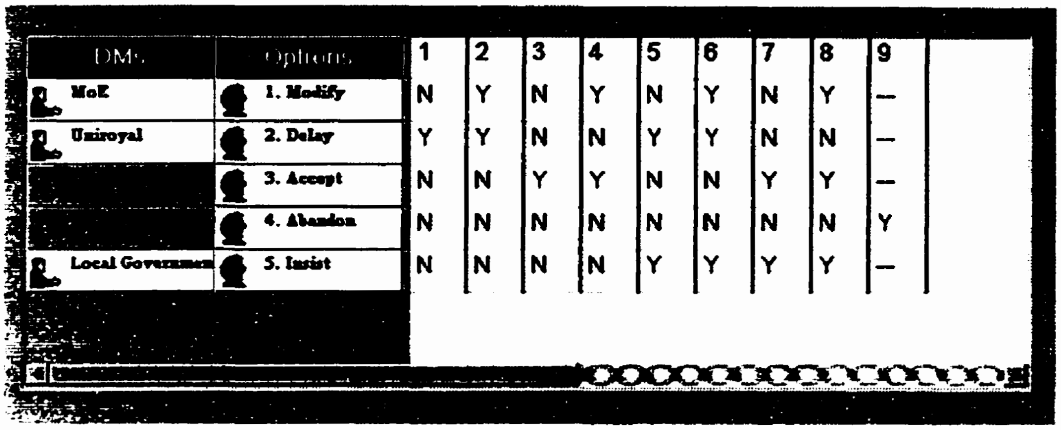
where $m = \sum_{i=1}^{n} m_i$ is the ***total number of options***. A ***state*** $u$ can be defined as a mapping

$$\begin{aligned} u: \mathbf{O} &\longrightarrow \{0,1\} \\ o_j^i &\longmapsto u(o_j^i) = \begin{cases} 1 & \text{if DM } i \text{ selects option } o_j^i \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

Therefore, the set of all ***mathematically possible states*** in a conflict model is $\{0,1\}^{\mathbf{O}}$. Since $\{0,1\}^{\mathbf{O}}$ is isomorphically equivalent to the power set $2^{\mathbf{O}}$, every state $u$ can also be equivalently expressed as a subset of $\mathbf{O}$, for which the mapping $u$ is the characteristic function. Obviously, the total number of mathematically possible states is $2^{|\mathbf{O}|} = 2^m$. In practice, however, only a small portion of mathematically possible states may be feasible due to various possible option constraints. Section 4.3 explains how GMCR II identifies and removes infeasible states. The set of all ***feasible states*** is denoted as $\mathcal{U} \subseteq 2^{\mathbf{O}}$.

In practice, each state can be represented by a Y-N column indicating which available options are selected (denoted by Y for yes) or not taken (denoted by N for no). For example, Figure 4.1 shows the list of feasible states that GMCR II generated for the Elmira conflict model, described in Section 2.3.

In Figure 4.1, each column of Ys and Ns represents a feasible state. Clearly, each column, except the right-most, corresponds to a particular mapping from $\mathbf{O}$ to $\{0,1\}$, and is thus an equivalent representation of a state, as defined earlier. The right-most column stands for a single state that represents a group of formally distinct but practically "indistinguishable" mappings, and is discussed in detail in Section 4.3.3.



| DMs | Options | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| MoE | 1. Modify | N | Y | N | Y | N | Y | N | Y | — |
| Uniroyal | 2. Delay | Y | Y | N | N | Y | Y | N | N | — |
| | 3. Accept | N | N | Y | Y | N | N | Y | Y | — |
| | 4. Abandon | N | N | N | N | N | N | N | N | Y |
| Local Government | 5. Insist | N | N | N | N | Y | Y | Y | Y | — |

Figure 4.1: Displaying Feasible States

## 4.2 Implementation of Option Form in GMCR II

The option form can present problems in memory and execution time. For example, INTERACT [8] (as mentioned in Section 1.3) has difficulty obtaining a list of all feasible scenarios, even in models with only 10 options. Fraser and Hipel [27] have devised algorithms and data structures for DecisionMaker based on a binary tree structure that produces significant savings in memory and execution time in a stability analysis using the solution concept of SEQ, but the trade-off is that only one solution definition is applicable. Moreover, the list of feasible states and the preference rankings cannot be shown. In GMCR II, data structures and related algorithms are carefully designed to allow efficient execution of modeling operations.

### 4.2.1 *Data Structure*

For each feasible state $u \in \mathbf{U}$, a unique integer $b(u)$ can be defined as

$$b(u) = \sum_{i=1}^{m} u(o_i) \cdot 2^{i-1}$$

For example, let $u$ be state 1 of the illustrative model shown in Figure 4.1, then we have

$$b(u) = 0 \cdot 2^0 + 1 \cdot 2^1 + 0 \cdot 2^2 + 0 \cdot 2^3 + 0 \cdot 2^4 = 2 \ (decimal)$$

If represented in binary format,

$$b(u) = \overbrace{\cdots\cdots \underbrace{0}_{DM_3}\ \underbrace{001}_{DM_2}\ \underbrace{0}_{DM_1}}^{32\ bits} \ (binary)$$

The brackets indicate which of the three DMs controls the option. Each bit in the applicable range equals 1 or 0 to indicate whether the option is selected by the decision maker controlling it, or rejected, respectively. GMCR II uses a 32-bit DOUBLEWORD to represent the specific option selection defining a state. Because there are 32 bits, this format can handle up to 32 options, which seems more than sufficient for real-world applications. This straightforward data representation, combined with the bit-wise operation features of the $C/C^{++}$ language, constitutes a basis for a range of efficient algorithms in GMCR II. Detailed discussion of each algorithms is located in the section to which it is most relevant.

### 4.2.2 *Pattern Matching*

Often a group of states with some common characteristics must be identified from the model. In GMCR II, "common characteristics" refers to a partial specification of the options being taken or not taken, called a "pattern". A *pattern p* can be defined as a mapping

$$\begin{aligned} p: \mathbf{O} &\longrightarrow \{0,1,2\} \\ o^i_j &\longmapsto p(o^i_j) = \begin{cases} 1 & \text{if decision maker } i \text{ selects option } o^i_j \\ 0 & \text{if decision maker } i \text{ does not select option } o^i_j \\ 2 & \text{option } o^i_j \text{ may or may not be selected} \end{cases} \end{aligned}$$

In many operations in the modeling stage, it is important to determine whether a state "matches" a particular pattern, or whether the state is one of the states specified by that pattern. A state $u$ is said to *match* a pattern $p$, denoted as $u \models p$, iff

1) $u^{-1}(\{1\}) \supseteq p^{-1}(\{1\})$, and

2) $u^{-1}(\{0\}) \supseteq p^{-1}(\{0\})$

or equivalently,

1) $\forall o_i \in \mathbf{O}, p(o_i) = 1 \Rightarrow u(o_i) = 1$, and

2) $\forall o_i \in \mathbf{O}, p(o_i) = 0 \Rightarrow u(o_i) = 0$.

In practice, a pattern is most easily expressed in a binary format. For instance,

$$\cdots\cdots 00 - -1 \ \text{(or } NN - -Y \text{ when exposed to users)}$$

is a pattern representing those states in which option 1 is chosen and options 4 and 5 are not. In other words, a pattern in binary format consists of 1's (Ys), 0's (Ns), and −s. The dash "−" indicates that the entry can be either a 0(N) or 1(Y). States 2 and 4 in Figure 4.1 match the above pattern.

A pattern cannot be directly recorded in a simple data structure. However, a pair of *masks*, each of which is essentially a DOUBLEWORD and thus very easy to store or manipulate, can be defined to equivalently represent a pattern in an extremely efficient manner. Recall that

$$sgn(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ -1 & \text{if } x < 0 \end{cases}$$

For any pattern $p$, define

$$m_0(p) = \sum_{i=1}^{m} sgn(p(o_i)) \cdot 2^{i-1}$$

as the *zero-mask* of $p$, and

$$m_1(p) = \sum_{i=1}^{m} (1 - \mid 1 - p(o_i) \mid ) \cdot 2^{i-1}$$

as the *one-mask* of $p$. For example, for the above-mentioned pattern with binary format $\cdots\cdots 00 - -1$, the two masks are

$$m_0(p) = 1 \cdot 2^0 + 1 \cdot 2^1 + 1 \cdot 2^2 + 0 \cdot 2^3 + 0 \cdot 2^4 = \cdots\cdots 00111 \ (binary)$$

$$m_1(p) = 1 \cdot 2^0 + 0 \cdot 2^1 + 0 \cdot 2^2 + 0 \cdot 2^3 + 0 \cdot 2^4 = \cdots\cdots 00001 \ (binary)$$

As can be seen, $m_0(p)$ retains the 0-bits of $p$ with all other bits set to 1, while $m_1(p)$ retains the 1-bits of $p$ with all other bits set to 0. The pair of masks is "equivalent" to the pattern because

**Proposition 4.1** *A state $u$ matches pattern $p$: $u \models p$, iff*

$$1) \quad b(u) \mid m_0 \ = m_0, \text{ and,} \tag{4.1}$$

$$2) \quad b(u) \,\&\, m_1 \ = m_1, \tag{4.2}$$

*where " | " and " & " are bit-wise OR and bit-wise AND, respectively.*

Because of the $C/C^{++}$'s capacity of bit-wise operation, the this pattern-matching test is very efficient.

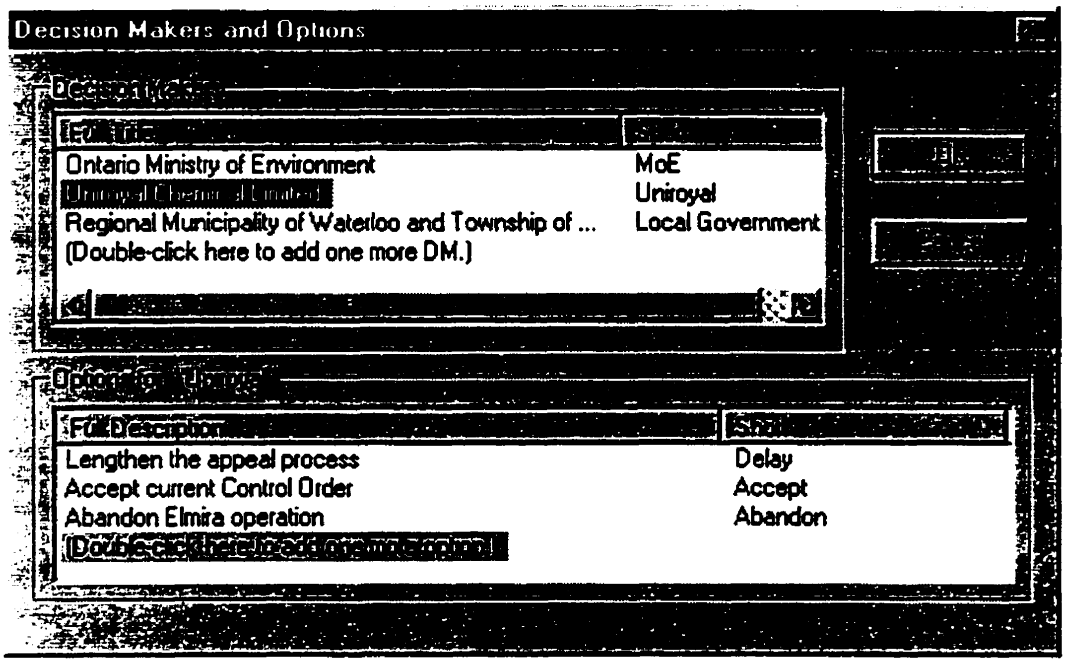
## 4.3 Scenario Generation and Reduction

### 4.3.1 *DMs and Options*

To formulate a conflict in option form, one must first identify the DMs in the model and the options, or courses of action, that fall under the control of each one. Figure 4.2 shows the GMCR II dialog box for entering DMs and options, where "Uniroyal Chemical Limited (Uniroyal)" under Decision Makers is highlighted; accordingly, the lower area contains Uniroyal's options. This dialog allows the user to input a full title and a short title for each DM, as well as a full description and a short description for each option. From this dialog box, a user can

- *add* a DM, or an option controlled by the highlighted DM, into the model, by double-clicking the last item on the corresponding list;
- *modify* an existing DM name or option description, by double-clicking on that item;
- *remove* a highlighted DM or option from the model, by pressing the "Delete" key on the keyboard.

Note that when a DM is removed, the options belonging to this DM are removed automatically. While this dialog box is active, a simple description of how to modify and remove items appears on the status bar located at the bottom of the GMCR II main window, which is not shown in Figure 4.2.


Decision Makers and Options

Decision Makers

| Full Title | Short |
|---|---|
| Ontario Ministry of Environment | MoE |
| Uniroyal Chemical Limited | Uniroyal |
| Regional Municipality of Waterloo and Township of ... | Local Government |
| (Double-click here to add one more DM.) | |

Options for "Uniroyal"

| Full Description | Short |
|---|---|
| Lengthen the appeal process | Delay |
| Accept current Control Order | Accept |
| Abandon Elmira operation | Abandon |
| (Double-click here to add one more option.) | |

Figure 4.2: The DM/Option Input Dialog Box

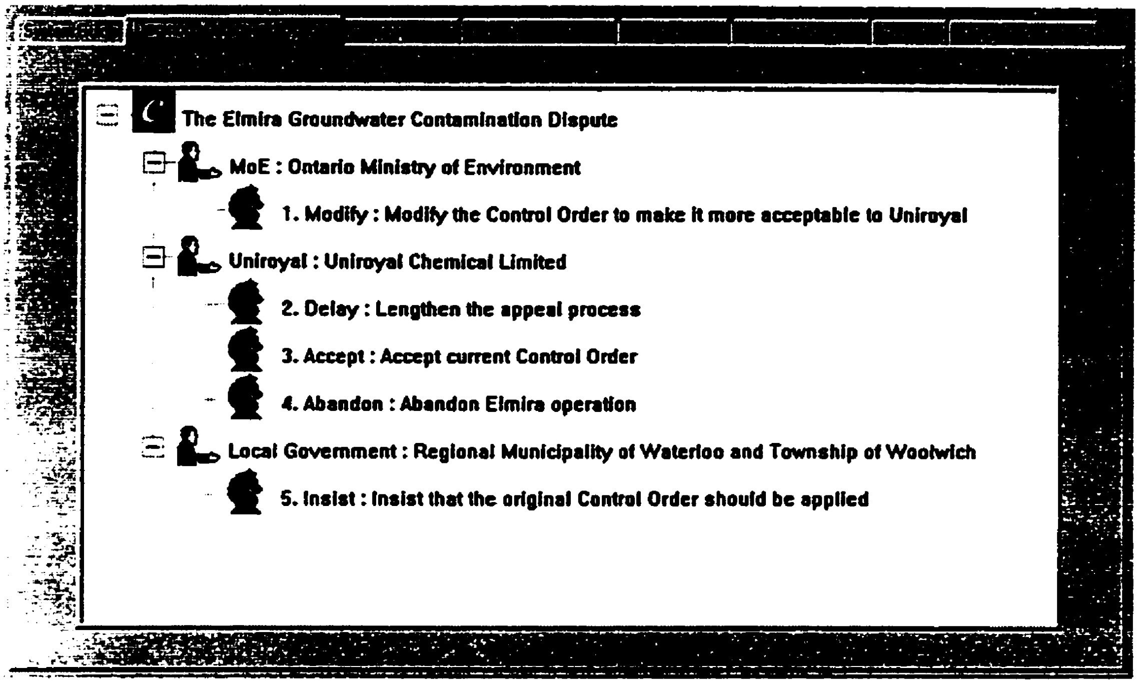

The Elmira Groundwater Contamination Dispute
MoE : Ontario Ministry of Environment
1. Modify : Modify the Control Order to make it more acceptable to Uniroyal
Uniroyal : Uniroyal Chemical Limited
2. Delay : Lengthen the appeal process
3. Accept : Accept current Control Order
4. Abandon : Abandon Elmira operation
Local Government : Regional Municipality of Waterloo and Township of Woolwich
5. Insist : Insist that the original Control Order should be applied

Figure 4.3: Displaying DMs and Options in the Elmira Conflict Model

The input from the Decision Makers and Options dialog box is read and stored by the system. For simplicity, GMCR II uses only the short title or short description to indicate a DM or option; longer titles and descriptions are kept as a reference. A user can refer to this information at any later time by activating the property page tabbed by "DMs and Options", as shown in Figure 4.3. In this page, a tree view displays all the DMs, all the options, and their relationships.

The dialog box to input DMs and options is invoked via the menu "Modeling | States | Generate Feasible ...". The necessary information is essentially the number of DMs involved in the conflict model, $n$, and number of options each DM controls:

$m_i, i = 1, 2, \cdots, n$, plus labels. These are the necessary input for the generation of all mathematically possible states and the determination of allowable state transitions, as will be discussed in Sections 4.3.4 and 4.4.

### 4.3.2 *Infeasible State Removal*

Since each option can be either selected or not selected, $m$ options imply a total of $2^m$ mathematically possible states, constituting the set $\{0, 1\}^{\mathbf{O}}$ . In practice, however, many of these states typically cannot occur. States that are infeasible for various reasons should be identified and removed from the model.

In GMCR II, infeasible states can be specified by applying options constraints under one or more of the following four categories:

1) Mutually Exclusive Options,
2) "At Least One" Option,
3) Option Dependence, and
4) Direct Specification.

Figure 4.4 shows GMCR II's starting dialog box for the specification of infeasibilities. This dialog is invoked via the menu item "Modeling/States/Remove Infeasible...". A user has the opportunity to indicate under which categories the infeasibilities are to be specified. For the illustrative Elmira conflict, only the first two categories are used, and therefore only the upper two check boxes are turned on in Figure 4.4.

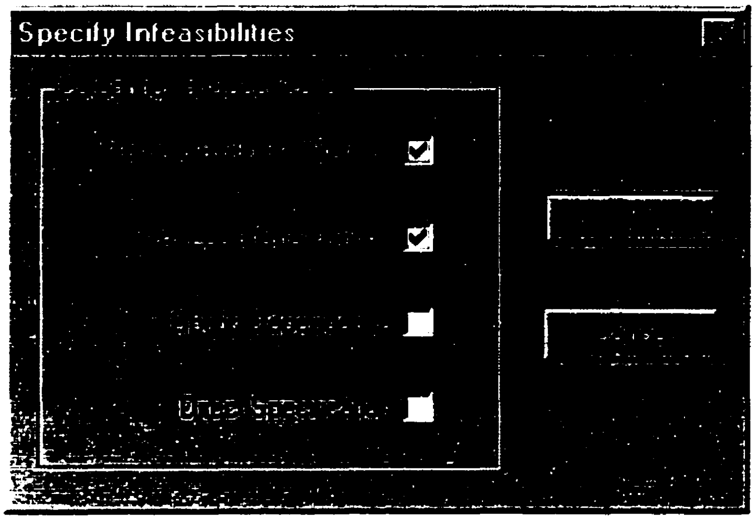


Figure 4.4: Starting Dialog Box for the Specification of Infeasibility

#### 4.3.2.1 Mutually Exclusive Options

A set of options is *mutually exclusive* if at most one option from the set can be taken. If a set of options $\mathbf{O}_{me} \subseteq \mathbf{O}$ is designated as mutually exclusive options, then all the states in the set

$$\{u \in \{0,1\}^{\mathbf{O}} \mid \exists o_{j_1} \in \mathbf{O}_{me}, o_{j_2} \in \mathbf{O}_{me}, j_1 \neq j_2, \textit{ such that } u(o_{j_1}) = u(o_{j_2}) = 1\}$$

are specified as infeasible.

The dialog box for the entry of mutually exclusive options is shown in Figure 4.5. For the Elmira conflict model, only one set of mutually exclusive options is specified in Figure 4.5. In this case, Uniroyal can at most select one of its options Delay, Accept and Abandon. In the dialog box, the user can input and maintain a list of columns, each representing a set of mutually exclusive options, to the right

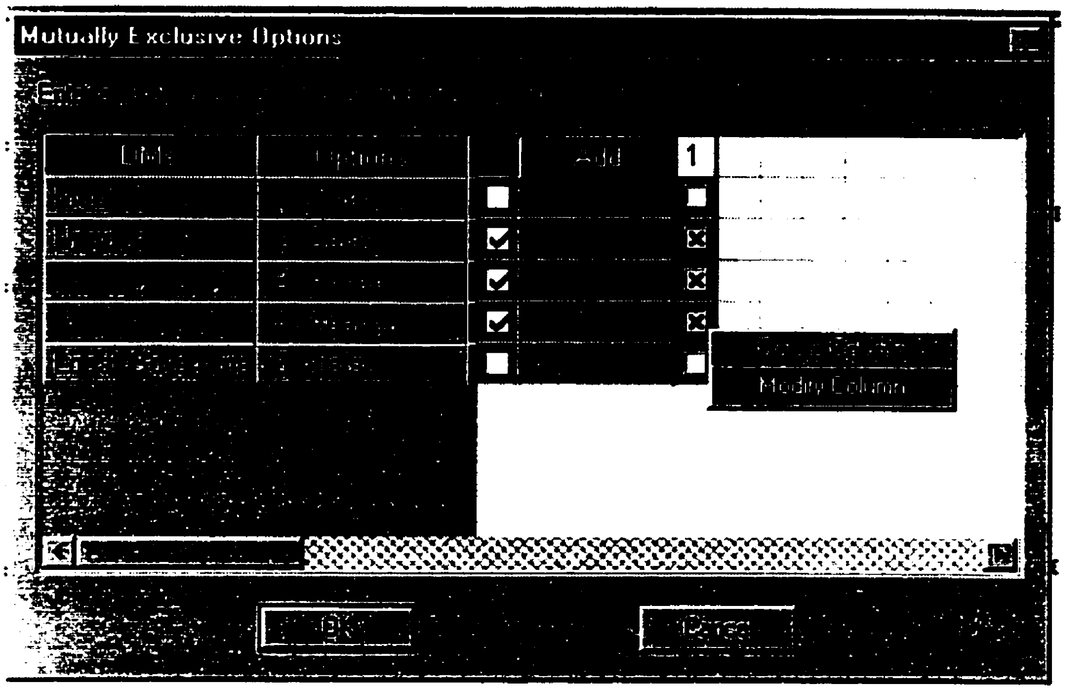


Figure 4.5: Dialog Box for the Entry of Mutually Exclusive Options

of the "Add" button. The user can check on the relevant boxes to the left of the "Add" button, and the selection will be shown as a new column on the list, where the "x" marks are initially read-only. A highlighted column can be either deleted or modified. In the case of modification, the "x" marks on the column will be open for any necessary changes, and will resume read-only status when the highlight moves to another column. Just like for any other dialog box in GMCR II, a simple description of the operations appears on the status bar of the main window, while the dialog box is active.

#### 4.3.2.2 "At Least One" Option

In the "at least one" dialog box, sets of options can be specified such that any state must contain at least one option from each set. If one of the sets specified is $\mathbf{O}_{al} \subseteq \mathbf{O}$, then any state belonging to

$$\{u \in \{0,1\}^{\mathbf{O}} \mid \forall o_j \in \mathbf{O}_{al}, u(o_j) = 0\}$$

is considered infeasible.

The dialog box for "at least one" input is shown in Figure 4.6. For the Elmira conflict, Uniroyal must choose at least one from its listed options. The input mechanism for this dialog box is basically the same as for mutually exclusive options.

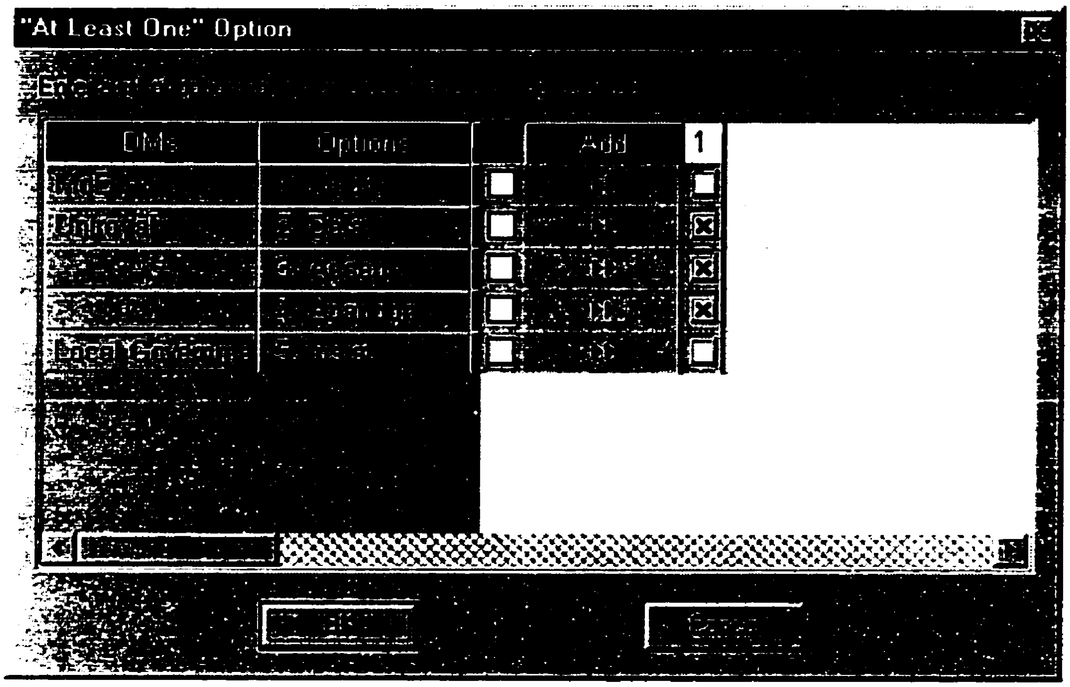
In the implementation, determining whether a state is infeasible under this category is very fast and straightforward. Each column on the list actually corresponds to a mask $m$(a binary format DOUBLEWORD) that designates a range of options in which the compliance with the "at least one" option specification is checked. For example, the "1" column in Figure 4.6 corresponds to a mask

$$m = \overbrace{\cdots\cdots 0 \ \underbrace{111}_{Uniroyal} \ 0}^{32 \ bits} \ (binary)$$

which indicates that among options 1, 2, and 3 the "at least one" constraint is applied. Then whether a state $u$ is infeasible under this specification simply depends on whether $b(u) \ \& \ m = 0$ is true. Here, & means the bit-wise *AND* operator, which is available in $C/C^{++}$.

#### 4.3.2.3 Option Dependence

Under option dependence, two patterns ($p_A$ and $p_B$) are specified. A state that matches pattern $p_A$ is feasible only if it also matches pattern $p_B$; or, pattern $p_A$


"At Least One" Option

[illegible]

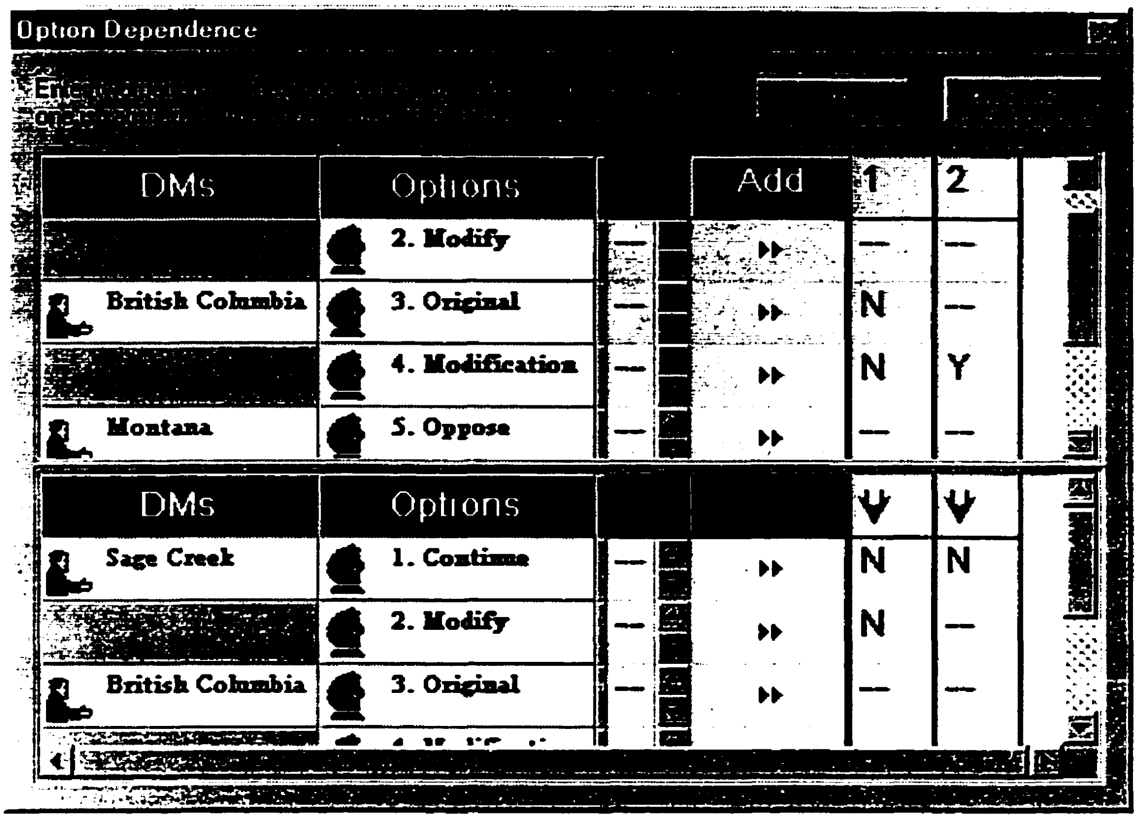
| DMs | Options | | Add | 1 |
|---|---|---|---|---|
| [illegible] | [illegible] | ☐ | [illegible] | ☐ |
| Uniroyal | [illegible] | ☐ | [illegible] | ☒ |
| [illegible] | [illegible] | ☐ | [illegible] | ☒ |
| [illegible] | [illegible] | ☐ | [illegible] | ☒ |
| Local Governme | [illegible] | ☐ | [illegible] | ☐ |

Figure 4.6: Dialog Box for the Entry of "At Least One"

implies pattern $p_B$. In this specification, any state $u$ is infeasible if it belongs to the following set:

$$\{u \in \{0,1\}^{O} \mid u \models p_A, u \not\models p_B\}$$

Figure 4.7 shows the dialog box for entering option dependence in GMCR II. In this dialog box, a user can use the spin buttons to cycle through "Y", "N" or "–" to specify the upper and lower patterns. The context menu can be invoked by a right-click to delete or modify an existing entry. Since there is no option dependence in the Elmira conflict model, the two entries in this figure are for the Flathead River Development Conflict Model, the context of which is outlined in [35, 53]. There are two possible ways to interpret each column entry in this dialog box. The first is that the lower pattern is a necessary condition of the upper pattern; the second is that the upper pattern is a sufficient condition of the lower pattern. In the case of the Flathead River Development Conflict Model (refer to [35] for relevant background information of this case) shown in Figure 4.7, both entries use the second interpretation. In this model, the Sage Creek Coal Limited (Sage Creek) has three choices: to take Option 1 to *continue* the original development, to take Option 2 to *modify* the project to reduce environmental impacts, or to take neither option to simply stop the project. Likewise, another DM, the British Columbia Provincial Government (British Columbia) has three available strategies: to issue a full license for the original project (option 3: Original), to issue a limited license for a modified project (option 4: Modification), or to deny a license for the project (neither option 3 or 4). The two input columns in Figure 4.7 mean that Sage Creek cannot build a project that exceeds the license issued by British Columbia. If British Columbia denies a license $(- - NN - - - -)$, then Sage Creek has to stop$(NN - - - - - -)$. Meanwhile, if British Column issues a limited license for a modified project $(- - NY - - - -)$, then Sage Creek has to take a choice that


Option Dependence

| DMs | Options | | Add | 1 | 2 |
|---|---|---|---|---|---|
| | 2. Modify | — | ▸▸ | — | — |
| British Columbia | 3. Original | — | ▸▸ | N | — |
| | 4. Modification | — | ▸▸ | N | Y |
| Montana | 5. Oppose | — | ▸▸ | — | — |

| DMs | Options | | | ↓ | ↓ |
|---|---|---|---|---|---|
| Sage Creek | 1. Continue | — | ▸▸ | N | N |
| | 2. Modify | — | ▸▸ | N | — |
| British Columbia | 3. Original | — | ▸▸ | — | — |

Figure 4.7: Dialog Box for Entering Option Dependence

excludes full original development ($N - - - - - - -$).

Using the pattern matching technique introduced in Section 4.2.2, the assessment of the feasibility of a state is simple and efficient. What is interesting is that the information taken from this dialog box can be used not only for the generation of feasible states, but also in the calculation of allowable transitions between feasible states, particularily in determination of "forcing moves", for which details will be discussed in Section 4.4.4.

#### 4.3.2.4 Direct Specification

The three methods of specifying infeasible states discussed earlier in this section are relatively user-friendly, and able to handle most of the real-world cases for infeasibility specification. However, there is no guarantee that all types of infeasibilities can be easily expressed under the above three categories. As a supplementary means, the Direct Specification category is offered, under which statements each consisting of a logical combination of options can be input to specify infeasible states.

The dialog box for direct specification of infeasible states is shown in Figure 4.8. For the Elmira conflict, all infeasible states were specified under the first two categories, so there is no need to use direct specification. Nonetheless, for illustration purposes, two statements are shown in Figure 4.8. The one already on the list is equivalent to the specifications in Figure 4.5, and the one being added to the list equivalent to the specification in Figure 4.6.

A direct specification statement is expressed as a combination of available option numbers and logical connectives including negation ("NOT" or −), conjunction ("AND" or & ) and disjunction ("OR" or | ). Brackets are used to control the priority of operations in a statement. The interpretations of the two statements input in Figure 4.8 are given in Table 4.1. As can be seen, each direct specification statement corresponds to one or more patterns which represent all states at which the statement is true. The transformation from a direct specification statement to its corresponding infeasible pattern(s) is carried out automatically by GMCR II. Again, using the pattern-matching techniques (4.2) introduced in Section 4.2.2, those states that are infeasible under this category can be easily identified.

A specification entered in the edit box on the upper right of the dialog box will be put into the list box below when the "Add to List" button is pressed. A

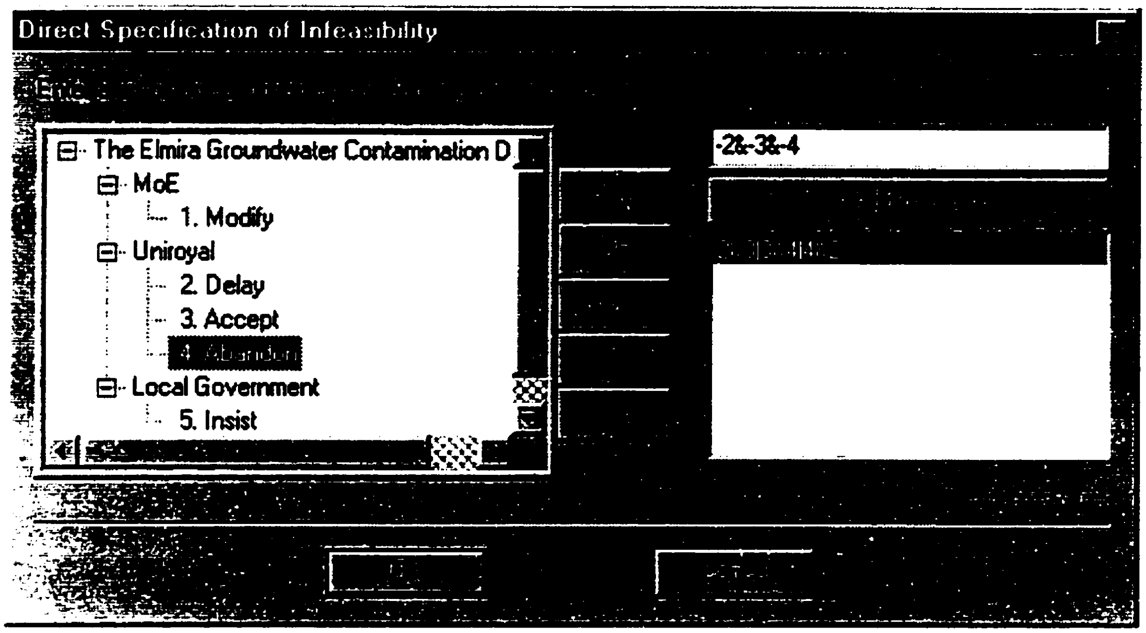


Figure 4.8: Dialog Box for Direct Specification of Infeasibilities

Table 4.1: Interpretation and Corresponding Patterns of Direct Specification Statements

| **Statement** | **Interpretation** | **Infeasible Pattern(s)** |
|---|---|---|
| 2 & 3 \| 3 & 4 \| 4 & 2 | Uniroyal cannot both "Delay" and "Accept", or both "Accept" and "Abandon", or both "Abandon" and "Delay". In other words, it can take at most one from its options. | ··· − −YY−<br>··· − YY − −<br>··· − Y − Y− |
| −2 & −3 & −4 | Uniroyal has to take at least one of the options "Delay", "Accept" and "Abandon" | ··· − NNN− |

highlighted list item can be deleted or modified by invoking the context menu. In the case of modification, the original text of the list item will be put back to the edit box for re-editing, and the label of the "Add to List" changed to "Confirm Modification".

A direct specification can be entered to the edit box in two different ways. A user could simply type into the edit box the logical combination of options. In this case, the options shown in the tree view on the left of the dialog box are used as a reference. The other way is to only use the mouse as input device - a click on a leaf of the tree view will enter the option number in the edit box, while a click on the buttons in the middle of the dialog box will enter a logical operator or a bracket. An "error preventing" feature is implemented in this dialog box for the second input method, as outlined in Table 4.2. When a row item has just been clicked to enter a character, there are always some column items to be disabled, preventing many possible input errors. For example, in Figure 4.8, since the option number 4 has just been entered, the "NOT" button, the "(" button and the tree view control (for entering option number) are disabled (grey), because they are irrelevant at this moment.

### 4.3.3 *Indistinguishable State Combination*

Sometimes in a conflict model a group of states with a common pattern is indistinguishable and, thus, should be treated as a single state. For example in the Elmira Conflict (Section 2.4), once Uniroyal abandons its Elmira plant, MoE's and Local Government's choices are irrelevant. Thus, all states containing the information that Uniroyal abandons its Elmira plant should be considered indistinguishable.

The dialog box for coalescing indistinguishable states is shown in Figure 4.9. To

Table 4.2: Relevance of Input Sequences in Direct Specification Dialog

| | AND | OR | NOT | ( | ) | option tree |
|---|---|---|---|---|---|---|
| AND | disabled | disabled | | disabled | disabled | |
| OR | disabled | disabled | | | disabled | |
| NOT | disabled | disabled | disabled | disabled | disabled | |
| ( | disabled | disabled | | disabled | disabled | |
| ) | disabled | | disabled | disabled | disabled | disabled |
| option tree | | | disabled | disabled | | disabled |

specify which states are to be coalesced, users are required to provide the common characteristics of the states in the form of a pattern. As can be seen, a user can use the spin buttons to specify a pattern in the form of a column of "Y", "N" and "-"s. Then the group of feasible states that matches the pattern will be combined into a single state.

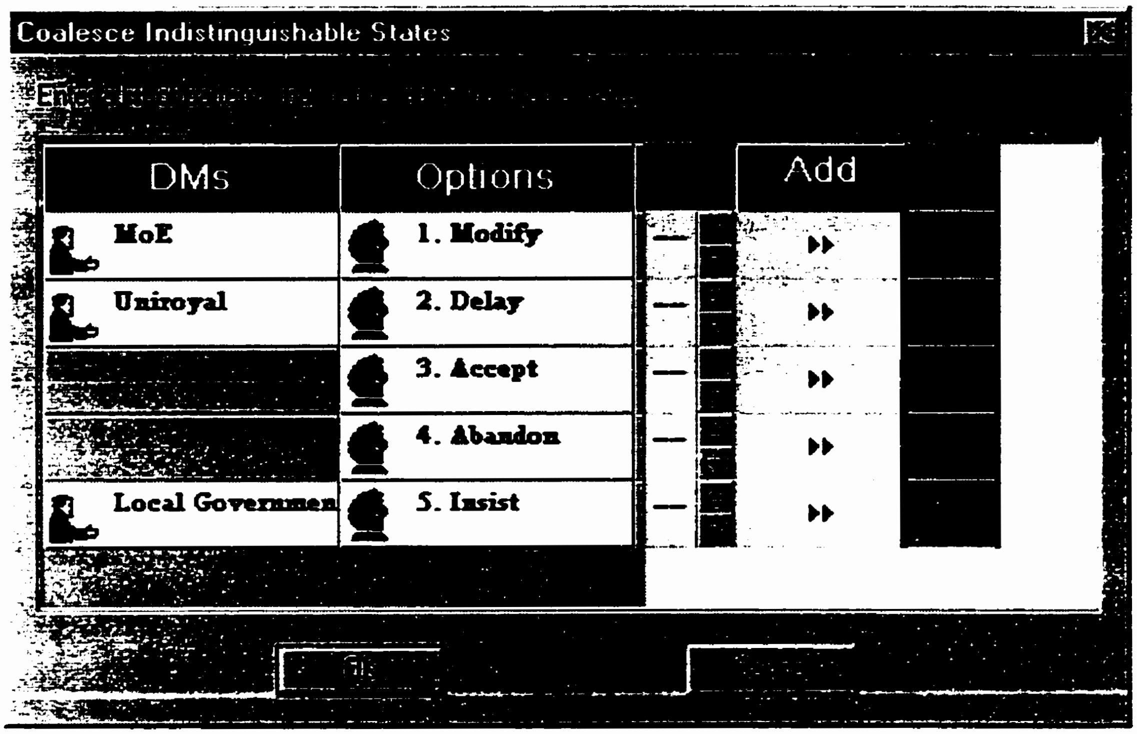
Because of the introduction of indistinguishable states, the definition of feasible states given in Section 4.1 now needs to be refined. If two infeasible states $u_1, u_2 \in \mathcal{U}$ are indistinguishable, write $u_1 \sim u_2$. Obviously, from the nature of indistinguishablity, we have:

1) $u_1 \sim u_2 \Rightarrow u_2 \sim u_1$ $(symmetry)$;

2) $u_1 \sim u_1$ $(reflexivity)$;

3) $u_1 \sim u_2, u_2 \sim u_3 \Rightarrow u_1 \sim u_3 (transitivity)$.

Consequently, $\sim$ is an *equivalence relation*. For $u \in \mathcal{U}$, denote the equivalence



Coalesce Indistinguishable States

| DMs | Options | | Add |
|---|---|---|---|
| MoE | 1. Modify | — | ▸▸ |
| Uniroyal | 2. Delay | — | ▸▸ |
| | 3. Accept | — | ▸▸ |
| | 4. Abandon | — | ▸▸ |
| Local Governmen | 5. Insist | — | ▸▸ |

Figure 4.9: Dialog Box for the Specification of Indistinguishable States

class containing $u$ as $\bar{u} = \{u' \mid u' \in \mathcal{U}, u' \sim u\}$. The set of ***feasible and distinguishable states*** is now defined as:

$$\mathbf{U} = \mathcal{U}/\sim\, = \{\bar{u} \mid u \in \mathbf{U}\}$$

An equivalence class $\bar{u} \in \mathbf{U}$ is called an ***atomic state*** if $|\bar{u}| = 1$; or a ***composite state*** if $|\bar{u}| > 1$. The feasible states referred to earlier in this thesis are actually atomic states.

Since an atomic state $\bar{u}$ is a single-element set, $\bar{u} = \{u\}$, it can be equivalently denoted by $u$. Hence the binary format representation and pattern-matching technique (4.2) presented in Section 4.2 remain valid for atomic states.

Since a composite state and a pattern all represent a group of option combinations, there is no fundamental difference in their representation. For example, state 9 in Figure 4.1 is identical in representation to the pattern used to specify it in Figure 4.9. With respect to implementation, the two-mask data structure for patterns could also be used for composite states. However, in GMCR II, the data structure for composite states is slightly different from the one for patterns. For a composite state $\bar{u}$, define two DOUBLEWORD ***dash-indicator*** $d(\bar{u})$ and ***Y-indicator*** $b(\bar{u})$ as follows:

$$d(\bar{u}) = \sum_{i=1}^{m} D(\bar{u}, o_i) \cdot 2^{i-1}$$

where

$$D(\bar{u}, o_i) = \begin{cases} 1 & \text{if } \exists u', u'' \in \bar{u}, \text{ such that } u'(o_i) = 0 \text{ and } u''(o_i) = 1 \\ 0 & \text{otherwise} \end{cases}$$

and

$$b(\bar{u}) = \sum_{i=1}^{m} B(\bar{u}, o_i) \cdot 2^{i-1}$$

where

$$B(\bar{u}, o_i) = \begin{cases} 1 & \text{if } \forall u' \in \bar{u},\ u'(o_i) = 1 \\ 0 & \text{otherwise} \end{cases}$$

Obviously, when $d(\bar{u}) = 0$, $\bar{u}$ reduces to an atomic state, of which the binary representation is $b(\bar{u})$. In this thesis, an atomic state is also considered a special case of composite state under relevant context.

A composite state $\bar{u}$ is said to *match* a pattern $p$ ( $\bar{u} \models p$), if

$$\forall u' \in \bar{u}, u' \models p$$

Again, in the implementation, the test of whether a composite state matches a pattern is broken down into a few simple bit-wise operations:

**Proposition 4.2** *A composite state $\bar{u}$ matches pattern $p$ (i.e. $\bar{u} \models p$), iff*

$$1) \quad d(\bar{u})\ \&\ (m_0(p) \wedge m_1(p)) = d(\bar{u}) \tag{4.3}$$

$$2) \quad b(\bar{u}) \mid m_0 = m_0 \tag{4.4}$$

$$3) \quad b(\bar{u})\ \&\ m_1 = m_1 \tag{4.5}$$

*where "& ", "^", and " | " are bit-wise AND, bit-wise exclusive OR, and bit-wise inclusive OR, respectively.*

For an atomic state, the condition 1) above is always true, so Proposition 4.2 is consistent with Proposition 4.1.

In the remaining part of this thesis, "state" refers to a feasible and distinguishable state, unless otherwise indicated. For simplicity, the use of $u$ to denote a state will be continued.

### 4.3.4 *Generating the Feasible and Distinguishable State List*

When the information about DMs and options, as well as infeasibility and indistinguishablity specifications have been entered, GMCR II automatically generates a listing of all feasible and indistinguishable states. Figure 4.10 depicts the major steps of this procedure:

**Step 1.** Based on the number of options available to all DMs involved, $m$, GMCR II generates a flow of all $2^m$ mathematically possible states in a straightforward manner. These are equivalent to all possible option combinations, each represented by a DOUBLEWORD:

$$\begin{array}{llcll}
\cdots\cdots 00\cdots 000 & (\textit{binary}) & = & 0 & (\textit{decimal}) \\
\cdots\cdots 00\cdots 001 & (\textit{binary}) & = & 1 & (\textit{decimal}) \\
\cdots\cdots 00\cdots 011 & (\textit{binary}) & = & 2 & (\textit{decimal}) \\
 & & \vdots & & \\
\cdots\cdots 11\cdots 111 & (\textit{binary}) & = & (1 \ll m) - 1 & (\textit{decimal})
\end{array}$$

Here "$\ll$" means bit-wise left shift, another low level operator available in $C/C^{++}$. "$1 \ll m$" is used to implement $2^m$ more efficiently than the power function.

**Step 2.** The system loops through each candidate from the above-mentioned flow to determine whether it is feasible and distinct. The option constraints obtained from the infeasibility specification are used as a filter. Infeasible states do not pass this filter and are removed from the model. The filtering procedure was discussed in Section 4.3.2. In real-world conflict models, usually a considerable number of candidates are eliminated at this step.

**Step 3.** The coalescing of indistinguishable states from the feasible states resulting from step 2 is done using a "remove and add" approach. Since the feasible states that pass step 2 come one-at-a-time, those that match an indistinguishable pattern, and hence are to be included in a composite state, are removed first.

**Step 4.** The composite states specified by the indistinguishable patterns are added to the model with the representation discussed in Section 4.3.3. However, a possible pit-fall here is that an indistinguishable pattern entered by the user may not be feasible. Therefore, each indistinguishable pattern is subject to a feasibility check before the composite state it corresponds is added to the list. GMCR II conducts an alternative feasibility check for each indistinguishable pattern by keeping track of the number of feasible candidates removed under it in step 3. A pattern under which no candidate is eliminated is infeasible. Patterns that pass the feasibility check are added to the list as representations of composite states.

Following this procedure, and taking advantages of the pattern matching techniques described in Section 4.2.2, the generation of the feasible and distinguishable list is quite efficient, even for very large conflict models. For example, the model for the Trade in Services Conflict (documented and explained in [34, 40]) possesses 6 DMs and 20 options, and hence $2^{20} = 1,048,576$ mathematically possible states. On a personal computer with a 266mHz Pentium II processor, it takes less than 30 seconds to generate the 184,320 feasible states.

Let the list of feasible and distinguishable states generated be

$$\mathrm{U} = \{u_1, u_2, \cdots, u_\nu, u_{\nu+1}, \cdots, u_\mu\} \tag{4.6}$$

Due to the way the list is generated, the atomic states come first in the list. Assume that the first $\nu$ states in the above list are atomic ones, and the rest are composite

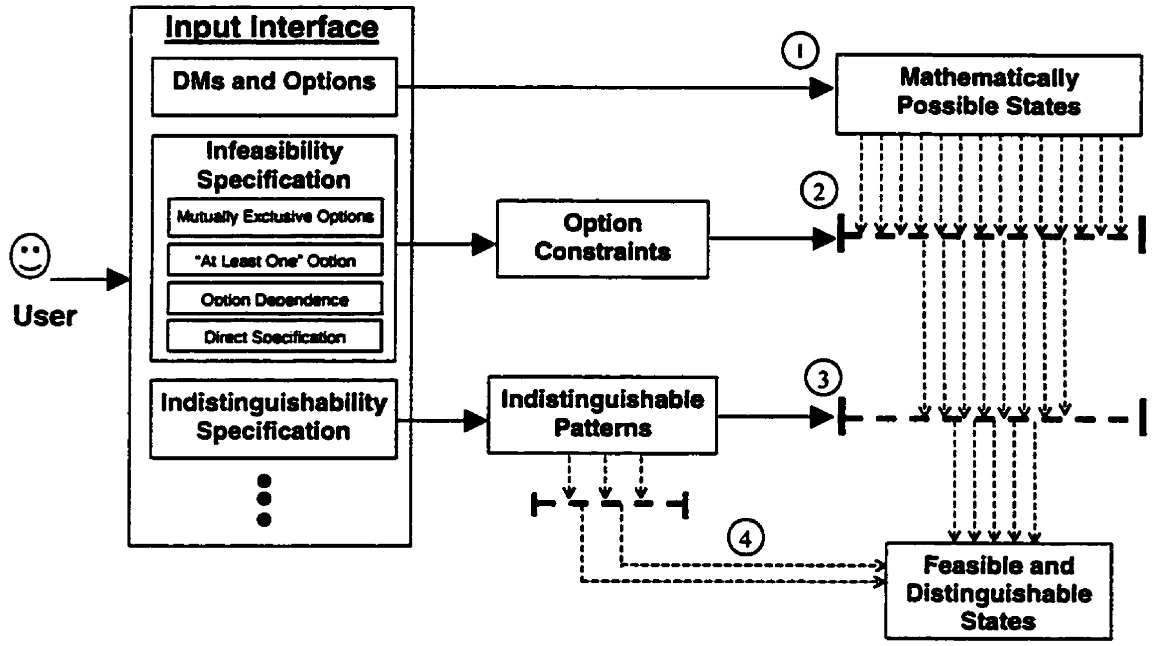

Input Interface
DMs and Options
Infeasibility Specification
Mutually Exclusive Options
"At Least One" Option
Option Dependence
Direct Specification
Indistinguishability Specification
User
Option Constraints
Indistinguishable Patterns
Mathematically Possible States
Feasible and Distinguishable States
1
2
3
4

Figure 4.10: Generating the List of Feasible and Indistinguishable States

ones. Typical models have only a few composite states; most of the list is occupied by atomic states. Observe that the order of the candidates produced in step 1 above gives

$$0 \leq b(u_1) < b(u_2) < \cdots < b(u_\nu) < 2^m \qquad (4.7)$$

Recall that sometimes the index is also used to identify a state, as discussed in Section 4.6. GMCR II uses a dynamically allocated array to store the set of feasible and distinguishable states. The access time for an indexed state is constant and is independent of the array size.

In GMCR II, the list of feasible and distinguishable states once generated, is made available in a user-friendly display as shown in Figure 4.1. This display can be brought up at any later time. Outside the context of this section, the feasible and distinguishable states are sometimes also referred to as feasible states for short.

## 4.4 Allowable Transitions

### 4.4.1 *Irreversible Transitions*

*State transition* is the process by which a conflict model moves from one state to another. If a DM can cause a state transition on his or her own, then this transition is called a *unilateral move (UM)* for that DM. As denoted in Section 4.6, if there is a UM by DM $i$ from $u_k$ to $u_q$, then $R_i(u_k, u_q) = 1$. The list of all states to which DM $i$ has a UM from state $u_k$ is $\mathbf{S}_i(u_k)$. Allowable state transitions constitute an important modeling component, which determines the structure of a graph model.

In the original option form, it is assumed that a DM has a UM from one state to another if and only if the two states differ only in one or more options controlled by that DM. In other words, the original option form assumes that

$$R_i(u_k, u_q) = 1 \ (i.e. \ u_q \in S_i(u_k)) \ \Leftrightarrow \ \forall o_j \in \mathbf{O} \backslash \mathbf{O}_i, u_k(o_j) = u_q(o_j) \quad (4.8)$$

Note that the original option form did not include the concept of indistinguishable states; all states were considered atomic. Thus, the state transitions for a model are implied by the input information about DMs and options, as discussed in Section 4.3.1. Also

- a UM is always *reversible*:

$$R_i(u_k, u_q) = 1 \Leftrightarrow R_i(u_q, u_k) = 1 \quad (i.e. \ u_q \in S_i(u_k) \Leftrightarrow u_k \in S_i(u_q))$$

- UMs are always *transitive*:

$$R_i(u_k, u_q) = 1, R_i(u_q, u_l) = 1 \Rightarrow R_i(u_k, u_l) = 1$$

$$(i.e. \ u_q \in S_i(u_k), u_l \in S_i(u_q) \Rightarrow u_l \in S_i(u_k))$$

In fact, it is these assumptions that lead to the limitations of the option form outlined in Section 2.4. In GMCR II, the transition aspect of option form is improved by introducing option-based irreversibility.

In an *irreversible transition* or *irreversible move*, a DM can, for instance, unilaterally cause a state transition from state $u_k$ to $u_q$, but cannot make the reverse move from $u_q$ to $u_k$. In GMCR II, the right hand side of (4.8) is used as a default necessary condition but not a sufficient condition. A user can specify irreversibility by applying some restrictions to this condition.

### 4.4.2 *Specification of Irreversibility*

In GMCR II, irreversibility can be specified based either on a single option or on multiple options.

#### 4.4.2.1 Single Option Based Irreversibility

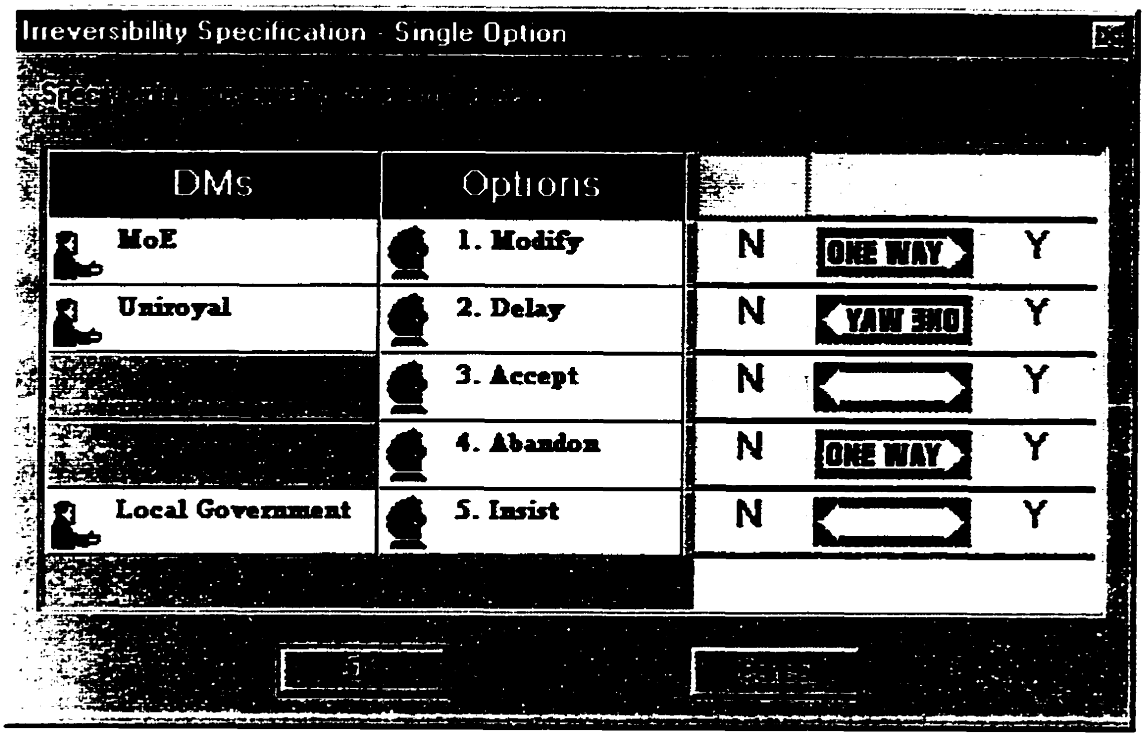
A state transition by a DM from one state to another could be infeasible even if the two states differ only in one or more options controlled by that DM. This occurs, for instance, when an option is irreversible – after the option is selected it cannot be reversed. For example, in the Elmira model, if the DM Uniroyal chooses the option "delay", it cannot take this move back later, because time is irreversible. Figure 4.11 shows the GMCR II dialog box where a user can indicate the irreversible options. A "ONE WAY" arrow from "N" to "Y" beside an option means that changes in the selection of that option are permitted only from N (not selected) to Y (selected). Likewise, an arrow from "Y" to "N" means the option can only be changed from Y (selected) to N (not selected). A bidirectional arrow, the default status, means no restriction on the change of the option. The user can easily double-click an arrow to toggle an arrow type.

The information contained in this dialog box can be summarized as a mapping:

$$
\begin{aligned}
\imath : \mathbf{O} &\longrightarrow \{0,1,2\} \\
o_j &\longmapsto \imath(o_j) = \begin{cases} 0 & \text{if the selection of option } o_j \text{ can only} \\ & \text{be changed from not selected to selected} \\ 1 & \text{if the selection of option } o_j \text{ can only} \\ & \text{be changed from selected to not selected} \\ 2 & \text{otherwise} \end{cases}
\end{aligned}
$$

GMCR II applies the following restrictions on the allowable state transitions between two atomic states $u_k$ and $u_q$, $1 \le k \le \nu, 1 \le q \le \nu$, based on the above specification:

- $\imath(o_j) = 0, u_k(o_j) = 1, u_q(o_j) = 0 \Rightarrow R_i(u_k, u_q) = 0 (i.e.\ u_q \notin \mathbf{S}_i(u_k)), 1 \le i \le n;$



Irreversibility Specification - Single Option

| DMs | Options | | | |
|---|---|---|---|---|
| MoE | 1. Modify | N | ONE WAY ▶ | Y |
| Uniroyal | 2. Delay | N | ◀ YAW ƎNO | Y |
| | 3. Accept | N | ◀▶ | Y |
| | 4. Abandon | N | ONE WAY ▶ | Y |
| Local Government | 5. Insist | N | ◀▶ | Y |

Figure 4.11: Dialog Box for Specifying Irreversible Options

- $\imath(o_j) = 1, u_k(o_j) = 0, u_q(o_j) = 1 \Rightarrow R_i(u_k, u_q) = 0 (i.e.\ u_q \notin \mathbf{S}_i(u_k)), 1 \leq i \leq n.$

To check whether a state transition is infeasible based on these restrictions is very efficient in implementation:

$$u(o_j) = 1 \quad \Leftrightarrow \quad b(u)\ \&\ (1 \ll (j-1)) \neq 0$$

where " & " and "$\ll$" are bit-wise AND and bit-wise LEFT-SHIFT, respectively.

When composite states are involved, the following conventions are made based on modeling practice:

- For an atomic state $u$ and a composite state $\bar{u}$,

$$\bar{u} \in \mathbf{S}_i(u) \quad \Leftrightarrow \quad \exists u' \in \bar{u}, u' \in \mathbf{S}_i(u) \tag{4.9}$$

$$u \in \mathbf{S}_i(\bar{u}) \quad \Leftrightarrow \quad \forall u' \in \bar{u}, u \in \mathbf{S}_i(u')$$

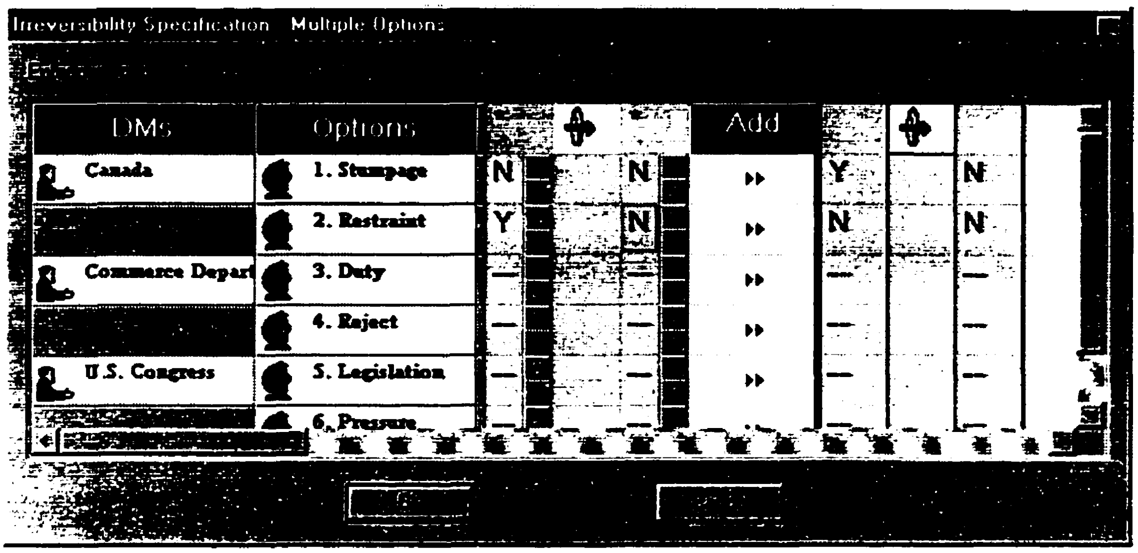
- For two composite states $\bar{u}'$ and $\bar{u}"$,

$$\bar{u}" \in \mathbf{S}_i(\bar{u}') \quad \Leftrightarrow \quad \forall u \in \bar{u}', \bar{u}" \in \mathbf{S}_i(u)$$

The information shown in the above dialog box happens to be sufficient to define all irreversible moves in the illustrative Elmira model (refer to Table 2.3 and Figure 2.1). However, in some situations, selection restrictions on multiple options, rather than on a single option, are needed.

#### 4.4.2.2 Multiple Option Based Irreversibility

The GMCR II dialog box for irreversibility specification based on multiple options is shown in Figure 4.12. In this dialog, the user is asked to enter a pair of patterns


Irreversibility Specification Multiple Options

| DMs | Options | | | Add | | |
|---|---|---|---|---|---|---|
| Canada | 1. Stumpage | N | N | ▸▸ | Y | N |
| | 2. Restraint | Y | N | ▸▸ | N | N |
| Commerce Depart | 3. Duty | — | — | ▸▸ | — | — |
| | 4. Reject | — | — | ▸▸ | — | — |
| U.S. Congress | 5. Legislation | — | — | ▸▸ | — | — |
| | 6. Pressure | — | — | | — | — |

Figure 4.12: Dialog Box for Irreversibility Specification based on Multiple Options

such that the move from the left pattern to the right pattern is not allowed. A list of pattern pairs can be maintained in the area to the right of the "Add" button. This dialog offers easy deletion and modification functions. The two pairs displayed in Figure 4.12 (one is already on the list, the other is being added to list) are for the Softwood Lumber model, details of which can be found in [38]). Note that this dialog box has the capacity to specify infeasible movement between any pair of states, because even an atomic state can be envisioned as a pattern – one without "–". Also, any irreversible option specified in Figure 4.11 can also be equivalently input in this dialog, in the form of a infeasible movement between two patterns. For example, the first right arrow in Figure 4.11 can be represented as $(N----) \not\rightarrow (Y----)$. Of course, for irreversible options, the dialog box in Figure 4.11 is more efficient.

Let the list of pattern pairs input in this dialog box be $\{(p_1^L, p_1^R), (p_2^L, p_2^R), \cdots, (p_h^L, p_h^R)\}$. According to GMCR II,

$$\exists k, 1 \leq k \leq h, u' \models p_k^L, u'' \models p_k^R \Rightarrow R_i(u', u'') = 0 \ (i.e.\ u'' \notin \mathbf{S}_i(u')), 1 \leq i \leq n.$$

Again, the pattern-matching technique (4.2, 4.5) is used to implement this method of picking out prohibited transitions.

### 4.4.3 *Calculation of Reachability*

With the information elicited from the specification of irreversibility based on both single option and multiple options, the reachable list from each state for each DM can be calculated. A straightforward implementation for calculating a reachable list $\mathbf{S}_i(u)$ would be:

**Step 1.** Obtain a state from the feasible state list as candidate. If the end of the list is reached, stop.

**Step 2.** Check whether the incoming candidate differs from $u$ only on one of more options that DM $i$ controls. If yes, go to 3; otherwise, go to 1.

**Step 3.** Check whether the candidate passes the test based on the specification in Section 4.4.2.1. If yes, go to 4; otherwise, go to step 1.

**Step 4.** Check whether the candidate passes the test based on the specification in Section 4.4.2.2. If, yes, add the candidate into $\mathbf{S}_i(u)$; otherwise, go to step 1.

The above procedure was actually adopted in earlier versions of GMCR II. In the worst case, when all mathematically possible states are feasible, the complexity for the above implementation is $O(2^m)$. This can harm the performance of the system when the model is large.

To improve the system's capacity to deal with large models, an innovative strategy is used to reduce the complexity of the above algorithm. Instead of looping through all feasible states, the scope of the search can be limited to a much smaller candidate set. For simplicity, only atomic states are discussed below. Composite states required a different treatment, but since only a very small portion of states are composite, their contribution to the complexity is negligible in any case.

Denote the binary representation of $u$ by

$$b(u) = \mathrm{XXXX}\ \underbrace{\mathrm{X}\cdots\mathrm{XX}}_{DM_i}\ \mathrm{XXX}\ \ (binary).$$

Then each state that differs from $u$ only on options that DM $i$ controls has one of the following binary representations:

$$\begin{array}{l} \mathrm{XXXX}\ \ \overbrace{0\cdots00}^{DM_i}\ \mathrm{XXX} \\ \mathrm{XXXX}\ \ 0\cdots01\ \mathrm{XXX} \\ \mathrm{XXXX}\ \ 0\cdots11\ \mathrm{XXX} \\ \qquad\qquad\vdots \\ \mathrm{XXXX}\ \ 1\cdots11\ \mathrm{XXX} \end{array} \tag{4.10}$$

Therefore the full-range loop in step 1 of the original algorithm is unnecessary; the loop need only cover the entries in (4.10). To build such a loop,

- the starting entry is

$$\begin{aligned} & \mathtt{XXXX\ 0\cdots00\ XXX} \\ = \;& b(u)\ \&\ \mathtt{1111\ 0\cdots00\ 111} \\ = \;& b(u)\ \&\ (\mathtt{1111\ 0\cdots00\ 000} \mid \mathtt{0000\ 0\cdots00\ 111}) \\ = \;& b(u)\ \&\ (((1 \ll m) - (1 \ll \textstyle\sum_{k=1}^{i} m_k)) \mid ((1 \ll \textstyle\sum_{k=1}^{i-1} m_k) - 1)) \end{aligned}$$

- the ending entry is

$$\begin{aligned} & \mathtt{XXXX\ 1\cdots11\ XXX} \\ = \;& b(u) \mid \mathtt{0000\ 1\cdots11\ 000} \\ = \;& b(u) \mid ((1 \ll \textstyle\sum_{k=1}^{i} m_k) - (1 \ll \textstyle\sum_{k=1}^{i-1} m_k)) \end{aligned}$$

- the increment for each entry is $1 \ll \sum_{k=1}^{i-1} m_k$.

In particular, all the components needed to form the limited-range loop are conveniently obtained from bit-wise operations AND ( & ), INCLUSIVE-OR ( | ) and LEFT-SHIFT ($\ll$). Using this new loop to replace steps 1 and 2 in the original algorithm reduces the complexity from $O(2^m)$ to average $O(2^{m/n})$ in the worst case. This is a considerable saving in execution time because when the size of a model increases, the number of DMs, $n$, usually also increases. Moreover, this saving appears in the calculation for the reachable list from *each* states for *each* DM.

The above technique works well when all mathematically possible states are feasible. In this case, the binary representation of a state directly corresponds to the index of the state on the feasible state list (see Section 4.3.4), and hence identifies the state. However, the state index does not depend on the binary representation, when infeasible states are removed. In this case, how to identify the candidate at the feasible state list becomes a new question. Of course, an exhaustive search over the feasible state list for each binary candidate is the easiest way to find its index. But the extra complexity introduced would diminish the validity of this new approach.

GMCR II takes advantage of the monotonic relation between the binary representations and the indices of atomic states (4.7), and use binary search [1] to address this problem. A procedure based on "decremental binary searches" is carried out in GMCR II and depicted in Figure 4.13. The main steps of this procedure are outline as follows:

I. First, let the binary candidates (4.10) go through the feasibility tests as described in Section 4.3.2. Usually quite a few candidates are eliminated at this step.

II. Second, Steps 3 and 4 above also rule out a considerable portion of the remaining candidates.

III. Third, for remaining binary candidates $c_k$ $(k = 1, \cdots, r)$, "decremental binary searches" are used to identify their indices $I(c_k)$ $(k = 1, \cdots, r)$ before they are added to the reachable list.

  - For each $c_k$, let the starting index of binary search be $i_S(c_k)$, $1 \leq i_S(c_k) < \nu$, where $\nu$ is the maximum index for atomic states (4.6). Notice that we have $I(c_k) > I(c_{k-1}), k = 2, \cdots, r$. Therefore, if

$$\imath_S(c_k) = \begin{cases} 1 & if k = 1 \\ I(c_{k-1}) & 1 < k \leq r \end{cases} \tag{4.11}$$

    then

$$I(c_k) \in [i_S(c_k), \nu], k = 1, \cdots, r. \tag{4.12}$$

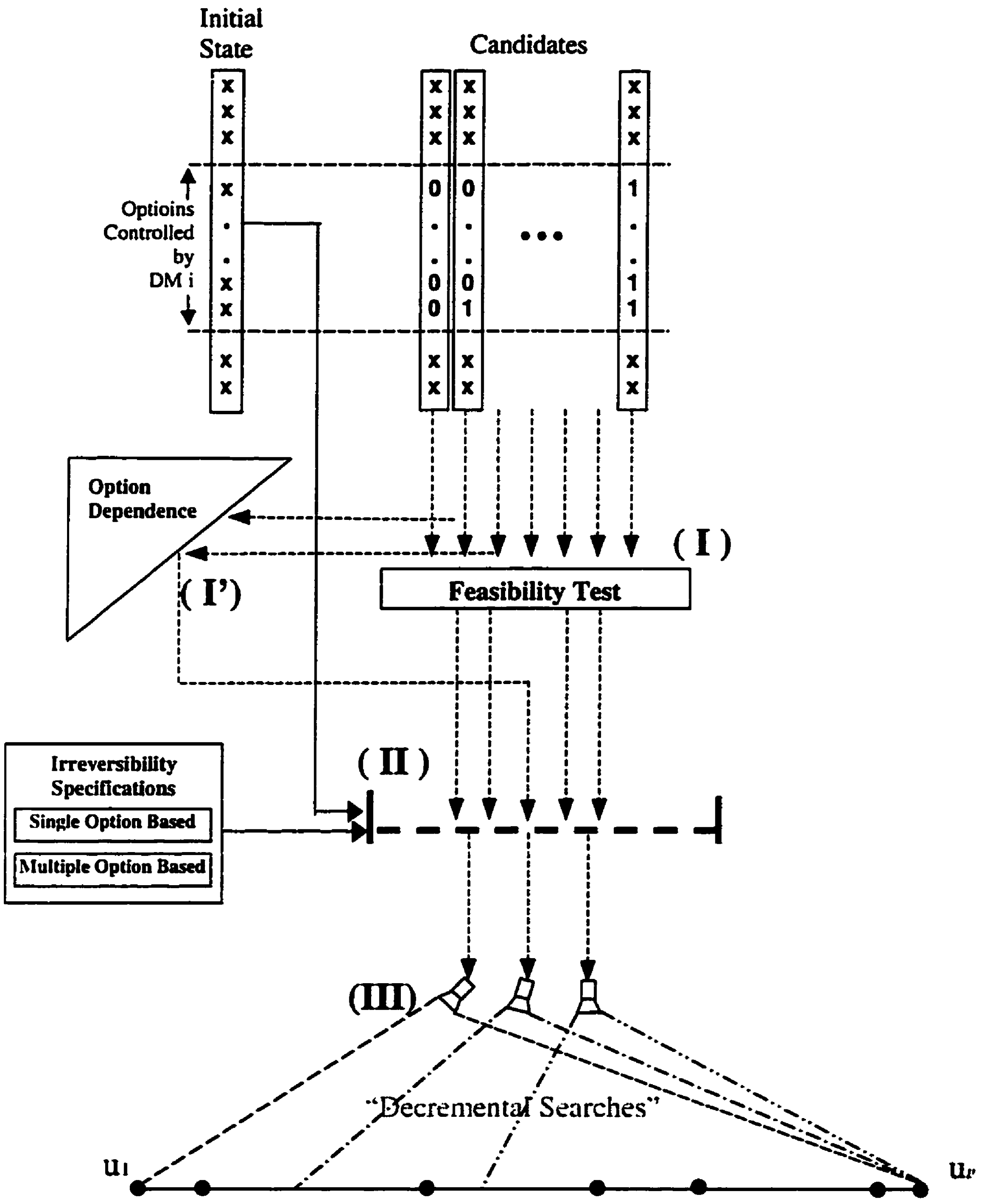


Figure 4.13: The Improved Procedure for Calculation of a Reachable List

This means that the binary searches described below for the indices of the remaining candidates can be performed on decreasing nested scopes. That is why this approach is called "decremental binary searches".

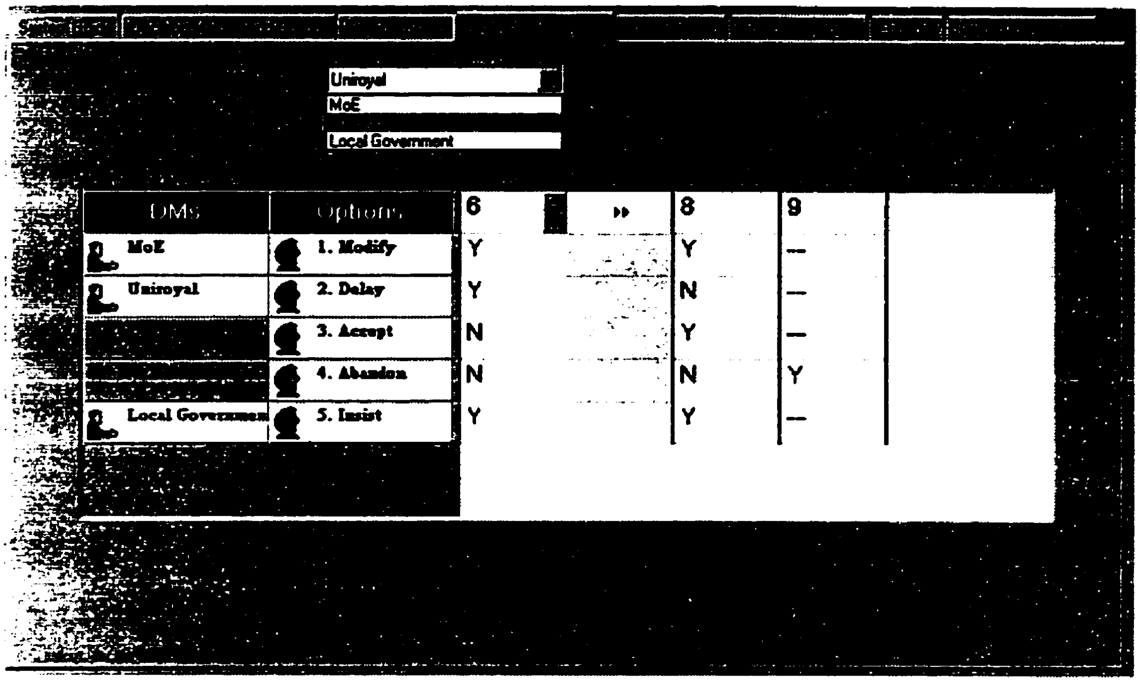
- For $c_k$, perform a binary search on $[i_S(c_k), \nu]$:
  - i. Let $B = i_S(c_k)$ and $E = \nu$. Take the middle index $M = [(B+E)/2]$ and compare $c_k$ with $b(u_M)$.
  - ii. If $c_k < b(u_M)$, let $E = M$, and repeat 1. If $c_k > b(u_M)$, let $B = M$, and repeat 1. If $c_k = b(u_M)$, then $I(c_k) = M$, end.
  - iii. This procedure always finds $I(c_k)$, because $c_k$ has been pre-checked for feasibility.

Because of the pre-checks and the decreasing scopes, the extra complexity factor added by this procedure to the new algorithm would be much smaller than $O(m)$. This innovative strategy for calculating reachable lists greatly enhances GMCR II's capacity in handling large scale models.

The "Allowable Transitions" property page in GMCR II, as shown in Figure 4.14, allows the user to view the reachable list of any state by any DM. In this page, the user can pull down the combo box to select the focal DM. The initial state shown as the column to the left of the double-arrow can be altered by changing the state number using the spin button or direct editing. The states on the reachable list appears instantly with the selection of the focal DM and/or the initial state.

## 4.4.4 *Forcing Moves*

Under some circumstances, a DM who makes a certain move from a state may give one or more other DMs no choice but to respond with "forced" moves that result



Uniroyal
MoE
Local Government

| DMs | Options | 6 | 8 | 9 |
|---|---|---|---|---|
| MoE | 1. Modify | Y | Y | — |
| Uniroyal | 2. Delay | Y | N | — |
| | 3. Accept | N | Y | — |
| | 4. Abandon | N | N | Y |
| Local Governmen | 5. Insist | Y | Y | — |

Figure 4.14: Displaying a Reachable List

in another state. In this way, the DM who initiates the *forcing move* achieves a transition from the initial state to the final states as if it were "unilateral", even though the two states differ on the options that he or she does not control. Forcing moves constitute an exception to the default necessary condition (right hand side of (4.8)) for determining UMs, so the original option form, and even earlier versions of GMCR II, did not include this concept. The term "forcing move" was first used in the proposal of this thesis, and later in [37].

Table 4.3 gives an example of a forcing move in the context of the Flathead River Resource Development Conflict model, which was studied using GMCR II in [35]. In the model, the provincial government (British Columbia) can move from state 3 by changing its strategy from issuing a full license (YN) to denying any license (NN). Another DM, Sage Creek, has no choice but to move accordingly, from developing a full project (YN) to stopping the project (NN). The model thus moves from state 3 to state 1 as a consequence of British Columbia's initial move. State 3 and 1 differ not only on British Columbia's options, but also on Sage Creek's options, yet British Columbia can be modeled as achieving the transition from state 3 to state 1 "unilaterally".

The dialog box presented in Figure 4.7 was developed for specification of infeasible states under the option dependence category. However, looking at the first specification in Figure 4.7 and envisioning the upper pattern ($p_A$: − − NN − − − −) as a "forcing pattern" and the lower pattern ($p_B$: NN − − − − − −)) "forced pattern", one finds that the forcing move to state 1 in Table 4.3 is actually implied in that specification. Therefore, no extra input dialog is needed to specify forcing moves; the option dependence dialog serves this purposes as well.

Since State 1 is not even on the candidate list (4.10) for calculating the reachable list for state 3 according to the procedure outlined in Section 4.4.3, a special

Table 4.3: Example of a Forcing Move

| | | | | | |
|---|---|---|---|---|---|
| **Sage Creek** | | | | | |
| 1. *Continue* | Y | | Y | $\longrightarrow$ | N |
| 2. *Modify* | N | | N | | N |
| **British Columbia** | | | | | |
| 3. *Original* | Y | $\longrightarrow$ | N | | N |
| 4. *Modification* | N | | N | | N |
| **Montana** | | | | | |
| 5. *Oppose* | N | | N | | N |
| **IJC** | | | | | |
| 6. *Original* | N | | N | | N |
| 7. *Modification* | N | | N | | N |
| 8. *No* | Y | | Y | | Y |
| State Numbers | 3 | | * | | 1 |

treatment must be attached to that procedure. In Table 4.3, British Columbia can be interpreted to make a UM from state 3, resulting in the state combination given in the middle column (*) in the table, from which Sage Creek has a UM to state 1. (*) was deemed infeasible by the first specification in Figure 4.7, and hence does not exist in the model. Nonetheless, this "ghost state" can be envisioned as a "refractor" that quickly transfers British Columbia's UM from state 3 to Sage Creek's UM to state 1.

Based on this interpretation, the identification of a forcing move in the calculation of a reachable list can start from identification of the "refractor". Note that the move by the focal DM from the initial state to the "refractor" is a genuine UM, and hence the "refractor" appears in the candidate list (4.10). Therefore, step I in the procedure for calculating reachable lists in Section 4.4.3 can be modified as follows (Figure 4.13):

I'. Test all binary candidates (4.10) for feasibility. Send those that pass to II. For each candidate that fails the test, compare it with each upper pattern ($p_A$, or "forcing pattern") specified under option dependence.

- If a candidate does not match a "forcing" pattern, eliminate it:
- If a candidate $b(u')$ matches a "forcing" pattern $p_A$, transform it to a new candidate $b(u'')$ based on the "forced" pattern $p_B$:

$$b(u'') = b(u') \ \& \ m_0(p_B) \ | \ m_1(p_B) \tag{4.13}$$

Remove $b(u')$ and send $b(u'')$ as a remaining candidate to step II.

For the forcing move example in Table 4.3, the candidate from state 3 representing the "refractor" does not pass the feasibility test, but does match a "forcing" pattern. Applying the transformation (4.13) to it yields state 1. State 1 then passes all remaining steps and eventually joins the reachable list of state 3. In this way, the latest version of GMCR II successfully incorporates forcing moves.

### 4.4.5 *Common Moves*

A *Common move* is a new concept introduced in the context of graph model. A common move refers to the possibility of two or more DMs to independently making unilateral moves that cause the model to move from a particular initial state to the same target state. Common moves do exist in the real world. For example, Fang, Hipel and Kilgour use common moves in a superpower nuclear confrontation model to illustrate the advantages in flexibility of graph model over the option form and normal form [20, Ch.2].

In the nuclear confrontation model, DMs 1 and 2 have three strategies: peace (label P), conventional attack (label C), and full nuclear attack (label W) which is assumed to trigger a nuclear winter. The five distinct states possible are labelled as (PP), (PC), (CP), (CC) and (W). The graph model is illustrated in Figure 4.15; note that the moves from (PP), (PC), (CP) and (CC) to (W) are common moves available to either player.

Since a common move must violate the default necessary condition for a UM, it cannot be handled in the original option form. Common moves are not implemented in the current version of GMCR II, due to the fact that they cannot be represented by single Y-N-"—" columns; also they do not occur very often in practice. However, the implementation of common moves in the GMCR II framework is in principle possible, as will be illustrated below.

Consider the option form model presented in Table 4.4. The columns on the

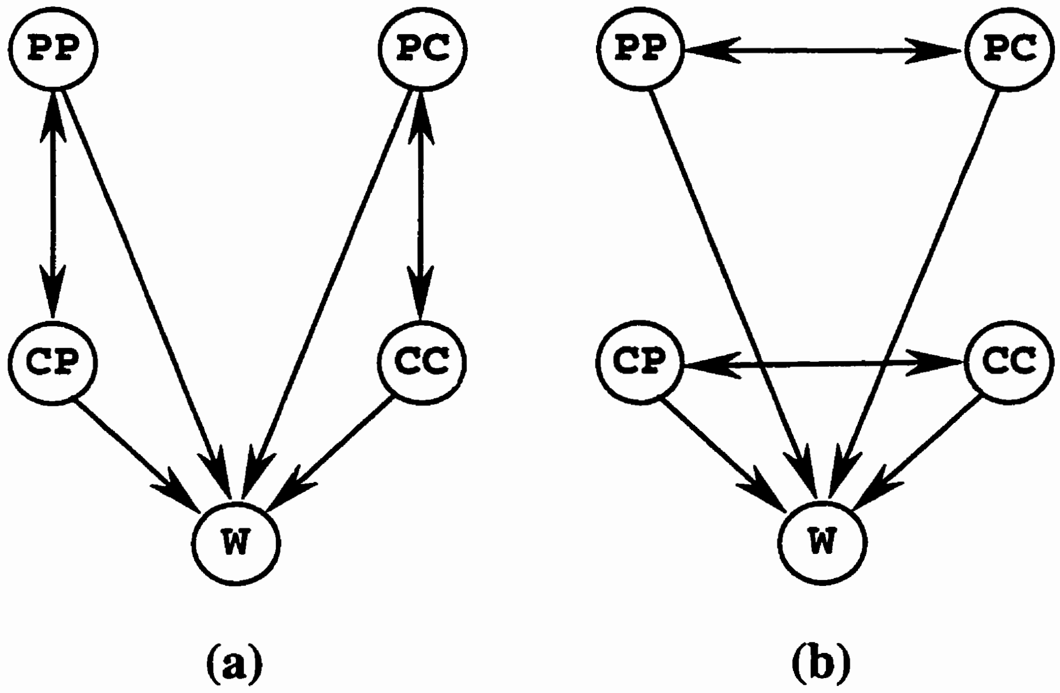


Figure 4.15: Graph Model for Superpower Nuclear Confrontation: (a) Graph for DM 1; (b) Graph for DM 2.

Table 4.4: Option Form for Superpower Nuclear Confrontation Model

| DMs and Options | Feasible States | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Superpower 1** | | | | | | | | | |
| 1. Conventional Attack | N | Y | N | N | Y | N | N | Y | N |
| 2. Full Nuclear Attack | N | N | Y | N | N | Y | N | N | Y |
| **Superpower 2** | | | | | | | | | |
| 3. Conventional Attack | N | N | N | Y | Y | Y | N | N | N |
| 4. Full Nuclear Attack | N | N | N | N | N | N | Y | Y | Y |
| State Number | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| | (PP) | (CP) | (W) | (PC) | (CC) | (W) | (W) | (W) | (W) |

right are the feasible states under the GMCR II specification that both options 1 and 2, and options 3 and 4, form two sets of mutually exclusive options. As can be seen, states 1, 2, 4 and 5 correspond to states (PP), (CP), (PC) and (CC) in the graph model in Figure 4.15. Recall that the logical combinations of options are used in GMCR II for the direct specification of infeasibility (Section 4.3.2.4, Figure 4.8). Obviously, the same design could be used to specify indistinguishability as well (it was actually implemented in an earlier version of GMCR II). If we specify that all states satisfying the statement "2 | 4" are indistinguishable, then a composite state consisting of the atomic states 3, 5, 6, 7, 8 and 9 in Table 4.4 will be obtained. This composite state represents the state (W) in Figure 4.15.

Now, consider the state transitions. Obviously, once a full nuclear attack is launched, the disastrous effect is irreversible. Consequently, options 2 and 4 should be specified as irreversible options that can only be changed from not selected to selected, and not vice versa. Recalling the convention (4.9) of Section 4.4.2.1, one now finds the specifications of indistinguishable states and irreversible options in GMCR II's improved option form, actually determine state transitions identical to those in the graph model (Figure 4.15) – even the common moves! This can be also

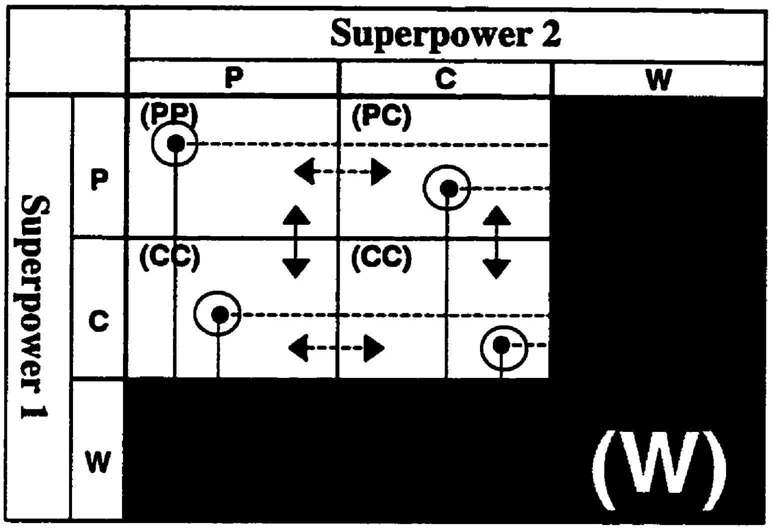


Table 4.5: State Transitions in Superpower Nuclear Confrontation Model

shown in a format similar to normal form as in Table 4.5. In this table, solid arrows and dotted arrows represent Superpower 1's and Superpower 2's UMs, respectively, with common moves indicated by circles. Note that the shaded area represents a single composite state (W).

In conclusion, there is no fundamental difficulty to incorporate common moves in GMCR II. One reason why common moves are not implemented in current version lies in the representation of states. Because of the logical connective OR ( | ) in the statement specifying a common move, a common move cannot be represented by a single Y-N-"–" column. If more occasions of common moves arise in real-world case studies, it should not be too difficult for future versions of GMCR II to get around this representation issue. An interesting observation here is that although option dependence and indistinguishability specification were originally intended for use in generating the state list, they also play a role in determining state transitions, specifically, forcing moves and common moves.

## 4.5 Preferences

### 4.5.1 *Ordinal Preferences in GMCR II*

Preference information about the feasible states is the last, and perhaps the most important input required for a stability analysis under the Graph Model for Conflict Resolution paradigm. In GMCR II, a flexible methodology is presented for conveniently eliciting a DM's relative preferences. More specifically, three techniques are available for ordering the states from most to least preferred, with ties permitted. One method is *Option Weighting*, in which weights are assigned to each option choice, and total weights used to determine an ordering of states. A second technique is to employ an *Option Prioritizing* scheme, based upon a set of lexicographic statements. Subsequently, one can rank the states manually using a process called *Fine Tuning* or *Direct Ranking*. Figure 4.16 depicts how these three techniques can be utilized to determine the preferences of a particular DM. The name of the focal DM, whose preference information is currently being elicited, appears on the title bar.

The most reliable method of ordinal preference elicitation would be to list exhaustively the pair-wise comparisons among feasible states. However, it is usually impractical to do so unless the model is very small. In fact, the most useful method that works explicitly with states is *Direct Ranking*, also called *Fine Tuning*.

But more efficient still are Option Weighting and Option Prioritizing, which are implicit ranking methods based on options. Because the number of options is

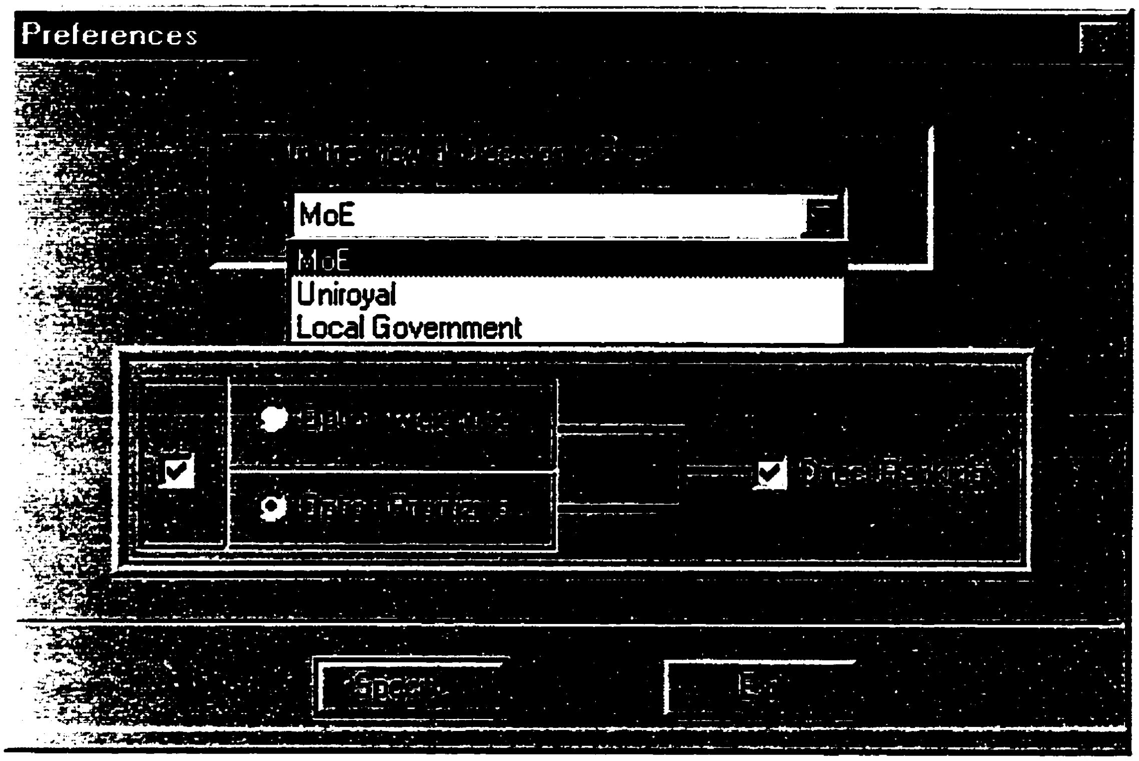


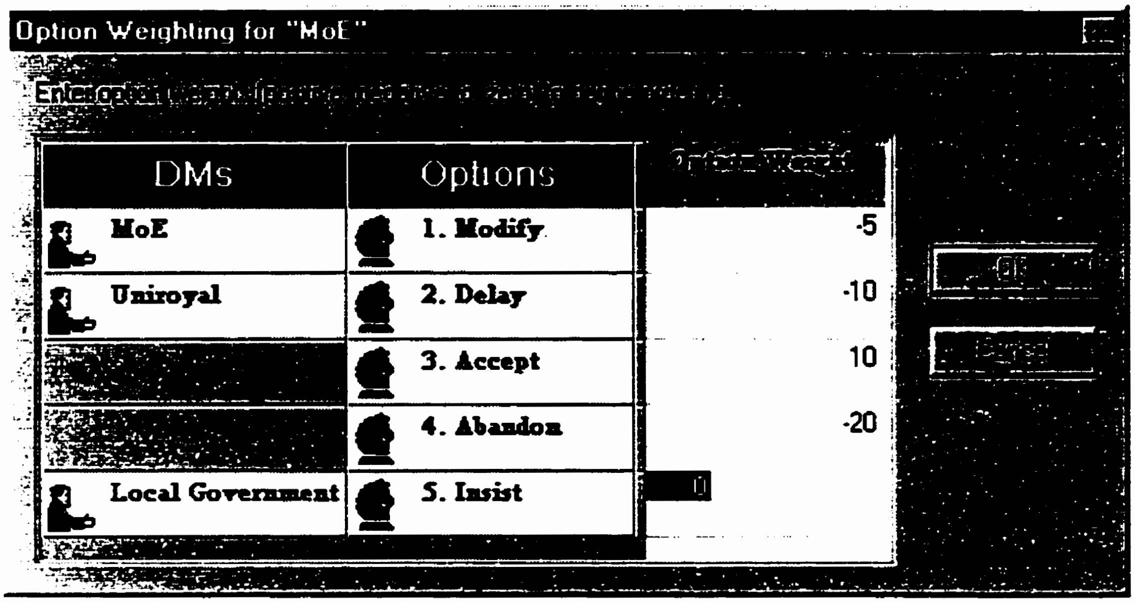
Figure 4.16: Dialog Box for Relative Preference Elicitation

much smaller than the number of states, Option Weighting and Option Prioritizing are much more efficient for moderate and large-size conflict models. As shown on the left of the dialog box in Figure 4.16, either of these two techniques can be chosen to input a preliminary ranking of states. Both Option Weighting and Option Prioritizing have their limitations and might not catch all the details of the preferences. Therefore, it is usually recommended that the state ranking obtained from either Option Weighting or Option Prioritizing be refined by Direct Ranking.

For conflict models of very large size, on-screen Fine Tuning may not be practical, and thus can be inactivated by unchecking the corresponding checkbox on the dialog box. On the other hand, if a conflict model is small so that on-screen manipulation alone could be sufficient, then the preliminary ranking using Option Weighting or Option Prioritizing can likewise be skipped. The detailed features of these three techniques are introduced in the following sections. The Elmira conflict model has only nine states, and is in fact typical of the group for which preference information is best input by Direct Ranking. However, Option Weighting and Option Prioritizing are performed on this model to illustrate how each technique works. An advantage of using this conflict for illustration purpose is that this model is small enough that complete preference rankings can be displayed on one screen. Application of these approaches to obtain and represent ordinal preference information allows GMCR II to model real-world conflicts expeditiously, and analyze them effectively. Initial research on ordinal preferences representation, elicitation and processing was previously presented in [71].

### 4.5.2 *Option Weighting*

Option Weighting is straightforward and simple to use in practice. For a given decision maker, each option $o_k \in O$ can be assigned a numerical weight $W(o_k)$, which may be a positive or negative number, according to its importance and desirability to that decision maker. The more important an option, the greater the magnitude of the weight assigned to it. Negative weights indicate options that the decision maker prefers not be selected. Figure 4.17 shows GMCR II's dialog box for entry of option weights. In this dialog, the default weight for each option is set to 0 and listed beside the option. A user can simply click on the cell to edit the number as desired. When the edit focus moves to another cell, or the return key is pressed, the editing is considered finished. If some illegal characters (e.g. non-number key) are entered, GMCR II resets the input weight to its default value 0. Contained in the dialog box in Figure 4.17 is an illustrative set of option weights input for MoE in the Elmira conflict model. Concerned about the economic benefits a company like Uniroyal can offer the local community and in turn the province, MoE is influenced by the firm's ultimate threat - abandonment. Consequently, option 4,in which the Uniroyal abandons its Elmira operation, is assigned a negative weight with a relatively high magnitude of $-20$. It is reasonable to expect that MoE would be happy if Uniroyal accepts the current Control Order, and, almost to the same degree, would dislike Uniroyal to delay the procedure. Therefore, the weights on options 2 and 3 are set to be of equal magnitude, but one negative (-10) and one positive (10). Likewise, option 1 is given a small negative weight (-5), because MoE is somewhat reluctant to modify its original Control Order. The local government's option is assigned a weight 0, reflecting that MoE is not really sure whether its position is improved if the Local Government sticks to the original Control Order.



Option Weighting for "MoE"

Enter option [illegible]

| DMs | Options | Option Weight |
|---|---|---|
| MoE | 1. Modify | -5 |
| Uniroyal | 2. Delay | -10 |
| | 3. Accept | 10 |
| | 4. Abandon | -20 |
| Local Government | 5. Insist | 0 |

Figure 4.17: Dialog Box for Option Weighting

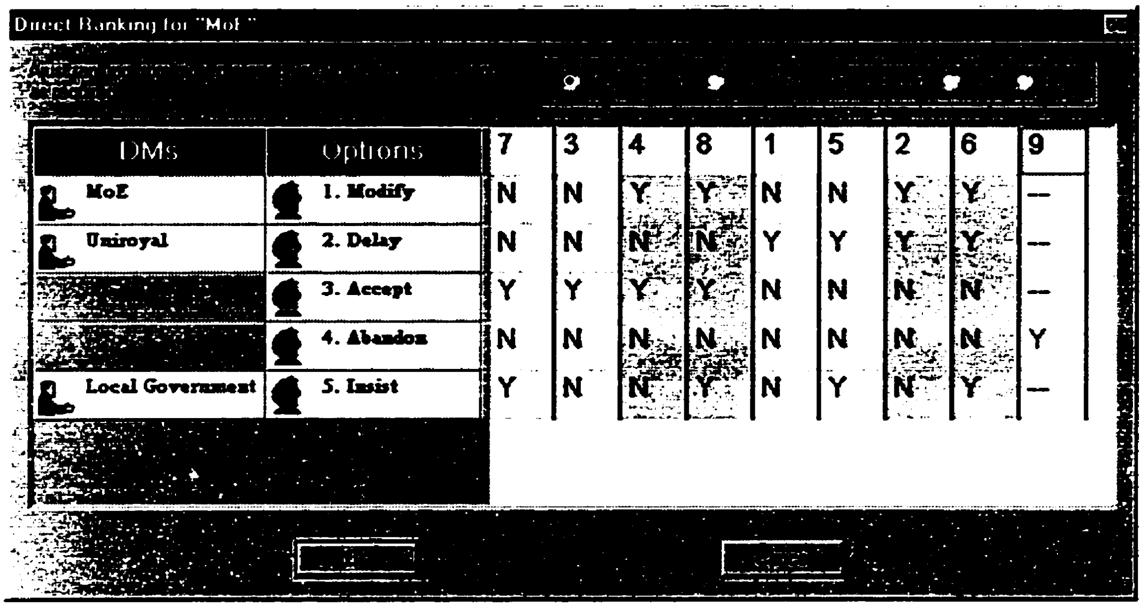
In Option Weighting, a score $M(u)$ for each feasible state $u \in \mathbf{U}$ is calculated based on the option weights assigned. It represents the focal decision maker's preferences over states as a consequence of his or her preferences on options. Analogous to many multiple criteria decision making (MCDM) problems, there are many possible combinations of weights that produce the same preference ranking. But Option Weighting has a "monotonicity" property - that is, if the weight on an option is increased, then states in which that option is chosen will either move upward in the ranking or stay in the same position.

In GMCR II, a natural scheme is chosen: for a given atomic state, the weights are summed across the options to obtain a score for that state

$$\forall u \in \mathbf{U}, M(u) = \sum_{o_k \in \mathbf{O}} W(o_k) \cdot u(o_k) \tag{4.14}$$

For a composite state, a representative atomic state that it contains is chosen to receive the score. In GMCR II, the practice is that the "$-$"s are considered "N" in calculating the score for a composite state. Subsequently, GMCR II ranks the states using quick-sort [1] from most preferred to least preferred based on their scores. For example, based on the option weights assigned as shown in Figure 4.17, Figure 4.18 displays the resulting state ranking generated by GMCR II for MoE. Of course, equally preferred states occur when scores are equal. A set if equally preferred states is indicated in GMCR II by a group of columns having a common colored backgrounds. In Figure 4.18, states 7 and 3, 4 and 8, 1 and 5, and 2 and 6 are each an equally preferred group. As can be seen, this ranking is quite close to the final ranking as given in Table 2.4. The Direct Ranking will take over the rest of the job.

It is not unusual that many techniques used in models designed to handle ordinal data, including option weighting, deal with the data in a cardinal fashion [16]. Note



| DMs | Options | 7 | 3 | 4 | 8 | 1 | 5 | 2 | 6 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| MoE | 1. Modify | N | N | Y | Y | N | N | Y | Y | — |
| Uniroyal | 2. Delay | N | N | N | N | Y | Y | Y | Y | — |
|  | 3. Accept | Y | Y | Y | Y | N | N | N | N | — |
|  | 4. Abandon | N | N | N | N | N | N | N | N | Y |
| Local Government | 5. Insist | Y | N | N | Y | N | Y | N | Y | — |

Figure 4.18: State Ranking for MoE in Elmira Conflict Model Obtained from Option Weighting

that the magnitude of the scores is not meaningful; scores are simply a useful device to determine relative preferences. It should also be noted that there are many preference orderings over states that cannot be constructed using option weighting. A detailed discussion on this issue can be found in [51].

### 4.5.3 *Option Prioritizing*

The Option Prioritizing approach in GMCR II [71] constitutes a generalization of the "preference tree" method originally suggested by Fraser and Hipel [27], and later reported upon by Hipel and Meister [40] and Fraser [23, 24]. The "preference tree" method was implemented in DecisionMaker [29]. However, due to unilaterally's particular implementation strategy and close coupling between modeling and analysis, the "preference tree" method is essentially "option-centered" as opposed to "statement-centered". Accordingly, a variety of restrictions appear in DecisionMaker which seriously handicap its capability to represent preference information. For example, in DecisionMaker, a logical conditionals must be interpreted as "if and only if"; logical connectives must be either AND's or OR's, but not a mixture of both; options appearing in the hypothesis of a conditional of statement must have appeared in a previous statement, and hence the first statement can never be conditional; etc. In contrast, GMCR II's Option Prioritizing is statement-centered, and none of the above limitations applies.

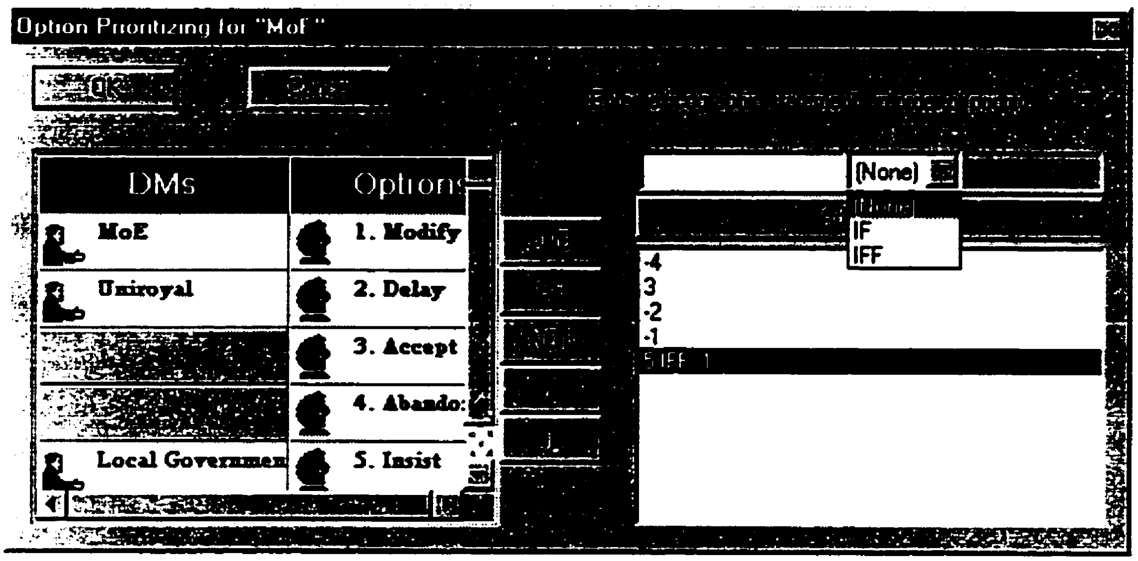
In Option Prioritizing, the user is asked to provide an ordered set of *preference statements* for each DM. Each preference statement $\Omega$ takes a truth value, either True (T) or False (F), at each particular state. A statement that has higher priority in determining preferences appears earlier in the set. The preferences over states can be determined in the following way:

Let $\{\Omega_1, \Omega_2, \ldots, \Omega_k\}$ be the set of statements. A state $u_1$ is *preferred to* a state $u_2 \neq u_1$ if and only if $\exists j$, $0 < j \leq k$, such that

$$\begin{array}{rcl} \Omega_1(u_1) & = & \Omega_1(u_2) \\ \Omega_2(u_1) & = & \Omega_2(u_2) \\ \vdots & \vdots & \vdots \\ \Omega_{j-1}(u_1) & = & \Omega_{j-1}(u_2) \\ \Omega_j(u_1) = T & \text{and} & \Omega_j(u_2) = F \end{array}$$

Figure 4.19 shows the input dialog box in GMCR II for eliciting a set of preference statements for a focal DM. The operations for this dialog box are similar to those of Figure 4.8. Extra controls in the input area include a combo box for specifying whether a conditional or a non-conditional statement is to be input, and, if conditional, what type of condition to be used. If the user chooses to input a non-conditional statement (default), the edit box for entering the condition, becomes irrelevant and therefore disabled. One important attribute of the set of preference statement is that order matters, so GMCR II allows the user to move a highlighted list item to any position in the list by dragging. The title of the focal DM is shown on the title bar, so the user is reminded whose preference information is being entered.

In GMCR II, preference statements are expressed in terms of options and logical connectives. A preference statement can be non-conditional, conditional, or bi-conditional. A non-conditional statement is expressed as a combination of available option numbers, plus connectives including negation ("not" or −), conjunction ("and" or &) and disjunction ("or" or | ). Brackets ("(" and ")") are used to control the priority of operations in a statement. In this sense, non-conditional preference statements are no different from direct specification statements as introduced in



-4
3
-2
-1
5 IFF 1

Figure 4.19: Dialog Box for Entering Option Prioritizing Preference Statements

Section 4.3.2.4. Negation (−) preceding an option number indicates that the decision maker prefers the option not to be taken. For example, the first preference statement (−4) for MoE shown in Figure 4.19 means that option 4, abandonment of the Elmira operation by Uniroyal, is what MoE prefers not to happen. The relative *importance* of a preference statement is reflected by its position in the list: a statement that occupies a higher place in the list, is more important in determining the decision maker's preferences. In the Elmira example, the ultimate threat of Uniroyal's abandonment of its Elmira operation, which would have a negative effect on the province's economy, is MoE's greatest concern. This statement is true (T) at any state for which option 4 is not selected, and false (F) otherwise. In the Elmira conflict model, for instance, this statement is true (T) at each of the first eight states in Table 4.1 and false (F) at state 9.

A conditional or bi-conditional statement consists of two non-conditional statements connected by an "IF" or "IFF". For example, the last statement in the list in Figure 4.19 is a bi-conditional statement "5 IFF -1", which means that MoE would like to see Local Governments's support of the original Control Order if and only if itself chooses not to modify the Control Order. Likewise, the third statement for Local Government, "3 IF -1", as will be shown in Table 4.7, is a conditional statement meaning that Local Government would like to see Uniroyal accept the Control Order (3), provided it is not modified (-1). The truth value of a conditional or bi-conditional statement at a state depends on the truth values of its two non-conditional components according to the conditional or bi-conditional truth tables defined in mathematical logic [78]. For example, according to the conditional truth table, statement "3 IF -1" is considered to be true at any state in which option 1 is selected (and hence -1 is false).

Even though GMCR II has the capacity to handle quite complicated logical combinations of options, most human DMs' preferences tend to be expressed in rather simple preference statements. Compare MoE's preference statements with its option weights as specified and interpreted in Section 4.5.2, and one can find that they almost reflect exactly the same reasonable preference information, except when option 5 is considered. Option weighting cannot assign a non-zero weight to option 5 because it cannot express conditional information. We will see that appropriate lexicographic preference statements for MoE produce a state ranking identical to the final ranking in Table 2.4– no fine-tuning is needed. Generally speaking, Option Prioritizing can express richer and more flexible preference information than option weighting, and hence should be preferred by users who feel comfortable with this methodology. The lexicographic preference statements for the other two DMs, Uniroyal and Local Government, are given in Tables 4.6 and 4.7. respectively, together with interpretations. These statements all make good sense in the context of the conflict model, and the resulting state rankings are generally close, and often identical, to the final rankings. Note that the first statement in Table 4.6. and third and fourth statements in Table 4.7 would not be allowed in DecisionMaker [29].

A tree presentation of ordinal preferences can be described as follows. For a particular DM, any set of states or alternatives can be split into two subsets, where any state in the first subset is preferred to any state in the second subset. This splitting process can be applied successively to the subsets, forming a binary tree. A complete ranking can eventually be achieved. Each time a set or subset of states is split (*i.e.* at each level of the binary tree), a preference statement serves as the criterion governing this bifurcation. In other words, a set of preference statements controls the structure of the preference tree. The tree presentation that corresponds to the set of preference statements in Table 4.7 is shown in the upper part of Figure

Table 4.6: Lexicographic Preference Statements and Interpretation for Uniroyal

| Statement | Interpretation |
|---|---|
| 3 *IFF* 1 | Uniroyal would be ready to accept the Control Order if and only if it has been modified to be more acceptable to Uniroyal |
| −4 | Uniroyal does not like to abandon its Elmira operation |
| −5 | Uniroyal hates to see Local Government insisting that the original Control Order not be modified |
| 2 *IFF* −5 | Uniroyal would like t·· delay the procedure if and only if Local Government's attitude is softened |

Table 4.7: Lexicographic Preference Statements and Interpretations for Local Government

| Statement | Interpretation |
|---|---|
| −4 | Concerned about the negative consequence on the local economy, Local Government does not like to see Uniroyal abandon its Elmira operation. |
| −1 | Local Government prefers that the original Control Order not be modified. |
| 3 *IF* −1 | Local Government likes to see Uniroyal accept the Control Order, if it is an unmodified one. |
| 5 *IF* 1 | Local Government would insistently ask for the original Control Order if MoE tends to modify it. |
| −2 | Local Government does not like to see the delay of the procedure. |
| 5 | After all, Local Government still tends to insist on the original Control Order. |

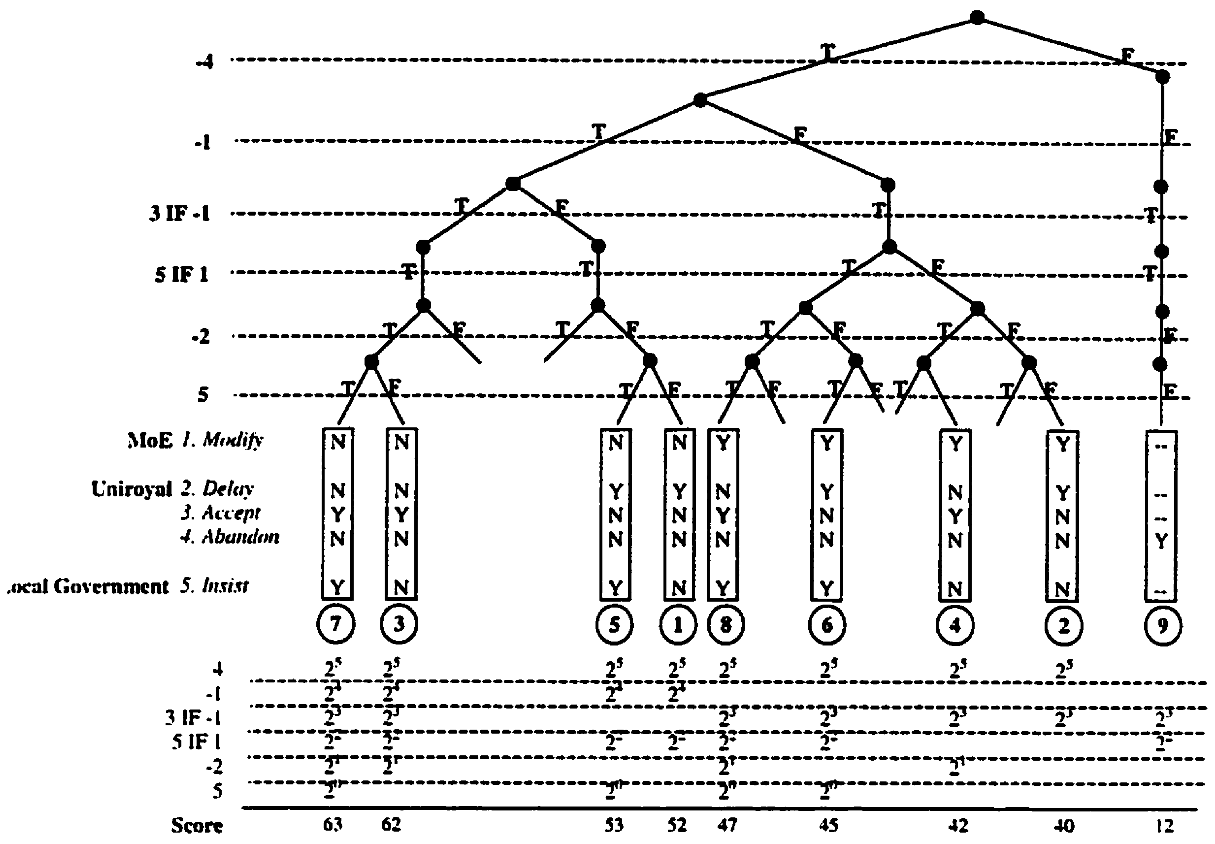


Figure 4.20: Tree Presentation of Option Prioritizing

4.20. Note that the tree has some "empty leaves" because of the prior removal of infeasible states. As can be seen, a few preference statements can represent a rather complex tree. In other words, lexicographic preference statements are a compact representation of a preference tree. Another observation here is that a conditional statement (e.g. "3 IF -1"), as opposed to a bi-conditioinal (iff) statement, can help to localize preference judgements.

An equivalent scheme that can result in the same ranking as in the tree presentation is to assign a "score" $\Psi(u)$ to each state according to its truth values when the statements are applied. Assume $k$ is the total number of statements that have been provided, and denote by $\Psi_j(u)$ the incremental score to state $u$ based upon statement $\Omega_j, 0 < j \leq k$. Define

$$\Psi_j(u) = \begin{cases} 2^{k-j} & \text{if } \Omega_j(u) = T \\ 0 & \text{otherwise} \end{cases}$$

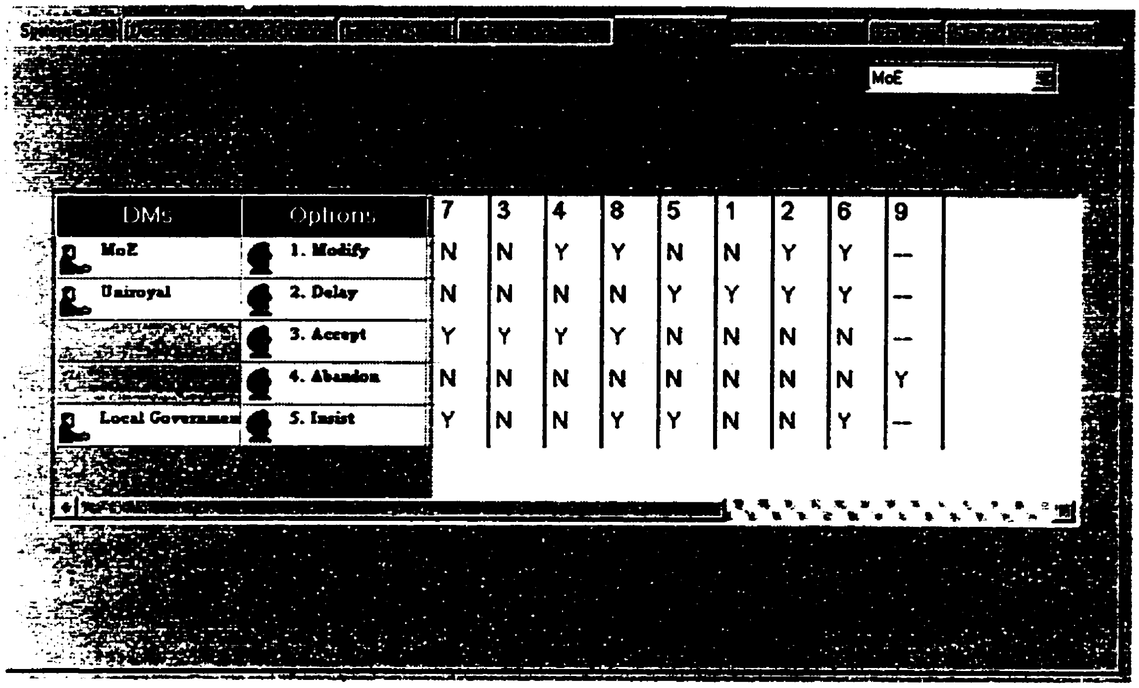
and

$$\Psi(u) = \sum_{j=1}^{k} \Psi_j(u)$$

The states can then be sorted according to their scores using quick-sort [1], which will result in exactly the same ranking as that from the tree presentation. This "scoring" scheme is illustrated in the lower part of Figure 4.20. The implementation of Option Prioritizing in GMCR II is based on this scheme. Again, it is emphasized that even though a cardinal "score" is involved, it only plays a temporary role in determining the ranking; GMCR II requires only ordinal preference information. The scores given at the bottom of Figure 4.20 as well as the preference tree at the top list the states from the most preferred on the left to the least preferred on the right. This resulting ranking is identical to the final ranking given in Table 2.4.

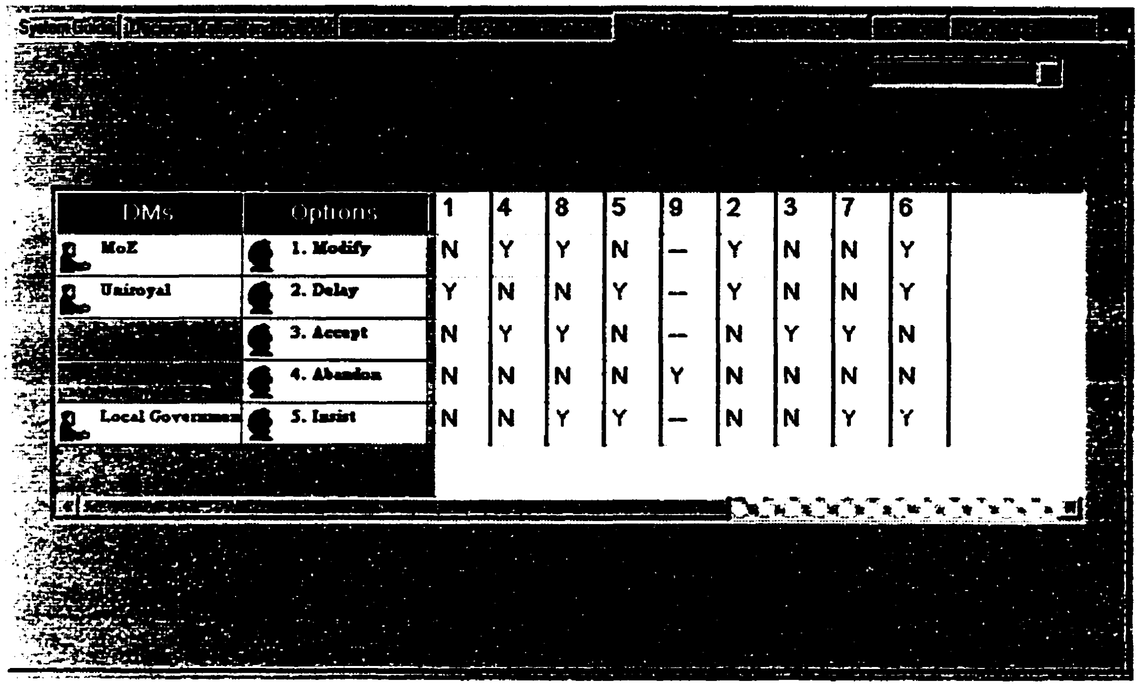
The state rankings GMCR II produces for MoE and Uniroyal, based on the lexicographic preference statements for each of them provided earlier, are shown in Figures 4.21 and 4.22, respectively. Compared with the final rankings in Table 2.4, MoE's ranking is identical, and only a single preference reversed occurs on a pair of states (2 and 7) in Uniroyal's ranking, which can be easily fixed by Fine Tuning.

As for how to decide whether a preference statement returns true at a particular state in the program, the efficient pattern matching techniques (4.2, 4.5) play a key role again. Since a conditional or DecisionMaker statement is comprised of a pair of



| DMs | Options | 7 | 3 | 4 | 8 | 5 | 1 | 2 | 6 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| MoE | 1. Modify | N | N | Y | Y | N | N | Y | Y | — |
| Uniroyal | 2. Delay | N | N | N | N | Y | Y | Y | Y | — |
| | 3. Accept | Y | Y | Y | Y | N | N | N | N | — |
| | 4. Abandon | N | N | N | N | N | N | N | N | Y |
| Local Governmen | 5. Insist | Y | N | N | Y | Y | N | N | Y | — |

Figure 4.21: State Ranking for MoE Resulting from Option Prioritizing



| DMs | Options | 1 | 4 | 8 | 5 | 9 | 2 | 3 | 7 | 6 |
|---|---|---|---|---|---|---|---|---|---|---|
| MoE | 1. Modify | N | Y | Y | N | — | Y | N | N | Y |
| Uniroyal | 2. Delay | Y | N | N | Y | — | Y | N | N | Y |
|  | 3. Accept | N | Y | Y | N | — | N | Y | Y | N |
|  | 4. Abandon | N | N | N | N | Y | N | N | N | N |
| Local Government | 5. Insist | N | N | Y | Y | — | N | N | Y | Y |

Figure 4.22: State Ranking for Uniroyal Resulting from Option Prioritizing

non-conditional statements plus the connective, its truth value can be easily decided based on the conditional or DecisionMaker truth table, once the truth values of its component unconditional statements are returned. The problem is, therefore, to focus on how to calculate the truth value of an unconditional statement at a state. As pointed out earlier, an unconditional preference statement is no different from direct specification statements introduced in Section 4.3.2.4. As discussed in Section 4.3.2.4, such a statement corresponds to one or more patterns which represent all the states at which the statement is true. GMCR II automatically transforms a statement into its corresponding patterns, and then uses pattern-matching (4.2, 4.5) to determine whether a statement is true for a state.

Using sufficiently complex preference statements, any ordering of the states can be represented by Option Prioritizing. However, some rankings may require many statements – the maximum is one per state. Nevertheless, Option Prioritizing is especially useful for large models, where the DMs' preferences typically fall into regular and consistent patterns.

### 4.5.4 *Direct Ranking*

The Direct Ranking or Fine Tuning in GMCR II allows on-screen manipulation to refine the ranking list for a given decision maker. By invoking one of the radio boxes in the upper-right of the dialog box in Figure 4.18, the user can perform the following operations:

**Move Group** Moves a group of equally preferred states, including a single-state group, from one location in the ranking to another, by dragging the group to the desirable location.

**Move State within Group** To move a state within an equally preferred group from one location to another (within the same group). This operation is useful when combined with Splitting.

**Split** To split an equally preferred group into two groups. In this mode, when the mouse is on the boundaries within the group, the cursor changes to a scissors shape, and a left-click splits the group into two, which are distinguished by different background colors. The cursor resumes to default afterward.

**Join** To highlight a range of contiguous states to create an extra equally preferred group, which overrides the initial groups within the range.

These four basic operations enable the user to achieve any desired ranking via on-screen direct manipulation. It is worth pointing out that these four user-friendly functionalities involve a tremendous amount of implementation effort. The existence of *indistinguishable states* further justifies the necessity of direct ranking. For example, Option Weighting (4.14 can only be performed on a "representative" state of the coalesced group, resulting in a somewhat arbitrary location of the indistinguishable states. Direct Ranking is thus necessary to adjust the ranking of the indistinguishable states.

### 4.5.5 *Summary*

In this section, a novel and flexible methodology for conveniently eliciting a domain expert's ordinal preference ranking is presented. This methodology is valuable in the modeling of strategic conflicts, when the user's assessment of the preferences of each DM over all states is generally required. The methodology is comprised of three techniques: Option Weighting, Option Prioritizing, and Direct Ranking.

Option Weighting and Option Prioritizing can be utilized for moderate- and large-size models (many different states to be ranked). Direct Ranking can be used for small models or for fine tuning the preliminary ranking obtained from either option weighting or option prioritizing.

## 4.6 Graph-Based Modeling

In some situations, decision makers may prefer to enter a model by directly specifying the states, and the possible movements among them, based upon their own perceptions of the dispute. For instance, high-level executives or policy makers often like to think directly about possible states in free form, and about how more preferred outcomes can be achieved, without worrying about the details of each state's definition. Sprague and McNurlin [84] point out that direct manipulation interface (DMI) is most suitable for EISs (Executive Information Systems) – DSSs for executives. According to [84], in an EIS: only highly summary performance data are accessed; graphics should be used to display and visualize the data; only a minimum amount of analysis for modeling is required beyond the capability to "drill down" in summary data to examine components. Graph-based modeling of states may be very helpful in these cases, especially for smaller conflicts at early stages of development, and when the modeling is carried out in a "brain-storming" session. The Graph Model for Conflict Resolution, which takes states as basic building blocks, makes graph-based modeling possible.

Graph-based modeling permits a user to conveniently input a conflict model by drawing graphs directly on the screen, and to visually interpret the output. Each component (vertex or arc) of a graph would respond to the user's command, via a pointing device (usually a mouse, or maybe a touch-screen interface). Highly summarized models, that may be appropriate for higher-level executives or policy makers, are usually smaller. Graph-based modeling is not intended for large models, not only because the graphs would be difficult to display, but also because over-ambitious modeling could make the model's behavior inexplicable and would overwhelm rather than help the user.

An annotated bibliography by [18] gives a comprehensive review of algorithms for graph drawing. To produce aesthetically pleasing drawings of graphs is actually a difficult task [18]. A graph-drawing software development tool is needed for this key component of graph-based modeling. The desired development tools can be in the form of either a graph drawing tool that has an import/export interface, or a graph drawing development toolbox or library. For flexibility, the latter is preferable. Sander [80] summarized the commercial and non-commercial graph-drawing tools and libraries currently available. In fact, most graph drawing tools target non-Windows platforms, and many are for special purposes such as displaying control flow, visualizing program structures, etc. Moreover, the existing tool designs the nodes and arcs as components that can interact with the user. Therefore, a successful implementation of the graph-based modeling in GMCR II is subject to the availability of flexible and general-purpose graph drawing development tools for a Windows platform. Nonetheless, some basic design ideas can be discussed now, which may be helpful for future developments.

- Representation of states and state transitions in graph-based modeling can be done in a straight-forward manner. Clicking on the client area should cause a small circle to appear as a representation of a state (node). Dragging between a pair of states should create a line or curve connecting the states with an arrow pointing from the initial state to the target state, representing a directional state transition. A second arrow with the reverse direction can be added if desired. In this way, a graph model can be built for a focal DM. The chosen graph-drawing tool should be capable of updating the layout of the graph to keep it tidy whenever new components are added. GMCR II requires structural information - states and transitions - in order to build an internal analytical model for the analysis stage. A hit-test [72] function is required such that each node and arc can be treated as a control that can interact with the user. For example, a user can click on a node to enter a label for the node, or to invoke an popup window at any time to enter or review a brief description of the scenario represented by that node. The description may or may not be in terms of DM/option format. Graphs for different DMs (such as Figure 2.1 (a) (b) (c) ) can be drawn in separate windows. An integrated graph (like Figure 2.1 (d)) can be generated, where moves by different DMs are distinguished by different arc colors. In an integrated graph, which puts all the structural information of a model in same screen, normally there is only one arc connecting a pair of nodes. The only exception is a common move, in which case parallel arcs with different colors could be used. It should be the graph drawing tool's responsibility to coordinate layout between the integrated graph and the individual graphs such that the locations of the nodes and relevant arcs remain unchanged when the view is switched between two graphs.

- States and transitions are actually input while graphs are being drawn. This graphical information is then converted by GMCR II into a format which can be processed by the analysis engine. It should be pointed out that graph-based modeling can better exploit the flexibility of graph model. In option form modeling, even though the irreversibility specification provides a fair amount of flexibility in expressing state transitions, the feasible states and state transitions are determined before the structural model is built. In contrast, graph-based modeling supports a truly interactive modeling procedure. The user may identify new states not previously envisioned while contemplating the possible moves from an existing state. Likewise, when a new node (state) is added, he or she may discover moves not thought of until the graph is presented. This "modeling-while-drawing" approach could be very effective and helpful in a "brain-storming" setting.

- Sometimes, while carrying out graph-based modeling, the user may find that one state has become irrelevant, or that several scenarios are effectively indistinguishable. Thus state removal and state coalescing are required. In the graph-drawing interface, node deletion or node merging can be used. These can be done by combining the hit-test feature with a toolbar function, a shortcut, or a context menu. For node merging, a multiple highlighting function is also needed. When a node deletion or node merging is carried out, the arcs incident on the node should be processed accordingly. The job of updating the layout to maintain an aesthetically pleasing appearance should be undertaken by the graph drawing tool. GMCR II can update its internal structural model to reflect the changes.

- Kilgour *et al.* [52] and Fang *et al.* [20] present a general preference structure for the Graph Model for Conflict Resolution, explaining how the Graph Model for Conflict Resolution can handle transitive or intransitive preferences. However, the transitivity assumption can save a user a considerable amount of effort in specifying the preferences, and therefore is recommended for practical use. A graph-based preference specification method is proposed here for the graph-based modeling. The user can specify a focal DM's preference or indifference over two states by assigning a directed or undirected arc between the nodes for the two states, which will be retained by the system. With the transitivity assumption, users do not need to specify preferences over each pair of states. Actually, the most efficient preference ordering over $\mu$ states requires only $\mu - 1$ edge assignments. This method can prevent the assignment of a pair-wise preference contradicting the previous preference input in the sense of transitivity. It can also remind the user when the preference arc assignments are sufficient for the ranking. Figure 4.23 shows the flowchart for this graph-based preference specification method. For each node $u_k \in \mathrm{U}$, the system keeps track of two lists, $L(u_k)$ and $E(u_k)$, which contain the states that have been judged to be less preferred or equally preferred to $u_k$, respectively. As can be seen from Figure 4.23, nodes marked earlier are more preferred. A preview window will be useful to enable the user to look at his or her preference specifications. One way is to show the graph with each node labeled by the focal DM's preference level. Another approach is to present all the UIs for the focal DM on the graph. The user should also be allowed to cancel or modify some pair-wise comparison judgements.

- A graph-based model can be connected to the analysis engine and then displayed again for interpretation of the analysis results. It is certainly much more powerful for interpretation purposes, especially for status quo analysis and visualization of the stability analysis on a state of interest. Most existing graph drawing tools [80] take the data equivalent to the reachability information as input and generate a graph layout based on the information. Consequently, another option is to export the reachability result from an option-form model, which is already state-based, to a suitable graph-drawing tool to draw the desired graph.

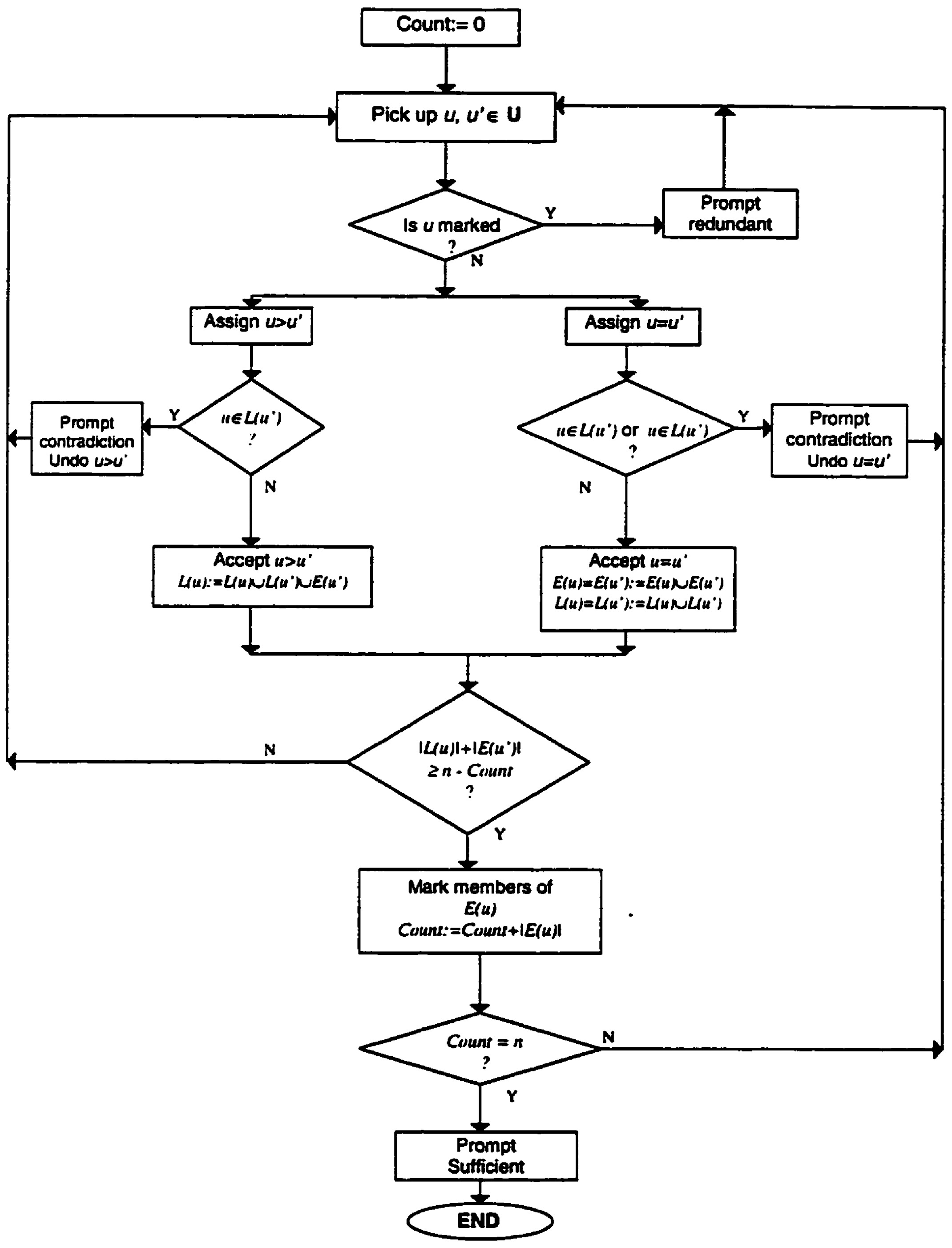

Count:= 0
Pick up $u$, $u' \in$ U
Is $u$ marked ?
Y
Prompt redundant
N
Assign $u>u'$
Assign $u=u'$
$u \in L(u')$ ?
Y
Prompt contradiction Undo $u>u'$
N
$u \in L(u')$ or $u \in L(u')$ ?
Y
Prompt contradiction Undo $u=u'$
N
Accept $u>u'$
$L(u):=L(u) \cup L(u') \cup E(u')$
Accept $u=u'$
$E(u)=E(u'):=E(u) \cup E(u')$
$L(u)=L(u'):=L(u) \cup L(u')$
N
$|L(u)|+|E(u')| \geq n$ - Count ?
Y
Mark members of $E(u)$
Count:=Count+$|E(u)|$
Count = $n$ ?
N
Y
Prompt Sufficient
END

Figure 4.23: Graph-Based Preference Specification

# Chapter 5

# Analysis of Strategic Conflicts

A conflict model constructed via procedures described in Chapter 4, is a basic framework within which strategic interactions among the DMs can be analyzed in detail. The analysis engine of GMCR II can calculate the stability of every state from each DM's viewpoint. This in turn permits the straightforward calculation of the set of equilibria, consisting of the states that are stable for each DM.

The results of the stability analysis can then be interpreted by the user in order to understand better the real-world conflict. To facilitate the interpretation and refinement of the stability analysis, various follow-up analyses can then be carried out.

## 5.1 Stability Analysis

The stability analysis of a conflict is carried out by determining the stability of each state for every DM. A state is *stable* for a DM if and only if that DM has no incentive to deviate from it unilaterally, under a particular behavior model, usually referred to as a ***stability definition*** or ***solution concept.*** A state is an ***equilibrium*** or possible resolution under a particular stability definition iff all DMs find it stable under that stability definition. A broad range of stability definitions (Table 2.1, Section 2.2.3) are implemented in GMCR II, representing diverse human decision making characteristics. For example, stability definitions embody several different attitudes to strategic risk, as well as many levels of foresight. This range of stability definitions gives the analyst a clear picture of how the conflict can evolve, and how the DMs' decision styles and approaches are reflected in the evolution. Thus, the analyst's judgement and experience can be integrated into the modeling, interpretation, and prediction process.

### 5.1.1 *GMCR I as Analysis Engine*

GMCR I [20, Appendices] was the first computer program to carry out stability analysis under the paradigm of the Graph Model for Conflict Resolution. It was written in the C language and runs under the DOS operating system. Using DOS's I/O redirection, GMCR I's input and output are ASCII text files. The input text file requires information about

- the number of DMs involved in the conflict model;
- the number of states in the model;
- reachable list of each state for each decision maker, and
- ordinal payoff for each state for each decision maker.

The output file contains:

- a summary of equilibria for each of the stability types listed in Table 2.1;

- the stability of each state, under each stability type, for each DM.

In both input and output files, a state or a DM is identified by number. Since GMCR I takes an available conflict model as given, and does not have an interactive user interface, it has serious limitations on both the front and the back ends. However, the analysis part of GMCR I is quite powerful. It efficiently carries out an exhaustive stability analysis for a wide range of solution concepts. For this reason, GMCR I and its improved versions was used as an interim analysis engine for the earlier versions of GMCR II, until GMCR II's own analysis engine was developed using object-oriented design and programming. The modeling information established via the input interface of GMCR II is converted into a GMCR I input file such that GMCR I can take over stability analysis. The use of GMCR I as an interim analysis engine contributed greatly to the development of GMCR II, especially because it allowed the modeling subsystem and analysis subsystem to be implemented separately, and hence minimized the coupling between these two discrete subsystems (Section 3.2), which is considered good software design practice for both structured design and object-oriented design [90].

The GMCR II analysis engine was developed by improving and expanding, the original GMCR I program. While GMCR I's analysis is powerful in that a variety of stability definitions are incorporated, its capacity to handle larger models is limited. When more than two DMs are involved, GMCR I required [20, Appendices] that the maximum number of states be 100, and maximum number of DMs be 5. Of course, state number is not the only index that reflects the amount of memory that GMCR I requires – the length of reachable lists matters too. Sometimes, therefore, the program fails before reading these limits.

A main reason for size limitations in GMCR I is that not enough attention was paid to memory management, even though linked lists rather than a multi-dimensional array were used to represent allowable transitions [20, Chapter 2]. But in fact, very few human users would be willing to prepare manually a model with 100 states and 5 DMs anyway. Imagine 500 reachable lists and 5 preference rankings over 100 states without an interactive input interface! But when serving as an analysis engine in GMCR II, the situation for GMCR I is different.

Improvements that were made to GMCR I in terms of memory management include the following:

- Highly inefficient fixed-size multi-dimensional arrays used to represent the stability results were removed. Instead, a state's stability under one solution concept is simply recorded in one bit - a 1-bit indicates stable and a 0-bit unstable. A DOUBLEWORD is used to record a state's stability under the stability definitions of R. GMR, SMR, SEQ, NM and up to 10 levels of limited-move stability. In total 14 bits ($L_1$ overlaps with $R$) were used, with 18 bits reserved for future introduction of new solution concepts. Bit-wise operations are used to make the recording and retrieval of the stabilities simple and fast.

- Large numbers of redundant intermediate variables were replaced by a very small number of essential variables.

- Variables that previously occupied the stack and frequently caused stack-overflow were replaced by pointers, with memory dynamically allocated to heap.

- GMCR I was ported to Windows 3.1, on which global memory allocations can be used under protected mode [72] via rather complicated (and unstable) processes. Fortunately, in Windows 95's 32-bit memory addressing capacity and virtual memory management [73] eventually make this relatively hassle-free.(A Windows 95 version of GMCR II was started in late 1997.) Nonetheless, the previous memory-saving measures remain valid and worthwhile.

GMCR I and its improved versions have now been replaced by an O-O program developed using Visual C++/MFC. However, useful algorithms and techniques used in the improvement greatly contributed to the development and testing of the current version.

### 5.1.2 *Illustrative Implementation for Stability Analysis*

As can be seen from Section 2.2.3, the stability analysis for any of the stability definitions is basically a systematic examination of the permissible moves and countermoves by the DMs during possible evolution of the conflict. The basic components that underlie these stability definitions are individual unilateral moves, group unilateral moves and anticipation. The group unilateral move is taken as an example below to illustrate the implementation, because this concept is also essential to coalition analysis and status quo analysis which will be discussed later in this chapter.

The group reachable list for $H \subseteq N$ from state $u$, $\mathbf{S_H}(u)$, is defined recursively in Definition 2.1.

The original implementation in GMCR I uses a linked list, as shown in Figure 5.1, to represent $\mathbf{S_H}(u)$ [20]. In Figure 5.1, $\omega_{\mathbf{H}u}(u') = j$ if $\Omega_{\mathbf{H}u}(u') = \{j\}$ and $\omega_{\mathbf{H}u}(u') = 0$ if $|\Omega_{\mathbf{H}u}(u')| > 1$. Note that a state $u$ is always identified by its index in the implementation. The GMCR I algorithm is transcribed in pseudocode as listed in Figure 5.2.

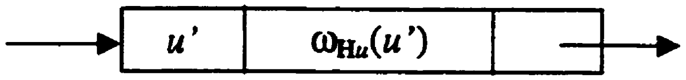


Figure 5.1: A Three Element Cell in the Linked List

This complicated algorithm simply adds to the link list all states that can result from any sequence of UMs by some or all members of the group **H**. But a large amount of effort is devoted to ensuring

- that the same state is not added twice, and
- that states resulting from two consecutive moves by the same DM are not included.

For these latter purposes, a large amount of searching and comparing is required, and making the use of the linked list collection shape somewhat awkward. For example, the simple condition *if* $u'' \notin \mathbf{S_H}(u)$ in the above algorithm adds considerable complexity, because the entire list must be searched to ensure that this condition holds. The program becomes very inefficient as the list builds up.

Table 5.1 summarizes the characteristics of the three commonly available collection shapes: list, array, and map. Comparing them on all three applicable features listed on the right of the table, one can find that the map collection shape is the most suitable for storing group UMs.

A map is a dictionary-based collection with hash table implementation [1], which maps unique keys to values. It only takes constant time, on the average, to insert, delete or retrieve a key-value pair (element) into the map. Iteration over all the

```
for $j \in \mathbf{H}$
...for $u' \in \mathrm{S}_j(u)$
......if $u' \notin \mathrm{S}_{\mathbf{H}}(u)$ then
.........$u' \in \mathrm{S}_{\mathbf{H}}(u)$;
.........$\omega_{\mathbf{H}u}(u') = j$;
......else
.........$\omega_{\mathbf{H}u}(u') = 0$;
......endif
...endfor
endfor
flag = 1;
while (flag == 1)
...flag = 0;
...for $u' \in \mathrm{S}_{\mathbf{H}}(u)$
......for $j \in \mathbf{H}$
.........for $u'' \in \mathrm{S}_j(u')$
............if $\omega_{\mathbf{H}u}(u') \neq 0$ then
...............if $j \neq \omega_{\mathbf{H}u}(u')$ then
..................if $u'' \notin \mathrm{S}_{\mathbf{H}}(u)$ then
.....................$u'' \in \mathrm{S}_{\mathbf{H}}(u)$;
.....................$\omega_{\mathbf{H}u}(u'') = j$;
.....................flag=1;
..................else
.....................if ($\omega_{\mathbf{H}u}(u'') \neq 0$) then
........................if ($j \neq \omega_{\mathbf{H}u}(u'')$) then
...........................$\omega_{\mathbf{H}u}(u'') = 0$;
...........................flag=1;
........................endif
.....................endif
..................endif
...............endif
............else
...............if $u'' \notin \mathrm{S}_{\mathbf{H}}(u)$ then
..................$u'' \in \mathrm{S}_{\mathbf{H}}(u)$;
..................$\omega_{\mathbf{H}u}(u'') = j$;
..................flag=1;
...............else
..................if ($\omega_{\mathbf{H}u}(u'') \neq 0$) then
.....................if ($j \neq \omega_{\mathbf{H}u}(u'')$) then
........................$\omega_{\mathbf{H}u}(u'') = 0$;
........................flag=1;
.....................endif
..................endif
...............endif
............endif
.........endfor
......endfor
...endfor
endwhile.
```

Figure 5.2: Original Algorithm for $\mathrm{S}_{\mathbf{H}}(u)$

Table 5.1: Collection Shape Features

| **Shape** | **Ordered?** | **Indexed?** | **Insert an element** | **Search for specified element** | **Duplicate elements?** |
|---|---|---|---|---|---|
| List | Yes | No | Fast | Slow | Yes |
| Array | Yes | By int | Slow | Slow | Yes |
| Map | No | By key | Fast | Fast | No (keys)<br>Yes (values) |

elements in the map is also allowed. In GMCR II's new analysis engine, a map class is built which takes the target state of a move (must be unique) as key, and the DM who controls the move (not necessary unique) as value. A *SetAt(key, value)* operation of this class first looks up the key (fast, see Table 5.1). If the key is found, then the corresponding value is changed; otherwise a new key-value pair is created. With all the suitable features of this new data structure, the algorithm is greatly streamlined, as can be seen from Figure 5.3.

The set of group UMs $\mathbf{S_H}(u)$ is a important concept used not only in the implementation of a series of stability definitions, but also in the coalition analysis and the status quo analysis to be introduced later this chapter. When used for stability analysis purposes, the above algorithm can actually be further optimized. For example, in calculating GMR stability, before $u'$ or $u''$ is added to the map, it can be checked whether whether the move to it is an effective sanction to the original UI. If yes, the loop can be broken, and the map does not grow further.

This improved algorithm for group UI, the efficient calculation of reachable lists (Section 4.4.3), and more efficient implementation of stability definitions, enable

```
for $j \in \mathbf{H}$
   for $u' \in \mathbf{S}_j(u)$
      $\mathbf{S_H}(u).SetAt(u',j)$;
   endfor
endfor

while $(|\mathbf{S_H}(u)| \neq c)$
   for $u' \in \mathbf{S_H}(u)$
      for $j \in \mathbf{H}$
         for $u'' \in \mathbf{S}_j(u')$
            $\mathbf{S_H}(u).SetAt(u'',j)$;
         endfor
      endfor
   endfor
   $c = |\mathbf{S_H}(u)|$;
endwhile.
```

Figure 5.3: Improved Algorithm for $\mathbf{S_H}(u)$

GMCR II to carry out stability analysis effectively and efficiently. The biggest real-world conflict model that has been documented is the Trade-in-Services model hipe90b, hipe94 with 184,320 feasible states, which was analyzed by DecisionMaker. GMCR II successfully analyzed the model and produced a stability result identical to the DecisionMaker results on Nash and SEQ (the only stability types that DecisionMaker can handle). The performance of GMCR II is obviously superior to DecisionMaker.

### 5.1.3 *Customized Equilibrium Types*

GMCR II's analysis engine implements a range of stability definitions, each representing a particular pattern of behavior, including foresight, attitude toward risk, etc.. as characterized in Table 2.1. Therefore, which stability definition(s) is suitable for a certain DM depends on the characteristics or behavior patterns of that DM.

Consequently, it is quite possible that DMs involved in a conflict will have different behavior patterns, and thus that different stability definitions will be required.

It is commonly received that a state is an equilibrium if it is stable for all the DMs in the model. However, it has so far been interpreted as: a state is an equilibrium *under a particular stability definition* if it is stable for all DMs under that particular stability definition. This interpretation implies that the DMs in conflict share the same behavior pattern, which is neither good nor necessary.

In the current applications of graph model for conflict resolution, equilibrium types are the same as stability types. For example, a state said to be SEQ equilibrium is SEQ stable for all DMs. A much richer variety of equilibrium types can and should be defined by allowing different DMs to use different stability types in equilibrium. For example, an equilibrium state could be SEQ stable for one DM, NM stable for another, etc.. To incorporate these new equilibrium types is not difficult, since the stability analyses remain the same. This requires only a naming and specification mechanism, which will be discussed further in next chapter.

The question that would affect the stability analysis, esp. its workload is: now that only a few solution concepts can possibly apply to a DM, can those be specified before the running of the analysis engine so that the non-applicable stability types can be left out in the first place? The answer is positive. However, except for *huge* models, the exhaustive analysis in GMCR II over the stability types can be done very fast in any case. Therefore the approach currently adopted is to do the exhaustive analysis first, and then the user can choose to display the applicable ones (Section 6.3).

## 5.2 Coalition Analysis

Coalition formation constitutes a common sociological phenomenon in strategic conflicts involving more than two DMs, especially in group decision and negotiation situations. Coalition analysis is a methodology that tries to predict which coalitions are likely to form, and which coalitions would benefit or harm a certain DM, by studying the impact of the coalition on the outcome. The main objective of this section is to present a new perspective on coalition analysis, one that adds an important dimension to the methodology of the Graph Model for Conflict Resolution as well as the decision support system GMCR II.

A coalition is a subset of two or more DMs who coordinate their actions in some way. The various approaches to coalition analysis constitute models of joint action by a coalition, usually leading to conclusions about how coalitions change the structure of a strategic conflict. Understanding the effect of coalitions is important, as coalitions have been observed frequently in military, political, economic, legal, environmental, and other disputes involving three or more DMs. Some coalitions seem to be extremely durable; others form and then dissolve, or fail to become operational due to inability to agree an joint actions. Finally, the threat of a coalition can be of strategic importance, even if the coalition itself never forms.

### 5.2.1 *Previous Work*

Early work on coalition analysis applicable to the ordinal approach to modeling preferences includes that of Brams [12] who argued that coalitions form in order to "win", that is, the existence of coalitions reflect strategic possibilities. Arrow's famous Impossibility Theorem [4] shows that, working ordinally, it is difficult in general to define the preferences of a coalition. Some insightful discussions on coalitions in complex decision situations can be found in [74, Ch.4].

Within the framework of conflict resolution methodologies, coalition analysis has been carried out from two perspectives. Kuhn *et al.* [63] proposed a state-based metric, measuring the similarity of preferences among the members of a proposed coalition as an indicator of the likelihood of the forming of this coalition. Meister *et al.* [66] and Hipel and Meister [40] proposed an option-based metric related to the preference tree [27], which depends on the "similarity of preference" for individual options and the "average importance" of individual options for coalition members. An overall coalition preference tree, which contains the coalition's ordinal preferences over the states, is then built up from the individual preference trees and this option-based metric.

The first of these perspectives is based on the assumption that DMs with more similar preferences are more likely to form a coalition. The second is a necessary step to redefining a conflict model with the coalition as a new player - the coalition's preferences must be determined in some way. However, Brams [12] proceeds from the assumption that a DM searches not for a coalition partner who has similar preferences, but who provides that DM with a greater chance of doing better. The study of the Elmira Groundwater Contamination Dispute, the model of which is used in this thesis, reveals clearly that preference similarity is not necessary for the members of a coalition to achieve a mutually preferable equilibrium. Moreover, treating a coalition as a single DM actually implies that the coalition has to be durable throughout the course of the conflict and the control of the members' options is absolutely centralized. The author believe that the world of strategic conflicts is far more versatile for this assumption to hold. Thus the new perspective on coalition analysis implemented here is based on the impact of the coalition on the outcome, instead of on similarity of preferences.

### 5.2.2 *A New Perspective*

The objective here is to provide a new perspective on the study of coalition formation using the graph model framework (or some other related conflict models), and to illustrate this new prospective using the Elmira case. Some definitions are presented, which lead to measures of tendency of certain coalitions to form, and the resulting tendency of certain outcomes to be unstable. By studying the possible impact of coalitions on stability results in a conflict involving more than two DMs, one can get a clearer picture of the decision situation.

A coalition is any subset $\mathbf{H} \subseteq \mathbf{N}$ such that $|\mathbf{H}| \geq 2$. (Thus a coalition is any set of two or more DMs). $\mathbf{S_H}(u)$ (Definition 2.1) was defined to model sanctions in the stability analysis of a multiple-DM model [20]. It will continue to be used in this coalition analysis context without alteration. The only difference is that a member of $\mathbf{S_H}(u)$ is now called a *coalition move* by **H** from state $u$.

**Definition 5.1** *A state* $u \in \mathbf{U}$ *is* **stable for coalition H** *iff*

$$\forall u' \in \mathbf{S_H}(u), \exists i \in \mathbf{H} \text{ so that } P_i(u') \leq P_i(u).$$

**Definition 5.2** *A state* $u' \in \mathbf{U}$ *is called a* **coalition improvement** *from state* $u$ *by* $\mathbf{H} \subseteq \mathbf{N}$ *if it is a member of*

$$\mathbf{S_H^{(+)}}(u) = \{u'' \in \mathbf{S_H}(u) | \forall i \in \mathbf{H},\ P_i(u'') > P_i(u)\}$$

Note that the set of coalition improvements by **H**, $\mathbf{S_H^{(+)}}(u)$ is different from $\mathbf{S_H^{+}}(u)$ (Section 2.2.2), which is the set of all group UIs by **H**. The latter requires that each member can be reached via a sequence of inividual UIs, but the member itself does not necessary make every coalition member better off. In contrast, the first has no requirement on the individual moves that form each of its members, but does required that each member must be an improvement for each coalition member. Obviously,

**Proposition 5.1** *A state $u' \in \mathbf{U}$ is stable for coalition* $\mathbf{H}$ *iff*

$$\mathbf{S}_{\mathbf{H}}^{(+)}(u) = \emptyset.$$

Proposition 5.1 means that a state is stable for a coalition if and only if that coalition does not have available improvement from this state.

**Definition 5.3** *A state $u \in \mathbf{U}$ is* **coalitionally stable** *iff it is stable for all coalitions* $\mathbf{H} \subseteq \mathbf{N}$.

Joining Definition 5.3 and Proposition 5.1 gives

**Proposition 5.2** *A state $u \in \mathbf{U}$ is* **coalitionally stable** *iff* $\forall \mathbf{H} \subseteq \mathbf{N}, \mathbf{S}_{\mathbf{H}}^{(+)} = \emptyset$..

Of particular interest are states that are stable for all DMs (according to some individual stability definitions) but are unstable for some coalition(s). These states are likely to persist as long as the DMs act independently, but are unlikely to endure if a coalition forms for which they are not coalitionally stable. The contribution of coalition analysis is to point out these states that are vulnerable to coalitions. This is particularly the case when, as in the Elmira conflict, the coalition improvement is itself highly stable - both individually and coalitionally.

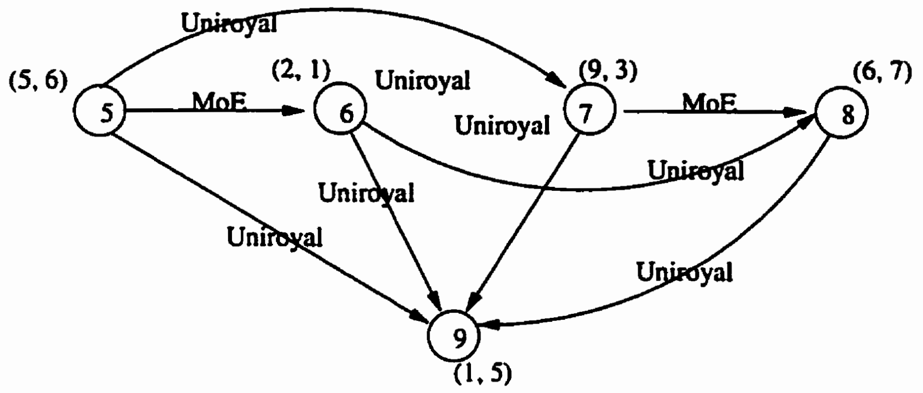


Figure 5.4: Relevant Part of Integrated Graph for Elmira Model

Figure 5.4 shows the relevant part of the integrated graph of the Elmira conflict, where the ordinal payoffs of MoE and Uniroyal are labeled beside each node. (The first and second numbers in brackets denote MoE's and Uniroyal's ordinal payoffs, respectively. Recall that a greater payoff means more preferred.) Use index values 1, 2, 3 to represent the three DMs, MoE, Uniroyal and Local Government (LG), respectively. Let **H** ={1, 2}. We have $u_8 \in \mathbf{S_H}(u_5), P_1(u_8) > P_1(u_5), P_2(u_8) > P_2(u_5)$. In other words, MoE and Uniroyal together control the transition from state 5 to 8, and that both are better off at state 8 than at state 5. Therefore, state 5 is not stable for coalition **H**; it is coalitionally unstable, even though it is strongly stable under individual stability definitions (Table 2.6. Furthermore, the consequence of the coalition improvement, state 8, is highly stable individually (Table 2.6) and coalitionally stable also. Figure 5.4 shows how the coalition move from 5 to 8 could take place.

This analysis of the Elmira model is supported by actual events. Historically, the status quo in mid-1991 was state 1. Local Government shifted quickly to support the original Control Order, resulting in state 5 for a protracted interval of time. Then MoE and Uniroyal dramatically announced agreement on a modified version of the original control order, thus moving to the equilibrium at state 8. This agreement caught Local Government by surprise; its protests were to no avail, and it was forced to reach a separate and quite unfavorable arrangement with Uniroyal. The coalition made both MoE and Uniroyal better off, and the outcome was no less stable than before. Local Government was harmed by the deal, but could do nothing to prevent it. Here, an interesting observation is that the similarity of preferences between MoE and Uniroyal is actually lower than that of any other pair, yet {MoE, Uniroyal} is the coalition that was formed. Moreover, both state 5 and state 8 are coalitionally stable for any coalition other than {MoE, Uniroyal}.

A coalition improvement by a subset of DMs indicates a threat to the stability of a state. Even though the coalition might find it difficult to co-operate to implement the improvement, the important fact, identified here, is that the coalition members have an incentive to co-operate. This information can form the basis for valuable advice to DMs in a conflict. This work is quite different from others who require the coalition's preference ranking in order to treat it as a single "DM". The view taken here is that an individual cooperates when it is in that individual's interest to cooperate. Other approaches begin by specifying the coalition, and then calculating how it affects behavior; here, a state is defined to be coalitionally stable only when there is no coalition that is capable of upsetting it, and actually prefers to do so.

Coalition stability for a coalition is related to Pareto Inferiority.

**Definition 5.4** *A state* $u \in \mathbf{U}$ *is said to be* **Pareto-inferior** *for a coalition* $\mathbf{H}$ *within* $\mathbf{S} \subseteq \mathbf{U}$ *if* $\exists u' \in \mathbf{S}$ *such that* $\forall i \in \mathbf{H}, P_i(u') \geq P_i(u)$ *and* $\exists j \in \mathbf{H}, P_j(u') > P_j(u)$.

It can be proven:

**Proposition 5.3** *If* $u \in \mathbf{U}$ *is Pareto-inferior for* $\mathbf{H}$ *within* $\mathbf{S_H}(u)$*, then* $u$ *is* **coalitionally unstable.**

This the significance of this proposition is, under the Graph Model for Conflict Resolution paradigm, group Pareto-inferior outcomes can probably be removed via proper communication and cooperation. This is useful when using the Graph Model for Conflict Resolution in negotiation and bargaining.

### 5.2.3 *Implementation in GMCR II*

Currently the intended role of coalition analysis in GMCR II is as a follow-up analyses that re-examines the states that have been identified as equilibria by the stability analysis, according to certain individual stability specifications. The main purpose is to challenge the coalitional stability of these equilibrium states, and possibly reduce the number of possible outcomes by identifying those that are coalitionally unstable. As a first implementation step, GMCR II will point out the equilibria that are vulnerable to a coalition improvement that leads to other equilibria. This kind of coalition improvments is also referred to as "equilibrium jumping" ([61]).

From the series of definitions and discussions in Section 5.2.2, it can be seen that the concept of $\mathbf{S_H}(u)$ is instrumental to this new coalition analysis approach. Therefore, the efficient implementation of $\mathbf{S_H}(u)$ in Section 5.1.2 greatly facilitates the implementation of the calculation of coalition stability.

Let the set of equilibria from stability analysis of GMCR II be $\mathcal{E}$, and denote the set of DMs who prefer equilibrium $e'$ over $e$ by $\mathrm{N}(e, e')$. The algorithm for

```
for $e \in \mathcal{E}$
...for $e' \in \mathcal{E}$
......$\mathrm{N}(e,e') = \emptyset$;
......for $i \in \mathrm{N}$
.........if $(P_i(e') > P_i(e))$
............$i \in \mathrm{N}(e,e')$;
.........endif
......endfor
......if $(e' \in G_{\mathrm{N}(e,e')}(e)$
.........mark $e$ as coalitionally unstable;
.........break;
......endif
...endfor
endfor.
```

Figure 5.5: Algorithm for Identifying Coaltionally "Vulnerable" Equilibrium

identifying the "vulnerable equilibria" according to the above discussion can be expressed as pseudo code in Figure 5.5.

Of course, the above algorithm can be easily extended, if desired, by replacing $\mathcal{E}$ with full state set U. This identification process provides the user a better picture of predicted resolutions of a conflict, and hence helps achieve better decisions.

## 5.2.4 *Possible Extensions*

There are two directions in which this work on coalition analysis could be extended. One is to consider countermoves to the coalition improvement; the other is to take into account the enforcability of the agreement that the coalition improvement is based upon.

What has been defined earlier in this chapter is a sort of "$L_1$" ("Nash") stability for coalitions: no counter-moves are considered. When they are included, other stability types can also be defined, although they may be applicable only under complex assumptions about behavior. For instance, Bernheim, Peleg and Whinston's [10, 11] "Coalition-proof Nash Equilibrium" against Aumann's "Strong Nash Equilibrium" [5] is a good reference to start. The following is the author's first try:

**Definition 5.5** *A state $u \in \mathbf{U}$ is* **coalitionally safe** *iff:* $\forall \mathbf{H} \subseteq \mathbf{N}$, *if* $\mathbf{S}_{\mathbf{H}}^{(+)}(u) \neq \emptyset$, *then* $\forall u' \in \mathbf{S}_{\mathbf{H}}^{(+)}(u), \exists \mathbf{H}' \subseteq \mathbf{H}$, *such that* $\mathbf{S}_{\mathbf{H}'}^{(+)}(u') \neq \emptyset$.

According to this definition, the original stability (if applicable) of a state can only be challenged by a coalition improvement that itself is internally coalitionally stable.

The "cooperative" viewpoint in game theory assumes DMs can make an agreement that is binding and enforcable. But, as pointed out by Brams [13, page 25], "their decision to cooperate in the first place should emerge as the result of 'non-cooperative' individual calculations, which would inform them, for example, that such an agreement is stable instead of their just assuming this to be the case." Building cooperative game theory on non-cooperative foundations is what is known as the "Nash program" in game theory. The DMs should plan their moves without presuming that other DMs are bound by agreements to cooperate; rather, other DMs will cooperate only if they find it in their interests to do so. A coalition improvement can be viewed as an agreement among coalition members. So a coalition improvement has to seem to be ***stable*** or ***enforcable***. An agreement is ***enforcable*** if it is not in any coalition member's interest to diverge from the agreement. For example, in Figure 5.4, agreement "$5 \rightarrow 6 \rightarrow 8$" would be more "enforceable" than "$5 \rightarrow 7 \rightarrow 8$". As a minimum condition, a coalition improvement is stable or enforceable because it makes the actual movers better off. But there could be different ways to define the enforceability of an agreement, depending on each coalition member's solution concept, as well as other assumptions. Additional study is required to define enforceability credibly in different contexts. In other words, much more content could be added to the definition of coalitional stability.

In conclusion, the new concept of coalition stability will contribute to the theory of multi-party negotiations and help to increase the performance of the related DSSs. It will make an important contribution to the effectiveness of GMCR II in aiding DMs in negotiation and other form of strategic conflict.

## 5.3 Status Quo Analysis

Status quo refers to the current situation of the conflict. "Status quo analysis" was occasionally used to refer to the identification of a path on which the conflict would move from the status quo to the predicted outcome. This kind of identification is often done in an *ad hoc* and intuitive manner; no systematic process has been suggested. The role of status quo in the analysis and interpretation of a conflict has been so far largely underestimated, or simply ignored. This section demonstrates how status quo and its associated analysis can be used in the analysis and interpretation of a conflict model.

### 5.3.1 *Status Quo Analysis and Irreversible Moves*

The importance of status quo analysis is highlighted due to the existence of irreversible moves in the Graph Model for Conflict Resolution. In general, status quo analysis concerns how far and where a conflict can reach from the status quo. In the the assumption (4.8) of original option form, a conflict can move from any status quo state to any other state via a joint effort of all DMs:

$$\forall u \in \mathbf{U}, \mathbf{S_N}(u) = \mathbf{U} \setminus \{u\}.$$

However, the introduction of irreversible moves greatly changes the landscape. For ease of reference, the integrated graph of the Elmira conflict and the table listing each DM's preference ranking are reproduced in Figure 5.6 and Table 5.2, respectively. From Figure 5.6, it can be seen that:

$$\mathbf{S_N}(u_1) = \mathbf{U} \setminus \{u_1\} \tag{5.1}$$

$$\mathbf{S_N}(u_2) = \{u_4, u_6, u_8, u_9\} \tag{5.2}$$

$$\mathbf{S_N}(u_3) = \{u_4, u_7, u_8, u_9\} \tag{5.3}$$

$$\mathbf{S_N}(u_4) = \{u_8, u_9\} \tag{5.4}$$

$$\mathbf{S_N}(u_5) = \mathbf{U} \setminus \{u_5\} \tag{5.5}$$

$$\mathbf{S_N}(u_6) = \{u_4, u_2, u_8, u_9\} \tag{5.6}$$

$$\mathbf{S_N}(u_7) = \{u_4, u_3, u_8, u_9\} \tag{5.7}$$

$$\mathbf{S_N}(u_8) = \{u_4, u_9\} \tag{5.8}$$

$$\mathbf{S_N}(u_9) = \emptyset \tag{5.9}$$

From this above list, one can surely better appreciate what a big difference the concept of irreversible moves has brought to the nature of conflict modeling.

### 5.3.2 *Status Quo and Stability Analysis*

Equilibria as an output of the stability analysis are intended to be the predicted possible resolution of a conflict model. However, the status quo of the conflict model is currently not considered part of the input for stability analysis. The consequence is that some of the "predicted resolutions" would be simply impossible to reach from

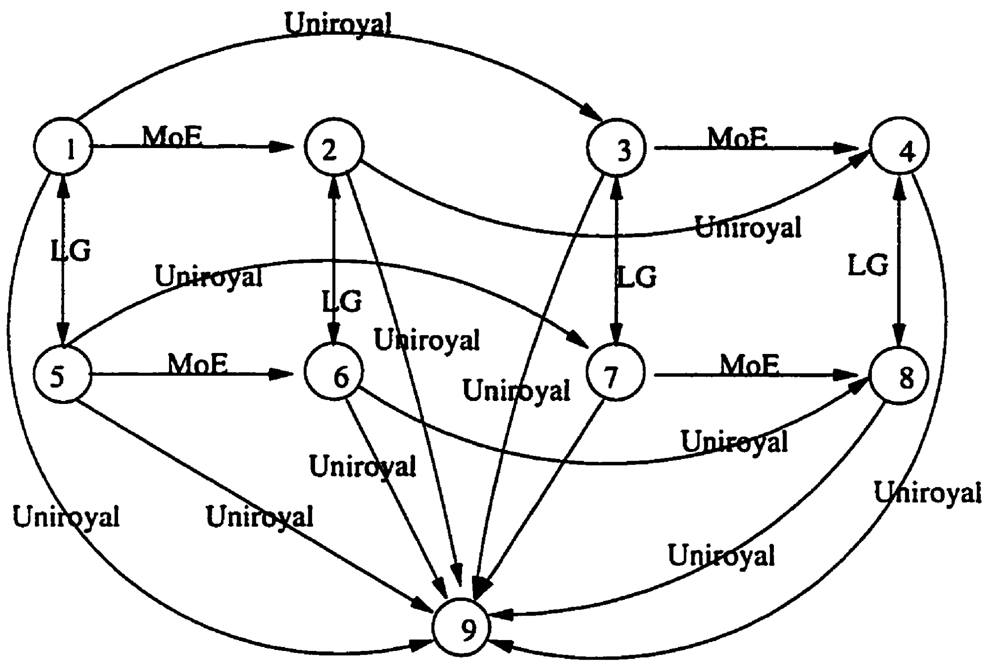


Figure 5.6: Integrated Graph of the Elmira Model

Table 5.2: Preference Ranking for the Elmira Conflict Model

| | *most preferred* ⟶ *least preferred* | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| MoE | 7 | 3 | 4 | 8 | 5 | 1 | 2 | 6 | 9 |
| Uniroyal | 1 | 4 | 8 | 5 | 9 | 3 | 7 | 2 | 6 |
| Local Government | 7 | 3 | 5 | 1 | 8 | 6 | 4 | 2 | 9 |

the starting point of the model. Therefore, the status quo should become part of the input, and the necessary scope of stability analysis can usually be significantly reduced. An extreme example is, in the Elmira model, if the status quo were state 9, then no stability analysis would be necessary – there is simply no way out!

Performing stability analysis only in a restricted scope based on status quo, rather than on the whole set, can mean a significant saving of computing resource. More important reason is that unreachable, and hence false, prediction should be ruled out. In the current version of GMCR II, exhaustive stability analysis is still performed. but a well designed status quo analysis interface is provided to facilitate the user's need. A conflict analysis without status quo analysis is an incomplete analysis.

### 5.3.3 *Evolution Path of Conflict Model*

A popular use of status quo analysis is to find the possible evolution path from the the starting point of the conflict to an equilibrium. Status quo analysis as a follow-up analysis is usually carried out after the stability results are available. It should be pointed out that an evolution path is not just a viable path in the integrated graph linking the status quo to the focal equilibrium; rather, the incentive of each DM to join the path based on his/her behavior style is important. For example, a DM should not be expected to move from a state that is stable for him or her. Conseqently, usually no evolution path can pass through an equilibrium state. However, one exception is the "equilibrium jump" as discussed in Section 5.2.3. GMCR II's coalition analysis takes into account "equilibrium jumping".

### 5.3.4 *Beyond Evolution Path*

Status quo analysis can also be used to provide strategic advice directly. For example, in the Elmira model (Figure 5.6), assume that the status quo is state 2 and the DMs' behavior patterns are all non-myopic. One can find that if Uniroyal move first to state 4, then the only possible states the conflict can move to are members of $\mathbf{G_N}(u_4) = \{u_8, u_9\}$. Notice that Uniroyal itself control the move to state 9, so the possible destinations now reduce to $\mathbf{G}_{\mathbf{N}\setminus\{2\}}(u_4) = \{u_8\}$. Since state 8 is an equilibrium (Table 2.6) that is favorable to Uniroyal, a possible strategy for Uniroyal would be, move quickly to state 4 so as to achieve a favorable equilibrium.

Status quo analysis can also be used to verify the stability results of a state by examining the possible moves and countermoves in sequence. Under careful study, much more helpful forms of status quo analysis can be discovered. The graph model is especially suitable for status quo analysis, due to its state-based nature. The implementation of various forms of status quo analysis basically concerns the calculation of $\mathbf{G}_i(u), \mathbf{G}_i^+(u), \mathbf{G_H}(u), \mathbf{G_H^+}(u)$, and $\mathbf{G_H^{(+)}}(u)$. The efficient implementation of these components in GMCR II (as discussed in Sections 4.4.3 and 5.1.2) is essential to this feature.

## 5.4 Other Follow-Up Analyses

Follow-up analyses refer to additional "what if" analyses that may be carried out after the initial stability analysis. Besides coalition analysis and status quo analysis, sensitivity analysis, dynamic analysis and hypergame analysis are other follow-up analyses that can provide useful interpretations and refinements of the stability results. This section provides a brief introduction to these techniques and their implement in the current version of GMCR II.

### 5.4.1 *Sensitivity Analysis*

The general purpose of ***sensitivity analysis*** is to identify the most sensitive parameters of a model in terms of impact on the predictions of the model. Information about most real-world conflicts is generally incomplete, and estimates of the parameters are likely to be subjective - depending on the perceptions, opinions, and experience of the analyst. Therefore, sensitivity analysis is importance in conflict resolution.

Sensitivity analysis for the graph model is usually carried out by asking "*what if?*" questions, or in other words, by varying the model input in the following categories:

- the identity of the DMs, and the options (or state transitions) available to them;
- the preferences of the DMs;
- the behavioral styles of the DMs.

Because conflict analysis is based on ordinal models, and because there is always a certain amount of user interaction between the input and output stage, sensitivity analysis is carried out in the form of sensitivity testing, instead of the parametric analysis used in quantitative methods. Nonetheless, it is helpful to develop a systematic procedure so that relevant aspects of the range of variation of input information can be suggested to the analyst.

In existing systems, sensitivity analyses can be carried out only in a separated session by rerunning the system. GMCR II is built as a multiple document interface (MDI) Windows application [72]. MDI applications allow multiple document frame windows to be open in the same instance of an application. An MDI application has a window within which multiple MDI child windows, which are themselves frame windows, can be opened. Each child window contains a separate document, permitting GMCR II to carry out sensitivity analyses in the same session, but on different documents. In this way, GMCR II offers more than existing systems.

Because in GMCR II all the input information in modeling stage is saved in a .gm (graph model) file, the reformulation effort for a slightly different model is nominal. However, assessment of the results of the variations needs an alignment facility, especially when the variation has changed the list of feasible states in the model. It is left to future development to design a genuine integration of sensitivity analysis features into this system.

Normally, sensitivity analysis is intended to assess the robustness of the stability results. However, sensitivity analysis can also be used to provide strategic insights to the analyst. For example, an interesting aspect of the Elmira conflict, from a policy point of view, is the effort to ensure that Local Government had an essential role in developing the resolution. Sensitivity analyses can be carried out to examine the importance of this role in the outcome of the conflict.

In the existing Elmira model, add one more "at least one" specification as highlighted in Figure 5.7 (compare with Figure 4.6). The resulting new model has five states (Figure 5.8, with the status on option 4 (*Insist*) fixed to "Y"). All the other aspects of the model remain the same. Call this model the Elmira Y sub-model. A Elmira N sub-model can be similarly obtained, in which all the states have "N" on option 5. The correspondence of states among the original model, the

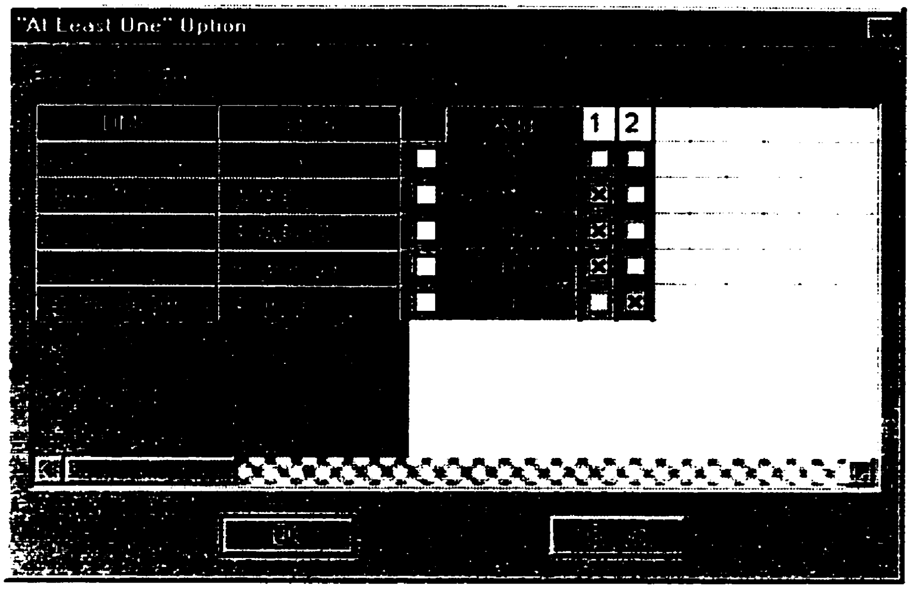
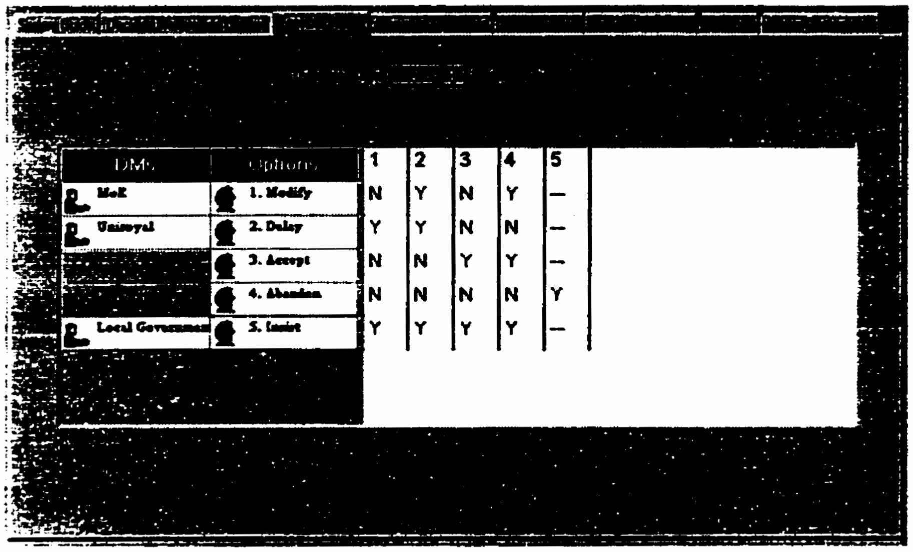


Figure 5.7: Model Variation for Sensitivity Analysis

Y sub-model and the N sub-model is given in Table 5.3.

Stability analyses are performed on these two sub-models and the results are listed in Table 5.4. Comparing Tables 5.4, 5.3 and 2.6, one can find that Local Government cannot possibly influence the result of the conflict. Therefore, decision advice for the Local Government based on this sensitivity analyses is that it should either keep out to avoid the cost of involvement, or that it should consider other more effective strategies.



| DMs | Options | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| MoE | 1. Modify | N | Y | N | Y | — |
| Uniroyal | 2. Delay | Y | Y | N | N | — |
|  | 3. Accept | N | N | Y | Y | — |
|  | 4. Abandon | N | N | N | N | Y |
| Local Government | 5. Insist | Y | Y | Y | Y | — |

Figure 5.8: Feasible States in Elmira Y Sub-Model

Table 5.3: Feasible States of the Elmira Model, Y Sub-Model and N Sub-Model

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1. **MoE** | | | | | | | | | |
| (1) *Modify* | N | Y | N | Y | N | Y | N | Y | – |
| 2. **Uniroyal** | | | | | | | | | |
| (2) *Delay* | Y | Y | N | N | Y | Y | N | N | – |
| (3) *Accept* | N | N | Y | Y | N | N | Y | Y | – |
| (4) *Abandon* | N | N | N | N | N | N | N | N | Y |
| 3. **Local Government** | | | | | | | | | |
| (5) *Insist* | N | N | N | N | Y | Y | Y | Y | – |
| State Number | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| Y Sub-Model | | | | | 1 | 2 | 3 | 4 | 5 |
| N Sub-Model | 1 | 2 | 3 | 4 | | | | | 5 |

Table 5.4: Equilibria of the Sub-Models

| Y Sub-Model | N Sub-Model | Equilibrium under |
|---|---|---|
| 1 | 1 | R, GMR, SMR, SEQ, $L_1 - L_{10}$, NM |
| 4 | 4 | R, GMR, SMR, SEQ, $L_1 - L_{10}$, NM |
| 5 | 5 | R, GMR, SMR, SEQ, $L_1 - L_{10}$,NM |

(Limited move stability up to level 10 only.)

### 5.4.2 *Hypergame Analysis*

Hypergames are modeling structures that incorporate information about misperceptions. A misperception can be related to the preferences of DMs, the available options of DMs, or even the relevance of a particular DM's role in a conflict model. An effective model of hypergames for use with conflict resolution has been developed by Wang et al. [88].

Hypergame analysis is a useful tool in strategic conflict. It can help a DM in assessing the possibly favorable effect of a misperception caused by his or her intentional bluffing, or in evaluating the negative impact of a misperception occurring accidentally through a lack of communication.

The most common misperception is misperception of preferences. In this case, different models can be built using GMCR II's MDI design. Similar to sensitivity analysis, only nominal effort is usually needed to establish a new model from the original. The stability results of different models can then be compared. In this case, the comparison is easier because these models share the same list of feasible states. In the case of misperception related to available options of DMs, or even the role of some DMs, the comparison can be more complicated due to differences in the sets of feasible states. Currently the user must line up comparable states. A future implementation of GMCR II should consider a genuine incorporation of hypergame analysis in which different misperception models can be analyzed in a same document and the alignment of comparable states is automated.

A new type of misperception that can arise in the context of GMCR II is the misperception of a DM's behavior pattern. In this case, equilibrium results from different stability type specifications based on this misperception can be compared. Since this type of hypergame analysis does not interfere the modeling stage, it can be readily carried out in a same session upon provision of comparison facilities. The appropriate use of hypergame analysis in GMCR II has yet to be studied carefully.

### 5.4.3 *Dynamic Analysis*

A strategic conflict can evolve over time. The goals and preferences of a DM can usually be established at a particular time, but changes could occur due to either revealed information or the effects of exogenous events. A form of dynamic analysis suggested in [65] to account the effects of exogenous events can be rea dily used in a DSS such as GMCR II. An imaginary DM named "Nature" or "Environment", with no preferences but rather with control over exogenous events can be included. Other DMs' preferences, expressed in option prioritizing or option weighting, could become dependent on the event option. For example, an "Environment" DM, who controls an event option "found more contamination evidence" could be introduced into the Elmira model. Then the possible changes of each DM's preferences due to that event can be integrated into the option-based preference specifications, such that surprise could be avoided. However, dynamic analysis in this format is applicable only in option form.

# Chapter 6

# Output Presentation and Interpretation

## 6.1 Output Information from Stability Analysis

The analysis engine of GMCR II generates a vast amount of output data – information about the stability of each state for each DM under a variety of solution concepts. Figure 6.1 shows the organization of stability results in the output data subsystem. The three dimensional structure of Figure 6.1 can be interpreted in a variety of ways, including:

1. For each DM, the ***DM's plane*** (parallel to the STATE/STABILITY TYPE plane) indicates the stability or instability of each state under each possible stability type for that specific DM.

2. For each stability type, the ***stability-type plane*** (parallel to the DECISION MAKER/STATE plane) provides a complete analysis of the model according

to that stability type.

3. For each state, the *state plane* (parallel to the DECISION MAKER/STABILITY TYPE plane) identifies all DMs for whom the particular state is stable, under each possible stability type.

4. The STABILITY TYPE/STATE plane itself, referred to as the *equilibrium plane*, contains the projection of stability results for each DM, indicating all equilibria for each stability type.

As can be seen, complete stability information is produced by the analysis engine. In practice, a user often wishes to view the stability results in different manners. He or she may also wish to have some additional analyses based on the initial stability results, which would be very helpful for inspiring insights into decision problems. As a decision support system, GMCR II is intended to provide DMs and analysts with decision advice, structural insights, and answers to "what if?" questions. This goal cannot be fully achieved without an interactive presentation of the analysis results. The following subsections discuss and illustrate how the stability results are presented and utilized in GMCR II in order to better achieve the goal of providing useful strategic decision advice. The Elmira model continues to be used as the illustration case.

## 6.2 Displaying Equilibria and Stable States

The Graph Model methodology takes states as its basic building units while ignoring further details about their option representations. By doing so, the methodology offers more flexibility, because the states could be represented in formats other

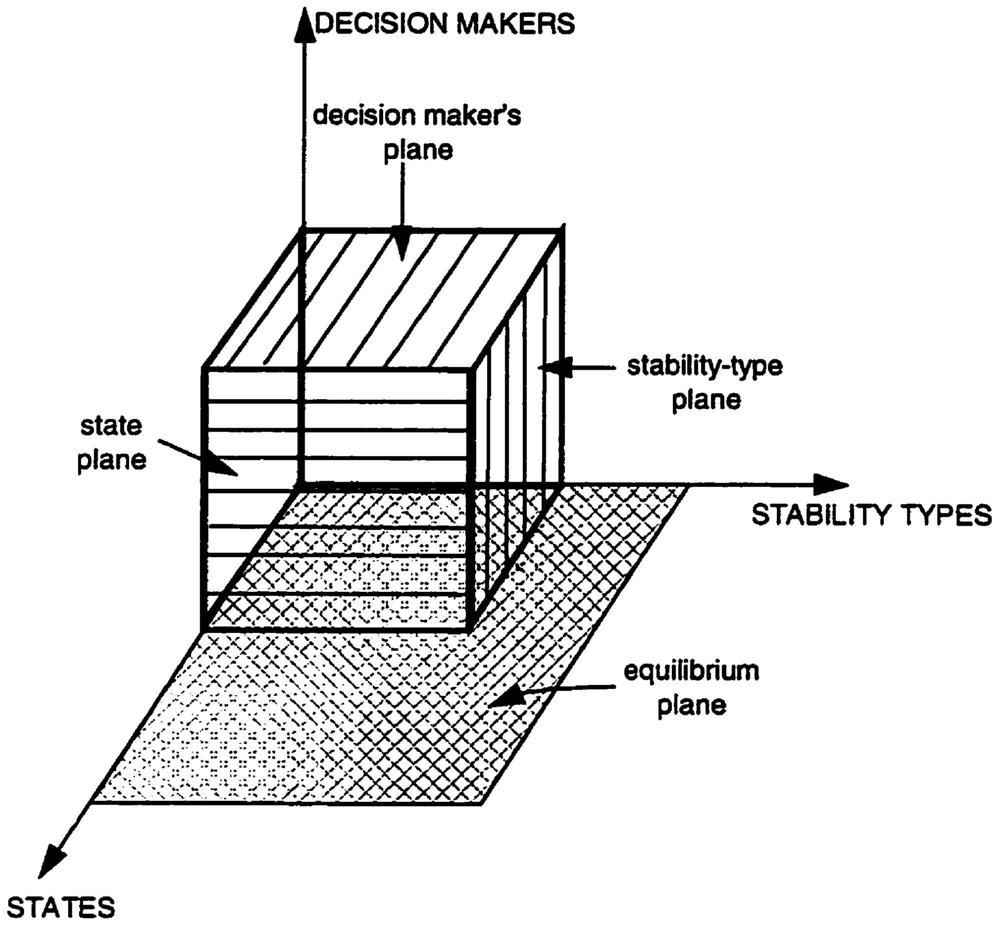
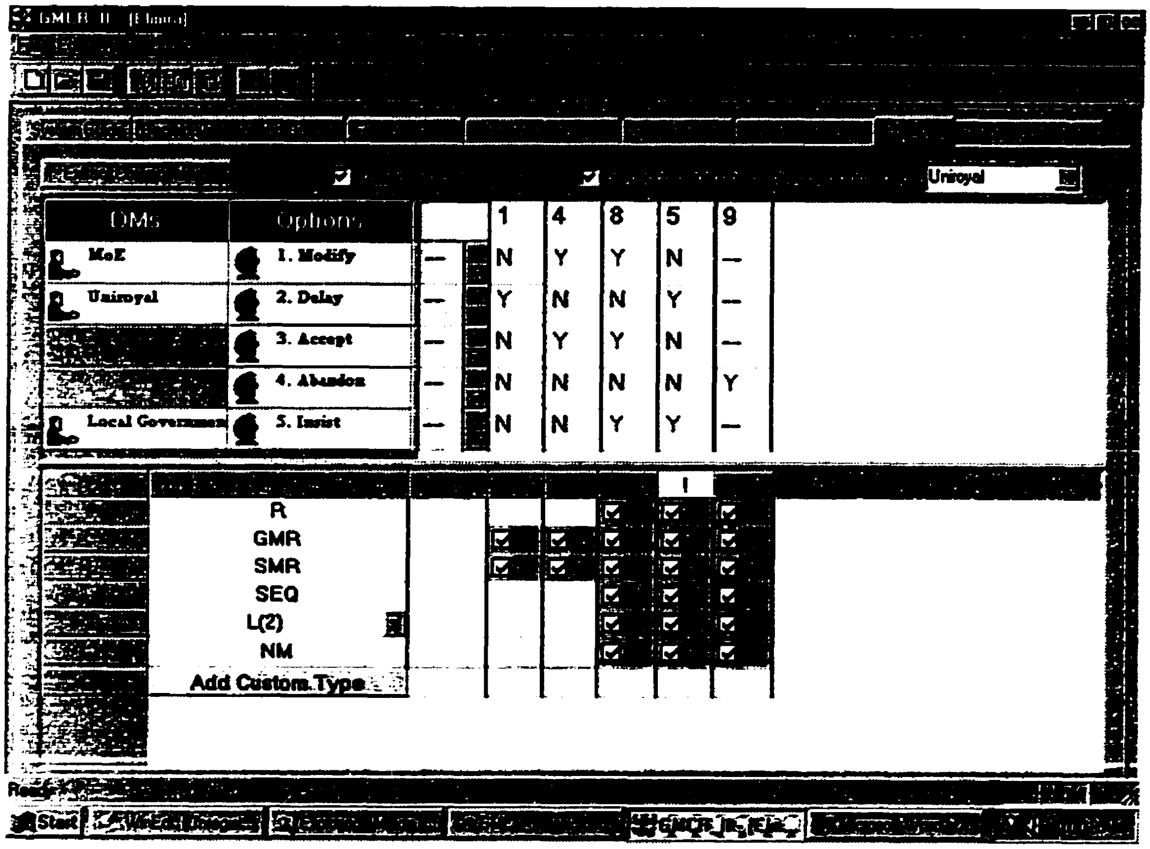


Figure 6.1: Stability Results Structure [20]

than the option form, without any impact on the stability analysis. As a result, the output of the last generation of this DSS, GMCR I [20], while possessing the same information as depicted in Figure 6.1, identifies a state only by the state number, which is an arbitrarily assigned integer and provides no information at all about the nature of the state. GMCR II, however, takes option form as a major representation format of states from the modeling stage. In this case, when displaying stability results, it is very important to associate a state with its option form representation rather than just the state number, now that the information is available. Usually, a user wants to see as much information as possible on the same screen, and preferences of DMs are always essential for an understanding of decision choices. Taking the above into account, the interface for displaying equilibrium states is designed as in Figure 6.2. Note that the interactive presentation of analysis results allows the user to examine different aspect of the output, but does not alter the established conflict model itself. Therefore, the display of output data takes the form of property pages, rather than dialog boxes, except for those dialogs used for specifying user options. The property page for displaying individually stable states (under the tab "Individual Stability" in Figure 6.2) is very similar to "Equilibrium" page, except that the "Coalition Stability" check box and "Add Custom Type" button are non-applicable, while a drop down list appears for choosing the DM for whom the states are stable.

A check box on the up-right of this property page allows a user to specify whether he or she wants the equilibria to be shown in the order of a focal DM's preferences. In Figure 6.2, this option is chosen, and a pull-down combo box allows the user to specify the focal DM, which is Uniroyal in this case. The equilibrium states, as shown in option form in the upper area of this property page, are hence ranked from most to least preferred based on the Uniroyal's preferences (refer to



GMCR II [Elmira]

Uniroyal

| DMs | Options | | 1 | 4 | 8 | 5 | 9 |
|---|---|---|---|---|---|---|---|
| MoE | 1. Modify | — | N | Y | Y | N | — |
| Uniroyal | 2. Delay | — | Y | N | N | Y | — |
| | 3. Accept | — | N | Y | Y | N | — |
| | 4. Abandon | — | N | N | N | N | Y |
| Local Government | 5. Insist | — | N | N | Y | Y | — |

| | 1 | 4 | 8 | 5 | 9 |
|---|---|---|---|---|---|
| R | | | ✓ | ✓ | ✓ |
| GMR | ✓ | ✓ | ✓ | ✓ | ✓ |
| SMR | ✓ | ✓ | ✓ | ✓ | ✓ |
| SEQ | | | ✓ | ✓ | ✓ |
| L(2) | | | ✓ | ✓ | ✓ |
| NM | | | ✓ | ✓ | ✓ |
| Add Custom Type | | | | | |

Ready

Start

Figure 6.2: Display of Equilibrium Results

Figure 4.22). If this box is unchecked, then the equilibria would be displayed simply according to the natural order of state numbers. In the lower part of this page, a check in a cell indicates that the state represented by the column is an equilibrium under the solution concept represented by the row. As a default, the equilibrium type for each solution concept that GMCR II analyzed (Table 2.1) is included. Note that another pull-down combo box is used for the specification of the depth of limited move stability so as to better organize the dialog box and save vertical space. The levels start at 2, because L1 overlaps with Nash stability (R), and can go up to 10 steps.

As shown in In Figure 6.2, in the Elmira model, stronger and longer term equilibria occur at states 5, 8 and 9, among which Uniroyal prefers state 8. Listing equilibria based on each DM's preferences can give the analyst a clear overall picture of each DM's intention, which is essential for he or her to better assess the decision situation.

## 6.3 Displaying User Customized Equilibria

The initial run of the analysis engine determines the stability of each state for each DM for all the solution concepts listed in Table 2.1. However, as pointed out in Section 5.1.3, the applicability of a solution concept to a particular DM actually depends on the pattern of this DM's behavior. As characterized in Table 2.1, a solution concept is usually associated with the DM's foresight, knowledge of preferences, and strategic risk attitude. It is very unlikely that all the solution concepts listed are applicable to any particular DM. Therefore, the specification of applicable solution concepts for each DM is necessary.

Since a solution concept may not be applicable for all the DMs involved, the concept of equilibrium types must be generalized. Besides the traditional ones, many more equilibrium types can be defined by considering a state that is stable for all DMs but under different solution concepts for different DMs. Since the GMCR II analysis engine calculates all individual stabilities for each state, no additional work needs to be done on the engine part. The problem can be addressed by providing a dialog box to allow the user to customize equilibrium types by identifying desirable solution concepts for each DM. For example, if a state is Nash stable for MoE, SEQ stable for Uniroyal, and NM stable for Local Government, it also constitutes an equilibrium, even though it is difficult to name its type. The introduction of custom-designed equilibria significantly generalizes the definition of equilibria. For example, with up to a 10 limited-move level, a 3-DM conflict model like the Elmira model has $14 \times 14 \times 14 = 2744$ distinct possible equilibrium types under this new concept, compared with only 14 choices originally.

In Figure 6.2, the "Add Custom Type" button appearing at the bottom of the existing equilibrium type list allows the user to specify one more custom designed type each time. Upon clicking on this button, a dialog box (Figure 6.3) will pop up to allow the user to specify a customized equilibrium type by choosing an appropriate (but not necessarily identical) stability type for each DM. For each DM, a corresponding drop list contains all available stability types. This specification method is easy and straightforward.

Figure 6.4 illustrates how the equilibria are displayed when three customized types are added. "(RQN)", for example, indicates an equilibrium type in which Nash (R) stability applies to the first DM, SEQ (Q) stability the second, and Non-myopic (N) stability the third. If a certain level, say level 2 of limited-move stability is requested, L2 will be used to indicate the choice in the brackets. The "Add Custom Type" now moves to the bottom of this updated list again, allowing

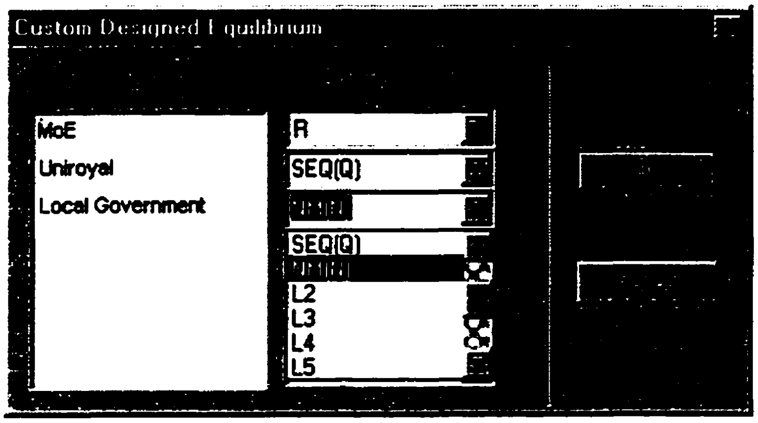


Figure 6.3: Specifying a Customized Equilibrium Type

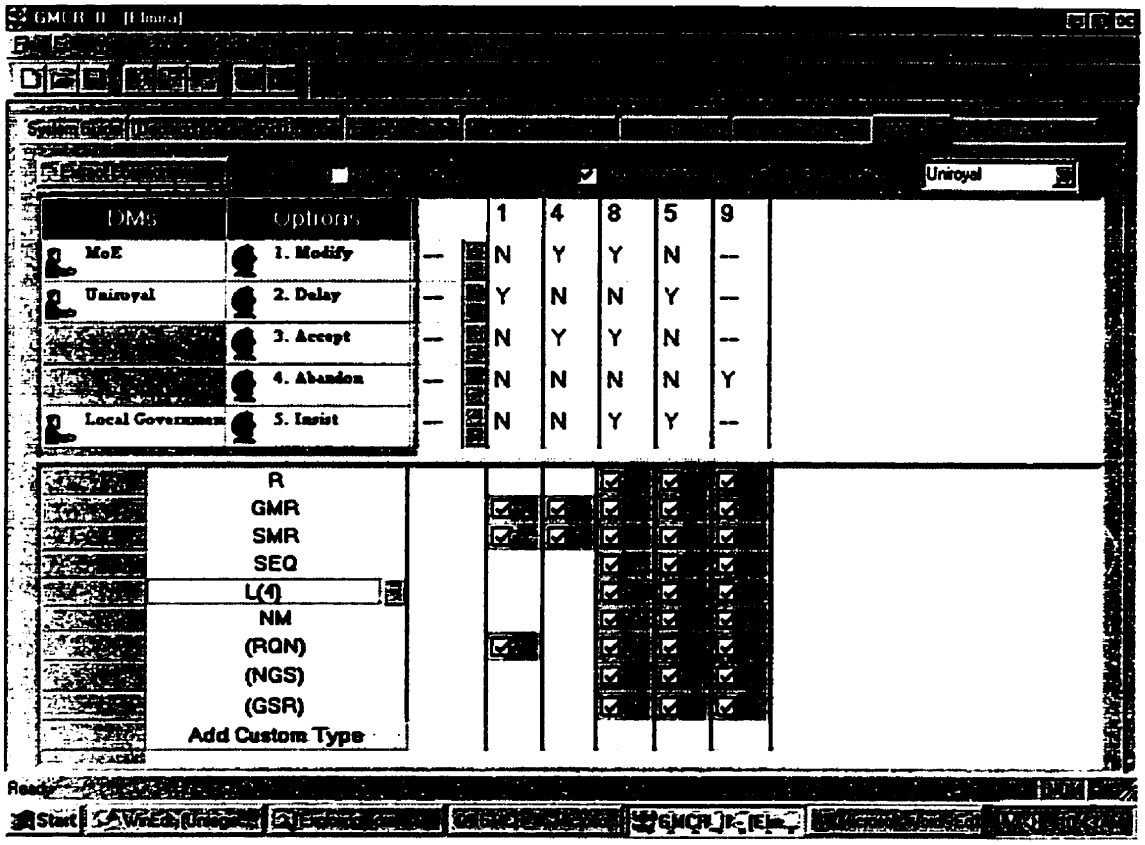
more types to be specified. The user can also remove an undesirable type from the display.

For a non-sophisticated user, an acronym like GMR or SEQ has little relevance. Studies are ongoing to design an on-screen questionnaire based on the attributes in Table 2.1 to help the user assign the most appropriate solution concept(s).

## 6.4 Extracting Commonalities

Some large real-world conflict models may have thousands (or even more) of feasible states, and a great number of equilibria may be identified. Under these circumstances, it is useful to discover common features of the equilibria so that key information can be extracted. In option form, the common features can be either of first or second order:

**First order:** A first order commonality refers to the consistent choice of one par-



GMCR II [Elmira]

Uniroyal

| DMs | Options | | 1 | 4 | 8 | 5 | 9 |
|---|---|---|---|---|---|---|---|
| MoE | 1. Modify | — | N | Y | Y | N | — |
| Uniroyal | 2. Delay | — | Y | N | N | Y | — |
| | 3. Accept | — | N | Y | Y | N | — |
| | 4. Abandon | — | N | N | N | N | Y |
| Local Government | 5. Insist | — | N | N | Y | Y | — |

| | 1 | 4 | 8 | 5 | 9 |
|---|---|---|---|---|---|
| R | | | ✓ | ✓ | ✓ |
| GMR | ✓ | ✓ | ✓ | ✓ | ✓ |
| SMR | ✓ | ✓ | ✓ | ✓ | ✓ |
| SEQ | | | ✓ | ✓ | ✓ |
| L(4) | | | ✓ | ✓ | ✓ |
| NM | | | ✓ | ✓ | ✓ |
| (RON) | ✓ | | ✓ | ✓ | ✓ |
| (NGS) | | | ✓ | ✓ | ✓ |
| (GSR) | | | ✓ | ✓ | ✓ |
| Add Custom Type | | | | | |

Ready

Start

Figure 6.4: Displaying Custom Designed Equilibria

ticular option, or not, over all equilibrium states.

**Second order:** A second order commonality means that, for all the equilibria, their status on two certain options always follows a particular pattern. Hence, if a second order commonality exists on options m and n, it means that one of the following situations has consistently happened on all equilibrium states:

**A)** The status on option m is always the same as that on option n;

**B)** The status on option m is always different to that on option n.

In Figure 6.2, the "Extract Commonalities" button located at the upper-left corner can bring up an information box which summarizes the commonalities found across the equilibrium states. There is no first order commonality among the equilibrium states in Figure 6.2. However, first order commonalities may occur when the model gets larger. The implementation of the discovery of first order commonality is relatively simple - just compare the status of each option in all equilibrium states.

In Figure 6.2, state 9, as explained in Section 4.3.3, actually represents a group of indistinguishable states. The treatment of the "–"s has to involve some conventions. Hence, let us ignore this state in the equilibrium list, and observe all the other equilibria in Figure 6.2. It is easy to identify three second order commonalties, one of Type A and two of Type B:

**Type A commonality:** The status on options 1 and 3 always coincides. Based on this, advice for MoE would be that a modification of the control order would tend to stimulate Uniroyal to accept it.

**Type B commonalities:** 1. The Y-N status against options 1 and 2 is always different. Possible advice that can be drawn from this commonality for

Uniroyal is that once it seizes the initiative to delay, MoE would very likely modify the control order.

2. The Y-N status against options 2 and 3 is always different. The is actually a "backgound commonality" – a commonality applies to all feasible states – that is caused by the user's specification of mutual exclusive options (Section 4.3.2.1).

Currently, it is up to the user's judgement to identify the "background commonality". However, development is ongoing to enable the system to filter out "background" commonalites automatically. The easiest way to implement is to draw commonalites over the feasible state list and subtract them from the commonalities for equilibrium states.

Commonalties of higher order can be defined, but generally increase computing complexity in implementation, while they may not provide much worthwhile information. Thus, they are not considered for incorporation into in GMCR II.

## 6.5 Categorizing Equilibria into Patterns

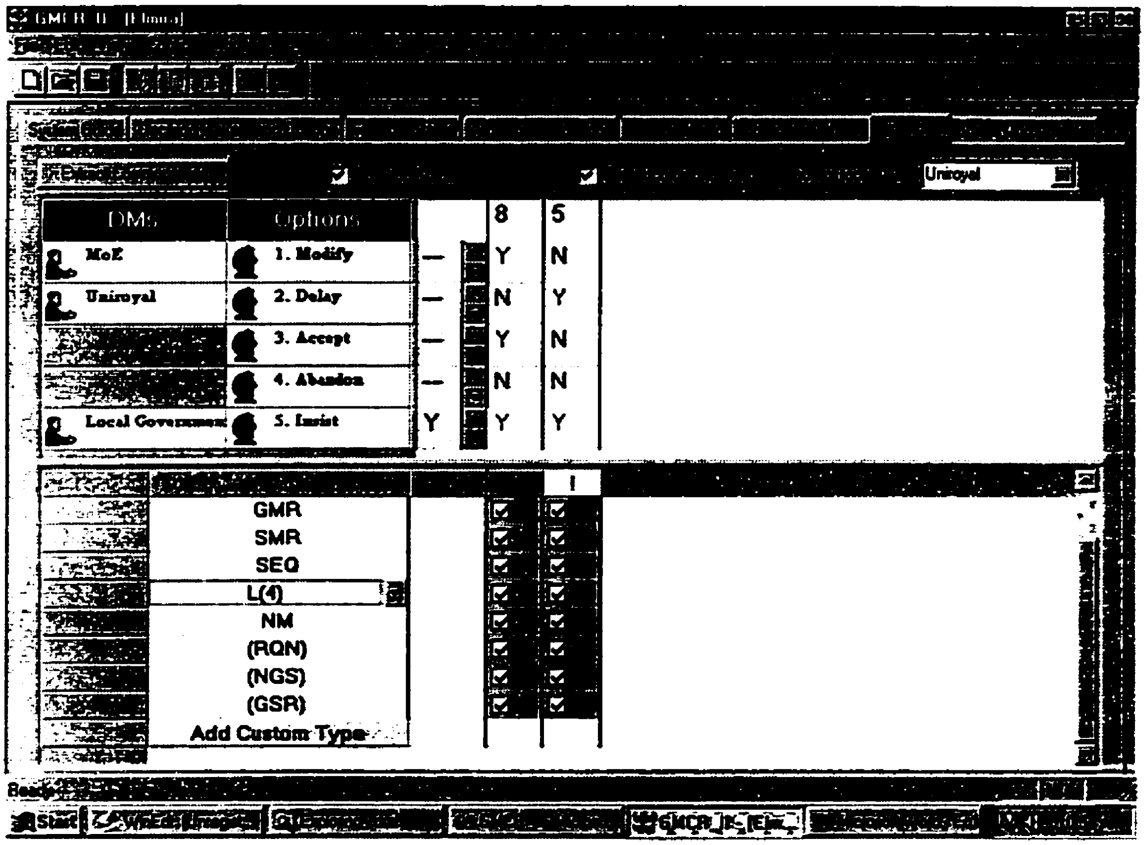
The Elmira conflict is small in size, and a user can examine each equilibrium individually. However, in a large-sized conflict model, especially when the equilibria cannot be displayed on a single screen, a user may wish to specify each time a pattern representing, say, favorable outcomes, and display all equilibria consistent with this specified pattern.

In the Elmira case, for example, if the analyst believes that Uniroyal is concerned about Local Government's attitude toward the original Control Order, he or she may wish to take a close look at those equilibria in which, say, Local Government insists on the original Control Order. On the upper display area of the "Equilibria" property page, the column to the right of the options contains a spin button for each option, by which a "filter pattern" can be specified such that only those equilibria matching this "filter" remain on the screen. In Figures 6.2 and 6.4, this filter is in its default status such that all equilibria of the specified types are shown without further restrictions. In Figure 6.5, however, the pattern is set so as to request the system to show only those equilibria (compared with Figure 6.4) in which Local Government sticks to the original Control Order. The display updates instantly. Recall that pattern specification using spin buttons is first mentioned in Section 4.3.2.3, and the pattern-matching technique (4.2, 4.5), which is conveniently used to implement this display feature, is extensively employed in the modeling subsystem.

## 6.6 Requesting Follow-Up Analyses

The interactive output presentation illustrated in the earlier sections of this chapter allows the user to examine many useful aspect of the output as he or she desires. After utilizing these carefully designed features, the user should be in a better position to understand the implications of the stability results, and hence ready to go forth to take an even closer look in a more active manner. This section will demonstrate how a user can request additional analyses, and how these analyses can generate useful insights and decision advice, of course, provided the user's own judgement is used.

Follow-up analyses are additional analyses that can provide further useful insights of a conflict after the initial run of the stability analysis. Among those types of follow-up analyses described in Chapter 5, sensitivity analyses, hypergame analysis, and dynamic analysis require the establishment of additional models and need



Uniroyal

| DMs | Options | | 8 | 5 |
|---|---|---|---|---|
| MoE | 1. Modify | — | Y | N |
| Uniroyal | 2. Delay | — | N | Y |
| | 3. Accept | — | Y | N |
| | 4. Abandon | — | N | N |
| Local Government | 5. Insist | Y | Y | Y |

| | | I |
|---|---|---|
| GMR | ✓ | ✓ |
| SMR | ✓ | ✓ |
| SEQ | ✓ | ✓ |
| L(4) | ✓ | ✓ |
| NM | ✓ | ✓ |
| (RQN) | ✓ | ✓ |
| (NGS) | ✓ | ✓ |
| (GSR) | ✓ | ✓ |
| Add Custom Type | | |

Figure 6.5: Displaying Equilibria according to a Specified Pattern

to run separate sessions of GMCR II. In this section will concentrate on the other two follow-up analyses: coalition analysis and status quo analysis.

### 6.6.1 *Coalition Analysis*

Section 5.2 provides a new perspective on coalition analysis. In contrast to previous approaches, this new method is based on the impact of a coalition on the outcome, instead of on similarity of preferences. A main objective of this new approach within the graph model paradigm is to identify states that may appear to be stable on an individual basis, but that fail to be coalitionally stable.

In the initial stability analysis, a state is an equilibrium if no individual DM has an incentive to move away from it unilaterally. However, a group of DMs might have both the motivation and ability to depart from an equilibrium. For example, state 5 in the Elmira conflict is a strong equilibrium (refer to Figure 6.2 and Section 2.2.3) - all unilateral moves from it will lead to a state that is less preferred by the DM who makes the move. However, a coalition of MoE and Uniroyal can move from this equilibrium to state 8 (see Figure 5.4), and this move makes both members of the coalition better off. The equilibrium state 5 is, therefore, upset by the coalition move from 5 to 8 (also known as an "equilibrium jumping"). State 5 is said to be coalitionally unstable, even though it is individually stable. This scenario actually coincides with what happened historically [57, 61]. The status quo in mid-1991 was state 1. Local Government shifted quickly to support the original Control Order, resulting in state 5 for a protracted interval of time. Then on October 7, 1991, MoE and Uniroyal dramatically announced an agreement on a modified version of the original Control Order, thus moving to the equilibrium at state 8. This agreement caught Local Government by surprise; its protests were to no avail, and it was forced to reach a separate and quite unfavorable agreement with Uniroyal.

In GMCR II, a user can request a coalition analysis on the existing equilibria, if necessary, to further investigate coalitional stability. States that are equilibria on an individual basis but not coalitionally stable, such as state 5 in the Elmira Conflict, can be identified in the equilibrium list. In Figure 6.2, the "coalition stability" check box is turned on, and hence the coalition analysis is automatically performed over the equilibrium, and each equilibrium state that is coalitionally unstable due to an "equilibrium jumping" is indicated by a cell with a "!" sign appeared at the first row of the lower display area. As can be seen from that figure, state 5 is singled out. The user has the option to ignore the coalition analysis by unchecking the "Coalition Stability" check box. In this case, as can be seen from Figure 6.4, the row originally use to indicate coalition unstability disappears. The GMCR II implementation of this type of coalition analysis related to "equilibrium jumpings" is described in Section 5.2.3.

## 6.6.2 *Status Quo Analysis*

Status quo analysis can be used to find the possible evolution paths from the starting point (Status Quo state) of the model to an equilibrium. The purpose can be

- to verify a predicted outcome (equilibrium) by examining the model's "reachability" to that outcome, from the status quo; or,
- if the answer to the above is true, and
  - if the predicted outcome is desirable, explore possible plans of implementation;

  - if the predicted outcome is undesirable, explore possible plans of deviation, in ways such as a "status quo switch" (Section 5.3) and "equilibrium jumping" (Section 5.2).

The Graph Model is especially suitable for this purpose. An evolution path is not just a viable path in the integrated graph linking the status quo state and the equilibrium under study; rather, the incentive of each DM to join the path based on his/her behavior style is important. In the Graph Model for Conflict Resolution, a more careful study is required for this purpose, since a variety of solution concepts are involved. A natural assumption for a fundamental Status Quo analysis is that each of the directional arcs that form the evolution path must be a unilateral improvement for a DM.

Due to high computing complexity, it would be too aggressive to require the system to automatically identify all possible evolution paths from the status quo to the target equilibrium. Therefore, a step-by-step interactive approach to status quo analysis is adopted in GMCR II. Besides the computing feasibility, the extra benefits of this approach include user involvement and the transparency of the process to the user, both of which are very helpful and thus highly welcome by a user. Figure 6.6 shows the user interface for status quo analysis. The left-most combo box allows the user to specify the state number of the status quo (which is 1 in the Elmira case), either by choosing from the drop list, or by directly entering it. The status quo is the initial "source state". A "move" button is located to the right of the status quo combo box. Clicking on this button brings up a "Moving Options" dialog box as shown in Figure 6.7. The "Moving Options" dialog allows the user to specify

- which DM(s) is considered involved in the current move – by using the mul-

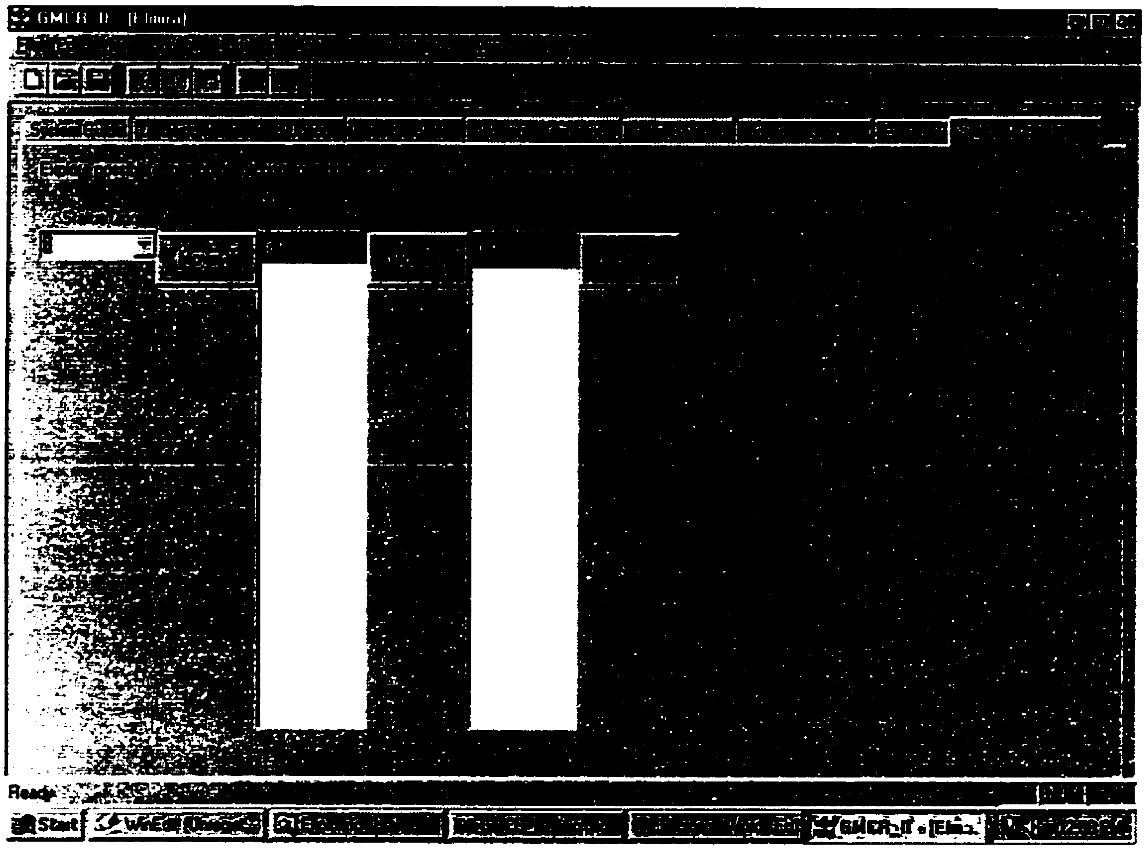


Figure 6.6: User Interface for Status Quo Analysis

tiple selection list of options. In Figure 6.7, for example, all three DMs are highlighted;

- whether to examine improvement only – via the "improvement only" check box. If selected, only UIs of single DMs, or group UIs of multiple DMs are considered in identifying possible target states of this current move; this option is actually recommended.
- whether to consider "equilibrium jumpings" – via the "equilibrium jumping" check box. This option is only applicable when a source state is an equilibrium.

The first specification actually allows the user to specify a group $\mathbf{H} \in \mathbf{N}$, and the second let the user choose to examine $\mathbf{S}_{\mathbf{H}}(u)$ or just $\mathbf{S}^{+}_{\mathbf{H}}(u)$. Of course, a single DM or the whole set $\mathbf{N}$ is also a special case of group $\mathbf{H}$. The third specification permits coalition stability to be considered in the course of status quo analysis.

When the OK button is pushed, and hence the options are confirmed, the "Moving Optioin" dialog box disapears. The system focus is back to the "Status Quo Analysis" property page, and a new multiple selection list box, containing the state number of all the possible target states can be reached by the specified current moves from the source state(s), appears to the right of the current "Move" button, together with another "move" button to its right (Figure 6.6). In the Elmira model, the only state that can be reached from state 1, by individual or group UIs, is state 5 ($\mathbf{S}^{+}_{\mathbf{H}}(u_1) = \{u_5\}$), which is an equilibrium and hence marked with a "*" to its right. By the way, whether a state is considered an equilibrium in this status quo analysis is based on the current equilibrium type setting in the "Equilibrium" property page. In most of the cases, there would be more than one item in this multiple selection list. One then can specify one or more items as the new source state(s),

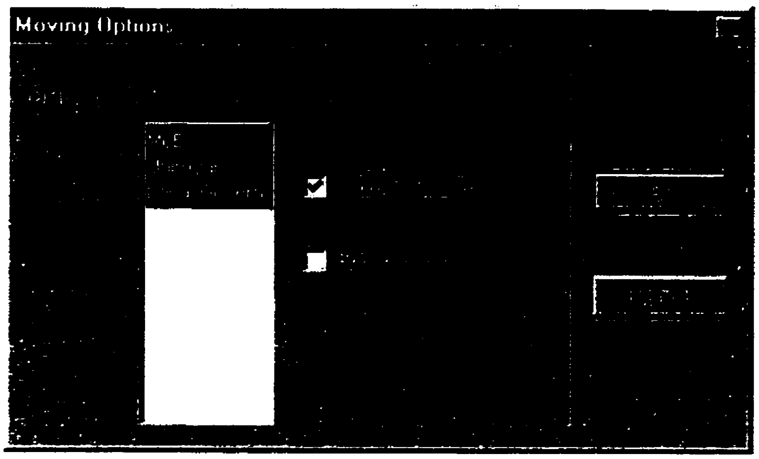


Figure 6.7: Specifying Moving Options in Status Quo Analysis

and then click on the new "Move" button to specify the new current move(s), and so on.

Table 6.1: Preference Ranking for the Elmira Conflict Model

| | *most preferred* ⟶ *least preferred* | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| MoE | 7 | 3 | 4 | 8 | 5 | 1 | 2 | 6 | 9 |
| Uniroyal | 1 | 4 | 8 | 5 | 9 | 3 | 7 | 2 | 6 |
| Local Government | 7 | 3 | 5 | 1 | 8 | 6 | 4 | 2 | 9 |

In the Elmira case, the only state in the list, state 5, is selected as the new source state. Since state 5 is an equilibrium, the option "equilibrium jumping" in Figure 6.7 for the new current move(s) becomes relevant and is selected. The "equilibrium jumping" by the coalition of MoE and Uniroyal thus brings the model from state 5

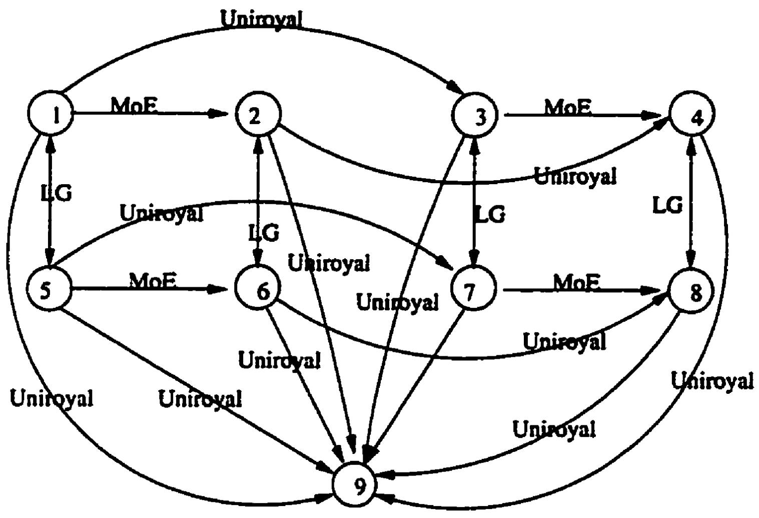


Figure 6.8: Integrated Graph of the Elmira Model

to state 8. It can also be discovered that there would have not been an improvement from equilibrium state 5, were an "equilibrium jumping" not considered. Further attempts to move would find no way out. Once again, the integrated graph of the Elmira model and each DM's preference ranking are reproduced in Figure 6.8 and Table 6.1 for ease of reference.

This status quo analysis on the illustrative Elmira model perfectly depicts how the conflict evolved (all based on reasonable judgements). Initial predictions that are not viable are ruled out, and the remainder is confirmed. The evolution path identified is exactly what happened historically [54, 57, 61]. The newly developed coalition analysis is thus operational for the first time. Much more, could, of course, be done. For example, if the user wants to find out which DM(s) initiates the first move from the status quo to state 5, and which group possibly implements the "equilibrium jumping" from state 5 to state 8, he or she can trace it down by narrowing the multiple selections in the moving options. Most of the analysis methods discussed in Section 5.3 can be facilitated by this interface. The user's necessary judgements are significantly reinforced by the power and flexibility of this type of status quo analysis. The great potential of various usages of the status quo analysis to generate structural insights and decision advices is yet to be discovered.

Depending upon a user's specification, the calculation of the set of target states by a "current move" is basically related to individual UM, $\mathbf{G}_i(u)$, individual UI, $\mathbf{G}_i^+(u)$, group UI, $\mathbf{G}_{\mathbf{H}}^+(u)$, and the "equilibrium jumping". The definitions and implementations of these components can be found in relevant sections in Chapters 4 and 5.

# Chapter 7

# Summary of Achievements and Future Research

## 7.1 Research Contributions

In this research, the author has developed a comprehensive decision support system. GMCR II, for systematically studying strategic conflicts, based on the Graph Model for Conflict Resolution paradigm. During the development of this DSS, new concepts in both the theory and the technology of conflict resolution were defined; modeling, analysis, and interpretation techniques were invented or improved; a range of efficient algorithms were designed and implemented to ensure that real-world conflicts can be effectively studied. As a result, GMCR II constitutes the next generation of a strategic DSS that can provide decision makers and analysts with decision advice, structural insights and answers to what-if questions. With this enhanced understanding, analysts can better explain strategic relationships and assist decision makers, who may have the opportunity to direct the evolution of a conflict toward more favorable results.

More specifically, the contributions of this research include:

- The user-centered design principles and user-friendly interface of GMCR II allow users to interact with the system effectively and efficiently, and make the system widely accessible to both professionals and practitioners.

- The strength of option form is that it is more efficient and contains richer detailed information for a state, while the state-based graph model is more general and more flexible. An improved option form used in the modeling stage efficiently elicits user input and automatically generates the analytical graph model that can be accepted by the analysis engine. This design ensures a loose coupling among the sub-systems of GMCR II: graph-based modeling or other new modeling approaches can be incorporated into the formulation subsystem in the future without changing the analysis engine; meanwhile, new solution concepts can be introduced into the analysis subsystem later, without altering the modeling stage. similar relationship exists between the analysis and interpretation subsystems. This architectural design provides this DSS with great adaptability, flexibility and potential for future improvement, and constitutes an innovative advance over the existing systems for conflict resolution.

- To ensure a smooth coupling between the option form and the graph model, the option form is improved, expanded in useful directions. In particular, the concepts of irreversible moves, forcing moves, and common moves are for the first time represented in the option form, which was earlier considered to be almost impossible.

- Specially designed data structures and corresponding algorithms are developed to facilitate option-based modeling, such as generating possible states, removing infeasible states, coalescing indistinguishable states and calculating allowable state transitions, in a highly efficient manner.

- Systematic approaches for ordinal preference elicitation are developed to effectively elicit a DM's relative preferences over states. Option prioritizing generalizes and improves the preference tree method such that unreasonable restrictions are removed.

- The efficiency of the analysis engine, which implements a variety of stability definitions to model different human behavior, is greatly improved to ensure that large scale real-world conflicts can be effectively analyzed. In fact, GMCR II analyses have been performed on the largest documented conflict model, the Trade in Service model, which was previously analyzed using DecisionMaker. The results perfectly coincide and GMCR II's performance proved to be superb despite the computing overhead caused by the extra features of the system.

- For the first time it is pointed out in this thesis that the correct definition of an equilibrium should allow different stability concepts be applied to different DMs in the same model. This concept actually provides a much richer variety of equilibrium types, and hence significantly contributes to the quality of decision support that the Graph Model for Conflict Resolution can offer.

- A novel approach to coalition analysis is presented, which emphasizes the impact on stability results of the potential coalition rather than preference similarity among coalition members. This approach has added a new dimension to GMCR II, and also provided a framework for a new research direction of conflict resolution theories and methodologies;

- The concept of status quo analysis is significantly expanded and equipped with carefully designed interface, which greatly enhance the system's ability to provide better structural insights and decision advices;

- Besides coalition analysis and status quo analysis, the use of GMCR II to carry out sensitivity analyses, hypergame analysis and dynamic analysis are illustrated.

- Informative presentation and interpretation facilities allow the user to better understand and utilize the analysis results; follow-up analyses facilities integrated into the output presentation and interpretation subsystem provide the user with decision advice, structural insights, and answers to "what-if?" questions.

## 7.2 Future Challenges

As pointed out by Sprague and Carlson [83], a DSS is an adaptive system which requires a unique iterative development approach. In contrast to the typical "prototyping" approach, the iterative development "*becomes the system*" over time. A DSS needs to continue evolving to accommodate much different decision environments, behavioral styles and capabilities. Some opportunities for future research and development are as follows:

- Based on the framework of the novel approach to coalition analysis introduced in Chapter 5, interesting research can be continued in two major directions: First, research can be carried out to take into account counter-responses such that a wide range of different "coalitional solution concepts" can be defined. To achieve this objective, reasonable assumptions must be set based on important theoretical and practical assumptions. Secondly, the "enforcability" of a coalition improvement can be defined based on the sequential information of the improvement as well as each coalition member's solution concept. In other words, much more content could be added to the definition of coalition stability which will make an important contribution to the theory of multiparty conflict resolution, thereby significantly increasing the performance of the related DSSs.

- The graph-based modeling approach proposed and outlined in Chapter 4 can be implemented in connection with the availability of suitable graph-drawing development tools. The brand new modeling and interpretation method based on an interactive graph display will bring a revolution to the arena of decision support for conflict resolution.

- Chapter 5 of this thesis explains that useful follow-up analyses, such as sensitivity analyses, hypergame analysis, and dynamic analysis, can be performed using GMCR II, but currently they have to be done in separate sessions other than the original model. Further development should be carried out to integrate these analyses such that they can be more conveniently and effectively used in decision support under strategic uncertainty.

- Cognitive research should be conducted on how to convey different solution concepts in a manner more acceptable for non-expert users.

- Section 4.4.5 demonstrates that there is no fundamental difficulty to incorporate common moves into the option form modeling of GMCR II. However, the representation issue has to be address such that this can be actually implemented.

- More empirical evaluation of GMCR II, especially operational evaluation [79], can be carried out. This evaluation will guide the future development of the system.

- The design of GMCR II, especially the loose coupling among the formulation, analysis and interpretation subsystems, yields the opportunities for new stability types and new modeling and interpretation methods to be incorporated into the DSS.

# Bibliography

[1] A. V. Aho, J. E. Hopcroft, and J. D. Ullman. *Data structures and Algorithms.* Anddison-Wesley Publishing Company, Reading, MA, 1987.

[2] J. Angus. Negotiate! This software teaches you the fine art of negotiation. *Portable Computing*, 4(3):56, 1990.

[3] R. Anson and M. T. Jelassi. A development framework for computer-support conflict resolution. *European Journal of Operational Research*, 46:181–199, 1990.

[4] K. J. Arrow. *Social Choice and Individual Values.* Yale University Press, New Haven, 2nd edition, 1963.

[5] R. Aumann. Accepatable points in general cooperative n-person games. In Luce R. D. and A. W. Tucker, editors, *Contributions to the Theory of Games IV*, pages 287–324. Princeton University Press, Princeton, NJ, 1959.

[6] P. G. Bennett. Toward a theory of hypergames. *OMEGA*, 5:749–751, 1977.

[7] P. G Bennett, editor. *Analysing Conflict and Its Resolution: Some Mathematics Contibutions.* Oxford University Press, New York, 1987.

[8] P. G. Bennett, A. Tait, and K. Macdonagh. INTERACT: Developing software for interactive decisions. Working Paper 35, Department of Management Science, University of Strathclyde, United Kingdom, 1992.

[9] P. G. Bennett, A. Tait, and K. Macdonagh. INTERACT: Developing software for interactive decisions. *Group Decision and Negotiation*, 3(4):351–372, 1994.

[10] B.D. Bernheim, B. Peleg, and M.D. Whinston. Coalition-proof Nash equilibria: I. Concepts. *Journal of Economic Theory*, 42:1–12, 1987.

[11] B.D. Bernheim and M.D. Whinston. Coalition-proof Nash equilibria: II. Applications. *Journal of Economic Theory*, 42:13–29, 1987.

[12] S. J. Brams. *Game Theory and Politics.* The Free Press, New York, 1975.

[13] S. J. Brams. *Theory of Moves.* Cambridge University Press, Cambridge, U. K., 1994.

[14] S. J. Brams and D. Wittman. Nonmyopic equilibria in 2 × 2 games. *Conflict Management and Peace Science*, 6(1):39–42, 1981.

[15] J. Bryant. Studios front end. *Coperateion or Conflict: The Research Supplement*, 10(1), Jan. 1996.

[16] W.D. Cook and M. Kress. *Ordinal Information and Preference Structures: Decision Models and Applications.* Prentice-Hall, Inc., Englewood Cliffs, NJ, 1992.

[17] E. De Bono. *Conflicts: A Better Way to Resolve Them.* Harrap, London, 1985.

[18] G. Di Battista, P. Eades, R. Tamassia, and I. C. Tollis. Algorithms for drawing graphs: An annotated bibliography. *Computational Geometry: Theory and Applications*, (4):235–282, 1994. Also available as PostScript file from ftp.cs.brown.edu.

[19] L. Fang, K. W. Hipel, and D. M. Kilgour. Conflict models in graph form: Solution concepts and their interrelationships. *European Journal of Operational Research*, 41(1):86–100, 1989.

[20] L. Fang, K. W. Hipel, and D. M. Kilgour. *Interactive Decision Making: The Graph Model for Conflict Resolution.* Wiley, New York, 1993.

[21] L. Fang, K. W. Hipel, D. M. Kilgour, and X. Peng. "Scenario generation and reduction in the decision support system GMCR II". In *Proceedings of the 1997 IEEE International Conference on Systems, Man and Cybernetics*, Orlando, Florida, U.S.A, October 1997.

[22] L. Fang, K. W. Hipel, D. M. Kilgour, and X. Peng. Presenting output information in the decision support system gmcr ii. In *Proceedings of the Symposium on Industrial Engineering and Management, Canadian Society for Mechanical Engineering (CSME) Forum, held at Ryerson Polytechnic University*, Toronto, Ontario, Canada, May 1998.

[23] N. M. Fraser. Applications of preference trees. In *Proceedings of the 1993 International Conference on Systems, Man and Cybernetics*, pages 132–136, LeTouquet, France, October 1993.

[24] N. M. Fraser. Ordinal preference representations. *Theory and Decision*, 36:45–67, 1994.

[25] N. M. Fraser and K. W. Hipel. Solving complex conflicts. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-9:805–817, 1979.

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

[27] N. M. Fraser and K. W. Hipel. Decision support systems for conflict analysis. In M. G. Singh, K. Hindi, and D. Salassa, editors, *Managerial Decision Support Systems, Proceedings of the IMACS/IFORS 1st International Colloquium on Managerial Decision Support Systems*, pages 13–21, North Holland, Amsterdam, 1988.

[28] N. M. Fraser and K. W. Hipel. Decision making using conflict analysis. *OR/MS Today*, 16(5):22–24, 1989.

[29] N. M. Fraser and K. W. Hipel. DecisionMaker: The conflict analysis program. University of Waterloo, Waterloo, Ontario, Canada, 1993.

[30] S. Gauvin, G. L. Lilien, and K. Chatterjee. The impact of information and computer based training on negotiator's performance. *Theory and Decision*, 28:331–354, 1990.

[31] K. W. Hipel, editor. *Multiple Objective Decision Making in Water Resources.* American Water Resources Association (AWRA) Monograph Series No. 18, Herndon, Virginia, 1992.

[32] K. W. Hipel, L. Fang, D. M. Kilgour, and M. Haight. Environmental conflict resolution using the graph model. In *Proceedings of the 1993 IEEE International Conference on Systems, Man and Cybernetics*, volume 1, pages 153–158, Le Touquet, France, October 1993.

[33] K. W. Hipel and N. M. Fraser. Cooperation in conflict analysis. *Applied Mathematics and Computation*, 43:181–206, 1991.

[34] K. W. Hipel, N. M. Fraser, and A. Cooper. Conflict analysis of the trade in service dispute. *Information and Decision Technology*, 4(16):347–360, 1990.

[35] K. W. Hipel, D. M. Kilgour, L. Fang, and X. Peng. Using the decision support system GMCR for resolving conflict in resource management. In S. A. El-Swaify and D. S. Yakowitz, editors, *Multiple Objecttive Decision Making for Land, Water and Environmental Management, Proceedings of the Malama 'Aina 95, First International Conference on Multiple Objective Decision Support Systems for Land, Water, and Environmental Management: Concept, Approaches and Applications.* St. Lucie Press, July 1995. published in 1997.

[36] K. W. Hipel, D. M. Kilgour, L. Fang, and X. Peng. Resolving water resources conflicts using the decision support system GMCR II. In *Proceedings of the International Conference on Water Resources and Environment*, volume II, Kyoto, Japan, October 1996.

[37] K. W. Hipel, D. M. Kilgour, L. Fang, and X. Peng. The decision support system GMCR in environmental conflict management. *Applied Mathematics and Computation*, 83(2 and 3):117–152, 1997.

[38] K. W. Hipel, D. M. Kilgour, L. Fang, and X. Peng. Conflict management for the services industry. In *Proceedings of the 1998 IEEE International Conference on Systems, Man and Cybernetics*, pages 4769–4775, La Jolla, California, U.S.A, October 1998.

[39] K. W. Hipel, D. M. Kilgour, L. Fang, and X. Peng. "Strategic decision sup-

port for the services industry". 1998. Submitted to IEEE Transactions on Engineering Management.

[40] K. W. Hipel and D. B. Meister. Conflict analysis methodology for modelling coalitions in multilateral negotiations. *Information and Decision Technology*, 19(2):85–103, 1994.

[41] N. Howard. *Paradoxes of Rationality.* MIT Press, Cambridge, Mass., 1971.

[42] N. Howard. The CONAN Manual, 1986. Copyright by Nigel Howard Systems, 10 Bloomfied Road, Birmingham, England B13 9BY.

[43] N. Howard. CONAN 3.0. Nigel Howard Systems, 10 Bloomfield Road, Birmingham. England, B13 9BY, 1989.

[44] N. Howard. Soft game theory. *Information and Decision Technologies*, 16(3):215–227, 1990.

[45] N. Howard. Negotiation as drama: How 'games' become dramatic. *International Negotiation*, 1:125–152, 1996.

[46] M. Jarke. M. T. Jelasi. and M. F. Shakun. MEDIATOR: Towards a negotiation support system. *European Journal of Operatinoal Research*, 31(3):314–334, 1987.

[47] M. T. Jelassi and A. Foroughi. Neogotiation support systems: An overview of design issues and existing software. *Decision Support Systems*, 5:167–181, 1989.

[48] D. M. Kilgour. Equilibria for far-sighted players. *Theory and Decision*, 16:135–157, 1984.

[49] D. M. Kilgour. Anticipation and stability in two-person noncooperative games. In M. D. Ward and U. Luterbacher, editors, *Dynamic Models of International Conflict*, pages 26–51. Lynne Rienner Press, Boulder, Colorado, 1985.

[50] D. M. Kilgour. Book review: Theory of moves. *Group Decision and Negotiation*, 4:283–284, 1995.

[51] D. M. Kilgour. Seperable and non-seperable preferences in multiple referenda. *Presented at Public Choice Society*, San Francisco, CA, March 1997.

[52] D. M. Kilgour, L. Fang, and K. W. Hipel. General preference structures in the graph model for conflicts. *Information and Decision Technologies*, 16(4):291–300, 1990.

[53] D. M. Kilgour, L. Fang, and K. W. Hipel. Analyzing the Flathead River resource development dispute using the graph model for conflicts. In M. G. Singh and L. Travé-Massuyès, editors, *Decision Support Systems and Qualitative Reasoning, Proceedings of the IMACS International Workshop on Decision Support Systems and Qualitative Reasoning, Toulouse, France*, pages 101–110, 1991.

[54] D. M. Kilgour, L. Fang, and K. W. Hipel. The decision support system GMCR and the management of strategic uncertainty. In *Proceedings of the Fifth International Conference IPMU: Information Processing and Management of Uncertainty in Knowledge Based Systems*, volume 2, pages 638–643, Paris, France, July 1994.

[55] D. M. Kilgour, L. Fang, and K. W. Hipel. GMCR in negotiations. *Negotiation Journal*, 11(2):151–156, 1995.

[56] D. M. Kilgour, K. W. Hipel, and L. Fang. The graph model for conflicts. *Automatica*, 23(1):41–55, 1987.

[57] D. M. Kilgour, K. W. Hipel, L. Fang, and X. Peng. A new perspective on coalition analysis. In *Proceedings of he 1996 IEEE International Conference on Systems, Man and Cybernectics*, volume III, pages 2017–2022, Beijing, China, October 1996.

[58] D. M. Kilgour, K. W. Hipel, L. Fang, and X. Peng. GMCR - The next generation in negotiation support. In H. Pirkuland and M. J. Shaw, editors, *Proceedings of the First INFORMS International Conference on Information Systems and Technology*, pages 152–156, Washington D.C., May 1996.

[59] D. M. Kilgour, K. W. Hipel, L. Fang, and X. Peng. "Applying the decision support system GMCR II to peace operations". In *CORNWALLIS II: Analysis for and of the Resolution of Conflict*, Pearson Peacekeeping Training Centre, Cornwallis Park, NS, Canada, April 1997.

[60] D. M. Kilgour, K. W. Hipel, L. Fang, and X. Peng. Peace support, GMCR II, and bosnia. In *Proceedings of the Cornwallis III Conference held at the Lester B. Pearson Canadian International Peacekeeping Training Centre, Corneallis Park, Clementsport, Nova Scotia, Canada*, 1998.

[61] D. M. Kilgour, X. Peng, L. Fang, and K. W. Hipel. Coalition analysis in the decision support system GMCR II. In *INFORMS 1997 Spring Meeting*, San Diego, U.S.A, May 1997.

[62] H. W. Kuhn. Extensive games and the problem of information. In H. W. Kuhn and A. W. Tucker, editors, *Contributions to the Theory of Games*, volume II of *Annals of mathematics Studies, 28*, pages 193–216. Princeton, N.J., 1953.

[63] J. R. D. Kuhn, K. W. Hipel, and N. M. Fraser. A coalition analysis algorithm with application to Zimbabwe conflict. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-13(3):338–352, 1983.

[64] J-P Langlois. Decision Systems Analysis. San Fransisco State University, September 1994. Version 3.3.

[65] D. B. Meister and N. M. Fraser. Conflict analysis technologies for negotiation support. *Group Decision and Negotiation*, 3(3):333–345, 1994.

[66] D. B. Meister, K. W. Hipel, and M. De. Coalition formation. *Journal of Scientific Industrial Research*, 51(8-9):612–625, 1992.

[67] S. S. Nagel and M. K. Mills. Multicriteria dispute resolution through computer aided mediation software. *Mediation Quaterly*, 7(2):175–189, 1989.

[68] J. F. Nash. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences of the U.S.A.*, 36:48–49, 1950.

[69] J. F. Nash. Noncooperative games. *Annals of Mathematics*, 54(2):286–295, 1951.

[70] J. F. Nunamaker Jr. Experience with and future challenges in GDSS (group decision support systems). *Decision Support Systems*, 5:115–118, 1989.

[71] X. Peng, K. W. Hipel, D. M. Kilgour, and L. Fang. Representing ordinal preferences in the decision support system GMCR II. In *Proceedings of the 1997 IEEE International Conference on Systems, Man and Cybernetics*, Orlando, Florida, U.S.A, October 1997.

[72] C. Petzold. *Programming Windows 3.1.* Microsoft Press, Redmond, Washington, third edition, 1992.

[73] C. Petzold and P. Yao. *Programming Windows 95: The Definitive Developer's Guide to the Windows 95 API.* Microsoft Press, Redmond, Washington, fourth edition, 1996.

[74] K. J. Radford. *Strategic and Tactical Decisions.* Springer-Verlag, New York, 2nd edition, 1988.

[75] K. J. Radford, K. W. Hipel, and L. Fang. Decision making under conditions of conflict. *Group Decision and Negotiation*, 3:169–185, 1994.

[76] A. Rapoport and M. J. Guyer. A axonomy of 2 x 2 games. *General Systems*, 11:203–214, 1966.

[77] A. Rapoport, M. J. Guyer, and D. G. Gordon. *The 2 x 2 Game.* The University of Michigan Press, Ann Arbor, MI, 1976.

[78] J. E. Rubin. *Mathematical Logic: Applications and Theory.* Saunders College Publishing, Philadelphia, 1990.

[79] A. P. Sage. *Decision Support Systems Engineering.* Wiley, New York, NY, 1991.

[80] G. Sander. Graph drawing tools and related work. Jan. 1998.

[81] H. A. Simon. *Models of Man: Social and Rational, Mathematical Essays on Rational Human Behavior in a Social Setting.* Wiley, New York, 1961.

[82] M. G. Singh, J. B. Singh, and M. Corstjens. MARK-OPT–A negotiating tool for manufaturers and retailers. *IEEE Transactions on Systems, Man and Cybernetics*, 15(4):483–495, 1985.

[83] R. H. Jr. Sprague and E. D. Carlson. *Building Effective Decision Support Systems*. Prentice Hall, Englewood Cliffs, N.J., 1982.

[84] R. H. Jr. Sprague and B. C. McNurlin. *Information Systems Management in Practice*. Prentice Hall, Englewood Cliffs, N.J., 3rd edition, 1993.

[85] E. M. Thiessen and D. P. Loucks. Computer-assisted negotiation of multi-objective water resources conflicts. *Water Resources Bulletin*, 28(1):163–177, 1992.

[86] J. Von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, Princeton, N. J., U.S.A., 1st edition, 1944.

[87] J. Von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, Princeton, N. J., U.S.A., 3rd edition, 1953.

[88] M. Wang, K. W. Hipel, and N. M. Fraser. Modeling misperceptions in games. *Behavioral Science*, 33(3):207–223, 1988.

[89] F. W. Winter. An application of computer decision tree models in management-union bargaining. *Interfaces*, 15(2):74–80, 1985.

[90] E. Yourdon. *Object-Oriented Systems Desigri, A Intergrated Approach*. Prentice-Hall, Eaglewood Cliffs, NJ, 1994.

[91] F. C. Zagare. Limited-move equilibria in $2 \times 2$ games. *Theory and Decision*, 22:1–19, 1984.