Towards the Qualitative, Plan-based Simulation of International Crises

Wolfgang Balzer
Institut für Philosophie, Logik und Wissenschaftstheorie, Universität München, Ludwigstr.31, D-80539 München

Jörg Sander
Institut für Informatik, Universität München, Oettingen Str.67, 80538 München

Manuscript 1997

ABSTRACT

We describe an algorithm for the calculation of whether a given state is the origin of a crisis or not. The algorithm is based on a qualitative, plan-based model of crises described in (Balzer, 1996) which is outlined as far as necessary. The algorithm is intended to serve as a basis for implementation. In the final part of the paper, we discuss perspectives and problems of the implementation of simulation- and support programs for international crises.

Keywords: Simulation, Decision Support Systems, Artificial Intelligence, Systems

Introduction

We describe the specification for a program to be used for the qualitative simulation of international crises based on a model of binary crises developed by Balzer, Sander and Gayhoff.[1] The model being complex we cannot repear here its detailed description. For a full account including substantive discussion and examples see (Balzer, 1996).

The model differs from other approaches to the conceptualization of crises, like (Brecher, 1977), (Brecher & Wilkenfeld, 1988, 1997), (Hermann, 1972), (Lebow, 1981), (Leng & Singer, 1988), (Singer, 1968) in several respects. Also, our approach to simulation is different from other simulation studies of social affairs, like (Eberwein,1990), (Fehr, Rosenberg, Wiegard, 1992), (Forrester, 1971), (Hegselmann, 1994), (Helling & von Wachter, 1996), (Huber & Miller, 1996), or (Mallery & Sherman, 1993). Our model is a micro-model centering on the plans of the parties involved, on the ways they choose new plans and delete blocked plans, and on their perceptions of their possibilities. For reasons of space we cannot discuss and compare all these differences in approach. Instead, we will point out those salient features of the present model which are not found in

[1]This research was funded by DFG-project Ba 678/4-1. The present specification is described in full detail in the technical report (Sander,1993) which can be obtained from the authors.

other approaches, and in Sec.1 we will describe the notion of a crisis captured by that model only as far as needed for understanding the algorithm.

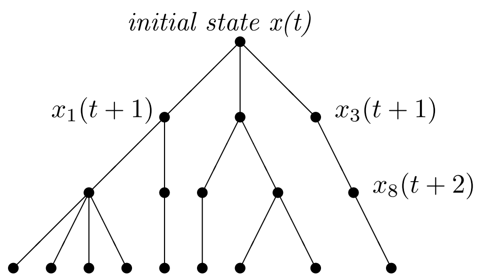
The model is, firstly, entirely qualitative in the sense of not using numerical equations to steer the dynamics. Instead, the system's dynamics is driven by a set of rules or procedures (i.e. sets of conditions and consequences) in the way found in expert systems or in knwoledge based systems. When the conditions get satisfied during a simulation run the rule fires, and its consequences are added to the database. In this approach, the only use of numbers is in the representation of facts in actions and plans.

Second, the model is built on an ontology of propositions, not of numbers. We take serious the ontological perspective of decision theory which assumes that alternatives are conceptualized as propositions, see (Luce & Raiffa, 1957), (Jeffrey, 1965), without committing ourselves to the particular behavioral and rationality assumptions made in decision theory. In the context of simulation, models being based on propositions raise the question of how to deal with propositions on the computer. Propositions -which we may identify with sentences for the sake of simplicity- are basically given by instantiated verbal phrases, and thus by verbs as occurring in natural language. There are many thousands of verbs in a natural language which can be used to generate a set of propositions exponentially exploding with the number of items used to instantiate the verbal phrases, like dates, persons, names of events, plans, beliefs and the like. Even with their substantially increased power present day computers quickly reach the bounds of their capacity in such a setting. In order to overcome this difficulty we propose to use a small base of action types which are relevant for crises. Such a base can be constructed using systems of categories developed by crisis researchers, e.g. in (Brecher, 1977), appendix, or (Pfetsch, 1994). Such types can be instantiated in various ways to yield a multitude of tokens and at the same time can be easily processed by means of rules referring to types only, not to tokens.

In our system the states of the parties involved in a crisis are represented in terms of propositions. This requires to check the consistency of sets of propositions. When an action is planned and a group decides to perform that action it also checks whether the action is feasible. A group usually would not decide for some alternative which is inconsistent with the facts the group knows about the present state of the world. We suggest to use a 'one-sided' consistency check based on a list of pairs of propositions which is empirically determined. Each pair in the list represents incompatible propositions, and the consistency of pairs of propositions not occurring in the list is checked by comparing them to pairs in the list by means of simple rules. If a given pair is not found similar to a pair in the list the propositions in the given pair are treated as being consistent. This is of course a notion of consistency much weaker than the ordinary, logical notion.

1 A global dynamical picture of crises

Considering some given state $x(t)$ of an international system at time t we may look for all its possible successor states $x(t+1)$. For each possible successor state we may again calculate its successor states, $x(t+2)$, and so forth. In this way we obtain a tree-like graph as depicted in figure 1 below. The top node represents the initial state, each node represents a possible state and the arcs indicate the relation of being a possible successor.
Fig.1



Running through one path in that graph from top to bottom amounts to running through one of those processes that might have originated from the initial state. The events taking place during a crisis correspond to one such process, that is, to a sequence (path) of states which occurs in a graph originating from a given state. However, one such path is not a model of the crisis. We must distinguish between the events taking place during a crisis, and the crisis itself which is a more comprehensive entity. The actual events are not sufficient to capture what is essential to the notion of a crisis. In order to describe a crisis, we have to refer to the *whole* graph, and to the *possible* paths or developments as well.

On our account a crisis is represented by such a graph if and only if *every* path from the origin contains a *crash state*, that is, a state in which force is exerted by at least one of the parties involved. Thus, a crisis is not given simply by one path of events. It is given by a *set* of *all possible* paths of events -all originating from one state- all of whose members contain a crash state. A less idealized notion of crisis requires that *almost all* paths, or a high percentage of paths, contain a crash state. In a different formulation, we can say that a crisis is implicit already in one single state, namely the initial state $x(t)$ of the graph. Such a state is critical, or $fatal$ as we shall say, if every path from it contains a crash state. Thus, the essential feature of a crisis is the *potential* to end in violence on whatever course of action pursued. A fatal state necessarily, or with high probability leads to the real development of a crisis. The path of the graph describing the *actual* development will end in violence because every path does so (or most paths do). This means that even if the events would take a different course violence would not, or hardly, be avoided, as soon as the initial, fatal state is realized.

When confronted with the many real crises that have been studied (Brecher & Wilkenfeld, 1988, 1997), (Lebow, 1981), (Leng & Singer, 1988), (Pfetsch, 1994) this picture seems to cover only a special case, for most real crises, in fact, do not end in violence or even full scale war, (Brecher, 1997), p.859. In many crises there is not even a threat to use force. However, the meaning of violence has little bearing on the overall structure just described. Violence in the structure is represented by 'crash states', i.e. states which are *interpreted* as involving the use of force. The only structural feature of this model is that crash states are distinguished as a sub-species of states in general. Extending the interpretation of these distinguished states we might speak of 'undesired' states without changing the formal structure of the model. Thus even those crises which do not tend to violence and in which there is no threat of using force may be directly accomodated by our model as long as they tend to some distinguished, undesired states. However, in order to stress the importance of violent crises we will stick to the 'violence' terminology.

Still, it could be objected that even under the 'non-violence' interpretation crises often end or are settled before the unintended states are reached. We believe that it is true also for these crises that they *tend* to reaching these unintended states. We therefore believe it is good strategy to focus on the 'ideal type' of crises which actually end in undesired states, and to obtain other, less 'complete' forms as special cases of the 'ideal' model. Starting from a model of 'full' crises' we can develop precise ideas of what it means for a system *to tend* to unintended states. On the basis of such a model we then can construct a more special model capturing crises that are settled before unintended states are reached. Such a model is described in (Balzer, 1996).

Our primary interest is not in the particular path which represents the actual course of events. We take it that the main goal of crisis research -at least in science- is to improve crisis prevention. In the language of graphs this means to find criteria for determining whether a given, original state of such a graph will be the origin of a crisis, i.e. a fatal state. Crisis prevention consists in detecting fatal states. If a fatal state is detected by a group before it gets realized the group may try to avoid it by taking decisions that lead to a state different from the fatal one. (The informal or even intuitive ability of leaders to recognize fatal states seems to account for most crises being settled without violence.) At the present, abstract level a fatal state is defined as a state such that all, or almost all paths originating from it contain a crash state in which at least one party resorts to force.

In order to bring this approach in line with ordinary talk we will identify a crisis not with a fatal state but with the path of real events starting from a fatal state, *as related* to the set of all possible paths from that state. We first define the graph of all possible paths originating from $x(t)$. Second, we define the notion of a fatal state by reference to crash states. This yields the notion of the graph of a fatal state. Finally, a crisis is defined as a path in the graph of a fatal state. That is, we define c to be a *crisis* iff there exists some fatal state $x(t)$ such that c is a path originating from $x(t)$.

This account imposes considerable structure on the notions of state and successor state. If every path originating from $x(t)$ contains a crash state then the *sets* of successor states must converge in a certain sense. They get 'smaller' in the sense of containing less and less non-crash states, where a non-crash state is a state that does not involve violence. 'Does not involve' here means that none of the propositions occurring in the chosen plans or in the realities describes a violent action (see below). In the following, we will take a state to consist -among other things- of sets of plans. The notion of a crash state then can be reduced to that of a *crash plan*, i.e. a plan the execution of which involves the use of force. (Under the 'non-violence' interpretation the execution of a crash state involves *undesired* actions or events.) A crash state is a state in which at least one of the parties involved is left only with plans that are crash plans, that is, the party has no plan left for execution that would not include the use of force. Clearly, if the number of non-crash plans, those not including force, decreases, violence and crash states become more probable. During a crisis the non-crash states vanish because during the passage from one period to the next some of the non-crash plans chosen in the first period either are successfully finished or become impossible to execute, mostly because of interference of the enemy. In this way more and more non-crash plans will be eliminated from the successor states and in the end only crash plans are left and a crash state is reached.

The notions of a crash state and a crash plan are less problematic and will be taken for granted. In general and with sufficiently coarse graining of time it is not difficult to tell whether a given plan includes the use of force, and whether a state is violent or not. So the burden of our definition of a fatal state and of crisis falls on the notion of a successor state, and this, in turn, requires to make explicit the internal structure of states. We will in the following specify the internal structure of the states and accordingly explain what is a successor state of a given state. For reasons of space we cannot present the full range of details, see (Sander,1993).

The previous explanations are stated at an analytical or conceptual level. Of course, one would like to include in a crisis model assumptions of more substantive, behavioral nature, for example an assumption stating that each party g tries to block those plans of the opposing parties which have harmful consequences for g. Such behavioral rules are *not* included in the present account. The specification presented here is most general in this respect. Put differently, we might say that our model of crises is a $frame\ model$ which can be specialized in many different ways by adding more special assumptions about the dynamics in a crisis. As there are many different special forms of crises, each form being characterized by its own, particular set of assumptions or 'dynamical laws', these assumptions cannot simply be put together in the general model. This would yield straightforward contradiction. The idea of specialization is to claim that special assumptions hold true only in a small part of the full domain of the theory's applications, that is, in a *restricted* domain of special applications (compare (Balzer, Moulines, Sneed, 1987), Chaps. 4 and 5). We want to stress that the methodological view originating from Popper has been superseded for a long time now, and may be called false with good reasons. The Popperian picture according to which scientists try to find universally true 'laws' is at best a heroic myth. What really happens in science (as proved by numerous case studies) is that scientists offer some new hypothesis as valid for a given, small set of *intended* systems, and after that try to extend the hypothesis' range of validity by checking whether it applies to 'new' systems that have not yet been investigated. If such an application fails the hypothesis is by no means 'falsified'; of course it remains valid for all the systems to which it was successfuly applied.

As already indicated, we think that the detection of fatal states is the primary end to which crisis research can contribute. From case studies we know that with hindsight it is often possible to say that the parties involved in a crisis could have seen that they are approaching a fatal state though actually they did not take action to avoid that state. Illuminating case studies are found in (Lebow, 1981). However, such considerations heavily rely on human understanding, and the evaluation of beliefs and perceptions. At present there is no formal tool that can support such evaluations, and the objection might be raised that it is hard to see how computer programs can be used and applied here. We think there are two ways in which programs can really help. First, a computer is much more powerful in searching all possible paths than is any group of humans. On the computer, the calculation of fatal states becomes possible, if sufficient data are available. Thus the computer may serve to replace, or back, intuitive considerations, completing the picture and pointing to alternatives that have been overlooked or gone unperceived. Second, by pointing to such alternatives which have been found by blind computation power the users in crisis management might be stimulated to think in a more disciplined way, trying to find further alternatives, and trying to check different perceptions which otherwise would go unnoticed. Our specification shows that such computations have become feasible now. It also shows that an important reason for this is the qualitative approach in which the 'calculation' of possible paths essentially proceeds in the way of theorem provers.

2 States

In order to generate states and successor states we consider two *groups*, g, g', to be interpreted as the leading groups of two states involved in a crisis. We restrict analysis to the binary case, the more general cases can be treated analogously but lead to considerably more complexity. The groups are not further analyzed with respect to their inner structure, they play the role of single actors. The groups can perform certain actions which lead to a change of state. Each state occurs at a point of time, or rather at a short period of time, t.

Each group acts in two ways. On the one hand, it chooses plans for execution and checks whether chosen plans are still feasible. The finite set of plans which at time t are chosen by group g we call $choice(g, t)$. The members of this set

$$choice(g, t) = \{P_1, ..., P_n\}$$

are plans P_i which are chosen by g for execution. A plan is *chosen* by a group if, whenever the conditions necessary for continuing in the execution of the plan are satisfied the group will actually continue in its execution. On the other hand each group has its own perception or image of reality, being given by a set of propositions which are believed to be true by the group. The set of propositions which group g at time t believes to be true we call g's *reality* at time t and denote it by $real(g,t)$.

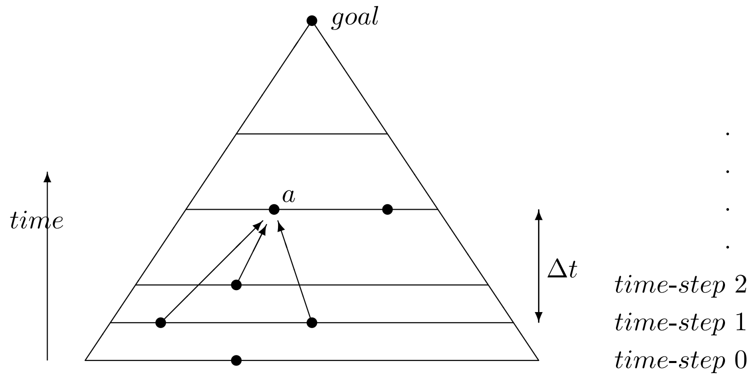
$$real(g,t) = \{a_1, ..., a_m\}$$

consists of propositions a_i with the interpretation that a_i, or the state of affairs described by a_i, is believed to true by group g at t.

The basic stuff all these components consist of are propositions. We assume that a finite set $\mathcal{P}$ of verbal phrases is given which describe action types and types of events. From each member of $\mathcal{P}$ propositions expressing concrete actions or events (tokens) can be obtained by adding suitable names (e.g. for actors and instances). Propositions can be of arbitrary complexity. For instance, a_i might express that the 'other' group g' believes that group g will not attack first, and a_i's being a member of $real(g,t)$ then would express: group g believes that group g' believes that g will not attack first.

Propositions are used to construct plans. Roughly, a plan is a tree like graph the nodes of which are propositions, as depicted in figure 2. The top node represents the plan's goal, the horizontal lines indicate time-steps. All actions on one such line have to be performed simultaneously, at one instant, and the vertical distance between two such lines denotes the time Δt that passes between the two corresponding instants. Moreover, for each node a (except those at the first, bottom time-step) we distinguish a set of nodes at the previous time-steps with the interpretation that exactly these nodes (describing actions or events) taken together will cause the event described by a, plus some action which also is specified in the plan. In figure 2 three nodes causing a are depicted. The time structure of a plan does not employ an 'objective' time scale but is relativized to the plan. The times and time distances internal to a plan are represented by integers, each plan beginning at time-step zero. When plans are executed during a simulation run they have to be appropriately embedded into the objective time scale of the simulation. This is done by mapping each plan's time-steps one-one into an 'objective' time scale.

Fig.2



Our account centers on plans. The development over time during a crisis is characterized by the choice of new plans, the performance of chosen plans, and the deletion of plans which have become impossible to perform. From the ingredients just described we can define a set of *all* plans, namely plans which can be constructed in terms of the propositions, groups, and times which are assumed to be given, and in terms of the further primitives just introduced, (see for instance (Kamphampati, 1994), (Wilkins, 1988) for notions of plans used in AI).

In addition to the notion of choice of plans we use a notion of feasibility for plans. A plan is *feasible for group g at time t* iff it is *chosen* by g and its time structure can be embedded into the objective time scale such that 1) the plan's first time-step 0 is identified with t, 2) all conditions (actions or events) occurring at the plan's first time-step are true in the sense of belonging to group g's reality $real(g,t)$, and 3) no other condition in the plan is incompatible with what is believed by the group at t. In the following, the particular definition of this notion will not matter, and we will treat it like a primitive. We write $feasible(g,t)$ to denote the set of all plans which, at t, are feasible for group g.

Finally, we characterize a *crash state* as a state in which all plans chosen by, and feasible for, one party are crash plans, where a crash plan is characterized as a plan containing (descriptions of) violent actions.

3 Perception and Successor States

It is general wisdom in crisis research that the perceptions which the parties involved have of themselves, their enemies, and the world, are essential and often decisive for the development of the crisis. We provide room for perceptions in two ways. First, we model a state not as consisting of objectively chosen plans and realities of both parties, but as consisting of the chosen plans and the realities *as perceived by both parties.* In principle, we should consider even the perception which a group has of the other group's choices and reality but for reasons of simplicity we will concentrate on each group's environment as perceived by *that* group. Compare (Balzer, 1996) for a more general treatment.

Let us introduce, for any time t, the *planning environment of group g at time t as perceived by g*, denoted by $PE_g(t)$. This environment is defined as the list

$$\langle choice(g,t), real(g,t), feasible(g,t)\rangle.$$

In order to include both groups' perceptions we have to model a state by two planning environments, one for each group. Thus a state $x(t)$ is defined to be a pair

$$\langle PE_g(t), PE_{g'}(t)\rangle$$

where g and g' denote the two groups under consideration. Implicit in this definition are the underlying sets of propositions and points of time which form the basis for the construction of both groups' plans and realities.

We can now define, for a state $x(t)$, the set of all its successor states, or, in other words, the notion of an arbitrary successor state of $x(t)$. Assuming time to be represented by integers, a *successor state* of $x(t)$ is a state $x(t+1)$ present at time $t+1$ which 1) is identical with $x(t)$ in the underlying set of propositions but 2) differs from $x(t)$ in the components *choice* and *real*. In first approximation we may say that *choice* and *real* get updated for both groups, i.e. the sets $choice(g,t), real(g,t), choice(g',t)$ and $real(g',t)$ are updated and replaced by new sets $choice(g,t+1), real(g,t+1), choice(g',t+1)$ amd $real(g',t+1)$.

We characterized fatal states as states all of whose paths contain a crash state, crash states being defined as having only crash plans chosen by at least one party. Now if we want to define or to calculate successor states there are two options. We first might allow for 'new' things, external events, to happen, things that were not predetermined by the previous state. On this account groups may choose new plans or delete plans that previously were chosen, or the world may change and create changes in the groups' realities. On the other hand, we might concentrate only on those successor states that are fully determined by the previous, given state. If we admit successor states obtained by new decisions or external influences the range of successor states will become very large and the probability of finding fatal states may decrease considerably. On the other hand, considering only successor states that are fully determined by the previous state yields a complete, analytic overview of the consequences of one state. We will focus here on the second alternative, and we will not allow for successor states to include newly chosen plans or new, externally triggered beliefs (see Sec.5 for comments on the first alternative).

On this account the updating of chosen plans becomes rather simple. In a successor state $x(t+1)$ exactly those plans from the previous period are still chosen which have not been finished in the previous period t. In other words: any plan chosen in $x(t)$ whose goal is not satisfied at t is taken over and also chosen at $t+1$, and no other plans are chosen at $t+1$. So the *choice* sets can only become smaller. Some chosen plans get satisfied and then are eliminated.

The updating of both groups' realities is much more complex. One reason for this is that we have no good knowledge yet of how beliefs are updated in the real world. In the present context, the task is to update beliefs on the basis of previous beliefs and of changes in the sets of chosen plans. The general methods for updating (for instance, (Gärdenfors, 1988), (Katsuno & Mendelzon, 1992)) do not take this particular setting into account and thus are not suitable for our purpose. We therefore developed a new algorithm by which the adjustment of beliefs is closely tied to the development of the sets of chosen plans.

We use a rather procedural presentation, focusing on the changes in the involved sets of propositions. It is not very difficult to transform this specification to be applicable for other programming paradigms. For instance, the *SWARM* simulation system (Minar et al., 1996) has become very popular for the simulation of complex adaptive systems. This is an object oriented framework which offers some advantages when implementing a concrete system (e.g. predefined librabries for some simulation specific tasks, or data collection tools). However, we think that for the purpose of presentation a 'lean' procedural model yields the most comprehensible and concise statement of the overall picture.

4 An Algorithm for Belief-Updating

We begin with the two planning environments of the two groups at time t:

$$PE_g(t) = \langle choice(g,t), real(g,t), feasible(g,t)\rangle,$$
$$PE_{g'}(t) = \langle choice(g',t), real(g',t), feasible(g',t)\rangle,$$

and we want to construct new realities $real(g, t+1)$, $real(g', t+1)$ for both groups in the following period.

Our algorithm for this proceeds in several steps. For reasons of computational simplicity we first deal with the updating of events which are not actions, that is, 'material' events which have been caused by actions or by other events during the execution of plans. In order to see why this leads to simplification we have to be more precise about the interpretation of a plan's being *chosen* by a group at time t. This does not mean that the group will start to execute the plan at t. We allow for 'conditional' plans which are chosen at t but depend on some condition which has to be realized before the plan can be executed. Our interpretation of *choice* is such that, if the condition for a plan chosen at t is realized at t *then* the plan will be executed with certainty. Clearly, according to this notion, a plan may be chosen at t but not executable at t simply because it depends on a condition which is not satisfied at t. Typical examples are contingency plans which are worked out for situations of emergency and are permanently 'chosen' without being executed (except in a real case of emergency). In the development of a crisis chosen plans may become feasible, their conditions may get satisfied. Accordingly, the respective group will perform some action which, in turn, has effects on the enemy. As a group may have chosen several such plans, and as we want to play through all possibilities of action sequences on both parties' sides we have to regulate the hypothetical order of hypothetical actions. If we would allow one group to perform all actions which are required by all its chosen plans which have become feasible at t this might lead to a rather severe restriction of the other group's alternatives. In reality, as both groups act simultaneously it is not likely that one group can perform all actions of all chosen plans which have become feasible. In order to obtain a somewhat more realistic range of hypothetical interactions, our algorithm allows each group to execute just one of those ('newly feasible, chosen') plans in one step and then to update its situation in the light of the enemy's reaction to the effects of this plan, before executing the next such plan. Now the separation between actions and events that are not actions in this context saves us several consistency checks. If, in the first steps of the algorithm, we would consider actions we later would have to check, for each plan and its execution, whether the actions required are consistent with actions previously selected in the algorithm.

The following steps are depicted in the flow diagram in figure 3), and also are stated in pseudo code in the appendix. The numbering here corresponds to the numbering in figure 3). In figure 3), each line containing the symbol g^* stands for two such lines in which g^* is replaced by symbols for the two groups: g and g', respectively. The expressions printed in capital letters, $REAL$, AUX, $EXPECTED$, FE, A, $EXCT$ are variables which take different values when the program is running. $REAL$ is a variable for realities, $EXPECTED$ for sets of expected propositions, FE for sets fo feasible plans, and, $EXCT$ for sets of executed plans, AUX and A are auxiliarly variables for sets of plans and sets of propositions. Italicized, small letter expressions stand for primitives, and expressions printed in fat for terms which can be explicitly defined in terms of the primitives. **add** X **consistently to** Y means that X, Y are sets of propositions, and that some subset Z of X is added to Y such that $Y \cup Z$ is consistent and maximal in this respect.

Fig.3

$REAL(g^*) := \{a/a \in real(g^*, t) \wedge a$ **is not about** $t+1\}$ (1)

↓

$AUX := \emptyset$
$EXPECTED := \{b/b \in$ **expected**$(g, t) \cap$ **expected**$(g', t) \wedge$ b **is about** $t + 1 \wedge b$ **is not an action** $\}$ (2)

↓

$AUX :=$ **add** $EXPECTED$ **consistently to** AUX
$REAL(g^*) := REAL(g^*) \cup AUX$ (3)

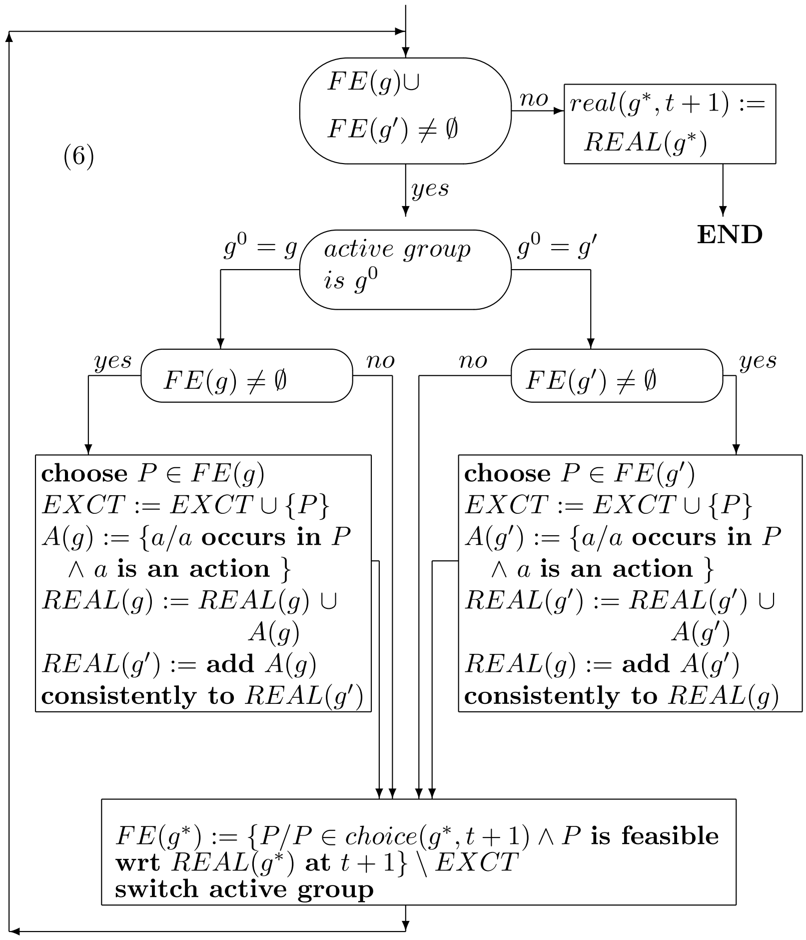
↓

$REAL(g^*) :=$ **add** $\{a/a \in real(g^*, t) \wedge a$ **is about** $t + 1 \wedge a$ **is not an action** $\}$ **consistently to** $REAL(g^*)$ (4)

↓

$FE(g^*) := \{P/P \in choice(g^*, t+1) \wedge P$ **is feasible wrt** $REAL(g^*)$ **at** $t+1\}$, $EXCT := \emptyset$
choose active group g^0 **randomly** (5)

↓



In step (1), those propositions are eliminated from $real(g,t)$ and $real(g',t)$ which 'are about' time $t+1$. Recall that the propositions, say, in $real(g,t)$ describe the beliefs of group g at time t. Among these there may well be beliefs about time $t+1$, for instance, the proposition 'At $t+1$ the enemy will attack' (i.e. group g at t believes that g' will attack at $t+1$). In order to adjust the beliefs about the next period $t+1$ we first delete those that are present in period t, and in later steps we introduce new, adjusted beliefs about $t+1$. Some of these may of course be identical with beliefs previously deleted.

In step (2), we consider all propositions which 1) are about time $t+1$, 2) are expected consequences of plans chosen and feasible at t and 3), do not describe actions. These propositions are calculated for both groups and are put into one big set. The notion of an expected consequence of a plan used in 2) is itself rather involved, and requires to go into all the details of plans which cannot be presented here. Intuitively, a plan consists of what we call *plan-elements*: triples of the form $\langle A, a, C\rangle$ where A denotes *assumptions* or conditions that must be satisfied, a describes the *action* which then is performed, and C the *consequences* resulting from that action. Moreover, a plan contains the duration of each plan-element, i.e. the time that passes from the realization of the 'first' condition to the realization of the 'last' consequence. In a plan, many such plan-elements are nested and their relations in time are made fully explicit. In order to calculate the expected consequences of a plan which is chosen and feasible at t we calculate, for any plan-element beginning at t in the plan, its consequences, i.e. the propositions in its set C. These propositions form the expected consequences of the plan, and the times for which they are expected are calculated from the information about the plan. The propositions calculated in step (2) do not describe actions, they only cover events which are expected as the results of previous actions. This is in line with the separation of actions and 'pure' events explained at the beginning of this section.

Step (3). The set of propositions created in step (2) is likely to be inconsistent, for it lumps together the expected consequences of both groups. In the third step, for each group, a consistent subset of the set of all expected consequences obtained in step (2) is constructed. The resulting set of propositions is obtained in a non-deterministic way. The algorithm used for its construction (labeled **add consistently to** in figure 3) takes an *arbitrary* proposition from those created in step (2) and checks whether this proposition can be consistently added to the set of propositions already present in AUX. Therefore, the result in step (3) depends on the random order in which new propositions are picked out and checked.

In step (4), those propositions in each group's reality which are about time $t+1$ and do not describe actions, are consistently added to the sets of propositions obtained in step (3). Recall that in step (1) *all* propositions about time $t+1$ have been removed from the realities. Now a consistent part of them is re-established, namely a part which is consistent with the set created in step (3) for each group. In other words, each group tries to keep a maximally consistent set of beliefs about time $t + 1$, removing as few of them as possible.

In the next step, (5), we determine for each group the set of plans chosen by the group at time $t + 1$ which still are feasible for the group in the light of the new reality created in steps (1) to (4). We begin with the set of plans chosen by g at $t + 1$ which was obtained by the updating rule for chosen plans described in the previous section. A plan was called feasible for group g at t if the plan's initial assumptions belong to the group's reality at t, and its other assumptions are consistent with the group's reality at t. This definition refers to the group's reality. At the present stage feasibility of plans will of course not be checked with respect to the original realities $real(g, t), real(g', t)$ but with respect to the realities updated by the previous steps. That is, whether a plan is feasible for g, now is checked on the basis of the set of propositions calculated for g in step (4). As the result of step (5), each group has a new, updated set FE of plans which it may consider at $t+1$. A change may have occurred in the set of feasible plans because plans, say, in $choice(g, t + 1)$ may have turned out as being not feasible with respect to the updated 'reality' of group g or because some chosen plans have become feasible in the meantime.

Up to now we dealt with propositions which do not describe actions, that is, propositions describing effects of previous actions. If a group believes that certain actions have been performed it can infer the corresponding effects from its knowledge of plans, and it will come to believe that these effect have occurred. (In our present model plans are deterministic, so strict inference is possible. If plans contain probabilistic elements an outcome a and its negation $\neg a$ both may have positive probability. This considerably more complicated situation requires a substantial extension of our model, in particular a component dealing with the addition of probabilities (with all its problems known in AI).) The previous steps were concerned with such effects, and thus do not cover the actions which are actually performed by the group (and by the other group). Clearly, descriptions of these actions must be included in the groups' realities. The reality of, say, group g at $t+1$ should include propositions describing those actions which g has performed during period $t+1$. In principle we might take the set of 'chosen' plans from step (5), calculate all the actions both groups will perform at $t+1$ according to these plans, and add all these actions to the groups' realities. However, a set of plans which are feasible for one or the other group may be inconsistent in the sense that assumptions in a plan of group g may be inconsistent with assumptions in a plan of group g', and this kind of inconsistency has not been eliminated in the previous steps. So at the point reached now the new sets of 'feasible' plans may be incompatible, and we cannot simply collect the actions as occurring in all plans that are present. These (descriptions of) actions might contradict each other.

In order to achieve a reasonable space of possibilities of actions we have to proceed step by step and to check for consistency in each step. Also, it would not be 'fair' to first calculate a maximal set of actions for one group, add this to the other group's reality, and only after that calculate the other group's possible actions. For in this way the 'other' group's array of possible actions will get rather severely restricted. In order to obtain a balance here we have to go back and forth between the two groups, considering in each step only those actions that are performed according to one single plan. This leads to the following loop which is performed alternately for the two groups. Alteration is provided by declaring one of the groups randomly as the 'active' group in step (5), and at the end of step (6) in the box at the bottom to switch this status to the respective other group so that the other group is the active one in the 'next' round.

Step(6): First, the escape condition for the loop is checked. As long as one of both groups still has feasible plans ($FE(g) \cup FE(g') \neq \emptyset$) the loop is entered through 'yes'. Only when all feasible plans of both groups have been processed the loop stops and the 'last' reality, $REAL(g^*)$, is taken as the desired value of $real(g^*, t+1)$. Entering the loop, there is a check for the active group. As there is complete symmetry in g and g' thereafter, we may consider just one of them, say g. Passing through the left-hand exit of the check '*active group is* g^0' we reach the next check: $FE(g) \neq \emptyset$. In the first case, if group g still has feasible plans, we come into the left-hand, big box, and the following happens.

A plan P is randomly choosen from those obtained for group g in step (5), and this plan is put into the dump '$EXCT$' of executed plans. All actions occurring among that plan's initial assumptions (the members of $A(g)$) are added to group g's set of propositions (its intermediate 'reality'). As these are actions from g's 'own' plans consistency is guaranteed by the way we deal with plans. The other group g' then also tries to add all these initial actions from P to its intermediate reality. Here, inconsistency is possible, and g' can add only a consistent subset of $A(g)$. With the new sets of propositions $REAL(g), REAL(g')$ so obtained (the new 'realities') this box is left and we come into the box at the bottom. Here, both groups re-evaluate feasibility. That is, they check which of the plans obtained in step (5) are still feasible with respect to the new sets $REAL(g), REAL(g')$ of propositions. The dependency of feasibility on such a set of propositions was explained previously. Those plans executed before (members of $EXCT$) are removed from the feasible plans, and the respective 'other' group g' is activated by **switch active group**. This box is also reached when, further up, no feasible plan had existed for g, and $FE(g) \neq \emptyset$ was left through the 'no' exit.

Now with the new sets of plans $FE(g), FE(g')$ obtained in the box at the bottom the loop starts again from above. This time, the 'other' group g' is the active one and execution proceeds along the right-hand side. After a plan was chosen from g''s plans, and the realities were updated according to this plan's actions, in the next loop group g is called up again, and performs step (6) now starting from the set $FE(g)$ of plans created in the previous turn. When step (6) was executed for a plan of group g, next a plan of the other group g' is called up, and its actions are added to both groups' (intermediate) realities as just described. After that, again a plan of group g is taken, and so on.

The loop in step (6) terminates for the following reason. The set of feasible plans $FE(g^*)$ in step (5), on which (6) begins, is finite, for $choice(g^*, t+1)$ is finite. Though $FE(g^*)$ may get larger in some executions of (6), in the long run this set must become smaller and finally empty because in every round at the end of (6) some plan from this set is executed and then gets eliminated (as a member of $EXCT$ in the lowest box of (6)).

When step (6) has terminated, new sets of propositions for both groups have been created, which are taken as their new, updated realities. We now have new sets $choice(g, t+1), real(g, t+1), choice(g', t+1), real(g', t+1)$ for period $t+1$. From these, the corresponding feasible plans can be determined, and we obtain a full, new state at time $t+1$ consisting of two planning environments for both groups. We thus have described an algorithm to construct a new state $x(t+1)$ from a given state $x(t)$.

In several steps of the algorithm random choices are made, and for each such choice a new path is opened in the overall graph. As each state is given by a finite configuration, it has only finitely many successor states, i.e. each branching point in the graph represents a finite branching. By itself, this does not guarantee the finiteness of the whole graph, however. In a non-crisis development, for instance, two kinds of states might alternate 'for ever' in each group. Put differently, in a 'normal' development there usually are non-crash plans which are not eliminated by the actions of the respective other group. This means that the development does not have the features of a crisis.

The presentation just given already contains a simplification of our real algorithm. In our original specification we use the additional notion of a *characterization*. Roughly, a characterization serves to replace a given and possibly rather abstract goal by several, more detailed propositions which, when taken together, may serve to characterize that goal. In this way we can deal with plans having rather abstract goals for which it is difficult to construct plan-elements that causally lead to the satisfaction of the goal. In such cases, our model allows to replace the goal by one of several possible characterizations. The propositions making up one characterization are much more concrete and may well be the goal of concrete plans.

5 Problems of Application

In order to implement the model described and in particular the above algorithm two problems as mentioned in Sec.1 have to be solved. First, we have to find a way of representing the propositions which are relevant to describe the actions and events during a crisis, and second, we need a tool which can be used to check consistency or compatibility of propositions.

For the solution of the first problem we suggest to work with a fixed and relatively small set of action types, each type being given by one verbal phrase. Of each type the computer can create and use many tokens by instantiating the type with names for suitable entities (dates, persons, events etc.) but the rules by which those tokens are processed during a simulation run basically are formulated in terms of types, and therefore in an economical way. Recently, students of crises have found several small sets of verbal phrases which seem to be sufficient (perhaps even complete) in order to describe what is going on during a crisis (see for instance (Brecher, 1977), appendix, and (Pfetsch, 1994), especially pp.42-56.) These systems can be used as providing a basis of verbal phrases for our approach. (This strategy of dealing with propositions has also proved fruitful in the simulation of social affairs where a similar problem comes up for the representation of actions.) For example, a general action type is given by the verb 'to attack'. If we agree to use this as a four place predicate in the form 'A attacks B at t with M' with 'A' as (a name for) the attacker, 'B' for the attacked, t for the time of attack and M as a -possibly complicated- list of items or means used in the attack (units, equipment, route etc.), then many different tokens are obtained by specifying the four variables to names of concrete objects. One token would be 'Germany attacks Poland in 1939 by army and air force', another would be 'The second battalion of the 5. tank division at XX.YY.1939 attacks the Polish frontier post at ZZ'. These tokens show that our approach is very flexible in dealing with descriptions at different levels of abstraction. Clearly, the second token describes just a small part of the action described by the first one. But under certain conditions one also would say that the first token expresses the same content as the second one, namely when the frontier post at ZZ in fact was the first being attacked when the war began.

The second problem, of checking the consistency or 'truth' of sets of propositions can be treated once a representation format for propositions is fixed. At the present state of computer science one would like to work with propositions of the kind of sentences of first-order predicate calculus, and use theorem provers (like OTTER (McCune, 1988) or SETHEO (Letz et al., 1992), (Ibens & Letz, 1996)) to check whether a sentence is true or consistent with other sentences. However, theorem provers are incomplete in not being able to perform such checks for arbitarary first-order sentences, and -what is more important- sentences in a crisis-model are typically *not* first-order sentences and thus cannot be processed by most present theorem provers.

We suggest to solve this problem by using a logic of propositions containing strong empirical elements. We begin with a base of pairs of propositions, where each pair in that base consists of propositions that are incompatible with each other, and a primitive system of rules for checking consistencies of arbitrary propositions by comparing them to pairs in that base. Two given propositions are compatible with each other if, by application of the rules, their pair is not found similar to a basic pair. For instance, the base might contain the pair ($attack$, $defend$) -with appropriate additional specifications of the arguments of these verbs depending on the chosen format- and the pair ($attack$, $retreat$) might be found similar to the first one by means of a rule saying that retreat is a special case of defence. This would mean that the action types of attacking and defending are incompatible with each other, and that the action types of attacking and retreating also will be found as incompatible by the system.

Besides these two problems crucial for the application, we want to discuss some further points concerning possibilities of extending our system. First, a practical problem arises from the computational complexity of our algorithm. In the flow-diagram of figure 3 (and the program in the appendix) there are several branching points at which a state has several different successor states, and the corresponding graph will contain several branches 'downward' from that state. Such branching occurs in all applications of the predicate **add consistently to**. Each such application begins with two given sets X, Y of propositions which usually are incompatible. A subset of propositions $Z \subseteq Y$ is then chosen such that X and Z are consistent. Clearly, this procedure in general is exponential in the number of propositions of Y. Another such branching point is given by the precidate **choose** where a plan is arbitarily chosen from a given set of plans.

Due to these features a program obtained by simply implementing our specification will have to be limited to rather small sets of propositions in order to be run on a (even large) computer. Considerable reduction in complexity can be expected, however, when more special forms of crises and crisis-dynamics are considered. In special cases, we may use techniques known from theorem proving, like setting priorities or pruning, to cut down the complexity.

A second point is the potential use of our model for crisis management. Our system primarily was developed as a comprehendable model or picture of crises; its computer implementation primarily is intended as a tool for the theoretical study of crises by means of simulation. These applications are quite different from real-life crisis management. However, we believe that our approach also could be used for the build-up of a crisis management tool for a particular group. The basic ingredient in this application would be a set of plans as really chosen by the group, a set of plans as really believed to be chosen by the other group, and a set of sentences as really believed by the group to be true. These sets would have to be implemented beforehand, in a possibly lengthy procedure in which the systems managers would have to be supported by real members of the group in order to get the relevant information.

Yet there is another, purely scientific possibility of application along these lines. Instead of getting the real plans and beliefs out of a group presently existing and active, one may try to extract data from historical crises, like Suez, Falkland, Corea, Cuba etc. In each case, by means of substantial, historical effort it seems possible to construct rather large data-bases containing descriptions of the chosen plans and beliefs of one or both groups. By running our program on such a data-base we might compare the possible developments produced by the program and the evaluation of the initial state with respect to being fatal with the actual course of events which took place in the real world. In this way the program can be tested and improved. The problem with this approach is that the production of data for one single case involves working power in the order of man-years eventhough substantial amounts of data are available (see, e.g. (Brecher & Wilkenfeld, 1988), (Lebow, 1981), (Pfetsch, 1994)). As the material in the literature is not presented in the primitives of our model, still substantial work is needed to extract actions, plans, believed propositions, choices etc. as data for an implementation. In (Balzer, 1996) a small example set of data from the Cuban missile crisis is described in the format needed here but a much larger set would be needed to allow for meaningful simulation runs.

The present algorithm contains a ‘closed world assumption’. All possible developments of the initial state are calculated purely in terms of information about that state. There is no further external interference of the world. In reality such external influences exist of course. There are external causes not residing in the actions of both groups, causes which may change the realities and the arrays of feasible plans of both groups. Also, each group may make up and choose a completely new plan, which was not contained in the initial stock of plans. When these possibilities are explicitly added to our model, the likelyhood of an initial state’s being fatal decreases.

Formally, it is not difficult to include this feature into our model. We simply have to admit a certain amount of new, additional plans and beliefs at each time in the development of the system, plans and believes which are not systematically linked to what happened in the system before, but which come up accidentally, out of the blue. In applications, this creates the additional problem of collecting and feeding in information about these external facts in real time when the system is running. While this further diminishes the probability of application in real crisis-management, historical applications as addressed in the previous paragraph could be adjusted by including data about external events which are felt relevant for the crisis under study.

We stress that our algorithm represents just one ‘simulation function’ out of many possible other such functions. By a simulation-function we mean any function which maps a given state into a set of successor states. Different algorithms will create different such functions. We just want to mention that among the possible simulation-functions there are functions much more specific than the one we used. For instance, other such functions might use a more biased behavior of the groups when they choose and perform their feasible plans in step (6) of the above algorithm. One group might be agressive and prefer to choose plans with goals directly linked to violence.

Finally, one of the referees pointed out that our exclusive reference to international crises is unconvincing. Indeed, it seems that the algorithm might possibly be applied to other kinds of crises as well, crises in which the agents' behavior is essentially determined by their choices of plans. Such crises might be found in the interaction of large corporate actors, whether intra- or international. However, our study is located in the field of international crises and we have no good judgement about applicability in other areas. Anyway, we do not think that a possibly extended domain of application of the algorithm speaks against it, nor does it have implications for its applicability to international crises.

Appendix

The following is a specification for a program that updates the two groups' beliefs (realities). Expressions indicating the flow of control in the program are printed in small-sized capital letters: WHILE, DO, IF THEN, ELSE, ENDIF, END and ENDWHILE

(1a) $REAL(g) := \{a/a \in real(g,t) \wedge a$ **is not about** $t+1\}$
(1b) $REAL(g') := \{a/a \in real(g',t) \wedge a$ **is not about** $t+1\}$

(2a) $AUX := \emptyset$
(2b) $EXPECTED := \{b/b \in expected(g,t) \cup expected(g',t) \wedge b$ **is about** $t+1 \wedge$ b **is not an action** $\}$

(3a) $AUX :=$ **add** $EXPECTED$ **consistently to** AUX
(3b) $REAL(g) := REAL(g) \cup AUX$
(3c) $REAL(g') := REAL(g') \cup AUX$

(4a) $REAL(g) :=$ **add**$\{a/a \in real(g,t) \wedge a$ **is about** $t+1 \wedge a$ **is not an action** $\}$ **consistently to** $REAL(g)$
(4b) $REAL(g') :=$ **add**$\{a/a \in real(g',t) \wedge a$ **is about** $t+1 \wedge a$ **is not an action** $\}$ **consistently to** $REAL(g')$
(5a) $FE(g) := \{P/P \in choice(g,t+1) \wedge P$ **is feasible wrt.** $REAL(g)$ **at** $t+1\}$
(5b) $FE(g') := \{P/P \in choice(g') \wedge P$ **is feasible wrt.** $REAL(g')$ **at** $t+1\}$
(5c) $EXCT := \emptyset$

(5.d) **set** g^* **randomly equal to** g **or** g'

(6.1) WHILE $FE(g) \neq \emptyset \vee FE(g') \neq \emptyset$ DO
(6.2) IF $g^* = g$ THEN
(6.3) IF $FE(g) \neq \emptyset$ THEN
(6.4) **choose** $P \in FE(g)$
(6.5) $EXCT := EXCT \cup \{P\}$
(6.6) $A(g) := \{a/a$ **occurs in** P **and is an action** $\}$
(6.7) $REAL(g) := REAL(g) \cup A(g)$
(6.8) $REAL(g') :=$ **add** $A(g)$ **consistently to** $REAL(g')$
(6.9) ENDIF
(6.10) ELSE
(6.11) IF $FE(g') \neq \emptyset$ THEN
(6.12) **choose** $P \in FE(g')$
(6.13) $EXCT := EXCT \cup \{P\}$
(6.14) $A(g') := \{a/a$ **occurs in** P **and is an action** $\}$
(6.15) $REAL(g') := REAL(g') \cup A(g')$
(6.16) $REAL(g) :=$ **add** $A(g')$ **consistently to** $REAL(g)$
(6.17) ENDIF
(6.18) ENDIF
(6.19) $FE(g) := \{P/P \in choice(g, t+1) \wedge P$ **is feasible wrt** $REAL(g)$ **at** $t+1\} \setminus EXCT$
(6.20) $FE(g') := \{P/P \in choice(g', t+1) \wedge P$ **is feasible wrt** $REAL(g')$ **at** $t+1\} \setminus EXCT$
(6.21) **switchgroup**(g^*)
(6.22) ENDWHILE

References

Balzer, W. 1996. A Theory of Binary Crises. In: (Huber & Avenhaus, 1996), pp. 233-52.

Balzer, W., Moulines, C. U., Sneed, J. D. 1987. An Architeconic for Science, Reidel, Dordrecht.

Brecher, M. 1977. Toward a Theory of International Crises Behavior, *International Studies Quarterly 21*, 39-73.

Brecher, M., Wilkenfeld, J., Moser, S. (Eds.) 1988. Crises in the Twentieth Century: Handbook of International Crises, 2 Volumes, Pergamon Press, Oxford.

Brecher M., Wilkenfeld, J. 1997. A Study of Crises, Michigan University Press, Ann Arbor.

Eberwein, W.-D. 1990. Globale Trends und Strukturbrüche: Weltmodelle als Forschungsinstrumente, Berlin: WZB papers.

Fehr, H., Rosenberg, C., Wiegard, W. 1992. Quantitative Aspekte einer Harmonisierung der Umsatzsteuern in der EG. In: Hansmeyer, K.-H. (Ed.), Ausgewählte Probleme der EG-Finanzen, Duncker & Humblodt, Berlin, pp. 11-113.

Forrester, J.W. 1971. World Dynamics, Wright-Allen, Cambridge/Mass.

Gärdenfors, P. 1988. Knowledge in Flux, MIT Press, Cambridge/Mass.
Hegselmann, R. 1994. Zur Selbstorganisation von Solidarnetzwerken unter Ungleichen. In: Homann, K. (Ed.), Wirtschaftsethische Perspektiven I, Duncker & Humblodt, Berlin, pp.105-29.
Helling, K. von Wachter, S. 1996. Crisis Management Gaming and Computer Support. In: (Huber & Avenhaus, 1996), pp. 291-302.
Hermann, C.F., (Ed.) 1972. International Crises: Insights from Behavioral Research, Free Press, New York.
Huber, R.K., Avenhaus, R. (Eds.), 1996: Models for Security Policy in the Post-Cold War Era, Nomos, Baden-Baden.
Huber R.K., Miller, G. 1996. A Decision-Theoretic Model of Peace Support Strategies. In: (Huber & Avenhaus, 1996), pp. 259-77.
Ibens, O., Letz, R. 1996. Subgoal Alternation in Model Elimination, Technical Report, Institut für Informatik, TU München, see also http://www.jessen... /forschung/reasoning/setheo.html
Jeffrey, R.C. 1965. The Logic of Decision, McGraw Hill, New York.
Kamphampati, S. 1994. Multi-contributor causal structures for planning: a formalization and evaluation, Artificial Intelligence 69, 235-78.
Katsuno, H., Mendelzon, A.O. 1992. On the Difference Between Updating a Knowledge Base and Revising it. In: Gärdenfors, P. (Ed.), Belief Revision, Cambridge Tracts in Theoretical Computer Science 29, pp. 183-203.
Lebow, R.N. 1981. Between Peace and War, Johns Hopkins UP, Baltimore.
Leng, R.J., Singer, J.D. 1988. Militarized Interstate Crises: the BCOW Typology and its Applications, International Studies Quarterly 32, 155-73.
Letz, R., Schumann, J., Bayerl, S., Bibel, W. 1992. SETHEO: A High-Performance Theorem Prover, Journal of Automated Reasoning 8, 183-212.
Luce, R.D., Raiffa, H. 1957. Games and Decisions, Wiley, New York.
Mallery, J.C., Sherman, F.L. 1993. Learning Historical Rules of Major Power Intervention in the Post-War International System, contribution to the 1993 Annual Meeting of the International Studies Association, Mexico.
McCune, W. 1988. OTTER's Users Guide, Technical Report, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne Ill.
Minar, N., Burkhart, R., Langton, C., Askenazi, M. 1996. The Swarm Simulation System: A Toolkit For Building Multi-Agent Simulations, http://www.santafe.edu/projects/swarm.
Pfetsch, F.R. 1994. Datenhandbuch nationaler und internationaler Konflikte, Nomos, Baden-Baden.
Sander, J. 1993. Theoretische Grundlagen für rechnergestützte Krisensimulation und -Beratung, Forschungsberichte aus dem Institut für Statistik und Wissenschaftstheorie der Universität München, Nr.44.
Singer, J.D., (Ed.) 1968. Quantitative International Politics: Insights and Evidence, Free Press, New York.
Wilkins, D.E. 1988. Practical Planning, Morgan Kaufmann, San Mateo.