CONFLICT RESOLUTION BETWEEN AGENTS: A BELIEF-THEORETIC PERSPECTIVE

by

Rajan Srikanth
Leonard N. Stern School of Business
Information Systems Department
New York University
90 Trinity Place
New York, NY 10006

July 1989

Center for Research on Information Systems
Information Systems Department
Leonard N. Stern School of Business
New York University

Working Paper Series

CRIS #212
STERN #89-81

Table of Contents

1. Introduction

Researchers seeking to build intelligent systems to solve real-world problems, have begun to move from the "single, intelligent agent" paradigm toward the "multiple, interacting agents" approach [Davis and Smith, 1983; Cammarata et al, 1983; Corkill and Lesser, 1983; Rosenschein and Genesereth, 1985; Ferrante, 1985]. Problems are modelled in terms of the 'distributed problem-solving' framework; there is no conflict of interests or goals, and agents freely exchange information and resources, to find solutions to shared problems.

These conditions, often do not hold in real-world situations. They are better modelled as a 'society' of interdependent, intelligent - sometimes specialized - systems (knowledge sources and problem solvers), often competing to safeguard conflicting 'opinions' or interests, but cooperating because of their interdependence. Strategic defense systems, sophisticated control systems for spacecraft and industrial robots, distributed remote-sensing applications, management of computer and communications networks, and devices for computer-assisted business management, are examples of applications that could be modelled along these lines.

In this paper, we are concerned with cooperative problem-solving situations, where conflict among agents exists but is not total; there is potential for 'integrative solutions' [Pruitt and Lewis, 1977; Walton and McKersie, 1965] or non-zero-sum outcomes. We address, in particular, the issue of how an agent conceptualizes conflict in a "dyadic" (two-agent) context, and the 'conflict resolution strategies' that may be used to reach solutions.

We model conflict resolution as problem-solving activity occuring in the 'belief-space' of an agent. Interdependence between agents implies a shared problem-space. The agent and its "opponent" impose constraints on the problem-space either explicitly or implicitly. An overconstrained state where there is no "feasible solution" represents a state of conflict. Conflict resolution is the process by which the problem-space is altered through an adjustment of constraints so that a solution emerges. Our focus is on the individual agent and how it goes about this conflict resolution activity.

The rest of the paper is organised as follows: in the next section, we justify the use of the 'problem-solving' paradigm to study conflict resolution. We then show how a 'beliefs' model provides flexibility, but still supports the use of powerful techniques for the issues we seek to address. A "protocol" for conflict resolution is then developed based on this model. An example from the world of business - the interactions of a Marketing manager and a Finance manager negotiating a budget allocation - is used to illustrate the discussion.

2. Conflict resolution as problem-solving

The concept of a 'problem' has been variously defined as: 'conflict' [Duncker, 1945], an 'obstacle' [Maier, 1970], an 'accepted task that a person does not know how to carry out' [Simon, 1976], 'dissatisfaction with a purposeful state' [Ackoff and Emery, 1972], 'the difference between what one wants and what one has' [de Bono, 1970], etc.

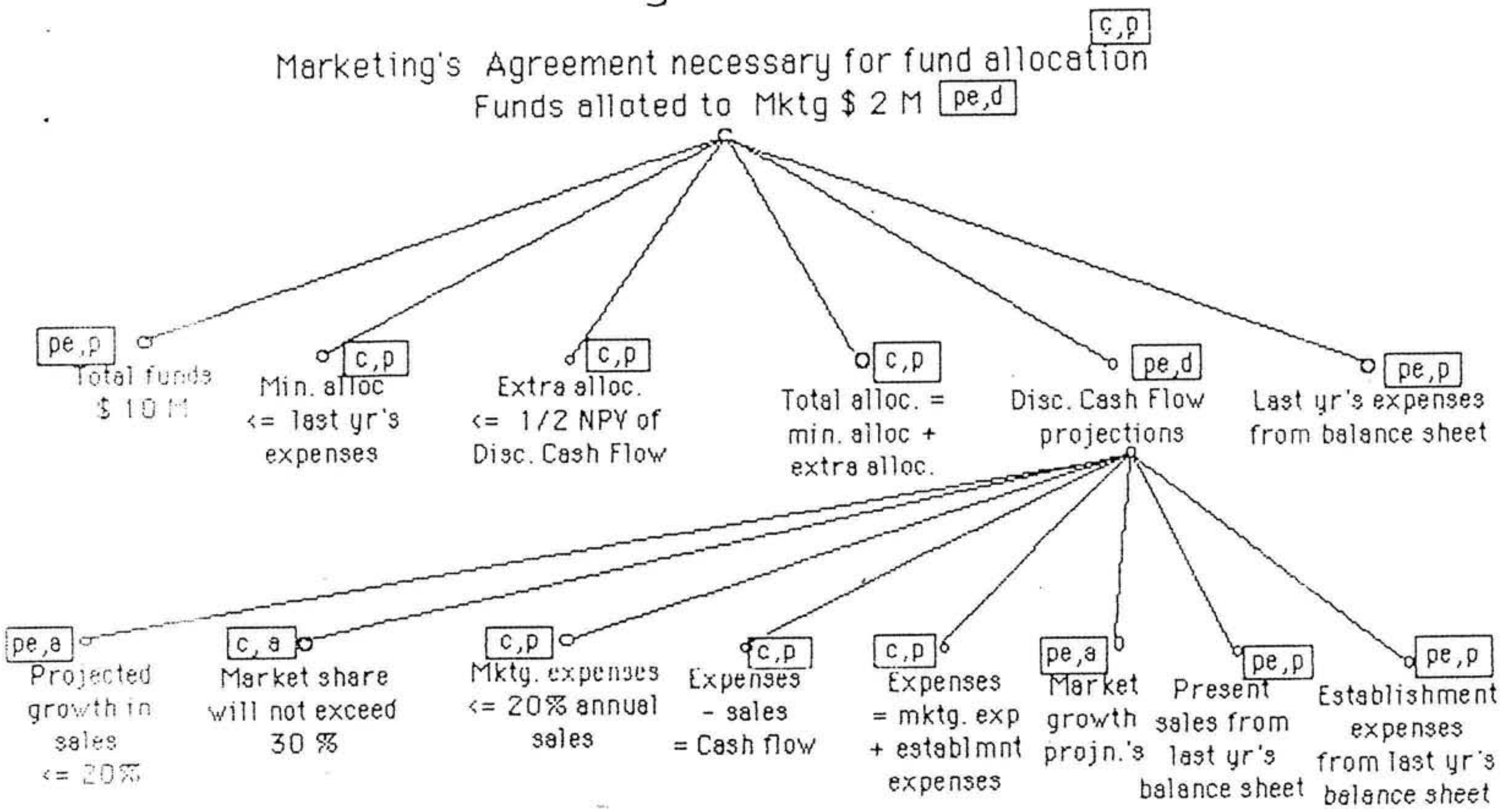
'Problem-solving' is the set of actions that an agent wilfully undertakes in response to a problem, with the express intention of mitigating the effects of the problem, or reaching a desired end-state. As an area of research interest, problem-solving has been extensively studied by economists, mathematicians, psychologists, and researchers in the field of Artificial Intelligence (**AI**).

AI researchers take three perspectives on problem-solving: one views problem-solving as a process of 'search' for a solution in the problem's solution-space. Another sees 'problem-reduction', or the decomposition of a problem into sub-problems whose solutions are known or trivial, as the key to problem-solving. A third perspective takes the view that problem-solving is the process of formulation, of 'constraints', their application, propogation and reformulation or adjustment, in the solution-space that bounds the problem.

In this paper, we adopt the third perspective, viewing problem-solving as *constraint propogation and reformulation.* We advance two reasons in support of this choice: Firstly, the use of constraints represents "negative reasoning" or reasoning by elimination. This allows us to partially specify a problem and incrementally modify it by adjusting or adding new constraints. Such an approach is best suited for dealing with ill-defined or *ill-structured* problems, where new information uncovered in the course of problem-solving significantly affects the problem-space. Conflict resolution is a typical ill-structured problem; the problem-space "evolves" constantly as information about each agent's position becomes known. Secondly, a whole body of powerful techniques is available for use with constraint-based problem formulations. Techniques such as "propogation analysis" and "dependency analysis" [Steele and Sussman, 1978] facilitate reasoning with constraint networks and provide efficient means to find problem solutions.

We take the approach that a 'problem-space' at any given instant, may be defined in terms of the "state" of 'problem entities' and 'constraints'. Any object (physical or abstract) that is of relevance to the problem-solving process is considered an **entity**. Entity states are determined by the values that are assigned to its 'attributes'. A **constraint** expresses a relationship that must obtain between attributes of one or more entities. Compound constraints or **constraint networks** may be built for modeling complex real-world systems, by combining simple constraints. A **hierarchy** of constraint networks may be used to model how any given problem is formulated. *Figure 1* presents an illustration of this idea for the example that we will use throughout this paper.

We spend a moment here to set forth the notation used in the figure. The problem-space is represented as a hierarchy of nodes. There are two kinds of "nodes": nodes representing *problem entity states*, and nodes representing *constraints* defined over problem entity states. Each node has a 'descriptor' that consists of two parts: the 'classifier', and the 'justification-class'. The classifier indicates whether a node represents a problem entity state or a constraint. The justification-class part of the descriptor specifies the origin of this problem "fragment," whether it was assumed, derived etc. (a more detailed discussion is offered in section 4.1). Each sub-tree represents a part of the problem-space; different "levels" in a sub-tree define 'problem-subspaces' since they consist of a collection of problem



Legend: The boxed "descriptor" [x,y] attached to a belief indicates the foll:

X : "classifier"
- c - constraint
- pe - problem entity state/value
- bc - believed constraint of opponent
- bpe - believed problem entity state of opponent.

Y: "justification-class"
- a - assumed belief
- d - derived belief
- p - premise
- df - default

Fig. 1 Constraint-based problem formulation : an illustration of an agent's belief-space

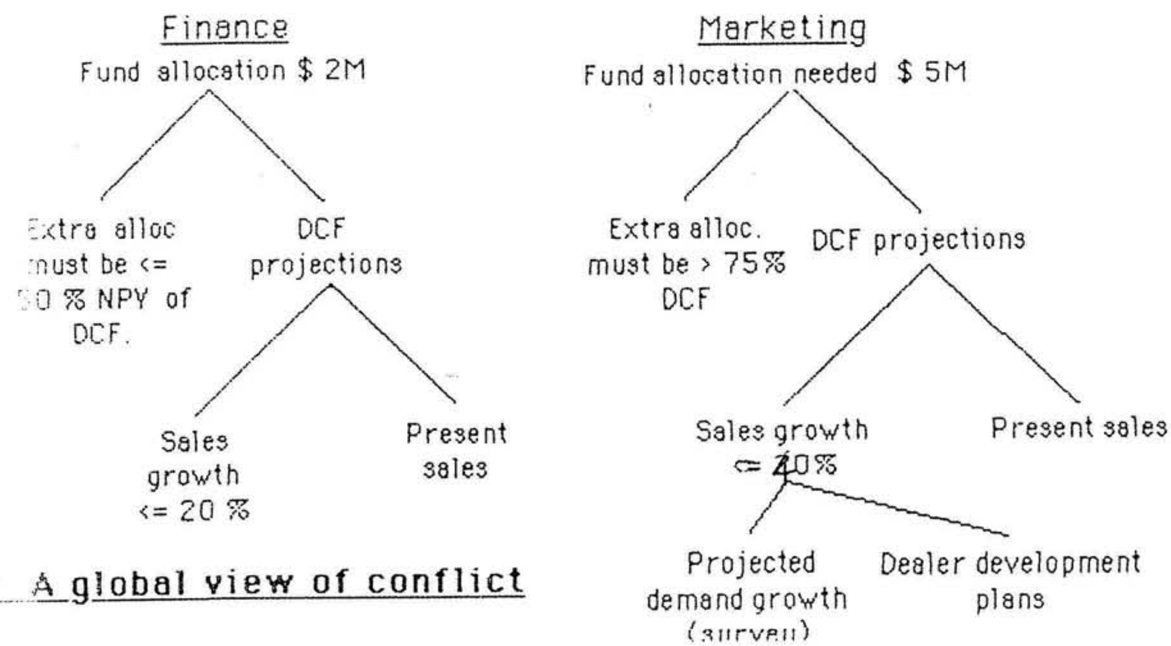


Fig. 2 A global view of conflict

entity states, and constraints defined over these problem entity states.

The example in *Figure 1* represents the formulation of the funds allocation problem, from the point of view of the Finance manager. At the root of the hierarchy is the constraint that expresses the 'problem statement'. Each level of every sub-tree in the hierarchy, represents a 'problem-subspace'. The two types of nodes represent the different aspects of the problem-space. For example, *Last yr's expenses from balance sheet* represents a problem entity state, while *Min.alloc <= last yr's expenses* is a constraint. The network of constraints and problem entity states at a particular level in a sub-tree, determine the 'state' of a problem entity at the next higher level, and so on. *Disc. Cash Flow projections*, for instance, is determined by applying the discounted cash flow model (represented by the network of constraints), to *present sales, projected growth in sales, market growth projn.'s*, etc., at the level immediately below it.

An "infeasibility" exists when the network of constraints at any level overconstrains the solution-space, or is unsatisfiable. Problem-solving then proceeds through a series of manipulations, that transform this overconstrained state into a satisfiable and feasible one.

When agents with disparate interests work together, each of them has his own formulation of the problem, though they also have a shared problem-space. This "shared" problem-space contains problem entities and constraints that are of concern to both agents. In the example (*Figure 2*), both the Finance manager and the Marketing manager have a hierarchy of constraint networks representing their own formulations of the funds allocation problem. However, they also share a problem-space since they are both concerned with the problem entities *Funds allotted to Mktg*, *Funds allocation needed*, and the constraint *Agreement needed.*

'Conflict' is the term used to describe the state when the shared solution-space is overconstrained. For example, since the Finance manager allocates $2M to marketing, which "needs" $5M, the constraint that the two must agree cannot be satisfied - there is a conflict. If one takes a "global view" of conflict by combining the two problem formulations, this infeasibility at the highest level of the problem-space arises because problem-spaces at some lower level, when taken together, overconstrain the problem. *Conflict resolution* therefore, is the series of adjustments and additions that are made, typically to both problem formulations, to make them jointly satisfiable.

To summarize, we have developed a model for an agent's problem formulation in terms of a hierarchy of constraint networks. Each level of the hierarchy represents a problem-subspace. The problem-subspace at one level determines the "state" of a problem entity at the next higher level. An overconstrained problem-subspace results in an infeasibility, or the inability to find a solution. When two agents with differing "interests" are interdependent, they share a part of their problem-space's - typically the "top-level" of their respective problem formulations. Conflict occurs when the problem entities and constraints in this problem-subspace represent an infeasible state. Conflict resolution is accomplished by transformations that make the two agents' problem formulations jointly satisfiable.

3. A belief-theoretic model of conflict

Based on his study of problem-solving protocols, Schoenfeld (1983) argues that there are three qualitatively different catagories of knowledge required for characterizing problem-solving: (1) 'control knowledge' which governs selection of goals and their pursuance or abandoning, (2) 'resources' - facts, algorithms and other knowledge that are possessed by the agent and may be brought to bear on the problem at hand and, (3) 'beliefs' that the agent may have about itself, about the problem and about the relevant environment. From his findings, he draws the conclusion that "one's beliefs establish the context within which one (a) selects from among one's resources and (b) (determines how to) employ them."

The role of *beliefs* as determinants of problem-solving context and consequently of problem-solving activity, has been studied by AI researchers studying non-monotonic reasoning and inference [Doyle, 1979; McCarthy, 1980; McAllester, 1982; and others]. A **belief** is a statement concerning the perceived truth of some proposition of interest. It may be believed and be 'in', or it may lose support and be 'out' (a third value of 'unknown' is also sometimes used [McAllester, 1978]).

Any agent engaging in problem-solving may be said to possess a set of beliefs about the various problem "fragments," or parts of the problem-space. In a constraint-based problem formulation, these beliefs concern problem entity states and constraints. The 'belief-space' of the agent is defined as the collection of all beliefs that are 'in', and those that are 'out'. This space undergoes modifications and additions, during the course of problem-solving. The set of beliefs that are 'in' at any point in time, constitute the agent's 'view' of the world, or 'problem formulation' at that point (*Figure 1*). Seen in this perspective, problem reformulation is equivalent to making transformations to the belief-space of the agent.

Modeling the agent's constraint-based problem formulation as a belief-space yields certain advantages. It is more 'flexible': a constraint can be thought of as a 'belief about the relationship among problem entities'. In addition, beliefs other than constraints may also be accommodated.

Second, it is more 'powerful'. By "powerful," we refer to two key factors: better information retention (or problem-solving "memory"), and "versatility" of problem-solving. The first derives from the principles of non-monotonic reasoning. Information "surfaced" during problem-solving, and inferences that are made from it are never lost. When the problem is reformulated, beliefs that are modified just lose support and become 'out'; if the context changes, they can be easily restored and the effect of this restoration is automatically propogated through the belief-space. The second argument for "powerfulness" is based on the strengths of currently available implementations of belief-maintenance systems such as the Assumption-based Truth Maintenance System (ATMS) [de Kleer, 1986a, 1986b] or one of the other variants of the Truth Maintenance System [Doyle, 1979]. With the ATMS implementation it is possible to attempt solutions in 'multiple world scenarios'. "Problem-solving can be restricted to a current context or all states can be explored simultaneously." [de Kleer, 1986a].

In the rest of this section, we develop a belief-space version of the constraint-based problem formulation, and a model for an individual agent's conceptualization of conflict. We will then use this model as a basis for proposing a protocol for conflict resolution.

Typically, an agent is not aware of the exact formulation that the opponent has given the problem. Conflict between an agent and its opponent is an overconstrained state of the shared problem-space, resulting from their respective problem formulations. Resolution of conflict requires reasoning with both problem formulations, and making adjustments so that they are jointly satisfiable. For an individual agent to initiate strategies for conflict resolution, it must model the problem as formulated by itself, and its opponent.

An agent may do this by articulating its 'beliefs' about the opponent, his problem formulation and, how they affect its (the agent's) own problem-solving status. A "subjective view" of the combined problem formulations, and the resulting shared problem-space, can therefore be constructed. The agent models this 'conceptualization of conflict' internally in its belief-space, in the form of a "tree" of beliefs. In a situation of dyadic conflict, the root of the tree has two children; one sub-tree of the root represents the agent's own problem formulation, the other represents its beliefs about the opponent's formulation of the problem. (Note: for notational purposes, we "classify" an agent's beliefs about its opponent's problem formulation in terms of "believed problem entity states" and "believed constraints"). Corresponding problem-subspaces from identical parts of the two sub-trees are referred to as 'sister' problem-spaces. At any point in time, by combining 'sister' problem-spaces from the two sub-trees, the agent can determine if conflict exists at that level of abstraction. The example in *Figure 3* shows the Finance manager's "internal" conceptualization of conflict. On the left, his own problem formulation is represented; the sub-tree on the right, shows his belief that the Marketing manager has assumed an extra allocation (for this year) equivalent to the NPV of discounted cash flows from projects. A comparison of 'sister' problem-spaces reveals to him that it is this assumption of his "opponent," that resulted in the demand for a $5M funds allocation, and caused the conflict.

An agent in a situation of conflict, attempts to resolve it typically through a combination of two strategies: (1) modifying its own problem-space by adjusting constraints and/or revising problem entity states, (2) trying to make the opponent alter his problem formulation. The generation of strategies for conflict resolution is the subject of the next section. Other than deliberate manipulation through such strategies, changes to the problem-space (belief-space) are also initiated in response to information uncovered in the course of problem-solving.

To summarize, in this section we built on the constraint-based view of conflict of the last section, and developed a model for an individual agent's "internal" conceptualization of conflict. We argued that this belief-space model provided 'flexibility' and 'power'. In the next section, we will develop a protocol that may be used by an intelligent agent, to generate strategies for resolving conflict with other agents.

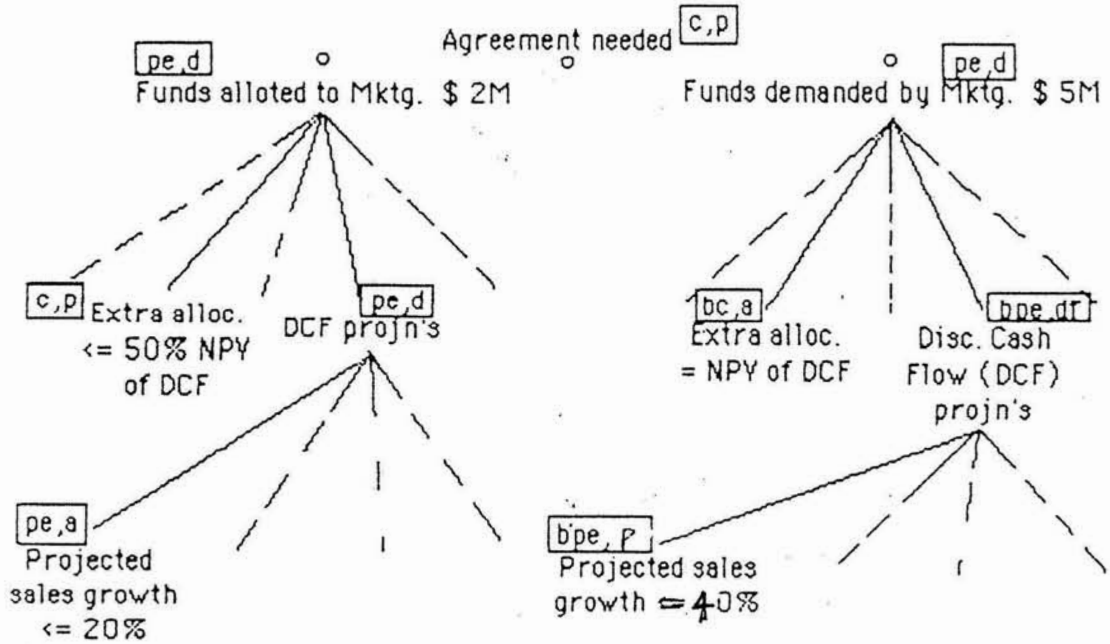


Fig. 3 Agent's conceptualization of conflict

Note: 1. For legend refer to Fig. 1
2. Only conflicting beliefs have been shown. "----" indicates default or non-conflicting beliefs
3. This figure should be seen in conjunction with Fig. 2

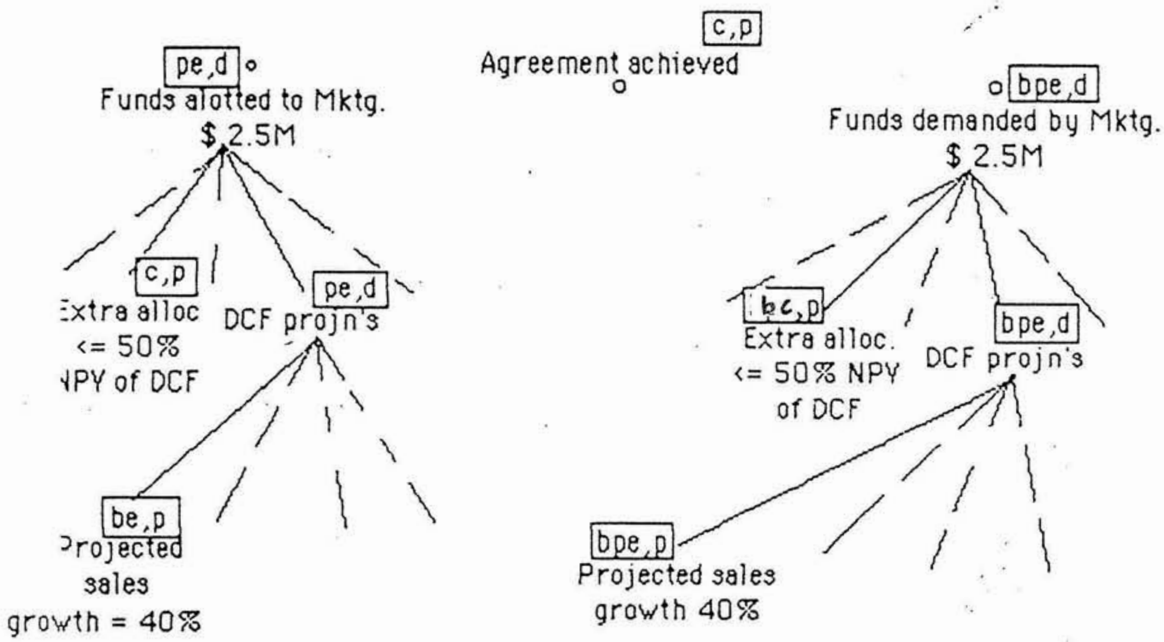


Fig. 4 The belief state of the agent upon conflict resolution

4. A protocol for conflict resolution

In order to develop a protocol for an agent to initiate conflict resolution, we need to know the exact nature of its internal conceptualization of conflict, how it is formed and how it might be changed.

4.1. Representation

Our basic representational unit is a 'belief'. One of the dimensions along which a belief may be characterized is in terms of how it originated - its 'justification-class'. It may be (1) a *premise* - something that is "taken for granted," (2) an *assumed belief* - an belief that is not necessarily grounded in fact, or (3) a *derived belief* - a conclusion that is derived from other beliefs.

A second dimension for characterizing beliefs is the aspect of reality they model. In the context of the constraint-based problem formulation, we have already identified two 'classes' of beliefs: beliefs about *problem entity states* or values, and beliefs about *constraints* that express relationship among problem entities. A third class of beliefs about *environmental factors*, or entities external to the problem, may also be represented.

Each belief in the belief-space of an agent engaged in problem-solving, may be characterized in terms of its classifier, and justification-class. In *Figure 1* for example, *Projected growth in sales <= 20%* is an assumption about a problem entity state; *Expenses - sales = Cash flow* is a premise about a constraint, etc. By examining the hierarchy of beliefs, one can perform a problem or situational analysis - by tracing "down" the justification of derived beliefs, the logic of the problem formulation becomes evident.

In the last section, we developed a model to understand how an individual agent could conceptualize conflict. We demonstrated that the agent may construct a "subjective view" of the combined problem formulations and the shared problem-space, by articulating beliefs about its opponent. The first step in developing a protocol for conflict resolution, is to address the questions: *How can the agent develop this internal conceptualization of conflict? How should it be represented?*

Our approach is based on the assumption that unless there is information to the contrary (past experiences, or information uncovered during an "exchange"), every intelligent agent believes an opponent to think and plan as he does. Therefore, *as the default,* an agent models its opponent's problem formulation as identical to its own. From a representational viewpoint, the agent creates a parallel, identical hierarchy of beliefs. Each belief in this new hierarchy is assigned the justification-class "default" to indicate its special origin. Typically, however, the agent does have some additional knowledge about the opponent; the belief-space is appropriately modified to reflect this knowledge. In *Figure 3*, for example, the Finance manager conceptualizes the problem formulation of the Marketing manager, as primarily identical to his own (note the "df" justification-class, representing that these beliefs are defaults), except for the belief about the constraint *Extra alloc. = NPV of DCF.*

Beliefs are represented using a modified form of the representation proposed by de Kleer (1986a). A belief is a 5-tuple:

<proposition, descriptor, label, justification, belief-state>

The *proposition*, is the logical expression whose "truth-state" is being represented. It consists of logical 'terms' (which take values true or false), linked by logical 'connectives' ("and," "or," and "not"). Each "term" consists of a combination of problem entity "attributes," and constants, connected by arithmetic and logical operators (less than, greater than, etc.). A special case is when the proposition consists of only one term and/or a term consists of only one element - beliefs about problem entity states are typically of this form.

The *descriptor*, the reader will recall, has two parts: <classifier, justification-class>. The 'classifier' indicates whether the belief concerns a constraint (*c*), a problem entity state (*pe*), a believed constraint of the opponent (*bc*), a believed problem entity state of the opponent (*bpe*), or an environmental factor (*ef*). The 'justification-class' specifies whether the belief is a premise (*p*), an assumption (*a*), derived from other beliefs (*d*), or a default assumed for an opponent's belief (*df*). This descriptor is a powerful representational feature that allows us to discriminate between the different beliefs in the agent's belief-space. Its importance for our work will become evident in the sections that follow.

The other 3 parts of a belief tuple are inherited from de Kleer's (1986a) representation. The *label* is a list of the (different) sets of assumptions under which the belief is 'in'. For a derived belief, this will include references to beliefs at the lower-level problem-subspace, from which it was derived, or the beliefs from which they were derived. The *justification* explains how the belief is derivable from other beliefs. It has three parts: the 'consequent' or the belief node itself, 'antecedents' from which the belief may be directly inferred, and the 'informant' or textual explanation of the justification. The *belief-state* parameter indicates whether the belief has support (is 'in'), or has lost support (is 'out').

4.2. Types of conflict, and their resolution

The second step in developing strategies for conflict resolution, is to ask: *How can conflict be detected? How may it be resolved?*

An agent may detect the existence of conflict between itself and the opponent, by introspectively "analyzing" its belief-space - combining 'sister' problem-spaces and examining them for infeasibility.

We identify two types of conflict that may occur in a problem-subspace: a *conflict of interests* and a *conflict of opinions*. When the 'constraints' formulated by the agent taken together with (its beliefs about) its opponent's constraints, overconstrain the problem-subspace, we call it a 'conflict of interests'. For example, in *Figure 3*, the two constraints *Extra alloc. <= 50% NPV of DCF*, and *Extra alloc. = NPV of DCF* produce an overconstrained state, and represent a conflict of interests. Such a set of constraints is basically unsatisfiable, whatever the state of problem entities - there is no feasible solution. When a conflict of interests exists, conflict resolution necessarily involves an adjustment of some of the constraints, so that they are jointly satisfiable. The issue of which constraint to adjust, and the process of such conflict resolution will be discussed shortly.

A 'conflict of opinion' occurs when the state assigned to a problem entity by the agent and (its belief about) the state assigned to the same entity by the opponent, differ. Since beliefs about problem entity states are often derived from problem-subspaces at a lower level, resolution of a conflict of opinion, may require consideration of corresponding 'sister' problem-spaces from the two sub-trees. Consider the belief *DCF projn.'s* in both sub-trees of *Figure 3*. If they differ, a conflict of opinion exists between the Finance and Marketing managers. Both these beliefs are 'derived' from a lower-level subspace. Upon considering the relevant 'sister' subspaces, it becomes apparent that the conflict of opinion, is a result of different beliefs about *Projected sales growth*.

If the "culprit" is not a derived belief as in the example above, but a belief that has been 'assumed', conflict may be resolved by analyzing, and possibly withdrawing support to the assumption.

If the agent possesses only 'default' beliefs concerning its opponent (it has no additional information), the only conflict it can detect is in the shared problem-space. Under such circumstances, the agent adopts an "inquiry mode." In this mode, the agent seeks corroboration of its beliefs about the opponent's problem formulation: it inquires about assumed beliefs first, and about derived beliefs in order of increasing "label size" (the objective is to minimize the number of beliefs examined). This inquiry proceeds until a conflict is observed or until all the beliefs in the problem-subspace are corroborated.

4.3. A protocol for conflict resolution

Determining answers to the questions: *What are the different strategies that may be used to resolve conflict? How can the agent determine which one to apply?*, is the third, and final step in developing a protocol for conflict resolution. In this section, we offer an algorithm or "protocol," that lays down the rules for how an agent may initiate strategies for resolving conflict, in "dyadic" situations.

Starting with the shared problem-space at the root of its 'internal conceptualization of conflict', the agent applies the following steps to each problem-subspace in its belief-space:

Step 1: If there are only defaults assumed for the opponent's problem formulation, the agent enters an 'inquiry mode' as explained earlier. It quits inquiry when conflict is "discovered," and passes on to the next step. If all the default beliefs are corroborated and no conflict is found, it skips Steps 2 & 3 and proceeds to Step 4.

Step 2: If "non-default" beliefs are held about the opponent, the agent first checks for a 'conflict of interests'. If such a conflict is found, the agent identifies the the subset of constraints that are inconsistent or "conflict." Some or all of the constraints in this 'conflict set' may be "adjusted" or modified to make them jointly satisfiable. A 'candidate' for adjustment is chosen based on a policy of "minimum modification to current context" (the beliefs that are currently 'in' define the context). This implies that a constraint is chosen such that its adjustment, may be accomplished with minimum modification of the belief-space. This is in keeping with the observed tendency of humans to resist change of belief-state.

The agent examines the 'conflict set' to identify an "assumed" constraint. If there is none, it tries to find a "derived" constraint with the smallest "label." If there are more than two "candidates" for adjustment, a choice is made using the following strategies:

1. If the candidates contain beliefs from the agent's own problem formulation, and (its beliefs about) the opponent's problem formulation, the agent chooses the latter for adjustment. This strategy is based on a "benefit of doubt" or "safeguard own interests" policy. The former acknowledges the possibility that the agent might be mistaken in its beliefs about its opponent, the latter is a reflection of a "better he adjusts than me" attitude.
2. If all the candidate beliefs are from the same sub-tree, a choice is made either on the basis of smaller label size, or at random.

Once a constraint has been chosen for modification, the agent switches to an 'interaction' mode where it either modifies a constraint and announces it to the other agent as an "offer," or tries to "convince" the opponent to change a constraint. The latter may be accomplished using one of the following *convincing strategies*:

1. Convincing by explanation or reasoned argument. It was mentioned earlier that the "logic" of a problem formulation can be obtained by "tracing down" the justification of derived beliefs. An argument for convincing the opponent could be made using such a trace.
2. Convincing by evidence - proving or counterproving by showing premises or 'true' beliefs as support.
3. Convincing by precedent - citing earlier instances if there exists a knowledge-base of such events, or by using "accepted norms."
4. Convincing by persuasion - by "threatening" or by "promising."
5. Convincing by gesture - the agent may be willing to "trade favors" by offering to adjust one of its own constraints for a "matching concession" by its opponent.

A detailed discussion of these 'convincing strategies' is beyond the scope of this paper, and is to be addressed in follow-up research.

If this step results in an adjustment of constraints, the belief-space is modified and the effects of revisions are propogated by using belief-maintenance mechanisms. Further conflict resolution continues by looping back to the beginning of Step 2.

Step 3: If no conflict of interests exists, the agent checks for a 'conflict of opinion'. A 'conflict set' consisting of beliefs that assign different "states" to the same problem entity, is identified if there is a conflict of opinion. The belief to be modified is chosen in much the same way as in Step 2, in accordance with the policy of "minimum modification to current context." An "assumed" belief may be modified by either making an 'offer' or by using convincing strategies. In order to modify a "derived" belief, the agent focusses attention on the constraints and problem entity states that the belief has been derived from. Conflict resolution now proceeds in this problem-subspace.

Step 4: If the agent finds "true agreement" (all beliefs have been corroborated, and there is no infeasibility) in the current problem-subspace, it "backtracks" to the problem-subspace at the level above and reexamines it for conflict. On the other hand, if there is no true agreement, the agent loops back to Step 1 and continues. Conflict resolution is complete when the agent backtracks all the way to the shared problem-space and finds no conflict.

We will now illustrate the use of this protocol in the context of our example. *Figure 3* illustrates the Finance manager's belief-space - his internal conceptualization of conflict with the Marketing manager, over funds allocation. The Finance manager must now initiate strategies aimed at resolving this conflict.

Upon examination of his own "subjective view" of the opponent's problem formulation, the Finance manager realizes that he does indeed possess some additional information (a "non-default" belief, in our terminology). He therefore proceeds directly to Step 2. The belief that Marketing manager has used *Extra alloc. = NPV of DCF* in his problem formulation, and the Finance manager's own belief that *Extra alloc. <= 50 % of NPV* give rise to a 'conflict of interests'. These two beliefs constitute the 'conflict set'. From the observation that the constraint *Extra alloc. = NPV of DCF* is an "assumption" he has made about the Marketing manager's formulation, the Finance manager decides to try and "modify" it. He now enters 'interaction mode' with the Marketing manager, and elects to use a strategy of "convincing by precedent" (because in his formulation, *Extra alloc. <= 50 % NPV of DCF* is a premise, based on say, last year's allocation).

Let us assume now that the Marketing manager is "convinced," and the constraint modified as shown in *Figure 4*. The Finance manager now proceeds to Step 4 since his "default" beliefs show no conflict of opinion. All the beliefs in this subspace have not been corroborated; he therefore returns to Step 1 and enters an "inquiry mode." In this mode, he "discovers" that beliefs about *DCF projn.'s* conflict.

Applying Step 3 (since this is a "conflict of opinion"), the Finance manager focusses his attention on the process by which these beliefs were "derived." In the lower-level problem-subspace shown in *Figure 3*, he finds a conflict of opinion between his belief that *Projected sales growth <= 20 %* and the Marketing manager's belief that *Projected sales growth = 40 %*. He seeks to modify the latter, and instead, is convinced by the arguments advanced by the Marketing manager, and revises his own belief as shown in *Figure 4*. In Step 4, the Finance manager now finds "true agreement" in this problem-subspace and backtracks to the higher level, and finally to the shared problem-space. Conflict resolution is now complete since there is no conflict in the revised belief-space.

5. Conclusion

In this paper we developed an argument for viewing conflict resolution as constraint-based problem-solving activity. The problem-space of an agent was modelled in terms of a hierarchy of problem-subspaces, in which problem entity states are bounded by constraints. A view of conflict as an overconstrained state of the "shared" problem-space, was presented. We then proposed a model of how an agent might conceptualize conflict internally, in his belief-space. Arguments were made that a belief-theoretic model offered advantages of "flexiblity" and "power." Using this modeling framework, we developed recommendations for a conflict resolution protocol that an intelligent agent may follow. Several alternative strategies were identified and prescriptions offered for their appropriate use.

As far as we are aware, this paper represents a first attempt to propose methodologies that may be used by competing, but interdependent, intelligent systems, to jointly solve problems where interests conflict. There are several issues that have been raised in this paper that are not addressed in adequate detail. Some of the major concerns that remain are: the operationalization of techniques for establishing minimal 'conflict sets' when there is infeasibility, development of detailed 'convincing strategies', and the implementation and evaluation of the protocol. Inspite of these shortcomings, we believe that this paper makes a contribution, by taking a first step towards beginning to understand the issues involved in automating conflict resolution between cooperating, intelligent agents.

BIBLIOGRAPHY

Ackoff, R.L. and F. Emery, *On Purposeful Systems*, Aldine-Atherton, Chicago, IL, 1972.

Cammarata, S., D. McArthur and R. Sleeb, "Strategies of cooperation in Distributed problem solving", IJCAI-83, Karlsruhe, WG, 1983, pp. 767-770.

Corkill, D.D. and V.R. Lesser, "The use of meta-level control for coordination in a distributed problem solving network, IJCAI-83, Karlsruhe, WG, 1983, pp. 748-756.

de Bono, E. *Lateral thinking: Creativity step by step*, Harper and Row, New York, NY, 1970.

de Kleer, J. "An assumption-based TMS", *Artificial Intelligence 28*, 1986, pp. 127-162.

de Kleer, J. "Problem solving with the ATMS", *Artificial Intelligence 28*, 1986, pp.197-224.

Doyle, J. "A truth maintenance system", *Artificial Intelligence 12*, 1979, pp. 231-272.

Duncker, K. "On problem solving", *Psychological monographs 58* (5), 1945, Whole No. 270.

Davis, R. and R.G. Smith, "Negotiation as a metaphor for distributed problem solving", *Artificial Intelligence 20*, 1983, pp. 63-109.

Ferrante, R.D., "The characteristic error approach to conflict resolution", IJCAI-85, 1985, pp. 331-334.

Maier, N.R.F. *Problem solving and creativity in individuals and groups*, Brookes/Cole Publishing Co., Belmont, CA, 1970.

McAllester, D. "A three-valued truth maintenance system", S.B. Thesis, Dept. of Electrical Engg., MIT, Cambridge, MA, 1978.

McAllester, D. "Reasoning utility package user's manual", Artificial Intelligence Laboratory, AI Memo 667, MIT, Cambridge, MA, 1982.

McCarthy, J. "Circumscription - a form of nonmonotonic reasoning", *Artificial Intelligence 13*, 1980, pp. 27-39.

Pruitt, D.G. and S.A. Lewis, "The psychology of integrative bargaining", in D. Druckman (ed.), *Negotiation: Social-psychological perspectives*, Sage Publications, Beverly Hills, CA, 1977.

Rosenschein, J.S, and M.R. Genesereth, "Deals among rational agents", IJCAI-85, 1985, Los Angeles, CA, pp. 91-99.

Schoenfeld, A.H. "Beyond the purely cognitive: belief systems, social cognitions, and metacognitions as driving forces in intellectual performance", *Cognitive Science 7*, 1983, pp. 329-363.

Simon, H.A. "Information processing theory of human problem solving", CIP Working paper No. 324, 1976.

Steele, G.L. Jr., and G.J. Sussman, "Constraints", Artificial Intelligence Laboratory, AI Memo-502, MIT, Cambridge, MA, 1978.

Walton, R.E. and R.B. McKersie, *A behavioral theory of labor negotiations*, McGraw-Hill, New York, NY, 1965.