December 2006

A Prescriptive Organizational Model for Transitional Negotiations

Tung Bui
University of Hawaii- Manoa

Siva Sankaran
California State University

A Prescriptive Organizational Model for Transitional Negotiations

Tung Bui
University of Hawaii, Manoa
tungb@hawaii.edu

Siva Sankaran
California State University
siva.sankaran@csun.edu

ABSTRACT

Customary organizational negotiations tend to focus efforts to attaining optimality in single-problem contexts that are both disparate and temporary in nature. Once the negotiations are settled, the process usually attains closure and the long-term impact of the outcome is rarely considered. In reality, however, decisions involving quid pro quos are made on a continuous basis. Since organizational environments are constantly in flux, negotiated solutions that appeared successful on a given problem at first no longer work out to be effective in the long run. We postulate that organizations evolve from one state to another and with it new negotiations are initiated on a continual basis, one negotiation transitioning into another. From this perspective, organizations can be modeled using sequential Markov chains that converge on homeostasis, leading to a prescriptive approach for transitional negotiations that suggest acceptance of short term losses in favor of the better payoffs that are to come.

Keywords

Negotiation support, computational modeling, cybernetics, organizational analysis.

INTRODUCTION

Organizations routinely make decisions that require consultations with multiple participants. Combining all points of view towards a consensus acceptable to all parties is always a challenge. A negotiation model imparting some degree of structure and transparency in organizational interactions and suggesting long-term choice policies in an otherwise irrational backdrop can be of valuable assistance to decision-makers.

Modern negotiation theory that finds its roots in decision theory and game theory focuses on interactive processes among antagonists with the attempt to reach compromises, or better yet, a win-win agreement (Raiffa,Richardson and Metcalfe, 2003). With the recent interests in developing Web-based tools to support various types of negotiation activities on the Internet (Ehtamo and Hamailainen, 2001; Kersten and Lo, 2003), the use of Negotiation Support Systems (NSS) tends to focus on ad-hoc problems (e.g., e-auctions) or local optimizations (e.g., labor contract renewal in an overall context of organizational decision-making).

In this paper, we look at negotiation in stochastic perspective. In each phase of the search for an organization action or policy, members of the decision teams negotiate to define the attributes that guide them in the choice process, the states of the organizations they want to reach, and the transitions and potential benefits associated with the selected action. At any given moment, an organization can be characterized as belonging to a discrete organizational state. During the transitory existence of the organization in such state, the decision makers endeavor to identify the state the organization is in and exercise one of the action-choices that are available to them in that state. Stated thus, the organizational flux can be described as consisting of a stream of theoretically infinite single-step state transitions in time, as the participants steer the organization through a series of decisions. The motivation of the decision-makers is to collectively choose actions that maximize benefit over time for their organizations.

ORGANIZATIONS AS COMPLEX SYSTEMS: COMPUTATIONAL THEORY BUILDING AND MODELING NEGOTIATION PROCESSES

With the advent of information technology and its use in organizations, scientists in organization theory have recently revised their 50-year old complex theory after a better understanding of the relation between the elements of organization design, decision making and performance (Anderson, 1999; Pines, Gowan and Meltzer, 1999). Other theorists explored the processes within organizations and their effects such as self-organization, bifurcation, and chaos (Dow and Earl, 1999). Another major direction of research leans toward building computational approaches to capture the dynamics of change and complexity, innovation and evolution, coordination and cooperation, organization learning and knowledge management (Carley, 2003). As the interest in computation increases, it is expected that new perspectives on organizations will emerge. We contend that the views that organizations are composed of intelligent, adaptive and computational agents in which learning and knowledge are distributed and where ecology of skills and social property come forward would be critical to designing effective negotiation processes (Bui, 2000; Epstein and Axtell, 1997; Hutchins, 1995).

From a stochastic modeling perspective, we can define the decision process to have four steps. The process is triggered by the discovery of organization members that the current state is not satisfactory and they engage in negotiating possible action strategies for future states. Second, they identify alternative action strategies available in each state and estimate inter-state transition probabilities. They then estimate the benefit when a certain action is taken, and finally, lay down a set of organizational choice-policies for each state that would maximize the overall benefits in the long run. The underlying idea is that what appears on the surface as random organizational behavior is most likely not random, but causal impacts of a series of external events and internal choices that can be modeled as probabilistic phenomena.

THE STOCHASTIC NEGOTIATION MODEL

States and Attributes

During a given time interval, an organization is assumed to be in any one of a large number of possible decision states. A state is represented by a specified collection of attributes and its values. A state can thus be designated by the notation, Z_i $[(a_1:v_1),\ldots,(a_n:v_n)]$, where i is the current state of the organization, a_1 thru a_n are the attribute names and v_1 thru v_n their corresponding values.

In a negotiation setting, each party has an agenda/ goals and proposes the attribute(s) each would like included in measuring a state. The outcome of this negotiation phase is the decision as to which attributes are to be included collectively in the model.

In our model, attribute values are in the range (0,1). Although infinite values are possible within this range, negotiators can limit their attention to specific discrete values of interest to them. If an agreement was reached to have, say r values uniformly distributed between 0 and 1, the possible attribute values can be calculated by the formula, $v_u = (u-1)/(r-1)$, where $u = 1,\ldots,r$. Thus, if there are n attributes to a state and each attribute can take r values, it can be concluded there will be r^n possible states among which the organization can transition.

To illustrate the concepts, let us consider an example. AgriHydro operates a dam that draws its income from selling water to farmers in its locality. The company is controlled by two board members, one representing the farmers and the other the shareholders. Both members want to collaborate in identifying the possible organizational states the company may face. In doing so, they identify two attributes that characterize each state (n=2). These are, a_1: level of water in the dam, a_2: level of demand for water from farmers. Further, they agree that each attribute can have three possible values (r=3) in the range [0,1)]. They are: 0 - low, 0.5 - medium and 1.0 - high. This leads to 9 (r^n) possible finite states that AgriHydro can be in. They can be expressed as, (a_1:0, a_2:0), (a_1:0, a_2:0.5), (a_1:0, a_2:1), (a_1:0.5, a_2:0), (a_1:0.5, a_2:0.5), (a_1:0, a_2:1), (a_1:1, a_2:0), (a_1:1, a_2:0.5), (a_1:1, a_2:1). Assuming the attributes are always denoted sequentially, we can further reduce the nomenclature to represent these states to (0,0,),…, (1,1) showing only the attribute values.

States and Choices

Decision-makers in the organization have any one of the possible action-choices, say c(1),…, c(m), that they may take for each state. Negotiators have to decide explicitly what these are. Taking no action can also be a choice. In a collaborative situation, the final actions-set will be the additive collection of the individual actions proposed by each participant. In an antagonistic negotiation context, this may need to be settled using some mutually acceptable value-based criteria. Turning to our example, two possible actions the board member for farmers may include are, (i) release water, and (ii) conserve water. On the other hand, the board member representing the shareholders may consider looking at the two possible actions of (i) raising the price for water and (ii) lowering the price of water.

State Transitions

The action chosen in a state moves the organization into a new state from among the possible states in the second time interval. The intervals need not be equal. Only one action is allowed per interval. The movement from one state to another is considered a probabilistic event. The probability that the current state will transition from i to j due to action choice c(k) is denoted by $q_{ij}^{c(k)}$. The new state j can be any of 1,…, n. The transition probabilities must satisfy the constraint, $\sum_j q_{ij}^{c(k)} = 1$ (see Table 1).

Current State	Choices	New State				
			Z_1	Z_2	…	Z_n
		Z_1,c(1)	$q_{11}^{c(1)}$	$q_{12}^{c(1)}$	…	$q_{1n}^{c(1)}$
		Z_1,c(2)	$q_{11}^{c(2)}$	$q_{12}^{c(2)}$	…	$q_{1n}^{c(2)}$)
		…	…	…	…	…
		Z_1,c(m)	$q_{11}^{c(m)}$	$q_{12}^{c(m)}$	…	$q_{1n}^{c(m)}$)
		Z_2,c(1)	$q_{21}^{c(1)}$	$q_{22}^{c(1)}$	…	$q_{2n}^{c(1)}$
		…	…	…	…	…
		…	…	…	…	…
		Z_n,c(m)	$q_{n1}^{c(m)}$	$q_{n2}^{c(m)}$	…	$q_{nn}^{c(m)}$

Number of states = $Z_{[1,\dots,n]}$

Number of choices in each state = c (1,… ,m)

Table 1. An example transition probability matrix

Determining probability for each cell is a multi-step process. Since each state consists of n attributes and r possible values, $q_{ij}^{c(k)}$ is computed using the formula,

$$q_{ij}^{c(k)}=(a_1{}^{(r)}{}_{ij}{}^{c(k)})*p(a_2{}^{(r)}{}_{ij}{}^{c(k)}))*..*p(a_n{}^{(r)}{}_{ij}{}^{c(k)}) \quad (1)$$

$q_{ij}^{c(k)}$ is to be computed for each n and r attribute combinations. The process must be repeated for each action choice c(k). The multiplicative form highlights the independence among the attribute values.

Even though the actions in each state affect the probabilities of transition from one state to another, it must be noted there may be external random events that can influence the outcome. Therefore, negotiators have the job of evaluating these events and adjusting the $q_{ij}^{c(k)}$ values while reaching a consensus. In addition to systematic elicitation of expert judgment, current and historical data may need incorporation. There are many algorithms in literature for accomplishing this step.

Let us return to our example. We saw that there are 9 states denoted by the coordinates (0,0),…,(1,1). Let the initial state be Z_1(0,0), and the terminal state be one of Z_j where j=1,…,9. Limit attention to choice c(1), conserve water. Using negotiation between the board members, assess the probabilities that conserving water will transition attribute a1 (level of water in the dam) from the current 0 (low) to 0.5 (medium) or 1.0 (high) states. We also need to adjust the probabilities to take into account random external events such as potential rains in the area. Let the following values be the result of such negotiation.

Initial State a1=0 ; Choice c(1) - Conserve water		
$p(a_1{}^{(r)}{}_{ij}{}^{c(k)})$		
Terminal State		
a1=0	a1=0.5	a1=1
0.6	0.3	0.1

Table 2. Assigning transition probabilities at the attribute level

We repeat the process for each of the attributes for the same choice. The transition probability is computed using formula (1). The $q_{ij}^{c(k)}$ values for the rest of the actions are determined using a similar approach.

Transition Benefit

Transition benefit is the income resulting from a transition during a period for implementing a specific choice. It can be either a tangible benefit such as a dollar income or it can be an intangible benefit such as a gain in utility for the negotiators, customer goodwill and the like. In the case of latter, external experts may need to be consulted in estimating the benefit value. As for measuring the transition benefit in terms of utility for the negotiators, several sophisticated techniques are available in literature. The simplest approach would be to take the average using the formula,

$$g_i^{\,c(k)} = \sum_e U^{Ne}{}_i^{c(k)} / N \quad (2)$$

where $g_i^{\,c(k)}$ is utility benefit in state i due to choice c(k), N is the number of negotiators, U^{Ne} is the utility value as estimated by the Ne^{th} negotiator.

Identification of a Choice Policy

Policy is a prescriptive function. Its purpose is to suggest which choice c(k) out of the possible set of choices must be acted upon, given the organization is in state Z_i. Such a policy can be mathematically stated as f(c|z) where $\sum_c$ f(c|z)=1. Even though states may be greater than n due to multiple attribute values, we adopt the conventional notation n in formulas 4-7.

If Z_n is the current state and action c(m) was adopted, the conditional probability of transition can be stated as,

$$P[(z', c') \mid Z_n, c(m)] = q\,(z'|z,c)^* f(c'|z') \quad (3)$$

z' represents the subsequent state and c' represents the action taken while in this state.

We assume the transition probability matrix to be Markov chain compliant. By the basic limit theorem of Markov chains, the long run mean income per unit interval g(f) would be equivalent to that under the stationary distribution.

$$g(f) = \lim_{n \to \infty} (1/n) \sum_{m=1}^{n} E[g(Zn, c(m))] = \sum_{z=1}^{n}\sum_{c=1}^{m} \pi(z,c) * g(z,c) \quad (4)$$

$$\text{where } \pi(z,c) \geq 0 \text{ and } \sum_{z=1}^{n}\sum_{c=1}^{m} \pi(z,c) = 1 \quad (5)$$

Using equations (4) and (5), we can generate (6) as follows,

$$\pi(z',c') = \sum_{z=1}^{n}\sum_{c=1}^{m} \pi(z,c) * q(z' \mid z,c) * f(c' \mid z') \quad (6)$$

From this, the policy f that maximizes the long run expected income per unit time can be calculated as shown in equation (7).

$$f(c'|z') = \pi(z',c') / \sum_{c=1}^{m} \pi(z',c) \quad (7)$$

Even if income levels were to fluctuate from one period to another over short periods, so long as the prescriptive policy is adopted, it will not adversely affect the asymptotic long range performance. Solutions to these equations can be generated using discrete stochastic dynamic programming formulations (Karlin and Taylor, 1984; Puterman, 1994).

From a cybernetic perspective, generating a choice policy is a learning process. The organization should continually examine the outcomes following from the choices it made in the previous periods, reinforce the assessments of the attribute elements and revise its battery of choices. Over an extended period, this results in the re-evaluation of the transition probability matrix and consequently leads to a new set of choice policies.

COMPUTATIONAL ILLUSTRATION OF THE MODEL

In this section, we shall complete the example we began earlier for the AgriHydro company. Since the purpose is to illustrate the model steps, we shall limit the problem further to two attributes, three states, and two actions per state. Note that we have two negotiators in the company, one representing the customers and the other company's owners.

Step 1: Negotiate the attributes and their value sets.

Let us assume that both participants collectively agreed to use (i) level of water in the dam, and (ii) level of demand for water from farmers as attributes. Each agrees to two possible values 0 and 1 for each attribute. This generates four possible states AgriHydro can be in: Z(0,0), Z(0,1), Z(1,0) and Z(1,1). The state (0,0) represents the state where the water level in the dam is low and the demand for water from the farmers is also low. The state (0,1) represents the state where the water level in the dam is low but the demand for water from the farmers is high. Similarly, the state (1,1) represents the state where both the water level in the dam and the demand for water from the farmers are high.

Step 2: For each state, negotiate possible choices, c(k), where k=1,…,m.

The resultant agreed choices will depend on how collaborative or antagonistic the negotiators were with respect to each choice. In the former case, the various choices would be collected additively into the basket of action strategies. In the latter case, give and take can be expected. The utility of having a choice in the basket for one negotiator at the expense of the other may be compensated by a reciprocating gesture with regard to other choices. Let us assume that in our case, two choices were agreed on: (i) release water, and (ii) conserve water.

Step 3: Evaluate the state transition probabilities.

We saw in Step 1 that AgriHydro can be in one of four states. In the current step, the task is to determine the probability that a given state will remain in status quo or move to another in the next time interval as a result of taking either the conserve or release-water actions. In our case, the structure of the transition probability matrix will look like below. (Note that in the interest of computational simplicity, the state (1,0) has been ignored. Further, this situation where the water availability is high but the demand is low is of little interest to decision-makers as its potential to generate income is trivial).

$q_{ij}^{c(k)}$ values for Table 3 can be calculated using formula (1) presented in the earlier discussion. For example, $q_{11}^{c(1)}$ is computed as shown below.

$$q_{11}^{c(1)}= p(a^{(1)}{}_{0,0}{}^{c(1)})*p(a^{(2)}{}_{0,0}{}^{c(1)}))$$

In words, this means that the probability of remaining in state 1 with water level value at zero and water demand at zero from one interval to the next when the action to release water has been taken is the product of the probability that the water level value will remain at zero despite the water release and the probability that the water demand will remain at zero despite the water release.

Starting State (i)	Attributes a1,a2 (Water level, Water demand)	Actions c(k)	Terminal state		
			1 (0,0)	2 (0,1)	3 (1,1)
1	(0,0)	k=1 Release	$q_{11}^{c(1)}$	$q_{12}^{c(1)}$	$q_{13}^{c(1)}$
		k=2 Conserve	$q_{11}^{c(2)}$	$q_{12}^{c(2)}$	$q_{13}^{c(2)}$
2	(0,1)	k=1 Release	$q_{21}^{c(1)}$	$q_{22}^{c(1)}$	$q_{23}^{c(1)}$
		k=2 Conserve	$q_{21}^{c(2)}$	$q_{22}^{c(2)}$	$q_{23}^{c(2)}$
3	(1,1)	k=1 Release	$q_{31}^{c(1)}$	$q_{32}^{c(1)}$	$q_{33}^{c(1)}$
		k=2	$q_{31}^{c(2)}$	$q_{32}^{c(2)}$	$q_{33}^{c(2)}$

Table 3. Structure of the transition probability matrix

The negotiators at this point will have to negotiate/decide on the values of $p(a^{(1)}{}_{0,0}{}^{c(1)})$ and $p(a^{(2)}{}_{0,0}{}^{c(1)})$. It is possible they would rely on historical data or seek assistance from experts in the field in completing this task. Let us assume that in our case $p(a^{(1)}{}_{0,0}{}^{c(1)}) = 0.75$ and $p(a^{(2)}{}_{0,0}{}^{c(1)})= 0.27$. Thus, $q_{11}^{c(1)}= 0.20$.

We repeat the above process for each transition cell in Table 3. The final transition probabilities are shown in Table 4.

Starting State (i)	Attributes a1,a2 (Water level, Water demand)	Actions c(k)	Terminal state		
			1 (0,0)	2 (0,1)	3 (1,1)
1	(0,0)	k=1 Release	0.2	0.6	0.2
		k=2 Conserve	0.4	0.3	0.3
2	(0,1)	k=1 Release	0.7	0.2	0.1
		k=2 Conserve	0.2	0.5	0.3
3	(1,1)	k=1 Release	0.1	0.7	0.2
		k=2 Conserve	0.8	0.1	0.1

Table 4. Final transition probability matrix for AgriHydro

Step 4: Determine the transition benefits.

Here, the participants have to assess the one period benefit function as a result of taking a specific action while in each of the possible states. As discussed earlier, the benefit may consist of either tangible or intangible incomes. The latter case is more complex. Formula 2 presented earlier can be used for this purpose. Let us assume the final agreed upon income matrix looks as shown below. Unless tangible values such as dollars are involved, incomes are represented using numbers relative to each other in size.

State	Action	
	Release water	Conserve water
(0,0)	1	2
(0,1)	5	3

Table 5. Benefit matrix for AgriHydro by state and action

Step 5: Apply a dynamic programming solution approach to determine the action-choice policy for each state.

The theory behind this step was presented in the earlier section. Formulas (4) thru (7) laid out the logic of the computations. Solving these equations require that they be restructured in linear programming format. A variation of the approach consists of solving the dual problem. The specific reiterative technique is referred to as the Howard's algorithm (Karlin and Taylor, 1984).

State Z_i i =	Choice Policy c(k)
1	Conserve water
2	Release water
3	Release water

Table 6. Long-term policy

The recommended long-run policy is shown in Table 6. These choices offer us the best chance that despite variations in benefits from one state to another over short term, in the long run sticking to the action policy will tend to maximize mean benefit of the inter-transitions.

ARCHITECTURE FOR A COMPUTERIZED SYSTEM

There are several benefits to implementing the above model as a computerized system. They are:

a. The computational requirements of dynamic programming can be handled easier
b. Negotiators who are dispersed geographically can participate in joint decisions over a network/Internet
c. Negotiations can take place asynchronously
d. Participants can maintain anonymity
e. Supporting experts in the field can join in discussions
f. External databases having historical data can be accessed
g. When negotiators want to perform modifications to inputs, the system can re-compute the results quickly leading to faster agreements

Another important advantage of having a computerized system is that the parties to a negotiation can try out different simulated scenarios for themselves. This will allow them to get better acquainted with the organizational context before they enter the real negotiation session. By simulating various portfolios of attributes, states, transitions and potential benefits in virtual sessions, they can familiarize themselves with the sensitivity of the different stochastic model parameters. It will also enable them to ascertain in advance which elements would matter most in the negotiation process and yield them insights into alternative stake out positions. They can build an understanding early on of what negotiation strategies might work for them and develop a plan of action accordingly before ever coming face-to-face with their counterparts. It could also give them an idea as to what the other parties might bring to the negotiation table and help them in preparing defensive/counter strategies ahead of time. This functionality of individual simulation using the computerized system is illustrated by the dotted rectangle in Figure 1.The figure also shows the architecture of the various system components and human negotiators/experts.

Attribute synthesizer

This module interacts with the negotiators and gathers the names of the attributes each participant considers as necessary to describe the organizational state. Once the attribute collection phase is completed from each human negotiator, the module identifies a common set of attributes acceptable to all parties. Only a single additive step is needed in a collaborative environment. When consensus is hard to reach, the module requests participants to rank the attributes and/or asks them to assign normalized weights to each. From this information, the module computes a starting set of attributes.

State composer

Once the attributes has been agreed upon, the next step is to compute all possible states for these attributes. The State composer component begins its work first by prompting the negotiators to input the number of intervals they would like to divide the attribute value range (0,1). Using a nested programming approach, the composer is able to enumerate every possible combination of these attribute values and coordinate them to capture the set of discrete states the organization can potentially be in. This gives a great advantage to the participants since they need not be limited by their inability to process a large number of organizational states on their own. However, since the number of states rises exponentially with the number of attributes and their intervals, the module gives the users the option to eliminate states that they believe are of little interest. In fact, in our AgriHydro example, such a situation was discussed whereby the negotiators decided to remove the state represented by the coordinates (Water level=1, Water demand=0).

Choices assembler

Having identified potential states the organization can be in, the next step is to synthesize what negotiators consider as action-choices applicable to each state. The Choices assembler module accomplishes this by presenting the individual states and receiving inputs from the users in the form of the different decisions they can make under each state. The module has access to the Knowledge manager component as well as internal and external databases which enable the system to benefit from decisions made in the historical past. External advice from consultants/ experts may also be incorporated. Thus, it can complement the actions directly suggested by the users through the system input interface. The Choice assembler has also the task to generate a consensual set of actions for each state if negotiators don't see eye to eye at the first round of bargaining.

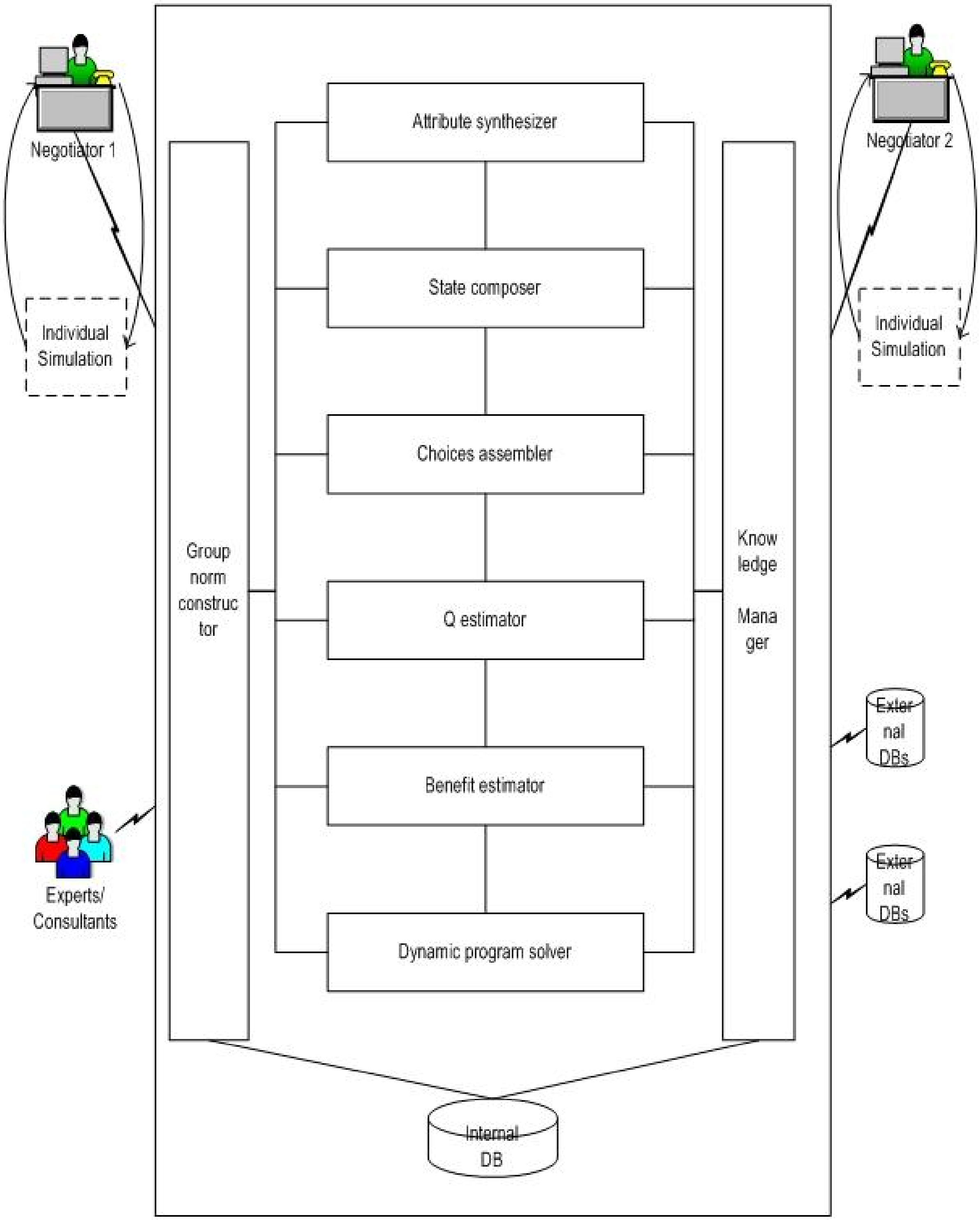

Figure 1. Components of the Stochastic Negotiation System

Q-estimator

The Q-estimator elicits from each negotiator the transition probabilities of each attribute moving from its current state into any of the other possible states that have been identified by the State composer module. The component asks the parties through its interface to input the estimates for each of the choices recognized by the Choices assembler. This component has also the traditional negotiation techniques built into it to assist the parties to converge to a common acceptable value should disagreements arise. It has also access to the Knowledge manager and the databases that contains data from previous periods. Further, the system can supply information from external consultants and experts to help the negotiators in generating the probability estimations. Once the probabilities of transition for each attribute have been determined, Q-estimator combines the information to derive the state transition probabilities using formula (1) and saves them to the system's internal database. Contradicting attributes will be assigned very near zero-values for transition probabilities to obviate their impact on final solution.

Benefit estimator

This module captures from the negotiators what they consider as the potential income during a period for every action while in a possible state. Once again, the component can also draw conclusions from system resources based on past data. Bayesian estimates can be used to refine data stored in the databases to improve the quality of prediction. Where negotiators diverge on their estimates, the system can help bridge the gap by facilitating iterative sessions.

Group norm constructor

This is a crucial module in that it controls the intercommunications among the rest of the components using a set of pre-drawn criteria mutually accepted by the negotiators. It facilitates exchange of information through use of blackboard memory as well as software agents to perform various services. One of the important functions it performs is to collect all the information generated by the state composer, choices assembler, q-estimator and benefit estimator modules and direct them to the dynamic program solver. When the group norms are violated, the appropriate agents are fired. Many of these agents are triggers and hence automatically get invoked when the occurrence of a designated event is detected by the system (Bui, 1987).

Knowledge manager

Organizations cumulatively perform an enormous number of transactions over long periods of time. Successful organizations store these information in data warehouses and detect patterns through data mining operations. These learnt patterns can be very helpful while assessing future organizational attributes, inter-state transition probabilities, what actions worked and didn't in the past and benefit matrices. The component also stores the expertise that decision-makers have learnt through their experience. They are typically represented in the form of if-then production rules using artificial intelligence techniques. Potential combinatorial explosion is greatly reduced using these rules. The knowledge manager has also access to both internal and external databases to complement its own data.

Dynamic program solver

This component has both linear programming formulation and solving abilities. Depending on the complexity of the problem at hand, the component can either solve the original linear programming equations or can reformulate them into a dual problem and then perform the simplex algorithm on it. When appropriate, it can also look for solutions using the trial and error approach as described by the Howard's algorithm. The module works seamlessly with the group norm constructor exchanging information back and forth. This feature is extremely useful when negotiators fiddle with their original inputs and want to assess the impact instantaneously. As mentioned earlier, when users want to perform simulation and sensitivity analysis either on an individual basis or in a real world negotiation context, dynamic program solver facilitates the generating of the ultimate solutions negotiators are looking for quickly and efficiently. The feasibility and reasonableness of solutions are assured through the currency and proper assignment of the transition probabilities by the Q-estimator module.

CONCLUSION

When organizations face consequences of their own past decisions as well as that of external random events, it is hard to predict where the organization will be at a future time. In this research, we assume that organizations progress from one state to another following a probabilistic transition pattern rather than in a totally random manner. By so hypothesizing, organizations can be modeled as using sequential Markov chains with a predisposition to achieve homeostasis. Understanding organizational behavior from this perspective can be greatly beneficial to negotiators by enabling them to accept short term losses in favor of the larger and better payoffs that may come to fruition in the distant future.

The model presented in this paper integrates techniques developed in negotiation support discipline with the benefits of depicting organizations as complex probabilistic systems. By representing an organization's evolution as conforming to stochastic processes, the proposed model obviates the assumptions of anarchy and irrationality ordinarily imposed to comprehend decision-making in such environments. Yet, it accounts for the randomness of the events in the organizational environment and guides decisions towards better states in the future for the organization to flourish. Lastly, far from being a mere theoretical tool, the model provides a practical approach to improving efficiency and effectiveness that directly benefits an organization's performance.

REFERENCES

1. Anderson, P. (1999) Complexity Theory and Organization Science, *Organization Science*, 10, 216-232.
2. Bui T. (1987) Co-oP – A Multiple Criteria Group Decision Support System, Springer Verlag.
3. Bui, T. (2000) Building Agent-based Corporate Information Systems, *European Journal of Operations Research*, 122, 242-257.
4. Carley, K. (2003) *Intra-organizational Computation and Complexity*, Working paper, Carnegie-Mellon University, April.
5. Dow, S. and Earl, P. (Eds.) (1999) Contingency, Complexity and the Theory of the Firm, Edward Elgar.
6. Ehtamo, H. and Hamailainen R. (2001) Interactive Multiple-Criteria Methods for Reaching Pareto Optimal Agreements in Negotiations, *Group Decision and Negotiation.*
7. Epstein, J. and Axtell, R. (1997) Growing Artificial Societies, MIT Press.
8. Hutchins, E. (1995) Cognition in the Wild, MIT Press.
9. Karlin, S. and Taylor, H.M. (1984) An Introduction to Stochastic Modeling, Academic Press, New York.
10. Kersten, G. and Lo G. (2003) Negotiation Support Systems and Software Agents in E-business Negotiations, *The First International Journal of Internet and Enterprise Management*, 1 (3).
11. Pines, D., Cowan, G., and Meltzer, D. (eds.) (1999) Complexity: Metaphors, Models and Reality, Addison-Wesley, Santa Fe Institute Series.
12. Puterman, M.L. (1994) Markov Decision Processes: Discrete Stochastic Dynamic Programming, John Wiley, New York.
13. Raiffa, H., Richardson J., and Metcalfe D. (2003) Negotiation Analysis: The Science and Art of Collaborative Decision Making, Belknap Press of Harvard University.