Brain and Mind **4:** 239–252, 2003.

Cognitive-Decision-Making Issues for Software Agents

BEHROUZ HOMAYOUN FAR[1] and ROMI SATRIA WAHONO[2]
[1]*Faculty of Engineering, University of Calgary, 2500 University Drive N.W., Calgary, Alberta, Canada, T2N 1N4, e-mail: far@enel.ucalgary.ca*
[2]*Faculty of Engineering, Saitama University, Saitama, Japan, e-mail: romi@romisatriawahono.net*

(Received: 29 January, 2003; in final form: 23 April, 2003)

Abstract. Rational decision making depends on what one believes, what one desires, and what one knows. In conventional decision models, beliefs are represented by probabilities and desires are represented by utilities. Software agents are knowledgeable entities capable of managing their own set of beliefs and desires, and they can decide upon the next operation to execute autonomously. They are also interactive entities capable of filtering communications and managing dialogues. Knowledgeability includes representing knowledge about the external world, reasoning with it, and sharing it. Interactions include negotiations to perform tasks in cooperative, coordinative, and competitive ways. In this paper we focus on decision-making mechanisms for agent-based systems on the basis of agent interaction. We identify possible interaction scenarios and define mechanisms for decision making in uncertain environments. It is believed that software agents will become the underlying technology that offers the capability of distribution of competence, control, and information for the next generation of ubiquitous, distributed, and heterogeneous information systems.

Key words: software agents, decision making, uncertainty and risk management.

1. Introduction

The rapid development in computer technologies has made it impossible to continue using the centralized, monolithic programming model that was adequate when treating computers as isolated entities. Presently, interconnected computer systems are normally distributed over a wide area leading to distribution of competence, control, and information. Therefore, for computing to become truly ubiquitous, new distributed, multitask programming methodologies must be developed. It is believed that distributed, multiagent technologies offer the capabilities needed. Consequently, there are many projects focusing on multiagent systems.

Traditional software systems can handle data and information. Data is defined as a sequence of quantified or quantifiable symbols. Information is about taking data and putting it into a meaningful pattern. Knowledge is the ability to use that information. Knowledgeability includes representing knowledge about the external world, reasoning with it, and sharing it. Interactions include the ability to directly communicate or collect data on the other agents. Software agents are knowledgeable entities capable of managing their own set of beliefs and desires and they can decide upon the next operations to execute autonomously. They are also interactive entities capable of filtering communications and managing dialogues. Various techniques and methodologies to handle the knowledgeability and interactivity have already been introduced (Jennings 1997, Weiss 1999, Subrahmanian 2000).

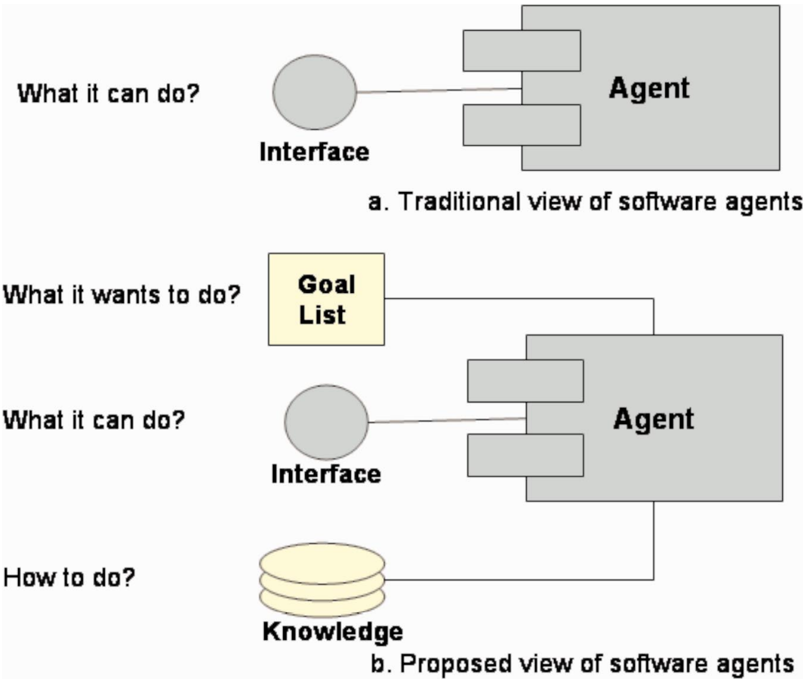


Figure 1. A view of software agent attributes.

When modeling software agents, a popular view is to model them as software components, i.e., a package with complete encapsulation of its behavior that has only one attribute called *interface* (i.e., what they can do). Then the component can only be accessed through its interface (Figure 1(a)). In this case, the other agents requiring services of a certain software agent may consult directory and naming services (i.e., agent yellow pages) and use its service by adhering to the strict rules specified in the interface document. Experience shows that this limits the scope and applicability of the software agents, in the sense that autonomy and interactivity may be compromised.

We propose another view in which two more attributes are also specified: a *goal list* (i.e., what they want to do) and *knowledge* (i.e., how to do) that an agent can utilize to perform tasks autonomously (Figure 1(b)).[3] When interacting with the other agents, either of the goals, interface, and knowledge attributes can be declared *public* or *private*. *Public* means that the attribute is accessible and readable by the other agents and *private* indicates otherwise. This leads to a maximum of eight interaction scenarios. On the basis of this view, a subset of useful and popular agent interaction scenarios are identified (Onjo and Far, 2001).

[3]Note that we may also add two more attributes: the thread of control (when) and identity (who). We do not consider them in this paper.

- *Cooperation:* Cooperation is revealing the agent goal and the knowledge to the other party, i.e., goal and knowledge are both public. In cooperation both agents share common goals.
- *Coordination:* Coordination is revealing the agent goals and the knowledge to the other party, i.e., goal and knowledge are both public. In coordination, agents have separate goals.
- *Loose competition:* Loose competition is revealing only the agent goals but masking the knowledge from the other party, i.e., goals are public and knowledge is private.
- *Strict competition:* Strict competition is revealing neither an agent's goals nor the knowledge to the other party, i.e., both goals and knowledge are private.

Using this view the agent autonomy can be preserved if a proper decision-making mechanism for the agents is devised and implemented. That is, the agent can decide upon the next task to perform using the current list of goals, interfaces, and knowledge of self and the other agents interacting with.

2. Decision-Making Techniques for Multiagent Systems

Agents engaged in *cooperative* and *coordinative* tasks can potentially have precise information about the other agents goals due to the fact that the goals and knowledge are accessible, usually through direct communication. Many techniques and methods to handle cooperation and coordination have already been proposed (Huhns and Singh, 1998). In this paper, we consider agent decision making in competitive and uncertain environment which normally arises in the case of competition. In this case, the agent must predict the other agents goals and this introduces uncertainty to the decision-making model. There are many sources of uncertainty including

- uncertainty inherent in the problem domain being modeled, and
- uncertainty about the outcome of decisions (i.e., risk).

There are a number of decision-making techniques based on uncertain data. A modest list may include the following: uncertainty management in knowledge-based systems using certainty factor (Shortliffe, 1979); Bayesian belief networks (BBN) and dynamic BBN (Pearl 1988, 1990); game theory, including static games of complete information; static games of incomplete information (static Bayesian game); dynamic games of complete and perfect/imperfect information; dynamic games of incomplete information (Gibbons, 1992; Osborne, 1994; Kuhn, 1997; Bierman, 1998); decision-making models under uncertainty and risk (French 1986); Dempster–Shafer's Theory of Evidence (Dempster, 1967, Shafer, 1976); Ordered Weighted Averaging (OWA) (Yager, 1988; 1990); and uncertain programming techniques (linear programming, nonlinear programming, multiobjective programming, goal programming, integer programming, multilevel programming, dynamic programming, expected value models, chance-constrained programming, dependant-chance

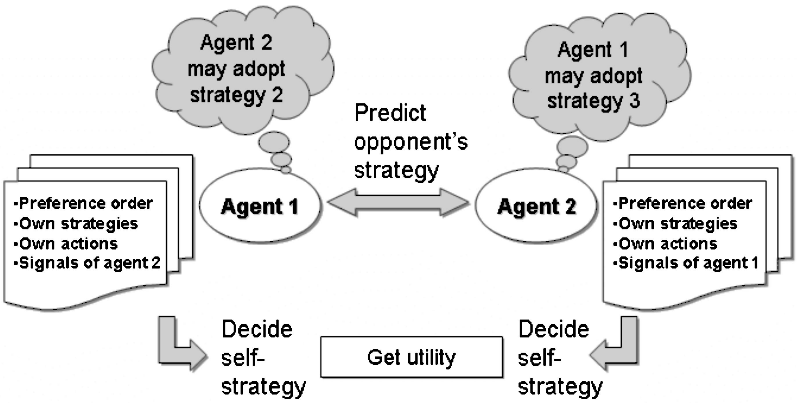


Figure 2. Overview of a competitive decision-making scenario.

programming) (Liu, 2002). Each of the techniques is appropriate for a class of agent interaction scenarios. Below we introduce the appropriate techniques for agent competition.

2.1. OVERVIEW OF MULTIAGENT COMPETITIVE ENVIRONMENT

Figure 2 shows the outline of agent competition. The process for deciding competitive strategy includes the following steps. First, each agent tries to predict opponent's strategy. Second, the agent chooses the best response strategy based on predictions. And finally, each agent will get a payoff, using a utility function.

From the decision-making viewpoint, since the amount of payoff in extended games is influenced by the opponent's moves, predictions of the other agent's strategies is crucial to guarantee a stable payoff. Information about opponent's moves is obviously uncertain. The law of maximum expected utility, i.e., selecting actions that yield the most preferred outcomes, given the actions of the opponent (French, 1986) will govern the decision for each agent.

2.2. MODELING COMPETITIVE ENVIRONMENT

Rational decision making depends on what one *believes*, what one *wants* (desires), and what one *knows*. In decision models, beliefs are represented by *probabilities* and desires are represented by *utilities*. Moreover, one needs a set of strategies to select from. Two cases may arise:

- *Opponent's goal (preference relation) known:* The problem can be reduced to static/dynamic game with perfect/imperfect information.
- *Opponent's goal (preference relation) unknown:* The competition scenario can be represented by extensive form of a game as shown in Figure 3. This is a simple but illustrative example of agents' competition model (*agent_1* versus *agent_2*).

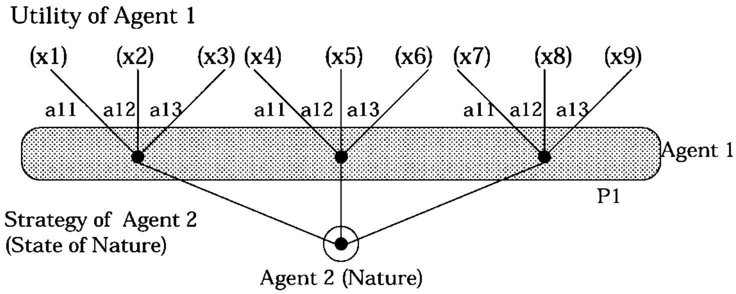


Figure 3. Example of agent competition scenario.

In this example, we consider that *agent_1* does not know the preference relation of *agent_2* and thus, *agent_1* is uncertain about which strategy *agent_2* might adopt. Here, P_1 is an information partition of *agent_1* and it is not sure which nodes it stays in (left, right, or center) within P_1. Under this uncertain environment, *agent_1* must decide which strategy to adopt so as to optimize its utility. Therefore, *agent_1* evaluates its *belief* over the state of nature and adopts the strategy which maximizes its expected utility. If all the agents decide upon strategy in the same way, there is a possibility that it leads to social Bayesian perfect Nash Equilibria (Kajii and Matsui, 2000).

A question which naturally arises here is how each agent assigns its belief autonomously. The answer can be achieved by dividing uncertainty into certain levels. Following Keynes (1883–1946), certainty is divided into three levels according to the amount of information about the state of nature or given signal observed before choosing among several strategies (Ichikawa, 1983).

- *Level 1*: Decision making under certainty. The agent knows exactly what the state of nature is. In this case, decision making is straightforward. The agent selects the strategy based on maximum expected utility (French, 1986).
- *Level 2*: Decision making under risk. It is assumed that the agent is not sure what state of nature is, but it has a probability distribution over the state of nature. In this case, the agent treats the known probability distribution as its belief and selects the strategy which again maximizes its expected utility but broadens the notion of value to include risk. In Section 3 we propose a risk management method to reflect each agent's *attitude toward risk*.
- *Level 3*: *Decision making under uncertainty.* In this level, it is assumed that the agent does not know anything about the state of nature except for that it is in some set, $N = \{\omega_1, \omega_2, \ldots, \omega_n\}$. In this case, the agent has to assign its belief without using a probability distribution. According to cognitive psychology, when probability distribution is not known, people evaluate belief on the basis of *degree of comfort* (i.e., selecting the one that needs the least effort) or *degree of optimism* (i.e., selecting the one that we think is the most fit). In Section 4 we propose a belief assignment method, which reflects agent's *degree of optimism*.

It should be noted that decision making under certainty (*Level 1*) is actually a special case of decision making under risk (*Level 2*).

3. Level 2: Decision Making Under Risk

In the case of decision making under risk, the agent naturally selects a strategy which maximizes its expected utility. Generally, utility is calculated as expected value of cost and/or benefit to indicate a general measure of value. However, problem may arise when expected values of two strategies are the same. In such a case, the attitude toward risk will play a crucial role. That is, the risk of failure influences decision making. The agent's attitude toward risk is categorized into the following three types:

- *Risk prone*: In this case, agents prefer high-risk high-return strategy to low-risk low-return strategy.
- *Risk aversion*: In this case, agents prefer low-risk low-return strategy to high-risk high-return strategy.
- *Risk neutral*: If expected value is the same, these strategies always become nondiscriminateable.

A number of models, such as maximin return (Wald, 1950), optimism–pessimism index (Hurwicz, unpublished discussion paper, 1951), minimax regret (Savage, 1972), and Laplace's principle of insufficient reason, all include the attitude toward risk. We define utility function that reflects the agent's attitude toward risk by

$$u(x) = E(x) - \eta V(x) \tag{1}$$

Where $u(x)$ is a pure benefit when agent adopts some strategy, $E(x)$ is an expected value when agent adopts some strategy, $V(x)$ is a variance, and η is a coefficient of degree of risk aversion taking values between -1 and $+1$. If η is plus, the function $u(x)$ becomes risk aversion, because the larger variance (i.e., the larger risk of failure), the smaller utility becomes. Conversely, if η is minus, function $u(x)$ represents risk prone because, the larger variance, the larger the utility becomes. And if η is zero, $u(x)$ represents risk neutral, because $u(x)$ is equal to the expected value when the agent adopts some strategy. Using this method, agents are allowed to select a strategy reflecting attitude toward risk and this simple representation can be implemented easily (Onjo and Far, 2001).

4. Level 3: Decision Making Under Uncertainty

In the case that the agent has to decide upon the strategy under uncertainty, it has to order its belief set without using a probability distribution. As mentioned earlier, the problem will be reduced to evaluating belief on the basis of *degree of comfort* or *degree of optimism*. The question is how to quantify each agent's *degree of optimism*.

To quantify degree of optimism, we use the OWA operator (Yager, 1988). The OWA operator of dimension n is defined as a function F that is associated with a weighting vector **W**,

$$W = (w_1,\ w_2,\ \ldots,\ w_n) \tag{2}$$

such that $0 \leq w_j \leq 1$ and $\sum_j w_j = 1$; $j \in \{1, \ldots, n\}$ and for any set of values $a_1, \ldots, a_n$

$$F(a_1, a_2, \ldots, a_n) = \sum_j w_j b_j \tag{3}$$

where b_j is the jth largest element in the set $\{a_1, \ldots, a_n\}$.

The OWA weights may be viewed as a pseudo-probability distribution (Yager, 1990). In particular we can view w_j as a probability that of the jth best thing happening. In this case, weights (pseudo-probability) are assigned not to a particular state of nature, but to a preference order of the utility. Thus, w_1 is the weight assigned to the best utility and w_n is assigned to a worst utility.

Here, another question that naturally arises is how the agent assigns the weights it is going to use. At the fundamental level, the answer is that a human expert interacting with the agent subjectively assigns it. But this may be a hard job in autonomous environments. Thus, we propose a method to assign the weight vector automatically reflecting *degree of optimism* of agents. Using the OWA operator, the degree of optimism is defined below.

$$\text{Opt}(W) = \sum_j w_j (n - j)/(n - 1); \qquad n \neq 1 \tag{4}$$

Using this definition, users of agents subjectively decide upon their degree of optimism $\text{Opt}(W)$. This value is fed into a following linear programming equation:

$$\text{maximize} \quad -\sum_j w_j \log_2 w_j \tag{5}$$

$$\text{Subject to} \quad \text{Opt}(W) = \sum_j w_j (n - j)/(n - 1); \quad n \neq 1$$

$$\begin{aligned} &\text{Opt}(W) \in [0, 1] \\ &\sum_j w_j = 1 \\ &w_j \geq 0 \quad \text{for} \quad j = 1, 2, \ldots, n \end{aligned} \tag{6}$$

This approach is closely related to the maximum entropy method used in probability theory, which is a commonly used rationale for selecting a canonical probability distribution from among a set of relevant ones.

Advantage of this method is that for various cardinalities of OWA, we can consistently provide weights corresponding to given $\text{Opt}(W)$.

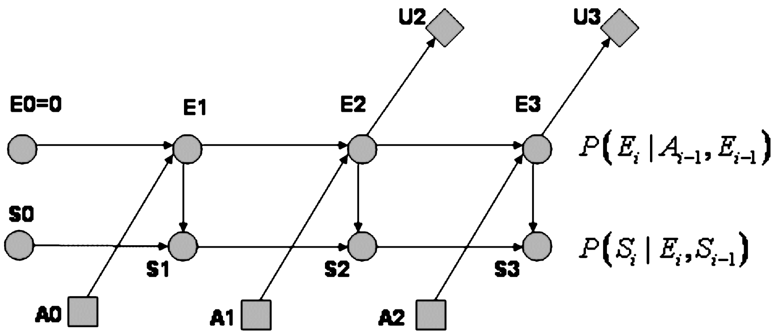


Figure 4. Structure of the DBN.

5. Analyzing Opponents Moves

Using decision-making method mentioned in Sections 3 and 4, the agent can decide upon optimal strategy on each step. The model involves beliefs represented by probabilities and desires represented by utilities. However, to get a stable payoff, the agent should reduce uncertainty by analyzing opponents' moves and updating its belief dynamically. BBN (Pearl, 1988) is added to the model to represent the knowledge in the form of nodes and causal relationships among them. BBN is a powerful computational model for representing causality and reasoning with uncertainty. A special form of BBN, called *Dynamic Belief Network* (DBN), is used for the belief updating process. DBN provides a mechanism to foresee the probability of interest in the next state with regard to the current beliefs. That mechanism is called probabilistic projection and can be performed by a three-step updating cycle called roll-up, estimation, and prediction phases (Russel and Norvig, 1995).

The high-level structure of DBN is shown in Figure 4. It is a directed acyclic graph composed of modeling nodes (E), Sensor nodes (S), decision nodes (A), and utility nodes (U). Both the decision and modeling nodes have a finite set of states. The utility nodes have no children and to each utility node is attached a real valued utility function. To each sensor node a probability value is attached. Each modeling node has a conditional probability table (CPT).

At the low level, each of the sensor node (S) and modeling node (E) is a DBN on itself. They are called *sensor DBN model* and *state evolution DBN model*, respectively. The role of the sensor DBN model is to obtain information related to other agent's strategy and/or behavior, and next state is estimated by state evolution DBN model based upon agents' action and prediction of current state. The two models are composed of the following nodes.

- Type of agent (T). To predict other agents' strategies, an agent has to know other agents' preference relation. Here, *type* is a value which decides each agent's preference relation. Specifically, *type* represents each agent's degree of attitude toward risk, as mentioned in Section 3. For instance, for an agent having three types (risk prone, risk neutral, and risk aversion), its type value T is represented by $T = -1$, 0, and 1, respectively.

Table I. Knowledge hirarchy structure

Agent_2 knows *Agent_1*?	No	Yes	Yes	No
Agent_1 knows *Agent_2*?	Yes	Yes	No	No
State	θ_1	θ_2	θ_3	θ_4

- Knowledge hierarchy structure (K). For the strategic decision making, an agent must analyze not only other agents' types but also the *knowledge hierarchy structure* (Kajii and Matsui, 2000). Knowledge hierarchy structure is the hierarchical structure of each agent's knowledge such as "whether or not all agents know all agents types," "whether or not all agents know 'all agents know all agents types'," . . . , etc.
  For instance, in case of competition among two agents (*agent_1* versus *agent_2*), the knowledge hierarchy structure is defined as follows:

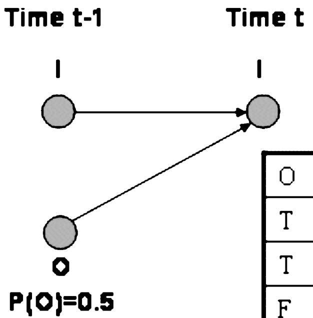
  1. whether or not *agent_1* knows type of *agent_2*
  2. whether or not *agent_2* knows type of *agent_1*
  3. whether or not *agent_1* knows "whether or not *agent_2* knows type of *agent_1* own"
  4. whether or not *agent_2* knows "whether or not *agent_1* knows type of *agent_2* own"

  We can represent knowledge hierarchy structure of this example into four states ($\theta_1 \sim \theta_4$) as shown in Table I. In case of state θ_2, it is said that types of all agent is *Common Knowledge*.
  Note that although four states exist, information partitions of each agent (P_1 and P_2) are

  $$P_1 = \{(\theta_1, \theta_2), (\theta_3, \theta_4)\} \quad \text{and} \quad P_2 = \{(\theta_1, \theta_4), (\theta_2, \theta_3)\}.$$

  Thus, all we have to do is to compute only two states within the information partition of each agent.
- Belief of belief (B). With analyzing knowledge hierarchy structure mentioned earlier, we can specify whether the opponent knows my type. Although it is better to analyze "if opponent does not know my type exactly, how does it misunderstand my type?" Here, *belief of belief* is a concept such as "I believe opponent believes my type is x." For example, if *agent_1* has three types, its belief of belief is represented by $B = -1$, 0, and 1, respectively.
- State of nature (N). State of nature is strategies of the other agents as mentioned in Section 2. For example in case of competition between two agents, if the opponent has three strategies, it is represented by $N = \{\omega_1, \omega_2, \omega_3\}$.
- Existence of signals (I). It represents whether or not an agent can get signals about other agent's strategies. Simply, it is represented by a Boolean variable. Namely, $I = \mathit{True}$ or *False*.

Considering these modeling elements, the sensor and state evolution models can be constructed. Example of the *sensor model* using two-slice temporal belief network (2DBN) is shown in Figure 5.



CPT

O	I	P(I)
T	T	1.0
T	F	1.0
F	T	1.0
F	F	0.0

Figure 5. Example of sensor model.

Agents get and analyze the signal using sensor model. But the agents may not always get signals successfully. Therefore, we add an event O into the sensor model. O is an event which shows agents obtain signals. And by adding CPT, we can construct the sensor model. In this example, we set that agents can get signals with a probability of 0.5.

An example of the 2DBN for the state evolution DBN is shown in Figure 6. It represents the effects of adoption of some strategies as direction of link. In the case of agent competition, we want to know the probability distribution over states of nature N at time t, denoted by N_t. As shown in Figure 6, value of N_t requires knowing T_{t-1} and N_{t-1}. Moreover, N_t and B_t are correlated and B_t and K_t are also correlated. Relations and correlations of each state is described using the conditional probability. Thus probability distribution of current state of nature is calculated as follows:

$$P(K_t, B_t, N_t | S_{t-1}) = P(N_t|B_t, S_{t-1}) \times P(B_t|K_t, S_{t-1}) \times P(K_t|S_{t-1}) \qquad (7)$$

where S_{t-1} is a previous state.

And finally, by multiplying probability of getting signals calculated by sensor model and probability of current state of nature obtained by state evolution model, agent can update its belief, calculate its utility, and select the strategy that maximizes the utility (Figure 4).

6. Example

Decision models presented earlier can be applied to competition among dealer agents in the electronic marketplace for a typical contracting scenario. In this scenario, a customer agent negotiates with a number of dealer agents. Each dealer agent tries to identify customer needs and takes parts in bidding to win the contract. Risk and uncertainty are integral parts of business and especially important in this scenario.

To win the contract, a dealer agent must not bid too high because it will fail to get the contract and loses the time and money spent in preparing the proposal. On the other hand, if the dealer agent bids much lower than its competitors, it loses again because it obtains the contract but it has undertaken to fulfill it at a price far lower than necessary. Therefore,

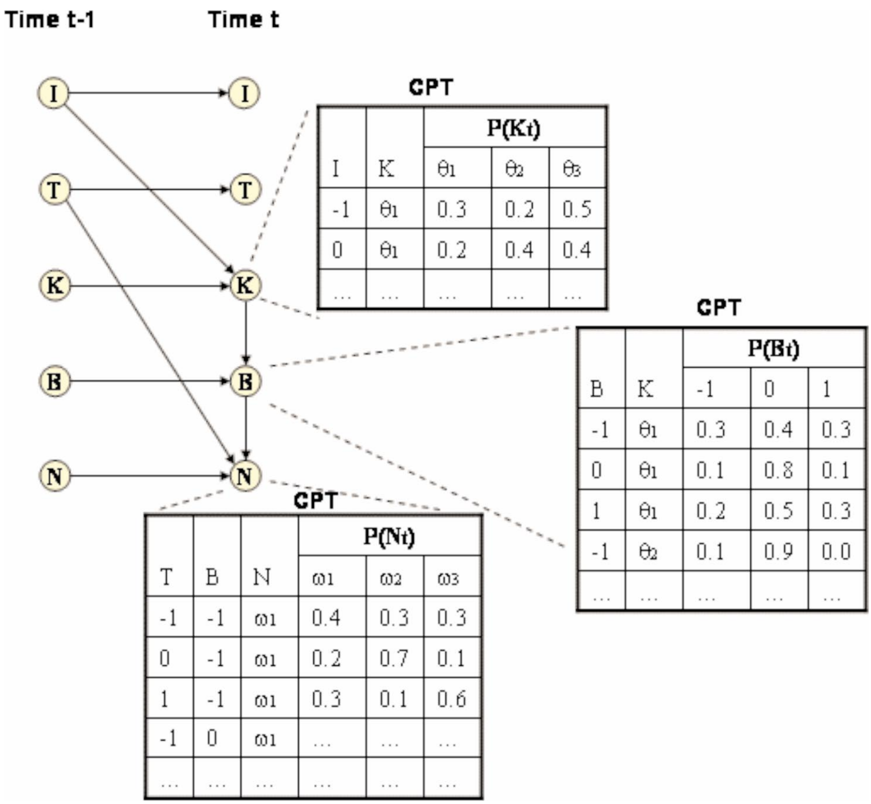


Figure 6. Example of state evolution model.

to win a contract, the dealer agent must bid high enough to make a profit but low enough to win simultaneously.

In this example, we assume that a customer agent negotiate with two dealer agents. Both dealer agents are selling the same product, meaning that there is no product differentiation. This is a simplified model but expressive enough to explain the ideas. Thus inevitably, the dealers will fall into the price competition.

Here, we assume each dealer agent has two strategies to sell the products with “High Price ($40,000)” and “Low Price ($30,000).” We assume that $30,000 is a limitation on price for both dealer agents. Payoff matrix for this price competition is shown in Table II. This game environment is analyzed as *Chicken Game* in economy.

Table II. Payoff matrix for price competition

	Agent_2	
Agent_1	High Price	Low Price
High price	(2, 2)	(0, 3)
Low price	(3, 0)	(−1, −1)

In the case that, both dealer agents adopt “High price” strategy, possibility to win the contract is half because of no product differentiation, meaning that both agents’ payoff is 2. In case that dealer *agent_1* adopts “Low price” and opponent adopt “High price,” dealer *agent_1* can win the contract and obtains payoff 3, and vice versa. In the case that both agent adopt “Low price,” since $30,000 is limitation on price, both agents’ benefit will be deficit because, although they are selling products at low price as much as possible, possibility to win the contract is very low (at most half). In this case, we assume that both payoffs are -1.

In this game environment, best response strategy is to adopt the different strategy from their opponents, “Low price, High price” and “High price, Low price.” At the first glance, it seems that this game is easy to solve, but since opponents’ strategy cannot be predicted, situation gets quite complicated. To cope with the uncertainty, we apply our proposed models. First, we apply decision-making model on the basis of degree of optimism introduced in Section 4. For instance, if Opt $(W) = 0.7$, $[w_1, w_2] = [0.7, 0.3]$. Thus for “High price,” expected utility is $0.7 \times 2 + 0.3 \times 0 = 1.4$ and in case of “Low price,” $0.7 \times 3 + 0.3 \times (-1) = 1.8$. Thus the agent should adopt “Low price.” The plot of expected utility with changing degree of optimism is shown in Figure 7.

As it is shown, until $\text{Opt}(W) < 0.5$, dealer agent adopts “High price” and when $\text{Opt}(W) > 0.5$, then “Low price” is adopted. This is because, since “Low price” has an aspect of high risk-high return, in case that dealer agent is optimistic ($\text{Opt}(W) > 0.5$), it prefers “Low price” but when it becomes pessimistic ($\text{Opt}(W) < 0.5$), it prefers “High price” which has a character of *low risk low return*. This experimental result proves that using the proposed model, interactive decision making is possible even though probability distribution over the competitors is completely unknown.

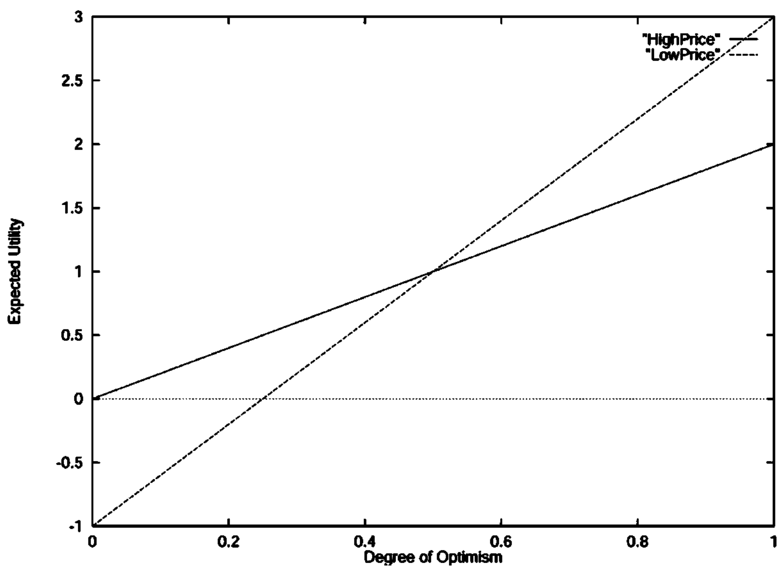


Figure 7. Expected utility with degree of optimism.

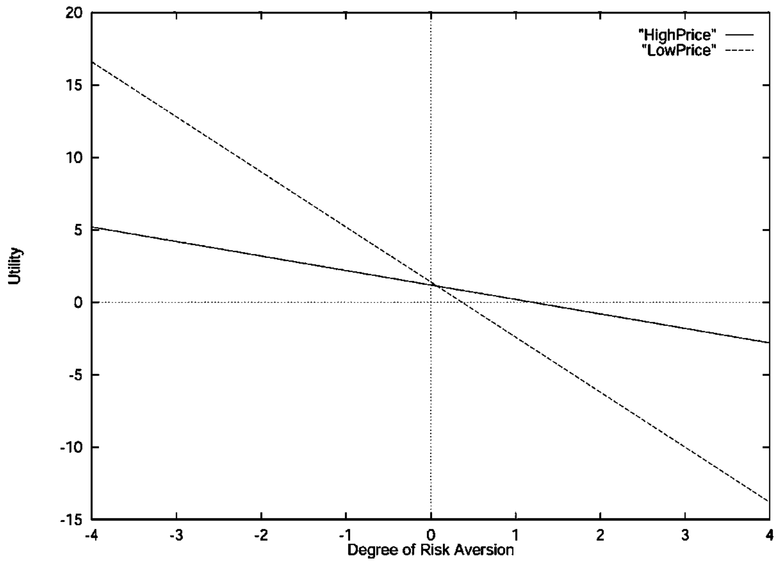


Figure 8. Expected utility with degree of risk aversion.

The next example is related to the risk management methodology presented in Section 3. We assume that, dealer *agent_1* believes opponent adopts "High price" with probability of 0.6 and "Low price" with 0.4. The utility with changing degree of risk aversion (η) is shown in Figure 8. When dealer agent's preference is risk aversion ($\eta > 0$), it prefers "High price" strategy which has a character of low risk low return, because utility of "High price" is higher that that of "Low price." And in case that dealer agent's preference becomes risk prone ($\eta \leq 0$), high risk high return strategy, i.e., "Low price," is adopted. This experimental example shows that our risk management model allows agents make decision with reflection of users' opinion within a simple representation.

7. Conclusions

Next generation of ubiquitous, distributed, and heterogeneous information systems rely on software agent technology. Software agents are knowledgeable entities capable of managing their own set of beliefs and desires and they can decide upon the next execution steps autonomously. They are also interactive entities capable of filtering communications and managing dialogues. Interactions among agents in a society of software agents can be learnt by closely investigating interactions in human societies. In this paper, we devised a realistic set of interaction scenarios for software agents with the focus on decision making in uncertain environments. Decision-making methods in the presence of both risk and uncertainty were introduced and a model and a method for belief update were presented. Because of the introduction of degree of risk and optimism, agents can select a competitive strategy even if probability of competitor's move is completely unknown. From the viewpoint of behavioral psychology, utilization of degree of risk and optimism is natural and from engineering point of view it is practical.

Acknowledgments

This research was partially supported by Natural Sciences and Engineering Research Council of Canada, NSERC.

References

Bierman, H. S. and Fernandez, L., 1998: *Game Theory With Economic Application*, Addison-Wesley, Reading, MA.

Dempster, A., 1997: Upper and lower probabilities induced by a multi-valued mapping, *Ann. Math. Stat.* **38**(2), 325–399.

French, S., 1986: *Decision Theory: An Introduction to the Mathematics of Rationality*, Wiley, New York.

Gibbons, R., 1992: *Game Theory for Applied Economists*, Princeton University Press, Princeton, NJ.

Huhns, M. N. and Singh, M. P., (eds.), 1998: *Readings in Agents*, Morgan Kaufmann, San Mateo, CA.

Hurwicz, L., : *Optimality Criteria for Decision Making Under Ignorance*, Unpublished discussion paper.

Ichikawa, A., 1983: *Decision Making Theory*, Kouritsu Publishing Co. (In Japanese)

Jennings, N. R. and Wooldridge, M. J., (eds.), 1997: *Agent Technology: Foundations, Applications and Markets*, Springer-Verlag, New York.

Kajii, A. and Matsui, A., 2000: *Micro Economics: A Strategic Approach*, Nihon-Hyouron Publishing Co. (In Japanese)

Kuhn, H. W., 1997: *Classics in Game Theory*, Princeton University Press, Princeton, NJ.

Liu, B., 2002: *Theory and Practice of Uncertain Programming*, Springer-Verlag, Berlin.

Onjo, H. and Far, B. H., 2001: A unified view of heterogeneous agents' interaction, *Trans. IEICE* **E84-D**(8), 945–956.

Osborne, M. J., 1994: *A Course in Game Theory*, MIT Press, Reading, MA.

Pearl, J., 1988: *Probabilistic Reasoning in Intelligent Systems*, Morgan-Kaufmann, San Mateo, CA.

Pearl, J., 1990: Reasoning under uncertainty, *Annu. Rev. Comput. Sci.* **4,** 37–72.

Russel, S. and Norvig, P., 1995: *Artificial Intelligence: A Modern Approach*, Prentice-Hall, Englewood Cliffs, NJ.

Savage, L. J., 1972: *The Foundations of Statistics*, 2nd ed., Dover, New York.

Shafer, G., 1976: *A Mathematical Theory of Evidence*, Princeton University Press, Princeton, NJ.

Shortliffe, E. H., Buchanan, B. G. and Feigenbaum, E. A.: Knowledge engineering for medical decision making: A review of computer-based clinical decision aids, *Proc. IEEE*, **67**(9), 1207–1224.

Subrahmanian, V. S., Bonatti, P., Dix, J., Eiter, T. and Ozcan, F., 2000: *Heterogeneous Agent Systems*, 1st edn., MIT Press, Reading, MA.

Wald, A., 1950: *Statistical Decision Functions*, Wiley, New York.

Weiss, G., (ed.), 1999: *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, MIT Press, Reading, MA.

Yager, R. R., 1988: On ordered weighted averaging aggregation operators in multi-criteria decision making, *IEEE Trans. SMC*, **18**(1), 183–190.

Yager, R. R., 1990: Decision making under Dempster–Shafer uncertainties, *Int. J. Gen. Syst.* **20,** 233–255.