May 2006, Vol.21, No.3, pp.412–417 J. Comput. Sci. & Technol.

Agent-Oriented Probabilistic Logic Programming

Jie Wang[1] (王 洁), Shi-Er Ju[2] (鞠实儿), and Chun-Nian Liu[1] (刘椿年)

[1] *Key Laboratory of Multimedia and Intelligent Software, Beijing University of Technology, Beijing 100022, P.R. China*

[2] *Institute of Logic and Cognition, Zhongshan University, Guangzhou 510275, P.R. China*

E-mail: {wj, ai}@bjut.edu.cn; hssjse@zsu.edu.cn

Received May 6, 2005; revised March 10, 2006.

Abstract Currently, agent-based computing is an active research area, and great efforts have been made towards the agent-oriented programming both from a theoretical and practical view. However, most of them assume that there is no uncertainty in agents' mental state and their environment. In other words, under this assumption agent developers are just allowed to specify how his agent acts when the agent is 100% sure about what is true/false. In this paper, this unrealistic assumption is removed and a new agent-oriented probabilistic logic programming language is proposed, which can deal with uncertain information about the world. The programming language is based on a combination of features of probabilistic logic programming and imperative programming.

Keywords agent, uncertainty, probabilistic logic programming, agent-oriented programming

1 Introduction

Agents are software or hardware systems that can autonomously take actions based on changes in their environment[1−3]. In the research community of agents, the concept of BDI-agents[4] is nowadays a well-known notion. A lot of theoretic and practical investigation have been done in order to make this notion both better understood and applicable to real-world problems. The main characteristics of BDI-agents:

- agents have internal mental states: beliefs, desires, plans, and intentions that may change over time;
- agents act pro-actively and reactively
- agents have reflective or meta-level reasoning capabilities.

Therefore, BDI agents can be regarded as goal-directed, belief-transforming entities. That is, agents have a set of goals and try to realize their goals by means-end reasoning or practical reasoning according to their belief and plans. Their belief must be updated in the light of the information about the current situation they find themselves in. Plans are recipes for achieving their goals, and therefore play a crucial role in this reasoning.

In BDI model of agents, it is assumed that there is no uncertainty in an agent's mental states and its environment. In other words, an agent can act through reasoning only with certain knowledge and information (i.e., the agent is 100% sure about what is true/false in the environment). However, in many real-world scenarios, agents developers have to face a partial, uncertain knowledge/information about the environment[3,5−7]. For example, when an image processing agent is asked to identify a military object, it might return the fact that the object could be a civil one with probability 60%–70% and military one with probability 30%–40%.

In order to deal with such uncertainties, this paper extends BDI-agents by using our method of probabilistic logic programming[8]. We chose probability theory as foundation to express uncertainty in the agents. This is because although during the recent decades many schemes for managing uncertainty have been proposed, the most sound one is still probability theory. Moreover, in our setting the uncertainties are estimated by interval probability rather than point probabilities. This is because of two reasons. Firstly, probability intervals are a kind of generalization of point probabilities and so more flexible. Secondly, the knowledge stored in agents are usually from the human users[9,10], and they like to estimate probabilities using intervals more than using points[6,11].

More precisely, we extend Hindriks *et al.*'s logic programming language for BDI-agents[12] to an interval probabilistic logic programming language for uncertain BDI-agents. Our interval probabilistic logic language consists of three components: belief updating, goal updating, and practical reasoning under uncertainty. That is, our language can express uncertain beliefs and goals, update uncertain beliefs and goals, and perform uncertain practical reasonings.

The rest of the paper is organized as follows. In Sections 2–4 we discuss functions of the programming language for the belief updating, goal revision and practical reasoning under uncertainty. Section 5 presents the basic framework of uncertain BDI logic programming language for agents and presents a program using this language. Section 6 discusses the related work. The final section concludes the paper.

2 Uncertain Beliefs

The uncertain beliefs of an agent can be expressed

Short Paper

This work is supported by the National Natural Science Foundation of China under Grand No. 60496322 and the Chinese Ministry of Education under Grand No. 05JZD720.4001.

in first-order formulae from a language L consisting of a finite set of predicate symbols and finite set of variable symbols. Let ρ be a sub-interval of $[0,1]$, the basic elements of the language are given by a signature $\Sigma = (P, F, C, A, \rho)$, where P is a set of predicate symbols, F is a set of function symbols, C is a set of constants, A is a set of action symbols, ρ is a set of probabilistic constants. Terms and atoms of a language L are defined as usual from a signature Σ and an infinite set of variables $TVar$. The notion of variable substitution, unifier, most general unifier, and variant are defined as usual[13]. The set of atoms is denoted by At.

A ρ-atom is an expression of the kind E/ρ with an atom E and interval probability ρ which is called the probability that E occurs. When $\rho = [1,1]$ (abbreviated for $E/1$), it means E is true; when $\rho = [0,0]$ (abbreviated for $E/0$), it means E is false; and when $\rho \subset (0,1)$, it means E probably occurs. Intuitively, E/ρ means that the probability of E being true lies in the interval ρ. The set of ρ-atoms is denoted by $\rho\text{-}At$.

After giving the notation, we turn to discuss how to calculate the uncertainties of one belief that is the combination of a number of uncertain sub-beliefs. The probability that a conjunction of two events is true depends on not only the probabilities of the individual conjuncts, but also the dependencies between the events denoted by these conjuncts[14]. The notion of a probabilistic conjunction strategy defined below[5] captures these different ways of computing probabilities via Definition 2.

Definition 1. *A probabilistic conjunction strategy is a mapping $\otimes$ which maps a pair of probability intervals to a single probability interval satisfying the following axioms*:

1. *Bottomline*: $[l_1, u_1] \otimes [l_2, u_2] \leqslant [\min(l_1, l_2), \min(u_1, u_2)]$, *where* $[x, y] \leqslant [x', y'] \Leftrightarrow x \leqslant x' \wedge y \leqslant y'$;
2. *ignorance*: $[l_1, u_1] \otimes [l_2, u_2] \subseteq [\max(0, l_1 + l_2 - 1), \min(u_1, u_2)]$;
3. *identity*: $[l_2, u_2] = [1,1] \Rightarrow [l_1, u_1] \otimes [l_2, u_2] = [l_1, u_1]$;
4. *annihilator*: $[l_1, u_1] \otimes [0,0] = [0,0]$;
5. *commutativity*: $[l_1, u_1] \otimes [l_2, u_2] = [l_2, u_2] \otimes [l_1, u_1]$;
6. *associativity*: $([l_1, u_1] \otimes [l_2, u_2]) \otimes [l_3, u_3] = [l_1, u_1] \otimes ([l_2, u_2] \otimes [l_3, u_3])$;
7. *monotonicity*: $[l_2, u_2] \leqslant [l_3, u_3] \Rightarrow [l_1, u_1] \otimes [l_2, u_2] \leqslant [l_1, u_1] \otimes [l_3, u_3]$.

Intuitively, $[l_1, u_1]$ and $[l_2, u_2]$ are intervals in which the probabilities of events e_1 and e_2 are located, and $[l_1, u_1] \otimes [l_2, u_2]$ returns a probability range for the concurrence of both these events. Now we use an example to explain the above setting further.

Example 1. A Robot called Orville wandering around in a two-dimensional grid world. There are only military and civil objects. Sometimes Robot Orville cannot make sure whether an object is a military one or not. That is, it has uncertain belief. The basic predicates which describe this world are: $Orville(x, y)/1$ for "Robot Orville is at coordinates (x, y)"; $military(x, y)/\rho_1$ for "there is a military-object at coordinates (x, y) with the probability of ρ_1"; and $civil(x, y)/\rho_2$ for "there is a civil-object at coordinates (x, y) with the probability of ρ_2". So in the probability space, $military(x, y)/\rho_1$ also means that there is a civil-object at coordinates (x, y) with the probability of $1 - \rho_1$. Similarly, $civil(x, y)/\rho_2$ also means that there is a military-object at coordinates (x, y) with the probability of $1 - \rho_2$.

In this example, Orville has uncertain knowledge of its environment. That is, although it knows its own position at any moment and the location of objects, Orville has only uncertain belief in the nature of the objects.

Example 2. The Robot Orville can take the following basic actions: *west* for moving west, *east* for moving east, *north* for moving north, *south* for moving south, and *destroy* for destroying an object that is believed to be a military object with the probability ρ. The goal of Orville is to destroy the military objects. The goal is denoted by the predicate *destroy-military*, which is a user-defined predicate and is in fact given below as a usual procedure definition.

3 Goals and Action Selection

In our setting, agents try to achieve two basic kinds of goals: goals to do some actions and goals to achieve some mental state of affairs. These two basic kinds can compose different complex ones in the following manner.

Definition 2. *Let $\sum = (P, F, C, A, \mu)$ be a signature, and Gvar an infinite set of goal variables ranging over goals. The set of goals L^g is inductively defined by*:

1. $A \subseteq L^g$, *called basic actions*;
2. $\rho\text{-}At \subseteq L^g$, *in particular*, $At \subseteq L^g$;
3. *if* $\varphi \in L$, *then* $\varphi? \in L^g$;
4. $Gvar \subseteq L^g$;
5. *if* $\pi_1, \pi_2 \in L^g$, *then* $\pi_1; \pi_2, \pi_1 + \pi_2, \pi_1 || \pi_2 \in L^g$.

Basic action $a \in A$, achievement goal $\varphi \in \rho\text{-}At$, and test goal $\varphi?$ are the basic goals in our language. Basic actions are updating operators on the uncertain belief base of an agent. Achievement goal φ is a goal to achieve a state where φ holds. Test goal $\varphi?$ is to check the uncertain belief base to see whether or not φ holds.

It is also possible to build more complex compositions of basic goals, by using the program to construct sequential composition(;), nondeterministic choice(+), and parallel composition(||). Assume π_1 and π_2 are ρ-atom E_1/ρ_1 and E_2/ρ_2, respectively. This allows us to specify conjunctive goals $\pi_1 || \pi_2$ by means of parallel composition and nondeterministic goals $\pi_1 + \pi_2$ by means of nondeterministic choice.

With respect to the nondeterministic action selection, we present an utility-theory-based approach for modeling. That is, an agent that chooses, for example, A_1 or A_2, based on not only the uncertain beliefs of the objects but also the utility of performing the action in pursuing the goal. For example, when an agent decides to destroy an object, it considers not only the probability that the object is a military one, but also the benefit (loss) for destroying the object.

First, we briefly review the basic idea of utility theory[14]. Utility is a measure of a decision maker's preferences on courses of action under uncertainty and risk (i.e., uncertainty with known probabilities). Utility theory assumes that the decision maker always chooses the alternative for which the expected value of the utility (expected utility) is maximum. In this paper, we can use utility function to determine what actions the agent should take — the ones with the highest expected utility (if more than one actions have the highest expected utility the agent will choose one randomly).

Usually, the probability is a real point in $[0,1]$. So, it is easy to calculating the expected utility of each action. But in our setting, the uncertainties of agents' beliefs are represented by means of probability intervals (see Table 1). Thus, we have to extend utility theory to deal with the setting of interval probability.

Table 1. Utility of Action in Uncertain States

state	b_1	$\cdots$	b_n
state's probability	$[l_1, u_1]$	$\cdots$	$[l_n, u_n]$
action A_1	u_{11}	$\cdots$	u_{1n}
$\vdots$	$\vdots$		$\vdots$
action A_m	u_{m1}	$\cdots$	u_{mn}

Now we present a new approach to calculating the interval that the expected utility falls in when the probabilities of an agent beliefs are intervals. We use operation on interval numbers[15] to calculate the maximum and minimum values of expected utility of decision action A_i $(i = 1, 2, \ldots, m)$. Let $\widetilde{m} = [a, b]$ and $\widetilde{n} = [c, d]$, then

$$\widetilde{m} + \widetilde{n} = [a + c, b + d], \tag{1}$$

$$\widetilde{m} - \widetilde{n} = [a - d, b - c], \tag{2}$$

$$\widetilde{m} \times \widetilde{n} = [\min\{ac, bc, ad, bd\}, \max\{ac, bc, ad, bd\}], \tag{3}$$

$$\widetilde{m}/\widetilde{n} = [a/d, b/c], \quad (a > 0,\ c > 0). \tag{4}$$

Thus, by (1) and (3), according to Table 1 the interval expected utility of decision action A_i is:

$$E(A_i) = \sum_{j=1}^{n} u_{ij}[l_j, u_j] = \Big[\sum_{j=1}^{n} u_{ij} l_j, \sum_{j=1}^{n} u_{ij} u_j\Big]. \tag{5}$$

That is, the interval that $E(A_i)$ falls into is $[\sum_{j=1}^{n} a_{ij} c_j, \sum_{j=1}^{n} a_{ij} d_j]$. In order to choose an action that has the biggest expected utility, we give the following method to compare two actions' interval expected utilities.

Definition 3. *Action A_i is more preferable than A_j if*

$$\alpha u_{E(A_i)} + (1 - \alpha) l_{E(A_i)} \geqslant \alpha u_{E(A_j)} + (1 - \alpha) l_{E(A_j)},$$

where α is a predetermined constant (called optimistic coefficient), $l_{E(A_k)}$ and $u_{E(A_k)}$ $(k = i, j)$ are the lower and upper boundaries that the action A_k's expected utility falls into respectively.

Intuitively, this definition means the greater the boundaries the better the action.

Now we examine a simple example using interval expected utility.

Example 3. Orville intends to destroy a military object. However, for some reasons, it cannot make sure whether or not the nearest object to its current position is a military one. The utility value and probabilities for each state and action are as shown in Table 2.

Table 2. Utility of Action in Uncertain Object

Object	Military	Civil
Object's probability	$[0.3, 0.6]$	$[0.4, 0.7]$
action A_1: *destroy*	3	2
action A_2: *look for*	1	3

Thus, by (5), we can obtain the interval expected utilities of each decision action:

$$E(A_1) = [1.7, 3.2], \quad E(A_2) = [1.5, 2.7].$$

Then, given $\alpha = 0.5$, by Definition 3, we get

$$\alpha u_{E(A_1)} + (1 - \alpha) l_{E(A_1)} \geqslant \alpha u_{E(A_2)} + (1 - \alpha) l_{E(A_2)}.$$

That is, action A_1 is more preferable than action A_2. So, Oriville will move to the object and then destroy it. However, clearly if Oriville is a traditional agent (given $\alpha = 0.2$), action A_2 will be more preferable.

In addition, the programming language for BDI-agent includes variables which range over goals. These variables can be used for several purposes. The first one is for reflective reasoning. This will become clear in the next section, where these variables are also allowed in the head of practical reasoning rules. The second is for communication. For example, an agent might receive in a goal variable a request to establish some goals.

4 Uncertain Practical Reasoning Rules

To achieve its goals an agent has to find the means for achieving them, and sometimes may have to revise its goals. This kind of reasoning is called practical reasoning. To perform this type of reasoning an agent uses a set of practical reasoning rules.

Definition 4. *Let $\varphi \in L, \pi, \pi' \in L^g$. Then an uncertain practical reasoning rule is in the form of*

$$\pi \leftarrow \varphi | \pi' \in L^p,$$

where

- *π is called the head of the rule;*
- *π' is called the body of the rule;*
- *φ is called the guard of the rule;*
- *the free variables in the head of a rule are called global variables of the rule; and*

- *all variables in a rule that are not global are called local variables.*

Since φ, π, π' are ρ-atoms with uncertainty, practical reasoning rules can actually be used to carry out the reasoning under uncertainty. The guard can be used to specify the context in which the rule might be applied and to retrieve data from the set of uncertain beliefs.

The function of a practical reasoning rule is two-fold. First, a rule can specify the means to achieve a particular goal under uncertainty. In this case, uncertainty is involved in both of the head and the body (a plan) of the rule. A plan rule encodes the procedural knowledge of an agent. Thus, an agent has a plan library it consults to find the means to execute actions and perform a kind of dynamic planning during execution.

Example 4. Robot Orville wants to destroy military objects in its world. A plan rule which specifies how to achieve this goal is the following:

$$\begin{aligned}&\textit{destroy-military}\\ \longleftarrow\, &\textit{Orville}(x_0,y_0)\wedge \textit{nearest}(x_0,y_0,x,y)\\ &|\,\textit{Orville}(x,y);\textit{military}(x,y)/\rho?;\\ &\textit{prefer}(\textit{destroy}(x,y))?;\textit{destroy-military}.\end{aligned}$$

Thus, when the agent has uncertain belief for objects, the plan tests action *destroy* is more preferable according to utility theory, destroys that military object, and then recursively starts destroying military objects again. The guard retrieves the current position of Orville and the position of the nearest object.

A plan for getting Orville at a specific position is the following:

$$\begin{aligned}&\textit{Orville}(x,y)\\ \longleftarrow\, &\textit{Orville}(x_0,y_0)|(x=x_0\wedge y=y_0)?+[(x<x_0)?;\\ &\textit{west}+(x_0<x)?;\textit{east}+(y<y_0)?;\\ &\textit{south}+(y_0<y)?;\ \textit{north}];\textit{Orville}(x,y).\end{aligned}$$

The plan (i.e., the body of the rule) states that to get at position (x,y), the agent should repeat making a move in the right direction until that position. This rule illustrates the two uses of predicates in our language. The predicate $\textit{Orville}(x,y)$ in the head denotes a possible subgoal of the agent, while the predicate $\textit{Orville}(x_0,y_0)$ in the guard denotes a test on the belief base. By using a predicate both in the head (as a goal) and in the guard (as a belief) in a practical reasoning rule an interface between the belief and goal bases is established. Thus, predicates in goals are not just procedurally defined, but can be related to their logical interpretation as beliefs via uncertain practical reasoning rules.

The second purpose of uncertain practical reasoning rules is to revise goals. Change of uncertain belief can lead to two types of situations in which a rational agent might wish to revise his goals (i.e., in case a more optimal strategy can be followed, or in case of failure). We show an instance of each of these cases of goal revision, also illustrating the use of goal variables. Firstly, we consider the former (i.e., a better strategy can be followed). We use the method of default logic which is based on decision-theoretic default[14]. The formulation of decision theoretic defaults takes probability and utility into consideration. Let e be a formula in the propositional calculus, and $A=\{a_1,\ldots,a_n\}$ a set of possible alternate action (the possible actions being primitive), and $a\in A$. Then

$$e\rightarrow{}_A a\quad \text{if } EU(a,e)=\max_{a_i\in A}EU(a_i,e).$$

In other words, given all that we know (contingently) is e, a is the action in A that maximizes expected utility. For uncertain practical reasoning rule, we modify the definition of decision-theoretic default to deal with interval-probability-oriented decision. That is,

$$e\rightarrow{}_A a\quad \textit{if }\ \epsilon(a,e)=\max_{a_i\in A}\epsilon(a_i,e).$$

where $\epsilon(a_i,e)=\alpha u_{E(a_i)}+(1-\alpha)l_{E(a_i)}$. So, the goal updating depends on not only the probability and utility of each state and action but also the optimistic attitude coefficient. Optimistic coefficient reflects the tendency of an agent.

Example 5. In Example 3, Orville has a goal to destroy an military object with the probability $[0.3,0.6]$. But with Orville going nearer to the object, the agent thinks the object is a civil one with greater probability $\rho_1=[0.6,0.8]$ and a military one with $\rho_2=[0.2,0.4]$. The utility values for each state and action are changeless. Now at the situation, for $\alpha=0.5$,

$$\alpha u_{E(a_1)}+(1-\alpha)l_{E(a_1)}\leqslant\alpha u_{E(a_2)}+(1-\alpha)l_{E(a_2)}.$$

This means action "look for another object" is more preferable. So a better strategy is to revise its current goal and look for another military object. The following rule makes this type of goal-revision possible:

$$\begin{aligned}&G:\textit{Orville}(x,y)\\ \longleftarrow\, &\textit{Orville}(x_0,y_0)\wedge \textit{military}(x,y)/\rho_2\\ &\wedge\neg\textit{prefer}(\textit{destroy}(x,y))\ |\ \textit{Orville}(x_1,y_1).\end{aligned}$$

The rule applies when Robot Orville has a goal of doing some (probably empty) list of actions G.

In another situation, for example, where Orville still has the goal of going west, but now a civil object is in the way and Orville cannot move west. If Orville tries to move west, it fails. Since Orville is assumed to have perfect knowledge for its position, it immediately detects the failure and knows that it is still at its old position. To avoid such failure Orville may use the following rule:

$$\begin{aligned}&\textit{West};\textit{Orville}(x,y)\\ \longleftarrow\, &\textit{Orville}(x_0,y_0)\wedge\textit{civil}(x_0-1,y_0)\\ &|\,[(y_0\leqslant y)?;\textit{north}\\ &+(y_0\geqslant y)?;\textit{south};\textit{Orville}(x,y).\end{aligned}$$

This rule detects the failure and makes it possible to revise the goal and employ a strategy to go around the civil object (not a very adequate one).

We need a number of technical definitions in order to proceed. A term, formula, or program, and *a free variable* are defined in the usual way. The *set* of free variables in an expression c is denoted by $Free(c)$. The notions of *variable substitution*, *unifier*, *most general unifier*, and *variant* are defined as usual[13].

5 Probabilistic Agent Programs

Agents are goal-directed and belief-transforming entities that are capable of means-end reasoning and have reflective capabilities. We assume that beliefs are updated by uncertain information, and goals are updated by execution and revision. This clearly separates the two different types of updating. The uncertain practical reasoning component is encoded in uncertain practical reasoning rules. Thus, an agent's beliefs and goals can change but practical reasoning rules and basic actions are fixed. Beliefs and goals constitute the mental state of an agent. Formally, we define:

Definition 5. *A mental state is a pair* $\langle \Pi, \sigma \rangle$, *where*

- $\Pi \subseteq L^g$ *is a set of goals called a goal base, and*
- $\sigma \subseteq L$ *is a set of beliefs called a belief base.*

The dynamics of behavior of an agent, therefore, is fully specified if the semantics of basic actions is given and the mechanisms for executing goals and applying rules are defined.

Definition 6. *The semantics of basic actions* A *is given by a transition function* $\mathcal{T}$ *of type:* $B \times B \to \varphi(A)$. *Let* $a \in A$. *We use the following notational convention, and write*: $\langle \sigma, \sigma' \rangle$ *a for* $a \in \mathcal{T}\langle \sigma, \sigma' \rangle$.

Example 6. Robot Orville is capable of performing five basic actions. The operational semantics of action "west" is given by:

$$(\{\cdots, Oriville(x,y),\ not(civil(X-1,Y),\cdots,\},\{\cdots,\ Oriville(X-1,Y),\ not(civil(X-1,Y),\cdots\})\,west.$$

This means action "west" can be performed if Orville believes it is at coordinates (X, Y), and after performing the action its beliefs change such that Orville believes it is at coordinates $(X-1, Y)$.

Similar definitions can be applied to actions east, north, and south,

$$(\{\cdots, military(X,Y)/\rho, Oriville(X,Y),\cdots\},\{\cdots,\ Orville(X,Y),\ not\{object(X,Y),\cdots\})\,destroy.$$

Note that in the case if there is a civil-object blocking the way to the west, although the west action will fail in the real world, Orville will still *believe* he has made a move to the west. Only by observing the environment Orville notices that his action of moving west has failed. Similar remarks are applied to the other actions.

Thus, to program an agent is to specify its initial mental state, the semantics of the basic actions that the agent can perform, and a set of uncertain practical reasoning rules. This is formally represented in the next definition.

Definition 7. *A probabilistic agent program is a quadruple* $(\mathcal{T}, \Pi_0, \sigma_0, \Gamma)$, *where*

- $\mathcal{T}$ *is a basic action transition function, specifying the effect of basic actions*;
- Π_0 *is the initial goal base*;
- σ_0 *is the initial belief base*; *and*
- Γ *is a set of practical reasoning rules* (*PR-base for short*).

Example 7. The agent program for Orville is the following:

- $\mathcal{T}$ is defined for the basic actions in Example 6;
- *the initial goal base is given by:* $\{destroy\text{-}military\}$;
- the initial belief base is given by

$$\{Orville(0,0),\ civil(1,5)/\rho_1,\ civil(3,3)/\rho_2,\ civil(2,1)/\rho_3, military(2,2)/\rho_4\};$$

- the PR-base contains the PR-rules as defined in Examples 4 and 5.

The agent program works according to the following steps. Firstly, it will test if the action "destroy" to the nearest object is more preferable according to utility theory. If yes, it will act according to the rule in Example 4; otherwise agent will looking for other object. Secondly, if agent's belief changes in the processing of go-ahead, it will revise the goal according to the rule in Example 5.

Notice that there are two special cases of uncertain practical reasoning rule. Firstly, the rule with an empty body. For example, Orville tests before destroying an object whether the action "destroy" is more preferable. In case it is not, however, Orville will fail in the test, and the goal becomes infeasible. This situation can be represented by the following rule:

$$prefer(destroy(x,y))? : destroy \longleftarrow not(prefer(destroy(x,y)))\ |\ .$$

Secondly, the rule with an empty head, for example, in case Orville happens to be at a position of a military-object, Orville should destroy it, regardless of what other goals Orville has at that moment. This can be achieved by the following rule,

$$\longleftarrow\ Orville(X,Y) \wedge military(X,Y)\ |\ destroy.$$

We call this type of rules *data-directed.*

Data-directed rules create new goals on the basis of beliefs only. These rules are a distinguishing feature of agents, although they do resemble interrupts. Data-directed rules could be used to make agents reactive to their environment (like interrupts) but the level of reactiveness accomplished also highly depends on the underlying control structure of the language.

6 Related Work

Some researchers also tried to extend the BDI agent model to the situation of uncertainty. For example, Parsons and Giorgini[16] extended the work of Parsons, Sierra and Jennings[17] on the use of argumentation in BDI agents[4] to uncertainty. In their work, the degree of belief is represented by a mass assignment in evidence theory[18]. When updating its belief set in the light of new information, the combination rule of evidence theory is used to calculate the degree of each piece of information in the belief set. Evidence theory is also employed to handle the issue of fusing an agent's belief with the information obtained from its sensors by Li and Zhang[9]. Further, in their work[15], the theory is used to fuse the information that is from an agent's own sensors, and uncertainty that it is told by other agents. The work of Li and Zhang[9,15] deals with the uncertain belief fusion, but disregards the uncertain belief revision while the issue of belief revision is dealt with in the work of Parsons and Giorgini[16]. However, all of them cannot handle the issues of goal and action selection and revision under uncertainty and the issue of uncertain practical reason. Nevertheless, these limitations have been removed in our work that is presented in this paper.

On the other hand, compared with previous work on logic programming for agents, our work is also novel. For example, although Dix and Subrahmanian[5] gave a proposal for programming probabilistic agents, and show how, given an arbitrary program written in any imperative program, they may build a declarative probabilistic agent program on top of which supports decision making in the presence of uncertainty. However, unlike our work, theirs is not based on BDI concepts.

In addition, although fuzzy information is put into consideration in the agent systems[3,7], no concepts of probability and BDI are involved in these agent systems.

7 Conclusion

Uncertainty is unavoidable in agent systems. In this paper, we extend the BDI agent model such that belief-updating, goal-updating and practice reasoning can be carried out under uncertainty. Our approach consists of three components for programming uncertain BDI agent. The first one is the function to update uncertain belief based on interval probability theory. The second one is that goal selection and revision based on interval utility theory. The third one is the propagation of uncertainty along practical reasoning.

References

[1] Ferber J. Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence. Addison-Wesley, 1999.

[2] Jennings N R, Sycara K, Wooldridge M. A roadmap of agent research and development. *Autonomous Agent and Multi-Agent Systems,* 1998, 1: 7–38.

[3] Luo X, Jennings N R, Shadbolt N, Leung H F, Lee J H M. A fuzzy constraint based model for bilateral, multi-issue negotiation in semi-competitive environments. *Artificial Intelligence*, 2003, 148(1-2): 53–102.

[4] Rao A S, Georgeff M P. BDI agents: From theory to practice. In *Proc. the First International Conference on Multi-Agent Systems*, San Franciso, CA, USA, 1995, pp.312–319.

[5] Dix J, Subrahmanian V S. Probabilistic agent programs. *ACM Transactions on Computational Logic*, 2000, 1(2): 207–245.

[6] Zhang C, Luo X. An issue on transformation of interval-based uncertainty in distributed expert systems. In *Poster Proc. the 10th Australian Joint Conference on Artificial Intelligence*, Perth, Australia, 1997, pp.38–43.

[7] He M, Leung H F, Jennings N R. A fuzzy logic based bidding strategy in continuous double auctions. *IEEE Transactions on Knowledge and Data Engineering*, 2003, 15(6): 1345–1363.

[8] Wang J, Ju S, Luo X. Probabilistic logic programming with inheritance. In *Proc. the 7th and 8th Asian Logic Conferences*, World Scientific, Singapore, 2003, pp.409–422.

[9] Li Y, Zhang C. Information fusion and decision making for utility-based agents. In *Proc. the Third World Multi-Conference on Systemic, Cybernetics and Informatics and the Fifth International Conference on Information Systems Analysis and Synthesis*, Orlando, USA, 1999, pp.377–384.

[10] Luo X, Jennings N R, Shadbolt N. Acquiring user tradeoff strategies and preferences for negotiating agents: A default-then-adjust method. *International Journal of Human Computer Studies*, 2006, 64(4): 304–321.

[11] Luo X, Zhang C, Jennings N R. A hybrid model for sharing information between fuzzy, uncertain and default reasoning models in multi-agent systems. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 2002, 10(4): 401–450.

[12] Hindriks K V, de Boer F S, der Hoek W van, Meyer J J C. Formal semantics of an abstract agent programming language. In *Proc. International Workshop on Agent Theories, Architectures, and Languages (ATAL '97)*, LNCS 1365, Springer, 1998, pp.215–229.

[13] Lloyd J. Foundations of Logic Programming. Springer, 1984.

[14] Poole D. Decision theoretic defaults. In *Proc. the 9th Biennial Canadian Artificial Intelligence Conference*, San Francisco, CA, USA, 1992, pp.190–197.

[15] Li Y, Zhang C. Information-based cooperation in multiple agent systems. *Advanced Topics in Artificial Intelligence*, LNAI 1747, Springer, 1999, pp.496–498.

[16] Parsons S, Giorigini P. An Approach to Using Degrees of Belief in BDI Agents. *Information, Uncertainty and Fusion*, Kluwer Academic Publisher, 2000, pp.81–92.

[17] Parsons S, Sierra C, Jennings N R. Agents that reason and negotiate by arguing. *Journal of Logic and Computation*, 1998 8(3): 261–292.

[18] Shafer G. A Mathematical Theory of Evidence. Princeton University Press, 1976.