ソフトウェア工学 121－4
(1998. 11. 5)

不確実な環境における対話と事例ベーズに基づくマルチエージェント学習言語

シデイ オ. スエナ　B.H. ファー　河野　善彌

〒338-8570 埼玉県浦和市下大久保255
埼玉大学情報システム工学科
Tel. 048-858-3489,　Fax. 048-858-3716
E-mail. sidi@cit.ics.saitama-u.ac.jp
http://www.cit.ics.saitama-u.ac.jp/~sidi/

あらまし:
エージェントとその競合的な環境における動作に関する情報は不完全で、雑音が多い。本研究においてエージェントが行動し相互作用し他のエージェントの環境、戦略を学びそれぞれのプランを構築できるような新たな言語を導入する。その行動と相互作用はスピーチアクト理論とゲーム理論に基づく。現在の環境の状態は事例として エージェントダイナミックメモリにおいて評価、保存される。その際事例ベース推論技術と他の推論を用いる。エージェントは不明なアクション（事例）、類似した、また競合する事例を処理しなければならない。このプロセスを通してエージェントは経験に基づいて事例を説明し、付加し、削除し、洗練することによってエージェントのバックグラウンドの知識との一貫性を増加させる。本研究においては、電子商取引はディーラー、マニュファクチャラ、サーチ、カスタマ、カタログ、デリバリ、そしてバンクの7つのエージェントのソサエティとして定義される。新しい言語はあらゆる状況においても使用可能である。本研究では電子商取引の中でも特に互いに競合しているディーラーとデリバリエージェントにおいて言語の実装を行う。新しい言語の構文、意味の定義を示す。

和文キーワード:　ゲーム理論、競争、不確実性、ソフトウェアエージェント、エレクツロニックスコマース。

Multi-agent Interaction and Case-Based Learning Language for Competitive and Uncertain Environments

Sidi O. Soueina　B. H. Far　Zenya Koono

Department of Information and Computer Science, Saitama University
255 Shimo-okubo, Urawa 338-8570, Saitama, Japan
Tel. +81-48-858-3489,　Fax. +81-48-858-3716
E-mail. sidi@cit.ics.saitama-u.ac.jp
http://www.cit.ics.saitama-u.ac.jp/~sidi/

Abstract:
Information about agent and actions they perform in competitive environments are incomplete and noisy. In this work we introduce a new language that enables agents to act, interact and learn the environment and strategies of their opponents, and to construct their own plans. The action and interaction are based on Speech Act Theory (SAT) and Game Theory formalisms. The current environment situations are assessed stored in agents dynamic memories as cases, and using the Case Based Reasoning (CBR) techniques, and other inferential methods, agents are to deal with unknown actions (cases), similar and conflicting cases, etc. This process results in increasing the consistency of the background knowledge of agents by explaining, adding, deleting and refining cases due to the experience that they encounter. Electronic Commerce (EC) is defined in our project as a society of seven agents, Dealer, Manufacturer, Search, Customer, Catalog, delivery and Bank agents. While the language can be used in any competitive and hostile situations, in this work we demonstrate an implementation of the language in electronic commerce, specifically in the case of dealer and delivery agents competing with each other. The definition of the language's syntax and semantics are demonstrated.

Key words　*Game Theory, Competition, Uncertainty, Software Agent, Electronic Commerce.*

1 Introduction

A lot of research have been done to satisfy societal needs of agents interacting in multi-agent environment [6][17][21]. Learning however is one of the topics that took several forms and perspectives, such as Federated and Reinforcement Learning [6][17]. Nevertheless a real tool that allows agents to accomplish useful competitive conclusions in hostile environments is still missing. A variety of methods and techniques have been suggested to deal with multi-agent environments [6], such as cooperation [19], coordination [1], and, Agent Oriented Programming (AOP) is defined for agents having mental qualities such as beliefs, commitments and desires [22]. Nevertheless, the hostile multi-agent environment creates challenging opportunities for learning, that can take various perspectives and forms, for example *agents learning the environment and each other's strategy.* In this research we introduce a new Case Based Reasoning (CBR) learning language that enables agent to bring the ultimate in competition. The language is called '*Multi-Agent Learning Language*' (MALL) and it combines game theory concepts, uncertainty and inferential learning issues and CBR, all together as a perfect frame for individual competitive learning. Using this language agents are able to explain other agents' actions, understand the environment, learn agents strategies and create an optimal plan despite the noise, hostility, and uncertainty.

1.1 Learning In CBR

There are many ways and tools that can be used to Model and to satisfy the different goals in a society of agent. For instance, using game theory slightly modified is found to be suitable for providing tools for various Multi-Agent Systems environments [12]. Learning and Interaction are found to be useful in multi-agent environments, where *Learning* and *uncertainty* are major factors to be treated. For example competitive agents wanting to learn each other's strategies should take into consideration the probability assessment in any evaluations they do.

Learning in CBR is most appropriate for this type of situations [25][26][27]. Cases provide sources of knowledge that can be used to support almost every type of inference and a case-based system can easily be made to learn by accumulating new experiences. That is to say, every time the agent deals with the environment given a case, it accumulates experiences in terms of dealing with the same case in the future or similar ones by applying adaptation. If the adaptation however fails the system needs to apply more sophisticated learning techniques. In this work we introduce a way of enabling the system to learn after adaptation failure.

1.2 Decision Making Under Uncertainty

In a multi-agent environment agents are to encounter uncertainty engendered by the lack of knowledge when devising optimal explanations to observable actions of other parties. In this project agents are categorized by *friendly*, *competitive* and *mistrusted* zones, in which certainty is divided into three levels according to the amount of information about the environment and about other parties, given the signal function observed before choosing among several choices [23]. These levels are: In case where the agent is certain (*decision making under certainty*); agent knows exactly what state of nature to choose, and therefore selects the alternatives that gives the highest payoffs. In case the agent is not quiet sure (*decision making under risk*); the case in which the agent knows with the probability $0 < P < 1$ that S is/was/will be the state of nature. Thus, after calculating the expected payoffs of each alternative (A), it selects the alternative with the largest payoffs $C^* = MaxC_A$ (where C_A is the payoff of alternative A). The third situation is when an agent does not know anything about the state of nature except that it is in some set $K = \{K_1, K_2, ..., K_n\}$, (*decision making under uncertainty*). Nevertheless, in order to deal with this defined uncertainty, a four decision strategies was suggested [24]:

1. *Pessimistic:* is selecting the worst possible outcome, C^*
2. *Optimistic:* is selecting the best possible outcome, C_*
3. *Hurwitz:* selecting the value $\alpha = [0, 1]$ for each alternative a weighted average of the best payoffs C^* and the worst C_* is taken, $MaxH = \alpha C^*_A + (1 - \alpha)C_{*A}$.
4. *Normative:* averaging the payoffs across n outcomes for each alternative and then se-

lecting the best one.

In this work we defined special keywords to deal with this uncertainty. (See `mentalstate()` in Table 1.)

2 Language Syntax and Semantics

2.1 Keywords

In this work, we used First Order Logic (FOL) [4] to represent the knowledge and to perform inference as shown in the coming sections. Table 1 depicts some of the basic keywords that formalize the actions, interactions and learning among a society of agents. The possible values if these keywords are declared in the declaration part in advance. We give a detailed explanation about the meaning of these keywords herewith.

- `agent[(name)]`:
  This keyword helps define two kinds of possible agents present in the environment. An agent with an identified type as in `dealer(A)` and another one with an anonymous type `agent(A)`.
- `action(subject [agent])`:
  Identifies the possible actions that can be performed in the environment. The action represents the change in the environment as in `lower, higher`. And the `subject` is the physical entity which is subject to the change. 'Dealer agent acts to change the price' of a commodity can be represented as `lower(price c)`. In a case where the doer of the action is not identified the action can be represented as `lower(price)` which means a change in the environment.
- `state(subject [agent])`:
  This keyword can be used to define the possible *states* of both the environment and the agents. For example in order to define the situation in which a given agent is an enemy we could write `enemy(c)` and describe situation of the market stock or the stock of a given agent we can write `low(stock)` and `low(stock c)` respectively.
- `signal([action()] [state()])`:
  This is either an action of a given agent as in `signal(lower(price c))` or a state as in `signal(stock(down))`.
- `ask(subject [agent])`:
  This is a communication keyword and it is used to ask friendly agents to pass the information about other agents. If the `[agent]` is omitted the command will have a role of broadcasting, i.e., asking all the agents.
- `profile(agent action(subject)`^{number}`)*`:
  The profile of agents can be identified by this notion that allows to define a sequence of actions performed by a given agent and the number of repetitions.
- `strategy(agent)`:
  This command is designed to name the strategy with which the player is playing. Whether it is *revealing*, *partially revealing* or *non-revealing* strategy.
- `mentalstate([action] [state] [type] [group])`:
  The mental state represents the degree of certainty maintained by the agent about certain alternatives. Especially when the agent is faced with expressing an assessment about actions and states. `mentalstate` can take attributes like `opti` (optimistic) and `pessi` (pessimistic).
- `lie(action())`:
  Lying maybe needed when the action is performed to fool enemy agents with some false impression. This happens usually by taking an action that in reality does not make a physical change in the environment, especially in the private environment. This command is mostly used within the command `bluff()`.
- `bluff(agent lie())`:
  Bluffing an agent is giving it a wrong impression with the help of the command `lie()`.
- `spy(agent ask() state())`:
  A way of collecting information about an agents with the help of another spying agent (a cooperating friendly agent). The `agent()` is asked by command `ask()` to communicate the state of a given agent.
- `type(agent())`:
  This keyword is used to identify the agents' type, i.e., customer, search, catalog, dealer, manufacturer, delivery and banker. For example, `dealer(c),customer(d)`.

Table 1: *The keywords are defined to satisfy the social and learning needs of agents present in multi-agent environment.* [] *is for optionality and* * *is for repetition.*

Keywords	Example
agent([name])	agent(c),broker(A), customer(k)
action(subject [agent])	lower(price c), sale(books)
state(subject [agent])	enemy(c), low(stock c)
signal([action()] [state()])	signal(lower(price c))
ask(subject[agent])	ask(stock b), ask(stock dealer())
profile(agent action(subject)^{number}*)	profile(c lower(price)^2)
strategy(agent)	revealing(c)
mentalstate([action()] [state()] [type()] [group()])	opti(lower(price c)), pessi(enemy(c))
lie(action())	lie(lower(price))
bluff(agent lie())	bluff(c lie(lower(price)))
spy(agent ask() state())	spy(c ask(stock) low(stock b))
type(agent())	dealer(c),customer(e)
group(agent())	enemy(c),competitive(e)
zone(action*)	friendly(high stable), hostile(sale)
classify([case] [fact] [plan] [sig])	classify(sig), classify(Fact)

- group(agent()):
  There may be some pre-defined group of agents. This keyword is used to identify to which group the agent belong to. For example, enemy(c).
- zone(action*):
  There are three security zones, hostile, friendly and competitive. The action* (where * is for repetition) are actions that characterize these zones. For example the zone hostile is characterized with actions lower, sale. According to the behaviors of agents, they can be classified in one the zones.
- classify([case] [fact] [plan] [sig]):
  This keyword helps implement learning method by returning the characteristics of environmental patterns. (See example in Section ?? for further explanation about this keyword.)

2.2 Syntax and Semantics

We defined the syntax of the language using Backus Naur Form (BNF), described in Table 2. The program consists of major declaration part started by the DECL: keyword (see example in Sec. 3) that indicates the declaration of knowledge format and knowledge types of given games. A background knowledge, BGK, of the games of which the name is indicated by an alphabetic string (game name), consisted of cases and facts that should follow. The items of BGK representing the expertise (the knowledge about the environments) are atomic sentences taking the forms <case> and <fact>.

The RESCAN keyword allowing the online execution, sensing, and cases matching of the signals and BGK cases, respectively. The learning process is reached by the use of the <assess> to indicate what triggers the learning and an applied

learning strategy. The structure of the <case> symbolizes three major things, the situation of the environment and/or the description of any eventual problem at a given time, symbolized by the ENVIRON. The solution of the problem or the suggested actions to the particular environment state, symbolized by GSOL and finally the changes that occur on the background knowledge after this solution using UPDATE. Each of these three parts might or might not be followed by a processing. Accordingly, the background knowledge is increasingly consistent and the agent is learning. The SIMILAR keyword is to symbolizes the matching of new cases.

3 An Implementation Example

An important area of implementation is the Electronic Commerce (EC). EC was defined by [11] as a society of seven agents that operate together to curry out the main interactions on the WWW to accomplish the physical EC, interact, compete and negotiate with each other (Fig.1). The agents are namely *dealer, manufacturer, customer, delivery, bank, catalog* and *search.*

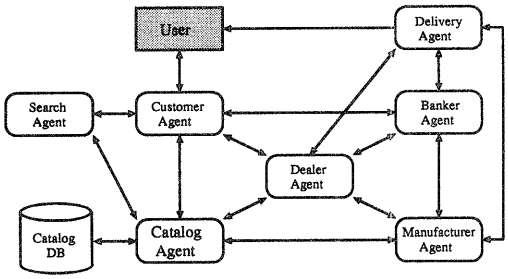

Figure 1: *7 types of EC agents in Ex-W-Pert Project.*

The Dealer, manufacturer and delivery agents are agent which are naturally in competition with each other attempting to attract more customers. For the sake of this competitive situation let us imagine the situation in which the knowledge of the environment (the market), plus, the signals of dealer agent A is sending, the agent B tries to learn the environment, A's strategy. Obviously, agent A is omniscient about it's own strategy and given that it's utility (benefit) is higher than B, we can say that the game is zero-sum (since the benefit of A is the loss of B) and B by analyzing the signals that A gives will be getting some hints. Agent A can play with any possible strategy, e.g., *play partially revealing* to make his opponent get the least indicating signals. On the other side, B in this situation would construct his own plan through a process of explanation and analysis of the information revealed by A and the information it basically knows about the environment.

```
–beginingof code–
DECL: GENV(price game)
 AGENT(NAME) A B C D
 ACTION(SUBJECT AGENT) lower higher sale
 STATE(SUBJECT AGENT) low high stable
 ZONE(ACTION) Friendly Competitive Hostile
 FRIENDLY(ACTION) up increase
 COPETITIVE(ACTION) stable
 HOSTILE(ACTION) lower down
 MENTALSTATE(ACTION STATE TYPE) opti pessi
 GROUP(AGENT) friendly competitive enemy
BGK(price game)
 <fact> agent(A) agent(B) agent(C) agent(D)
 <fact> profile(B higher(price)3 lower(price)1)
 <fact> profile(C higher(price)3 lower(price)1)
 <fact> profile(C stock(lower)1)
 <fact> high(stock C) unknown(stock A)
 <fact> enemy(D) friendly(B) opti(enemy C)
 <case> GENVIRON higher(price agent)
        GSOL enemy(agent)
        UPDATE <fact>> enemy(agent) low(price)
 <case> GENVIRON low(stock agent) enemy(agent)
        GSOL opti(enemy agent) ask(low price))
        UPDATE <fact>> opti(enemy(agent))
RESCANSIG(price game)
 <sig> signal(lower(price D))
 <sig> signal(low(stock C))
 <sig> signal(margin(up C))
 <sig> signal(higher(price B))
RETRIEVE(price game)
 <case> GENVIRON higher(price agent)
        GSOL enemy(agent)
        RESULT <fact>> enemy(agent) low(price>
ASSESS(price game)
 <ASSESS> IF NOT(SIMILAR(RESCAN(<sig>)))
        THEN IF SCANCASE( IDENTFY(<sig>))
        THEN <CASE>> GENVRON(<sig>)
              GSOL (opti(<sig>>))
              UPDATE (<fact>> opti(<sig>))
– end of code –
```

Table 2: *The syntax of MALL expressed by BNF.*

```
⟨program⟩::=      {DECL:-GENV(<game name>)<declaration>*}*
                  {BGK(<game name>){'⟨fact⟩'|'⟨case⟩'}*<processing>}*
                  {RESCAN[<game-environ>](<game name>)'⟨plan⟩'|'⟨sig⟩'|
                  {'⟨case⟩'|'⟨fact⟩'}<processing>}*
                  <ASSESS>*
<processing>::=   <statement>*
<statement>::=    <envStatement>
<envStatement>::= <declaration>|<case>|string
<declaration>::=  <atomicSentence>
<atomicSentence>::=<FunctionSymbol>
<FunctionSymbol>::=<keyword>
<keyword>::=      alph([alph])|alph(alph [alph])|
                  alph({[alph(alph [alph])]|[alph(alph [alph])]})|
                  profile(alph alph(alph Num)*)|alph(alph)|
                  alph({[alph(alph [alph])]|[alph(alph [alph])|
                  [alph(alph [alph])]})|
                  bluff(alph alph(alph alph(alph)))|
                  spying(alph ask(alph [alph]))|ask(aplh [alph]))|
                  aplh(alph*)
                  classify(⟨case⟩)|⟨case⟩|⟨⟨case⟩|⟨fact⟩⟩|⟨fact⟩|⟨⟨fact⟩|⟨plan⟩⟩|
                  ⟨plan⟩|⟨⟨plan⟩|⟨sig⟩⟩|⟨sig⟩|⟨⟨sig⟩⟩
<case>::=         ⟨case⟩ GENVIRON <atomicSentence>* GSOL <atomicSentence>*
                  UPDATE <atomicSentence>*
<ASSESS>::=       {if <learnStrategy>* then <learnStrategy>*}*
<learnStrategy>::=<processing>*
<game name>::=    alph
<game-environ>::= PLAN|SIG|CASE
```

The meta symbols alph *and* Num *stand for alphabetic and numeric respectively.*

4 Conclusion

A new action, interaction and learning language was defined for agents present in multi-agent environment. The learning is enforced by the continuous monitoring of the behaviors of both, agents and the environment. The language is suitable for a large variety of applications, and represent a suitable frameworks for situations where uncertainty, noise and hostility are major issues. It is being tested in Electronic Commerce's (EC) multi-agent applications, specifically in, dealer agents learning each other's strategies and interacting on the World Wide Web.

References

[1] M.F. Agraham, "Modern Sociological Theory", *Oxford University Press,* (1982).

[2] N. Adam, and Y. Yesha (Edts.), "Electronic Commerce: Current Research Issues and Application", *Springer Verlag, 155 p.,* (1996).

[3] J.L. Austin, "How to do things with words", *Harvard University Press, Cambridge, MA,* (1962).

[4] J. Barwise, and J. Etchemendy, "The Language of First Order Logic, Tarski's World",

Center for the Study of Language and Information, (1990).

[5] A.H. Bond and L. Gasser, (Edts). "Analysis of Problems and research in DAI", *Readings in Distributed Artificial Intelligence, pp. 3-35,* (1988).

[6] J.M. Bradshaw (Edt.), "Software Agents", *MIT Press, 480 p.,* (1997).

[7] J. G. Carbonell, and Y. Gill, "Learning By Experimentation", *Fourth Int. Workshop on Machine Learning, pp. 256-266,* (1987).

[8] D. Wilkins, "Readings in Knowledge Acquisition and Learning", *Automating the Constructions and Improvement of Expert systems, B.G. Buchanan and D.C. Wilkins, Edts.,* (1993).

[9] Encyclopedia of Artificial Intelligence "Learning, Machine", *John Wiley & Sons, Inc., vol. I, p. 799,* (1992).

[10] B.H. Far, J. Nagayama, and Z. Koono, "A WWW Based Intelligent Search Engine for Electronic Commerce" *Joint Conference on Knowledge Based Software Engineering, JCKBSE' 96, pp. 174-181,* (1996).

[11] B.H. Far, H. Mukai, Zenya Koono, "Intelligent Agents for Electronic Commerce", *JSAI-97, pp. 482-485,* (1997).

[12] G. Zlotkin, and J.S. Rosenschein, "Negotiation and Task Sharing in Cooperative Domains", *Proc., 11th Int. Joint Conf. on Artificial Intelligence, pp. 912-917 Detroit, MI,* (1989).

[13] J.C. Harsanyi, "Games with Incomplete Information Played by Bayesian Players", *Parts I, II, III. Management Science, vol. 14, no. 3,5,7,* (1967 & 1968).

[14] M.N. Huhns, and M.P. Singh, "Readings in AGENTS" *Morgan Kaufmann Publishers,* (1997).

[15] M.B. Morgan (Edt.), "Machine Learning, A Multistrategy Approach", *Volume IV,* (1994).

[16] D. Ourston, and J.R. Mooney, "Changing the rules: A comprehensive Rule to Theory of Reinforcement", *Eighth National Conference on Artificial Intelligence, pp. 815-820,* (1990).

[17] E. Plaza, J.L. Acros, and F. Martin, "Cooperation Modes among Case-based Reasoning Agents", *Proc. ECAI'96 Workshop on Learning in Distributed Artificial Intelligence Systems,* (1996).

[18] R. Pyle (Edt.), "Special Issue on Electronic Commerce on the Internet", *Communications of The ACM, vol. 39, no. 6,* (1996).

[19] J.S. Rosenschien, "Rational Interaction: Cooperation Among Intelligent Agents", *PhD. Thesis, Computer Sciences Department, Stanford University, Stanford CA,* (1985).

[20] S.J. Russell, and P. Morvig, "Artificial Intelligence: A Modern Approach" *Printice Hall Inc.,* (1995).

[21] T. Sandholm, and R. Crites, " Multiagent Reinforcement Learning in the Iterated Prisoner's Dilemma", *Biosystems 37: 147-166, Special Issue on the Prisoner's Dilemma,* (1995).

[22] Y. Shoham, "Agent-Oriented Programming", *Artificial Intelligence, vol. 60 no. 1, pp. 51-94,* (1993).

[23] K.J. Engimann, "Decision Making With belief Structures: An Application in Risk Management", *International Journal of Uncertainty Fuzziness and Knowledge based Systems, vol. 4, no. 1, pp. 1-25,* (1996).

[24] R.R. Yager, "On Ordered Weighted Averaging Aggregation Operators in Multi-criteria Decision Making", *IEEE Transactions on Systems, Man and Cybernetics, vol. 18, pp. 83-190,* (1988).

[25] J. Kolodner, "Case-Based Reasoning ", *Morgan Kaufman Publishers* ISBN 1 558 237 2, Copyright 1993.

[26] D.B. Leake, "Case-Based Reasoning, Experiences, Lessones and Future Directions ", *AAAI Press/ The MIT Press* ISBN 0 262 62110 X, Copyright 1996.

[27] F. Gebhardt et. Al., "Reasoning With Complex Cases", *Kluwer Academic Publishers* ISBN 0 7923 9882 3, Copyright 1993.