社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技法
TECHNICAL REPORT OF IEICE.
AI-52 (1998-11)

電子商取引エージェント用推論と学習環境

ベルーズ H. ファー

〒 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. 048-858-9612,　Fax. 048-858-3716
E-mail. far@cit.ics.saitama-u.ac.jp
http://www.cit.ics.saitama-u.ac.jp/~far/

あらまし:　小人数チームを対象にした分散情報処理システム (グループウエア) が多く存在するが多数のユーザーを中心とするシステムはまだ発表されていません。本研究は, 多数のメンバーから構成されるネットワーク上の分散型情報処理環境 (グループウエア) のモデルとそのパイロット版を構築する。様々な応用例は考えられますが, 電子商取引システムを実験対象とする。システム科学論からは，複数のオブジェクトを対象にするマスエントロピー法と Bayesian ネットワーク，人工知能及びソフトウェア工学からの応用技術は，階層構造を持った分散型知識ベースシステム，ゲーム理論，及び分散型オブジェクトである。ネットワークからの応用技術は，ドキュメントオブジェクトモデル (DOM)，ネットワークセキュリティーなどである。

和文キーワード　**ソフトウェアエージェント**, *WWW*, **エキスパートシステム**, **グループウェア**, **不確実情報管理**

Integrated Reasoning and Learning Environment for Electronic Commerce Agents

Behrouz H. Far

Department of Information and Computer Science,
Faculty of Engineering, Saitama University
255 Shimo-okubo, Urawa 338-8570, Saitama, Japan
Tel. +81-48-858-9612,　Fax. +81-48-858-3716
E-mail. far@cit.ics.saitama-u.ac.jp
http://www.cit.ics.saitama-u.ac.jp/~far/

Abstract:　Although there are many projects focusing on multi-agent systems, there are only a few focusing on the knowledge sharing. Furthermore, there is no project aiming at knowledge sharing and groupware system with a very large number of actors (i.e., agents or human counterparts). The internet, in general, and electronic commerce, in particular, are examples of such systems. In this project, we introduce a model and devise an implementation for the large scale knowledge sharing using mass entrophy, Bayesian networks, game theory, document object model (DOM) and network security. The agent model for the system components blends the traditional expert systems' reasoning engine with a multi-layer knowledge base, communication and documentation engines. We test the ideas via implementing electronic commerce (EC) agents.

Key words　*Software Agent, WWW, Expert system, Groupware, Uncertainty Management*

1 Introduction

There are many active projects focusing on multi-user groupware systems and multi-agent environment [7, 17] and various techniques have been introduced, such as cooperation [24], coordination [1], interoperability [15], open agent architecture (KAoS) [8], and Agent Oriented Programming (AOP) [26]. However, there are only a few projects focusing on the knowledge sharing. Furthermore, there is no project aiming at knowledge sharing and groupware systems with a very large number of actors (i.e., agents or human counterparts). The internet, in general, and electronic commerce, in particular, are examples of such systems [6].

In this project, we introduce a model and devise an implementation for the massive knowledge sharing using mass entropy method, game theory, Bayesian networks, document object model (DOM), network security and propose an agent model for the system components that blends the traditional expert systems' reasoning engine with a multi-layer knowledge base, communication and documentation engines.

Electronic Commerce (EC) is a potential application for such a system. EC is viewed as a society of software agents, such as customer, search, catalog, manufacturer, dealer, delivery and banker agents, distributed over the Internet and interact and negotiate with each other. Each EC agents is viewed as an expert in it's own field. It has a knowledge-base and a reasoning engine, a communication engine and a documentation engine. The knowledge-base is organized in three layers: skill layer, rule layer and knowledge layer (S-R-K layers). For each EC agent, we identify the class of problems to be solved and build the knowledge base gradually for each layer. We believe that using this multi-layer knowledge base system will speed up the reasoning and ultimately reduce the operation costs.

2 Knowledge Sharing Agent

Figure 1 depicts the basic structural components of the knowledge sharing agent, called an *Ex-W-Pert Agent*. Similar to conventional expert systems, each agent has a *knowledge-base* and a *reasoning engine*. Compared to the conventional expert systems, a main difference is that all agents have an additional *communication engine* and a *documentation engine*. The communication and documentation engines facilitate communication and navigation on the internet. Each agent also has a user friendly interface unit to communicate with a human counterpart.

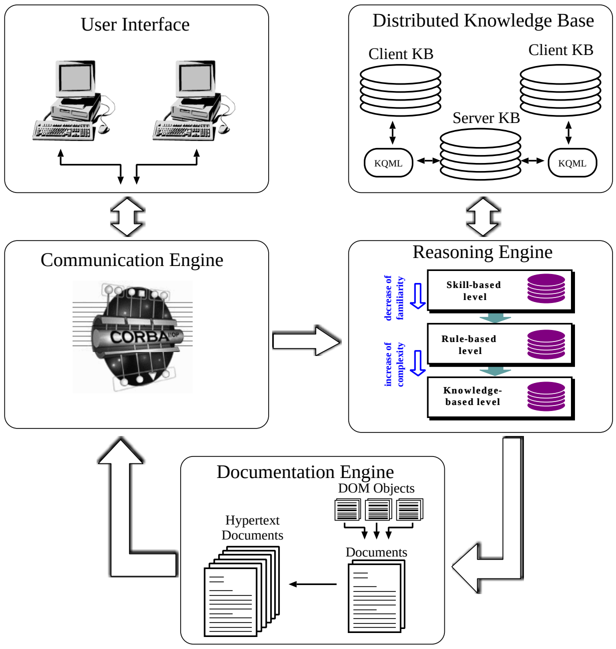

Figure 1: The knowledge-sharing agent

2.1 Reasoning engine and knowledge base

Human experts when engaged with a goal-oriented task, try to achieve the goal within the constraints imposed by the task and avoid the *cognitive overload* through selective utilization of their knowledge.

It is believed that human experts possess a conceptual (mental reference) model of how the objects in the external world interact based on standard operating procedures. Conceptual models have a hierarchical structure defined best by the Skill-Rule-Knowledge (S-R-K) levels [23] concerning with the routine, innovative and creative problem solving tasks, respectively.

2.1.1 Skill-based level:

This level deals with the *routine* tasks. It denotes the kind of task for which problem solving knowledge and strategies are well defined. At this level, reasoning is governed by stored patterns of predefined rules. Such context specific patterns are called *rules-of-thumb* (or symptomatic rules), that map directly from an observation to a ready-made solution.

At this level a query of an agent is accepted and by searching the knowledge-base, proper immediate action is selected. For instance, in case of a search agent, the query comes in the form of a list of keywords, submitted by the customer agent. Then search agent finds related keywords and conducts search using the new set of keywords.

2.1.2 Rule-based level:

This level deals with the *innovative* tasks when dealing with familiar or similar problems. It denotes the kind of task for which problem solving knowledge is well defined. Rule based behavior is conventionally described by case bases, decision tables, digraphs, fuzzy sets and natural language models [18].

At this level a query of an agent is accepted and a case data base is consulted to determine the action. For example, in case of a search agent, the initial query comes in the form of a sentence with reduced and restricted grammar [12]. This is adopted to avoid unnecessary overload of natural language processing. Then a set of similar cases are searched and cases matching the needs of the user are retrieved. Further search is conducted based on the instructions recorded on the matched cases.

2.1.3 Knowledge-based level:

This level deals with the *creative* tasks for which common patterns in stored knowledge form do not exist and reasoning should start from the so called *first principles*, starting from problem identification. In other words, neither problem solving knowledge nor the strategy is well defined.

At this level a query is accepted and the agent uses a KQML-CORBA based communication protocol (see section 4) to consult with the other agent to determine the proper action. For example, a search is conducted by consulting a catalog agent, which in turn, contacts the dealer and/or manufacturer agents for proper information.

2.2 Communication and documentation engines

knowledge-sharing agents must communicate with other agents in order to work flexibly and autonomously. Here we have considered building Java-based, KQML (Knowledge Query and Manipulation Language [20]) speaking and CORBA (Common Object Request Broker Architecture [9]) enabled agents that communicate over the Internet. The communication engine is mainly responsible for maintaining connection to the network, communicating with other agents and managing messages. The communication protocol is discussed in Section 4.2.

Documentation engine performs three main tasks:

- Acquiring data from the other agents, as requested by the reasoning and communication engines.
- Preparing and reformatting data items to be appropriate for transferring over the internet.
- Maintaining index of data items transferred and/or retrieved.

Using documentation and communication engines, the local knowledge can be shared with the other agents.

3 Agent Model of EC

Conventional EC models are built upon functional components, such as *commodity information*, *ordering*, *payment* and *electronic brokerage* [22]. *Commodity information* includes a showcase of goods and services offered, mainly in the form of a simple *home page* or a complicated electronic shopping mall. This is the heart of EC and must include interactive catalogs and directories. Electronic catalog system offers more flexibility as compared to the conventional catalog repository and can answer to search requests. *WWW based ordering system* includes a mechanism for placing and collecting orders, processing and/or distributing them. *Payment system* features a fail-safe mechanism for exchanging goods for electronic money. Finally, *electronic brokerage* involves special mechanism for offering a more efficient or safer service, and various add-on tasks such as bidding and bargaining.

In this project, EC is viewed as a society of software agents that interact and negotiate with each other. We have devised 7 types of EC agents, namely: *customer*, *search*, *catalog*, *manufacturer*, *dealer*, *banker* and *delivery* agents (see Fig. 2).

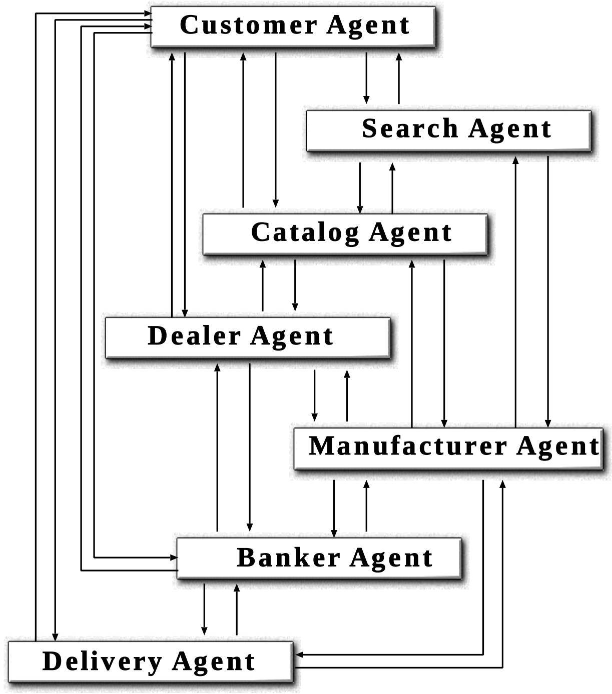

Figure 2: Society of EC Agent

Each EC agent is an expert in its own field and may interact with its human counterpart or behave autonomously. For instance, a *customer agent* receives instruction from a user to search the WWW space, to find a catalog and dealer for a desired product and to sign a contract. A *dealer agent*, on the other hand, identifies a customer's needs and takes parts in bidding to win the contract. A *delivery agent* competes to get a shipment contract.

Some distinguishing features of project are summarized below [13].

- Generating and customizing the menu items of the user interface based on users' utility model, in *Customer* agent, using Bayesian networks.

- The ability of off-line query processing, and using case-based knowledge to direct the search process, in *Search* agent. Together with a custom reporting facility that allows a user select the form and structure of the search report [12].

- Knowledge based guidance through the use of a case-based classifier, in *Catalog* agent.

- Formulating the Pickup and Delivery Problem (PDP) using integer programming, and devising an algorithm to handle PDP, in *delivery* agent.

- Dealing with uncertain and nondeterministic business practice, such as bidding and competing to sell a product, in *Dealer* and *Manufacturer* agents, using game theoretic techniques [25].

Concerning portability and modular design of the agents, we have used Java programming language to implement the functions and make use of HTTP and IIOP protocols for message based connections.

At present, the *Customer*, *Search*, *Catalog*, *Dealer* and *Delivery* agents are fully implemented. The *Manufacturer* and *Banker* agents are implemented partially.

4 Cooperation and Knowledge Sharing

4.1 Formal ontologies

Knowledge sharing, data integration, knowledge inter-operation, and reuse require a formal description of a representational vocabulary, including classes, relations and functions, which is called *ontology* [14]. Ontologies are considered as ways of specifying content-specific agreements for the sharing and reuse of knowledge [16].

In this project we have developed ontologies for both knowledge bases (e.g., *common-task* and *agency* ontologies) and commodities (e.g., *hardware_products* and *software_products*) that we experiment with. The *agency* and *common-task* ontologies describe inter-agent and intra-agent concepts, respectively. (See Section 4.3 for an example.)

4.2 Communication protocol

The EC agents interact based on a 3-step protocol composed of *agent identification*, *query pro-*

cessing and *payment processing*, which is novel to this project. In the agent identification step, necessary data for identifying an agent is submitted and acknowledged. In query processing step the kind and contents of the required service is specified and in the payment processing step the price and payment method is negotiated.

The above mentioned protocol is handled by a message management mechanism. An incoming message is added to the end of the queue and the message handler decides upon a proper process by consulting the reasoning engine and appropriate layer of the knowledge base.

Concerning portability, modularity, security and transaction handling issues, the HTTP and IIOP protocols are selected for message based connections. HTTP is used for Web based connections and IIOP allows establishes the client-server relationships, remote method invocation and integration of a wide variety of objects across a network that EC agents may require their services.

4.3 Knowledge sharing protocol

Knowledge sharing over the Internet is done via using KQML [20]. KQML is a semantics level communication language that is commonly used in knowledge sharing and agent communication projects [5].

We have developed a socket based and an object based communication API for KQML messages. The socket based one, similar to other projects, allows passing KQML messages encoded in ASCII form. The object based API for Java objects allows, first, simplifying the programming using command objects that take care of the communication issues in an abstract way, and second, as the API is installed on each machine, communication proceeds in the same manner whether or not an agent is on the same machine or on a different machine.

Here is a simple example illustrating the communication and knowledge sharing procedure. In this example, the *agency* and *hardware-products* ontologies (described in Section 4.1) are used.

Suppose that a *Customer* agent, say *Agent-a*, needs some additional information for a given product. In the first step, *Agent-a* asks for the information from an agent that most likely has such information, in this case, a *Catalog* agent *Agent-b* that *Agent-a* has already had it recorded in the friendly agents list. (See Section 4.4 for security zones.)

If *Agent-b* does not have that data and this request fails, it first informs *Agent-a* of the consequences. Then *Agent-b* passes a query to the naming server to find a proper agent that has such data. The name server, returns a list of the agents that may have the requested data and the security zone they belong to. Based on the answer of the naming server, *Agent-b* decides to ask *Agent-c* to acquire the data. Upon a proper answer from *Agent-c*, *Agent-b* can fulfill the original request of *Agent-a* by delivering this info to *Agent-a* and terminating the transaction.

Upon accepting the request, at the first place, *Agent-b* continues to search and check until the goal of the *Agent-a*, and consequently the user, is satisfied. Note that, considering user friendliness, only part of the process is visible to *Agent-a* and the user. They need not be aware of all the details of the communications and negotiations that *Agent-b* has gone through.

4.4 Security

Basically, the security service of CORBA's common services is adopted to build fine grained locally secure transaction and business processes.

Furthermore, we have devised three security zones for the agents to operate, i.e., *friendly*, *trustable* and *hostile* [25]. Each agents can operate in any of the security zones when dealing with the other agents. In the *friendly* zone the information is exchanged freely. In the *trustable* zone check and verification steps are added using public coding and decoding keys [10].

Agents competing for scare resources may apply individual strategies to increase their own utilities, therefore they may operate in the *hostile* zone. In this project we have used game theoretic and machine learning techniques to design a multi-agent learning language (MALL) [25] to handle this case.

For instance, a dealer applying a strategy with lower cost and better services to the customers is more likely to increase its own income. For the purpose of this example, let us say that given the basic knowledge of the environment (the market), plus, the signals of dealer agent X is sending, the dealer agent Y tries to learn X's strategy. We can say that the game is zero-sum (since the benefit of X is the loss of Y). Then Y by analyzing the signals that X gives will get some hints. Agent X can be said to *play partially revealing* to make its opponents get the least indicating signals [29]. On the other hand, agent Y in this case, would construct its own plan through a process of explanation and analysis of the information revealed by agent X.

The MALL language is currently bundled with the *Dealer* and *Manufacturer* agents when competing against each other by applying individual strategies attempting to increase their own *utilities*. Details of the language and implementation are given in [25].

5 Other Works and Discussion

5.1 Knowledge representation and reasoning concerns

In a multi-agent system using the Internet, dealing with novel situations in an innovative or creative way is a necessity. There are already some representations and models suggested to deal with innovation and creativity. For example, reinterpretation, evaluation, assimilation, strategic control [28]. The initial formulation of the problem to be solved may be incomplete, contradictory and under- constrained. The process of understanding and formulating the problem is reinterpretation. Evaluation is the process of weighting the alternative solutions. Assimilation is a means for calling relevant solution alternatives, and strategic control is defined as breaking typically rigid control structure and allow more interaction among processes.

Based on the idea of interacting the adaptation knowledge with the other knowledge sources, we have made a kind of network in which the nodes are the constraints which represent a new problem to be solved. We make the link between two nodes if there is any adaptation knowledge which may inform us having one of the constraint in the problem description allows us to adapt the solution in the way that satisfy the second constraint. The direction of each link shows that adaptation in which direction is possible. This allows us to use all adaptation knowledge available, combine them, and depict them in the network. Therefore, having a new constraint we can move through the network regardless of which knowledge we are using and find possible interpretation for the new problem [27].

5.2 Electronic commerce concerns

Some current issues of EC and representative projects are mentioned in [2, 3], and [22]. The generic EC model may be viewed as a set of *actors* using *media* performing business *actions*. The *actors* are providers (dealers, merchants, etc.), and consumers (users, information seekers). The *media* is composed of multimedia objects (documents, images, software, etc.) which represent goods or services. The *actions* are business processes, such as stock management and payment processing.

Almost all of the projects mentioned in [2] and [22], concentrate on either of the *actors*, *media* or *actions*. They mainly focus on identification and implementation of EC functions and business processes. At this moment, there is no proposed model other than ours to cover the whole set of *actors*, *media* and *actions*. Furthermore, in the other projects, the consumers are assumed to have a wide variety of domain expertise [3]. We, on the contrary, assume that the customers are ordinary and casual users and provide them with tools and techniques, borrowed from AI, to share the expertise with them. We think that implementing and applying AI techniques to EC is a new and challenging task for both fields. In this project we propose application of the Ex-W-Pert system architecture [11, 12] for such implementation, that is, adding the communication and documentation engines to the conventional expert systems to help the *actors* and put the burden of knowledge processing on the system rather than the *actor* itself.

The idea of EC, by means of Electronic Data Interchange (EDI) and private or local area networks (LAN) has been around for some years. What internet based EC adds to this is changing from private networks to a global network together with interactive data interchange, with a reasonable cost, but at the expense of lower security. Moving to a global network implies less control over the kind and amount of data available. We have proposed a simple but powerful solution to the security problem by devising the security zones.

5.3 Developing tools

There are already some agent developing tools available, mostly implemented in Java programming language, such as Agent Building Environment (ABE), Java Agent Template (JAT), Java Expert System Shell (JESS), and enhanced with Object Request Broker (ORB), suitable for distributed object platforms, such as Voyager and Open Agent Architecture (OAA), JKQML [19], AgentBuilder [4], etc. Although such tools and environments can facilitate the communication (for example, using KQML language in JAT as the knowledge level communication) and message passing (for example, using HTTP and IIOP protocols in ABE and OAA, respectively), but they fail to provide the appropriate aids to build extensive knowledge bases in the sense we use in this project.

One exception is Open Knowledge Base Connectivity (OKBC) [21] for accessing knowledge bases stored in knowledge representation systems using a model of knowledge representation systems based on a common conceptualization of classes, individuals, slots, facets, and inheritance. The protocol supports both networked and direct access to knowledge bases. However, OKBC is mostly an interface to the knowledge based system and does not help building the knowledge base itself in the multi-layer fashion described in Section 2.1.

6 Conclusion

In this paper, a general model for knowledge sharing agents was introduced and the ways of implementing intelligence into a Internet based agents was discussed. An architecture for Internet based Electronic Commerce (EC) agents was demonstrated. Present issues and implementation status of agents of the EC family were discussed.

References

[1] M.F. Abraham, "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] N. Adam, and Y. Yesha, "Strategic Directions in Electronic Commerce and Digital Libraries: Towards a Digital Agora", *ACM Computing Surveys*, vol. 28, no. 4, pp. 818-835, (1996).

[4] `http://www.agentbuilder.com`

[5] D. Benech, and T. Desprats, "A KQML-CORBA based Architecture for Intelligent Agents Communication in Cooperative Service and Network Management", *Proc. IFIP/IEEE MMNS'97*, (1997).

[6] T. Berners-Lee, R. Cailliau, H.F. Nielsen, and A. Secret, "The World-Wide Web", *Communications of The ACM*, vol. 37, no. 8, pp. 76-82, (1994).

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

[8] J.M. Bradshaw, S. Dutfield, P. Benoit, and J.D. Wooley, "KAoS: Toward An Industrial-Strength Open Agent Architecture", *Software Agents*, J.M. Bradshaw (Edt.), pp. 375-418, *MIT Press*, (1997).

[9] "The Common Object Request Broker: Architecture and Specification", *Object Management Group (OMG)*, Revision 2.2, (1998).

[10] Dorothy Denning, "Protecting Public Keys and Signature Keys", *IEEE Computer*, (1983).

[11] B.H. Far, and Z. Koono, "Ex-W-Pert System: A Web-Based Distributed Expert System for Groupware Design", *World Congress on Expert Systems' 96*, pp. 545-552, (1996).

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

[13] B.H. Far, S.O. Soueina, H. Hajji, S. Saniepour, and A.H. Hashimoto, "An Integrated Reasoning and Learning Environment for WWW Based Software Agents for

Electronic Commerce", **電子情報通信学会 英文論文誌（*D*）知能ソフトウェア工学特集号**, *98* **年** *12* **月号 – 採録** (1998).

[14] A. Farquhar, R. Fikes, and J. Rice, "The Ontolingua Server: A Tool for Collaborative Ontology Construction", *Knowledge Systems Laboratory*, KSL-96-26, (1996).

[15] M.R. Genesereth, "An Agent Based Framework for Interoperability", *Software Agents*, J.M. Bradshaw (Edt.), pp.317-345, *MIT Press*, (1997).

[16] T.R. Gruber, "Toward Principles for The Design of Ontologies Used for Knowledge Sharing", *Formal Ontology in Conceptual Analysis and Knowledge Representation*, N. Guarino and R. Poli, edts., *Kluwer Academic Publishers*, (1993).

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

[18] G. Johannsen, "Categories of Human Operator Behavior in Fault Management Situations", *Tasks, Errors and Mental Models*, L.P. Goodstein, H.B. Andersen, and S.E. Olsen, Edts., *Taylor and Francis*, ch. 16, pp. 251-258, (1988).

[19] `http://www.alphaWorks.ibm.com/formula/jkqml`

[20] "Draft Specification of the KQML Agent Communication Language", *The DARPA Knowledge Sharing Initiative*, (1993).

[21] `http://www.ai.sri.com/~okbc/`

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

[23] J. Rasmussen, "The Role of Hierarchical Knowledge Representation in Decision Making and System Management", *IEEE Transactions on System, Man and Cybernetics*, vol. SMC-15, no. 2, pp. 234-243, Mar./Apr., (1985).

[24] J.S. Rosenschien, "Rational Interaction: Cooperation Among Intelligent Agents", PhD. Dissertation, Computer Sciences Dept., Stanford University, Stanford CA, (1985).

[25] Sidi O. Soueina, B.H. Far, T. Katsube, and Z. Koono, "MALL: A Multi Agents Learning Language for Competitive and Uncertain Environment", **電子情報通信学会 英文論文誌（*D*）知能ソフトウェア工学特集号**, *98* **年** *12* **月号 – 採録** (1998).

[26] Y. Shoham, "Agent-Oriented Programming", *Artificial Intelligence*, vol. 60, pp. 51-94, (1993).

[27] S. Saniepour, H. Hajji, S.O. Soueina, H. Hashimoto, and B.H. Far, "CBR in Design: Towards more creative system using adaptation knowledge", *1998 JSAI Annual Conference*, pp. 154-155, (1998).

[28] L.M. Wills and J.L. Kolodner, "Towards More Creative Case Based Design Systems", *Case Based Reasoning Experiences, Lessons, and Future Directions*, D. Leake (edt.), *MIT Press*, (1996).

[29] S. Zamir, "Repeated Games Of Incomplete Information", *Handbook of Game Theory*, Volume I, Chapter IV, (1992).