

**1999年度**

**人工知能学会全国大会(第13回)論文集**

Proceedings of the 13th Annual Conference of JSAI, 1999

発行日 1999年6月15日

発行所 社団法人 **人工知能学会**

東京都新宿区津久戸町4-7 〒162-0821

OSビル402 ☎(03)5261-3401

## Towards an Agency Theory of an Agent Population

Behrouz H. Far  
ベルーズ H. ファーFaculty of Engineering, Saitama University  
埼玉大学情報システム工学科

**Abstract:** Although there are many projects focusing on multi-agent systems, there are only a few focusing on systematic design of large scale multi-agent system. Electronic commerce is an example of such systems. In this project, we introduce a model and devise an implementation for the large scale knowledge sharing using Bayesian networks, game theory 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. 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.

## 1 Introduction

Scalability is a central issue in multi-agent systems research. Although there are many projects focusing on multi-agent systems and groupware, it is quite hard to scale them up to include a very large number of actors (i.e., agents and/or human counterparts).

Reuse is another fundamental issue. The once designed agent configurations are to be used in another configurations constantly and repeatedly. Existing research projects do not specify how their proposed agent technology can be used for systematic design of large scale multi-agent systems.

Still other issues, such as dependency, flexibility, performance and evolution remain.

The goal of this research is to formalize the concept of agents and agency and devise standard building blocks for "agents" (AG) and certain patterns for "generalized agencies" (GAG) in order to use them effectively in a large scale system design.

AG agents are Java-based, KQML (Knowledge Query and Manipulation Language) speaking and CORBA (Common Object Request Broker Architecture) enabled.

Each GAG describes a problem, such as a business process that happens repeatedly, and describes the participant agents as well as the process towards the solution to the problem.

For each AG a basic configuration is specified and a basic building block for participating agents in a GAG is devised. Furthermore, for each GAG, alternative configurations, decisions and trade-offs are defined.

This paper describes the current status of this ongoing research project.

## 2 Generalized Agent (AG)

An Agent Reference Model, defining a framework within which agents exist and operate is specified by FIPA (<http://www.fipa.org>). It includes Agent Communication Channel (ACC), Directory Facilitator (DF), Agent Management System (AMS), Agent Platform Security Manager (APSM) and Internal Platform Message Transport (IPMT). It does not specify however the internal structure of the agents. The internal structure is specified in our project.

Fig. 1 shows the internal structure of GA agents. Similar to conventional expert systems, each agent has a *local 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*.

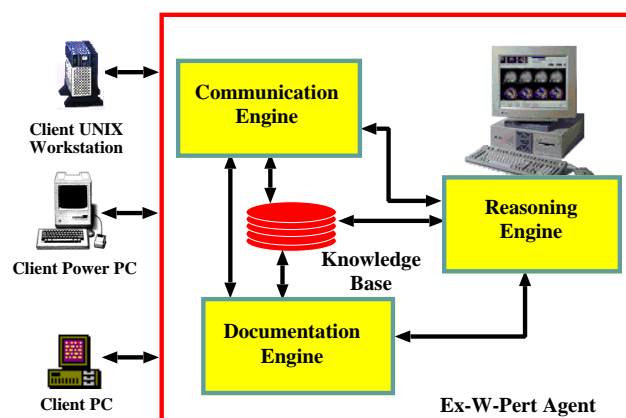


Figure 1: Internal Structure of a Generalized Agent

The reasoning engine includes procedures that the AG agent performs. Each procedure is triggered upon receiving a message from another agent. Each proce-

連絡先: Behrouz H. Far 〒 338-8570 埼玉県浦和市下大久保 255, 埼玉大学工学部情報システム工学科, Tel. 048-858-9612.  
E-mail. far@cit.ics.saitama-u.a.c.jp  
WWW. <http://www.cit.ics.saitama-u.ac.jp/~far/>

ture has a name, input parameters and the outputs. These together are called a *signature* and the set of all signatures of an AG is its *interface*. What is seen from outside is only AG's interface. How the job is actually done internally, is a matter personal to the agent itself.

The communication and documentation engines facilitate handling messages and run procedures. The communication engine is mainly responsible for maintaining connection to the network, communicating with other agents and managing messages. Documentation engine performs tasks, such as, acquiring data from the other agents, as requested by the reasoning and communication engines; and preparing and reformatting data items to be appropriate for transferring over the network.

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

### 3 Generalized Agency (GAG)

*Generalized agency* (GAG) describes a particular problem, such as a business process. GAG is a description of communicating and cooperating agents, which serve as the building blocks of the agency and customized to solve a general problem described by the GAG itself. For each GAG, the participating agents, their roles, collaborations and distribution of responsibilities are defined.

GAG is defined via the followings: a name, that is used to label it; intention, describing what goal it is supposed to fulfill; a list and description of participant agents; motivation, that describes the problem and how the participant agents solve the problem; a description of collaboration among the participants; a critique, describing whether the GAG can achieve its intended goal based on motivation and participant agents; and finally, a list of close or related GAGs and a description of their similarities and differences.

A library of highly customizable GAGs is under development.

#### 3.1 Knowledge sharing issues

Knowledge sharing, data integration, knowledge interoperation, and reuse requires a kind of *ontology* [2]. An ontology gives meanings to messages within a given domain language. In order for a message from one agent to be properly understood by another, the agents must ascribe the same meaning to the constants used in the message. Thus, the ontology performs the function of mapping a given constant to some well-understood meaning. AGs first discover whether or not they share a mutual understanding of the domain constants before further message passing.

#### 3.2 Security issues

Basically, the security service of CORBA's common services is adopted to build fine grained locally secure procedures. Furthermore, we have devised three security zones for the agents to operate, i.e., *friendly*, *trustable* and *hostile* [3]. 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.

Agents competing for scarce 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) [3] to handle this case.

### 3.3 Implementation issues

The choice of programming language is quite important. Flexibility, extedibility and interoperability are three main factors to select proper implementation language and platform.

There are already some agent developing tools available, mostly implemented in Java programming language, such as ABE, JAT and OAA [1]. 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) (<http://www.ai.sri.com/~okbc/>) for accessing knowledge bases stored in knowledge representation systems using a model of knowledge representation systems. However, OKBC is mostly an interface to the knowledge based system and does not help building the knowledge base itself.

## 4 Application and Conclusion

Electronic Commerce (EC) is a potential application area and test bed for GAGs. In this project, EC is viewed as a society of software agents, such as customer, search, catalog, manufacturer, dealer, delivery and banker agents. The EC agents interact based on a 3-step protocol composed of *agent identification*, *query processing* and *payment processing* [1].

Other applications such as multi-agent Computer Aided Instruction (CAI) system, distributed fault diagnosis system and Object Oriented CASE tool are under construction.

## References

- [1] B.H. Far, et al., "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), Vol.E81-D No.12, pp.1374-1386, (1998).
- [2] A. Farquhar, et al., "The Ontolingua Server: A Tool for Collaborative Ontology Construction", *Knowledge Systems Laboratory*, KSL-96-26, (1996).
- [3] Sidi O. Soueina, et al., "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), Vol.E81-D No.12, pp.1339-1349, (1998).