行政院國家科學委員會專題研究計畫成果報告

自動化商談協定： 注入協商原則於代理者談話協定中
Automated Negotiation Protocols: Embedded the Negotiation Policies into the Agent Conversation Protocols

計畫編號： NSC 87-2213-E-004-005
執行期限： 86/08/01-87/07/31

主持人：胡毓忠 國立政治大學資訊科學系
電郵： jong@cs.nccu.edu.tw
網址： http://www.cs.nccu.edu.tw/~jong

中文摘要

軟體代理者一般可以區分為： 人機介面代理者，常駐式代理者， 移動式代理者。 這些軟體代理者可以用層次的結構來表示。常駐式代理者可以視為智慧型代理者，它具有通訊，規劃，目標啟動及協商原則等數個模組。這些模組將會適當的分佈在常駐式代理者本身的各層。另一方面移動式代理者是可攜性高的移動程式碼，它讓代理者的合作及協商更具有效率。移動式代理者或許沒法攜帶所有的協商協定及原則於一身但是遠方的代理者系統將可以提供相同的服務模組而不損此移動式代理者的協商能力。對於軟體代理者每一層功能我們將用物件導向的模板技術來設計。

關鍵詞： 軟體代理者， 自動化商談協定，常駐式代理者， 移動式代理者

Abstract

The software agent can be classified as interface agents, stationary agent, and mobile agent. All of them can be modeled as a layer structure. Stationary agent is regarded as intelligence agent with complete set of communication, planning, goal activation and negotiation policy modules. These modules are suitably distributed on different layer of stationary agent. On the other hand, mobile agent is portable mobile code which allows the agent′s cooperation and negotiation more effective. Mobile agent might not carry all of the agent's negotiation protocols and policies but the remote agent system can provide similar modules services without reducing the mobile agent negotiation capabilities. Finally, we use object-oriented pattern design technique for each layer of software agent.

Keywords: Software Agents, Automated Negotiation Protocols, Stationary Agents, Mobile Agents

2. Project Objectives and Background

The aim of this research is to design a generalized and highly efficient generic model for executable agent negotiation protocols. DAI (Distributed Artificial Intelligence) addresses the problem of designing the open multi-agent systems, which provides agents to interact and cooperate effectively[Hewi91]. Agent's coordination does not necessarily result in perfect cooperation. Therefore, negotiation in multi-agent environments is quite common[Rose93][Rose94] [Sand95] [Zlot96]. There are several techniques for solving agent's negotiation problems. Game theory is suitable for small number of self-motivated agent's negotiation while operations research techniques are allowed medium number of cooperative agent's negotiation[Krau97].

The high level of agent's negotiation protocols can not be really operated in the Internet environment without the well-formatted executable communication protocols. Therefore, this research is to set up executable agent negotiation protocols, which can truly reflect the higher level agent negotiation's policies. Finally, in order to show the robustness of this generic negotiation model, we apply this negotiation model into Internet based securities trading electronic commerce.

A stationary agent is a computer system, *situated* in some environment, which is capable of *flexible autonomous* action in order to meet its design objectives. There are three key concepts in their definition: *siutatedness, autonomy* and *flexibility*. Situatedness means that the agent receives sensory input from its environment and that it can perform actions which change the environment in some way. Autonomy means that agent should be able to act without the direct intervention of humans (or other agents), and that it should have control over its own actions and internal state. Flexibility means that the agent is embedded with responsive, pro-active, and social characteristics[Jenn98]. On the other hand, a mobile agent does not emphasize its intelligent but require its mobility and security[Karn98][Thor97].

3. A Generic Framework for Agent and Agent System

A generic framework for agent and agent system is a fundamental agent internal architecture for any kinds of agent enable applications[[Brad97][Brad98] [Mull96]. Before we set up the agent negotiation model, we must define what is the agent internal structure. In DAI viewpoint, agent architecture can be classified as deliberative agent, reactive agent, and interacting agent. Recently, the hybrid agent with layered agent architecture is the most promising approach[Mull96]. So we adopt this architecture as our stationary agent. In DAI viewpoint, the agent system is embedded in the intelligent agent so there is no clear separation between agent and agent system concepts. Our stationary agent is modeled as 5 control layers: interaction layer, conversation layer, cooperation layer, negotiation layer, and application layer. Each layer is designed via object-oriented system patterns that will resolve the interoperability problem between heterogeneous agent system[Busc96] [Gamm95][Kend97]. Besides, there are several static data/knowledge bases for these agent control layers to store their associated communication protocol sets, policies, and management mechanisms, etc.

Similarly, the mobile agent can be defined as stationary agent's 5 control layers structure but without those associated static data/knowledge bases. This is due to the lightweight feature of the mobile agent. The mobile agents are executed in the agent system, which provides the management services for both stationary agents and mobile agents.

4. Discussion

The software agent is a multi-threaded active object, which is actor for concurrent object-oriented programming [Agha86] [Agha87] [Kafu98][Guess98]. The asynchronous message passing between peer communication agents is meta-token object. This meta-token object includes the meta information from each agent's different control layer. Interaction layer provides the sender/receiver agent's identity and the associated speech-act performative requested from the above agent control layers. The meta-token with speech-act perofrmative is a typed message which enrich the agent communication protocols[Sear69][Sear71] [Fini94]. Conversation layer is responsible for handling the chaining of each separate interaction protocols and make sure the entire agent conversation is meaningful and smooth. Besides the agent's conversation layer sometimes has to deal with the agent's security communication issues. Cooperation layer is designed for each agent to examine whether its goal (or plan) is achieved in the right direction. The negotiation layer is to handle the negotiation process and conflict resolution problems. Finally, the application layer is embedded the agent enable application's criteria and parameters. Each layer is modeled via object-oriented design patterns and is configured with some initial parameters once the agent is created.

Both the stationary agents and mobile agents can proceed the negotiation protocols with their peer communication agents. Due to the performance considerations, sometimes we might use mobile agents to execute the negotiation protocols. But we must solve the mobile agent's negotiation protocols and policies relocation problem. It is always infeasible for mobile agent to carry all of its negotiation protocols and policies modules from its original host because of the mobile agent lightweight characteristics. So we allow the mobile agent initiates the same modules of negotiation protocols and policies from the remote peer agent system. Thus, mobile agent must carry some of negotiation criteria while using the remote site standard negotiation protocols and policies modules. In case the mobile agent has to talk to its original host′s stationary agent to supply shortage information and criteria, the authentication mechanisms must provide to certify the mobile agent′s identity.

This generic framework agent-based software system was applied to Electronic Commerce(EC): Internet security trading system. Because the negotiation strategy for each agent is fixed in this EC example, we did not fully demonstrate the robustness of our generic agent negotiation framework.

5. Self Evaluation

The agents and objects relationship was clarified in this research. Active object (or actor) comes quite close to the concept of autonomous agents. The agent can be composed of objects component and the characteristics of agents can be expressed in terms of object patterns. In order to set up the automated negotiation protocols, we found the generic agent framework has to be built first. The layer agent architecture is the promising framework for agent developer to consider. At this moment, we did not fully implement our 5 agent control layers but the research is still going on. In the near future, we are expecting an executable agent negotiation protocols can be built. Finally, we also integrate the intelligent agent concepts and mobile agent technology to make the agent system framework more robust and flexible. These results are going to submit to the international well-known journal.

6. Conclusion

The robustness of the agent negotiation protocols must be implemented in the existing Internet environment. Even lots of agent negotiation strategies were studied none of them are really the executable agent negotiation protocols. We found the generic agent framework must be designed before the agent negotiation protocols were specified. The layer agent framework is the recent trend for agent architecture design besides each layer expressed in terms of object oriented design patterns is one of the most promising research directions. The marriage of intelligent agent concepts and mobile agent technology can make the agent services more flexible and robust.

References

[Agha86]Agha, Gul et al., *ACTORS:A Model of Concurrent Computation in Distributed Systems*, The MIT Press:

Cambridge, MA, 1986.

[Agha87]Agha, G and C. Hewitt, Concurrent Programming Using Actors, In *Object-Oriented Concurrent Programming*, Edited by Akinori Yonezawa and Mario Tokoro, MIT Press, 1987, pp. 37-53.

[Arno96]Arnold, Ken and Gosling, James (1996), *The Java Programming Language*, Addison-Wesley.

[Brad97]Bradshaw, M. Jeffrey, et al., KAoS: Toward An Industrial-Strength Open Agent Architecture, *Software Agents*, AAAI Press/The MIT Press, 1997.

[Brad98]Bradshaw, M. Jeffrey, et al., Mediating Representations for an Agent Design Toolkit, *Proceedings of the Eleventh Annual Knowledge Acquisition Workshop*, Banff, Canada, 18-23, Vol. I, SIS-8:1-21, April 1998.

[Busc96]Buschmann, Frank, R. Meunier, H. Rohnert, P. Sommerlad, and M. Stal, *Pattern-Oriented Software Architecture: A System of Patterns*, Wiley and Sons, 1996.

[Fini94]Finin, T., McKay, D., McEntire, R., KQML as an Agent Communication Language, *The Proceedings of the Third International Conference on Information and Knowledge Management*, (CIKM'94), ACM Press, 1994.

[Gamm95]Gamma, Erich, Richard Helm, Ralph Johnson, John Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software,* Addison-Wesley, 1995.

[Gass91]Gasser, Les, Social Conceptions of Knowledge and Action: DAI Foundations and Open Systems Semantics, *Artificial Intelligence* 47, 1991, pp. 107-138.

[Guess98]Guessoum, Z. and J.-P. Briot, From Active Objects to Autonomous Agents, *LIP6 Research Report, 1998/015*, Laboratoire d'Informatique de Paris 6 Paris 6 Univ.-CNRS, Paris, 1998; ftp://ftp.lip6.fr/lip6/reports/1998/lip6.1998.015.ps.gz.

[Jenn98]Jennings, R. Nicholas, Katia Sycara, and Michael Wooldridge, A Roadmap of Agent Research and Development, *Autonomous Agents and Multi-Agent Systems*, 1, 1998, pp. 7-38.

[Hewi91]Hewitt, Carl, Open Information Systems Semantics for Distributed Artificial Intelligence, *Artificial Intelligence*, 47, 1991, pp. 79-106.

[Hu95a]Hu, Y. J., Agent Communication Protocols for Automatic Services and Resources Allocation, *Technical Report NSC 83-0408-004-001 and NSC 84-2213-004-002*, Taipei, Taiwan, 1995.

[Hu95b]Hu, Y. J., Intelligent Autonomous Resources Allocation in Distributed Open Multi-Agent System, *International Workshop on Decentralized Intelligent* *and Multi-Agent Systems, DIMAS'95*, Cracow, Poland, 1995, pp. 435-442.

[Hu96a]Hu, Y. J., Cooperation, Negotiation, and Conflict Resolution Protocols for Automatic Services and Resources Allocation, *Position Paper, Proceeding of Seventh European Workshop on Modelling Autonomous Agents in a Multi-Agent World, MAAMAW'96*, Eindhoven, Netherlands, 1996.

[Hu96b]Hu, Y. J., Interaction Protocols for Intelligent Agents in Distributed Services Environment, *Proceeding of IFIP/IEEE International Conference on Distributed Platforms, ICDP'96*}, Dresden, Germany, 1996.

[Kafu98]Kafura, Dennis and Jean-Pierre Briot, Guest Editors' Introduction: Actors and Agents, *IEEE Concurrency,* Vol. 6, No. 2, April-June 1998, pp. 24-29.

[Karn98]Karnik, M. Neeran and Anand R. Tripathi, Design Issues in Mobile-Agent Programming System, *IEEE Concurrency*, Vol. 6, No. 2, April-June 1998, pp. 52-61.

[Kend97]Kendall, A. Elizabeth, et al., Multiagent system design based on object-oriented patterns, *Journal of Object-Oriented Programming*, June 1997, pp. 41-47.

[Krau97]Kraus, Sarit, Negotiation and cooperation in multi-agent environments, *Artificial Intelligence,* 94, 1997, pp. 79-97.

[Lea97]Lea, Doug, *Concurrent Programming in Java: Design Principles and Patterns*, The Java Series, Addison Wesley, 1997.

[Mull96] Muller, P. Jorg, *The Design of Intelligent Agents: A Layered Approach,* Springer-Verlag, 1996.

[Oliv96]Oliver, R. Jim, On Automated Negotiation and Electronic Commerce, http://opim.wharton.upenn.edu/~oliver27/dissertation/toc.html.

[Oust98]Ousterhount, K. John, Scripting: High Level Programming for the 21st Century, *IEEE Computer,* Vol. 31, No. 3, March 1998, pp. 23-30.

[Pico95][Picot, Arnold, Christine Bortenlaenger and Heiner Roehrl, The Automation of Capital Markets, *Journal of Computer-Mediated Communication*, Vol. 1, No. 3, Dec. 1995.

[Raif82]Raiffa, Howard, {\em The Art and Science of Negotiation}, Harvard University Press, Cambridge, MA, 1982.

[Rose93]Rosenschein, Jeffrey S. and Zlotkin, Gilad, Consenting Agents: Designing Conventions for Automated Negotiation," {\em IJCAI93}, Chambery, France, 1993.

[Rose94]Rosenschein, Jeffrey S. and Zlotkin, Gilad, *Rules of Encounter:*

Designing Conventions for Automated Negotiation Among Computers, MIT Press, 1994.

[Sand95]Sandholm, Tuomas and Lesser, Victor, Issues in Automated Negotiation and Electronic Commerce: Extending the Contract Net Framework, *First International Conference on Multiagent Systems*, San Franscisco, 1995.

[Sear69]Searle, J. R., *Speech Acts,* Cambridge University Press, Cambridge, UK, 1969.

[Sear71]Searle, J. R., *The Philosophy of Language,* Oxford University Press, 1971.

[Shoh93]Shoham, Y., Agent-Oriented Programming, *Artificial Intelligence*, 60, Elsevier Science Publishers, 1993, pp. 51-92.

[Thor97]Thorn, Tommy, Programming Languages for Mobile Code, *ACM Computing Surveys*, Vol. 29, No. 3, September 1997, pp. 213-239.

[Zlot96]Zlotkin, G. and Jeffrey S. Rosenschein, Mechanisms for Automated Negotiation in State Oriented Domains, *Journal of Artificial Intelligent Research*, 5, 1996, pp. 163-238.