Learning in Broker Agent

Xiaocheng Luan, **Yun Peng**, and **Timothy Finin**

Department of Computer Science and Electrical Engineering, University of Maryland Baltimore County
1000 Hilltop Circle
Baltimore, MD 21250
{xluan1, ypeng, finin}@cs.umbc.edu

One of the common ways to achieve interoperability among the autonomous agents is to use a broker agent (or a facilitator). The idea is, on the one hand, individual agents can advertise their capabilities to the broker agent; on the other hand, an agent can also ask the broker agent which agent(s) has certain capabilities (ask for recommendation). Simple broker agents provide match-making services based on the capability information volunteered by individual agents and the (recommendation) request. The problem is, even with a very good agent capability description language and a powerful match-making mechanism (such as LARKS), if the actual capability information volunteered by each individual agent is not accurate, it won't be of much help. Given that the autonomous agents might be written by different people, at different time, and for different purpose, this is likely to occur. For example, two Toyota car dealers (A and B) might give the same advertisement "We sell all models of Toyota cars, lowest price guaranteed". The fact might be that at dealer A, you have fewer or no choice of colors on popular models as you can at the other dealer. The service quality is different.

This work is an attempt to solve such problems by incorporating learning into broker agents so that the broker agents can capture more accurate information about the capabilities of individual agents. This is a result of an intuitive observation in the real world: don't be fooled by the beautiful words and colorful pictures in the commercials, consult a consumer report. In analogy, the broker tries to build agent consumer reports through learning. One of the learning methods we propose is based on concept-generalization and concept-specialization. In this learning method, we assume that the broker agent has limited domain knowledge, in the form of domain ontology, which is organized into tree-like class hierarchy. Based on this domain hierarchy, the broker agent can build service information trees. The capability information of an agent is associated with the appropriate nodes of the trees. Then, the broker agent continuously refines its knowledge about these agents through the interactions with them, and possibly by observing the interactions among the agents. The broker can learn the capability information from the following channels: advertisements from individual agents, interrogating/testing an agent about its capabilities, feedback for previous recommendations (volunteered or requested), and possibly the "past experience" of some agents and the (recommendation) requests from individual agents. One of the insights behind this learning method is the law of locality. The more frequently an aspect of an agent's capability is referenced, the more accurately the capability information around this aspect will be captured by the broker (through learning), and thus better overall system performance could be achieved.

We are also exploring the collaboration of multiple brokers. This includes the distribution of agent capability information over the brokers, collaborative learning, and knowledge sharing. With collaborative brokering, the bottle neck problem and the single point of failure problem could be avoided.

By incorporating learning techniques, a broker agent would be able to capture more accurate capability information of individual agents, build "agent consumer reports", and can thus help achieve more effective cooperation among agents.

References

Gruber, T. R. 1991. The Role of Common Ontology in Achieving Sharable, Reusable Knowledge Bases. In Allen, J. A., Fikes, R., and Sandewall, E. (Eds). Principles of Knowledge Representation and Reasoning: *Proceedings of the Second International Conference.* San Mateo, CA: Morgan KaufMann.

Lenzmann, B. and Wachsmuth, I. 1997. Contract Net Based Learning in a User-Adaptive Interface Agency. In Weiss, G., editor (*LNAI* 1221), *Distributed Artificial Intelligence Meets Machine Learning*, pages 223-241. Springer Verlag.

Weiss, G. 1995. Adaptation and Learning in Multi-Agent Systems: Some Remarks and a Bibliography. In Weiss, G. and Sen, S. editors (*LNAI* 1042), *Adaptation and Learning in Multi-Agent Systems*, pages 1-21. Springer Verlag.

http://www.cs.cmu.edu/~softagents/interop/matchmaking.html