Multi-agent System Development: Design, Runtime, and Analysis

K. S. Barber, J. Ahn, K. Fullam, T. Graser, N. Gujral, D. C. Han, D. N. Lam, R. McKay, J. Park, M. Vanzin

The Laboratory for Intelligent Processes and Systems
Electrical and Computer Engineering
The University of Texas at Austin
{barber, jsahn, kfullam, graser, ngujral, dhan, dnlam, rmckay, jisun, vanzin}@lips.utexas.edu

Introduction

Dynamic and unexpected events are defining characteristics of numerous application domains. These environments often require decision-makers to solve many problems with insufficient time and resources. To effectively assess options, decision-makers require situation analysis and decision-support tools that model the dynamism of these environments to make rapid, robust decisions. Autonomous agents and multi-agent systems satisfy these requirements for decision-making in dynamic environments.

Developing a multi-agent system (MAS) is a challenging task, considering sophisticated agent interactions and uncertain environmental dynamics and domain requirements. This demonstration addresses the comprehensive development process for multi-agent systems; illustrating tools for the initial design of the agent system, the capabilities encoded in the individual agents, and analysis tools that enhance developer comprehension of system behavior.

Design Tools

Two design activities supported by this research are categorization and comparison of agent technologies under a common ontology and construction of agent architectures through the selection of technologies that fulfill the designer's requirements. Developing agent systems requires the proper selection of agent technologies where selection is based on adherence to the agent architecture structure and satisfaction of domain and installation requirements. MAS designers are guided by specific desired "competencies" (capabilities) in the context of a particular domain, for the entire system and/or a particular agent. Thus, agent technologies (e.g., belief revision, organization, planning, etc.) are developed or selected for a MAS by considering their application to a particular domain and their ability to fulfill desired competencies.

The Technology Portfolio Manager (TPM) allows a designer to view and compare agent technologies with respect to both competencies provided and domains supported (Barber et al. 2004). The TPM is populated with four sets of information and the relationships between these sets: agent competencies, agent infrastructure services, domain tasks in a particular domain (in this case UAV surveillance) and agent technologies. Furthermore, the tool allows the designer to interrogate the repository from the perspective of: (1) technologies (displaying related domain tasks and competencies satisfied by the selected technologies), (2) domain tasks (displaying related competencies and technologies capable of delivering selected domain tasks), and (3) agent competencies (displaying related domain tasks and technologies capable of satisfying the selected competencies). The TPM provides the designer with a tool for browsing a repository of agent technology specifications allowing the designer to evaluate how well different technologies map to the desired agent competencies or infrastructure services.

The Designer's Agent Creation and Analysis Toolkit (DACAT) aids the designer in constructing agent designs based on the agent competencies (Barber and Lam, 2003b). Metrics such as component complexity and cohesion are prescribed to evaluate and compare agent designs based on agent competencies. Tasks and task dependencies resulting from inclusion of respective competencies in the agent architecture (design) can be analyzed and improved. This type of structural analysis can be performed during the architecting process to assist in forming agent components, during development (when more dependency details are known), or as a method to survey existing agent architectures.

Runtime – Agent Competencies

This demonstration also illustrates some novel methods (technologies loaded in TPM) for fulfilling agent competencies, such as belief maintenance and action selection, in the domain of UAV surveillance of military targets.

Accurate beliefs are required for an agent to correctly select actions. This demonstration presents a belief revision algorithm (Fullam 2003) based on information valuation "policies" accounting for information source reliability, information corroboration, information source certainty, and information timeliness. An algorithm based on policies for information valuation provides justification for derived beliefs by identifying influential policies as reasons why more valuable information is given priority over less valuable information. This research makes a major contribution toward producing quality beliefs for decision-making by examining information quality assessment in terms of a number of factors, not only information source reliability. In addition, basing information valuation on logical policies promotes traceability in algorithm construction. Justification for derived beliefs is provided, a useful feature for any domain in which humans are the ultimate decision-making authority.

In addition to the underlying belief revision algorithm, the organization of the agents affects which beliefs are adopted or revised. To maintain autonomy, an agent should control formation of its beliefs. The degree of an agent's belief autonomy is its degree of dependence on others to build its beliefs, which can be controlled by selecting the most appropriate agents on which to rely. Large, open environments make finding the most appropriate information sources difficult since agents may only know about a limited number of sources and sources may enter and leave the system. Finding partners can be facilitated by decomposing the agents in a system into small groups so that the scale and openness can be efficiently managed within the group. The partner-finding process benefits from trustworthiness information with respect information quality assurance, increasing the agents' goal achievement probability (Barber and Park, 2004).

Agents should behave in a rational manner, meaning an agent should perform actions that enable it to achieve its goals. Action selection is complicated by the existence of multiple goals of different priorities and interaction with other agents. In order to account for all these factors in the UAV demonstration, action selection is performed in a decision-theoretic fashion. Goal priorities are represented in the expected rewards obtained through goal achievement while agent coordination is performed through transformations on the reward and cost structure the agent uses in its action selection process (Barber and Han, 2003).

Analysis Tools

Software comprehension is the analysis of the implementation and extraction of a design, which ideally should be equivalent to the original design. In addition to extracting a design, software comprehension for MAS involves understanding the behaviors of the constituent agents with respect to their individual goals. Knowing the reasons for agent behaviors in the implementation is important for the development of MAS because those behaviors must agree with designed behaviors. The Tracing Method can be used to validate the implemented behaviors against the designed behaviors as well as facilitating agent software comprehension (Barber and Lam, 2003a).

The Tracer tool aims to automate the developer's task of analyzing run-time data and relating it to models of agent structure and behavior. The method involves logging agent behavior during execution and associating the log entries and run-time data with agent concepts in the design models. As a result, the agent concepts (e.g., beliefs about the current state of the environment) are instantiated with run-time data. Explanations of run-time agent behavior can be generated using abductive reasoning on this connection between the implementation and design models.

Agent software comprehension can be performed as soon as the initial implementation is done to verify that the agents are performing basic tasks correctly. During and after implementation, agent comprehension can be used to help debug the implemented MAS and to insure that the agents are behaving as expected and for the right reasons.

Acknowledgements

This research was funded in part by the Defense Advanced Research Projects Agency and Air Force Research Laboratory, Air Force Materiel Command, USAF, under agreement number F30602-00-2-0588. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed on implied, of the Defense Advanced Research Projects Agency (DARPA), the Air Force Research Laboratory, or the U.S. Government.

References

Barber, K. S. and Han, D. C. 2003. *Decision-Theoretic Action Selection for Interacting Planning Agents*, Technical Report, TR2003-UT-LIPS-003, Laboratory for Intelligent Processes and Systems, The University of Texas at Austin.

Barber, K. S.; Ahn, J; Gujral, N; Lam, D. N.; and Graser, T. 2004. *Technology Portfolio Manager*, Technical Report, TR2004-UT-LIPS-007, Laboratory for Intelligent Processes and Systems, The University of Texas at Austin.

Barber, K. S. and Lam, D. N. 2003a. Enabling Abductive Reasoning for Agent Software Comprehension. In *Proceedings of the IJCAI-03 Workshop on Agents and Automated Reasoning*, 7-13.

Barber, K. S. and Lam, D. N. 2003b. Specifying and Analyzing Agent Architectures using the Agent Competency Framework. In *Proceedings of the 15th International Conference in Software Engineering and Knowledge Engineering*, 232-239.

Barber, K. S. and Park, J. 2004. Finding Partners to Form Information Sharing Networks in Open Multi-Agent Systems. In *Proceedings of the The 17th Florida Artificial Intelligence Research Symposium*. Forthcoming.

Fullam, K. (2003) An Expressive Belief Revision Framework Based on Information Valuation, Master's Thesis, The University of Texas at Austin.