Design, Runtime, and Analysis of Multi-Agent Systems

K. S. Barber, N. Gujral, J. Ahn, D. DeAngelis, K. Fullam, D. C. Han, D. N. Lam, J. Park
The Laboratory for Intelligent Processes and Systems
Electrical and Computer Engineering
The University of Texas at Austin

{barber, ngujral, jsahn, dave, kfullam, dhan, dnlam, jisun}@lips.utexas.edu

1. INTRODUCTION

Developing a multi-agent system (MAS) is a challenging task, considering sophisticated agent interactions, uncertain environmental conditions, and dynamic domain requirements. This set of demonstrations offers tools for the design and analysis of agent systems and implementations. The design tools help to quickly select agent functionality, assign functionality to components, and then evaluate agent technologies for potential inclusion in the agent design. Search and evaluation is based on the ability of candidate technologies to deliver required functionality and satisfy deployment constraints (e.g. interoperability with database choice, operation on selected OS, etc). The analysis tool, Tracer, is used to validate the implemented behaviors against the designed behaviors and can be used for software comprehension. Sample technologies illustrated in this work include novel work in belief revision, information source selection, and action selection. These technologies are encoded in each agent to form a reliable information network and to support agents' action-selection in an Unmanned Aerial Vehicle (UAV) target tracking simulation.

2. DESIGN TOOLS

Developing agent systems requires the proper selection of agent technologies based on functional data. MAS designers are guided by specific desired *agent competencies* in the context of a particular domain. Thus, agent technologies (e.g., belief revision algorithms, coordination algorithms, planners, etc.) are developed or selected for a MAS by considering their ability to fulfill desired competencies.

This research is sponsored in part by the Defense Advanced Research Project Agency (DARPA) Taskable Agent Software Kit (TASK) program, F30602-00-2-0588. 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 or implied, of DARPA.

The Technology Portfolio Manager (TPM[1]) allows a designer to view and compare agent technologies with respect to both competencies provided and domains supported [2]. For this demonstration, the TPM has been populated with the DARPA TASK Agent Technology Repository, which is a collection of agent technology specifications acquired and represented by The University of Texas at Austin. The idea of the Designer's Agent Creation and Analysis Toolkit (DACAT) is to aid the designer in quickly creating an agent architecture given a selected set of agent technologies, considering software metrics such as coupling and cohesion among agent components [4]. Leveraging the technology repository in the TPM and the competency-based agent architecture from DACAT, the Application Architecture Creation and Evaluation Toolkit (ACET) aids the designer in building an agent-based system by selecting appropriate agent technologies according to their coverage of and compliance to both the functional and structural requirements prescribed by the competency-based agent architecture [1]. The Implementation Architecture Creation and Evaluation Toolkit (ICET) aids the architect to efficiently deploy agent technologies (suggested by ACET) based on adherence to infrastructure constraints such as the ability of technologies to integrate and interoperate with one another and their ability to deploy at specific sites.

The TPM, DACAT, ACET and ICET are useful to multi-agent system designers, for constructing agent-based system by selecting and assembling agent technology components written at different times by various developers. Specifically, this research offers methods and tools that assist the designer in comparing and selecting various agent technologies to construct and evaluate agent-based systems. Whereas DESIRE [6] and MaSE [7] offer agent-oriented software engineering methodologies that include creating agent architectures, DACAT offers the designer a practical tool for creating custom agent architectures from selected agent technologies while considering functional dependencies among components [1]. TPM, ACET, and ICET are highly related to and inspired by SEPA 3D Architecture [3].

These design tools have been applied to a limited set of application domains, future work will expand their application.

3. ANALYSIS TOOL

Software comprehension (understanding software structure and behavior) is essential for developing, maintaining, and improving software. 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. During and after implementation, agent

[1] TPM is copyrighted in 2005 by Automation Through Software.

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. With the aid of the Tracer Tool, many of the manual tasks, such as scanning for unexpected behavior, are automated. The research on software comprehension combines and extends ideas of empirical analysis from reverse engineering and abstraction from model-checking. The Tracer tool assumes the user inserts logging statements in correct locations in the code. Also, it does not perform an exhaustive state space search, but instead verifies behaviors in tested scenarios.

4. RUNTIME – AGENT COMPETENCIES

This demonstration also illustrates some novel agent technologies (also specified in the TPM) for fulfilling agent competencies in the domain of UAV surveillance: belief maintenance, information source selection, and action selection. For the following demonstrations and the design tools discussed above, the main infrastructure requirement is the availability of version 1.4 of the Java Runtime Environment (JRE).

Accurate beliefs are required for an agent to correctly select actions. This demonstration presents a belief revision algorithm [8] based on information valuation "policies" accounting for information source reliability, information corroboration, information source certainty, and information timeliness. Thus, it helps human decision-makers deal with overwhelming amounts of information. It also demonstrates recommended decisions by agents and their certainty on those decisions in real-time.

In addition to the underlying belief revision algorithm, agent interaction dependencies among agents affect which beliefs are adopted or revised. Dynamic environments result in unreliable information quality, unpredictable changes in network topology, and changes in information requirements. Therefore, it is not a simple task to select the set of information sources to request or receive information from [5]. The Topology viewer shows the dynamic information source selection and evaluation. It is intended for visualizing the information source selection process by the different users based on interest. It shows which information sources are selected based on the level of trustworthiness. Since the trustworthiness is represented by different colors and the selection is represented by lines, user can easily identify the logic behind the information source selection. The Topology viewer can be improved by incorporating coverage and cost factor visualization as the current version only considers trustworthiness.

Rational action means an agent should perform actions in the agent's best interests. This demonstration is intended for researchers interested in planning and action selection in response to changing objectives and agent interactions. Multiple goals and goal level interactions among the agents are handled in a decision-theoretic fashion for the application of UAV Surveillance. Macro actions are used to abstract domain behavior into task-level behavior allowing agents to perform trade-off reasoning about the value of each target during target selection [9]. The domains addressed by this research are restricted to those with independent goals. In the worst case, computation is exponential in the number of goals an agent is considering. This research is related to work on "over-subscription planning" [10], where agents select and plan for a subset of their goals. When the goal-set gets large (increasing the number of targets), bounded reasoning and value estimation heuristics are also used to speed calculation at the cost of sub-optimal behavior.

5. SUMMARY

In the design phase, the Technology Portfolio Manager (TPM) allows a designer to view and compare agent technologies with respect to both competencies list provided and domains (e.g. Unmanned Aerial Vehicles) supported. The Designer's Agent Creation and Analysis Toolkit (DACAT) aids the designer in constructing competency-based agent architectures. The Application Architecture Creation and Evaluation Toolkit (ACET) aids the designer in building an agent-based system by selecting appropriate agent technologies. The Implementation Architecture Creation and Evaluation Toolkit (ICET) aids the architect in helping to plan and efficiently deploy agent technologies. In the analysis phase, 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.

Three novel agent technologies are also demonstrated in the UAV target tracking domain. A belief revision algorithm based on integrating information valuation "policies" is demonstrated. Organizational aspects of multi-agent systems are also illustrated with an information source selection scheme. Finally, action selection based on macro actions and decision theory is also demonstrated.

6. REFERENCES

[1] Ahn, J., D.Lam, Graser, T., and Barber, K. S. System Engineering Processes Activities for Agent System Design: Component based development for rapid prototyping. In *ICEIS 2005* (Miami, Florida, 2005).

[2] Barber, K. S., Ahn, J., Gujral, N., Graser, T., and Lam, D. Agent Technology Portfolio Manager. In *SEKE 2004* (Banff, Canada, 2004), 37-44.

[3] Barber, K. S. and Graser, T. J. The Systems Engineering Process Activities (SEPA) Methodology and Tool Suite. In *AAAI 2000* (Austin, TX, 2000), 1117 - 1118.

[4] Barber, K. S. and Lam, D. N. Specifying and Analyzing Agent Architectures using the Agent Competency Framework. In *SEKE 2003* (San Francisco, USA, 2003), 232-239.

[5] Barber, K. S. and Park, J. Agent Belief Autonomy in Open Multi-Agent Systems. In *ACE 2004, Lecture Notes in Computer Science* (Springer-Verlag, 2004), 7-16.

[6] Brazier, F. M. T., Dunin-Keplicz, B., Jennings, N. R., and Treur, J. DESIRE: Modelling Multiagent systems in a Compositional Formal Framework. *International Journal of Cooperative Information Systems*, 6, 1 (1997), 67-94.

[7] DeLoach, S. A., Wood, M. F., and Sparkman, C. H. Multiagent Systems Engineering. *SEKE 2001*, 11, 231-258.

[8] Fullam, K. *An Expressive Belief Revision Framework Based on Information Valuation*. Master's, Electrical and Computer Engineering, University of Texas at Austin, 2003.

[9] Han, D. C. and Barber, K. S. Desire-space Analysis and Action Selection for Multiple, Dynamic Goals. In *Proc. of 5th International Workshop on Computational Logic in Multi-Agent Systems* (Lisbon, Portugal, 2004), 182-195.

[10] Smith, D. E. Choosing Objectives in Over-Subscription Planning. In *APS 2004* (Whistler, British Columbia, Canada, 2004), AAAI, 393-401.