The Metaglue Software Agent System

Michael Coen, Luke Weisman, Stephen Peters, Krzysztof Gajos & Nicholas Hanssens

Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

http://www.ai.mit.edu

The Problem: Traditional programming languages provide no support for managing systems of distributed computations, i.e., those in which different components run asynchronously on a heterogeneous collection of networked computers.

Motivation: Metaglue [4] is an extension to the Java programming language that provides very high-level support for writing large groups of software agents that interact with one another.

Metaglue was developed as part of the AI Lab's Intelligent Room Project [3, 2]. The Intelligent Room has literally dozens of hardware and software components that run on a variety of networked workstations. We needed a system that could provide the *computational glue* for linking all of these components and coordinating the flows of data among them.

The computational needs of the Intelligent Room, while not unique, were not satisfied by any pre-existing software systems or programming environments. We wanted the Intelligent Room's software infrastructure to be persistent, robust, and dynamically reconfigurable. We needed the ability to modify (or even introduce) individual components without bringing the whole system down, and we wanted to have tools for understanding and debugging the behavior of large groups of interacting software agents.

Previous Work: There are currently several other pragmatic research systems for creating software agents [7, 11]. They provide low-level functionality, e.g., support for *mobile agents* and directory services, which is necessary but not sufficient. These systems have no ability to manage groups of agents, to modify their activities in principled ways, or to abstractly and concisely describe their behaviors. Others provide some of the grouping functionality [10], but have not been used for extended research into agent frameworks. The Open Agent Architecture [9] has been used in intelligent systems, but is highly task-centric and not as robust.

Approach: Metaglue adds several new primitives and capabilities to the Java programming language. At the lowest level, it allows distributed agents to locate and refer to one another by their functions and capabilities, without respect to where they are physically running.

Metaglue replaces Java's remote method invocation (RMI) mechanism with one that allows dynamic reconnection. This allows individual Metaglue software agents to invisibly resume previously established connections to other Metaglue agents that have been lost for some reason. Useful not only for ensuring reliability, this capability is essential for debugging a persistent, distributed software agent system. It makes it possible to stop, debug, and then seamlessly restart components in a running system.

Using the current version of Metaglue, we have written a robust, distributed controller for the Intelligent Room that typically contains dozens of software agents running simultaneously on four or more networked workstations.

Impact: Because Metaglue is an extension to the Java programming language, it shares Java's portability and flexibility. Its low computational overhead, coupled with the ease of transforming pre-existing software into "Metaglue agents," has made it a very attractive platform for creating systems of distributed, interacting, and persistent software agents. For this reason, Metaglue has been of enormous benefit in the Intelligent Room.

However, there are growing communities of researchers and implementers with similar computational needs, in areas ranging from information retrieval to online commerce to industrial manufacturing. We believe Metaglue's capabilities will be equally useful for them as well.

Metaglue has already been used as the bedrock for research into adaptive behavior mechanisms [8], intelligent help systems [1], as well as innovative user interface designs [5, 12]. All in all, over three hundred software agents

have been written using the Metaglue system, and the system has been deployed in a dozen intelligent environments at the MIT AI Lab and Nokia's research facility.

Future Work: We are currently adding a resource management system [6] to make it simpler for agents to request hardware devices or other agents with more localized knowledge of their requirements, and are also extending some of the base Metaglue concepts for use in multi-spatial systems [13].

The existing system required that all agents in the system be compiled and made available in advance of system startup. We are currently removing this requirement by implementing a remote classloading framework, thus allowing new agents to be created remotely and inserted into a running system.

We are also implementing a testing and assertion framework, to make designing and debugging agents easier.

Research Support: This material is based on work supported by the MIT Project Oxygen.

References:

[1] Andy Chang. A help system for the intelligent room. In *Proceedings of Student Oxygen Workshop* [14].

[2] Michael Coen. Design principles for intelligent environments. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, Madison, Wisconsin, 1998. AAAI-98.

[3] Michael Coen. The future of human-computer interaction or how i learned to stop worrying and love my intelligent room. *IEEE Intelligent Systems*, March / April 1999.

[4] Michael Coen, Brenton Phillips, Nimrod Warshawsky, Luke Weisman, Stephen Peters, Krzysztof Gajos, and Peter Finin. Meeting the computational needs of intelligent environments: The metaglue system. In Paddy Nixon, Gerard Lacey, and Simon Dobson, editors, *Managing Interactions in Smart Environments: 1st International Workshop on Managing Interactions in Smart Environments (MANSE '99)*, pages 201–212. Springer-Verlag, London, 1999.

[5] Mark Foltz. Ligature: Gesture-based configuration of the e21 intelligent environment. In *Proceedings of Student Oxygen Workshop* [14].

[6] Krzysztof Gajos. Rascal - a resource manager for multi agent systems in smart spaces. In *Proceedings of CEEMAS 2001*. Springer, 2001. To appear.

[7] General Magic. *Odyssey (Beta 2)* agent system documentation. http://www.genmagic.com/agents.

[8] Ajay Kulkarni. A reactive behavioral system for intelligent spaces. In *Proceedings of Student Oxygen Workshop* [14].

[9] David L. Martin, Adam J. Cheyer, and Douglas B. Moran. The Open Agent Architecture: A framework for building distributed software systems. *Applied Artificial Intelligence*, 13(1–2):91–128, January–March 1999.

[10] Nelson Minar, Matthew Gray, Oliver Roup, Raffi Krikorian, and Pattie Maes. Hive: Distributed agents for networking things. In *Proceedings of ASA/MA'99, the First International Symposium on Agent System and Applications and Third International Symposium on Mobile Agents*, August 1999.

[11] ObjectSpace, Inc. Objectspace voyager core package technical overview (version 1.0). http://www.objectspace.com/voyager/whitepapers, December 1997.

[12] Alice Oh, Rattapoom Tuchinda, and Lin Wu. Meetingmanager: A collaborative tool in the intelligent room. In *Proceedings of Student Oxygen Workshop* [14].

[13] Stephen Peters. Infrastructure for multi-user, multi-spatial collaborative environments. In *Proceedings of Student Oxygen Workshop* [14].

[14] SOW 2001. *Proceedings of Student Oxygen Workshop*, Gloucester, MA, July 2001.