The following paper was originally published in the
Proceedings of the Fifth Annual Tcl/Tk Workshop
Boston, Massachusetts, July 1997

Agent Development Support for Tcl

R. Scott Cost, Ian Soboroff, Jeegar Lakhani, Tim Finin,
Ethan Miller and Charles Nicholas
Computer Science and Electrical Engineering
University of Maryland Baltimore County

Agent Development Support for Tcl

R. Scott Cost, Ian Soboroff, Jeegar Lakhani, Tim Finin, Ethan Miller and Charles Nicholas
Computer Science and Electrical Engineering
University of Maryland Baltimore County
Baltimore, Maryland 21250
{*rcost1, ian, jlakha1, finin, elm, nicholas*}*@cs.umbc.edu*

In the past few years, the explosive growth of the Internet has allowed the construction of "virtual" systems containing hundreds or thousands of individual, relatively inexpensive computers. The agent paradigm is well-suited for this environment because it is based on distributed autonomous computation. Although the definition of a software agent varies widely, some common features are present in most definitions of agents. Agents should be autonomous, operating independently of their creator(s). Agents should have the ability to move freely about the Internet. Agents should be able to adapt readily to new information and changes in their environment. Finally, agents should be able to communicate at a high level, in order to facilitate coordination and cooperation among groups of agents. These aspects of agency provide a dynamic framework for the design of distributed systems.

Tcl is an ideal language with which to build agents, because scripts written in Tcl may be used on any machine that can run Tcl, and because the Tcl language environment itself is highly portable. Additionally, Tcl/Tk greatly facilitates rapid prototyping and quick development of small applications.

We present TKQML, the integration of an agent communication language, KQML [6] (Knowledge Query Manipulation Language) into Tcl/Tk. TKQML can be used to build KQML-speaking agents that run within a TKQML shell. TKQML can also be used to bind together diverse applications into a distributed framework, using KQML as a communication language. Tcl's embeddable nature allows one to easily add agent communication facilities to existing code. As such, TKQML can be used to enhance the functionality of new or existing systems built using a Tcl framework, by allowing easy integration with agent-based systems.

KQML is a language for general agent communication. It was developed as part of the Knowledge Sharing Effort [7], a DARPA project exploring agent communication and knowledge reuse. KQML is a language based on speech acts, such as "tell", "ask", and "deny", which describe the nature of a message without reference to its content. Agents communicate application-specific information embedded in general, higher-level KQML messages. A comprehensive semantics [4] for KQML outlines protocols for agent "conversation." Additionally, most implementations provide facilities for message handling, agent naming and resource brokering.

Problems of software mobility, communication, and autonomy have not been neglected within the Tcl community. Existing Tcl-based solutions to agent issues, such as AgentTcl [2] and Tacoma [3] have emphasized security and mobility, but fall short with respect to communication. AgentTcl agents, using TCP/IP, exchange bytes strings which have no predefined syntax. In Tacoma, agents must meet in order to communicate. Others projects, such as Tcl-DP [9, 5] provide excellent packages for communication, but lack sufficiently flexible support for higher level languages. TKQML bridges this gap.

The CARROT project (Co-operative Agent-based Routing and Retrieval of Text, formerly CAFE) is an ongoing effort at UMBC to develop a distributed architecture for text-based information retrieval [1], and has served as a testbed for TKQML. This project employs a brokered environment of clients and servers. Users make queries through a World-Wide Web-based client, which are routed intelligently by a broker agent to an appropriate information source. The broker makes these decisions by gathering information, or metadata, from each source, and deciding which database the query most resembles. A ranked set of results is returned to the client.

A heterogeneous set of text-indexing engines, such as Telltale [8] and mg [10] manage large sets of text data. These engines have been augmented into agents with TKQML. The broker agent communicates transparently with these information servers via KQML. All components consist of C/C++ applications bound to TKQML with a Tcl/Tk shell. One agent, the Agent Control Agent (ACA) is written entirely as a TKQML script. Our experience with CARROT has shown that TKQML can facilitate quick prototyping and rapid development of agents and their GUIs, reducing the time necessary to build large agent-based systems. Figure 1 presents a sample system, in which agents of varying types communicate via KQML.

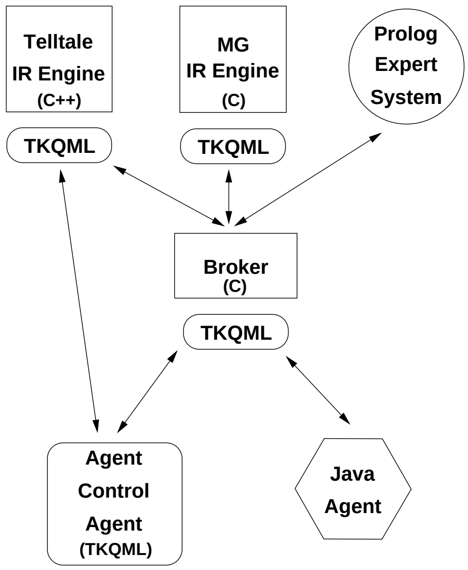

Figure 1: A system of agents speaking KQML. Some entities have been augmented with TKQML, while other have native KQML abilities.

Both Tcl and KQML are powerful tools in the development of agent-based systems. TKQML combines the two, making it possible to benefit from both the light weight and portability of Tcl scripts and the high-level communication support of KQML with one package. We feel that its power, simplicity and potential for future development make it an ideal platform for the development of agent-based systems.

References

[1] Grace Crowder and Charles Nicholas. Resource selection in CAFE: An architecture for network information retrieval. In *Proceedings of the Network Information Retrieval Workshop, SIGIR 96*, August 1996.

[2] Robert Gray. Agent Tcl: A flexible and secure mobile-agent system. In *The Fourth Annual Tcl/Tk Workshop Proceedings*. The USENIX Association, 1996.

[3] Dag Johansen, Robbert van Renesse, and Fred B. Schneider. An introduction to the TACOMA distributed system. Technical report, University of Tromso, June 1995.

[4] Yannis Labrou. *Semantics for an Agent Communication Language*. PhD thesis, University of Maryland Baltimore County, 1996.

[5] Peter Liu, Brian C. Smith, and Lawrence A. Rowe. Tcl-DP name server. In *Proceedings of the 1995 Tcl/Tk Workshop*. The USENIX Association, July 1995.

[6] James Mayfield, Yannis Labrou, and Tim Finin. Evaluation of KQML as an agent communication language. In J. P. Woolridge and M. Tambe, editors, *Intelligent Agents Volume II – Proceedings of the 1995 Workshop on Agent Theories, Architectures and Languages*, Lecture Notes in Artificial Intelligence. Springer-Verlag, 1996.

[7] Ramesh S. Patil, Richard E. Fikes, Peter F. Patel-Schneider, Don Mckay, Tim Finin, Thomas Gruber, and Robert Neches. The DARPA knowledge sharing effort: Progress report. In Charles Rich Bernhard Nebel and William Swartout, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Third International Conference (KR92)*. Morgan Kaufman, 1992.

[8] Claudia Pearce and Charles Nicholas. TELLTALE: Experiments in a dynamic hypertext environment for degraded and multilingual data. *Journal of the American Society for Information Science*, April 1996.

[9] Brian C. Smith, Lawrence A. Rowe, and Stephen C. Yen. Tcl distributed programming. In *Proceedings of the 1993 Tcl/Tk Workshop*. The USENIX Association, June 1993.

[10] Ian H. Witten, Alistair Moffat, and Timothy C. Bell. *Managing Gigabytes: Compressing and Indexing Documents and Images*. Van Nostrand Reinhold, 1994.