COLLABORATIVE DISCUSSION 2: AGENT COMMUNICATION LANGUAGES

Initial Post

Murthy Kanuri

Intelligent Agents

University of Essex

KQML (Knowledge Query and Manipulation Language) is a well-known agent communication language (ACL), designed to give autonomous agents a standardised way of exchanging information and coordinating tasks. Unlike low-level mechanisms such as sockets or RPC, ACLs operate at a higher semantic level, supporting concepts like *inform*, *request*, or *subscribe* (Finin, Labrou and Mayfield, 1994).

Advantages of ACLs.

ACLs offer **flexibility and interoperability**, enabling agents from different developers or domains to collaborate without hard-coding communication details. They also separate **communication from computation**: an agent interprets the performative to grasp intent, while the specific response remains internal. Finally, ACLs are **extensible**, as new performatives or ontologies can be introduced to suit domain needs (Labrou, Finin and Peng, 1999).

Disadvantages of ACLs.

Overhead and complexity are among the drawbacks. KQML's layered structure (performatives, content, ontology) increases verbosity and processing overhead, which may slow high-frequency communication (Russell and Norvig, 2020). The need for a shared ontology presents another difficulty since efficient communication between agents depends on their agreement on a common vocabulary. Constructing and maintaining such ontologies is often difficult and time-consuming (Finin and Fritzson, 1994). As a result, building a full ACL stack requires far more effort than simply calling functions in Python or Java.

Comparison with Method Invocation.

By contrast, method calls in Python and Java are straightforward, efficient, and tightly coupled. Communication is rapid and low overhead since functions or methods are invoked using preset signatures. Systems constructed on a single framework benefit significantly from this (Wooldridge, 2009). However, flexibility is limited by tight coupling, and cross-language calls necessitate bridging. Moreover, method calls only convey **operations**, not **intent**. For example, calculate_total() specifies an action but not the reason. ACLs, however, embed semantics; a query-if performative carries both intent and an extra layer of meta-communication absent in method calls.

Conclusion.

In summary, ACLs are better suited to heterogeneous, distributed environments, while method invocation is most effective in tightly integrated, monolithic systems.

References

- Finin, T., Labrou, Y. and Mayfield, J. (1994) 'KQML as an agent communication language', in *Proceedings of the 3rd International Conference on Information and Knowledge Management (CIKM '94)*. New York: ACM, pp. 456–463.
- Finin, T. and Fritzson, R. (1994) 'KQML – A language and protocol for knowledge and information exchange', in *Proceedings of the 13th*

International Distributed Artificial Intelligence Workshop (DAI-94), pp. 127–136.

- Labrou, Y., Finin, T. and Peng, Y. (1999) ‘Agent communication languages: The current landscape’, *IEEE Intelligent Systems*, 14(2), pp. 45–52.
- Russell, S. and Norvig, P. (2020) *Artificial intelligence: a modern approach*. 4th edn. Harlow: Pearson.
- Wooldridge, M. (2009) *An introduction to multiagent systems*. 2nd edn. Chichester: Wiley.