Initial Post

Knowledge Query and Manipulation Language (KQML) and similar Agent Communication Languages (ACL) were designed to let autonomous agents share information and work together. They are particularly useful because they offer a semantically rich framework where communication actions such as asking and telling are associated (Finin, Labrou and Mayfield, 1994). This abstraction aids in collaboration in heterogeneous multi-agent systems, where agents designed by different developers are able to interact in a meaningful way.

Dynamic systems such as supply chain or smart grid systems require negotiation and goal delegation. These real-world functionalities are core components of ACLs. By offering 'intent' rather than an intention procedure, ACLs enable communication, making them ideal for the autonomy expected in intelligent agents (Wooldridge, 2009). On the other hand, they come with drawbacks like the need for message parsing and interpreting, standardization issues as the transition from KQML to FIPA-ACL.

In contrast, invoking methods in Python or Java allows for easy and efficient direct function calls and is, with minimal processing. On the downside, the lack of ability to denote social intention makes it impractical for distributed autonomous systems. In sum, executing with verbal commands and function calls is prioritized methods ACL focus on the inter-system semantic communication.

References

Finin, T., Labrou, Y. and Mayfield, J. (1994) *KQML as an agent communication language*. Proceedings of the 3rd International Conference on Information and Knowledge Management. ACM, pp. 456–463. Available at: https://doi.org/10.1145/191246.191322 (Accessed: 26 August 2025).

Wooldridge, M. (2009) *An introduction to multiagent systems*. 2nd edn. Chichester: Wiley. Available at: https://www.wiley.com/en-us/An+Introduction+to+MultiAgent+Systems%2C+2nd+Edition-p-9780470519462 (Accessed: 26 August 2025).