COLLABORATIVE DISCUSSION 2: AGENT COMMUNICATION LANGUAGES

Peer Responses

Murthy Kanuri

Intelligent Agents

University of Essex

Table of Contents

1. RESPONSE FROM PEERS

Response from Ali Yousef Ebrahim Mohammed Alshehhi

Thank you, Linga, for your elaborate and objective post. I particularly liked how you mentioned the semantic richness of ACLs compared with the simplistic invocation of a method. Your comments about how both the intention and the action are embedded in ACLs, whereas a method invocation only encodes the action represent a very insightful take and accurately reflect the fundamental aims of both agent communication languages and their design.

I agree with you that ontology management is still a major issue. Without a consensus of shared vocabulary, even messages which are syntactically correct could be interpreted completely differently. My previous evaluations of KQML identified this as a continual problem encountered in real-world implementations (Mayfield, Labrou and Finin, 1995). One way that could potentially lessen this impact, however, is domain-specific ontologies as well as mediation, allowing agents to jointly configure their concepts (Gruber, 1993). This could be another avenue to consider, to augment the disadvantages you mentioned.

I appreciated your comparison to method invocation. Another perspective on this analysis is around scalability. Method invocation is a fine way to launch an operation when the systems are tightly coupled, and efficient processes exist. ACL's open a door to looser coupling and flexibility of design for future evolution (Jennings, Sycara and Wooldridge, 1998). This flexibility is ultimately why the ACL form is still useful among highly distributed heterogeneous systems, although we may bemoan some of the complexity and performance costs. Overall, you gave me some food for thought. I think the human value of simply saying a few words about ways to tackle ontology is the only thing missing from your overall survey of the trade-offs between ACLs and method invocation.

References

Gruber, T.R., 1993. A translation approach to portable ontology specifications. Knowledge Acquisition, 5 (2), pp.199–220.

Jennings, N.R., Sycara, K. and Wooldridge, M., 1998. A roadmap of agent research and development. Autonomous Agents and Multi-Agent Systems, 1 (1), pp.7–38.

Mayfield, J., Labrou, Y. and Finin, T., 1995. Evaluation of KQML as an agent

communication language. In: International Workshop on Agent Theories, Architectures, and Languages. Berlin: Springer, pp.347–360.

Response from Jaafar El Komati

Hi Murthy,

You brought up some important challenges with ACLs, especially the difficulty in building and maintaining shared ontologies. That's a big issue—if different agents use different vocabularies, they can easily misunderstand each other's intentions. That kind of miscommunication can lead to serious coordination problems.

One way to help prevent this is by pushing for stronger standardization of core ontologies across different systems. When developers can agree on a common baseline, agents are less likely to misinterpret each other and more likely to work together effectively (Labrou, Finin and Peng, 1999). Another helpful solution is to use automated ontology alignment tools. These tools reduce the need for manual adjustments by helping agents translate between different vocabularies (Russell and Norvig, 2020).

You also mentioned the overhead and verbosity in KQML. To prevent slowdowns, one strategy could be adopting hybrid communication models. That means using lightweight messages for simple tasks and saving the richer ACL structures for more complex conversations (Singh, 2013). This kind of layered approach keeps communication efficient without losing semantic detail.

In short, while ACLs offer more flexibility and depth than basic method calls, approaches like shared standards, semi-automated ontology alignment, and hybrid protocols could have helped avoid many of the issues you mentioned.

References

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. Pearson.

Singh, M.P. (2013) 'An ontology for commitments in multiagent systems', *Artificial Intelligence and Law*, 21(2), pp. 193–231.

Response from Ahmed Ayman Abdelrahman

Linga, thank you for your clear and balanced discussion of ACLs versus method invocation. I found your point about ontologies particularly important: without a common vocabulary, even the most sophisticated communication protocol fails. This echoes a key challenge in multi-agent systems: semantic interoperability is as critical as the communication channel itself (Labrou, Finin & Peng, 1999).

Your emphasis on the overhead and verbosity of ACLs also resonated with my initial post. As you noted, the layered structure of KQML does make it less efficient than direct method invocation in Python or Java. In tightly coupled environments, the simplicity and speed of method calls remain unmatched. However, I would add that with advances in distributed middleware and lightweight frameworks, some of this overhead can now be mitigated — for example, through optimised parsing or modular ontologies that focus on domain-specific tasks.

One additional angle relates to feature representation, which we studied in Unit 7. The effectiveness of ACLs depends not only on the communication protocol but also on how knowledge is structured and exchanged. Even if the ontology is agreed upon, poorly engineered representations can reduce both interpretability and efficiency (Wooldridge, 2009). This highlights that the real challenge is not ACLs alone but the broader ecosystem of knowledge engineering.

Overall, I agree with your conclusion: ACLs are better suited to heterogeneous, distributed systems, while method invocation excels in monolithic or performance-driven contexts. Both approaches remind us that design choices should align with the intended domain, balancing efficiency, flexibility, and semantic richness.

References

- Labrou, Y., Finin, T. and Peng, Y. (1999) ‘Agent communication languages: The current landscape’, IEEE Intelligent Systems, 14(2), pp. 45–52.
- Wooldridge, M. (2009) An Introduction to MultiAgent Systems. 2nd edn. Chichester: Wiley.

2. RESPONSE TO PEERS

Response to Jaafar El Komati

The focus on KQML is particularly appropriate because of its conversational depth. By using performatives such as *inform*, *request*, or *propose*, agents can communicate data, intentions, and commitments within distributed multi-agent systems (Finin, Labrou and Mayfield, 1997).

This highlights how KQML enables richer and more expressive interactions than straightforward method invocation.

However, semantic interoperability is one of the primary issues. Lack of interoperable or standardised ontologies might lead to misinterpretations, even though agents may exchange KQML messages without technical problems. Sycara (1998) asserts that this challenge emphasises how important ontology construction is to enabling effective agent communication. These issues are being addressed by semantic web technologies and linked data standards developments, which could make KQML-style communication more practical.

A clear trade-off also exists between performance and flexibility. Method invocation in Python or Java is fast, simple, and well-suited to tightly coupled systems. In contrast, open and heterogeneous settings that require asynchronous interaction, negotiation, and discovery are ideal for agent communication languages (Jennings, Sycara, and Wooldridge, 1998). In such cases, the flexibility gained justifies the additional overhead of parsing and ontology management.

Hybrid solutions have been suggested between efficiency and expressiveness. For example, a more expressive ACL such as KQML may be utilised to govern high-level negotiation and coordination, while lightweight method invocation could govern routine contact (Wooldridge, 2009). Such hybrid approaches preserve efficiency while still supporting semantically rich communication.

The decision to use KQML or direct method invocation ultimately depends on the deployment context. Distributed, autonomous, and negotiation-intensive environments benefit from the expressiveness of agent communication languages, whereas simpler, tightly coupled systems function more effectively with direct method calls.

References

- Finin, T., Labrou, Y. and Mayfield, J. (1997) ‘KQML as an agent communication language’, in Bradshaw, J.M. (ed.) *Software agents*. Cambridge, MA: MIT Press, pp. 291–316.
- Jennings, N.R., Sycara, K. and Wooldridge, M. (1998) ‘A roadmap of agent research and development’, *Autonomous Agents and Multi-Agent Systems*, 1(1), pp. 7–38.
- Smythos AI Research Group (2025) *Ontology alignment challenges in multi-agent communication*. Smythos Institute.

- Sycara, K. (1998) ‘Multi-agent systems’, *AI Magazine*, 19(2), pp. 79–92.
- Wooldridge, M. (2009) *An introduction to multiagent systems*. 2nd edn. Chichester: John Wiley & Sons.

Response to Jaco Espag

Your analysis fairly evaluates the advantages and disadvantages of ACLs. One of the most appealing features of ACLs, particularly in distributed and heterogeneous systems, is interoperability. Agents developed in different programming languages and contexts can interact meaningfully through ontological communication (Souza et al., 2016). Because diverse ontological interpretations may make communication difficult, semantic alignment remains challenging.

The contrast you draw between ACLs and direct method invocation is also insightful. Method calls in languages such as Java or Python offer efficiency and simplicity, but they are best suited for tightly coupled and homogeneous environments (Berna-Koes, Nourbakhsh and Sycara, 2004). By comparison, ACLs provide richer semantics that support autonomy and flexible coordination. However, they introduce additional complexity, particularly in developing and maintaining parsers, semantic managers, and other supporting infrastructure (Fatras, Ma and Jørgensen, 2022).

Examining hybrid approaches in greater detail might be beneficial. While ACLs can handle interactions across more diversified or distributed infrastructures, direct method calls may be more appropriate for managing communication within homogenous subsystems. Performance and interoperability can be practically balanced using such a tiered strategy.

Overall, your piece successfully integrates theory and practice with solid backing from the well-chosen sources. You demonstrate how various communication techniques can be used based on the system's requirements and clearly outline the trade-offs.

References

- Berna-Koes, M., Nourbakhsh, I. and Sycara, K. (2004) ‘Communication efficiency in multi-agent systems’, *Proceedings. ICRA ’04. 2004 IEEE International Conference on Robotics and Automation*, 3, pp. 2129–2134. doi:10.1109/ROBOT.2004.1307377.

- Fatras, N., Ma, Z. and Jørgensen, B.N. (2022) ‘An agent-based modelling framework for the simulation of large-scale consumer participation in electricity market ecosystems’, *Energy Informatics*, 5(4), p. 47. doi:10.1186/s42162-022-00229-0.
- Souza, M. et al. (2016) ‘Integrating Ontology Negotiation and Agent Communication’, in Tamma, V. et al. (eds) *Ontology Engineering*. Cham: Springer, pp. 56–68. doi:10.1007/978-3-319-33245-1_6.