Peer Response by Mohamed Alzaabi

Your post highlights the strengths of ACLs in dynamic systems like supply chains and smart grids, where negotiation and goal delegation are vital. I agree with your observation that the abstraction of "intent" makes ACLs ideal for heterogeneous environments. A potential measure to overcome the disadvantages you mention—such as parsing complexity and the transition from KQML to FIPA-ACL—would be stronger standardisation efforts and middleware support. Middleware frameworks could act as translators between different ACLs or ontologies, ensuring backward compatibility and reducing the cost of migration for developers (Singh, 1998).

Another preventive measure would be to design domain-specific ontologies alongside ACLs. For example, in a smart grid application, a well-defined ontology for energy resources would prevent ambiguity when agents negotiate load balancing (Poslad, 2007). This ensures that semantic misunderstandings do not undermine the autonomy and efficiency of the system.

While method invocation in Python or Java may remain efficient for tightly integrated systems, ACLs supported by robust standards and middleware could help overcome the barriers to real-world adoption. Your examples clearly show why ACLs are essential in distributed, autonomous settings.

References

Poslad, S. (2007) Specifying protocols for multi-agent systems interaction. ACM Transactions on Autonomous and Adaptive Systems (TAAS), 2(4), pp.15–29.
Singh, M.P. (1998) 'Agent communication languages: Rethinking the principles', IEEE Computer, 31(12), pp.40–47.

Peer Response by Abdulla Alshaibani

Hi Koulthoum,
Thank you for presenting such a clear and insightful explanation of ACLs and their contrast with direct method invocation. I really appreciate how you emphasized the role of intent in communication, which sets ACLs apart from simple programming calls in languages like Python or Java. Your discussion highlights a critical point: distributed autonomous systems require more than just function execution they need meaningful interaction.

Building on your analysis, it's worth noting that the Foundation for Intelligent Physical Agents (FIPA) framework has been instrumental in addressing the interoperability and standardization challenges you mentioned(Lizán, 2017). By formalizing communication protocols, FIPA ensures that heterogeneous agents can interpret one another's messages consistently, reducing the parsing complexity and ambiguity inherent in earlier ACL versions. This strengthens ACL adoption in large-scale applications.

A practical example can be seen in Siemens' smart grid solutions, where autonomous agents negotiate and delegate load balancing tasks to stabilize energy demand and supply. Similarly, IBM's autonomic computing initiatives use ACL-inspired communication for self-managing systems, enabling agents to handle fault recovery, resource optimization, and service negotiation with minimal human intervention(Sarvapali Ramchurn et al., 2011). These organizational implementations show how ACLs transition from theory into robust enterprise applications.

As a suggestion, linking ACLs with emerging domains such as autonomous vehicles or decentralized logistics could further illustrate their future relevance(Nuraini Diah Noviati et al., 2024). These systems rely heavily on negotiation, trust, and intent representation areas where ACL frameworks excel.

Question for Further Discussion:

Do you think ACLs like FIPA will remain central for intelligent systems, or will lighter alternatives take over?

References:

Lizán, F.J.M. (2017). Intelligent Buildings: Foundation for Intelligent Physical Agents. *International Journal of Engineering Research and Applications*, 07(05), pp.21–25. doi:https://doi.org/10.9790/9622-0705022125.

Nuraini Diah Noviati, Putra, F.E., Sadan Sadan, Ridhuan Ahsanitaqwim, Nanda Septiani and Santoso, L. (2024). Artificial Intelligence in Autonomous Vehicles: Current Innovations and Future Trends. *International Journal of Cyber and IT Service Management*, 4(2), pp.97–104. doi:https://doi.org/10.34306/ijcitsm.v4i2.161.

Sarvapali Ramchurn, Perukrishnen Vytelingum, Rogers, A. and Jennings, N. (2011). Agent-based control for decentralised demand side management in the smart grid. *The Tenth International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2011)*. [online] Available at: https://www.researchgate.net/publication/313363756_Agent-based_control_for_decentralised_demand_side_management_in_the_smart_grid.

Peer Response by Rayyan Alnaqbi

You have pointed out some important differences between traditional method invocation approaches and Agent Communication Languages (ACLs) like KQML, and I completely agree with your analysis. ACLs have a lot of semantic depth, which makes interactions more flexible and independent. This is especially helpful in multi-agent environments where systems are made separately and need to work together without any problems (Finin, Labrou, and Mayfield, 1994).

Your point about how things like supply chains and smart grids work in the real world is very important. Negotiation, delegation, and adaptability are all things that these systems need. ACLs help with these by focusing on intent and performatives instead of fixed procedures. This lets agents not only do their jobs, but also think about the communication, which makes coordination smarter (Wooldridge, 2009).

But I agree that the difficulty of implementation is still a problem. To understand performatives like "inform" or "request," you need to be able to parse and understand their meaning. This adds extra work and makes things more complicated than they need to be, which direct method invocation in Python or Java mostly avoids.

Method invocation works well for systems that are tightly coupled, but it doesn't have the flexibility and freedom needed for dynamic, decentralized environments. ACLs are still better for systems where agents need to make decisions on their own and think about what other people believe and want.

In conclusion, both methods have their pros and cons, but the choice mostly depends on whether the system needs collaborative autonomy or just following procedures.

References:

- Finin, T., Labrou, Y. and Mayfield, J. (1994) KQML as an agent communication language. [Online] Available at: https://www.cs.umbc.edu/kqml/papers/kqml93.pdf [Accessed 8 Sep. 2025].
- Wooldridge, M. (2009) An Introduction to MultiAgent Systems. 2nd ed. Chichester: John Wiley & Sons. [Online] Available at: https://onlinelibrary.wiley.com/doi/book/10.1002/9781118720833 [Accessed 8 Sep. 2025].