

# Collaborative Discussion 2: Agent Communication Languages

## Discussion Topic: Agent Communication Languages

What are the potential advantages and disadvantages of the use of agent communication languages such as KQML? How do they compare with method invocation in Python or Java?

### Learning Outcomes

- An understanding of the main agent models in use today and their grounding in artificial intelligence research.
- The knowledge and skills required to develop, deploy and evaluate the tools and techniques of intelligent systems to solve real-world problems.
- An understanding of contemporary research issues in the area of intelligent agent systems.

### Initial Post

by Anastasia Rizzo - Sunday, 25 June 2023, 4:58 PM

Number of replies: 3

There are dedicated languages designed for communication between intelligent agents within a system, enabling them to exchange information, ask and answer questions, and coordinate their actions. Some of the most well-known communication languages for such agents include:

KQML (The Knowledge Query and Manipulation Language) offers a flexible and straightforward approach to message exchange and querying. It facilitates structured information transfer and efficient management of dialogues between intelligent agents (Chalupsky et al., 1993).

KIF ( Knowledge Interchange Format) serves as a language for formalised knowledge representation. It provides syntax and semantics for expressing logical assertions, facts, and rules that can be exchanged among intelligent agents (aiforanyone.org, 2023).

FIPA-ACL (The Agent Communication Language), developed by the Foundation for Intelligent Physical Agents, establishes standardised formats, protocols, and agent attributes. It enhances agents' interaction and coordination capabilities within a system (fipa.org, 2002).

FIPA-SL (The Agent Specification Language), also developed by the Foundation for Intelligent Physical Agents, allows for the specification of agent properties, capabilities, and skills. It serves as a means to represent crucial information about agents and their roles in a given system (fipa.org, 2002).

AgentSpeak: is a programming language specially designed for specifying the behaviour of intelligent agents. It allows agents to define their goals, behaviour rules, and interactions with other agents (Vo & Ghose, 2004).

Table 1: Advantages and Disadvantages of Agent Communication Languages.

Language	Advantages	Disadvantages
KQML	<ul style="list-style-type: none"> <li>• Simplicity</li> <li>• Flexibility</li> <li>• Structured Information Transfer</li> <li>• Dialog Management</li> </ul>	<ul style="list-style-type: none"> <li>• Limited Adoption</li> <li>• Lack of Standardization</li> <li>• Limited Advanced Features</li> <li>• Limited Community Support</li> </ul>
FIPA-ACL	<ul style="list-style-type: none"> <li>• Standardization</li> <li>• Comprehensive Features</li> <li>• Well-Defined Semantics</li> <li>• Community Support</li> </ul>	<ul style="list-style-type: none"> <li>• Learning Curve</li> <li>• Complexity for Simple Systems</li> <li>• Performance Overhead</li> <li>• Compatibility with Legacy Systems</li> </ul>
FIPA-SL	<ul style="list-style-type: none"> <li>• Agent Specification</li> <li>• Role Representation</li> <li>• Interoperability</li> <li>• Community Support</li> </ul>	<ul style="list-style-type: none"> <li>• Learning Curve</li> <li>• Limited Adoption</li> <li>• Potential Complexity</li> <li>• Compatibility with Legacy Systems</li> </ul>
KIF	<ul style="list-style-type: none"> <li>• Formal Knowledge Representation</li> <li>• Interoperability</li> <li>• Expressive Power</li> <li>• Compatibility with Logic-Based Systems</li> </ul>	<ul style="list-style-type: none"> <li>• Complexity</li> <li>• Learning Curve</li> <li>• Scalability and Efficiency</li> <li>• Limited Adoption</li> </ul>
AgentSpeak	<ul style="list-style-type: none"> <li>• Agent-Oriented Programming</li> <li>• Goal-Oriented Behavior Specification</li> <li>• Rule-Based Programming</li> <li>• Interaction with Other Agents</li> </ul>	<ul style="list-style-type: none"> <li>• Learning Curve</li> <li>• Limited Tooling and Resources</li> <li>• Scalability</li> <li>• Integration with Existing Systems</li> </ul>

*Table 1: Advantages and Disadvantages of agent communication languages.*

The use of communication languages for intelligent agents offers both potential advantages and challenges. On the positive side, these languages provide standardisation, ensuring common conventions and standards for agent communication and integration. They enable interoperability by allowing agents to exchange information and knowledge regardless of their internal implementation. Communication flexibility is another benefit, as these languages support various communication methods and patterns among agents. Additionally, they facilitate distribution and parallelism in multi-agent systems, enhancing resource utilisation and task management.

However, challenges exist as well. The complexity of these languages can be overwhelming, requiring developers to acquire additional knowledge and understanding. Limited tooling support may hinder the development and maintenance process. Moreover, some languages may have a small community and limited resources,

reducing the availability of documentation and assistance. Integration complexity arises when attempting to connect agent systems that utilise different communication languages (Finin & Labrou, 1999).

When comparing agent communication languages with method invocation in Python (runestone.academy, 2023) or Java (javatpoint.com, 2021), there are notable differences. While Python or Java method invocation focuses on direct communication between objects or components within a single system, agent communication languages enable communication and coordination among autonomous agents in distributed environments. Unlike method invocation, which follows a synchronous and deterministic approach, agent communication languages embrace asynchronous and non-deterministic communication patterns. Agents can send messages and continue their tasks without waiting for immediate responses, allowing for concurrent and parallel processing.

Furthermore, agent communication languages provide higher-level abstractions that go beyond simple method calls. They allow agents to exchange complex messages, make queries, and engage in dialogues to negotiate and coordinate their actions towards shared goals. This facilitates flexible and adaptable interactions between agents, empowering them to cooperate, compete, and adapt to changing environments.

## **References:**

aiforanyone.org, (2023). Knowledge Interchange Format (KIF). Available from: <https://www.aiforanyone.org/glossary/knowledge-interchange-format> [Accessed 22 June 2023].

Chalupsky, H., Finin, T., Fritzson, R., McKay, D., Shapiro, S., & Wiederhold, G. (1993). An Overview of KQML: A Knowledge Query and Manipulation Language.

Finin, T. & Labrou, Y. (1999). Agent Communication Communication Languages. University of Maryland Baltimore County. Available from: <https://redirect.cs.umbc.edu/courses/pub/finin/papers/papers/asama99tutorial.pdf> [Accessed 22 June 2023].

fipa.org, (2002). FIPA ACL Message Structure Specification. Available from: <http://www.fipa.org/specs/fipa00061/SC00061G.html> [Accessed 22 June 2023].

fipa.org, (2002). FIPA SL Message Structure Specification. Available from:  
<http://www.fipa.org/specs/fipa00008/SC000081.html> [Accessed 22 June 2023].

javatpoint.com, (2021). Java Method invoke() Method. Available from:  
<https://www.javatpoint.com/java-method-invoke-method> [Accessed 22 June 2023].

runestone.academy, (2023). Method Invocations. Available from:  
<https://runestone.academy/ns/books/published/fopp/AdvancedFunctions/MethodInvocations.html> [Accessed 22 June 2023].

Vo, D., & Ghose, A. (2004). Agent Programming Language within Complete Knowledge - AgentSpeak(I). In Proceedings of the Sixth International Conference on Enterprise Information Systems (pp. 356-363).

### **Peer Response:**

by Samuel Adeniyi - Sunday, 25 June 2023, 10:47 PM  
Thanks, Ana,

This post provides an overview of communication languages for intelligent agents and discusses their advantages and challenges. It mentions several well-known languages, including KQML, KIF, FIPA-ACL, FIPA-SL, and AgentSpeak, highlighting their specific features and benefits. The advantages of using these languages include standardization, interoperability, communication flexibility, and support for distribution and parallelism in multi-agent systems. However, language complexity, limited tooling support, and integration difficulties are drawbacks to ACLs.

You also provided an excellent comparison between agent communication languages and method invocation in Python and Java, emphasizing the differences in their approaches and capabilities. Agent communication languages enable asynchronous and non-deterministic communication, support higher-level abstractions, and facilitate flexible agent interactions.

### **Peer Response:**

by Maksym Dudkin - Monday, 26 June 2023, 4:25 PM  
Hello Anastasia,

Thank you for such a deep and detailed explanation of ACL

You perfectly noticed how ACL could improve multi-agent systems messaging, providing tools for structured and standardised communication among agents.

The only disputable moment is that ACLs facilitate parallelism and distribution; at least, I haven't found strong evidence about such things in mentioned sources. Moreover, in the case of KQML, it is a high-level language which only describes the structure and principles of messaging independent of the transport mechanism and content language (Finin et al. 1999). However, ACLs perfectly fit Domain-Driven Design (DDD), where ACLs could be implemented on the corresponding domain level of the application. At the same time, parallelism and distribution are supported by other application levels (Steinegger, 2017).

Finin, T., Labrou, Y. (1999). Agent Communication Communication Languages.

University of Maryland Baltimore County. Available from:

<https://redirect.cs.umbc.edu/courses/pub/finin/papers/papers/asama99tutorial.pdf>

[Accessed 26 June 2023].

Steinegger, R., Giessler, P., Hippchen, B., Abeck, S. (2017). Overview of a

Domain-Driven Design Approach to Build Microservice-Based Applications. Available from:

[https://www.researchgate.net/publication/316492773\\_Overview\\_of\\_a\\_Domain-Driven\\_Design\\_Approach\\_to\\_Build\\_Microservice-Based\\_Applications](https://www.researchgate.net/publication/316492773_Overview_of_a_Domain-Driven_Design_Approach_to_Build_Microservice-Based_Applications) [Accessed 26 June 2023]