Agent Communication Languages (ACLs) such as the Knowledge Query and Manipulation Language (KQML) have become central to the development of intelligent, cooperative multi-agent systems. Their primary advantage lies in providing a **standardised protocol** for exchanging knowledge, intentions, and goals among autonomous agents (Finin et al., 1994). By defining explicit performatives (e.g. *inform*, *ask*, *recommend*), KQML allows agents to interpret not only the message content but also its **communicative intent**, thereby supporting coordination, negotiation, and reasoning at a higher semantic level (Labrou and Finin, 1997). This facilitates interoperability across heterogeneous systems—a key strength over conventional programming models.

However, ACLs also present **disadvantages**. Their abstraction and semantic richness introduce processing overheads, increasing computational complexity and slowing performance (Wooldridge, 2009). Additionally, the absence of universally agreed ontologies can lead to **semantic misinterpretation**, limiting scalability in real-world deployments. In contrast, **method invocation** in Python or Java is faster and simpler, enabling direct procedure calls with clear syntactic bindings but lacking the autonomy and flexibility inherent in ACL-based communication.

In summary, ACLs like KQML trade efficiency for **intelligence and interoperability**, making them ideal for distributed, adaptive environments where understanding intent is as important as executing instructions.

**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*, pp. 456–463.  
Labrou, Y. and Finin, T. (1997) ‘A proposal for a new KQML specification’, *University of Maryland Baltimore County Technical Report TR CS-97-03*.  
Wooldridge, M. (2009) *An Introduction to MultiAgent Systems*. 2nd edn. Chichester: John Wiley & Sons.