Agent-Oriented Programming in Linear Logic: An Example

Abdullah-Al Amin   Michael Winikoff   James Harland

Department of Computer Science, RMIT University
GPO Box 2476V, Melbourne, 3001, AUSTRALIA
{amin, winikoff, jah}@cs.rmit.edu.au

Agent-oriented programming (AOP) represents a new way of analysing, designing, and implementing complex software applications. This approach has been successfully used in complex applications with distributed components, which require concurrent behaviour, efficient reasoning, dependable communication, sharing and integration of knowledge. Such applications range from from comparatively small systems such as personalised email filters to large, complex, mission critical systems such as air traffic control. Since its introduction, various approaches have been proposed to determine a suitable architecture for agent-based systems. There have been a number of attempts to model AOP, some of which also involve an implemented development environment (such as JACK [1], or dMARS) whilst others are an abstract specification of necessary constructs such as Agent0 [4].

In order to develop and reason about such programs, an ability to take into account concurrency and dynamic state is required. *Linear logic*, introduced by Girard in 1987 [2] has such properties, and has been successfully applied to modelling updates, reasoning about the environment, and implementing concurrent behaviour. Linear logic is often described as *resource-sensitive*, and has been the basis for a number of programming languages including Lygon [3], Lolli, Forum, ACL, $\mathcal{LC}$, and LO.

In this paper we investigate the advantages and disadvantages of developing agent-oriented programs in Lygon, a programming language based on linear logic. In particular, we develop a bank deposit example which serves as an illustration of the advantages and disadvantages of this approach. Due to its basis in linear logic, Lygon can easily and naturally model states, express concurrency, and describe actions. These are the primitive building blocks of agents and hence Lygon appears promising as an agent-oriented programming language.

References

1. Paolo Busetta, Ralph Ronnquist, Andrew Hodgson, and Andrew Lucas. JACK Intelligent Agents - Components for Intelligent Agents in Java. Technical report, Agent Oriented Software Pty. Ltd, Melbourne, Australia, 1998.
2. Jean-Yves Girard. Linear Logic. *Theoretical Computer Science*, 50:1–102, 1987.
3. James Harland, David Pym, and Michael Winikoff. Programming in Lygon: An overview. In Martin Wirsing and Maurice Nivat, editors, *Algebraic Methodology and Software Technology*, LNCS 1101, pages 391–405. Springer, July 1996.
4. Yoav Shoham. Agent-oriented Programming. *Artificial Intelligence 60*, 60(1):51–92, 1993.

---

[0] This paper is based on the first author's Honours thesis.