Chapter XVII
Modeling and Programming with Commitment Rules in Agent Factory

Rem Collier
University College Dublin, Ireland

Gregory M.P. O'Hare
University College Dublin, Ireland

ABSTRACT

Agent-Oriented Programming (AOP) is a relatively new programming paradigm, proposed by Yoav Shoham, which views software systems as consisting of a set of agents that interact with one another to solve problems beyond their individual capabilities. Since the inception of the paradigm, a number of AOP languages have been proposed. This chapter focuses on one such language, the Agent Factory Agent Programming Language (AFAPL), a practical rule-based language that has been applied to a wide range of problem domains including robotics, virtual and mixed reality environments, and mobile computing. AFAPL is placed in context through a general introduction to the state-of-the-art in AOP. The chapter finishes with a discussion of some future trends for AOP and some concluding remarks.

INTRODUCTION

Agent-Oriented Programming (AOP) is a relatively new programming paradigm introduced by Yoav Shoham (1993) in which software systems are viewed as consisting of a set of agents that interact with one another to solve problems that are beyond their individual capabilities. More specifically, agents are viewed as high-level autonomous software entities that encapsulate a set of capabilities and whose internal state is comprised of a set of mental components such as beliefs, capabilities, choices and commitments. This view of agents as mentalistic entities is a

common perspective within multi-agent systems research and underpins many of the most prominent agent theories (Cohen & Levesque, 1990; Rao & Georgeff, 1991; Wooldridge, 2000). These theories model the internal decision-making process of an agent in terms of the interplay between the constituent components of the underlying mental state. Their objective is to define how an agent is able to act in a rational goal-directed manner and to tease out various desirable properties that emerge from that action. Thus, the objective of AOP is to present a framework for developing a new class of programming languages that are derived from these theories.

This view of programs that consist of components whose state is defined by a set of mental qualities and in which computation is realized through speech act based message passing is not unique to Shoham. John McCarthy, who is widely acknowledged as being one of the first to associate mental qualities with machines (McCarthy, 1979) had, as early as 1990, written an draft proposal for a programming language he entitled Elephant 2000 (McCarthy, 1992). That said, McCarthy recognized at the time that his language could be implemented, and to this day, no implementation of Elephant 2000 exists (McCarthy, 2007). In contrast, a number of AOP languages have been proposed, implemented and successfully used to build a range of agent-oriented applications.

This chapter focuses on one such AOP language, entitled the Agent Factory Agent Programming Language (AFAPL) (Collier, 2001; Ross & Collier & O'Hare, 2004; Collier & Ross & O'Hare, 2005). AFAPL has its origins as part of a larger software engineering framework entitled *Agent Factory* (O'Hare, 1996; Collier, 1996; O'Hare & Collier & Conlon & Abbas, 1998; Collier, 2001; Collier & O'Hare & Lowen & Rooney, 2003). Both AFAPL and the associated framework have been designed specifically to support the fabrication of agent-oriented applications, and have been used extensively in the development of a number of prototype systems in areas such as: robotics (O'Hare & Duffy & Collier & Rooney & O'Donoghue, 1999; Dragone & Holz & O'Hare, 2007), mobile computing (O'Hare & O'Grady, 2003; Muldoon & O'Hare & Phelan & Strahan & Collier, 2003), virtual and mixed reality (Duffy & O'Hare & Campbell & Stafford & O'Grady, 2005), wireless sensor networks (Marsh & Tynan & O'Kane & O'Hare, 2004; O'Hare & O'Grady & Marsh & Ruzzelli & Tynan, 2006) and information retrieval (Peng & Collier & Mur & Lillis & Toolan & Dunnion, 2004; Lillis & Collier & Toolan & Dunnion, 2007).

The remainder of this chapter starts with an overview of Agent-Oriented Programming. This is followed by a general overview of the Agent Factory framework. The next section presents details of AFAPL, and after that some of the tools that have been developed to support its use are discussed. The penultimate section, discusses future trends in this area, and finally, some concluding remarks are presented.

AGENT-ORIENTED PROGRAMMING

In his seminal paper, Shoham (1993) introduces the concept of Agent-Oriented Programming as apecialisation of Object-Oriented Programming (OOP). As is highlighted in Figure 1, the core computational unit becomes an agent rather than an object, and the state is constrained to mental components. The second significant restriction that is applied to AOP is the requirement that the only valid types of message are speech acts. Speech act theory (Searle, 1969) is a pragmatic theory of how humans communicate with one another, and has become the defacto standard within the multi-agent systems research community (Wooldridge & Jennings, 1995). Ultimately, the adoption of speech act theory has led to the specification of various Agent Communication Languages (ACLs), such as Knowledge Query Meta Language (KQML) (Finin & Labrou & Mayfield, 1997) and Foundation for Intelligent