Agent Technology

J.-J. Ch. Meyer
Utrecht University, The Netherlands

INTRODUCTION

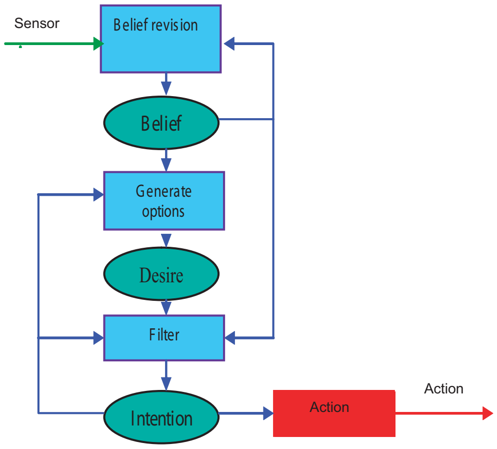
Agent technology is a rapidly growing subdiscipline of computer science on the borderline of artificial intelligence and software engineering that studies the construction of intelligent systems. It is centered around the concept of an (intelligent/rational/autonomous) *agent*. An agent is a software entity that displays some degree of autonomy; it performs actions in its environment on behalf of its user but in a relatively independent way, taking initiatives to perform actions on its own by *deliberating* its options to achieve its goal(s).

The field of agent technology emerged out of philosophical considerations about how to reason about courses of action, and human action, in particular. In analytical philosophy there is an area occupied with so-called practical reasoning, in which one studies so-called practical syllogisms, that constitute patterns of inference regarding actions. By way of an example, a practical syllogism may have the following form (Audi, 1999, p. 728):

Would that I exercise.
Jogging is exercise.
Therefore, I shall go jogging.

Although this has the form of a deductive syllogism in the familiar Aristotelian tradition of "theoretical reasoning," on closer inspection it appears that this syllogism does not express a purely logical deduction. (The conclusion does not follow logically from the premises.) It rather constitutes a representation of a *decision* of the agent (going to jog), where this decision is based on mental attitudes of the agent, namely, his/her beliefs ("jogging is exercise") and his/her desires or goals ("would that I exercise"). So, practical reasoning is "reasoning directed toward action—the process of figuring out what to do," as Wooldridge (2000, p. 21) puts it. The process of reasoning about what to do next on the basis of mental states such as beliefs and desires is called *deliberation* (see Figure 1). The philosopher Michael Bratman has argued that humans (and more generally, resource-bounded agents) also use the notion of an intention when deliberating their next action (Bratman, 1987). An intention is a desire that the agent is committed to and will try to fulfill till it believes it has achieved it or has some other rational reason to abandon it. Thus, we could say that agents, given their beliefs and desires, choose some desire as their intention, and "go for it." This philosophical theory has been formalized through several studies, in particular the work of Cohen and Levesque (1990); Rao and Georgeff (1991); and Van der Hoek, Van Linder, and Meyer (1998), and has led to the so-called Belief-Desire-Intention (BDI) model of intelligent or rational agents (Rao & Georgeff, 1991). Since the beginning of the 1990s researchers have turned to the problem of realizing artificial agents. We will return to this hereafter.

Figure 1. The deliberation process in a BDI architecture



BACKGROUND: THE DEFINITION OF AGENTHOOD

Although there is no generally accepted definition of an agent, there is some consensus on the (possible) properties of an agent (Wooldridge, 2002; Wooldridge & Jennings, 1995): Agents are hardware or software-based computer systems that enjoy the properties of:

- **Autonomy:** The agent operates without the direct intervention of humans or other agents and has some control over its own actions and internal state.
- **Reactivity:** Agents perceive their environment and react to it in a timely fashion.
- **Pro-Activity:** Agents take initiatives to perform actions and may set and pursue their own goals.

- **Social Ability:** Agents interact with other agents (and humans) by communication; they may coordinate and cooperate while performing tasks.

Thus we see that agents have both informational and motivational attitudes, namely, they handle and act upon certain types of information (such as knowledge, or rather beliefs) as well as motivations (such as goals). Many researchers adhere to a stronger notion of agency, sometimes referred to as *cognitive* agents, which are agents that realize the aforementioned properties by means of mentalistic attitudes, pertaining to some notion of a mental state, involving such notions as knowledge, beliefs, desires, intentions, goals, plans, commitments, and so forth. The idea behind this is that through these mentalistic attitudes the agent can achieve autonomous, reactive, proactive, and social behavior in a way that is mimicking or at least inspired by the human way of thinking and acting. So, in a way we may regard agent technology as a modern incarnation of the old ideal of creating intelligent artifacts in artificial intelligence. The aforementioned BDI model of agents provides a typical example of this strong notion of agency and has served as a guide for much work on agents, both theoretical (on BDI logic) and practical (on the BDI architecture, see next section).

CURRENT AGENT RESEARCH: MULTI-AGENT SYSTEMS

Agent-based systems become truly interesting and useful if we have multiple agents at our disposal sharing the same environment. Here we have to deal with a number of more or less autonomous agents interacting with each other. Such systems are called multi-agent systems (MAS) (Wooldridge, 2002) or sometimes also agent societies. Naturally, these systems will generally involve some kind of *communication* between agents. Agents may communicate by means of a communication primitive such as a *send* (agent, performative, content), which has as semantics to send to the agent specified the content specified with a certain illocutionary force, specified by the performative, for example, inform or request. The area of agent communication (and agent communication languages) is a field of research in itself (Dignum & Greaves, 2000). Further, it depends on the application whether one may assume that the agents in a multi-agent system cooperate or compete with each other. But even in the case of cooperation it is not a priori obvious how autonomous agents will react to requests from other agents: Since they ultimately have their own goals, it may be the case that they do not have the time or simply do not want to comply. This also depends on the *social structure* within the agent system/society, in particular the type(s) of coordination mechanisms and power relations employed, such as a market, network, or hierarchy, and the *role* the agent plays in it (Dignum, Meyer, Weigand, & Dignum, 2002).

Another issue related to agent communication, particularly within heterogeneous agent societies, concerns the language (ontology) agents use to reason about their beliefs and communicate with each other. Of course, if agents stem from different sources (designers) and have different tasks they will generally employ different and distinct ontologies (concepts and their representations) for performing their tasks. When communicating it is generally not efficacious to try to transmit their whole ontologies to each other or to translate everything into one giant *universal* ontology if this would exist anyway. Rather it seems that a kind of "ontology negotiation" should take place to arrive at a kind of minimal solution (i.e., sharing of concepts) that will facilitate communication between those agents (Bailin & Truszkowski, 2002; Van Diggelen, Beun, Dignum, Van Eijk, & Meyer, 2006).

Next we turn to the issue of constructing agent-based systems. Since the philosophical and logical work on intelligent agents mentioned in the introduction, researchers have embarked upon the enterprise of realizing agent-based systems. Architectures for agents have been proposed, such as the well-known BDI architecture (Rao & Georgeff, 1991) and its derivatives procedural reasoning system (PRS) and the InteRRap architecture (Wooldridge, 2002). Other researchers devised dedicated agent-oriented programming (AOP) languages to program agents directly in terms of mentalistic notions in the same spirit as the ones mentioned previously. The first researcher who proposed this approach was Shoham with the language AGENT0 (Shoham, 1993). Other languages include AgentSpeak(L)/Jason, (Concurrent) METATEM, CONGOLOG, JACK, JADE, JADEX, and 3APL (Bordini, Dastani, Dix, & El Fallah Seghrouchni, 2005; de Giacomo, Lespérance, & Levesque, 2000; Fisher, 1994; Hindriks, de Boer, Van der Hoek, & Meyer, 1999). One may also adhere to programming agents directly in generic programming languages such as JAVA and C++. Interestingly, one may now ask questions such as how to program agents in these languages in the same way that these questions are asked in software engineering with respect to traditional programming. The thus emerging subfield is called agent-oriented software engineering (AOSE) (Ciancarini & Wooldridge, 2001). In our opinion this makes the most sense in the interpretation of engineering agent-oriented software rather than trying to engineer arbitrary software in an agent-oriented way. So we advocate using an agent-oriented design together with an explicit agent-oriented implementation in an AOP language for those applications that are fit for this approach (Dastani, Hulstijn, Dignum, & Meyer, 2004). Obviously we do not advocate performing all programming (e.g., a sorting algorithm) in an agent-oriented way.

This brings us to the important question, what kind of application is particularly suited for an agent-based solution?