Review of:
Software Agents
Jeffrey M. Bradshaw Editor
AAAI Press / The MIT Press

Claudia Roda
American University of Paris

Nowadays software applications are characterised by their increasing complexity, their coverage of newer domains and the distribution of both data and control. At the same time, given the different backgrounds of possible users, a great emphasis falls on the design of human/computer interfaces which not only present software applications in an easy and widely accessible way, but also adapt to the different users and actively support their work. These characteristics imply that the software must be engineered into systems which are adaptable to new situations and are easy to maintain. These systems must also manage physical distribution and the coordination of the activity of several programs to achieve a common overall goal (satisfy the user). Software agents researchers aim at designing such systems.

In his book "Software Agents", Bradshaw identifies two main objectives of software agents. On the one hand the simplification of distributed computing by the definition of methods that allow "intelligent cooperation among systems to optimally achieve specified goals". On the other hand the improvement of user interfaces moving forward from direct manipulation interfaces toward indirect management, possibly adaptive, interaction where the user could give general guidelines instead of specifying each action explicitly. To introduce the reader to the state of the art achievements in these areas, Bradhsaw's book covers research on agent oriented programming, inter-agent communication, agent mobility, intelligent interoperability between loosely coupled components of distributed systems, antropomorphic and adaptive interfaces, knowledge acquisition, end-user programming, and programming-by-demonstration.

The book is structured as a collection of papers (one per chapter) some previously published, some new for this book. All papers are authored by leading researchers and developers in the field of software agents. Universities (MIT, Stanford, University of Maryland, USC, University of Toronto), computing companies (Apple, Microsoft, General Magic), and industrial research centres (AT&T Bell Labs, Boeing, EURISCO, Interval) are represented.

The style of all the papers in the book is informal in the effort, I assume, to make the book readable for the general public and not only for the software agent community. Notwithstanding the informal style, the papers in the central section do give a good overall description of specific systems and methodologies. For example, Chapter 9 describes KidSim, a symbolic simulation tool kit designed to be used by children;

Chapter 10 reports on Microsoft's Persona project which implements a "lifelike" computer assistant; Chapter 12 overviews the M system, which focuses on the integration of several reasoning processes (society of agents) and its application at AT&T Bell Laboratories to assist work in a virtual meeting room.

For those who are acquainted with the software agents technology, Bradshaw's work will be an easy to read book which offers the occasion to enlarge their view out of their specific field of research. In particular the discussions on philosophical and sociological concerns in the first section of the book, are a good reminder to all of us of the big picture: who are we designing agents systems for and why are we designing agent systems. In this section the discussion ranges from "should we be working on agents systems at all?" to "are people going to be comfortable working with the agents systems we design?".

I found it refreshing to find out that, whilst much of the effort within the software agents research community goes toward ensuring that it keeps its feet firmly on the ground (let's not call our agents 'intelligent'! let's not set too high goals: start from something simple! let's find a verifiable semantic for our messages!, etc.) Shneiderman's paper, the one in the book that argues against software agents, tells us that we should have "much greater ambition than to make a computer behave like an intelligent butler"! ... and we would have been happy to settle for a butler, not even an intelligent one! But to make matters worse, in Chapter 3, Negroponte one of the leading advocates of agent systems, tells us exactly the same thing as Shneiderman: that dreaming of computer interfaces that will be more like people is "shooting too low"! After years concentrating on software agents it may even take a bit of lateral thinking to see what the "greater ambition", or "higher shooting" should be. However computers can do different things than human can do and those things are exactly the ones that we should remember to exploit (no matter which metaphor we choose to use).

Shneiderman also touches a soft spot when he points out that predictability is a necessary quality of software systems. In fact, human control and predictability are a recurrent subject in the book. In Chapter 2 Norman goes as far as asserting that the main difficulties for agent systems "are social, not technical". People will use software agents systems only if they feel in control, i.e. they understand the system and are reassured that the system is "working according to plan". Users want to be able to predict the behaviour of their software and, as Negroponte and Maes (Chapter 8) point out, they also want to be able to decide if and when a task should be delegated. Obviously, what one person may perceive as understandable and working according to a possibly efficient or smart plan, may be different from another person's perception. As a consequence software agents must be able to adapt to their users. Several authors in the book (e.g. Negroponte, Laurel, Maes) concentrate on the dual knowledge (competence) that agents systems should have: knowledge about a specific domain, e.g. searching on the web, maintaining a file system, etc. and knowledge about its user, e.g.

his/her taste, his/her interest, etc.

The unaware reader may wonder what various systems described in the book such as the Oval system (Chapter 7) and the Persona project (Chapter 10) have in common. Oval is a system which allows end-users to program agents for information management applications. The Persona project aims at the design of life-like computer assistant. What do these systems have in common that makes them both agent systems? The answer is that the book describes both works where the agent metaphor is the main issue: the system presents a life-like character, and works where the system is actually structured like an agent, e.g. is semi-autonomous, is designed as having certain mental properties, etc. These two aspects of agent systems are very clearly discussed by Erickson in Chapter 5.

On the side of the applications represented, I found the book biased toward interface agents and generally toward the issues related to the interaction between the human user and the agent systems. Both end-user programming and machine learning techniques for adaptive systems are quite well represented in the book. Much less space has been given to industrial applications designed for specialised users (e.g. the pilot associate program (Smith and Broadwell 1988), particle accelerator control system (Jennings 1994), telecommunication systems (Huhns et al. 1994, Sugawara and Murakami 1992, Weihmayer and Veltguijsen 1994) ) where the focus is on timeliness, agents coordination, partial knowledge of the environment, etc.

Several issues of agents micro-theory are discussed in the various chapters. Aspects of agents macro-theory are also tackled, especially in the last section of the book where a collection of excellent papers gives an overview of the current debate and achievements in multi-agent systems research and development.

Shoham (Chapter 13) sets the scene with an overview of agent-oriented programming. Whilst the question “why agents and not objects?” is still one of the most frequently asked to members of the software agents community, I believe that Shoham’s paper (but see also Erickson’s paper and Chapter 17 authored by Bradshaw, Dutfield, Benoit and Woolley) can be a very quick and clear answer.

Topics related to agents communication are quite well covered. Two papers (Chapter 14 and 15) relate on the results obtained by the DARPA Knowledge Sharing Effort project and its applications. Finin, Labrou and Mayfield (Chapter 14) report on the de-facto standard agent communication language KQML. Genesereth (Chapter 15) reports on his Agent Communication Language (ACL) designed to allow interoperability amongst heterogeneous systems through the construction of federation architecture based on facilitators. Cohen and Levesque (Chapter 18) analyse KQML and point out how its lack of a formal semantic specification engenders several problems for its use as an agents communication language. They briefly introduce their very influential theory of

rational action and explain how it could form the basis for a formal semantic of an agent communication language.

Some more aspects of communication as well as agents modelling, coordination, cooperation, learning, agent mobility etc. are discussed in various chapters of the book although their coverage is not so thorough.

Overall I found the book very readable and a good source of information thanks also to a good index and the many references in the chapters of the book. If you intend to use the book as a text for a lecture course you will probably need to add to the introduction provided by Bradshaw a few more introductory papers taken, for example, from the ones listed in the reference section of Chapter 1.

Huhns M. N., Singh M. P. Ksiezyk T. (1994) “Global information management via local autonomous agents” Proceedings ICOT international symposium on fifth generation computer systems: Workshop on heterogeneous cooperative knowledge bases.

Jennings N. R. (1994) “Cooperation in industrial multi-agent systems” World scientific series in computer science, Vol.43 1994.

Smith D., Broadwell M. (1988) “The pilot association - an overview” Proceedings SAE Aerotech conference, Los Angeles, Ca. 1988.

Sugawara T., Murakami K. (1992) “A multiagent diagnostic system for internetwork problems” Proceedings of the Internet Society - INET 1992 International networking conference, Reston, Va.

Weihmayer D., Veltguijsen H. (1994) “Application of distributed AI and cooperative problem solving to telecommunication” Proceeding 13th international DAI workshop, Seattle Washington.