Editorial

Intelligent Agents: Software Technology for the new Millennium

Boi Faltings, Guest Editor

This article gives an introduction to intelligent agents and its history. It shows why agents are the logical next step in the evolution of computer software, and what role intelligence will play in this development. A brief review of the history and state of the art in intelligent agents provides concrete entry points for applying and developing the technology.

1 Why agents?

Computing has evolved from the age of calculating machines (1950–60) through information processing (1970–80) to information environments (1990–2000). Information environments, such as the world wide web, provide the basis for *autonomous* software systems. Examples of such systems are systems for mail and news delivery, document indexing (web crawlers), and consistency maintenance. Autonomous systems have also appeared in robotics, in particular for mobile robots.

At the same time, software is becoming complex to the point of being unmanageable. For example, the Microsoft Windows 2000 system contains 35 million lines of code with many bugs. Its release date has been pushed back several times and it is not clear whether it will ever become sufficiently reliable for commercial success. To avoid the difficulties of maintaining consistency in such large software projects, the tendency is to decompose systems into small components which can be understood independently of each other. However, component paradigms are still lacking intuitive metaphors which make component behaviour easy to understand.

Both developments – autonomous software and robotic agents as well as software components – are now converging into a single technology: *agents*. Agents turn software components into proactive processes that are:

- *autonomous* in that they react themselves to observations of their environment without requiring explicit commands,
- *proactive* in that they recognize and react to changes in the environment which present opportunities,
- *embedded* in that their actions respect the real-time constraints imposed by the environment,
- *heterogeneous* in that many different kinds of agents can work together in the same system, and be added or removed without interrupting it.

Boi Faltings is professor of computer science at the Swiss Federal Institute of Technology in Lausanne (EPFL), where he founded and directs the Artificial Intelligence Laboratory. He has published over 100 papers in different areas of AI and has recently focussed his attention on intelligent agents and their application.

According to the agent vision, complex homogeneous software systems will be replaced by networks of communicating agents. Agents can be written independently as long as they conform to a standard communication language, and they can be integrated or modified even in a running system. Infrastructures for agent-based computing are being developed with very heavy investment from large companies such as SUN and IBM and are beginning to appear on the market (for example, JINI from SUN).

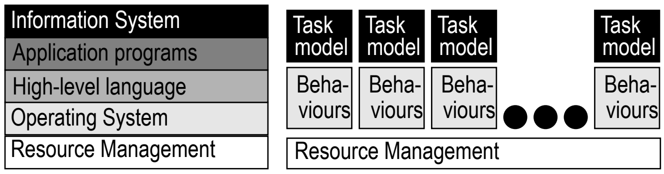

Fig. 1: Homogenous software systems are replaced by a collection of simple agents specific to certain tasks.

2 From objects to agents

When Adam Smith published his classic work *The Wealth of Nations*, he introduced the argument that local decision-making by groups of individuals was more likely to lead to fruitful results than central planning by governments – thus laying the foundations of modern market economies and unprecedented wealth. Agents will bring about a similar revolution in computer software.

Figure 1 shows how agent technology changes the way software is structured. In conventional technology, software is built as layers of abstraction. They require careful system-wide design since each layer has to conform to one single interface metaphor: it is complex to integrate different applications, difficult to combine programs written in different languages, and impossible to integrate programs written for different operating systems. As the number of functionalities increases, so does the complexity of interfaces and user interaction.

In contrast, agent systems (right) are a multitude of independent programs built around *task models*. They require standardization only at the level of resource management (i.e.

networks, files, processes), but can otherwise be written using different languages and metaphors. Each agent can be structured so as to best achieve the functionality it is designed for, rather than having to conform to numerous constraints motivated by different functions.

New functionalities increase the number of agents, but do not influence the complexity of each individual agent or its interfaces. A given application would only instantiate agents that correspond to functionalities that are actually used. Since agents are themselves proactive, they can decide themselves to become active and propose their functions whenever this appears sensible given the observed context. Similar to a market economy, localized decisions and self-interestedness lead to greater efficiency.

3 Agents with intelligence

Beyond these basic characteristics of agent technology, agents are likely to become more *intelligent*:

- *adaptive* agents which learn by themselves to fulfil their user's desires, or to react or survive in their environment,
- *learning* agents which are capable of improving their performance on the same task by reusing earlier experiences,
- *rational* agents which have explicit goals and reasoning capabilities, thus giving them the ability to receive explicit instructions or explain their behaviour to human users in very flexible ways,
- *communicating* agents which can cooperate and negotiate using powerful agent communication languages.

Intelligence is important for two reasons. The first is that adaptive and learning systems will be essential for customizing complex software to user requirements and their environment of use. The second is that just like economies of people, efficiency of agent systems is greatly increased when agents are *self-interested* and rational beings. This requires explicit recognition of goals and intentions so that they can be negotiated and communicated.

4 Where do agents come from?

Agents have their origins in four different research areas: robotics, artificial intelligence, distributed systems, and computer graphics.

Agent work in robotics and artificial intelligence was originally strongly interrelated. Robots such as SHAKEY [Nilsson 84] were programmed to exhibit autonomous behaviour in well-defined environments, and laid the groundwork for AI planning systems to this day. The first software agent was probably ELIZA [Weizenbaum 65], a program which could engage in a conversation with a user. Another influential program, SHRDLU [Winograd 73], allowed a person to have a conversation with a simulated robot.

The notion of *multi-agent* systems was brought to the forefront by Marvin Minsky in his work on the "Society of Mind" [Minsky 87]. His vision was that a complex system such as the human mind should be understood as a collection of relatively simple agents, each of which was a specialist in certain narrow domains. Through structures called K-lines, agents would activate each other whenever their context became relevant. The work of Minsky showed remarkable vision, but was ahead of its time since software complexity had not yet reached the level where the advantages of such structures would have a practical impact.

However, the idea of decomposing a complex system into simple agents found willing takers in robotics. Frustrated with the complexity of robots built around general and thus large homogeneous software systems, Rodney Brooks [Brooks 91] proposed a radically different design. In his view, intelligent and complex behaviour would be emergent in the interplay of many simple *behaviours*. Each behaviour is a simple agent whose activation is decided by a control architecture. Complex general vision systems were replaced by simple detectors specialized in particular situations, and actions were taken based on very simple rules. Brooks showed that using this approach, one could very easily build robust autonomous robots which had not been possible otherwise.

A group of researchers has developed approaches that fall in between the purely behaviour-oriented approach proposed by Brooks and knowledge-based systems otherwise prevalent in AI. For example, Stuart Russel has concentrated on rational behaviour in real-time environments using techniques such as Bayesian networks. Leslie Kaelbling and others have developed reinforcement learning approaches for robots and other problem-solving agents. These approaches have had a significant influence on the AI community.

In artificial intelligence, however, agents also developed in other directions. Research on *distributed* artificial intelligence started to make cooperation and communication among agents the primary focus of attention. Distributed algorithms had always generated some interest in the AI community, but were now growing into much more ambitious distributed problem-solvers. The goal here is to get a set of agents with limited capabilities and knowledge to develop a coherent plan of joint action that will best achieve their objectives, e.g. combined payoff. The advantage of such a framework is that knowledge can be distributed and can be kept partially confidential, that the system is open and can accept additional agents, that it is more robust in case of failure, and that it is more efficient due to the parallelism of many agents. Wooldridge and Jennings have shown how the model of distributed problem-solving can be implemented for a wide variety of problems [Jennings/Wooldridge 95]. Researchers such as Durfee et al. [Durfee et al. 89], Yokoo et al. [Yokoo et al. 90] and others have developed distributed methods for planning and constraint satisfaction.

The approach gains additional appeal when agents are self-interested and make decisions to increase their own local benefit. This leads to frameworks for coordination and negotiation among agents. Rosenschein and Zlotkin [Rosenschein/Zlotkin 94], Lesser, Krauss and others have proposed mechanisms by which agents can find agreement on a common objective in spite of conflicts of interest; in some cases these protocols can be shown to guarantee optimal and fair results. With the increasing interest in electronic commerce, this work has been directed towards auction and contract mechanisms, as developed for example by Maes, Wellmann, Sandholm and others. This has become an extremely active area of research.

Frameworks such as the BDI framework (Rao & Georgeff, [Rao/Georgeff 95], [d'Inverno 97]) and the framework of Cohen & Levesque [Cohen/Levesque 90] have provided logical tools for analysing multi-agent systems and their behaviour, and at the same time fixed some of the basic notions for programming and modelling agent behaviour.

Another important body of work deals with the interaction between agents through communication languages. The DARPA knowledge sharing initiative produced mechanisms for sharing knowledge, in particular KIF (knowledge interchange format [Genesereth/Fikes 92]), KQML (knowledge query and manipulation language [Finin et al. 93]) and ONTOLINGUA [Gruber 92] for modeling ontologies. These have provided the first frameworks for *interoperability* of agents in heterogeneous environments. They have given rise to standardization efforts: KQML is undergoing the process of ANSI standardization, while an international organization called FIPA (Foundation for Intelligent Physical Agents, [FIPA]) is developing a general set of standards for interoperable agents with heavy involvement from large companies (e.g. Siemens, NEC, IBM, Sun).

The agent approach has also had its impact on software engineering, for example in simulation software [Durfee et al. 98] and for process control such as air traffic control [Georgeff 94]. Such systems show that the agent approach can bring remarkable simplification not only to intelligent systems, but to software in general. The notion of *agent-oriented programming* has been introduced by Shoham in 1993 [Shoham 93].

Finally, agents also appeared in computer graphics and user interfaces. Computer animation has produced movies with virtual actors whose behaviour albeit had to be programmed manually [Magnenat-Thalmann/Thalmann 87]. In artificial life, researchers are working on autonomous actors and their group behaviour. Artificial characters in user interfaces guide customers through artificial shopping malls. Microsoft software has included various agents to help users with their work.

Progress in agent research is evident in several annual agent conferences, in particular the ICMAS and AGENTS conference series. In Europe, the AGENTLINK network [AgentLink] brings together more than 40 research institutions active in agent research. We believe that this large body of research results is now ready for synthesis into more widely applicable methodologies.

An important driver for intelligent agents has been the success of the internet and distributed systems [Petrie 96]. Telecommunications has been using agents for some time (e.g. the TINA architecture) to reduce signalling overhead in networks. Standards such as CORBA are providing more general infrastructure for distributed computation and communication between agents.

Recently, the lead has been taken over by the major companies in the computing industry such as Microsoft, SUN, IBM and Hewlett-Packard. They are proposing their own standards and investing heavily to promote them. Thus, Microsoft's COM, while arguably a lot less powerful than the open CORBA standard, has rapidly become more common. It is competing heavily with JINI, proposed by SUN but also strongly supported by IBM, a more radical design which completely eliminates operating systems. Following in the same line as JINI, Hewlett-Packard is proposing e-speak, focusing more on the semantic layer. This strong activity in basic development can be expected to rapidly lead to the development of usable and widely available platforms for agent systems. It also means that the stakes of entry in platform development are now raised beyond what academic research can afford.

Based on the distributed infrastructure, a large body of work has addressed useful applications of agent and multi-agent systems. Researchers such as Oren Etzioni [Etzioni/Weld 94] and Pattie Maes [Maes 94] promoted the idea of software robots, or *softbots*. These are agents for helping users with a variety of tasks, such as finding information (e.g. the AHOY! homepage finder [Shakes et al. 97]), synthesizing community information (e.g. the Firefly collaborative filtering mechanism [Firefly]). Several successful companies have been founded around software agent technology, for example JANGO and Agents Inc., showing that the technology is becoming commercially viable.

5 Where do we go from here?

Intelligent agents are a very active research topic in laboratories around the world. This special issue gives several examples of technologies and applications of intelligent agents, mostly in Switzerland.

The first two articles show examples of true autonomous artificial agents. The article by *Thalmann, Musse and Kallmann* shows how intelligent agents can be the core of full-scale virtual humans acting as believable characters, opening the door to entirely new ways of communication and interaction with computers. *Dario Floreano* shows how robotic agents can take on life-like qualities, including learning and evolution, thus evolving computers from being mere tools to individuals in their own right.

Besides such agents with a physical or at least virtually physical existence, there is much interest in pure software agents. The important issue in software agents in their ability to cooperate in a heterogeneous environment, a capacity which requires communication. The article by *Labrou and Finin* give an overview of the current state of the art in agent communication languages, an key area for large-scale deployment of agent systems in the future.

There are two main commercial application areas of intelligent agents today. The first is in telecommunications, and the article by *Steven Willmott and Monique Calisti* presents a survey of current work in this very active area. The other is in electronic commerce, where agents act as smart intermediaries for their users. The article by *Thomas Steiner* shows an example of this in tourism. It also points out the interest of agent technology from a business perspective.

Another application of agents in business are electronic auctions, where agents will be required to automate the burden of bidding for goods in different marketplaces. The article by

Rodriguez and Cortes shows an example of how such agents and their platforms may look, illustrating another promising application of agent technology.

I hope that this issue will stimulate your interest in intelligent agent technology and make you look at applying it in your own environment. Activity in this area is rapidly building up in Switzerland and Europe, and you will find competent partners at most Universities. There are also numerous activities organized by the Agentlink network (http://www.agentlink. org/), and major conferences such as AGENTS (this year held in Barcelona June 3–7, http://www.iiia.csic.es/agents2000), ICMAS (this year held in Boston July 7–12, http://icmas.lania.mx), ECAI (this year held in Berlin August 20-25, http://www.ecai2000.hu-berlin.de/) and AAAI (this year held in Austin July 30–August 3, http://www.aaai.org/). I would be glad to meet you at one of these events in the future!

References

[AgentLink]
AgentLink: Network of Excellence for agent-based computing http://www.agentlink.org/

[Brooks 91]
R. Brooks: Intelligence without representation. *Artificial Intelligence* **47**, pp. 139–159, 1991

[Cohen/Levesque 90]
P. R. Cohen, H. J. Levesque: Intention is choice with commitment. *Artificial Intelligence* **42** (1990), 213–261.

[Durfee et al. 98]
E. H. Durfee, P. G. Kenny, K. C. Kluge: Integrated Premission Planning and Execution for Unmanned Ground Vehicles. Autonomous Robots, 5:1-14, 1998.

[Durfee et al. 89]
E. H. Durfee, V. R. Lesser, D. D. Corkill: Trends in cooperative distributed problem solving. *IEEE Transactions on Knowledge and Data Engineering KDE-1*, 1(March 1989), 63–83.

[Etzioni/Weld 94
O. Etzioni, D. A. Weld: softbot-based interface to the Internet. *Communications of the ACM 37*, 7 (July 1994), 72–76.

[Finin et al. 93]
T. Finin et al.: Specification of kqml agent communication language. Tech. rep., DARPA Knowledge Sharing Initiative, External Interfaces Working Group, 1993.

[FIPA]
FIPA. Foundation for Intelligent Physical Agents. http://www.fipa.org/

[Firefly]
Firefly http://www.firefly.com/

[Genesereth/Fikes 92]
M. R. Genesereth, R. E. Fikes: Eds. Knowledge Interchange Format, Version 3.0 Reference Manual. Computer Science Department, Stanford University, Technical Report Logic-92-1, June 1992.

[Georgeff 94]
M. Georgeff: dmars: Distributed multiagent reasoning system. Tech. rep., Australian AI Institute, 1994.

[Gruber 92]
T. Gruber: Ontolingua: A mechanism to support portable ontologies. Tech. Rep. 91–66, KSL, Stanford University, 1992.

[d'Inverno et al. 97]
M. d'Inverno, D. Kinny, M. Luck, M. A. Wooldridge: Formal Specification of dMARS Proceedings of Forth International Workshop on Agent Theories, Architectures and Languages (ATAL'97), 1997

[Jennings/Wooldridge 95]
N. Jennings, M. Wooldridge: Intelligent agents: Theory and practice. *The Knowledge Engineering Review 10*, 2 (1995), 115–152.

[Maes 94]
P. Maes: Agents that reduce work and information overload. *Communications of the ACM 37*, 7 (94), 31–40.

[Magnenat-Thalmann/Thalmann 87]
N. Magnenat-Thalmann, D. Thalmann: The direction of synthetic actors in the film rendez-vous in montreal. *IEEE Computer Graphics and Applications 7*, 12 (Dec. 1987), 9–19.

[Minsky 87]
M. Minsky: *The Society of Mind*. Heinemann, London, 1987.

[Nilsson 84]
N. J. Nilsson: Shakey the robot. Technical Note 323, SRI International, Menlo Park, California, Apr. 1984.

[Petrie 96]
C. Petrie: Agent-Based Engineering, the Web, and Intelligence, *IEEE Expert* **11** (6), pp. 24–29, 1996

[Rao/Georgeff 95]
A. Rao, M. Georgeff: BDI agents: From theory to practice. In *Proceeding of the 1st International Conference on Multi-Agent Systems* (San Francisco, CA, June 1995), V. Lesser, Ed., AAAI Press, pp. 312–319.

[Rosenschein/Zlotkin 94]
J. S. Rosenschein, G. Zlotkin: *Rules of Encounter: Designing Conventions for Automated Negotiation among Computers*. The MIT Press, 1994.

[Shakes et al. 97]
J. Shakes, M. Langheinrich, O. Etzioni: Dynamic Reference Sifting: a Case Study in the Homepage Domain. *Proceedings of the Sixth International World Wide Web Conference*, pp.189–200, 1997; also: http://ahoy.cs.washington.edu:6060/doc/home.html

[Shoham 93]
Y. Shoham: Agent-oriented programming. *Artificial Intelligence 60*, 1 (1993), 51–92.

[Weizenbaum 65]
J. Weizenbaum: ELIZA – a computer program for the study of natural language communication between man and machine. *Communications of the Association for Computing Machinery 9*, 1 (Jan. 1965), 36–45.

[Winograd 73]
T. Winograd: A procedural model of language understanding. In *Computer Models of Thought and Language*, R.Schank and K. Colby, Eds. W.H.Freeman, New York, 1973, pp. 152–186.

[Yokoo et al. 90]
M. Yokoo, T. Ishida, K. Kuwabara: Distributed constraint satisfaction for DAI problems. In *Proceedings of the 1990 Distributed AI Workshop* (Bandara, TX, Oct. 1990).