Software agents are computer programs capable of flexible, autonomous action. In their most complex form, agents may persist over time, are capable of timely internal context-dependent reaction to sensed events, plan and initiate unique series of actions to achieve stated goals, and communicate with other agents (or people) toward those ends. The growth of interest in software agent technology throughout the 1990s has gone hand-in-hand with the growth of the Internet.

With this special issue of IEEE Internet Computing, we investigate issues surrounding the use of agent technology for Internet and intranet-related applications. The scope of the issue is deliberately broad, including issues ranging from the technological (what are the right technologies for agent-based Internet applications, and how are these deployed?) to the pragmatic (how will users interact with and control agent-based Internet systems?)
BREAKTHROUGH

In many respects, 1994 was a breakthrough year for intelligent autonomous agents. It was the year a special issue of Communications of the ACM introduced several papers that proved to have enormous influence on the agent research community’s agenda. It was also the year the World Wide Web emerged from its academic origins into the wider public consciousness.

In the wake of this activity, a major conference on agent systems was established the following year. The first International Conference on Multi-Agent Systems (ICMAS), held in San Francisco in June 1995, marked a certain degree of maturity in the field.

Of the many reasons for agent technology’s explosive growth in the late 1990s, the most important is undoubtedly the easy-to-use front end and powerful underlying infrastructure the Internet has provided for building agent applications. In addition to this technology push, the problems presented by the everyday use of the Web (in particular, the need to develop personalized, cooperative, proactive tools for information gathering and management) have generated a rich set of issues that are now driving research teams across the globe.

SIX YEARS AFTER

It seemed appropriate, at the turn of the millennium, to present an overview of the state of the art in agents, and see how the original vision has matured and changed over the past six years. In this special issue of IC, we have explicitly attempted to include articles that cover the whole spectrum of agent research, from information gathering and management agents, through applications of mobile agent technology, to what you might call traditional cooperative multiagent systems.

Probably the most important observation we can make on the basis of guest editing this special issue is that agents have become ubiquitous. Although the Internet remains the dominant application domain for deploying agents, many key ideas of the technology—such as transparent distributed processing, robust

---

Resource-Bounded Searches in an Information Marketplace

Victor Lesser, Bryan Horling, Anita Raja, Xiaoqin Zhang, and Thomas Wagner

The BIG information-gathering agent processes Web documents to create product models and recommend purchases. New capabilities allow control of the money spent in acquiring information from sites that charge an access fee.

Domain Name eXchange: A Mobile-Agent–Based Shared Registry System

Lassaâd Gannoun, Julien Francioli, Stanislav Chachkov, Fréderic Schutz, Jarle Hulaas, and Jürgen Harms

DNX is a platform-independent system designed to extend a shared registry system for managing domain names. DNX aims to provide several value-added services to the market, including intelligent registration.

Building Dynamic Agent Organizations in Cyberspace

Milind Tambe, David V. Pynadath, and Nicolas Chauvat

Karma-Teamcore provides a development framework for heterogeneous multiagent systems that use wrappers to encapsulate general teamwork reasoning in the execution of an abstract team plan.

Nomad: Mobile Agent System for an Internet-Based Auction House

Qianbo Huai and Tuomas Sandholm

The Nomad system allows mobile agents to travel to the eAuctionHouse site and participate in auctions on the user’s behalf.

NetChaser: Agent Support for Personal Mobility

Antonella Di Stefano and Corrado Santoro

NetChaser is an agent-based infrastructure for supporting personal mobility in accessing Internet information services. It exploits agent capabilities to follow users when they change working terminals.

JATLite: A Java Agent Infrastructure with Message Routing

Heecheol Jeon, Charles Petrie, and Mark R. Cutkosky

JATLite is a tool for creating agent systems. It includes a message router that supports message buffering, allowing agents to fail and recover, and a name-and-password mechanism that lets agents move freely between hosts.
and open service provision and selection, and standardized content languages—have found their way into almost every aspect of modern computing and information technology (IT). As the technology becomes ever more widespread and better understood, we can only expect this trend to continue.

There remain, however, aspects of agent-based technology that are not yet commonplace in non-agent-based systems. These fall primarily on the line between agent-based and mere object-based systems. Thus, while standards such as CORBA and Jini have brought distributed service provision and selection to non-agent-based systems, other key ideas exist only in the agent-based world. These include more complex and dynamic coordination, higher level communication protocols, and structured, persistent action. (A true agent will be somewhat “stubborn” about achieving a goal—retrying after failures, attempting different activities, and so on—in a quite different manner than the typical computational object).

Still, the development of agents as a new software paradigm is somewhat analogous to the rise of object-oriented techniques. Originally, object-oriented programming was perceived as only applicable to a few niche areas, such as user-interface development. As the technology matured, however, it became increasingly clear that the object metaphor was useful across the whole spectrum of computer science and IT.

The embodiment of object-oriented programming today—the Java language—has become ubiquitous, and we might expect to see agent technology develop similarly. Just as we can conceptualize the world as a collection of interacting (but essentially passive) objects, so we can conceptualize it as a collection of interacting autonomous agents.

Where To Learn More About Software Agents

The references in each of the theme articles in this issue are all good places to begin further inquiry. The following resources should be useful as well.

Books

Survey papers

Journals

Conferences
- Int’l Conf. on Multi-Agent Systems, held biannually (ICMAS-2000 will be held in Boston); see http://icmas.lania.mx/.
- Int’l Conf. on Autonomous Agents (Agents), held annually (Agents-2000 will be held in Barcelona); see http://www.iiia.csic.es/agents2000/.

There are also several long-standing workshop series held annually or biannually, including
- Agent Theories, Architectures, and Languages (ATAL) http://www.atal.org
- the U.S. Distributed AI workshops
- the European Modeling Autonomous Agents in a Multi-Agent World workshops (MAAMAW)
- the Cooperative Information Agent (CIA) workshop http://www.dfki.de/~klusch/cia2000.html

Websites
The premier information website is the UMBC Agent Web, run by Tim Finin and located at http://www.cs.umbc.edu/agents/.

Michael Wooldridge is a professor in the Computer Science Dept. at the University of Liverpool, U.K., where he heads the agent research group. He has been active in the research and development of multiagent systems for 10 years, gaining his PhD for work in the theoretical foundations of multiagent systems from the University of Manchester in 1992.

Keith Decker is an assistant professor in the Dept. of Computer and Information Sciences at the University of Delaware. His research interests include distributed problem solving, multiagent systems, real-time problem solving, computational organization design, concurrent engineering, parallel and distributed planning and scheduling, and distributed information gathering and retrieval. Decker received his PhD from the University of Massachusetts at Amherst.