Introduction to This Special Issue

The Many Faces of Agents

Katia P. Sycara

The recent interest and excitement in the area of agent-based systems has resulted from the confluence of a variety of research disciplines and technologies, notably AI, object-oriented programming, human-computer interfaces, and networking. Developing agents that could perceive the world, reason about what they perceive in relation to their own goals and acts, has been the Holy Grail of AI. Early attempts at such holistic intelligence (for example, SRI International's SHAKEY robot in the late 1970s) proved frustrating in part because of the immaturity of the AI technologies but most importantly because of the unreliability of the hardware. AI researchers turned their attention to component technologies for structuring a single agent, such as planning, knowledge representation, diagnosis, and learning. Although most of AI research was focused on single-agent issues, a small number of AI researchers gathered at the Massachusetts Institute of Technology Endicott House in 1980 for the First Workshop on Distributed AI. The main scientific goal of distributed AI (DAI) is to understand the principles underlying the behavior of multiple entities in the world, called *agents* and their *interactions.* The discipline is concerned with how agent interactions produce overall multiagent system (MAS) behavior. The presence of other agents in the world that must be taken into account in each agent's reasoning forced DAI researchers to confront issues of agent situatedness in a multiagent environment, perception of other's behavior, communication, and action that affects the behavior of others. In addition, because of limited local information, agents in an MAS always operate in the presence of some degree of uncertainty.

Meanwhile, the object-oriented community was defining and refining the notion of objects as units of software design that encapsulate state. They have some control over their state in the sense that the state can only be accessed or modified using the methods that the object provides. Objects could be distributed and invoked remotely through remote procedure calls or message passing. Networking and distributed systems technology was developing a fast and reliable infrastructure that facilitates fast and secure communication and efficient distributed computation. Human-computer interface research was coming up with task delegation as an alternative to direct manipulation as a way for humans to interact with computer systems.

All these separate strands of research and technology gave rise to the realization that these communities were concerned with different aspects of the notion of agency. Agency was defined by Eisenhardt (1989) and mathematically modeled. An agency relationship is present when one party (the principal) depends on another party (the agent) to undertake some task on the principal's behalf. The notion of agency covers cooperative coordination in MASs (agents depend on each other's cooperation to perform their tasks); delegation in human interface design; object-oriented programming, where an object uses another; and self-interested coordination through contracting in MASs.

Since the early 1990s, there has practically been a deluge of research papers dealing with agents and implemented systems that claim to be agent based (spanning e-mail filtering, information retrieval from the web, electronic commerce, entertainment, and spacecraft control). This diversity, albeit demonstrating the vitality and excitement of the field, also contributes to the confusing picture it provides. This confusion is intensified by the "overhyping" of the term *agent.* There have been many attempts at defining an agent and endless discussions about what constitutes agenthood. Although no agreed-on definition exists yet, there seems to be a convergence of opinion that an agent is a computer software system whose main characteristics are situatedness, autonomy, adaptivity, and sociability. We hold that all these characteristics must be present simultaneously for a system to qualify as an agent. *Situatedness* means that the agent receives some form of sensory input from its environment, and it performs some action that changes its environment in some way. The physical world and the internet are examples of environments in which an agent can be situated. *Autonomy* means that the agent can act without direct intervention by humans or other agents and that it has control over its own actions and internal state. *Adaptivity* means that an agent is capable of (1) reacting flexibly to changes in its environment; (2) taking goal-directed initiative, when appropriate; and (3) learning from its own experience, its environment, and interactions with others. *Sociability* means that an agent is capable of interacting in a peer-to-peer manner with other agents or humans.

Many researchers emphasize different aspects of agency, such as mobility (such additional properties might nevertheless be useful for certain applications). We believe that the previous four properties, when present in a single software entity, are what uniquely characterize an agent as opposed to related software paradigms, such as object-oriented systems, or expert systems (see also Jennings, Sycara, and Wooldridge [1998] for more detailed discussion). The agent paradigm offers a new promise for building complex software because of the abstraction and flexibility it provides. These complex systems are conceived as organizations of coordinating agents. A complex domain could be decomposed into modular, functionally specific software components—agents. When interdependencies arise, the agents adaptively coordinate to handle them. The agent-based view could offer a powerful set of tools, techniques, and metaphors that has the potential to redefine the ways people interact with and build software. This prospect is exciting for us because it brings concepts and techniques developed by AI researchers into the forefront of software development.

The Articles in This Issue

This issue brings together an eclectic set of articles on topics that, in general, have not been in the mainstream of AI.

The article by Clark Elliot and Jacek Brzenzski surveys research issues and approaches in building believable agents that communicate and interact with humans. Such software is especially attractive to humans because of its redeeming social qualities that enable humans to interact with agents in entertaining and innovative ways. The authors argue that these qualities will not only facilitate creative applications, for example, in entertainment and teaching, but also could elucidate issues and provide useful insights in the development of agent-based systems as such.

The article by Eaton, Freuder, and Wallace explores the synergies between constraint-based reasoning and agents and argues for increased interaction among the constraint research community and the agent research community so that productive cross-fertilization of ideas and techniques can ensue. Constraint-based agents are suitable for solving a variety of interesting and practically significant problems that can be modeled as constraint satisfaction or constraint optimization. These problems include resource-allocation problems (where only resource constraints must be taken into account); more complex scheduling problems, such as job-shop scheduling and meeting scheduling, where both resource and temporal constraints must be considered; configuration problems; and parametric design. For such applications, agents whose architecture is based on constraints are most effective; however, constraint-based reasoning can be useful in agent-based systems to allow agents to reason about available computational resource constraints (for example, network bandwidth) or reach agreeable compromises in negotiations over limited resources.

My article presents the basic motivations and research issues in MASs. It surveys research in multiagent problem solving and coordination that has its roots in the DAI community. Interactions based on cooperative, as well as self-interested, agents are presented. It is argued that as computer applications and research move increasingly toward more dynamic and open environments, MASs will become the dominant paradigm for software structuring. This development leads to a set of open issues and future directions for developing MASs.

The article by Michael Lewis critically presents the agent metaphor for human-computer interaction. Interactions based on an agent metaphor treat the computer as an intermediary that responds to user requests. The term *agent* has come to refer to the automation of aspects of human-computer interaction, such as anticipating commands or autonomously performing actions. Norman's 1984 model of human-computer interaction is introduced as reference to organize and evaluate research in human-agent interaction. A wide variety of heterogeneous research involving human-agent interaction is shown to reflect automation of one stage of action or evaluation within Norman's model. Improvements in human-agent interaction are expected to result from more heterogeneous use of methods that target multiple stages simultaneously.

The paper by Yoshiyasu Nishibe et al. presents the experiences of the Mobile Assistant Project that took place during the Second International Conference on Multiagent Systems (ICMAS96) in Japan. The goal of the project was to use mobile digital assistants to provide various types of electronic support for a large number of diverse users in a real environment. Each project participant was given a personal digital assistant that had various services, such as sending and receiving e-mail, finding conference-related information, and finding and interacting with other participants. Large amounts of data were collected from the use of these services by participants in the Mobile Assistant Project during the five days of the conference. These data pertained to (1) single-user information seeking and the types of support that individual conference participants used and (2) the social interactions of participants. Analysis of the data, conclusions about the lessons learned regarding the utility of the technology, and design guidelines are presented in the article.

Conclusions

Agent-based software systems offer the promise of revolutionizing software-engineering technology as they make substantial progress toward the long-held goal of AI, namely, the development of holistic entities that interact intelligently with their environment, the users, and each other. What is currently lacking is (1) a coherent set of problem-solving principles that can profitably be combined to enable the effective structuring of agents and MASs; (2) the consolidation of techniques into a coherent and systematic methodology that enables system designers to define and structure their applications in terms of agents; and (3) a set of agent behaviors and human-agent–interaction paradigms that allow humans to interact naturally with agents, develop trust in them, and ease the acceptance of agent technology. I hope that the articles in this special issue have contributed toward these goals. I would like to thank my colleagues for contributing their articles to this issue.

References

Eisenhardt, K. 1989. Agency Theory: An Assessment and Review. *Academy of Management Review* 14(1): 57–74.

Jennings, N.; Sycara, K.; and Wooldridge, M. 1998. A Roadmap for Agent Research and Development. *Autonomous Agents and Multiagent Systems* 1(1). Forthcoming.