Jeffrey O. Kephart

Jeffrey O. Kephart manages the Agents and Emergent Phenomena group at the IBM Thomas J. Watson Research Center. He studied electrical engineering and physics at Princeton University and Stanford University. He was a visiting scientist at Xerox PARC for two years, and then joined IBM Research in 1990. His primary research interest is the exploration and exploitation of analogies between large-scale computer systems and large-scale natural systems like ecologies and economies. He has pursued a number of theoretical and practical issues relating to computer viruses, and led the development of some novel anti-virus technologies, including a neural net virus detector and an immune system for cyberspace. His work has been cited or published in the Wall Street Journal, The New York Times, Wired, The Atlantic Monthly, IEEE Spectrum, and Scientific American.

Recently, Kephart has led IBM's Information Economy research effort, aimed at understanding and designing the future of software agents and electronic commerce. His other research interests include emergent phenomena in computer networks and intelligent mail agents.

EDUCATION

Ph.D. in Electrical Engineering, Ph.D. in Physics (minor), Stanford University, 1987.

Ph.D. thesis: Beyond the continuum potential approximation: channeling radiation from non-ideal crystals.
Advisor: Richard H. Pantell

M.S. in Electrical Engineering, Stanford University, 1981.

B.S. summa cum laude, Electrical Engineering and Computer Science, Engineering Physics program, Princeton University, 1980.

POSITIONS HELD

1997-. Manager, Agents and Emergent Phenomena (IBM Research)
1994-1997. Manager, Computer Virus Science and Technology (IBM Research)
1990-1994. Research Staff Member, IBM Research, Yorktown Heights, New York.
1987-1989. Visiting scientist, Xerox Palo Alto Research Center.
1987. Instructor, Stanford University. Designed and taught graduate course on Quantum Electronics.

AWARDS

2001. IBM Research. 2000 Computer Science Best Paper award.
1996. IBM Corporate Award for IBM AntiVirus Technology and System Design
1996. Finalist, Discover Magazine Awards for Technological Innovation
1995. IBM Outstanding Innovation Award for IBM AntiVirus Products
1990. Le Prix Cecoia for Best Paper at 2nd International Conference on Economics and Artificial Intelligence, July 4-6, 1990.
1980-1983. National Science Foundation Fellowship.
1980-1981. Stanford University Fellowship.
1976. 1st place, Philadelphia Science Council competition.

PATENTS: 8 U.S. Patents granted.

PUBLICATIONS: Approximately 50 journal papers or book chapters, plus approximately 35 conference papers. A sampling of papers appears on the next page.

A.R. Greenwald and J.O. Kephart, *Shopbots and Pricebots*. **Agent-Mediated Electronic Commerce**, A. Moukas, ed. Springer Verlag, 2000.
Received IBM Research Computer Science Best Paper award, 2001..

J.O. Kephart and A.R. Greenwald, *When bots collide*. **Harvard Business Review**, July-August, 2000.

J.O. Kephart, G.B. Sorkin, D.M. Chess and S.R White, *Fighting Computer Viruses*
Scientific American, November 1997, pp. 88--93.

J.O. Kephart, S.R. White, and D.M. Chess. *Computers and Epidemiology*. **IEEE Spectrum** (cover article), May 1993, pp. 20-26.

C.A. Waldspurger, T. Hogg, B.A. Huberman, J.O. Kephart, and W.S. Stornetta, *Spawn: A Distributed Computational Economy*. **IEEE Transactions on Software Engineering**, 18(2):103--117, 1992.

J.O. Kephart, T. Hogg, and B.A. Huberman, *Can Predictive Agents Prevent Chaos?*. **Economics and Cognitive Science**, P. Bourgine and B. Walliser, eds., Pergamon Press, New York (1991).
Received best paper award, Proceedings of CECOIA 2, Second International Conference on Economics and Artificial Intelligence, Paris, July, 1990.

1 The grand challenge of autonomic computing

Create computer systems as powerful as today's large enterprise and inter-enterprise systems that are *autonomic*, or completely self-managing. More specifically, at every scale from the enterprise level down to the application and resource (e.g. database and storage) levels, the computing system should be:

- **Self-configuring**—capable of configuring itself automatically in accordance with high-level policies that specify goals rather than means, and supporting automated, seamless incorporation of new components, and new types of components;
- **Self-maintaining**—maintaining and adjusting its operation in the face of changing workloads, demands and external conditions, and availing itself of minor or major upgrades on the fly;
- **Self-optimizing**—continually seeking ways to improve its operation, monitoring itself to identify and seize opportunities to make itself more efficient in terms of performance or cost; and
- **Robust**—healing or protecting itself in the face of hardware or software failures of innocent or malicious origin.

In order to meet the overall grand challenge of autonomic computing, it will be necessary to meet several significant subsidiary challenges in engineering and science, particularly within the realms of software architecture and software engineering, programming languages, multi-agent systems, machine learning, nonlinear dynamics, economics, and human factors. Here is a small sampling.

2 Engineering challenges for autonomic computing

1. Define a set of fundamental architectural principles for autonomic computing from which the properties of self-configuration, self-maintenance, self-optimization and robustness emerge.

   These principles should apply generally across multiple scales of computation from the device level up to the enterprise and inter-enterprise level, and should serve as the basis for mechanisms and standards that facilitate platform-independent and level-independent interactions among *autonomic elements* (the constituent parts of autonomic systems).

2. Develop new languages, metaphors and translation technologies that enable humans to specify goals and objectives to autonomic computing systems and visualize their potential effect.

   The techniques must provide a sufficient degree of expressiveness of preferences, cost and performance tradeoffs, security policies, and matters of risk and reliability, yet they must be sufficiently structured and/or naturally suited to human psychology and cognition to keep specification errors to an absolute minimum.

3. Develop appropriate software engineering concepts and programming tools for composing autonomic elements and systems, including support for expressing and understanding goals and strategies, and support for various aspects of relationships with other elements, especially the establishment, monitoring and enforcement of agreements.

4. Develop methods for testing and verifying behavior of autonomic elements, including autonomic testbeds and simulation environments, formal verification methods, and mechanisms that permit new versions of software to run alongside old versions until they have established their trustworthiness.

5. Develop a programming model and runtime architecture that facilitates proactive management of the lifecycle of autonomic elements and their service relationship with one another.

6. Develop robust, non-invasive, scalable approaches to monitoring and controlling the security and performance of heterogeneous, distributed systems. Monitoring approaches may include techniques for discovering and representing service relationships among autonomic elements, and for correlating measurements across elements that may be under separate ownership.

7. Develop methods for expressing and integrating into autonomic elements a knowledge of local, national, and international laws, to ensure that the elements behave legally in large-scale autonomic systems.

3 Scientific challenges for autonomic computing

1. Define appropriate abstractions and/or models for understanding, controlling, and exploiting emergent behavior in autonomic systems.

   More specifically, establish how the autonomic properties of self-configuration, self-optimization, and self-maintenance and robustness, as well as the stability and performance of the system and its constituent elements, depend upon

   - the behaviors, goals, and degree of adaptivity of the individual constituent elements of the system;
   - the pattern and type of interactions among them; and
   - the external influences or demands placed upon the system.

2. Establish to what extent (and by what methods) a designer can induce desired system behavior by imbedding the right local behavior and interaction rules into autonomic elements, and by creating the right topological pattern of interactions among them.

3. Establish a theoretical foundation for understanding and performing learning and optimization in cooperative and competitive multi-agent systems in which the individual adaptive agents must take into account one another's ability to adapt.

4. Establish a theoretical foundation for negotiation from the perspective of individual autonomic elements, and from the perspective of the system as a whole. In particular,

   - develop and analyze negotiation protocols and algorithms;
   - determine how system behavior depends on the mixture of negotiation algorithms employed by the population of autonomic elements; and
   - establish the conditions under which multilateral (as opposed to bilateral) negotiations among elements are necessary and/or desirable in both cooperative and competitive multi-agent systems.

5. Automate to the fullest possible extent the construction and learning of adaptive statistical models of large networked systems that allow overall performance problems to be detected or predicted from a stream of sensor data from individual devices.

6. Develop appropriate theories and theoretical constructs for measuring, understanding, and proving properties of autonomic systems. These may include

   - a process algebra with general primitives for initiating, monitoring, moving, killing, retrying, restarting, compensating autonomic elements, and
   - a general theory of tasks and services, including representation, composition/decomposition rules, conflict graphs, and an algebra and logic of tasks.

7. Develop a theory of robustness for autonomic systems, including definitions, analyses and relationships among robustness, optimality, diversity, and redundancy.