Given the scale and complexity of today’s information systems, it’s increasingly important that they handle system management problems and tasks themselves—intelligently and autonomously. They must be able to work around errors and failures as well as adapt their configuration for optimal response to actual system state or environment—with minimal or no human intervention. Self-management can significantly reduce costs and increase robustness. Moreover, it’s sometimes required because time or location constraints make human management impossible. Bringing self-management about in a variety of systems such as wired or wireless networks, peer-to-peer systems, the computational Grid, and distributed and embedded systems and applications in general is an exciting and quickly growing research area.

This special issue focuses on one particular approach, motivated by observations of the self-organizing and self-managing systems that surround us—multicellular organisms, social insects, market economies, human societies, ecosystems, and so on. These systems are made of components that obey some local rules and act on the basis of local observations—often selfishly. Yet the system as a whole exhibits global properties such as self-healing, self-tuning, and self-organization—exactly the properties we’re after for complex information systems. Distilling the key ideas from these systems and incorporating them into information systems often leads to inexpensive, straightforward, and highly robust solutions.

**Self-organization**

It’s difficult to define self-organization in a way that satisfies everyone. Perhaps a definition isn’t so important or productive. Instead, we can work with an intuitive concept that emphasizes decentralization and emergent functionality. In this sense, a self-organizing system consists of many interacting com-
ponents that have partial or no global system knowledge. The components interact locally with other system components according to some network of possible interactions. The components are autonomous in that no central controller directs their actions toward global goals according to current system state information. Instead, the components determine their own actions, taking as input only local information available from their neighbors in the communication topology and the environment. Despite this local behavior, the system as a whole meets its global goals.

In most cases, the components don’t even include representations of the system’s global state and global goals. They might not even have individual goals. In other cases, the components might know the global goals completely or partially but not know exactly how to cooperate to achieve them. In yet other cases, components might have individual goals that contradict the system goals in one way or another.

Self-managing systems are unquestionably interesting simply because many of them are crucial to us—society, a living cell, ecosystems, and so on. These are by definition self-organizing, and we need to understand them as best as we can.

But do we really want or need to build self-organizing information systems? After all, controlling a system from a central location is much easier and more convenient. It’s a natural thing to do, and many times it’s also the right thing to do. However, in some important cases, self-organization is useful, and at times it’s necessary.

For example, central coordination isn’t feasible in systems such as sensor networks, mobile ad hoc networks, and other embedded or spatial networks. The devices that form the network simply aren’t powerful enough for their radio signals to reach a common central location, and relay stations might not be available.

Nor is central control feasible when system components belong to different administrative domains that aren’t willing to cede control. Prime examples of this are the Internet itself and, more recently, peer-to-peer networks.

Even when not required, a self-organizing decentralized design might be useful. For example, if reliability is a top system priority, you want to avoid any single points of failure, such as a controller. The traditional way of dealing with this is to replicate functionality, say with backup controllers; but a truly self-organizing design can achieve significantly higher reliability levels with no critical system elements. Or when performance is important, self-organizing decentralized designs can offer greater parallelism and avoid communication bottlenecks more easily.

In this issue

This issue includes five articles that represent some of the most important techniques and application areas of self-organization.

“Infrastructure for Engineered Emergence on Sensor/Actuator Networks,” by Jacob Beal and Jonathan Bachrach, discusses the Proto language and its implementation for high-level programming of sensor-actuator networks. Proto helps in designing complex coordination and data processing operations over spatial networks of arbitrary size and density. The authors demonstrate the approach by generating synchronized wave patterns, both in simulations and in real hardware.

“Extended Stigmergy in Collective Construction,” by Justin Werfel and Radhika Nagpal, presents algorithms that allow for completely independent mobile robots to collaborate in building physical structures. The robots communicate with each other implicitly, through the structure itself. The authors compare and analyze several possibilities for implementing this stigmergic communication.

“SLACER: A Self-Organizing Protocol for Coordination in P2P Networks,” by David Hales and Stefano Arteconi, presents a protocol that lets a peer-to-peer system optimize the sum of the peers’ utility, where individual utility is a function of mutual interactions among the peers. In a Prisoner’s Dilemma game, the protocol maintains a high cooperation level and therefore good overall utility.

“Self-Organization in Wasp and Open Source Communities,” by Sergi Valverde, Guy Theraulaz, Jacques Gautrais, Vincent Fourcassié, and Ricard V. Solé, demonstrates universal self-organization principles by drawing parallels between wasps and open source developer communities. This work is an excellent reminder that self-organization is at work in large and complex information systems, whether we apply it consciously or not.

Finally, “Customer-Driven Sensor Management,” by Tracy Mullen, Viswanath Avasarala, and David L. Hall, applies a market-based approach to allocating sensors to tasks. They transform a bidding procedure into an optimization problem that, when solved or approximated, results in an optimal or good assignment.

The Authors

Márk Jelasity is a researcher in the Department of Computer Science at the University of Bologna, Italy, and a senior researcher scientist in the Research Group on Artificial Intelligence at the University of Szeged and Hungarian Academy of Sciences. His research interests include peer-to-peer systems, reliable large-scale distributed systems and computing, and bioinspired and evolutionary computing. He received his PhD in computer science from the University of Lie- den. Contact him at the Università di Bologna, Dipartimento di Scienze dell’Informazione, Mura Anteo Zamboni 7, I-40126 Bologna, Italy; jelasity@cs.unibo.it.

Ozalp Babaoglu is a professor of computer science at the University of Bologna, Italy. His research activities include distributed computing and complex adaptive systems. He received his PhD in computer science from the University of California at Berkeley, where he was a principal designer of BSD Unix. In 1993, he received the Usenix Association Lifetime Achievement Award for his contributions to the Unix system community and to open standard industry standards. He’s an ACM fellow and serves on the editorial boards for ACM Transactions on Computer Systems, ACM Transactions on Autonomous and Adaptive Systems, and Distributed Computing. Contact him at the Università di Bologna, Dipartimento di Scienze dell’Informazione, Mura Anteo Zamboni 7, I-40126 Bologna, Italy; babaooglu@cs.unibo.it.

Robert Laddaga is a research scientist at the Massachusetts Institute of Technology Computer Science and Artificial Intelligence Laboratory. His research activities include self-adaptive software technology, perceptually enabled interfaces, software for physically embedded autonomous systems, and programming language design. He received his PhD in philosophy of science from Stanford University. Contact him at MIT CSAIL, 32 Vassar St., Cambridge, MA 02139; rladdaga@csail.mit.edu.

To give readers a broader spectrum of ideas from this exciting field, we invited leading experts from various disciplines to informally summarize their research positions. Under the general title “Interdisciplinary Research: Roles for Self-Organization,” these four invited contributions close the issue.