University of South Carolina

Scholar Commons

Faculty Publications

Computer Science and Engineering, Department of

2005

Research Directions for Service-Oriented Multiagent Systems

Michael N. Huhns
University of South Carolina - Columbia, huhns@sc.edu

Munindar P. Singh

Mark Burstein

Keith Decker

Edmund Durfee

See next page for additional authors

Follow this and additional works at: https://scholarcommons.sc.edu/csce_facpub

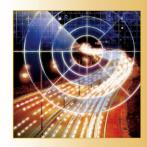Part of the Computer Engineering Commons

Publication Info

Author(s)

Michael N. Huhns, Munindar P. Singh, Mark Burstein, Keith Decker, Edmund Durfee, Tim Finin, Les Gasser, Hrishikesh Goradia, Nick Jennings, Kiran Lakkaraju, Hideyuki Nakashima, H. Van Dyke Parunak, Jeffrey S. Rosenschein, Alicia Ruvinsky, Gita Sukthankar, Samarth Swarup, Katia Sycara, Milind Tambe, Tom Wagner, Laura Zavala, and MAS Research Roadmap Project

Research Directions for Service-Oriented Multiagent Systems

Today's service-oriented systems realize many ideas from the research conducted a decade or so ago in multiagent systems. Because these two fields are so deeply connected, further advances in multiagent systems could feed into tomorrow's successful service-oriented computing approaches. This article describes a 15-year roadmap for service-oriented multiagent system research.

We've already seen service-oriented computing (SOC) take hold in cross-enterprise business settings, such as the use of FedEx and UPS shipping services in e-commerce transactions; the aggregation of hotel, car rental, and airline services by Expedia and Orbitz; or book-rating services for libraries, consumers, and bookstores. Given the widespread interest in and deployment of Web services and service-oriented architectures that are occurring in industry, the scope of SOC in business settings will expand substantially. However, the emphasis has been on the execution of individual services and not on the more important problems of how services are selected and how they can collaborate to provide higher levels of functionality. Fortunately, four major trends in computing are addressing this problem:

- Online ontologies are enabling meaning and understanding, arguably the last frontier for computing, to be captured and shared in more refined ways — via the Semantic Web initiative, for example, with the development of languages and representations for marking up heterogeneous content. In an alternative approach, shared representations are emerging from the works of (millions of) independent content developers. These ontologies will form models for numerous real-world entities and systems, as well as for the meanings of documents and content.

- Ubiquitous computing, consisting of widespread embedded processing with local awareness, is making huge strides in global deployment. Most of the world's objects with distinct identities and exhibiting state or behavior will likely soon have processors or RFID tags. The processors themselves consider only narrow domains of intelligence — a door, for example, could have a processor that knows whether it's currently locked and under what

Michael N. Huhns,
Munindar P. Singh,
Mark Burstein, Keith Decker,
Edmund Durfee, Tim Finin,
Les Gasser,
Hrishikesh Goradia,
Nick Jennings,
Kiran Lakkaraju,
Hideyuki Nakashima,
H. Van Dyke Parunak,
Jeffrey S. Rosenschein,
Alicia Ruvinsky,
Gita Sukthankar,
Samarth Swarup,
Katia Sycara, Milind Tambe,
Tom Wagner,
and Laura Zavala
MAS Research Roadmap Project

conditions it should be unlocked.

- Entities, from corporations to individuals, will provide numerous computational behaviors in the form of Web services and service architectures that can be discovered, engaged, and enacted by others.[1]
- The widespread availability of many different types of sensors and effectors (including actuators and robotic devices) will enable online entities to not only become aware of the physical world, but also to manipulate, change, and control it.

These trends are the new enablers that will drive SOC and multiagent system (MAS) research in the next decade and beyond. They portend an era in which complex systems will be modeled and simulated not just to understand them, but also to form predictions and interpretations that guide the monitoring and managing of them. SOC brings to the fore additional considerations, such as the necessity of modeling autonomous and heterogeneous components in uncertain and dynamic environments. Such components must be autonomously reactive and proactive yet able to interact flexibly with other components and environments. As a result, they're best thought of as agents, which collectively form MASs. Additionally, the key MAS concepts are reflected directly in those of SOC:

- *ontologies* (simplified representations of knowledge in a domain, developed with the purpose of facilitating interoperation);
- *process models* (simplified representations of activities and their enactment);
- *choreography* (simplified business protocols through which services can interact);
- *directories and facilitators* (simplified "middle agents" from MASs); and
- *service-level agreements* and *quality-of-service measures* (automated negotiation and flexible service execution in dynamic environments).

SOC represents an emerging class of approaches with MAS-like characteristics for developing systems in large-scale open environments. Indeed, SOC presents several challenges that can't be tackled without MAS concepts and techniques. Viewed in this light, MASs offer many ways in which to change the face of computing.

Multiagent Systems

The history of MASs mirrors the history of comput-ing in general. In the 1980s, distributed computing over LANs and advances in expert systems motivated the initial interest in distributed agents. Because the resulting systems functioned in single organizations, cooperation was the main focus. In the 1990s, opening LANs to the Internet ushered in an interest in MASs, which have a dynamic topology whose agents potentially could be implemented and maintained by more than one organization. The research focus shifted to interaction in general, with possibilities for emergent behaviors: the global behavior of a system consisting of many agents emerges from the interactions among agents and isn't always obvious from the agents' individual behaviors.

The problems that MASs address aren't new — ancient societies and economies encountered the same basic challenges of autonomy and heterogeneity. What is new is how the emergence of the Internet and ubiquitous computing have rendered traditional, manual approaches ineffective against such challenges, placing these problems squarely in the realm of modern IT.

MASs have reached a level of maturity in which they're now entrusted with spacecraft control, medical-record processing, military-mission scheduling, supply-network management, and multimillion-dollar-auction staging. MASs have affected several other areas as well, including entertainment, education, training simulations, mechanism design, distributed constraint satisfaction, user interfaces, and agent-based models of social networks and human organizations. In spite of these successes, MASs haven't yet become the mandatory architectural approach for information system construction. Fundamental problems remain, ranging from engineering individual agents to scaling them to open, enterprise-wide applications. Several core MAS challenges, arising in concepts such as autonomy, cooperation, commitments, and joint action, continue to pose challenges.

Elements of an MAS Vision

The roadmap for multiagent-based SOC research, devised at a retreat we attended in May 2005 in South Carolina, describes the research challenges that must be met before developers can produce application classes such as pervasive service environments, society-inspired systems, and computational service mechanisms.

Pervasive Service Environments

Beyond today's SOC installations, an obvious next step is the development of pervasive service envi-

ronments, in which services are widely available in everyday home, office, and play environments. In the future, independently developed services will be dynamically selected, engaged, composed, and executed in a context-sensitive manner. Such services could support several applications, including

- heterogeneous information management within and across enterprises, thereby facilitating superior process management and e-business;
- scientific computing with large, dynamically reconfigurable resources (such as in grid computing), thereby supporting the solution of grander scientific problems;
- mobile computing, in which mobile users obtain desired information at the right time from fixed or mobile information resources; and
- pervasive computing, in which computational resources are associated with components of the (physical) infrastructure that surrounds us, thereby leading to optimized management of resources and an improved user experience.

Instead of passively waiting for discovery, services could proactively contribute to applications, thereby behaving like agents in an MAS.

Society-Inspired Systems

Societal representations of large-scale systems facilitate the exploration and understanding of relationships between elements of our world whose complexity has kept them a mystery. Such representations are appropriate for a vast range of system domains, including

- global environmental phenomena, such as climate change and extreme weather prediction and detection;
- biological networks, such as tools for epidemiological modeling (for example, contagion of bird flu or mad-cow disease); and
- computational pathologies, such as viruses or spam.

Large-scale simulations inform decision making by enabling "what if" analyses that help people understand the consequences of possible military, economic, political, or environmental actions. Understanding the effect of mobile phone usage on traffic accidents in a given area, for example, could help choose between banning cell phone usage while driving and lowering the speed limit. MASs improve the verisimilitude of such simulations

because agents have more of the same characteristics of the entities involved in the simulation.

Computational Service Mechanisms

As services become increasingly "alive" and their interactions become increasingly dynamic, they'll begin to do more than just manage information in explicitly programmed ways. In particular, MASs or services acting in concert can function as computational mechanisms in their own right, thus significantly enhancing our ability to model, design, build, and manage complex software systems. Think of such MASs as providing a new approach for constructing complex applications wherein developers concentrate on high-level abstractions, such as overall behavior and key conceptual structures (the active entities, their objectives, and their interactions), without having to go further into individual agents' details or interactions.

This vision becomes more compelling as the target environments become more

- populous (a monolithic model is intractable, whereas developers can construct an MAS modularly);
- distributed (pulling information to a central location for monitoring and control is prohibitive, whereas techniques based on interaction among agents and the emergence of desired system-level behaviors are much easier to manage); and
- dynamic (an MAS can adapt in real time to changes in the target system and the environment in which it is embedded).

Ashby's principle of requisite variety[2] states that a system's controller must have complexity at least equal to the system itself, or "every good regulator of a system must be a model of that system."[3] From this perspective, using an MAS to manage complex distributed systems isn't just feasible, but necessary.

Table 1 shows the ways in which MAS properties can benefit complex system engineering. Potential applications and application domains that can also benefit from an MAS approach include meeting scheduling, scientific workflow management, distributed inventory control and supply chains, air and ground traffic control, telecommunications, electric power distribution, water supplies, and weapon systems.

Past as Prologue

To see where MASs are headed (and thus predict

Table 1. Reasons for complex system development based on multiagent systems.

Multiagent system properties	Benefits for system development
Autonomous, objective-oriented behavior; agent-oriented decomposition	Autonomous, active functionality that adapts to the user's needs; reuse of whole subsystems and flexible interactions
Dynamic composition and customization	Scalability
Interaction abstractions; statistical or probabilistic protocols	Friction-free software; open systems; interactions among heterogeneous systems; move from sophisticated and learned e-commerce protocols to dynamic selection of protocols
Multiple viewpoints, negotiation, and collaboration	Robustness and reliability
Social abstractions	High-level modeling abstractions

SOC's future), we must look at what they can accomplish today. In agent-based software engineering, MASs form the fundamental building blocks of software systems, even those systems that don't require agent-like behaviors.[4,5] Another success is in simulation technology: modern MAS simulation platforms can support 10^4 to 10^5 concurrent agents. By modeling such a large number of active components, an MAS-based simulation helps us understand system-level behavior in settings with nonlinear interactions among many parts, in which behavior generally can't be understood analytically.

To date, a sizable body of MAS research hasn't yet made many inroads into conventional system development:

- *Simulated evolution.* Defining individual agent behaviors that will yield desired system behavior is analytically intractable, but methods are maturing by which agents can evolve with respect to fitness functions, thus reducing the engineering burden.
- *Stigmergic methods* are communication methods in emergent systems in which the individual parts of the system communicate with each other by modifying their local environment, such as how ants communicate with each other by depositing pheromones along their paths. Our understanding of how environmentally mediated interactions among agents can yield emergent coordination is growing; such interactions have the added benefit of enforcing locality and thus ensuring tractability.
- *Consensus software for robustness.*[6,7] MASs have the potential to revolutionize the way in which software is produced, developed, and executed — simultaneously improving large-scale, mission-critical, and complex system reliability. Initial experiments show an improvement in robustness due to redundancy.
- *Frameworks for describing and controlling societal-level computations.* Controlling a large-scale MAS requires a distributed means for assessing and managing computation by an agent society. TAEMS (Task Analysis and Environment Modeling System) provides a mechanism that could enable the realization, progress, and achievement of societal-level goals by helping agents relate their own tasks to those of other agents in their society via formally defined hard and soft relationships.
- *Engineering tools and frameworks for managing MASs.* The Multi-Agent Survivability Simulator (MASS) and Multi-Agent Computing Environment (MACE3J) are successful environments for designing and simulating society-level MASs.[8] These technologies focus on real-world social variables.

In disciplines such as economics, sociology, and marketing, MASs serve as models for understanding the behavior of highly populous, distributed, dynamic systems. Decision makers could use them to design city-wide traffic controls, choose among distribution schemes for water and electricity in large utility districts, and manage trade relationships among nations.

Challenges

MAS and SOC researchers must address several challenges to realize the applications we've described. We won't discuss infrastructure challenges here because processing and communication capacity, as well as that of sensors and effectors, improves daily.

The development of large-scale MASs calls for new engineering methods. Such systems must be self-organizing and support runtime reconfiguration and design. Techniques such as evolutionary computation or population-based search could serve as design tools for building MASs.

Interaction should be given primacy in terms of representation and reasoning. To develop open sys-

tems, we must capture interactions independently of the components that might ultimately perform those interactions. Along these lines, we must also develop notions of commitments, protocols, and new ways of modeling and specifying desirable interaction goals; understanding teamwork and coordinated problem solving and how both can help make a service-oriented system more robust in the face of various exceptions; and monitoring the behavior of large-scale systems composed of heterogeneous components with their own aims and objectives.

Current multiagent and service-oriented systems operate according to fixed protocols, but many application domains are uncertain and don't have that luxury. This requires additional work on probabilistic discourse, negotiation, and interaction protocols (here, protocols are considered to be statistical entities) — specifically, embedded instructions (systems should present interaction norms to users and provide for discoverable interaction, negotiation, and discourse protocols) and system-wide properties and emergent phenomena.

Predicting the behavior of large-scale agent systems is extremely difficult — sometimes impossible — because of emergent behavior. Tools and techniques from statistical physics could help, because concepts such as phase transitions, universality, entropy, and convergence have analogs in MASs. Network theory is relevant as well, because of its study of the interplay between structure and interaction, as is sociology (for understanding social dynamics).

The human aspects of these technologies — such as modeling emotion and incorporating it into social-system simulations — are crucial for ensuring that people are comfortable with an expanded role for agents. Key trade-offs between trust and autonomy inform adjustable autonomy. Likewise, privacy is inherently tied to social norms and expectations, which can be expected to evolve as people reassess the risk/benefit trade-off between trust and autonomy.

Future research will extend MASs in scale, heterogeneity, embodiment in the physical world, and lifetime, and as MASs take SOC into new dimensions we'll be able to explore, understand, and control the complex and intertwined relationships of the real world in unimaginable new ways.

A large community of academic and industrial researchers is poised to conduct the research outlined in this article, and much of it will likely be accomplished within five years. The most important MAS and SOC applications in the near term will be in focused domains, such as the interpretation of information from sensor networks, in which the MAS already has a decision-support role. As a result, the human-agent-robotic interface will require a means for modeling, enacting, and monitoring high-level protocols. We'll also need models of reputation and trust that agents can maintain by themselves in a distributed fashion. Such models could assist with service location and selection based on additional considerations such as provenance and empirical evaluation of information sources.

We'll also see the emergence of a stable infrastructure for large-scale, multiagent simulation and control. Agents will represent active entities (such as people and traffic controllers), animate the inanimate parts of an environment (such as retail goods on a store shelf), represent individuals' interests (such as in e-commerce or auctions), and even model and represent environments. For the special case of e-government, agents representing constituents could help elected officials gauge their constituents' opinions and preferences, resulting in better buy-in for legislated decisions.

Within 10 years from now, the scope of MAS and SOC applications will increase in terms of functionality (a smart house, for example, is a multifunction system) or geography (such as intelligent highway systems and alert forwarding). For such applications to scale to realistic size, we'll need concepts and techniques for teamwork that can be applied routinely, including negotiation among team members, planning and replanning in the face of environmental uncertainty, and dynamism. Adaptive, incentive-based mechanisms for producing trustworthy behavior will also be crucial.

In 15 years, we can expect to see complete incorporation of the MAS into the system it supports, producing an awareness of its system and itself that would let it detect and correct any misbehavior. To realize this full immersion, we'll need to understand system-level properties and social dynamics with respect to adaptive corrective behaviors and reputation-based trust. We'll achieve this understanding and awareness via effective models of social systems that let developers adjust simulations to align them better with the real world. In such contexts, our concern will be with ethical agents that operate according to articulated and desirable philosophies. Ethics will also characterize the behavior of successful services as they interact and collaborate with each other in SOC.

References

1. M. Singh and M. Huhns, *Service-Oriented Computing: Semantics, Processes, Agents*, John Wiley & Sons, 2005.
2. W.R. Ashby, "Requisite Variety and Its Implications for the Control of Complex Systems," *Cybernetica*, vol. 1, no. 2, 1958, pp. 83–99.
3. R.C. Conant and W.R. Ashby, "Every Good Regulator of a System Must Be a Model of that System," *Int'l J. Systems Science*, vol. 1, no. 2, 1970, pp. 89–97.
4. N.R. Jennings and M. Wooldridge, eds., *Agent Technology: Foundations, Applications and Markets*, Springer-Verlag, 1998.
5. H.V.D. Parunak, "Industrial and Practical Applications of Distributed AI," *Multi-Agent Systems*, G. Weiss, ed., MIT Press, 1999, pp. 377–421.
6. V.T. Holderfield, M.N. Huhns, and R.L. Zavala, "Achieving Software Robustness Via Large-Scale Multiagent Systems," *Software Eng. for Large-Scale Multi-Agent Systems*, LNCS 2603, Springer-Verlag, 2003, pp. 199–215.
7. R.L. Zavala and M.N. Huhns, "On Building Robust Web Service-Based Applications," *Extending Web Services Technologies: The Use of Multi-Agent Approaches*, L. Cavedon et al., eds., Kluwer Academic, 2004, pp. 293–310.
8. L. Gasser and K. Kakugawa, "MACE3J: Fast Flexible Distributed Simulation of Large, Large-Grain Multi-Agent Systems," *Proc. 1st Int'l Conf. Autonomous Agents and Multi Agent Systems*, ACM Press, 2002, pp. 745–752.

Michael N. Huhns is the NCR professor of computer science and engineering at the University of South Carolina. Contact him at huhns@sc.edu.

Munindar P. Singh is a professor of computer science at North Carolina State University. Contact him at singh@ncsu.edu.

Mark Burstein is a division scientist and director of the Human Centered Computing Group at BBN Technologies. Contact him at burstein@bbn.com.

Keith Decker is an associate professor in computer and information sciences at the University of Delaware. Contact him at decker@cis.udel.edu.

Edmund Durfee is a professor of electrical engineering and computer science at the University of Michigan. Contact him at durfee@umich.edu.

Tim Finin is a professor of computer science and electrical engineering at the University of Maryland Baltimore County. Contact him at finin@umbc.edu.

Les Gasser is an associate professor at the University of Illinois. Contact him at gasser@uiuc.edu.

Hrishikesh Goradia is a PhD candidate in computer science at the University of South Carolina. Contact him at goradia@engr.sc.edu.

Nick Jennings is a professor in the School of Electronics and Computer Science at Southampton University. Contact him at nrj@ecs.soton.ac.uk.

Kiran Lakkaraju is a PhD candidate in computer science at the University of Illinois. Contact him at klakkara@uiuc.edu.

Hideyuki Nakashima is the president of Future University in Hakodate, Japan. Contact him at h.nakashima@aist.go.jp.

H. Van Dyke Parunak is the chief scientist for Altarum in Ann Arbor, Michigan. Contact him at van.parunak@altarum.org.

Jeffrey S. Rosenschein is an associate professor and Chairman of Studies for Computer Engineering at Hebrew University. Contact him at jeff@cs.huji.ac.il.

Alicia Ruvinsky is a PhD candidate in computer science at the University of South Carolina. Contact her at ruvinsky@engr.sc.edu.

Gita Sukthankar is a PhD student in robotics at Carnegie Mellon University. Contact her at gitars@cs.cmu.edu.

Samarth Swarup is a PhD candidate in computer science at the University of Illinois. Contact him at swarup@uiuc.edu.

Katia Sycara is a research professor at Carnegie Mellon University and holds the Sixth Century Chair in Computing Science at the University of Aberdeen. Contact her at katia@cs.cmu.edu.

Milind Tambe is an associate professor in computer science at the University of Southern California. Contact him at tambe@usc.edu.

Tom Wagner is a program manager in the IPTO office at Darpa. Contact him at twagner@darpa.mil.

Laura Zavala is a PhD candidate in computer science at the University of South Carolina. Contact her at zavalagu@engr.sc.edu.