The Art of Agent-Oriented Modeling

Leon S. Sterling and Kuldar Taveter

The Art of Agent-Oriented Modeling

Intelligent Robotics and Autonomous Agents
Edited by Ronald C. Arkin

Dorigo, Marco, and Marco Colombetti, Robot Shaping: *An Experiment in Behavior Engineering*

Arkin, Ronald C., *Behavior-Based Robotics*

Stone, Peter, *Layered Learning in Multiagent Systems: A Winning Approach to Robotic Soccer*

Wooldridge, Michael, *Reasoning about Rational Agents*

Murphy, Robin R., *An Introduction to AI Robotics*

Mason, Matthew T., *Mechanics of Robotic Manipulation*

Kraus, Sarit, *Strategic Negotiation in Multiagent Environments*

Nolfi, Stefano, and Dario Floreano, *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*

Siegwart, Roland, and Illah R. Nourbakhsh, *Introduction to Autonomous Mobile Robots*

Breazeal, Cynthia L., *Designing Sociable Robots*

Bekey, George A., *Autonomous Robots: From Biological Inspiration to Implementation and Control*

Choset, Howie, Kevin M. Lynch, Seth Hutchinson, George Kantor, Wolfram Burgard, Lydia E. Kavraki, and Sebastian Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementations*

Thrun, Sebastian, Wolfram Burgard, and Dieter Fox, *Probabilistic Robotics*

Mataric, Maja J., *The Robotics Primer*

Wellman, Michael P., Amy Greenwald, and Peter Stone, *Autonomous Bidding Agents: Strategies and Lessons from the Trading Agent Competition*

Floreano, Dario, and Claudio Mattiussi, *Bio-Inspired Artificial Intelligence: Theories, Methods, and Technologies*

Sterling, Leon S., and Kuldar Taveter, *The Art of Agent-Oriented Modeling*

The Art of Agent-Oriented Modeling

Leon Sterling and Kuldar Taveter

The MIT Press
Cambridge, Massachusetts
London, England

This book was set in Times New Roman on 3B2 by Asco Typesetters, Hong Kong.
Printed and bound in the United States of America.

Library of Congress Cataloging-in-Publication Data

Sterling, Leon.
The art of agent-oriented modeling / Leon S. Sterling and Kuldar Taveter.
p. cm. — (Intelligent robotics and autonomous agents series)
Includes bibliographical references and index.
ISBN 978-0-262-01311-6 (hardcover : alk. paper) 1. Intelligent agents (Computer software) 2. Computer software—Development. I. Taveter, Kuldar. II. Title.
QA76.76.I58S757 2009
006.3—dc22 2008044231

10 9 8 7 6 5 4 3 2 1

To Miriam, Danya, Sara, and Emily

To Siiri, Eliise, and Sanne

Contents

Foreword

The software systems that we need to run our society in the twenty-first century are becoming increasingly complex. We need to build large systems by integrating existing systems and components from a range of suppliers. These systems have to be responsive, dependable, and secure. Our reductionist, top-down software engineering techniques—which have helped us create the large, reliable software systems that we depend upon—are now struggling to cope. To meet the demands for twenty-first century systems, we need new approaches to software engineering that can cope autonomously with a rapidly changing environment and that can take advantage of new features as they become available. We need systems that can cope with the unexpected—in terms of both reacting autonomously to problems and communicating these problems to other agents in the system. Agent-oriented software engineering is an important emerging technology that can help us build such systems.

In any developing technology, books in the area go through stages. Initially, the only works are the proceedings of specialist conferences for researchers, where the book includes advanced research papers. The second stage is typified by multiauthor books and specialist books aimed at researchers and early adopters, which focus on a specific topic. These works assume that readers understand basic concepts and principles. Finally, student textbooks are written, aimed at people who need to start from the beginning. Agent-oriented software engineering has now reached this final stage of maturity. This book is one of the first textbooks on agent-oriented software engineering. It aims to disseminate knowledge about this important topic to a wide audience.

The Art of Agent-Oriented Modeling is an introduction to agent-oriented software development for students and for software developers who are interested in learning about new software engineering techniques. Although the principal focus of the book is agent-oriented modeling, it is an excellent general introduction to all aspects of practical, agent-oriented software engineering. Building on Leon Sterling's and Kuldar Taveter's courses in this topic, this book uses everyday examples to illustrate the notion of agents and how agents can interact to create complex, distributed systems.

A key difference between this book and other books in the field is that this book recognizes that if agent-oriented development is to enter the mainstream, it must be done with consideration of practical software engineering issues. As well as delivering the required functionality, agent-oriented systems have to be reliable and secure; their development has to fit into a software engineering life cycle, where systems must meet externally imposed requirements and have to be maintained over many years.

This book has many important insights and it is perhaps invidious to pick on only one of them as an example. However, I think the key point that Leon and Kuldar make in this book is that agent-oriented technology is an effective way to construct sociotechnical systems, which take into account organizational and human issues as well as technical issues. Agents are a useful abstraction that helps us think about how sociotechnical systems deliver the services required by their users. We should think about this without regard to whether the agents that deliver these services are people or automated systems. Modeling a system as agents gives us a way of establishing flexible boundaries for the automated system that can be extended as our understanding of human tasks improve. Over time, these systems can evolve with more and more functionality being taken over by automated agents.

Barry Boehm, one of the most respected software engineering researchers and practitioners, has called the twenty-first century the "software century." The challenges for humanity are to improve the lives of people around the world without making unsustainable demands on the environment. We can achieve this only with the help of large and small software systems that coordinate resources from different places and different providers and that break down cultural barriers between different civilizations. Many of these systems will be agent-oriented and Leon Sterling and Kuldar Taveter have made an important contribution in this book to the development of agent-oriented software engineering.

Ian Sommerville, *St. Andrews, Scotland, July 2008*

Preface

The introduction of the personal computer changed society's view of computer systems. No longer was a computer a complicated machine doing a task, tucked away in a separate room or a far-off corner of an office or work environment. A computer was a machine on your desk that was ready for daily interaction. No longer was a computer only for specialist programmers—it was a tool to be used by virtually all office workers.

The rapid rise of the World Wide Web in the mid-1990s changed society's view still further. No longer was the personal computer restricted to a desk or a small office network. The machine on your desk could be part of a global network. In addition, the use of computers at home skyrocketed. Almost every home in the developed world acquired a computer. Laptops were mandated for school children in many schools. The volume and visibility of email addresses and domain names exploded.

In the mid-2000s, as this book is written, our interaction with computers is ever-expanding. Businesses depend on software for many—if not most—of their core activities. Cars and appliances have embedded software of which we are barely aware. Teenagers and young adults engage in social networking, and share music and videos on Web sites such as LimeWire and YouTube. We look up knowledge on Wikipedia. Skype is a common method for making telephone calls over the Internet.

So how do we develop computer software to interact with the ever-increasing complexity of the technical world and the increased fluidity of social organizations? It is not an easy task. Having been involved in research and teaching of software engineering and software development projects, we find a need for changing conceptual models. We want to develop software that is open, intelligent, and adaptive. We are becoming convinced that such software needs to be conceived, designed, and developed differently from existing applications that assume fixed requirements and an unchanging environment.

One consideration we believe is important is the need to take a systems view. The system of people and technology has a life cycle of inception, design, implementation, testing, and maintenance. Processes need to be understood for each of the stages of the life cycle. A systems view implies being able to take a holistic approach and to include considerations of all stakeholders.

In this book, we advocate an agent-oriented view. An *agent* is a concept and metaphor that is familiar for people with and without technical background or interest. An agent allows the blurring of the boundary between people and technology to allow more people to be engaged with the process of building software. Thinking in terms of agents can change the way that people think of software and the tasks it can perform.

The concept of an agent is not new to computing. Agents have been discussed in regard to artificial intelligence (AI) from its early days. More recently, agents have been discussed in the context of object-oriented technology, with many researchers and practitioners viewing them as an extension of object-oriented technology. Agent models have been added to established object-oriented modeling approaches, most notably based around UML (Unified Modeling Language). Agent programming extensions have been developed for languages such as Java and Prolog.

We prefer viewing an agent-oriented approach as different from an object-oriented approach. Agent-oriented models need to be described independently of object-oriented models, though sensible reuse of notation is advisable. The reason for differentiation is that people can be sometimes trapped into thinking only in ways familiar to them. As the saying goes, if all you have is a hammer, everything looks like a nail. In our experience, thinking in terms of agents requires a different mindset. Students and practitioners who insist in thinking about building from their existing (object-oriented) methods are less successful in developing agent-oriented applications.

Our interest in writing this book is to encourage a wide variety of stakeholders in the software development process to engage with an agent-oriented approach. This is a challenge. Software developers, software acquirers, software users, and software maintainers have different concerns. Models are needed that are understandable for all stakeholders. The variety of concerns suggests a variety of models. If the models are understandable, the stakeholder is more likely to be engaged. We believe that engagement throughout the process will improve the development of systems consisting of people and software that will interoperate and evolve successfully.

So why another book? One factor in the adoption of a technology is having textbook material in support. There has been an increase in agent-oriented theory, language, and methodology books over the past several years. This progress is to be encouraged. Nonetheless we feel that this book fills a niche not currently being filled.

There are three main features of this book that we feel add significantly to the literature. The first is the presentation of substantial new examples that range from requirements through design to deployment. The more examples that potential users of technology see, the more likely a technology is to be adopted. The second is the use of abstract agent-oriented models at various levels of abstraction and geared to different stakeholders. The models relate to a range of agent-oriented software engineering methodologies that are being actively developed. The third is related—namely, the emphasis on models rather than languages—and the philosophy that models can be mapped to a variety of deployment platforms, possibly including conventional languages, makes the concepts accessible to a wider audience. Our recent discussions with colleagues in both academia and industry have reinforced our appreciation for the need for different models for open, distributed domains.

There are often forces working against trying to show how an approach covers a range of implementation languages. Agent language vendors want companies to be locked into their technology. Indeed, one of the initial motivations for the research underlying this book was a request by an industrial partner on what agent models would be "future-proof." The industry partner had been burned by using an agent-based vendor platform that was no longer supported. Research groups also want to encourage others to use their specific methods. Though this motivation is understandable, the agent community will benefit from broadly supportable models.

Why do we use the word "art" in the title of the book? We have experienced modeling as a creative process that does not always follow clear-cut rules. Many decisions are left to the discretion of the modeler and his or her background and intuition. Modeling is also iterative. Usually one cannot—and should not—end up with final models right away. Even some of the models in the book would improve through further rounds of iteration. Given that we cannot give definitive advice on building "perfect" models, we settle for the more modest aim of providing guidance for creating models.

Having explained some of the motivation that prompted the book, we address the issue of timing. Why now? We have each been working with agent applications for the past ten years. Only now, in our opinion, do we perceive that agent concepts are known widely enough to allow agent-oriented modeling. Modeling tools are emerging: an essential development, if agent concepts are to be used by industry. Also, recent developments in the research areas of autonomic systems, event-based systems, and multiagent systems seem to indicate that the paradigm of peer-to-peer computing in the broad sense is gaining momentum.

Our conclusion that agent concepts have matured sufficiently has been developed through our delivering of seminars and courses on agent technology in academia and industry. We have been involved in teaching graduate students (and advanced undergraduates) for more than ten years. In 2000, Leon was involved in an Australian government project on the rapid diffusion of technology from academia to industry using agent technology as an example. With the benefit of hindsight, the project was premature. Providing students with the models presented in this book has greatly increased their ability to conceptualize and design agent-oriented systems. The models have been progressively polished over the past three years and are now ready for broader engagement. By the time the reader finishes reading this book, we hope that his or her capacity and confidence for the design and modeling of agent-oriented systems will have improved.

Leon Sterling, *Melbourne, Australia, November 2008*
Kuldar Taveter, *Tallinn, Estonia, November 2008*

Acknowledgments

Any book that is of this length and that has taken almost two years to write owes its appearance to many people, events, and circumstances. We gladly take this opportunity to acknowledge and thank the many people who have contributed to this book, in ways both big and small. Doubtless there will be people who have contributed that we have unintentionally omitted from mention. For that we request forgiveness in advance.

First and foremost, we acknowledge the dedication and support of our families. Leon appreciates the love and support of his wife Miriam over his career, and the interactions with his daughters Danya, Sara, and Emily. They have had a positive impact on the book in many ways, including influencing examples.

Kuldar appreciates the love and patience of his wife Siiri, who, not too long after Kuldar's Ph.D. thesis writing process, went through a similar experience. Of enormous importance for the content of this book were Kuldar's interactions with his daughters Eliise and Sanne, now 10 and 7, who at the request of Daddy made experiments with Tamagotchis and consulted Daddy. Kuldar is also thankful to his parents for their prayers and thoughts. And last but not least, Kuldar thanks God for blessing the authors and their families and for providing us with the opportunity to write this book.

We next thank our colleagues who have been instrumental in the appearance of this book. For Leon, the story starts with his return to Melbourne to a professorship in Computer Science and the establishment of the Intelligent Agent Lab, jointly with Professor Liz Sonenberg. Liz and Leon co-taught an Intelligent Agents graduate class in 1996, and to a large extent, the origins of the book lie in the lack of an appropriate textbook for that class. Liz has been a tremendous colleague and indirectly affected the book in many ways. Over the past twelve years, many people have influenced the book's direction. Silvio Salom from Adacel Technologies gave substantial support. The Intelligent Home examples arose from a consulting project with Silvio and many related discussions. The ROADMAP methodology was a response to Silvio's request for a vendor-free agent methodology. There was extensive interaction

with the Australian Defence Science and Technology Organisation (DSTO), especially with Simon Goss, and later Clint Heinze and Michael Papasimeon. Adrian Pearce joined the Melbourne Agent Lab in 2000 and formed part of the formative discussions with DSTO. The University of Melbourne Software Engineering group deserves thanks, especially Philip Dart, Ed Kazmierczak, and more recently Shanika Karunasekera. All have been good colleagues who have contributed to Leon's perspective on software engineering. Two software engineering students who later became Ph.D. students, Thomas Juan and Kendall Lister, deserve special mention. This book definitely would not have happened without Thomas's Ph.D. research and surrounding discussions about agent-oriented software engineering methodologies. At RMIT University down the road, Lin Padgham and Michael Winikoff followed some similar paths, and discussions with them have undoubtedly contributed to the book. More recently, we have had helpful interaction with Knobby Clarke from Oakton Computing, previously Agentis, who gave useful feedback on our agent concepts.

Internationally, a special thanks to the Italian Logic Programming Community for inviting Leon to the Italian summer school in 1997, which initiated an ongoing collaboration with researchers at the University of Genoa led by Maurizio Martelli, and supported by Viviana Mascardi and Floriano Zini during their Ph.D. research. Leon has also enjoyed attending two workshops at Dagstuhl in Germany, at the first of which Leon met Kuldar.

For Kuldar, his initial interest in agent-oriented modeling was focused and shaped by Gerd Wagner, currently a professor at Brandenburg University of Technology in Cottbus, Germany. Their collaboration started in 1999 and Gerd later became a co-supervisor of Kuldar's Ph.D. thesis. The cooperation with Gerd over many years has been very influential in the emergence of this book, and is certainly reflected by the contents of the book. It is also not possible to underestimate the contribution of the Academician Boris Tamm, the late supervisor of Kuldar's thesis. He helped Kuldar in getting the real-life case study of manufacturing simulation, which is also present in this book, and was also instrumental in obtaining research funding from the Estonian Science Foundation.

Kuldar would also like to thank his former colleagues at the Technical Research Centre of Finland, VTT Information Technology, who have been involved in forming Kuldar's industry-oriented research attitude. Especially deserves mentioning Aarno Lehtola, now an entrepreneur, who has been very supportive over many years. Other supportive colleagues from VTT Information Technology include Pekka Silvennoinen, Seppo Linnainmaa, and Markus Tallgren.

During the past three years, Kuldar has enormously enjoyed the work atmosphere at the Agent Lab of the University of Melbourne. Particularly his discussions with Leon have been extremely enlightening with respect to the book and to agent-

oriented software engineering more broadly. Cooperation with the Lochard and Jeppesen companies in Melbourne has been a good sequel to Kuldar's earlier innovation projects. Kuldar would also like to thank Professors Brian Henderson-Sellers and Aditya Ghose from Australia, Dr. Jens Dietrich from New Zealand, and Professor Eric Yu from Canada for the opportunities to give seminars and receive useful feedback at their universities.

Funding from several research organizations has contributed to Leon's and Kuldar's collaboration. Kuldar was supported as a research fellow through the Australian Research Council Centre for Perceptive and Intelligent Machines in Complex Environments. Funding was also received from the Australian Research Council Discovery and Linkage programs, the Smart Internet Technology Cooperative Research Centre, and Adacel Technology. Earlier, Kuldar's research projects, two of which are overviewed as case studies in this book, were on several occasions supported by the Technology Agency of Finland and VTT Information Technology and once by the Estonian Science Foundation.

Next, we thank research students. In addition to Thomas Juan and Kendall Lister, mentioned earlier, we thank Kevin Chan, Ayodele Oluyomi, Anne Boettcher, Yu Xiu Luo, Bin Lu, Wai Shiang Cheah, Joost Westra, Muthukuruppan Annamalai, Maia Hristozova, and Nicole Ronald. The following research students read drafts of the book and made useful comments: Mohammed Arif, Bin Lu, Yu Xiu Luo, Golriz Rezaei, and Rebekah Velicescu. The coursework and research students who attended the evolving Intelligent Agents class are too many to mention, but their influence is noted. Finally, thanks to Alison Handley who helped with last-minute formatting, and who has been a presence for many years.

We would also like to thank the following external proofreaders who made many useful remarks: Knobby Clarke, Brian Berenbach, Dragan Gasevic, and Jeni Paay. Help with MaSE figures was provided by Scott DeLoach and help with Tropos figures was provided by Anna Perini. A couple of proofreaders deserve special mention. The comments and suggestions by Giancarlo Guizzardi considerably improved the ontological grounding for modeling given in the book. Also, proofreading by the Emeritus Professor Brian V. Hill produced many constructive remarks and proved that the book is also understandable for an audience without a background in information technology.

We would like to thank the publisher and the proofreader for many small improvements. We also acknowledge Skype by Skype Technologies S.A. for enabling us to continue interaction once Kuldar returned to Estonia.

I MODELS

This book presents an approach for modeling complex systems consisting of people, devices, and software agents in changing environments. In part I, a set of models is presented at three levels of abstraction: a motivation layer, where the purpose, goals, and requirements of the system are described; a design layer; and an implementation layer. The first chapter advocates conceiving of the world in which software operates as a multiagent system operating in an environment subject to rules and policies. The conceptual space that we look at will be discussed in more detail in chapter 2. The models themselves will be presented in chapter 3 and applied in later chapters. Chapter 4 focuses on nonfunctional requirements or quality goals. Chapter 5 describes four different platforms, and chapter 6 presents a viewpoint framework that is used in part II to analyze methodologies.

References

Arnold, K., Gosling, J., and Holmes, D. 2005. *The Java Programming Language*. 4th edition. Reading, MA: Addison-Wesley.

Asimov, I. 1950. *I, Robot*. New York: Gnome Press.

Bauer, B., and J. Odell. 2005. UML 2.0 and agents: How to build agent-based systems with the new UML standard. *Journal of Engineering Applications of Artificial Intelligence*, *18*(2), 141–157.

Baylor, A. L. 2001. Investigating multiple pedagogical perspectives through MIMIC (Multiple Intelligent Mentors Instructing Collaboratively). In J. D. Moore, C. L. Redfield, and W. L. Johnson (eds.), *Artificial Intelligence in Education: AI-ED in the Wired and Wireless Future*. Amsterdam, The Netherlands: IOS Press.

Bellifemine, F., G. Caire, and D. Greenwood. 2005. *Developing multiagent systems with JADE*. Chichester, UK: John Wiley and Sons.

Bellifemine, F., A. Poggi, and G. Rimassa. 2001. Developing multiagent systems with a FIPA-compliant agent framework. *Software—Practice and Experience*, *31*(2), 103–128.

Bergenti, F., M.-P. Gleizes, and F. Zambonelli. 2004. *Methodologies and Software Engineering for Agent Systems: The Agent-Oriented Software Engineering Handbook*. Norwell, MA: Kluwer Publishing.

Blaha, M., and J. Rumbaugh. 2005. *Object-Oriented Modeling and Design with UML*. Upper Saddle River, NJ: Prentice Hall.

Blake, M. B. 2002. B2B electronic commerce: Where do agents fit in? In M. B. Blake (ed.), *Agent-Based Technologies for B2B Electronic Commerce: Papers from the AAAI Workshop, Technical Report WS-02-01* (1–8). Menlo Park, CA: American Association for Artificial Intelligence.

Bordini, R. H., J. F. Hübner, and R. Vieira. 2005. Jason and the golden fleece of agent-oriented programming. In R. H. Bordini, M. Dastani, J. Dix, and A. E. F. Seghrouchni (eds.), *Multiagent Programming: Languages, Platforms, and Applications* (3–37). Berlin, Germany: Springer-Verlag.

Bratman, M. E. 1987. *Intentions, Plans, and Practical Reason*. Stanford, CA: CSLI Publications.

Bresciani, P., A. Perini, P. Giorgini, F. Giunchiglia, and J. Mylopoulos. 2001. A knowledge level software engineering methodology for agent oriented programming. In *Proceedings of the Fifth International Conference on Autonomous Agents*, May 28–June 1, Montreal, Quebec, Canada (648–655). New York: ACM.

Bresciani, P., A. Perini, P. Giorgini, F. Giunchiglia, and J. Mylopoulos. 2004. Tropos: An agent-oriented software development methodology. *Autonomous Agents and Multiagent Systems*, *8*(3), 203–236.

Brooks, R. 1991. Intelligence without representation. *Artificial Intelligence*, *47*(1–3), 139–159.

Bubenko, J. A. 1993. Extending the scope of information modelling. In A. Olivé (ed.), *Fourth International Workshop on the Deductive Approach to Information Systems and Databases, September 20–22, Lloret de Mar, Catalonia, Proceedings* (DAISD 1993). Report de recerca, LSI/93-25-R, Departament de Llenguatges i Sistemes Informatics (73–97). Barcelona, Spain: Universitat Politecnica de Catalunya (UPC).

Bubenko, J. A., Jr., and M. Kirikova. 1994. "Worlds" in requirements acquisition and modelling. In H. Kangassalo and B. Wangler (eds.), *Proceedings of the 4th European–Japanese Seminar on Information*

Modelling and Knowledge Bases, May 31–June 3, Stockholm, Sweden (159–174). Amsterdam, The Netherlands: IOS Press.

Caire, G., W. Coulier, F. Garijo, J. Gómez-Sanz, J. Pavón, P. Kearney, and P. Massonet. 2004. The MESSAGE methodology. In F. Bergenti, M.-P. Gleizes, and F. Zambonelli (eds.), *Methodologies and Software Engineering for Agent Systems: The Agent-Oriented Software Engineering Handbook* (177–194). Norwell, MA: Kluwer Publishing.

Cernuzzi, L., T. Juan, L. Sterling, and F. Zambonelli. 2004. The GAIA methodology: Basic concepts and extensions. In F. Bergenti, M.-P. Gleizes, and F. Zambonelli (eds.), *Methodologies and Software Engineering for Agent Systems: The Agent-Oriented Software Engineering Handbook* (69–88). Norwell, MA: Kluwer Publishing.

Chan, K., and L. Sterling. 2003a. Specifying roles within agent-oriented software engineering. In *Proceedings of the 10th Asia-Pacific Software Engineering Conference* (APSEC 2003), December 10–12, Chiang Mai, Thailand (390–395). Washington, DC: IEEE Computer Society.

Chan, K., and L. Sterling. 2003b. Light-weight agents for e-learning environments. In N. Zhong, Z. W. Ras, S. Tsumoto, and E. Suzuki (eds.), *Foundations of Intelligent Systems, 14th International Symposium, ISMIS 2003, Maebashi City, Japan, October 28–31, Proceedings* (LNAI 2871, 197–205). Berlin, Germany: Springer-Verlag.

Chen, P. P. 1976. The Entity-Relationship Model: Toward a unified view of data. *ACM Transactions on Database Systems*, *1*(1), 9–36.

Chung, L., B. Nixon, E. Yu, and J. Mylopoulos. 2000. *Non-functional Requirements in Software Engineering*. Norwell, MA: Kluwer Publishing.

Ciancarini, P., and M. Wooldridge. 2001. Agent-based software engineering: Guest editors' introduction. *International Journal of Software Engineering and Knowledge Engineering*, *11*(3), 205–206.

Clancey, W. J. 2002. Simulating activities: Relating motives, deliberation, and attentive coordination. *Cognitive Systems Research*, *3*(3), 471–499.

Cockburn, A. 2001. *Writing Effective Use Cases*. Reading, MA: Addison-Wesley.

Cossentino, M. 2005. From requirements to code with the PASSI methodology. In B. Henderson-Sellers and P. Giorgini (eds.), *Agent-Oriented Methodologies* (79–106). Hershey, PA: Idea Group.

Curtis, W., M. I. Kellner, and J. Over. 1992. Process modelling. *Communications of the ACM*, *35*(9), 75–90.

Dardenne, A., A. van Lamsweerde, and S. Fickas. 1993. Goal-directed requirements acquisition. *Science of Computer Programming*, *20*(1–2), 3–50.

Dastani, M., M. B. van Riemsdijk, and J.-J. Ch. Meyer. 2005. Programming multiagent systems in 3APL. In R. H. Bordini, M. Dastani, J. Dix, and A. E. F. Seghrouchni (eds.), *Multiagent Programming: Languages, Platforms, and Applications* (39–67). Berlin, Germany: Springer-Verlag.

DeLoach, S. A. 2008. Developing a multiagent conference management system using the O-MaSE process framework. In M. Luck and L. Padgham (eds.), *Agent-Oriented Software Engineering VIII: The 8th International Workshop on Agent Oriented Software Engineering, AOSE 2007, Honolulu, HI, May 14, 2007, Revised Selected Papers* (LNCS 4951, 168–181). Berlin, Germany: Springer-Verlag.

DeLoach, S. A., and M. Kumar. 2005. Multiagent systems engineering: An overview and case study. In B. Henderson-Sellers and P. Giorgini (eds.), *Agent-Oriented Methodologies* (317–340). Hershey, PA: Idea Group.

DeLoach, S. A., and J. L. Valenzuela. 2007. An agent-environment interaction model. In L. Padgham and F. Zambonelli (eds.), *Agent-Oriented Software Engineering VII: 7th International Workshop, AOSE 2006, Hakodate, Japan, May 8, 2006, Revised and Invited Papers* (LNCS 4405, 1–8). Berlin, Germany: Springer-Verlag.

DeLoach, S. A., M. F. Wood, and C. H. Sparkman. 2001. Multiagent systems engineering. *International Journal of Software Engineering and Knowledge Engineering*, *11*(3), 231–258.

De Wolf, T., and T. Holvoet. 2005. Emergence versus self-organisation: Different concepts but promising when combined. In S. Brueckner, G. Di Marzo Serugendo, A. Karageorgos, and R. Nagpal (eds.), *Engineering Self Organising Systems: Methodologies and Applications* (LNCS 3464, 1–15). Berlin, Germany: Springer-Verlag.

Dignum, V. 2004. A model for organizational interaction: Based on agents, founded in logic. Ph.D. thesis, Utrecht University, The Netherlands.

d'Inverno, M., and M. Luck. 2001. *Understanding Agent Systems*. Berlin, Germany: Springer-Verlag.

d'Inverno, M., M. Luck, M. Georgeff, D. Kinny, and M. Wooldridge. 2004. The dMARS architecture: A specification of the distributed multiagent reasoning system. *Journal of Autonomous Agents and Multiagent Systems*, *9*(1–2), 5–53.

Dornbush, S., K. Fisher, K. McKay, A. Prikhodko, and Z. Segall. 2005. XPOD: A human activity and emotion aware mobile music player. In *2nd International Conference on Mobile Technology, Applications and Systems*, November 15–17 (1–6). Washington, DC: IEEE Computer Society.

Farber, Barry. 2003. *Diamond Power: Gems of Wisdom from America's Greatest Marketer*. Franklin Lakes, NJ/Career Press.

Fielding, R. T. 2000. Architectural styles and the design of network-based software architectures. Ph.D. thesis, University of California, Irvine, CA.

FIPA ACL. 2002. *FIPA Agent Communication Language (ACL) Message Structure Specification*. Foundation for Intelligent Physical Agents (FIPA), Standard, December 3, 2002. Retrieved January 13, 2008, from http://www.fipa.org/specs/fipa00061/.

Firesmith, D. G., and B. Henderson-Sellers. 2002. *The OPEN Process Framework: An Introduction*. London, UK: Addison-Wesley.

Franklin, S., and A. Graesser. 1997. Is it an agent, or just a program? A taxonomy for autonomous agents. In J. P. Müller, M. J. Wooldridge, and N. R. Jennings (eds.), *Proceedings of the ECAI '96 Workshop on Agent Theories, Architectures, and Languages: Intelligent Agents III, Heidelberg, Germany, August 12–13, 1996* (LNAI 1193, 21–36). Berlin, Germany: Springer-Verlag.

Garcia-Ojeda, J. C., S. A. DeLoach, Robby, W. H. Oyenan, and J. Valenzuela. 2008. O-MaSE: A customizable approach to developing multiagent development processes. In M. Luck and L. Padgham (eds.), *Agent-Oriented Software Engineering VIII: The 8th International Workshop on Agent Oriented Software Engineering, AOSE 2007, Honolulu, HI, May 14, Revised Selected Papers* (LNCS 4951, 1–15). Berlin, Germany: Springer-Verlag.

Genesereth, M. R., and S. P. Ketchpel. 1994. Software agents. *Communication of the ACM*, *37*(7), 48–53.

Giorgini, P., Kolp, M., Mylopoulos, J., and Castro, J. 2005. Tropos: A requirements-driven methodology for agent-oriented software. In B. Henderson-Sellers and P. Giorgini (eds.), *Agent-Oriented Methodologies* (20–45). Hershey, PA: Idea Group.

Guarino, N., and Welty, C. A. 2001. A formal ontology of properties. In R. Dieng and O. Corby (eds.), *Knowledge Acquisition, Modelling and Management, 12th International Conference, EKAW 2000, Juan-les-Pins, France, October 2–6, Proceedings* (LNCS 1937, 97–112). Berlin, Germany: Springer-Verlag.

Guizzardi, G. 2005. *Ontological foundations for structural conceptual models*. Telematica Instituut Fundamental Research Series, No. 15. Enschede, The Netherlands.

Guizzardi, G., R. Falbo, and R. S. S. Guizzardi. 2008. Grounding software domain ontologies in the Unified Foundational Ontology (UFO): The case of the ODE Software Process Ontology. In *Proceedings of the XI Iberoamerican Workshop on Requirements Engineering and Software Environments (IDEAS 2008)*, February 11–15, Recife, Brazil (electronic edition).

Guizzardi, G., and G. Wagner. 2005a. Towards ontological foundations for agent modeling concepts using the Unified Foundational Ontology (UFO). In P. Bresciani, P. Giorgini, B. Henderson-Sellers, G. Low, and M. Winikoff (eds.), *Agent-Oriented Information Systems II, 6th International Bi-Conference Workshop, AOIS 2004, Riga, Latvia, June 8, 2004, and New York, NY, July 20, 2004, Revised Selected Papers* (LNAI 3508, 110–124). Berlin, Germany: Springer-Verlag.

Guizzardi, G., and G. Wagner. 2005b. Some applications of a unified foundational ontology in business modeling. In P. Green and M. Rosemann (eds.), *Business Systems Analysis with Ontologies* (345–376). Hershey, PA: Idea Group.

Guizzardi, G., G. Wagner, N. Guarino, and M. van Sinderen. 2004. An ontologically well-founded profile for UML conceptual models. In A. Persson and J. Stirna (eds.), *Advanced Information Systems Engineering, 16th International Conference, CAiSE 2004, Riga, Latvia, June 7–11* (LNCS 4408, 148–164). Berlin, Germany: Springer-Verlag.

Guizzardi, R. S. S. 2006. Agent-oriented constructivist knowledge management. Ph.D. thesis, Centre For Telematics and Information Technology, University of Twente, The Netherlands.

Guizzardi, R. S. S., and G. Guizzardi. 2008. Ontology-based transformation framework from Tropos to AORML. In P. Giorgini, N. Maiden, J. Mylopoulos, and E. Yu (eds.), *Tropos/i*: Applications, Variations and Extensions*, Cooperative Information Systems Series. Cambridge, MA, and London, England: MIT Press.

Guizzardi, R. S. S., G. Guizzardi, A. Perini, and J. Mylopoulos. 2007. Towards an ontological account of agent oriented goals. In R. Choren, A. Garcia, H. Giese, H. Leung, C. Lucena, and A. Romanovsky (eds.), *Software Engineering for Multiagent Systems V: Research Issues and Practical Applications* (LNCS 4408, 148–164). Berlin, Germany: Springer-Verlag.

Heinze, C., M. Cross, S. Goss, T. Josefsson, I. Lloyd, G. Murray, M. Papasimeon, and M. Turner. 2002. Agents of change: The impact of intelligent agent technology on the analysis of air operations. In L. Jain, N. Ichalkaranje, and G. Tonfoni (eds.), *Advances in Intelligent Systems for Defense*, Series on Innovative Intelligence (Volume 2, 229–264). River Edge, NJ: World Scientific.

Henderson-Sellers, B., and P. Giorgini. 2005. *Agent-Oriented Methodologies*. Hershey, PA: Idea Group.

Høydalsvik, G. M., and G. Sindre. 1993. On the purpose of object-oriented analysis. *ACM SIGPLAN Notices*, *28*(10), 240–255.

Huhns, M. N., and L. M. Stephens. 1999. Multiagent systems and societies of agents. In G. Weiss (ed.), *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence* (79–120). Cambridge, MA, and London, England: MIT Press.

Iglesias, C. A., and M. Garijo. 2005. The agent-oriented methodology MAS-CommonKADS. In B. Henderson-Sellers and P. Giorgini (eds.), *Agent-Oriented Methodologies* (46–78). Hershey, PA: Idea Group.

Ingrand, F., M. Georgeff, and A. Rao. 1992. An architecture for real-time reasoning and system control. *IEEE Expert*, *7*(6), 34–44.

Jacobsson, M., M. Rost, M. Håkansson, and L. E. Holmquist. 2005. Push!Music: Intelligent music sharing on mobile devices. In *Adjunct Proceedings of the Seventh International Conference on Ubiquitous Computing* (UbiComp 2005), September 11–14, Tokyo, Japan (electronic edition).

Jafari, A. 2002. Conceptualizing intelligent agents for teaching and learning. *EDUCAUSE Quarterly*, *25*(3), 28–34.

Jayatilleke, G. B., L. Padgham, and M. Winikoff. 2005. A model driven component-based development framework for agents. *Computer Systems Science and Engineering*, *20*(4), 273–282.

Jennings, N. R. 2000. On agent-based software engineering. *Artificial Intelligence*, *117*(2), 277–296.

Jennings, N. R., P. Faratin, M. J. Johnson, T. J. Norman, P. O'Brien, and M. E. Wiegand. 1996. Agent-based business process management. *International Journal of Cooperative Information Systems*, *5*(2–3), 105–130.

Jennings, N. R., K. Sycara, and M. Wooldridge. 1998. A roadmap of agent research and development. *Autonomous Agents and Multiagent Systems*, *1*(1), 7–38.

Juan, T. 2008. ROADMAP: Agent oriented software engineering for intelligent systems. Ph.D. thesis, The University of Melbourne, Australia.

Juan, T., A. R. Pearce, and L. Sterling. 2002. ROADMAP: Extending the Gaia methodology for complex open systems. In *The First International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS 2002, July 15–19, Bologna, Italy, Proceedings* (3–10). New York: ACM.

Juan, T., and L. Sterling. 2003. The ROADMAP meta-model for intelligent adaptive multiagent systems in open environments. In P. Giorgini, J. Muller, and J. Odell (eds.), *Agent-Oriented Software Engineering IV, 4th International Workshop, AOSE 2003, Melbourne, Australia, July 15, Revised Papers* (LNCS 2935, 53–68). Berlin, Germany: Springer-Verlag.

Juan, T., L. Sterling, M. Martelli, and V. Mascardi. 2003. Customizing AOSE methodologies by reusing AOSE features. In J. S. Rosenschein, T. Sandholm, M. Wooldridge, and M. Yokoo (eds.), *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-03), Melbourne, Australia, July 14–18* (113–120). New York: ACM.

Juan, T., L. Sterling, and M. Winikoff. 2002. Assembling agent oriented software engineering methodologies from features. In F. Giunchiglia, J. Odell, and G. Weiss (eds.), *Agent-Oriented Software Engineering III, Third International Workshop, AOSE 2002, Bologna, Italy, July 15, Revised Papers and Invited Contributions* (LNCS 2585, 198–209). Berlin, Germany: Springer-Verlag.

Kagal, L., T. Finin, and A. Joshi. 2003. A policy based approach to security for the Semantic Web. In D. Fensel, K. P. Sycara, and J. Mylopoulos (eds.), *The Semantic Web—ISWC 2003, Second International Semantic Web Conference, Sanibel Island, FL, October 20–23, Proceedings* (LNCS 2870, 402–418). Berlin, Germany: Springer-Verlag.

Kasinger, H., and B. Bauer. 2005. Towards a model-driven software engineering methodology for organic computing systems. In M. H. Hamza (ed.), *Computational Intelligence: IASTED International Conference on Computational Intelligence*, July 4–6, Calgary, Alberta, Canada (141–146). Calgary, Alberta, Canada: IASTED/ACTA Press.

Kavakli, V., and P. Loucopoulos. 1998. Goal-driven business process analysis–application in electricity deregulation. In B. Pernici and C. Thanos (eds.), *Advanced Information Systems Engineering, 10th International Conference CAiSE '98, Pisa, Italy, June 8–12, Proceedings* (LNCS 1413, 305–324). Berlin, Germany: Springer-Verlag.

Kendall, E. A. 1999. Agent roles and role models: New abstractions for intelligent agent system analysis and design. In A. Holsten, G. Joeris, C. Klauck, M. Klusch, H.-J. Müller, and J. P. Müller (eds.), *International Workshop on Intelligent Agents in Information and Process Management, TZI-Report No. 9*. Bremen, Germany: University of Bremen.

Kinny, D., M. Georgeff, and A. Rao. 1996. A methodology and modelling technique for systems of BDI agents. In W. Van de Velde and J. W. Perram (eds.), *Agents Breaking Away, 7th European Workshop on Modelling Autonomous Agents in a Multiagent World (MAAMAW), Eindhoven, The Netherlands, January 22–25, Proceedings* (LNAI 1038, 56–71). Berlin, Germany: Springer-Verlag.

Kirikova, M. 2000. Explanatory capability of enterprise models. *Data and Knowledge Engineering*, *33*(2), 119–136.

Kuan, P. P., S. Karunasakera, and L. Sterling. 2005. Improving goal and role oriented analysis for agent based systems. In *16th Australian Software Engineering Conference (ASWEC 2005), March 31–April 1, Brisbane, Australia* (40–47). Washington, DC: IEEE Computer Society.

Labarthe, O., E. Tranvouez, A. Ferrarini, B. Espinasse, and B. Montreuil. 2003. A heterogeneous multiagent modelling for distributed simulation of supply chains. In V. Marik, D. C. McFarlane, and P. Valckenaers (eds.), *Holonic and Multiagent Systems for Manufacturing, First International Conference on Industrial Applications of Holonic and Multiagent Systems, HoloMAS 2003, Prague, Czech Republic, September 1–3, Proceedings* (LNCS 2744, 134–145). Berlin, Germany: Springer-Verlag.

Labrou, Y., and T. Finin. 1997. A proposal for a new KQML specification. TR CS-97-03, February, Computer Science and Electrical Engineering Department. Baltimore, MD: University of Maryland.

Laird, J. E., K. J. Coulter, R. M. Jones, P. G. Kenny, F. V. Koss, and P. E. Nielsen. 1998. Integrating intelligent computer generated forces in distributed simulation: TacAir-Soar in STOW-97. In *Proceedings of the 1998 Simulation Interoperability Workshop*. Orlando, FL.

Leontief, A. N. 1979. The problem of activity in psychology. In Wertsch, J. V. (ed.), *The concept of activity in soviet psychology* (37–71). Armonk, NY: M. E. Sharpe.

Lister, K., and L. Sterling. 2003. Tasks as context for intelligent agents. In *Proceedings of the 2003 IEEE/WIC International Conference on Intelligent Agent Technology* (IAT 2003), October 13–17, Halifax, Canada (154–160). Washington, DC: IEEE Computer Society.

Luck, M., and L. Padgham, eds. 2008. *Agent-Oriented Software Engineering VIII: The 8th International Workshop on Agent Oriented Software Engineering, AOSE 2007, Honolulu, HI, May 14, Revised Selected Papers* (LNCS 4951). Berlin, Germany: Springer-Verlag.

Luo, Y., G. Antoniou, and L. Sterling. 2007. Incorporating security requirements into communication protocols in multiagent software systems. In *Proceedings of the Eighth International Conference on Parallel and Distributed Computing, Applications and Technologies* (PDCAT '07), December 3–6, Adelaide, Australia (159–160). Washington, DC: IEEE Computer Society.

Luo, Y., and L. Sterling. 2007. *Modelling a Smart Music Player with a Hybrid Agent-Oriented Methodology*. Agentlab Technical Report, Department of Computer Science and Software Engineering. Melbourne, Australia: The University of Melbourne.

Luo, Y., L. Sterling, and K. Taveter. 2007. Modeling a smart music player with a hybrid agent-oriented methodology. In *Proceedings of the 15th IEEE International Requirements Engineering Conference*, October 15–19, Delhi, India (281–286). Washington, DC: IEEE Computer Society.

Markham, S., J. Ceddia, J. Sheard, C. Burvill, J. Weir, B. Field, L. Sterling, and L. Stern. 2003. Applying agent technology to evaluation tasks in e-learning environments. In *Proceedings of the Exploring Educational Technologies Conference*. Melbourne, Australia: Monash University.

Martin, D., and I. Sommerville. 2004. Patterns of cooperative interaction: Linking ethnomethodology and design. *ACM Transactions on Computer-Human Interaction*, *11*(1), 59–89.

Marchewka, J. T. 2006. *Information Technology Project Management: Providing Measurable Organizational Value*. 2nd edition. Chichester, UK: John Wiley and Sons.

Masolo, C., S. Borgo, A. Gangemi, N. Guarino, and A. Oltramari. 2003. Ontology library. *EU IST Project 2001-33052 WonderWeb: Ontology Infrastructure for the Semantic Web, Deliverable D18*.

Medina-Mora, R., T. Winograd, R. Flores, and F. Flores. 1992. The action workflow approach to workflow management technology. In *Proceedings of the Conference on Computer Supported Cooperative Work (CSCW), October 31–November 4, Toronto, Canada* (281–288). New York: ACM.

Metsker, S. J. 1997. Thinking over objects. *Object Magazine*, *7*(3), 56–59.

Morandini, M., D. C. Nguyen, A. Perini, A. Siena, and A. Susi, 2008. Tool-supported development with Tropos: The conference management system case study. In M. Luck and L. Padgham (eds.), *Agent-Oriented Software Engineering VIII: The 8th International Workshop on Agent-Oriented Software Engineering, AOSE 2007, Honolulu, HI, May 14, Revised Selected Papers* (LNCS 4951, 182–196). Berlin, Germany: Springer-Verlag.

Nguyen, D. C., Perini, A., and Tonella, P. 2008. A goal-oriented software testing methodology. In M. Luck and L. Padgham (eds.), *Agent-Oriented Software Engineering VIII: The 8th International Workshop on Agent-Oriented Software Engineering, AOSE 2007, Honolulu, HI, May 14, Revised Selected Papers* (LNCS 4951, 58–72). Berlin, Germany: Springer-Verlag.

Nwana, H. S. 1995. Software agents: An overview. *Knowledge Engineering Review*, *11*(2), 205–244.

Odell, J. 2002. Objects and agents compared. *Journal of Object Technology*, *1*(1), 41–53.

Odell, J., H. Van Dyke Parunak, and B. Bauer. 2001. Representing agent interaction protocols in UML. In P. Ciancarini and M. Wooldridge (eds.), *Agent-Oriented Software Engineering, First International Workshop, AOSE 2000, Limerick, Ireland, June 10, Revised Papers* (LNCS 1957, 121–140). Berlin, Germany: Springer-Verlag.

Oja, M., B. Tamm, and K. Taveter. 2001. Agent-based software design. *Proceedings of the Estonian Academy of Sciences: Engineering*, *7*(1), 5–21.

OMG. 2003. MDA Guide Version 1.0.1. June 12, 2003. Retrieved June 28, 2007, from http://www.omg.org/cgi-bin/doc?omg/03-06-01.

OMG. 2006. Business Process Modeling Notation (BPMN) Specification. Final Adopted Specification, February 6, 2006. Retrieved February 10, 2008, from http://www.omg.org/bpmn/.

OMG. 2007. Unified Modeling Language: Superstructure. Version 2.1.1, February 2007. Retrieved June 28, 2007, from http://www.omg.org/cgi-bin/doc?formal/07-02-05.

Ow, S. P., S. F. Smith, and R. A. Howie. 1998. CSS: A cooperative scheduling system. In M. D. Oliff (ed.), *Proceedings of the 2nd International Conference on Expert Systems and the Leading Edge in Production Planning and Control*, May 3–5, Charleston, SC (43–56). New York: North-Holland.

Padgham, L., and M. Winikoff. 2003. Prometheus: A methodology for developing intelligent agents. In F. Giunchiglia, J. Odell, and G. Weiss (eds.), *Agent-Oriented Software Engineering III, Third International Workshop, AOSE 2002, Bologna, Italy, July 15, Revised Papers and Invited Contributions* (LNCS 2585, 174–185). Berlin, Germany: Springer-Verlag.

Padgham, L., and M. Winikoff. 2004. *Developing intelligent agent systems: A practical guide*. Chichester, UK: John Wiley and Sons.

Padgham, L., J. Thangarajah, and M. Winikoff. 2005. Tool support for agent development using the Prometheus methodology. In *Proceedings of the 2005 NASA/DoD Conference on Evolvable Hardware* (EH 2005), June 29–July 1, Washington, DC (383–388). Washington, DC: IEEE Computer Society.

Padgham, L., J. Thangarajah, and M. Winikoff. 2008. The Prometheus Design Tool—A conference management system case study. In M. Luck and L. Padgham (eds.), *Agent-Oriented Software Engineering VIII: The 8th International Workshop on Agent-Oriented Software Engineering, AOSE 2007, Honolulu, HI, May 14, 2007, Revised Selected Papers* (LNCS 4951, 197–211). Berlin, Germany: Springer-Verlag.

Parunak, H. Van Dyke. 2000. *International Journal of Cooperative Information Systems*, *9*(3), 209–227.

Parunak, H. Van Dyke, A. D. Baker, and S. J. Clark. 1997. The AARIA agent architecture: An example of requirements-driven agent-based system design. In *Proceedings of the First International Conference on Autonomous Agents*, February 5–8, Marina del Rey, CA (482–483). New York: ACM.

Pavón, J., J. J. Gómez-Sanz, and R. Fuentes. 2005. The INGENIAS methodology and tools. In B. Henderson-Sellers and P. Giorgini (eds.), *Agent-Oriented Methodologies* (236–276). Hershey, PA: Idea Group.

Peirce, C. S. 1935. *Scientific Metaphysics, Vol. VI of the Collected Papers*. Cambridge, MA: Harvard University Press.

Penserini, L., A. Perini, A. Susi, and J. Mylopoulos. 2006. From stakeholder intentions to software agent implementations. In E. Dubois and K. Pohl (eds.), *Advanced Information Systems Engineering, 18th International Conference, CAiSE 2006, Luxembourg, Luxembourg, June 5–9, Proceedings* (LNCS 4001, 465–479). Berlin, Germany: Springer-Verlag.

Perugini, D. 2007. Agents for logistics: A provisional agreement approach. Ph.D. thesis, The University of Melbourne, Australia.

Picard, G., and M.-P. Gleizes. 2004. The ADELFE methodology. In F. Bergenti, M.-P. Gleizes, and F. Zambonelli (eds.), *Methodologies and Software Engineering for Agent Systems: The Agent-Oriented Software Engineering Handbook* (157–175). Norwell, MA: Kluwer Publishing.

Putman, J., R. 2001. *Architecting with RM-ODP*. Upper Saddle River, NJ: Prentice Hall.

Raffel, W.-U. 2005. Agentenbasierte Simulation als Verfeinerung der Diskreten-Ereignis-Simulation unter besonderer Berücksichtigung des Beispiels Fahrerloser Transportsysteme. Ph.D. thesis, Institut für Informatik, FU Berlin, Germany.

Rahwan, I., T. Juan, and L. Sterling. 2006. Integrating social modelling and agent interaction through goal-oriented analysis. *International Journal of Computer Systems Science and Engineering*, *21*(2), 87–98.

Rao, A. S., and M. P. Georgeff. 1991. Modeling rational agents within a BDI architecture. In J. Allen, R. Fikes, and E. Sandewall (eds.), *Proceedings of Knowledge Representation 91* (KR-91) (473–484). San Francisco, CA: Morgan Kaufmann.

Rothwell, W. J., and H. C. Kazanas. 1997. *Mastering the Instructional Design Process: A Systematic Approach*. San Francisco, CA: Jossey-Bass.

Rumbaugh, J., M. Blaha, W. Premerlani, F. Eddy, and W. Lorensen. 1991. *Object Oriented Modeling and Design*. Englewood Cliffs, NJ: Prentice-Hall.

Russell, S. J., and P. Norvig. 2002. *Artificial Intelligence: A Modern Approach*. 2nd edition. Upper Saddle River, NJ: Prentice Hall.

Shoham, Y. 1993. Agent-oriented programming. *Artificial Intelligence*, *60*(1), 51–92.

Sierhuis, M., W. J. Clancey, and R. van Hoof. 2003. Brahms: A multiagent modeling environment for simulating social phenomena. In *Proceedings of the First Conference of the European Social Simulation Association* (SIMSOC VI), September 18–21, Groningen, The Netherlands.

Silva, V., and C. Lucena. 2004. From a conceptual framework for agents and objects to a multiagent system modeling language. *Journal of Autonomous Agents and Multiagent Systems*, *9*(1–2), 145–189. Norwell, MA: Kluwer Publishing.

Singh, M. P. 1999. An ontology for commitments in multiagent systems: Toward a unification of normative concepts. *Artificial Intelligence and Law*, *7*(1), 97–113.

Smith, R. D. 1998. Essential techniques for military modeling and simulation. In D. J. Medeiros, E. F. Watson, J. S. Carson, and M. S. Manivannan (eds.), *Proceedings of the 30th Conference on Winter Simulation*, December 13–16, Washington, DC (805–812). New York: ACM.

Smith, S. 1995. Reactive scheduling systems. In D. E. Brown and W. T Scherer (eds.), *Intelligent Scheduling Systems* (155–192). Norwell, MA: Kluwer Publishing.

Smith, S. F., and M. A. Becker. 1997. An ontology for constructing scheduling systems. In *Working Notes of the 1997 AAAI Spring Symposium on Ontological Engineering* (120–129), Stanford, CA.

Smith, S., P. S. Ow, N. Muscettola, J. Y. Potvin, and D. Matthys. 1990. OPIS: An opportunistic factory scheduling system. In *Proceedings of the 3rd International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems* (IEA/AIE '90), July 15–18, Charleston, SC (Vol. 1, 268–274).

Sommerville, I. 2007. *Software Engineering*. 8th edition. Reading, MA: Addison-Wesley.

Sowa, J. F., and J. A. Zachman. 1992. Extending and formalizing the framework for information systems architecture. *IBM Systems Journal*, *31*(3), 590–616.

Sterling, L., and T. Juan. 2005. The software engineering of agent-based intelligent adaptive systems. In G.-C. Roman, W. G. Griswold, and B. Nuseibeh (eds.), *27th International Conference on Software Engineering (ICSE 2005), May 15–21, St. Louis, Missouri, Proceedings* (704–705). New York: ACM.

Sterling, L., and E. Shapiro. 1994. *The Art of Prolog: Advanced Programming Techniques*. 2nd edition. Cambridge, MA, and London, England: MIT Press.

Sterling, L., K. Taveter, and the Daedalus Team. 2006a. Building agent-based appliances with complementary methodologies. In E. Tyugu and T. Yamaguchi (eds.), *Knowledge-Based Software Engineering, Proceedings of the Seventh Joint Conference on Knowledge-Based Software Engineering* (JCKBSE '06), August 28–31, Tallinn, Estonia (223–232). Amsterdam, The Netherlands: IOS Press.

Sterling, L., K. Taveter, and the Daedalus Team. 2006b. Experience from building industry strength agent-based appliances. *An Industry Experience Report at the Australian Software Engineering Conference* (ASWEC 2006), April 18–21, Sydney, Australia. Retrieved June 7, 2008, from http://www.cs.mu.oz.au/~kuldar/Daedalus-paper.pdf.

Stern, L., and K. Lam. 2007. A framework for evaluating multimedia software: Modeling student learning strategies. In *ED-MEDIA 2007: World Conference on Educational Multimedia, Hypermedia and Telecommunications*, Vancouver, BC, Canada, June 25–29 (3301–3309). Chesapeake, VA: Association for the Advancement of Computing in Education.

Stern, L., and L. Naish. 2002. Visual representation for recursive algorithms. In *SIGCSE '02: Proceedings of the 33rd SIGCSE Technical Symposium on Computer Science Education*, Cincinnati, OH (196–200). New York: ACM.

Stern, L., L. Naish, and H. Sondergaard. 1999. A strategy for managing content complexity in algorithm animation. In B. Manaris (ed.), *Proceedings of the Fourth Annual SIGCSE/SIGCUE Conference on Innovation and Technology in Computer Science Education* (ITiCSE99), Cracow, Poland (127–130). New York: ACM.

Stern, L., and L. Sterling. 2006. Towards agents for educational software. In *ED-MEDIA 2006*: *World Conference on Educational Multimedia, Hypermedia and Telecommunications*, Orlando, FL, June 26–30 (2040–2047). Chesapeake, VA: Association for the Advancement of Computing in Education.

Tamm, B. G., R. Puusepp, and R. Tavast. 1987. *Analiz i modelirovanije proizvodstvennyh sistem* (*Analysis and modeling of production systems*). Moscow: Finansy i Statistika.

Taveter, K. 1997. From object-oriented programming towards agent-oriented programming. In G. Grahne (ed.), *Proceeedings of the Sixth Scandinavian Conference on Artificial Intelligence* (SCAI '97), August 18–20, Helsinki, Finland (288). Amsterdam, The Netherlands: IOS Press.

Taveter, K. 2004a. A multi-perspective methodology for agent-oriented business modeling and simulation. Ph.D. thesis, Tallinn University of Technology, Estonia.

Taveter, K. 2004b. From business process modeling to business process automation. In J. Cordeiro and J. Filipe (eds.), *Computer Supported Activity. Coordination, Proceedings of the 1st International Workshop on Computer Supported Activity Coordination* (CSAC 2004). In conjunction with ICEIS 2004, April, Porto, Portugal (198–210). Porto, Portugal: INSTICC Press.

Taveter, K. 2005a. Business process automation with representing and reasoning on trust. In *Proceedings of the 3rd International IEEE Conference on Industrial Informatics* (INDIN '05), August 10–12, Perth, Western Australia (electronic edition). Washington, DC: IEEE Computer Society.

Taveter, K. 2005b. CONE: Improving human-computer interaction by integrated ontological models. In *Proceedings of the 11th International Conference on Human-Computer Interaction*, July 22–27, Las Vegas, Nevada (electronic edition). Las Vegas, NV: Mira Digital Publishing.

Taveter, K. 2006a. A technique and markup language for business process automation. In *Proceedings of the Workshop on Vocabularies, Ontologies, and Rules for the Enterprise* (VORTE 2006). In conjunction with the Tenth IEEE International EDOC (The Enterprise Computing) Conference, October 16–20, Hong Kong (electronic edition). Washington, DC: IEEE Computer Society.

Taveter, K. 2006b. Application of RAP/AOR to the modeling and simulation of a ceramic factory. In J.-P. Rennard (ed.), *Handbook of Research on Nature-Inspired Computing for Economy and Management* (541–556). Hershey, PA: Idea Group.

Taveter, K., and R. Hääl. 2002. Agent-oriented modelling and simulation of a ceramic factory. In E. Juuso and L. Yliniemi (eds.), *Proceedings of the 43rd Conference of Simulation and Modelling* (SIMS 2002), September 26–27, Oulu, Finland (102–110). Helsinki, Finland: Finnish Society of Automation and Scandinavian Simulation Society.

Taveter, K., and L. Sterling. 2008. An expressway from agent-oriented models to prototype systems. In M. Luck and L. Padgham (eds.), *Agent-Oriented Software Engineering VIII: The 8th International Workshop on Agent-Oriented Software Engineering, AOSE 2007, Honolulu, HI, May 14, Revised Selected Papers* (LNCS 4951, 147–163). Berlin, Germany: Springer-Verlag.

Taveter, K., and G. Wagner. 2006. Agent-oriented modeling and simulation of distributed manufacturing. In J.-P. Rennard (ed.), *Handbook of Research on Nature Inspired Computing for Economy and Management* (527–540). Hershey, PA: Idea Group.

Taveter, K., and G. Wagner. 2001. Agent-oriented enterprise modeling based on business rules. In H. S. Kunii, S. Jajodia, and A. Sølvberg (eds.), *Conceptual Modeling—ER 2001, 20th International Conference on Conceptual Modeling, Yokohama, Japan, November 27–30, Proceedings* (LNCS 2224, 527–540). Berlin, Germany: Springer-Verlag.

Taveter, K., and G. Wagner. 2005. Towards radical agent-oriented software engineering processes based on AOR modelling. In B. Henderson-Sellers and P. Giorgini (eds.), *Agent-Oriented Methodologies* (277–316). Hershey, PA: Idea Group.

Tamagotchi. 2008. *Tamagotchi Connection*. Retrieved February 5, 2008, from http://www.tamagotchi.com.

Vetere, F., M. R. Gibbs, J. Kjeldskov, S. Howard, F. Mueller, S. Pedell, K. I. Mecoles, and M. Bunyan. 2005. Mediating intimacy: Designing technologies to support strong-tie relationships. In G. C. van der Veer and C. Gale (eds.), *Proceedings of the 2005 Conference on Human Factors in Computing Systems* (CHI 2005), April 2–7, Portland, OR (471–480). New York: ACM.

Vrba, P. 2003. MAST: manufacturing agent simulation tool. In *Emerging Technologies and Factory Automation, 2003, Proceedings, ETFA '03, IEEE Conference*, September 16–19 (Volume 1, 282–287). Washington, DC: IEEE Computer Society.

Wagner, G. 2003. The agent-object-relationship meta-model: Towards a unified view of state and behavior. *Information Systems*, *28*(5), 475–504.

Weyns, D., Omicini, A., and Odell, J. 2007. Environment as a first class abstraction in multiagent systems. *Autonomous Agents and Multiagent Systems*, *14*(1), 5–30.

Wikipedia. 2006a. *Electronic Commerce*. Retrieved October 11, 2006, from http://en.wikipedia.org/wiki/E-commerce.

Wikipedia. 2006b. *Home Automation*. Retrieved October 11, 2006, from http://en.wikipedia.org/wiki/Intelligent_home.

Wikipedia. 2006c. *Computer Virus*. Retrieved December 17, 2006, from http://en.wikipedia.org/wiki/Software_virus.

Wikipedia. 2008a. *Tamagotchi*. Retrieved February 5, 2008, from http://en.wikipedia.org/wiki/Tamagotchi.

Wikipedia. 2008b. *Procedural Reasoning System*. Retrieved February 25, 2008, from http://en.wikipedia.org/wiki/Procedural_Reasoning_System.

Winikoff, M. 2005. JACK™ intelligent agents: An industrial strength platform. In R. H. Bordini, M. Dastani, J. Dix, and A. E. F. Seghrouchni (eds.), *Multiagent Programming: Languages, Platforms, and Applications* (175–193). Berlin, Germany: Springer-Verlag.

Winograd, T., and F. Flores. 1986. *Understanding Computers and Cognition: A New Foundation for Design*. Reading, MA: Addison-Wesley.

Wood, M. F., and S. A. DeLoach. 2001. An overview of the multiagent systems engineering methodology. In P. Ciancarini and W. Wooldridge (eds.), *Agent-Oriented Software Engineering, First International Workshop, AOSE 2000, Limerick, Ireland, June 10, 2000, Revised Papers* (LNCS 1957, 207–221). Berlin, Germany: Springer-Verlag.

Wooldridge, M. 1999. Intelligent agents. In G. Weiss (ed.), *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence* (27–78). Cambridge, MA, and London, England: MIT Press.

Wooldridge, M. 2002. *An Introduction to Multiagent Systems*. Chichester, UK: John Wiley and Sons.

Wooldridge, M., and N. R. Jennings. 1995. Intelligent agents: Theory and practice. *Knowledge Engineering Review*, *10*(2), 115–152.

Wooldridge, M., N. R. Jennings, and D. Kinny. 2000. The Gaia methodology for agent-oriented analysis and design. *Autonomous Agents and Multiagent Systems*, *3*(3), 285–312.

W3C. 2003. Architecture of the World Wide Web, First Edition. W3C Working Draft, 9 December 2003. Retrieved July 30, 2007, from http://www.w3.org/TR/2003/WD-webarch-20031209/.

W3C. 2007. Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language. W3C Recommendation, June 26, 2007. Retrieved July 31, 2007, from http://www.w3.org/TR/wsdl20/.

Yu, E. 1995. Modeling strategic relationships for process reengineering. Ph.D. thesis, Department of Computer Science, University of Toronto, Canada.

Zachman, J. A. 1987. A framework for information systems architecture. *IBM Systems Journal*, *26*(3), 276–292.

Zambonelli, F., N. R. Jennings, and M. Wooldridge. 2001. Organizational abstractions for the analysis and design of multiagent systems. In P. Ciancarini and M. Wooldridge (eds.), *Agent-Oriented Software Engineering, First International Workshop, AOSE 2000, Limerick, Ireland, June 10, 2000, Revised Papers* (LNCS 1957, 235–251). Berlin, Germany: Springer-Verlag.

Zambonelli, F., N. R. Jennings, and M. Wooldridge. 2003. Developing multiagent systems: The Gaia methodology. *ACM Transactions on Software Engineering and Methodology*, *12*(3), 317–370.