Int. J. Engng Ed. Vol. 22, No. 5, pp. 1097–1104, 2006
Printed in Great Britain.

0949-149X/91 $3.00+0.00

Expanding Our Horizons in Teaching the Use of Intelligent Agents for Simulation Modeling of Next Generation Engineering Systems*

LEVENT YILMAZ
M&SNet: Auburn Modeling and Simulation Laboratory, Department of Computer Science and Software Engineering, Auburn University, Auburn, AL 36849, USA. E-mail: yilmaz@auburn.edu

Multi-agent systems are advocated as a model for designing complex, distributed engineering systems. Yet the practice of teaching the use of intelligent agents in modeling and simulation of next generation open, dynamic, adaptive, and intelligent engineering applications is still in its infancy. In this paper we present a unified and coherent framework for teaching a graduate level agent-directed simulation course for computer science and engineering students. The framework aims to: (1) promote extending our horizons by introducing multiple dimensions for the use of agents in simulation; (2) emphasize focusing on teaching the theory, methodology, and fundamental principles underlying the agent-based modeling framework, and (3) suggest a shift from a predictive modeling worldview toward a new computational epistemology perspective that advocates exploratory experimentation with agent-based models. Based on these premises, a synopsis of the structure, delivery strategy, and the underlying rationale for the design of the course are presented.

Keywords: agent-based simulation; modeling and simulation; multi-agent systems; agents; education

INTRODUCTION

The premise of the agent paradigm and its related theory and methodologies are opening up new frontiers for advancing the physical, natural, social, military, and information sciences and engineering [1]. Agents are being espoused in product and process design, environmental management [3], electrical power grid and transport network management [4], steel production management [5], transportation systems and road traffic control [6].

At present, there is much debate about exactly what constitutes agenthood. Bradshaw [7] discussed approaches to defining agents. Intelligent agents are often defined as encapsulated entities that are situated in some environment and that are capable of flexible, autonomous action in order to meet their design objectives. An increasing number of engineering systems are being viewed in terms of autonomous entities. The multi-agent systems approach is being advocated as a next generation model for engineering complex, distributed systems [8]. Yet, despite the growing interest, the transition of agent research into engineering education requires further impetus.

Social scientists have already embraced the agent paradigm to study and teach the dynamics of complex adaptive systems, social complexity [9], economics [10], organizations, institutes, and societies [11]. However, existing agent-based simulation courses in social sciences revolve around application domain problems, as opposed to agent theory, methodology, and associated technologies that constitute the foundation of multi-agent systems. This is a satisfying approach within the social science curriculum since the primary objective is to explore social complexity via computer simulation. However, as computer science and engineering educators, we need to take into account the learning objectives involving the design principles, conceptual frameworks, and common mechanisms underlying such systems. This is mainly due to our interest in further developing and advancing the theory and methodology of multi-agent systems as well as exploring their complex behavior in the context of specific engineering applications. This extended view brings new challenges, as well as avenues, for teaching the use of agents in the simulation and analysis of next generation engineering applications. Fortunately, there already exist a number of engineering courses involving the use of agents [12]. MaterialSim toolkit [13] and the strategies for teaching agents to industrial engineers [14] demonstrate the increasing level of acceptance of agents in simulation modeling of engineering phenomena. In software engineering, agent-based

* Accepted 9 March 2006.

software engineering courses are also emerging [15]. Topics of interest include agent architectures, communication, knowledge sharing, computing, and uncertainty management.

There are general-purpose agent-based simulation courses that emphasize the use of agents as design metaphors [16]. Similar courses in the military domain also exist [17]. Such courses, however, focus on military applications of Artificial Intelligence techniques. Integrated simulation environments are proposed to allow students to develop and test AI algorithms in dynamic, uncertain, visual environments [18, 19]. There are also short-tem courses that focus on using agents to facilitate modeling complex adaptive social phenomena [20, 21] and business complexity [22].

Based on the review of existing courses in agent-based simulation modeling, we observe that agents are exclusively used as design metaphors to model systems. This is a rather limited viewpoint regarding the potential of intelligent agents in engineering. The framework proposed in this paper takes a small step towards expanding our horizons in teaching the use of agents in Modeling and Simulation.

- First, we present a dichotomy that depicts an expanded viewpoint for teaching different ways in which agents can be used in design and simulation modeling of modern engineering systems. This viewpoint, which is called Agent-Directed Simulation (ADS), extends the common narrow view of using agents simply as design metaphors.
- Second, critical theoretical aspects of multi-agent systems are used to guide the formulation of the course. This is in sharp contrast with existing strategies, where agent-based simulation courses are structured around the application domain without examination of the design principles and conceptual frameworks underlying agent systems.
- Finally, we promote viewing simulations as exploratory computational experiments, as opposed to predictive tools that mirror the system of interest.

Predictive modeling comes from the context of theoretical science, with a bias toward deductive reasoning and a resulting preference for validity as a standard quality. Agent-based simulation modeling treats the use of computer models as experimental science. The purpose of agent-based models is not necessarily to predict the outcome of a system, rather it is to reveal and understand the complex and aggregate system behaviors that emerge from the interactions of the various individuals involved. This viewpoint is based on the observation that emergent engineering applications are becoming dynamic, adaptive and open systems [23], for which the tools of traditional closed systems are limited. More specifically, it is suggested in [23] that if our critical infrastructures are to continue to provide vital services safely and reliably, the linkages between people, organizations, and technology needs to be fully understood and managed holistically. As we start exploring the state space of such systems, the types of courses, as espoused in this paper, will gradually increase in numbers and find their place within the engineering curriculum.

AGENT-DIRECTED SIMULATION

In developing a course for teaching the use of agents in modeling and simulation, we take into account: (1) the different roles that agents can play in M&S; (2) the underlying theory, methodology, and principled model design strategies for multi-agent systems, and (3) the need for revisiting the engineering view on computational epistemology under the agent-based conception of engineering systems.

Dimensions involving the use of agents

Agents are often viewed as design metaphors in the development of models and scenarios for simulation. Yet, this narrow view limits the potential of agents in improving various other dimensions of simulation. To this end, the three-tier dichotomy shown in Fig. 1 presents the framework of Agent-Directed Simulation that consists of three distinct, yet related, areas that can be grouped under two categories as follows:

Simulation for Agents (agent simulation), i.e., simulation of systems that can be modeled by agents in engineering, human and social dynamics, military applications etc.;

Agents for Simulation that can be grouped under two groups: *agent-based simulation*, which focuses on the use of agents for the generation of model behavior in a simulation study; and *agent-supported simulation*, which deals with the use of agents as a support facility to enable computer assistance by enhancing cognitive capabilities in problem specification and solving.

Agent simulation involves the use of conventional simulation frameworks (i.e., discrete-event scheduling, activity scanning, process interaction) to simulate the behavioral dynamics of agent systems [24].

In multi-agent systems, agents learn from and about other agents, find proper ways to cooperate, negotiate, establish, and manage coalitions for effective problem solving. Simulating agent systems requires understanding such basic mechanisms and technologies underlying such systems. Agent-based simulation involves the use of agents as design metaphors in developing simulation models. That is, models and their simulations are designed around autonomous, communicative entities that are flexible and effective in open dynamic environments. This brings a radically new solution to the very concept of simulation modeling of engineering applications,

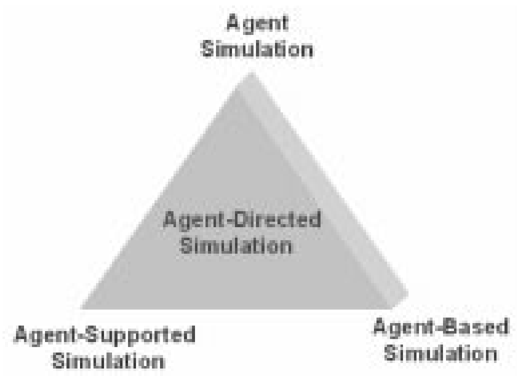

Fig. 1. Agent-directed simulation.

by offering the possibility of directly representing individuals, their behavior, and interaction. The use of agents as supplementary components to improve adaptivity and learning capabilities of the simulated system are the common themes promoted under agent-supported simulation.

The theory and methodology of multi-agent systems

Teaching modeling, design, and simulation of multi-agent systems involves the analysis of a variety of problems. Take, for instance, an automated railcar system [25]. How do the agents perceive one other? How do they cooperate? Can they sustain their viability? Are they capable of adapting their behavior to the changes in their environment? One can classify such questions into four main categories: (1) the behavior and action, (2) cognitive decision-making, (3) interaction, and (4) adaptation.

The action and decision-making functions involve the conceptualization and specification of reactive and deliberative behavior of agents. The interaction deals with describing the elementary mechanisms that allow agents to communicate and cooperate. Cooperation is the general form of interaction most studied in multi-agent systems. Depending on the available resources and skills of agents involved in modeling a multi-agent system, cooperation deals with task allocation, coordination of actions, and resolution of conflicts. The methods underlying these components of cooperation pervade in most multi-agent systems. Therefore, teaching how to model collaboration via task allocation, coordination, and negotiation for conflict resolution are critical. Realistic representation of such systems requires understanding requisite adaptivity and evolution strategies used in various engineering application areas such as collective robotics and distributed decision-making.

COMP8700: Agent-directed simulation

The above dimensions and issues are taken into consideration in formulating a graduate level special topics course on Agent-Directed Simulation taught at the Computer Science and Software Engineering Department of Auburn University. Fifteen students registered and successfully completed this advanced-level course. The course requires basic understanding about the fundamental principles of discrete-event simulation, object-oriented programming paradigm, set theory, and algorithm design. Set theory is used to model abstract and concrete agent architectures, whereas knowledge about algorithms is needed to understand mechanisms for cooperation. Knowledge of Unified Modeling Language (UML) is not necessary but can be useful in communicating the design of a model. UML is a graphical language for visualizing, specifying, constructing, and documenting the artifacts of a software intensive system [26]. Discrete-event simulation and object-oriented programming skills are needed to develop simulation programs that implement the proposed models. The course has the following objectives:

- acquiring modeling and simulation skills to develop simulation-based solutions to analyze complex systems that can be modeled by agent methodology;
- acquiring the ability to use software agents to not only represent intelligent entities within simulation models but also to enhance modeling and simulation environments, and
- acquiring the ability to use agent theoretic model design principles and conceptual frameworks.

It is expected that having successfully completed this course, students will be able to

- solve complex problems by way of using computational agent theory and methodology under the discrete-event simulation framework;
- demonstrate advanced knowledge in the modeling and simulation of agent technologies;
- simulate agent systems using a discrete-event model development environment (i.e. RePast), and participate in agent-based simulation model design and development projects.

The course involves a comprehensive group project, in which students apply the theoretical aspects of agent-directed simulation within the realm of the selected projects. The projects span a broad range of topics such as software processes, transportation networks, facility design for physical security, and anticipatory network fault management [27].

A FRAMEWORK FOR TEACHING THE USE OF AGENTS IN M&S

In COMP8700, the course delivery and projects proceed simultaneously, so that we can bridge the gap between the theory and practice. The challenge in designing COMP8700 was to have a coherent and unified framework by which the theory and practice of various aspects of multi-agent systems can be bridged in a meaningful and seamless manner.

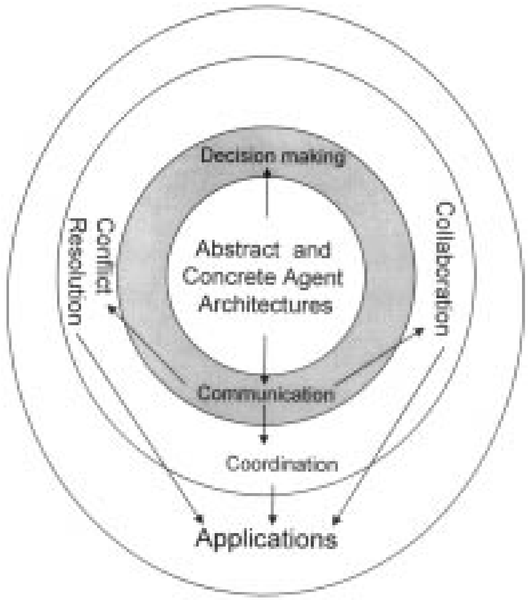

Fig. 2. A framework for teaching ADS.

The framework shown in Fig. 2 depicts the learning modules of the course. The modules involve the conceptualization of agents as well as the interaction design. To facilitate teaching design of Multi-agent Systems we develop seven major learning modules, which are depicted in Fig. 2. The modules are organized in a framework that aims to provide a holistic approach to understanding agent system. The framework is influenced (1) by the experience of the instructor in developing agent simulations and (2) by Wooldridge's philosophical arguments [8] on making the case for an agent-based approach to modeling symbolic systems. Students first learn the means to conceptualize and design agents with reactive and deliberative mechanisms. The specification of agent behavior, and its deliberation mechanisms are taught as part of the internal architecture of agents.

The rest of the learning modules pertain to interaction design that involves various forms of communication and cooperation strategies among a society of agents. Learning about various forms of communication such as point-to-point, multi-cast, propagation by signal diffusion, and notice-board, enable students to choose the proper communication mechanism among a society of agents. With these two components, students gain insight about how to structure their simulations around autonomous, decentralized, and communicative actors. The learning modules that constitute the cooperation aspect can be used in conventional simulations to model agent systems. Such mechanisms can also be embedded within the agent organization that constitutes the infrastructure of agent-based simulations.

Conceptualization and design of agents

Introducing intelligent agents to students requires a definition of intelligent behavior. A conceptual template is presented as an abstract architecture. The abstract template constitutes the perception, deliberation, and action components.

Equipped with formal set theoretic representation of each component, students develop an understanding of the common characteristics of agents. By varying the abstract specification with distinct forms of realization, they observe how alternative concrete architectures can be developed. In particular, we focus on three types of architectures: logic-based architecture, reactive subsumption-based architecture, and practical deliberation architecture that uses the Belief–Desires–Intentions framework. The advantages and disadvantages of the architectures along with the application domains for which they have proved to be useful are discussed to help students make informed selection for their group projects.

Teaching agent interaction design

Modeling and designing a multi-agent system requires detailed analysis of the forms and types of interaction between agents. Interaction is viewed as the most critical and fundamental aspect of a multi-agent system, as intelligence is viewed as an emergent property that is a direct manifestation of local interactions among agents. The challenge in interaction design for multi-agent systems is that complex agent systems involve a changing web of relations for flexibly forming, maintaining, and disbanding organizations.

Communication in multi-agent systems is indispensable to expand the perceptive capacities of agents by allowing them to benefit from the information and know-how that other agents possess. The cooperation module comprises collaboration, coordination, and conflict resolution components. Each one of these components is discussed in isolation with significant detail before we demonstrate their collective usage in a case study.

- **Communication:** Various forms of communication are explored. While cognitive agents use symbolic messages that are routed directly between the sender and recipient agents, reactive agents use signal propagation and perception. Furthermore, students are briefly introduced with the notion of speech acts that designate intentional actions (i.e., request, affirmation) along with conversation protocols that realize such high-level communication primitives.
- **Cooperation:** Most multi-agent systems promote cooperative interaction among agents. The form of cooperation discussed in COMP8700 is a specific type of interaction, called coordinated collaboration. The situations in which the resources and skills of agents are insufficient to solve a problem require task allocation and coordination of actions to synchronize agents in their collective actions. While coordination minimizes the possibility of conflicts, agents may have incompatible goals that may result in conflicting situations. Arbitration and negotiation methods are presented to demonstrate how such situations can be avoided, managed, and resolved. In particular, various forms of auctions (i.e., English, Dutch auctions) and negotiation techniques are examined.

To illustrate the methods of cooperation, a case study (Predator–Prey model) is used to demonstrate how agents at increasing levels of sophistication can improve the problem-solving performance. The case study is influenced by the web-based agent simulation reported in [28]. Ferber's [29] formulation of the model provided further impetus to develop the case study. Originally, the study is based on a popular game, which is called Hunt [30]. The game describes interactions between two species in an ecosystem: a predator and a prey. The objective of the case study is to provide a powerful test bed to teach various forms of collaboration and coordination that can be extended to other application domains.

Bridging the gap between theory and practice: predator–prey model

In the case study, the coordinated collaboration problem is examined as a game, where agents, the prey animals, like the predators, move over a space represented in the form of a grid as shown in Fig. 3. The objective of the game is for the predators to capture the prey animals by following and surrounding them. The predators need to cooperate to improve their performance in capturing prey animals. There are various application areas for this simple game. For instance, in designing physical security systems one needs to define sophisticated security protocols to improve the likelihood of detecting and isolating intruders by security guards, while at the same time optimizing the cost of operating the facility. To study this

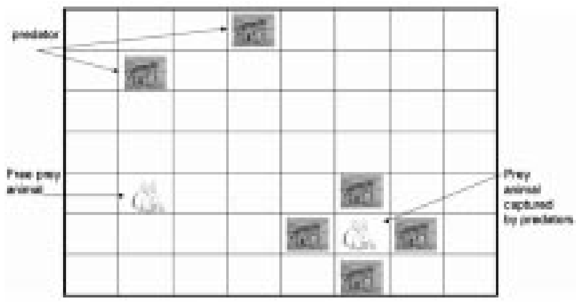

Fig. 3. The predator–prey game.

problem a number of hypotheses are laid out to set the basic rules of the game to facilitate achieving the learning objectives.

The dimensions of the environment (that is, of the grid) are finite,
The predators and prey animals move at fixed speeds.
The prey animals move in a random manner.
The predators can use the corners and edges to block a prey animal's path.
The predators have a limited perception of the world that surrounds them.

Specifically, the problem consists of coordinating the actions of the predators so that they can surround the prey animals as quickly as possible. The interesting thing about this problem is that it is very well defined and yet leaves open the design of possible modes of cooperation. The students are asked to specify and design predator agents with increasing levels of sophistication In particular, they are asked to extend and stepwise refine the architecture shown in Fig. 4. For reactive agents, the perception subsystem and the control model define the overall behavior of agents. Students suppose that the prey animals emit a signal, whose intensity decreases with distance. The signal plays the role of an attractor. Students use the reactive (emergent) task allocation technique covered earlier in the class to design reactive agents. The design of communicative agents introduces the communication subsystem to improve the perceptive capacities of agents. At that level, predator agents are capable of subscribing to other

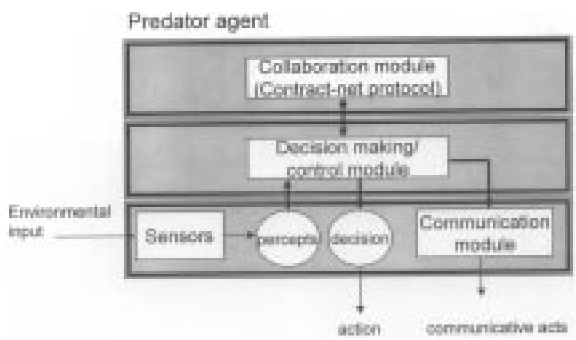

Fig. 4. The predator agent.

predator agents to be notified by symbolic messages about the location of the perceived prey animals.

Collaborative agents bring in the concept of a contract net discussed earlier in the class. The contract net provides a market-like protocol for task allocation, in addition to the management of subscriptions of communicative agents. For collaborative agents, the organization subsystem of the agent is introduced to realize the contract-net protocol to establish teams. The students also extend the control module to implement the following rules.

1. If the agent is a leader and it perceives a prey animal then it moves towards it and sends the information concerning its position to its team members.
2. If the agent is not a leader, but is committed, then after receiving information concerning the position of the prey animal, it moves towards it.
3. If the agent is not committed and perceives a prey animal, then it becomes a leader and establishes a team using the contract-net protocol.
4. If the agent is a leader and perceives captured prey, then it breaks the team.

Applications

The course concludes with an overview of various types of applications, for which agents play critical roles. These applications include workflow and business process management, distributed sensing, agents for information retrieval and management, and e-commerce. After a brief synopsis of such applications, agent-supported simulation is illustrated via a sample project conducted at the Auburn Modeling and Simulation Laboratory. The study involves the use of specific types of intelligent agents such as a mediator, broker, and matchmaker who facilitate interoperability of simulations [31].

STUDENT EXPERIENCE

Students use the experience gained through the case study along with the various models of agent architectures and cooperation strategies presented in class to conceptualize, design, and implement agent-based simulation models. As for experimentation with the models, they follow a unique strategy that is radically different from the approach advocated in conventional engineering simulation modeling courses (i.e., predictive modeling). The advocated strategy is based on the premise of computational epistemology, which deals with the question, 'How can we learn about a phenomena by performing computational experimentation via simulation?'

Students immediately observe that agent-based simulations bring a very different framework for teaching how to learn and understand engineering systems. From this viewpoint simulations are viewed as computational experiments by which one can explore and gain insight about the system. While the use of trace-driven simulations that use field-collected data to drive a model is highly encouraged in their introductory simulation courses, the potential avenue of exploratory modeling, which provides an alternative rationale for using models to understand complex systems, is a new perspective.

The students find it difficult to accept the fact that, in exploratory analysis, the value of a simulation does not depend on the degree of correspondence between the simulation input data and observed system behavior. Rather, the value of a simulation model is in clarifying and specifying new questions that students might want to explore. Gradually, students learn to ask interesting questions by performing, for instance, agent-based simulation of processes. Such questions include: 'What are the minimal conditions for the emergence of allegiance and trust within and between teams?', 'What tends to promote such emergence?', 'How is the project dynamics affected by the number of teams?', and 'What aspects of the team coordination or task allocation behavior can lead to the failure of the overall project?' These questions may not have previously been considered and, if explored with the model, they might lead to new ways of thinking about comparable questions in the real world.

The case study enabled students to observe how cooperation strategies proceed from a top-down design by defining functions that the system needs to carry out: detection of prey animals, setting up teams, allocation of roles, reorganization of teams, and so on. Students further observe how these functions can be realized in the form of adapted behaviors requiring a communication system that allows for dialogue and distributed decision-making. A noteworthy learning experience that proceeds from the case study involves the use of reactive agents. The process of emergent task allocation and coordination on the basis of propagated signals illustrates how reactive strategy develops bottom-up in comparison with the top-down design of the cognitive agents.

The feedback from students indicates various limitations pertaining to the current state of the art of the agent tools, toolkits, and environments. Many of the conceptual ideas such as speech acts, conversation protocols, and deliberation mechanisms are not supported by existing simulation toolkits and libraries. This requires them to develop their own applications over existing fine-grain agents. Students find this quite time-consuming and prone to error.

CONCLUSIONS

Teaching the use of intelligent agents in modeling, design, and simulation of engineering systems and promoting exploratory analysis of such phenomena present two distinct yet related challenges. In this paper we presented a strategy to expand our horizons in using agents and its associated technologies to support Modeling and Simulation of engineering applications. The primary distinction of the proposed course structure is its focus on the theory and methodology underlying agents. It is argued that learning and understanding the fundamental principles underlying the agent paradigm and multi-agent systems is a prerequisite to developing agent-based simulations or to simulating agent systems in various application domains. To this end, the structure and the contents of the ADS course is based on methods for specifying, designing, and implementing various types of agents, their communication, and cooperation mechanisms. We illustrate how to bridge the gap between the theory and practice by using a case study that provides a sound and powerful test bed to study various forms of collaboration, coordination, and conflict resolution techniques covered in class.

REFERENCES

1. G. Weiss, Multi-Agent Systems: A Modern Approach to Distributed Artificial Intelligence, (1999).
2. H. V. D. Parunak, A. Ward, M. Fleischer, and J. Sauter, A marketplace of design agents for distributed concurrent set-based design, *Proceedings of ISPE/CE97: Fourth ISPE International Conference on Concurrent Engineering: research and Applications*, (1997).
3. N. Avouris, Cooperating knowledge-based systems for environmental decision support, *Knowledge-based Systems*, **8**(1), 1995, 39–54.
4. L. Varga, N. Jennings, and D. Cockburn, Integrating intelligent systems into a cooperative community for electricity distribution management, *International Journal of Expert Systems*, **7**(4), 1994, 168–179.
5. J. Mori, H. Torikoshi, K. Nakai, K. Mori, and T. Masuda, computer control system for iron and steel plants, *Hitachi Review*, **37**(4), 1988, 251–258.
6. J. Cuena, J. Hernandez, and M. Molina, Knowledge-based models for adaptive traffic management, *Transportation Research*, **3**(5), 1995, 311–337.
7. J. M. Bradshaw, An introduction to software agents, in *Software Agents*, J. M. Bradshaw (ed.), AAAI Press / MIT Press, Menlo Park, CA, (1997), pp. 3–46.
8. M. Wooldridge, Agent-based Software Engineering. *IEE Proceedings in Software Engineering*, **144**, (1), 26–37. (1997).
9. R. Axelrod, Complexity theory in the social sciences, http://www.personal.umich.edu/~axe/PS793.htm. (2005).
10. L. Tesfatsion, ACE: Agent-based computational economics, http://www.econ.iastate.edu/classes/econ308/tesfatsion/syl308.htm (2005).
11. CASOS, The CASOS Ph.D program, http://www.casos.cs.cmu.edu/education/phd/phd_casos.html (2005).
12. ADS, Agent-Directed Simulation courses, http://www.eng.auburn.edu/~yilmaz/ADS-Course.htm (2005).
13. P. Blikstein and U. Wilensky, MaterialSim: An agent-based simulation toolkit for materials science learning, *Proceedings of International Conference on Engineering Education University of Florida*, (2004).
14. R. Xuefeng. Teaching intelligent agents to industrial engineering majors, *Int. J. Eng. Ed.*, **21**, 2005, 758–766,
15. B. H. Far, Agent-oriented software engineering, http://www.unel.ucalgary.ca/people/far/lectures/SENG609-22/index.html, (2006)
16. W. Wakeland. Agent-based simulation, https://webct.pdx.edu/public/sysc610abs, (2006).
17. MAAI. Military applications of artificial intelligence, http://carlisle-www.army.mil/usacsl/divisions/std/branches/keg/maai.htm, (2006).
18. L. B. Holder and D. J. Cook, An integrated tool for enhancement of artificial intelligence curriculum, *Journal of Computing in Higher Education,* **12**(2), 2001.
19. Serenko and B. Detlor. Agent toolkit satisfaction and use in higher education, *Journal of Computing in Higher Education*, **15**(1), 2003, 65–88.
20. K. Kollman, S. Page, and R. Riolo, Agent-based computer intensive modeling, http://www.h-net.org/announce/show.cgi?ID=125833, (2006).
21. B. LeBaron, Agent-based modeling, http://people.brandeis.edu/~blebaron/classes/econ326f/syl.html, (2006).
22. ANL, Agent-based modeling and simulation for business complexity, http://www.dis.anl.gov/abms, (2006).
23. R. G. Little, Organizational culture and the performance of the critical infrastructure: Modeling and simulation in socio-technological systems, *Proceedings of the 38th Hawaii International Conference on System Sciences, (2005).*
24. L. Yilmaz and T. Ören (eds) Agent-directed simulation, special issue of the *Transactions of the Society for Computer Simulation International*, **81**(7), 2005.
25. B. Faverjon, Hierarchical object models for efficient anti-collision algorithms, *Proc. IEEE Int. Conf. on Robotics and Automation*, Scottsdale, AZ, (1989) pp. 333–340.
26. G. Booch, J. Rumbaugh, and I. Jacobson, *The Unified Modeling Language User Guide*, Addison-Wesley, (1999).
27. COMP8700 Projects. http://www.eng.auburn.edu/~yilmaz/comp8700S05, (2006)

28. A.M.C. Campos and D. R. C. Hill, Web-based simulation of agent behaviors, *1st International Conference on Web-based Modeling and Simulation* (P. Fishwick, D.R.C. Hill, and R. Smith (eds)), SCS, San Diego, CA, (1998) pp. 9–14.
29. L. Gasser, N. Rouquette, R. Hill, and J. Lieb, Representing and using organizational knowledge in distributed AI systems, in *Distributed Artificial Intelligence, Vol. 2*, Pitman, (1989).
30. J. Ferber, Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence, Addison-Wesley, (1999).
31. L. Yilmaz and S. Paspuletti, Toward a meta-level framework for agent-supported interoperation of defense simulations, *Journal of Defense Modeling and Simulation* (forthcoming) (2006).

Levent Yilmaz is Assistant Professor of Computer Science and Software Engineering in the College of Engineering at Auburn University. Dr Yilmaz was awarded his Ph.D. and M.S. degrees from Virginia Polytechnic Institute and State University (a.k.a. Virginia Tech). He received his B.S. degree in Computer Engineering from Bilkent University, in Turkey. His research interests are in advancing the theory and methodology of simulation modeling, agent-directed simulation (to explore and understand conflicts, dynamics of organizations, and human behavior), and education in simulation modeling. He is a member of ACM, IEEE Computer Society, Society for Computer Simulation International, and Upsilon Pi Epsilon.