Agent-Directed Simulation for Systems Engineering

Philip S. Barry,[1] Matthew T. K. Koehler[1] and Brian F. Tivnan[1]

[1]The MITRE Corporation
7515 Colshire Dr.,McLean, VA, USA
pbarry@mitre.org, mkoehler@mitre.org, btivnan@mitre.org

Keywords: Agent Directed Simulation, Infrastructure for Complex-systems Engineering, Stadium Defense.

Abstract

As the fielding of enterprise systems of systems becomes common it becomes increasingly important to understand the interactions between the systems as well as the important role that human behavior plays. This paper suggests that Agent-Directed Simulation is a valuable and crucial analysis tool for the Systems Engineer. The paper examines the concept of Agent-Directed Simulation for Systems Engineering and then introduces the notion of Human Complex Systems. An analysis infrastructure is described and a case study is provided to illustrate the concepts.

1 INTRODUCTION

The development of Systems of Systems (SoS) is becoming commonplace; the government and private industry are increasingly faced with the problem of investing large amounts of money in SoS that cannot be fully specified in a requirements document, and cannot be fully tested or in many cases even prototyped. Compounding this problem is that SoS are not simply very complicated systems like a racecar or a communications satellite. While complicated systems have many parts that interact with each other in nontrivial ways, we can describe the interaction using well-understood laws of mechanics and physics. SoS on the other hand present a constantly changing topology, evolving over time and space. Moreover, human are an important, if not the most important, component of these systems. Humans interact with each other as well as with technology to provide new systemic capabilities in ways that are often impossible to define a priori. This differs from the classical view of systems where the human element was considered outside the system or as a user of the system.

With this expansion of scope there has come a recognition that new approaches are needed (e.g., [1], [2], [3]and[4]). There has been good progress in developing paradigms that provide relatively static descriptions of the SoS, hierarchically decomposed views of the enterprise with robust descriptions of the interfaces. While providing a valuable base for analysis, these approaches do not provide a clear path for gaining insight into the dynamic and evolutionary behavior of SoS.

To 'engineer' a SoS we propose that additional techniques are required. The sheer number of components in fielded SoS and their associated complex behaviors argues strongly against using closed forms of analysis. However, systems engineers are often required to advise in areas such as portfolio management that require a quantitative understanding of the possibility space of complex interactions. We suggest the employment of agent-directed simulation (ADS) alongside rigorous, proven systems engineering techniques and enterprise architecture development to gain insight into the specification, design and evolution of systems of systems.

Our argument for the necessity, and utility, of ADS is built upon a foundation of work from Wegner, Simon, Doyle, Schelling , Buss, et al., Epstein and Axtell. Wegner [5, 6] showed that as powerful as closed form algorithmic analysis is, that it essentially represents the system as a Turing machine; whereas, allowing the algorithms to interact, (e.g. an interaction machine or ADS), is a far more powerful analytic representation. Simon [7] argued that complex systems can be meaningfully represented as collections of subsystems; hierarchies of nearly decomposable systems. Within these collections the state of each subsystem through time is only weakly influenced by the other subsystem. The ability to represent these complex systems (or SoS) in this way makes systems engineering of them possible and creating ADSs of them meaningful and useful. Csete and Doyle [8] stresses not only the importance of modules in complex SoS but also the criticality of common interfaces between the components. These interfaces are critical for large collections of systems to function properly. Schelling [9] argued that the interaction of subcomponents of a system, without meaningful centralized control, can have enormous impact on the behavior and performance of the system as a whole.

Moreover, Buss et al. [10] proved that under ideal conditions a model of SoS as a collection of homogeneous automata with a global control rule that is independent of the states of any of the automata the system will be predictable (i.e. the future state of the system is computable) in polynomial time; if, however, the global control rule is not independent of the states of the automata then the prediction of the system become PSPACE-complete. This means that one can do no better to understand the future states of the system than to simulate it. More recently, Epstein [11] has argued, and demonstrated, that it can be far more insightful to examine the dynamics of a system as it moves through time than to "solve" for the system's equilibria. It is trivial to specify a system that may not be able to obtain it's equilibria from plausible initial conditions or that it may take longer than the universe has been in existence for the system to achieve an equilibrium state. This being the case, simulations can prove far more useful from a systems engineering perspective to gain insight into the actual, or meaningful, performance of the system in question. Finally, Epstein and Axtell [12], and Epstein [13] argue for a generative nature to our understanding of social systems—"If we didn't grow it, we didn't explain it." They argue that understanding the emergent properties of a SoS (the macro-dynamics) comes from specifying the components and then allowing the system to move forward in time. In this way one generates a sufficiency theorem—this specification with this set of input is sufficient to generate this output. Again, this argues that ADS for SoS engineering is not only useful, but may be one of the only ways to meaningfully understand and design complex SoS.

2 AGENT-DIRECTED SIMULATION FOR THE SYSTEMS ENGINEERING OF HUMAN COMPLEX SYSTEMS

By contrast to the static models described above, simulation often represents the method of choice for researchers (e.g., Carley and Svoboda [14], Epstein and Axtell[12], Levinthal [15], March [16] and McKelvey [17,18] among others) to explore complex dynamics often found in Human Complex Systems [19]. Human Complex Systems (HCS) are systems of systems that include active, human participants beyond simple roles of systems operator (e.g, large, public venue including crowd and security personnel; metropolitan area experiencing a pandemic; and financial exchange including traders and regulators). Analyzing HCS requires a refactoring of tools; McKelvey and Cyert and March call for a specific class of simulation for HCS, namely simulation with ADS [18, 20, 21]

While simulations applied to the study of HCS first occurred as much as forty years ago (e.g., Cyert and March [21]), only recently has it begun to generate a broader acceptance [22]. Not only special issues but entire journals are now dedicated to simulation and its application to the science of HCS (e.g., Carley [23], Lissack [24] and Gilbert [25]). This acceptance stems from two critical aspects of simulation research: (a) simulation allows researchers to explore the inherent complex dynamics of HCS [22, 26], hence (b) simulation research allows for the conduct of experiments that would typically be impossible or impractical in the physical world [27].

Stressing the value of simulations for theorizing [28], Axelrod [29, p. 23-24] believes that simulation offers a new vehicle for conducting scientific research that differs from induction (i.e., the "discovery of patterns in empirical data") and deduction (i.e., "specifying a set of axioms and proving consequences that can be derived from those assumptions"). On the one hand, simulation research resembles deduction in that simulations start with a set of assumptions. On the other hand, the simulation generates data to be inductively analyzed. Axelrod [29, p. 24] refers to simulation research as "thought experiments" since the assumptions might seem simple but the results are often counter-intuitive (i.e., the nonlinear, macro-level effects of interacting agents known as emergent properties).

Axelrod [29] provides further support for simulation as an alternative to the rational actor / choice assumptions. Because the rational actor / choice assumption allows for deductive, closed-form analysis, researchers are willing to overlook the bounded rational limitations of their actors [30]. The primary alternative to the rational actor / choice assumption lies in some form of adaptive behavior. Due to the complex effects of social interactions, Axelrod [29] asserts that ADS offers the only vehicle to study sets of actors who possess an adaptive capacity.

With the growing acceptance of simulation in the design and engineering of HCS due in no small part to March's research, several leading scholars have called for the formal use of ADS (e.g., Anderson [32], Axelrod[29], Dooley [22] and McKelvey [17, 18]). As the primary tool of complexity theorists, ADS assume that agents behave in a stochastic, nonlinear manner and that agents possess a nonlinear capacity to adapt over time. This stochastic, nonlinear behavior of agents is consistent with the stochastic, idiosyncratic microstates of HCS. That is, despite institutional influences [33, 34], strong forces remain to idiosyncratically steer both the behaviors of individuals and the conduct of aggregate processes [31]. Among others, such forces might include unique organizational cultures, the unique set of suppliers and customers (i.e., organizations are each embedded within a unique social network) and the unique interaction network of different individuals each with his/her own personal history in different contexts. Therefore, agent activity in an ADS can offer an excellent representation of the adaptive and idiosyncratic behavior of an HCS and that of its human agents.

3 AN INFRASTRUCTURE FOR THE ENGINEERING OF HUMAN COMPLEX SYSTEMS

ADS typically represent the *only* way one may experiment and test the SoS in question. With that in mind, we developed the Infrastructure for Complex-systems Engineering (ICE). With its model-centered core, the ICE is a collection of software tools, computational hardware, and methodologies that allow one to move from abstract thought experiments to "operational" testing and optimization of an HCS.

Consistent with solid SE practices, an application of the ICE starts with an assessment of what information about the SoS in question is currently known; this includes Subject Matter Expert understanding of the components of the SoS and how they are interconnected, measures of performance for individual components (usually in isolation), and so on. Once enough information is amassed about the SoS in question, one moves to the prototyping stage. What constitutes an adequate prototype will be driven largely by what the SoS is and the questions to be asked about it. For example, if the prototype does not need to be overly large in scale (less than 10,000 entities) we have had very good luck with NetLogo [35]. However, there are times when even the prototype must be very large scale. In those cases, we move to Repast [36]. Eventually as the prototype stabilizes we, typically, port the NetLogo simulation to Repast for deployment on a high-performance cluster computer. Though Repast is more tightly integrated with our cluster computer, NetLogo can be run on the cluster, also. Therefore, if scale or high-performance is not a driving concern we may not move away from our prototype.

We use NetLogo and/or Repast to handle the representation of the SoS as a whole. Usually, however, there are components of the SoS that are of particular importance to the functioning and performance of the SoS. These components are handled specially in the ICE framework. These high importance components are modeled separately at as high a resolution as possible. The family of simulations is now run together to represent the functioning of the system.

Use of a cluster computing system allows us to scale very large if necessary and also to run many replicates of the simulation to perform a Monte Carlo analysis of the modeled SoS. Intelligent design of experiments is very important here as the parameter space associated with these models can be nearly infinite. As part of the cluster computing system we have an automated genetic optimization framework. Optimization over a complex space requires a reasonable degree of verisimilitude. Consequently, our methodology employs detailed physical models and vetted behavioral models where possible. We also seek to provide realistic physical models of the geometry where that level of detail is important to the question at hand.

Concurrent with the model development is the development of a preference model. Most decisions involve a number of (frequently competing) criteria. Our methodology characterizes the goodness of a given decision by developing a preference structure and an overall utility equation through interaction with subject matter experts (SMEs). This step is described in detail in the next section.

To explore the complex interaction of the components we employ an ADS framework. We begin with rapid prototyping to facilitate communication with the subject matter experts and gain additional insight into the problem. Subsequent to the initial runs, we develop a higher fidelity model that is more scalable and designed to make use of a high-performance computing environment. Additional tools such as Matlab are employed to develop additional high-resolution models as warranted. Parallel model development is important in systems where emergence is a feature of study. This is the case because once the scalable model is developed we can compare its dynamics with those of the prototype and determine their similarities [23]. If both models demonstrate the same dynamics then we can be more confident that these results are a true attribute of the system rather than a bug in our code or an artifact of the modeling framework we chose.

To explore a variety of system responses, formal design of experiments [37] are created to use statistical techniques to manage the inherent combinatorics. These experiments are then run in a high-performance computing environment to investigate a wide variety of behaviors that are implied from the stochastic and/or adaptive behavior of the agents. Frequently, the number of individual simulation runs will approach 10^5 or more.

As the simulation runs are completed, two types of analyses are conducted. The first uses optimization techniques such as genetic algorithms to drive to areas of optimal parameter combinations, optimality being defined using the utility analysis described below. The second conducts statistical analysis to characterize the relationship of the various parameters within the simulation. While the optimization algorithms will generally find families of solutions, the statistical analyses will provide additional insight into areas of concern as well as statistical anomalies that might have catastrophic consequences. This overall methodology is illustrated in Figure 1.

4 CASE STUDY: DEFENDING THE STADIUM

ADS enhanced systems engineering has been used to explore the best ways to best defend a stadium against possible terrorists. In this case, stadium defense is a complex interaction of sensors, tactics, and decision making. Typically, a heterogeneous mix of sensors is fielded. The mix represents different modalities of detection (e.g., infrared imaging, millimeter wave radar, etc.) as well as different performance profiles. In fact, the exact performance of the sensor is highly dependent upon the placement as well as environmental conditions such as crowd flow, occlusion by spurious objects, humidity, ambient temperature and even sunlight.

Initially a straight-forward agent model of sensors was considered for this analysis. However, early studies conducted using simple parameterized sensor models resulted in overly optimistic estimates for systemic detection of prohibited materials. Consequently, the decision was taken to replace simple sensor agents with high fidelity deterministic models of the sensors which resulted in much more realistic results.

On the other hand, agent models were used to represent the decisions of law enforcement officers who decide which individuals to examine with the sensors, and if the individual is suspicious enough to interdict, to keep tracking, or to break contact. Further, as the decisions are done in a resource constrained environment, there is not a straightforward deterministic algorithm for making the decisions; they are often fraught with uncertainty and error.

The decision that a law enforcement officer takes is a study in complexity. Dedicating sensors to a potential person of interest reduces the opportunity to scan other individuals. To complicate matters even more the number of law enforcement officers available to interdict suspicious persons is relatively small; if an interdiction decision is taken there is an opportunity cost for interdicting other potential persons of interest. Further, interdiction decisions will affect the behavior of other individuals in the crowd, potentially providing warning to individuals with prohibited materials.

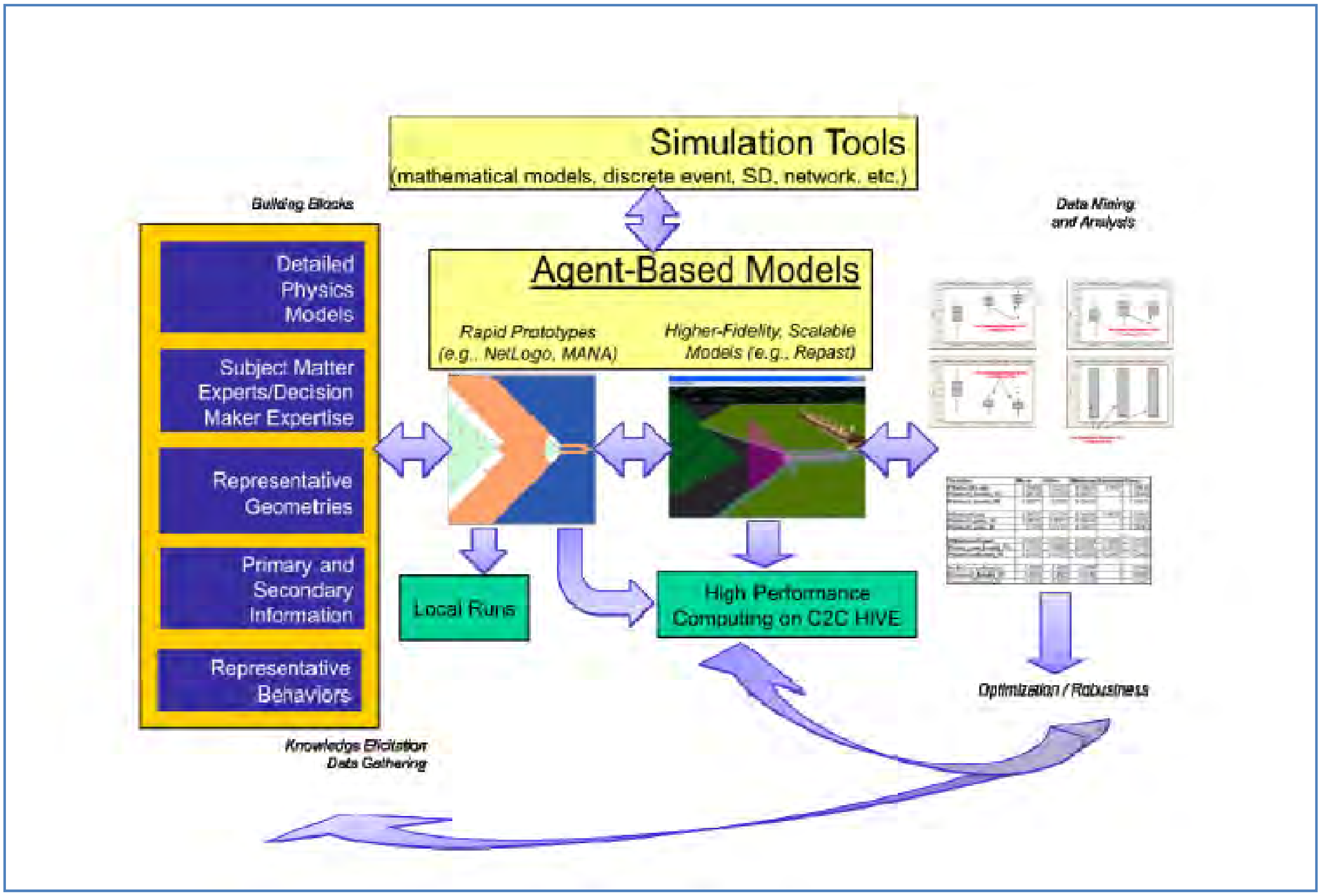

Figure 1. The Infrastructure for Complex-systems Engineering.

As the combinatorics of the possible sensor placements and the wide variety of tactics was daunting, it was determined that simulation could be used to provide recommendations to optimize stadium defense. An experiment was formally designed and a number of simulation excursions were run. The various combinations of sensor placements and tactics were then scored in a utility framework, with specific recommendations briefed to guide live experimentation.

4.1 Simulation Setup

Figure 2 illustrates the three types of agents in the prototype model: Security, Civilians, and “bad actors”. Bad actors model terrorists; agents that have a goal of getting through the turnstiles undetected carrying explosives. Bad actors must traverse a corridor from left to right where they are likely to encounter sensors as well as security guards. If a bad actor successfully crosses the turnstile, he is considered successful.

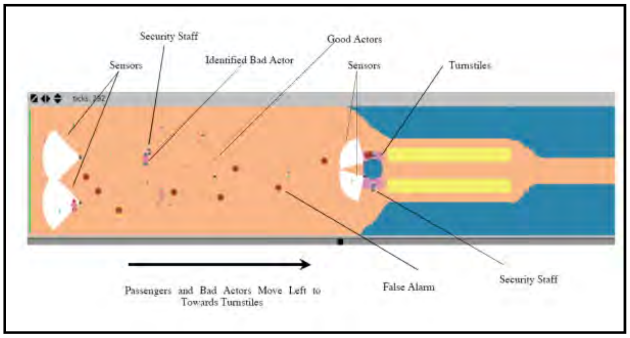

Figure 2. Simulation Setup.

Bad actors will take evasive measures to avoid security guards. If the security guards close off potential escape routes and the bad actor considers the situation hopeless, then the bad actor will exhibit satisficing behavior and detonate the explosives. In this simulation, bad actors do not act collaboratively. Security agents have a primary goal to interdict the bad actors. Security agents are throughout the corridors as well as around the turnstiles. Security agents use fused data from the sensors to evaluate whether a passenger agent is a bad actor or not. As false alarms are possible, non-bad actors may be targeted as bad actors and stopped by the security agents. The remainder of the agents are "innocent passengers" that are instantiated as a rate per unit time. The number created each time-step is drawn from a random-exponential distribution. Upon instantiation a small percentage of the passengers (0.005%) may be designated as an individual with a bomb or a bad actor.

4.2 Experimental Results

The simulation was run for a wide variety of sensor placements, varying the x and y coordinates of two passive infrared (IR) sensors and a passive millimeter wave (mmw) sensor as well as their respective angle with respect to the incoming traffic. Additionally, the amount of evidence necessary for the security guards to interdict a possible suspect was varied. In other words, varying the sensors directly affected the amount of information received. Changing the evidence threshold resulted in varying the likelihood of false positives or false negatives. As discussed above, the "goodness" of a configuration of sensors in combination with the likelihood of interdiction was modeled as a utility function. Two primary measures of effectiveness were considered; for a given configuration the probability that an explosive would be detected and the probability that there would be a false alarm, meaning an explosive was indicated where no explosive existed. Two thousand possible combinations were modeled; each design point was run thirty times as there were a number of stochastic features in the simulation.

The data indicated a number of interesting aspects. First, for any given configuration there was wide variability in the results. However, some general trends were observed. Figure 3a plots utility against numbered sensor configurations. From sensor configurations 0 – 26, the general trend is that the mmw sensor starts out significantly in front of the IR sensors and then progressively moves closer. At sensor configuration 27-53, the mmw sensor is behind the IR sensors. As a general practice, it would appear that placing the IR sensors behind the mmw sensor is preferred.

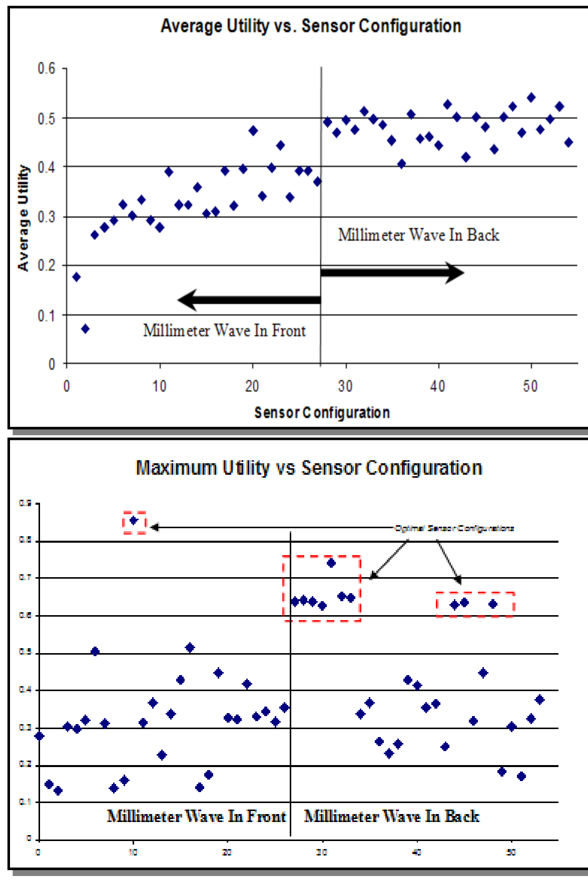

Figure 3. a) Average Utility vs. Sensor Configuration, b) Maximum Utility vs. Sensor Configuration.

Figure 3b illustrates the maximum utility plotted against sensor configuration. Consistent with the indication from shown in Figure 3a, the best performing configurations from a maximum utility perspective are where the mmw sensor is behind the

IR sensors. There are two groupings of similar configurations that performed roughly the same. However, the highest utility occurred during a run using sensor configuration 10, where the mmw sensor is in front of the IR sensors. This can be viewed as an anomalous result, possible but unlikely as the average utility is much less.

For both the average utility as well as the maximum utility, optimal temperament configuration for the security forces was a mildly aggressive but not overly aggressive profile. Overly aggressive security forces result in too many false alarms whereas passive security forces miss too many threats. This result seemed relatively consistent across all of the sensor configurations.

5 SUMMARY

As our development of systems becomes increasingly complex, the practice of systems engineering must evolve. Our tools and techniques must expand to handle issues of combinatorial complexity, long term system evolution, managerial independence and emergent behavior. Increasing recognition that viable models of the organizational element of an enterprise require concordant maturation of our toolset. We have suggested here that Agent Directed Simulation can provide a valuable tool to gain insight into complex enterprises. The application of ADS builds on a long history of successful work in organizational theory married with traditional systems engineering and enterprise architecture. We have found that with significant computational capabilities we can find statistically interesting insights into complex SoS, allowing us to provide quantitative engineering direction to all phases of the systems lifecycle for a SoS. The use of ADS has been shown to provide significant value for the design and implementation of SoS and will continue to gain importance as our systems grow in scope and complexity.

References

[1] M. W. Maier, "Architecting Principles for Systems-of-Systems," *Systems Engineering*, vol. 1, pp. 267-284, 1998.

[2] P. G. Carlock and R. C. Fenton, "Systems of Systems (SoS) Enterprise Systems Engineering for Information Intensive Organizations," *Systems Engineering*, vol. 4, pp. 242-251, 2001.

[3] A. P. Sage and C. D. Cuppan, "On the Systems Engineering and Management of Systems of Systems and Federations of Systems," *Information, Knowledge, and Systems Management*, vol. 2, pp. 325-345, 2001.

[4] A. J. Kriegel, *Behind the Wizard's Curtain: An Integration Environment for a System of Systems*. Washington, DC: CCRP, 1999.

[5] P. Wegner, "Why Interaction is More Powerful than Algorithms," *Communications of the ACM*, vol. 40, pp. 80-91, 1997.

[6] P. Wegner and D. Goldin, "Computation Beyond Turing Machines: Seeking Appropriate Methods to Model Computing and Human Thought," *Communications of the ACM*, vol. 46, pp. 100-102, 2003.

[7] H. A. Simon, *The Sciences of the Artificial*, 3rd ed. Cambridge, MA: MIT Press, 1996.

[8] M. E. Csete and J. C. Doyle, "Reverse Engineering of Biological Complexity," *Science*, vol. 295, pp. 1664-1669, 2002.

[9] T. C. Schelling, *Micromotives and Macrobebahavior*. New York: Norton, 1978.

[10] S. R. Buss, C. H. Papadimitriou, and J. N. Tsitsiklis, "On the Predictability of Coupled Automata: An Allegory about Chaos," *Complex Systems*, vol. 5, pp. 525-539, 1991.

[11] J. M. Epstein and R. A. Hammond, "Non-Explanatory Equilibria: An Extremely Simple Game with (Mostly) Unattainable Fixed Points," *Complexity*, vol. 7, pp. 18-22, 2002.

[12] J. M. Epstein and R. L. Axtell, *Growing Artificial Societies: Social Science from the Bottom Up*. Washington: Brookings Institution Press, 1996.

[13] J. M. Epstein, "Agent-Based Computational Models and Generative Social

[14] K. M. Carley and D. M. Svoboda, "Modeling Organizational Adaptation as a Simulated Annealing Process," *Sociological Methods & Research*, vol. 25, pp. 138-168, 1996.

[15] D. A. Levinthal, "Adaptation on Rugged Landscapes," *Management Science*, vol. 43, pp. 934-950, 1997.

[16] J. G. March, "Exploration and Exploitation in Organizational Learning," *Organization Science*, vol. 2, pp. 71-87, 1991.

[17] B. McKelvey, "Avoiding Complexity Catastrophe in Coevolutionary Pockets: Strategies for Rugged Landscapes," *Organization Science*, vol. 10, pp. 294-321, 1999.

[18] B. McKelvey, "Self-Organization, Complexity Catastrophe, and Microstate Models at the Edge of Chaos," in *Variations in Organization*

Science: In Honor of Donald T. Campbell, J. A. C. Baum and B. McKelvey, Eds. Thousand Oaks, CA: Sage, 1999, pp. 279-310.

[19] B. F. Tivnan, "Coevolutionary Dynamics and Agent-Based Models in Organization Science," in *Proceedings of the 2005 Winter Simulation Conference*, M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, Eds. Piscataway, NJ: Institute for Electrical and Electronics Engineers, 2005, pp. 1013-1021.

[20] B. McKelvey, "Complexity Theory in Organization Science: Seizing the Promise or Becoming a Fad?," *Emergence*, vol. 1, pp. 5-32, 1999.

[21] R. M. Cyert and J. G. March, *A Behavioral Theory of the Firm*. Englewood Cliffs, NJ: Prentice-Hall, 1963.

[22] K. J. Dooley, "Simulation Research Methods," in *Companion to Organizations*, J. A. C. Baum, Ed. Oxford, UK: Blackwell, 2002.

[23] K. M. Carley, "Computational and Mathematical Organization Theory: Perspective and Directions," *Computational and Mathematical Organization Theory*, vol. 1, pp. 39-56, 1995.

[24] M. R. Lissack, "Editor's Note," *Emergence*, vol. 1, pp. 3-4, 1999.

[25] N. Gilbert, "Editorial," *Journal of Artificial Societies and Social Simulation*, vol. 1, 1998.

[26] K. J. Dooley and A. H. Van de Ven, "Explaining Complex Organizational Dynamics," *Organization Science*, vol. 10, pp. 358-375, 1999.

[27] N. Gilbert and K. G. Troitzsch, *Simulation for the Social Scientist*. Philadelphia, PA: Open University Press, 1999.

[28] K. E. Weick, "What Theory is *Not*, Theorizing *Is*," *Administrative Science Quarterly*, vol. 40, pp. 385-390, 1995.

[29] R. Axelrod, "Advancing the Art of Simulation in the Social Sciences," in *Simulating Social Phenomena*, R. Conte, R. Hegselmann, and P. Terna, Eds. Berlin: Springer, 1997, pp. 21-40.

[30] H. A. Simon, *Administrative Behavior*, Third ed. New York: Free Press, 1976.

[31] B. McKelvey, "Quasi-Natural Organization Science," *Organization Science*, vol. 8, pp. 352-380, 1997.

[32] P. Anderson, "Complexity Theory and Organization Science," *Organization Science*, vol. 10, pp. 216-232, 1999.

[33] L. G. Zucker, *Institutional Patterns and Organizations: Culture and Environment*. Cambridge, MA: Ballinger, 1988.

[34] W. R. Scott, *Institutions and Organizations*. Thousand Oaks, CA: Sage, 1995.

[35] U. Wilensky, "NetLogo." Evanston, IL.. Center for Connected Learning and Computer-Based Modeling, Northwestern University, 1999.

[36] E. Tatara, M. J. North, T. R. Howe, N. T. Collier, and J. R. Vos, "An Introduction to Repast Modeling by Using a Simple Predator-Prey Example," presented at Agent 2006 Conference on Social Agents: Results and Prospects, Chicago, IL, 2006.

[37] J. P. C. Kleijnen, S. M. Sanchez, T. W. Lucas, and T. M. Cioppa, "State - of - the - Art Review: A User's Guide to the Brave New World of Designing Simulation Experiments," *INFORMS Journal on Computing*, vol. 17, pp. 263-289, 2005.