Modeling Technique

Agent-Based Simulation Comes of Age

Software opens up many new areas of application.

By Douglas A. Samuelson and Charles M. Macal

Agent-based simulation is establishing itself as a serious, useful area of study. On Aug. 21-26, the first World Congress on Social Simulation will be held in Kyoto, Japan, jointly sponsored by the three major international professional societies in the subject: the Pacific Asian Association for Agent-based Approach in Social Systems Sciences (PAAA), the European Social Simulation Association (ESSA) and the North American Association for Computational Social and Organization Sciences (NAACSOS).

Regular readers of *OR/MS Today* may recall that computational mathematical organization theory (CMOT) arose in significant part from a special interest group within TIMS and ORSA. Now, however, the field has expanded far beyond its earlier boundaries, to link up with other groups of people and encompass work in a range of disciplines including computer science, artificial intelligence, economics, sociology, physics, biology, anthropology and even archaeology.

One of the most important reasons for this expansion is that software toolkits released within the last few years have made agent-based modeling easy enough to be attractive to many practitioners from a variety of subject areas. Among the most significant of these toolkits are Swarm, Repast, NetLogo, AnyLogic, MASON and Ascape. Of these, only Swarm, introduced in 1996, has been available for more than six years.

The now-classic Gilbert and Troitzsch book on simulation in social science, published in 1999, had a chapter of examples of cellular automata and agent-based modeling — with a few examples in LISP, reflecting the artificial intelligence roots of the field. At the time, Swarm was the only agent-based modeling language available. The new edition, published last year, has many more example models, now in NetLogo.

Complex Adaptive Systems and Agent-Based Modeling and Simulation

The essential idea of agent-based modeling and simulation (ABMS) is that many phenomena, even very complex ones, can best be understood as systems of autonomous agents that are relatively simple and follow relatively simple rules for interaction. Applications range from modeling agent behavior in the stock market (Arthur et al. 1997) and supply chains (Macal 2003, 2004), to predicting the spread of epidemics (Bagni et al. 2002) and the threat of bio-warfare; from modeling bacterial cell behavior (Emonet et al. 2005) to consumer behavior to understanding the fall of ancient civilizations (Kohler et al 2005), to name just a few.

Agent-based modeling and simulation is also an experimental technique, a framework for developing electronic laboratories in which the most detailed assumptions about individual agents, their behaviors and interactions can be varied and explored in silico. Some commentators have gone so far as to suggest that ABMS, with the computational experimental framework it offers, is a new way of doing science (Axelrod 1997; Epstein and Axtell 1996, Wolfram 2005).

The fundamental feature of an agent is the capability to make independent decisions: An agent is a discrete individual with a set of characteristics and rules governing its behaviors and decision-making capability. An agent may have additional rules that modify its rules of behavior. An agent can function independently in its environment and in its dealings with other agents, at least over a limited range of situations. An agent is goal-directed, having goals to achieve (not necessarily objectives to maximize) with respect to its behaviors. An agent is flexible and has the ability to learn and adapt its behaviors over time based on experience, which generally requires some form of memory.

The field of complex adaptive systems (CAS) was originally motivated by investigations into adaptation and emergence of biological systems. CAS have the ability to self-organize and dynamically reorganize their components in ways better suited to survive and excel in their environments. John Holland, a pioneer in the field, identified properties and mechanisms common to all CAS (Holland 1995). CAS properties are:

1. aggregation: allows groups to form,
2. nonlinearity: invalidates simple extrapolation,
3. flows: allow the transfer and transformation of resources and information, and
4. diversity: allows agents to behave differently from one another and often leads to the system property of robustness.

CAS mechanisms are:

1. tagging: allows agents to be named and recognized,
2. internal models: allows agents to reason about their worlds, and
3. building blocks: allows components and whole systems to be composed of many levels of simpler components.

These CAS properties and mechanisms provide a useful framework for designing agent-based models.

Later, Holland (1992) also proposed genetic algorithms as a basic mechanism for adaptation in nature. These are generic search procedures based on the mechanics of genetics and natural selection and the bases for many "swarm optimization" algorithms (Bonabeau 1999).

ABMS also borrows from the fields of multi-agent systems (MAS), robotics and artificial intelligence, but ABMS is not only concerned with designing and understanding "artificial" agents. Agent-based modeling is primarily concerned with modeling human social behavior and individual decision-making. With this comes the need to represent social interaction, group behavior, collaboration and the emergence of higher order social structures. This is why specialized languages and toolkits are needed.

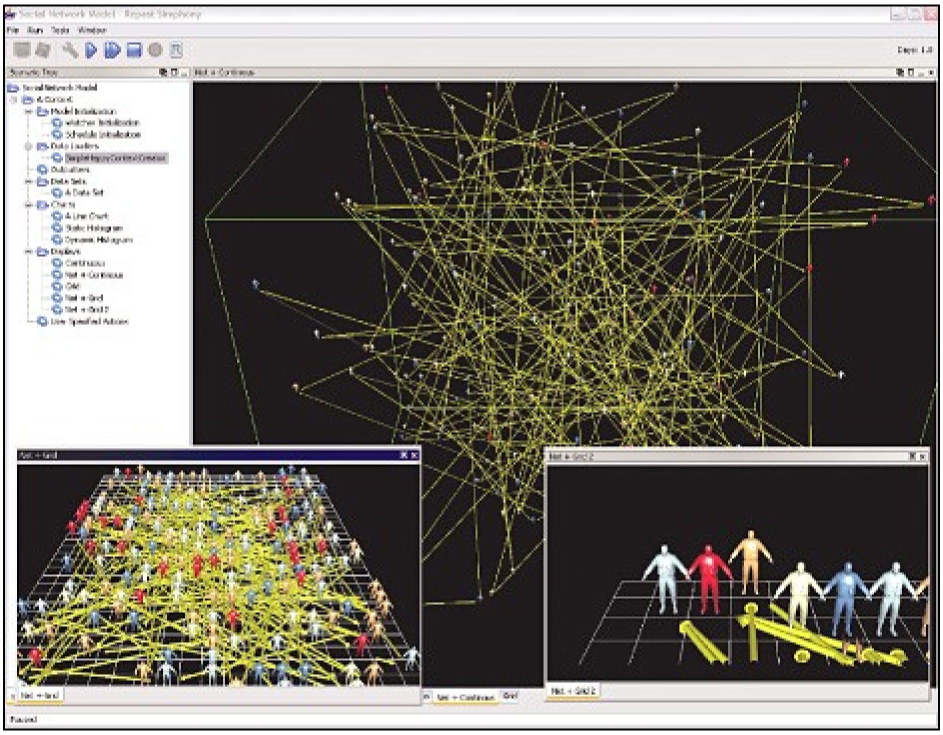

Figure 1: Autonomous agents interact over a self-organizing social network in this Repast simulation of social influence.
Image courtesy of Kurt Severance, NASA, and Kenneth Zick, U. Michigan

Swarm, Repast and ABMS Software

Swarm was the first widely used software toolkit for modeling complex adaptive systems. It was conceived by Chris Langton at the Santa Fe Institute and was designed specifically to explore "artificial life," an approach to studying biological systems that attempts to infer mechanism from biological phenomena using the elaboration, refinement and generalization of these mechanisms to identify unifying dynamical properties of biological systems. The first version of Swarm was available by 1996, and since then it has been used by researchers in not only biology, but also anthropology, computer science, defense, ecology, economics, geography, industry, political science and many other fields. Swarm is a free and open source software library (Minar, et al. 1996) and is currently maintained by the Swarm Development Group (SDG 2005).

Thanks to substantial public research and development investments, many more ABMS software environments are now freely available, each having a somewhat different focus. These environments support several features specific to agent modeling: a time scheduler, communications mechanisms, flexible interaction topologies, a range of architectural choices, facilities for storing and displaying agent states, large-scale development support and in some cases special topic support. These new environments maintain the Swarm tradition of being freely available and, in some cases, open source. They capitalize on the Swarm experience, making the most of modern principles of software design and implementation, emphasizing reusability and extensibility through the object-oriented software design model. Proprietary agent-based modeling toolkits are also becoming available.

One of the most widely used open source toolkits is Repast (the REcursive Porous Agent Simulation Toolkit), originally developed by Sallach, Collier and others (Collier et al., 2003) at the University of Chicago in 2000, and expanded by Argonne National Laboratory, as a reusable software infrastructure to support "rapid social science discovery" based on extensive computational experimentation (Sallach and Macal 2001). Recent releases of Repast have extended it to handle large-scale agent simulation application development. Repast is now managed by the non-profit volunteer Repast Organization for Architecture and Design (ROAD). ROAD is led by a board of directors that includes members from a wide range of government, academic and industrial organizations; Argonne National Laboratory and the University of Chicago remain especially active in maintaining and developing Repast. The Repast system, including the source code, is available directly from the Web at http://repast.sourceforge.net/.

Repast is available in pure Java and pure Microsoft .NET forms. Users build simulations by incorporating Repast library components directly into their own Java programs or by using a higher-level visual scripting environment with a limited syntax not requiring users to understand Java coding that is easier for the uninitiated users to pick up. The unique features of Repast have made it particularly attractive to modelers. For example, Tobias and Hofmann performed a review of several agent modeling toolkits, including Repast, and found that "we can conclude with great certainty that according to the available information, Repast is at the moment the most suitable simulation framework for the applied modeling of social interventions based on theories and data" (Tobias and Hofmann 2004). The Repast community is large and growing. For example, Repast 3 has had 7,000 downloads versus 3,000 for the previous version. Repast has been used extensively in social simulation and related applications (North and Macal 2005).

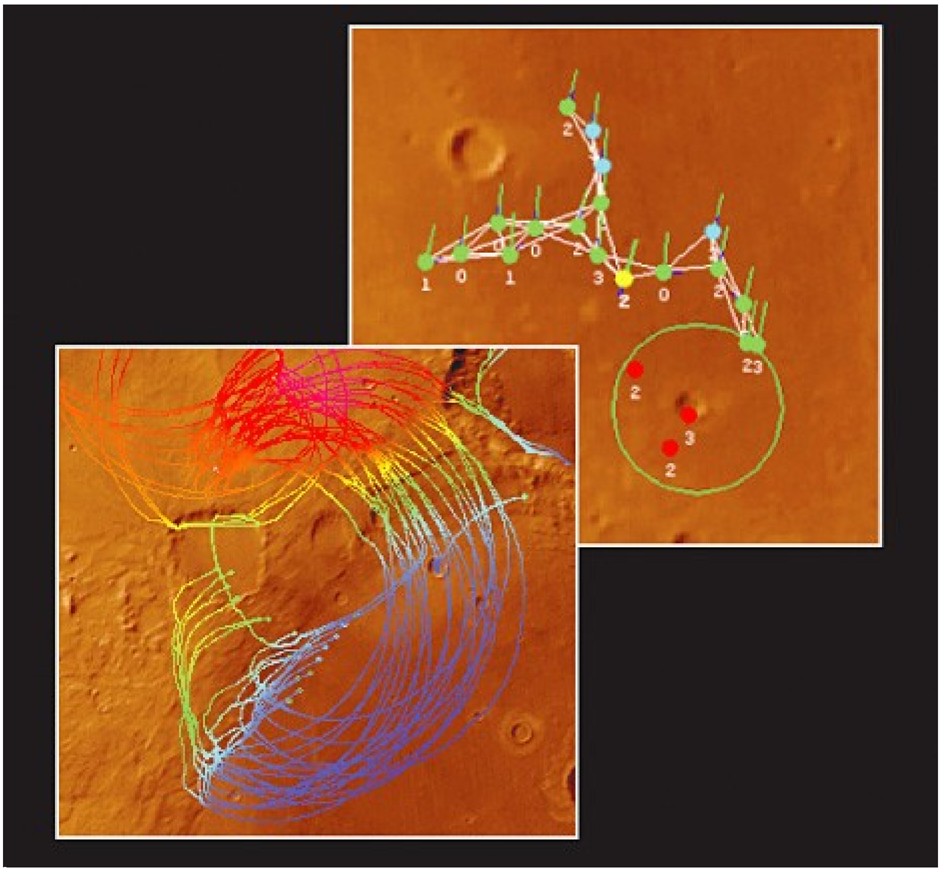

Figure 2: Swarms of autonomous rovers establish communication links and coordinate movements over the surface of a virtual Mars in this Repast simulation of self-organizing agent behavior.
Image courtesy of Kurt Severance, NASA, and Kenneth Zick, U. Michigan

Practical ABMS

Many of the tools and techniques for developing and applying agent-based models are coming into their own only recently (North and Macal, in press). Getting a good software toolkit or toolkits is a good first step in developing an agent-based model but is only part of the bigger process. A phased approach in developing an agent model is often the most workable, and often requires more than a single software approach. Many analysts recommend starting with NetLogo, which has excellent documentation, including tutorials, available as a free download. Others suggest starting with desktop tools such as general spreadsheets and computational mathematics systems such as MATLAB and Mathematica. Desktop systems require no compilation or linking steps, unlike general programming languages or agent-based modeling toolkits; however, desktop systems are generally limited to dozens to hundreds of agents and generally do not provide dedicated agent facilities.

Large-scale ABMS extends agent modeling beyond simple desktop environments and allows thousands to millions of agents to engage in sophisticated interchanges. Large-scale agent modeling is usually done with computer-based agent simulation environments and toolkits, such as Repast. Toolkits differ along various dimensions such as available features, the ease of learning and use, performance and licensing requirements.

Coming Attractions

The open source nature and architecture of Repast facilitates updates and upgrades. The respective development teams are exploring possible links between NetLogo and Repast. The Repast development team is also preparing to introduce a new version, named Repast Simphony, to make Repast easier to learn and use by non-computer specialists but also to make it useable at a higher level of abstraction. The goal is to make Repast more user-oriented and visually oriented and to make behavioral rules easier to enter into the model. This may, in turn, facilitate having a process description layer over the more simple components, enabling a broader class of users — analysts who are not necessarily experienced modelers and definitely not computer programmers — to use the toolkit. Thus, the leading open source ABMS toolkits' developers are trying to move in the direction conventional simulation took 10 to 15 years ago, replacing much coding with an intuitive visual interface and on-screen menus.

ABMS and Traditional M&S Techniques

Agent-based modeling can provide an overarching framework for model components based on other modeling techniques. For example, models may be composed of agents whose decision-making behaviors are represented by formal optimization problems or by informal decision heuristics. Another example is agent behaviors represented as statistical models deriving agent behaviors from the agents' input information. Agent-based modeling can also be used as a complement to other modeling techniques: for example, an agent model that builds system behavior from the behaviors of the individual agents can be "docked" (used in conjunction) with a more aggregate Systems Dynamics model of the system, to see whether the two approaches yield similar results over a range of test cases.

Conclusions

Agent-based modeling and simulation combines novel and traditional modeling approaches in ways that have not been previously possible. With the new software toolkits recently released, this approach is rapidly becoming accessible to a much larger set of analysts, with less programming experience needed. As problem specification becomes less tied to and constrained by the tools' capabilities and the user's knowledge of the tools, the field holds great promise for many interesting new developments and opportunities.

Douglas A. Samuelson *is president of InfoLogix, Inc., Annandale, Va., a senior* *analyst for the Homeland Security Institute, Arlington, Va., author of the "ORacle" column and a frequent feature writer for* OR/MS Today.

***Charles M. Macal** is a senior system engineer at Argonne National Laboratory, Argonne, Ill., and the director of the Center for Complex Adaptive Agent Systems Simulation (CAS*^{2}*), an applied research organization specializing in development and application of agent-based modeling techniques.*

References

1. Arthur, W.B., J.H. Holland, B. LeBaron, R. Palmer, and P. Taylor, 1997, "Asset pricing under endogenous expectations in an artificial stock market," in W.B. Arthur, S.N. Durlauf, and D.A. Lane (eds), "The Economy as an Evolving Complex System II," Addison-Wesley, pp. 15-44: Reading, Mass.
2. Axelrod, R., 1997, "The Complexity of Cooperation: Agent-Based Models of Competition and Collaboration," Princeton University Press, Princeton, N.J.
3. Bagni, R., R. Berchi and P. Cariello, 2002, "A comparison of simulation models applied to epidemics," *Journal of Artificial Societies and Social Simulation,* Vol. 5, No. 3, http://jasss.soc.surrey.ac.uk/5/3/5.html, June 30, 2002.
4. Bonabeau, E., M. Dorigo and G. Theraulaz, 1999, "Swarm intelligence: from natural to artificial systems," Oxford: Oxford University Press.
5. Bonabeau, E., 2001, "Agent-based modeling: methods and techniques for simulating human systems," in *Proc. National Academy of Sciences,* Vol. 99, No. 3, pp. 7,280-7,287.
6. Collier, N., T. Howe, et al., 2003, "Onward and upward: the transition to Repast 2.0," in *Proc. First Annual North American Association for Computational Social and Organizational Science Conference,* Pittsburgh, Pa.
7. Emonet, T., C. M. Macal, M. J. North, C. E. Wickersham and P. Cluzel, 2005, "AgentCell: A Digital Single-Cell Assay for Bacterial Chemotaxis," *Bioinformatics,* Vol. 21, No. 11, pp. 2,714-2,721.
8. Epstein, J. M. and R. Axtell, 1996, "Growing artificial societies: social science from the bottom up," Cambridge, Mass.: MIT Press.
9. Gilbert, N., and Troitzsch, K., 1999, "Simulation for the Social Scientist," Open University Press, second edition, 2005.
10. Holland, J. H., 1992, "Genetic Algorithms," *Scientific American,* Vol. 267, pp. 66-72.
11. Holland, J. H., 1995, "Hidden Order: How Adaptation Builds Complexity," Addison-Wesley: Reading, Mass.
12. Kohler, T. A., G. J. Gumerman and R. G. Reynolds, 2005, "Simulating ancient societies," *Scientific American,* July.
13. Macal, C., 2003, "Effects of global information availability in networks of supply chain agents," in *Proc. Agent 2003: Conf. on Challenges in Social Simulation,* Eds., D. Sallach, C. Macal

and M. North, Chicago, Oct. 2-4, pp. 235-252, Argonne National Laboratory.

14. Macal, C., 2004, "Emergent structures from trust relationships in supply chains," in *Proc. Agent 2004: Conf. on Social Dynamics: Interaction, Reflexivity and Emergence,* Eds., C. Macal, D. Sallach and M. North, Chicago, Oct. 7-9, pp. 743-760, Argonne National Laboratory, MASON, 2005, http://cs.gmu.edu/~eclab/projects/mason/.
15. Minar, N., R. Burkhart, et al., 1996, "The Swarm simulation system, a toolkit for building multi-agent simulations, www.santafe.edu/projects/swarm/overview/overview.html.
16. NetLogo, 2006, http://ccl.northwestern.edu/netlogo.
17. North, M. J., and C. M. Macal, (in press), "Managing business complexity: discovering strategic solutions with agent-based modeling and simulation," Oxford: Oxford University Press.
18. North, M. J. and C. M. Macal, 2005, "Escaping the accidents of history: an overview of artificial life modeling with Repast," in *Artificial Life Models in Software,* Eds., A. Adamatzky and M. Komosinski, Springer-Verlag: Dordrecht, Netherlands.
19. Pacific Asian Association for Agent-based Approach in Social Systems Sciences (PAAA), First World Conference on Social Simulation, www.paaa.econ.kyoto-u.ac.jp/wcss06/wcss06.html.
20. Repast, 2006, http://repast.sourceforge.net.
21. Sallach, D. and C. Macal, 2001, "The simulation of social agents: an introduction," *Special Issue of Social Science Computer Review,* Vol. 19, No. 3, pp. 245—248.
22. Samuelson, D., 2000, "Designing Organizations," *OR/MS Today,* December 2000.
23. Samuelson, D., 2005, "Agents of Change," *OR/MS Today,* February 2005.
24. SDG (Swarm Development Group), 2005, www.swarm.org.
25. Tobias, Robert and Carole Hofmann, 2004, "Evaluation of free Java-libraries for social-scientific agent based simulation," *Journal of Artificial Societies and Social Simulation,* Vol. 7, No. 1, Jan. 31.