Proceedings of the 2011 Winter Simulation Conference
S. Jain, R.R. Creasey, J. Himmelspach, K.P. White, and M. Fu, eds.

INTRODUCTORY TUTORIAL: AGENT-BASED MODELING AND SIMULATION

Charles M. Macal
Michael J. North

Argonne National Laboratory
Center for Complex Adaptive Systems Simulation (CAS^2)
Decision & Information Sciences Division
Argonne, IL 60439 USA

ABSTRACT

Agent-based modeling and simulation (ABMS) is an approach to modeling systems comprised of individual, autonomous, interacting "agents." There is much interest in many application problem domains in developing agent-based models. Agent-based modeling offers ways to model individual behaviors and how behaviors affect others in ways that have not been available before. Applications range from modeling agent behavior in supply chains and the stock market, to predicting the success of marketing campaigns and the spread of epidemics, to projecting the future needs of the healthcare system. Progress in the area suggests that ABMS promises to have far-reaching effects on the way that businesses use computers to support decision-making and researchers use agent-based models as electronic laboratories to aid in discovery. This brief tutorial introduces agent-based modeling by describing the basic ideas of ABMS, discussing some applications, and addressing methods for developing agent-based models.

1 INTRODUCTION

Agent-based modeling and simulation (ABMS) is a modeling approach that has gained increasing attention over the past 15 years or so. This growth trend is evidenced by the increasing numbers of applications (Table 1), articles appearing in modeling and applications journals, funded programs that call for agent-based models incorporating elements of human and social behavior, the growing number of conferences on or that have tracks dedicated to agent-based modeling, the demand for ABMS courses and instructional programs, and the number of presentations at conferences such as the WSC that reference agent-based modeling. Some contend that ABMS "is a third way of doing science" and could augment traditional deductive and inductive reasoning as discovery methods (Axelrod 1997). This tutorial provides an introduction to agent-based modeling and simulation. The goals are to show that ABMS is:

- Useful: Why ABMS is an appropriate modeling approach for a large class of problems and has advantages over conventional modeling approaches in many cases,
- Usable: How ABMS is advancing to the point of producing portable, extensible, and transferable software, with better integrated development environments, and
- Used: How good ABMS applications are being developed to solve practical problems.

Table 1: A sample of recent agent-based applications available on the web (all application use the Repast agent-based modeling toolkit)

Application Area:	Model Description:
Agriculture	A spatial individual-based model prototype for assessing potential exposure of farm-workers conducting small-scale agricultural production (Leyk, Binder, and Nuckols 2009).
Air Traffic Control	Agent-based model of air traffic control to analyze control policies and performance of an air traffic management facility (Conway 2006)
Anthropology	Agent-based model of prehistoric settlement patterns and political consolidation in the Lake Titicaca basin of Peru and Bolivia (Griffin and Stanish 2007)
Biomedical Research	*The Basic Immune Simulator*, an agent-based model to study the interactions between innate and adaptive immunity (Folcik, An, and Orosz 2007)
Crime Analysis	Agent-based model that uses a realistic virtual urban environment, populated with virtual burglar agents (Malleson 2010).
Ecology	Agent-based model to investigate the trade-off between road avoidance and salt pool spatial memory in the movement behavior of moose in the Laurentides Wildlife Reserve (Grosman et al. 2011). Agent-based model of predator-prey relationships between transient killer whales and other marine mammals (Mock and Testa 2007). A risk-based approach for analyzing the intentional introduction of non-native oysters on the US east coast (Opaluch, Anderson, and Schnier 2005).
Energy Analysis	Agent-based model to identify potential interventions for the uptake of wood-pellet heating in Norway (Sopha et al. 2011). Agent-based model for scenario development of offshore wind energy (Mast et al. 2007).
Epidemiology	Synthetic age-specific contact matrices are computed through simulation of a simple individual-based model (Iozzi et al. 2010).
Evacuation	A simulation of tsunami evacuation using a modified form of Helbing's social-force model applied to agents (Puckett 2009).
Market Analysis	A large-scale agent-based model for consumer marketing developed in collaboration with a Fortune 50 firm (North et al. 2009). An illustrative agent-based model of a consumer airline market to derive market share for the upcoming year (Kuhn et al. 2010). Agent-based simulation that models the possibilities for a future market in sub-orbital space tourism (Charania et al. 2006).
Organizational Decision Making	An agent-based model to allow managers to simulate employee knowledge-sharing behaviors (Wang et al. 2009). An agent-based model to evaluate the dynamic behavior of a global enterprise, considering system-level performance as well as components' behaviors (Behdani et al. 2009). Agent based modeling approach to allow negotiations in order to achieve a global objective, specifically for planning the location of intermodal freight hubs (van Dam et al. 2007).
Social Networks	An agent-based model of email-based social networks, in which individuals establish, maintain and allow atrophy of links through contact-lists and emails (Menges, Mishra, and Narzisi 2008).

Agent-based modeling is being applied to many areas, spanning human social, physical and biological systems. Applications range from modeling ancient civilizations that have been gone for hundreds of years, to designing new markets for products that do not exist right now. Heath, Ciarallo, and Hill (2009) provide a review of agent-based modeling applications. Selected applications that use the Repast agent-based modeling toolkit are listed in Table 1. All of the cited publications make the case for agent-based modeling as the preferred modeling approach versus other modeling techniques for the problem addressed. They argue that agent-based modeling is used because only agent-based models can explicitly incorporate the complexity arising from individual behaviors and interactions that exist in the real-world.

We refer the reader to previous papers on other introductory topics in ABMS not covered here, such as the history of ABMS, connections of ABMS to complex adaptive systems and artificial life, and the relationships of ABMS to other modeling and simulation techniques (Macal and North 2010, Macal and North 2009, Macal 2009, North and Macal 2007, Macal and North 2007). Publications that are especially good introductions to agent-based modeling include Epstein and Axtell (1996), Gilbert and Troitzsch (2005), and Bonabeau (2001), as well as the ACE web site (Tesfatsion 2011).

This tutorial is organized into two parts. The first part is a tutorial on how to *think* about ABMS. The background on ABMS and its motivating principles are described along with some exemplary applications. The second part is a tutorial on how to *do* ABMS. It addresses modeling approaches and toolkits for developing agent-based models.

2 HOW TO THINK ABOUT AGENT-BASED MODELING

2.1 Structure of an agent-based model

A typical agent-based model has three elements:

1. Ag*ents*, their attributes and behaviors.
2. A*gent relationships* and methods of interaction. An underlying topology of connectedness defines how and with whom agents interact.
3. *Agents' environment*. Agents live in and interact with their environment in addition to other agents.

A model developer must identify, model, and program these elements to create an agent-based model. The structure of a typical agent-based model is shown in Figure 1.

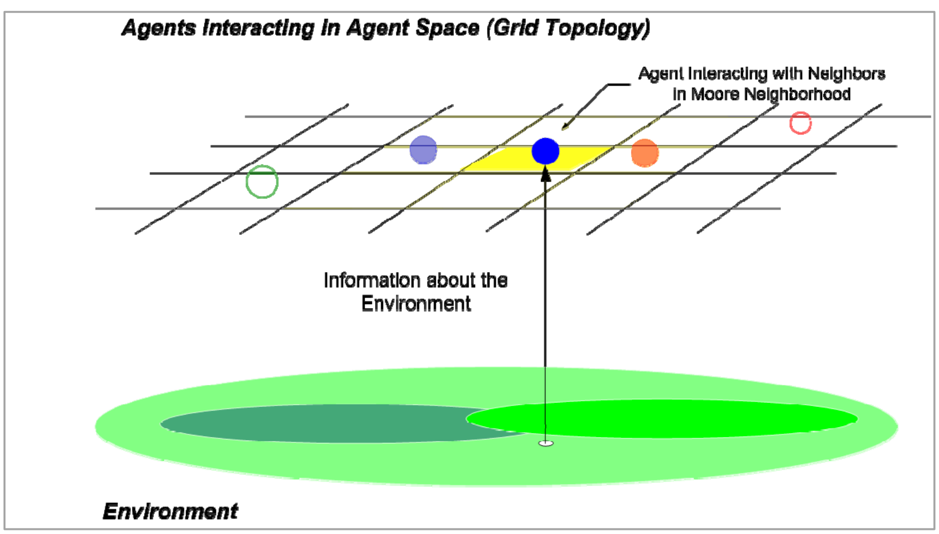

Figure 1: General elements of an agent-based model

A computational engine for simulating agent behaviors and agent interactions is then needed to make the model run. An agent-based modeling toolkit, programming language or other implementation provides this capability. To run an agent-based model is to have agents repeatedly execute their behaviors and interactions. This process often does, but is not necessarily modeled to, operate over a timeline, as in time-stepped, activity-based, or discrete-event simulation structures.

2.2 Agents

There is no universal agreement on the precise definition of the term *agent* in the context of ABMS. It is the subject of much discussion and occasional debate. The issue is more than an academic one, as it often surfaces when one makes a claim that their model is *agent-based* or when one is trying to discern whether such claims made by others have validity. There are important implications of the term agent-based when used to describe a model in terms of the model's capabilities or potential capabilities that could be attained through relatively minor modification. A formal definition of agent is beyond the scope of this paper; in the literature informal descriptions of agent tend to agree on more points than they disagree. Some modelers consider any type of independent component, whether it be a software component or a model to be an agent (Bonabeau 2001). Some authors insist that a component's behavior must also be adaptive in order for it to be considered an agent. Casti (1997) argues that agents should contain both base-level rules for behavior as well as a higher-level set of "rules to change the rules." The base-level rules provide responses to the environment, while the rules-to-change-the-rules provide adaptation. Jennings (2000) provides a computer science view of agent that emphasizes the essential characteristic of autonomous behavior.

For practical modeling purposes, we consider agents to have certain properties and attributes, as follows (Figure 2).

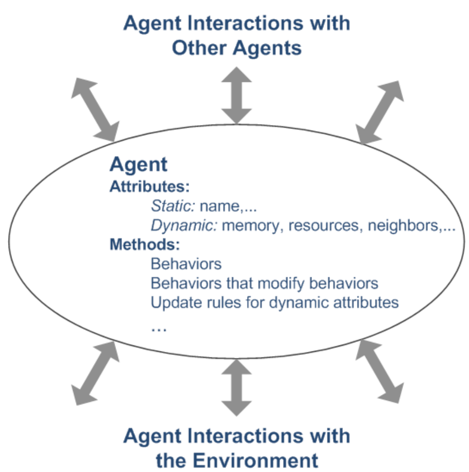

Figure 2: A Typical Agent

Autonomy. An agent is autonomous and self-directed. An agent can function independently in its environment and in its interactions with other agents, generally from a limited range of situations that are of interest and that arise in the model. When we refer to an agent's *behavior*, we refer to a general process that links the information the agent senses from its environment and interactions to its decisions and actions.

Modularity. Agents are modular or self-contained. An agent is an identifiable, discrete individual with a set of characteristics or attributes, behaviors, and decision-making capability. The modularity requirement implies that an agent has a boundary, and one can easily determine whether something (that is, an element of the model's state) is part of an agent, is not part of an agent, or is a characteristic shared among agents.

Sociality. An agent is social, interacting with other agents. Common agent interaction protocols include contention for space and collision avoidance, agent recognition, communication and information exchange, influence, and other domain-or application-specific mechanisms.

Conditionality. An agent has a *state* that varies over time. Just as a system has a state consisting of the collection of its state variables, an agent also has a state that represents its condition, the essential variables associated with its current situation. An agent's state consists of a set or subset of its attributes. The state of an agent-based model is the collective states of all the agents along with the state of the environment. An agent's behaviors are conditioned on its state. As such, the richer the set of an agent's possible states, the richer the set of behaviors that an agent can have.

Agents often have additional properties, which may or may not be considered as defining properties or necessary for agency. An agent may have explicit *goals* that drive its behavior. The goals are not necessarily objectives to maximize as much as criteria against which to assess the effectiveness of its decision and actions. An agent may have the ability to *learn and adapt* its behaviors based on its experiences. Individual learning and adaptation requires an agent to have memory, usually in the form of a dynamically updated attribute of the agent. Interesting and useful agent models are developed regularly that do not contain adaptive agents. Another optional feature concerns agent heterogeneity. Interesting and useful agent models are developed regularly that contain nearly identical agents.

2.3 Agent Relationships

Agent-based modeling concerns itself with modeling agent relationships and agent interactions as much as it does modeling agents and agent behaviors. The primary issues of modeling agent interactions are specifying who is, or could be, connected to who, and the dynamics governing the mechanisms of the interactions. For example, an agent-based model of Internet growth would include mechanisms that specify who connects to who, why, and when.

Common topologies for representing social agent interactions are as follows:

1. Soup. A nonspatial model in which agents have no locational attribute.
2. Grid or lattice. Cellular automata represent agent interaction patterns and available local information by a grid or lattice; cells immediately surrounding an agent are its neighborhood.
3. Euclidean space. Agents roam in 2D or 3D spaces.
4. Geographic Information System (GIS). Agents move over realistic geo-spatial landscapes.
5. Networks. Networks may be static (links pre-specified) or dynamic (links determined endogenously).

No matter what agent-interaction topology is used in an agent-based model to connect the agents, the essential idea is that agents only interact at any given time with a limited (small) number of other agents out of all the agents in the population. This notion is implemented by defining a local neighborhood and thereby limiting interaction to a small number of agents that happen to be in that neighborhood. This is not to say that agents need to be located in close proximity to one another spatially to be able to interact. The network topology allows agents to be linked on the basis of relationships other than proximity. Most empirically-based social networks have characteristic network topologies with individual agents having limited numbers of connections.

2.4 Emergence in Agent-based Models

One of the motivating concepts for agent-based modeling is its ability to capture *emergence*. Agent-based models that are completely described by simple, deterministic rules and based only on local information can produce sustainable patterns that self-organize themselves and have not been explicitly programmed into the models. Emergence refers to the emergence of *order*. Emergence can be illustrated by simple agent-based models such as *Life* and *Boids*. These observations have practical implications for developing and interpreting agent-based models. More complex models of the kind that people are likely to build to represent real-world phenomenon can also exhibit emergent behavior resulting from agent interactions.

Based on simple rules of behavior and the nature of agent interactions, natural systems seemingly exhibit collective intelligence, or *swarm intelligence*, even without the existence of or the direction provided by a central authority. Swarm intelligence has inspired agent-based modeling as well as practical optimization techniques, such as ant colony optimization and particle swarm optimization that have been used to solve practical scheduling and routing problems (Bonabeau, Dorigo, and Theraulaz 1999). These types of algorithms can be implemented in agent-based frameworks.

3 HOW TO DO AGENT-BASED MODELING

3.1 Thinking Through an Agent Model

It is useful to ask a series of agent-specific questions before developing an agent-based model (Table 2). The answers to these questions help define the scope and level of detail for which it is appropriate to model the system. They imply the resources required for successfully completing the project as well as help identifying likely bottlenecks to development.

Table 2: Questions to Ask Before Developing an Agent-based Model

Model Purpose and Value-added of Agent-based Modeling: What specific problem is the model being developed to address? What specific questions should the model answer? What kind of information should the model provide to help make or support a decision? Why might agent-based modeling be a desirable approach? What value-added does agent-based modeling bring to the problem that other modeling approaches cannot bring?
All About Agents: What should the agents be in the model? Who are the decision makers in the system? What are the entities that have behaviors? Where might the data come from, especially on agent behaviors, for such a model?
Agent Data: What data on agents is simply descriptive (static attributes)? What agent attributes are calculated endogenously by the model and updated for the agents (dynamic attributes)? What is the agents' environment? How do the agents interact with the environment? Is agent mobility through space an important consideration?
Agent Behaviors: What agent behaviors are of interest? What decisions do the agents make and what information is required to make such decisions? What behaviors are being acted upon? What actions are being taken by the agents? How would we represent the agent behaviors? By If-Then rules? By adaptive probabilities, such as in reinforcement learning? By regression models or neural networks?
Agent Interactions: How do the agents interact with each other? How do the agents interact with the environment? How expansive or focused are agent interactions?
Agent Recap: How do we design a set of experiments to explore the importance of uncertain behaviors, data and parameters? How might we validate the model, especially the agent behaviors and the agent interaction mechanisms?

Developing an agent-based model requires getting behavioral theory and modeling frameworks for the agents included in the model, getting a process for developing the agent model, and getting a platform for building the model (see figure 3).

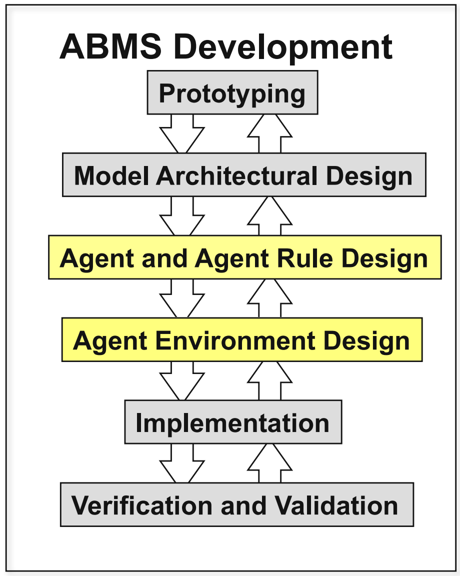

Figure 3: Agent-based Model Development Process

3.2 Designing Agent-based Models

Modern software practices are based on a template design approach in which recurring elements are codified and reused for new applications; this approach has proven very valuable in designing models as well as software. Several formats have been proposed for describing agent-based model designs. Chief among these standards is Grimm et al.'s (2006) Overview, Design Concepts, and Details (ODD) protocol. ODD describes models using a three-part approach: overview, concepts, and details. The model overview includes a statement of the model's intent, a description of the main variables, and a discussion of the agent activities and timing. The design concepts include a discussion of the foundations of the model, and the details include the initial setup configuration, input value definitions, and descriptions of any embedded models (Grimm et al. 2006).

North and Macal (2011) discuss product design patterns for agent-based modeling. For example, design patterns that have proven themselves useful for agent-based modeling include:

- *Scheduler Scramble:* The problem addressed is when two or more agents from the ABM pattern can schedule events that occur during the same clock tick. Getting to execute first may be an advantage or disadvantage. How do you allow multiple agents to act during the same clock tick without giving a long-term advantage to any one agent?
- *Context and Projection Hierarchy:* The problem addressed is how to organize complex spaces into a single unified form such that individual agents can simultaneously exist in multiple spaces and the spaces themselves can be seamlessly removed and added.
- *Strategy*: The problem addressed is how to let clients invoke rules that may be defined long after the clients are implemented? There are a set of rules that need to be dynamically selected while a program is running. There is a need to separate rule creation from rule activation.

- *Learning*: The problem addressed is to how to model agents that adapt or learn. There is a need for agents to change their behavior over time based on their experiences.

An agent-based model developer uses design patterns in the following way. The developer decides on how to model important features of a real-world system, such as, for example, how time should be treated in the model. The real world process moves forward in continuous time, but the modeler may decide that for the given application, important events only occur at discrete time points that can be scheduled to occur in advance at constant time intervals (as in time-stepped simulation), according to assumed distributions for activity times (as in discrete event simulation), or according to thresholds of agent state variables being reached (as in combined continuous/discrete event simulation). If, for example, the time-stepped approach is taken, other technical questions must be answered about how to deal with simultaneous events. The Scheduler Scramble pattern above is a possible solution.

3.3 Modeling Agent Systems

Identifying agents, accurately specifying their behaviors, and appropriately representing agent interactions are the keys to developing useful agent models. One begins developing an agent-based model by identifying the agent types (classes) along with their attributes. Agents are generally the decision-makers in a system whether they be human, organizational, or automated. Once the agents are defined, agent behaviors are specified. One needs to have a theory of agent behavior as a basis for modeling agent behavior. For example, a normative model in which agents attempt to optimize a well-defined objective can be a useful starting point to eventually developing more descriptive and domain-specific behavioral heuristics. Alternatively, one may begin with a bounded rationality model or a generic behavioral heuristic, such as anchoring and adjustment, to describe agent behavior or more broadly a formal behavioral modeling framework such as the Belief-Desire-Intent (BDI) model (Wooldridge 2009).

In addition to agents, an agent-based model consists of agent relationships. One defines the agent relationships and then adds the methods that control which agents interact, when they interact, and how they interact.

3.4 Advanced Agent-based Modeling

Often, an agent-based modeler would like to include a variety of advanced capabilities in their model. These capabilities include distributed computing implementations, artificial intelligence and machine learning algorithms, geographical information systems (GIS), connections to relational databases, version control systems (especially if there are multiple developers working on a project), and integrated development environments (IDEs). It is often useful to first develop a core model that includes these capabilities as connections or "stubs" to ensure the core model design is acceptable and to verify that scaling up the design appears feasible. Agent-based modeling and software toolkits often provide advanced capabilities such as these.

3.5 ABMS Software and Toolkits

Agent-based modeling can be done using general, all-purpose software or programming languages, or it can be done using specially designed software and toolkits that address the specific requirements for modeling agents. Agent modeling can be done in the small, on the desktop, or in the large, using large-scale computing clusters, or it can be done at any scale in-between. Projects often begin small, using one of the desktop ABMS tools, or whatever tool or programming language the developers are familiar with. The initial prototype then grows in stages into a larger-scale agent-based model, often using dedicated ABMS toolkits. Often one begins developing their first agent model using the approach that one is most familiar with, or the approach that one finds easiest to learn given their background and experience.

We distinguish several approaches to building ABMS applications in terms of the scale of the software that one can apply according to the following continuum:

Desktop Computing for ABMS Application Development:

- Spreadsheets: Excel using the macro programming language VBA
- Dedicated Agent-based Prototyping Environments: Repast Simphony, NetLogo
- General Computational Mathematics Systems: MATLAB, Mathematica

Large-Scale (Scalable) Agent Development Environments:

- Repast
- Swarm
- MASON
- AnyLogic
- Other

General Programming Languages:

- C++
- Java
- Python

Desktop ABMS can be used to learn agent modeling, prototype basic agent behaviors, and perform limited analyses. Desktop agent-based models can be simple, designed and developed in a period of a few days by a single computer-literate modeler using tools learned in a few days or weeks. Desktop agent modeling can be used to explore the potential of ABMS with relatively minor time and training investments, especially if one is already familiar with the tool.

Spreadsheets, such as Microsoft Excel, are in many ways the simplest approach to modeling. It is easier to develop models with spreadsheets than with many of the other tools, but the resulting models generally allow limited agent diversity, restrict agent behaviors, and have poor scalability compared to the other approaches designed specifically for agent modeling. Agent-based modeling in spreadsheets requires some macro-programming to be done in a language such as VBA (Visual Basic for Applications), the macro programming language for Excel and other Microsoft Office applications. Significant agent models have been developed entirely using spreadsheets. In previous WSC papers, we described an spreadsheet implementation of a spatial agent-based shopper model (Macal and North 2007).

Special-purpose agent tools, such as NetLogo, provide special facilities focused on agent modeling. NetLogo is a free ABMS environment developed at Northwestern University's Center for Connected Learning and Computer-Based Modeling (Wilensky 1999). The most directly visible common trait shared by the various prototyping environments is that they are designed to get first-time users started as quickly as possible. The NetLogo language uses a modified version of the Logo programming language (Harvey 1997). NetLogo was originally developed to support teaching, but it can be used to develop a wide range of applications. NetLogo provides a graphical environment to create programs that control graphic "turtles" that reside in a world of "patches" that is monitored by an "observer." NetLogo continues as a subject of active development and new versions with expanded capabilities are released periodically.

General-purpose desktop computational mathematics system (CMS) with integrated development environments (IDEs), such as MATLAB and *Mathematica*, can be used to develop agent models, although the agent-specific functionality has to be written by the developer from scratch, as there are no dedicated libraries or modules that focus on agent-based modeling. The basic requirements is knowledge of how to program in a scripting language. CMS environments have rich mathematical functions, and nearly any mathematical relation or function that can be numerically calculated is available within these tools or their add-on libraries. In some cases, the tools even support symbolic processing and manipulation, which is useful for systems of equations that can be solved analytically and can be exploited quite effectively to do agent-based modeling (Macal 2004). If a CMS environment is already familiar to a developer, this can be a good place to start agent-based modeling.

Many large-scale ABMS software environments are now freely available. These include Repast (North, Collier and Vos, 2006), Swarm (SDG 2006; Minar et al. 1996), NetLogo (NetLogo 2009) and MASON (GMU 2006) among many others. Proprietary toolkits are also available such as AnyLogic (XJ Technologies 2009). A review and comparison of Java-based agent modeling toolkits is provided by Tobias and Hoffman (2004) and Nikolai and Madey (2009).

Swarm was the first ABMS software development environment, launched in 1994 at the Santa Fe Institute. Swarm was originally written in Objective C and was later fitted with a Java interface.

Following the original Swarm innovation, the Repast (REcursive Porous Agent Simulation Toolkit) toolkit was developed as a pure Java implementation (North, Collier, and Vos 2006). Repast Simphony (Repast S) is the latest version of Repast, designed to provide visual point-and-click tools for agent model design, agent behavior specification, model execution, and results examination. The Repast S agent model designer allows users to visually specify the logical structure of their models, the spatial (e.g., geographic maps and networks) structure of their models, the kinds of agents in their models, and the behaviors of the agents themselves. Once their models are specified, users can use the point-and-click Repast S runtime environment to execute model runs as well as visualize and store results. The Repast S runtime environment includes automated results analysis and connections to a variety of spreadsheet, visualization, data mining, and statistical analysis tools, virtually all of which are free and open source. Repast Simphony 2.0 also includes a ReLogo, a new Logo-like interface for specifying agent models (Ozik 2011).

As computational capabilities continue to advance in both hardware and software, new capabilities are continuously being incorporated into the latest versions of ABMS toolkits. The field is advancing rapidly toward highly scalable, high productivity agent development environments.

4 WHY AND WHEN ABMS

We conclude by offering some ideas on the situations for which agent-based modeling can offer distinct advantages to conventional simulation approaches such as discrete event simulation (Law 2007), systems dynamics (Sterman 2000) and other quantitative modeling techniques. Axtell (2000) discusses several reasons for agent-based modeling especially compared to traditional approaches to modeling economic systems. When is it beneficial to think in terms of agents? When one or more of the following criteria are satisfied:

- When the problem has a natural representation as being comprised of agents
- When there are decisions and behaviors that can be well-defined
- When it is important that agents have behaviors that reflect how individuals actually behave (if known)
- When it is important that agents adapt and change their behaviors
- When it is important that agents learn and engage in dynamic strategic interactions
- When it is important that agents have dynamic relationships with other agents, and agent relationships form, change, and decay
- When it is important to model the processes by which agents form organizations, and adaptation and learning are important at the organization level
- When it is important that agents have a spatial component to their behaviors and interactions
- When the past is no predictor of the future because the processes of growth and change are dynamic
- When scaling-up to arbitrary levels is important in terms of the number of agents, agent interactions and agent states
- When process structural change needs to be an endogenous result of the model, rather than an input to the model

ACKNOWLEDGMENTS

This work was supported by the U.S. Department of Energy under contract number DE-AC02-06CH11357. Portions of this tutorial have appeared in previous tutorial papers presented at the Winter Simulation Conference for 2005 - 2009.

REFERENCES

Axelrod, R. 1997. *The Complexity of Cooperation: Agent-Based Models of Competition and Collaboration*. Princeton, NJ: Princeton University Press.

Axtell, R. 2000. *Why Agents? On The Varied Motivations for Agent Computing in the Social Sciences*, Working Paper 17, Center on Social and Economic Dynamics, Brookings Institution, Washington, D.C.

Behdani, B., Z. Lukszo, A. Adhitya, and R. Srinivasan. 2009. "Agent-Based Modeling To Support Operations Management In A Multi-Plant Enterprise." In *International Conference on Networking, Sensing and Control, ICNSC '09*, 323 – 328, Okayama, 10.1109/ICNSC.2009.4919295. Accessed August 12, 2011. www.nextgenerationinfrastructures.eu/download.php?field=document&itemID=515584.

Bonabeau, E. 2001. "Agent-Based Modeling: Methods and Techniques for Simulating Human Systems." In *Proc. National Academy of Sciences* 99(3): 7280-7287.

Bonabeau, E., M. Dorigo, and G. Theraulaz. 1999. *Swarm Intelligence: From Natural To Artificial Systems*. Oxford: Oxford University Press.

Casti, J. 1997. *Would-Be Worlds: How Simulation Is Changing the World of Science*. New York: Wiley.

Charania, A. C., J. R. Olds and D. DePasquale. 2006. "Sub-Orbital Space Tourism Market: Predictions of the Future Marketplace Using Agent-Based Modeling." SpaceWorks Engineering, Inc., Atlanta, GA. Accessed August 12, 2011. www.sei.aero/uploads/archive/IAC-06-E3.4.pdf.

Conway, S. R. 2006. "An Agent-Based Model for Analyzing Control Policies and the Dynamic Service-Time Performance of a Capacity-Constrained Air Traffic Management Facility." In *Proceedings of ICAS 2006 - 25th Congress of the International Council of the Aeronautical Sciences*. Hamburg, Germany, 3-8 Sep. 2006. Accessed August 12, 2011. http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20060048296_2006250468.pdf.

Epstein, J. M., and R. Axtell. 1996. *Growing Artificial Societies: Social Science From The Bottom Up*. Cambridge, MA: MIT Press.

Folcik, V. A., G. C. An, and C. G. Orosz. 2007. "The Basic Immune Simulator: An Agent-Based Model to Study the Interactions between Innate and Adaptive Immunity." *Theoretical Biology and Medical Modelling*, 4(39). Accessed August 12, 2011. www.tbiomed.com/content/4/1/39.

Gilbert, N., and K. Troitzsch. 2005. *Simulation for the Social Scientist*. McGraw-Hill. 2nd ed.

GMU (George Mason University). 2009. MASON Home Page. Accessed August 12, 2011. http://cs.gmu.edu/~eclab/projects/mason/.

Griffin, A. F., and C. Stanish. 2007. "An Agent-Based Model of Prehistoric Settlement Patterns and Political Consolidation in the Lake Titicaca Basin of Peru and Bolivia, Structure and Dynamics." *eJournal of Anthropological and Related Sciences*, 2(2). Accessed August 12, 2011. http://repositories.cdlib.org/imbs/socdyn/sdeas/vol2/iss2/art2.

Grimm, V., U. Berger, F. Bastiansen, S. Eliassen, V. Ginot, J. Giske, J. Goss-Custard, T. Grand, S. K. Heinz, G. Huse, A. Huth, J. U. Jepsen, C. Jørgensen, W. M. Mooij, B. Müller, G. Pe'er, C. Piou, S. F. Railsback, A. M. Robbins, M. M. Robbins, E. Rossmanith, N. Rüger, E. Strand, S. Souissi, R. A. Stillman, R. Vabø, U. Visser, and D. L. DeAngelis. 2006. "A Standard Protocol for Describing Individual-Based and Agent-Based Models." *Ecological Modelling* 198(1-2):115-126.

Grosman, P. D., J. A. G. Jaeger, P. M. Biron, C. Dussault, and J.-P. Ouellet. 2011. "Trade-off Between Road Avoidance and Attraction by Roadside Salt Pools in Moose: An Agent-Based Model to Assess Measures for Reducing Moose-Vehicle Collisions." *Ecological Modelling* 222(8):1423-1435.

Harvey, B. 1997. *Computer Science Logo Style*. Boston: MIT Press.

Heath, B. L., F. Ciarallo, and R. R. Hill. 2009. "A Survey Of Agent-Based Modeling Practices (January 1998 to July 2008)." *Journal of Artificial Societies and Social Simulation* 12(4) October.

Iozzi F., F. Trusiano, M. Chinazzi, F. C. Billari, E. Zagheni, S. Merler, M. Ajelli, E. Del Fava, and P. Manfredi. 2010. "Little Italy: An Agent-Based Approach to the Estimation of Contact Patterns - Fitting Predicted Matrices to Serological Data." *PLoS Comput Biol* 6(12): e1001021. doi:10.1371/journal.pcbi.1001021. Accessed August 12, 2011. http://openaire.cern.ch/record/773/files/journal.pcbi.1001021.pdf.

Jennings, N. R. 2000. "On Agent-Based Software Engineering." *Artificial Intelligence* 117:277-296.

Kuhn Jr, J. R., J. F. Courtney, B. Morris, and E. R. Tatara. 2010. "Agent-Based Analysis And Simulation of the Consumer Airline Market Share for Frontier Airlines." *Knowledge-Based Systems* 23(8): 875-882.

Law, A. M. 2007. *Simulation Modeling and Analysis*. 4th ed. New York: McGraw-Hill.

Leyk, S., C. R. Binder, and J. R. Nuckols. 2009. "Spatial Modeling Of Personalized Exposure Dynamics: The Case Of Pesticide Use In Small-Scale Agricultural Production Landscapes Of The Developing World." *International Journal of Health Geographics*, 8:17. doi:10.1186/1476-072X-8-17. Accessed August 12, 2011. http://www.ij-healthgeographics.com/content/pdf/1476-072X-8-17.pdf.

Macal, C. M. 2004. "Agent-Based Modeling and Social Simulation with *Mathematica* and MATLAB." in *Proc. Agent 2004 Conference on Social Dynamics: Interaction, Reflexivity and Emergence,* edited by C. Macal, C., D. Sallach, and M. North, Chicago, IL, Oct. 7-9, http://www.agent2004.anl.gov/, pp. 185-204.

Macal, C. M. 2009. "Agent Based Modeling and Artificial Life." in *Encyclopedia of Complexity and Systems Science*, edited by R. Meyers, 112-131. Springer. ISBN 978-0-387-75888-6.

Macal, C. M., and M. J. North. 2007. "Agent-based Modeling and Simulation: Desktop ABMS." In *Proceedings of the 2007 Winter Simulation Conference*, edited by S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, 95-106. Piscataway, New Jersey: Institute of Electrical and Electronic Engineers, Inc.

Macal C. M., and M. J. North. 2009. "Agent-Based Modeling and Simulation." In *Proceedings of the 2009 Winter Simulation Conference,* edited by M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin and R. G. Ingalls, 86-98. Piscataway, New Jersey: Institute of Electrical and Electronic Engineers, Inc.

Macal C. M. and M. J. North. 2010. "Tutorial on Agent-Based Modeling and Simulation." *Journal of Simulation* 4:151-162. New York, NY USA: Palgrave Macmillan.

Malleson, N. 2010. "Agent-Based Modelling of Burglary." PhD Thesis, School of Geography, University of Leeds. Accessed August 12, 2011. http://www.geog.leeds.ac.uk/fileadmin/downloads/school/people/postgrads/n.malleson/thesis-final.pdf.

Mast, E. H. M., G.A.M. van Kuik, and G.J.W. van Bussel. 2007. *Agent-Based Modelling for Scenario Development of Offshore Wind Energy*. Delft University of Technology, The Netherlands.

Menges, F., B. Mishra, and G. Narzisi. 2008. "Modeling And Simulation of E-Mail Social Networks: A New Stochastic Agent-Based Approach." In *Proceedings of the 2008 Winter Simulation Conference*, edited by S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, and J. W. Fowler, 2792-2800. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc. Accessed August 12, 2011. https://collaboration.vbi.vt.edu/download/attachments/25953407/menges08.modeling-and-simulation-of-email-social-networks.pdf.

Minar, N., R. Burkhart, C. Langton, and M. Askenazi. 1996. "The Swarm Simulation System, A Toolkit for Building Multi-Agent Simulations." Working Paper 96-06-042, Santa Fe Institute, Santa Fe, NM. Accessed August 12, 2011. www.santafe.edu/projects/swarm/overview/overview.html.

Mock, K. J., and J. W. Testa, 2007. *An Agent-Based Model of Predator-Prey Relationships between Transient Killer Whales and Other Marine Mammals*. University of Alaska Anchorage, Anchorage, AK, May 31, 2007. Accessed August 12, 2011. www.math.uaa.alaska.edu/~orca/.

NetLogo. 2009. NetLogo Home Page. Accessed August 12, 2011. http://ccl.northwestern.edu/netlogo/.

Nikolai, C., and G. Madey. 2009. "Tools of the Trade: A Survey of Various Agent Based Modeling Platforms." *Journal of Artificial Societies and Social Simulation* 12(2). Accessed August 12, 2011. http://jasss.soc.surrey.ac.uk/12/2/2.html.

North, M. J., and C. M. Macal. 2007. *Managing Business Complexity: Discovering Strategic Solutions with Agent-Based Modeling and Simulation*. Oxford: Oxford University Press.

North, M. J. and C. M. Macal. 2011. "Product Design Patterns for Agent-Based Modeling." In *Proceedings of the 2011 Winter Simulation Conference*, edited by S. Jain, R. R. Creasey, J. Himmelspach, K. P. White, and M. Fu. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

North, M., N. Collier, and J. Vos. 2006. "Experiences in Creating Three Implementations of the Repast Agent Modeling Toolkit." *ACM Transactions on Modeling and Computer Simulation* 16(1):1-25.

North, M., C. Macal, J. St. Aubin, P. Thimmapuram, M. Bragen, J. Hahn, J. Karr, N. Brigham, M. Lacy, and D. Hampton. 2009. "Multi-scale Agent-based Consumer Market Modeling." *Complexity* 15(5): 37-47 (May/June).

Opaluch, J. J., J. L. Anderson, and K. Schnier. 2005. "A Risk-Based Approach to Managing the Intentional Introduction of Non-Native Species." Paper presented at the *AERE Workshop Natural Resources at Risk*, Grand Teton National Park, Wyoming, June 12-14.

Ozik, J. 2011. *ReLogo Getting Started Guide*. Accessed August 12, 2011. http://repast.sourceforge.net/docs/ReLogoGettingStarted.pdf.

Puckett, R. R. 2009. "Multi-Agent Crowd Behavior Simulation For Tsunami Evacuation." Masters Thesis, Department of Information and Computer Sciences, University of Hawaii, May. Accessed August 12, 2011. http://www2.hawaii.edu/~puckett/research/past/tsunami.pdf.

SDG. 2009. Swarm Development Group Home Page. Accessed August 12, 2011. www.swarm.org/.

Sopha, B. M., C. A. Klockner, and E. G. Hertwich. 2011. "Exploring Policy Options for a Transition To Sustainable Heating System Diffusion Using an Agent-Based Simulation." *Energy Policy* (in press).

Sterman, J. D. 2000. *Business Dynamics: Systems Thinking and Modeling for a Complex World*. Boston: Irwin McGraw-Hill.

Tesfatsion, L. 2011. "ACE web site: Agent-Based Computational Economics: Growing Economies From The Bottom Up." Accessed August 12, 2011. www.econ.iastate.edu/tesfatsi/ace.htm.

Tobias, R., and C. Hofmann. 2004. "Evaluation Of Free Java-Libraries For Social-Scientific Agent Based Simulation." *Journal of Artificial Societies and Social Simulation* 7(1), Jan. 31.

van Dam, K. H., Z. Lukszo, L. Ferreira, and A. Sirikijpanichkul. 2007. "Planning the Location of Intermodal Freight Hubs: An Agent Based Approach." In *Proceedings of the 2007 IEEE International Conference on Networking, Sensing and Control*, 187-192. London, UK, 15-17 April.

Wang, J., K. Gwebu, M. Shanker, and M. D. Troutt. 2009. "An Application Of Agent-Based Simulation To Knowledge Sharing." *Decision Support Systems* 46:532–541. Accessed August 12, 2011. www.personal.kent.edu/~mshanker/personal/Zip_files/DECSUP11534.pdf.

Wilensky, U. 1999 *Netlogo*, Center for Connected Learning and Computer-Based Modeling, Northwestern University: Evanston, IL USA. Accessed August 12, 2011. http://ccl.northwestern.edu/netlogo/.

Wooldridge, M. 2009. *An Introduction to MultiAgent Systems*. 2nd ed. John Wiley.

XJ Technologies. 2009. AnyLogic Home Page. Accessed August 12, 2011. http://www.xjtek.com/

AUTHOR BIOGRAPHIES

CHARLES M. MACAL is the Director, Center for Complex Adaptive Agent Systems Simulation (CAS^2), Argonne National Laboratory. He is a member of the INFORMS-Simulation Society, Association for Computing Machinery, the Society for Computer Simulation International, and the Systems Dynamics Society. Charles has a Ph.D. in Industrial Engineering & Management Sciences from Northwestern and a Master's Degree in Industrial Engineering from Purdue. Contact: macal@anl.gov.

MICHAEL J. NORTH is the Deputy Director of CAS^2 at Argonne. Michael has over 15 years of experience developing advanced modeling and simulation applications for the federal government, international agencies, private industry, and academia. Michael has a Ph.D. in Computer Science from the Illinois Institute of Technology. Contact: north@anl.gov.