Published as: Brown, D.G. 2006. Agent-based models. In H. Geist, Ed. *The Earth's Changing Land: An Encyclopedia of Land-Use and Land-Cover Change.* Westport CT: Greenwood Publishing Group, pp. 7-13.

Agent-based models

Agent-based models (ABMs) are computer representations of systems that are comprised of multiple, interacting actors (i.e., agents). In a land-use/cover change (LUCC) context, agents can include land owners, farmers, collectives, migrants, management agencies, and/or policy making bodies, all of whom make →decisions or take actions that affect land-use patterns and processes. By simulating the individual actions of many diverse actors, and measuring the resulting system behavior and outcomes over time (e.g., the changes in patterns of land cover), ABMs can be useful tools for studying the effects on land-use/cover processes that operate at multiple →**scales** and organizational levels and their effects.

Agents are discrete entities that are defined in terms of both their attributes and their behaviors. The attributes of agents can be continuous measurements, like age or wealth, or discrete categories, like farmer or member of a particular interest group. The actions of agents can be scheduled to take place synchronously (i.e., every agent performs actions at each discrete time step) or asynchronously (i.e., agent actions are scheduled with reference to a clock or to the actions of other agents). The behaviors of agents can vary from completely reactive, i.e., agents only perform actions when triggered to do so by some external stimulus (e.g., actions of another agent), to goal-directed (e.g., through seeking of a particular goal). For example, an farmer agent could be programmed to plant corn every spring (i.e., a relatively reactive agent) or choose whether to plant and which crop to maximize return on →**investment**, and plant at the time that is expected to produce the highest yield (i.e., a goal-directed agent). Agent-based models for LUCC research are nearly always →**spatially explicit**, which means that the agents and/or their actions are referenced to particular locations on the Earth's surface. For this reason, many agent-based models have either direct or indirect interaction with →**geographic information systems**.

ABMs belong to a category of models known as discrete event simulations, which run with some set of starting conditions over some period of time, allowing the programmed agents to carry out their actions until some specified stopping criterion is satisfied, usually indicated by either a certain amount of time or a specified system state. As such, ABMs are similar to discrete-event models used in a wide range of fields, including individual-based models (IBMs) and gap models, which have been used in biology and ecology to simulate the behavior of animals and plants, respectively.

ADVANTAGES OF AGENT-BASED MODELS

ABMs have a number of strengths that contrast with traditional methods for →**modeling land-use/cover change**. Important among these strengths is the ability to represent agent behavior explicitly while, at the same time, providing a range of options for representing behavior. Traditional land-use models derive from rational utility theory, in which a land-use pattern is explained in terms of actors who maximize utility obtained from various land uses at various locations (in some models the agent decision is what land use to engage in at a location, in others it is where to locate a particular land use). For example, rational utility theory is the basis for von →**Thünen**'s models of agricultural land-use patterns around a city center, and later variants of this model. In such traditional models, homogeneity is assumed in both the →**decision-making** approach and in the factors that influence the utility calculation for each member of the population. Further, all agents are assumed to have perfect information about alternatives and about future utility, and to maximize utility in all situations. Research in the behavioral sciences has produced a range of alternative behavioral models that can be implemented and evaluated with agent-based models. For example: (a) heterogeneity in agent behavior can be represented by drawing parameters of a utility function from a statistical distribution, or using alternative decision making approaches for different agent types (e.g., residents evaluate aesthetics and distance to work and farmers evaluate crop price and yield); (b) rationality can be bounded by imposing limits on the amount of effort agents use to search for and/or evaluate alternatives; (c) alternatives to utility maximization can be included by, for example, using satisficing behavior, in which agents select alternatives that are "good enough" using heuristics to determine agent choices (e.g., based on simple psychological models); (d) randomness in environmental conditions, information availability, or decision outcomes can be incorporated through inclusion of stochastic processes.

In addition to the richer behavioral representations afforded by ABMs, ABMs inherit several advantages from their computer-simulation heritage. Because an ABM is a dynamical system, the model can incorporate positive and negative →**feedbacks**, such that the behavior of an agent has an influence on the subsequent behavior of other agents. These feedbacks can be used to represent the →**endogeneity** of various →**driving forces** of land-use and land-cover change. For example, if the existing amount of developed land is a variable that predicts how much new land will be developed, then developed land area is an endogenously coupled variable. By representing the sequential effects of land development at each time step in the model, a simulation model can explicitly represent this endogeneity. Other simulation models, like →**cellular automata**, share this ability. An important characteristic that distinguishes ABMs from cellular automata (CA) is that, whereas CA usually have a fixed interaction topology (i.e., which neighbors a cell interacts with is fixed by the cellular geometry), interactions in the ABM can be dynamically changed as the model runs, because they are defined at the agent level, rather than in terms of the partitioning of space.

PURPOSE AND USE OF AGENT-BASED MODELS

Like other modeling approaches, the use of ABMs might be summarized in two broad categories, namely explanatory and prognostic purposes. The explanatory uses of models involve their use as a means to codify understanding about processes, evaluate the implications of assumptions, and evaluate the efficacy of this understanding relative to observations. Prognostic uses of models involve their use for extrapolation of trends, evaluation of →**scenarios**, and prediction of future states. Whether explanatory or prognostic, many projects that use ABMs begin with exploratory uses of the models to understand system processes and dynamics before building truly explanatory or prognostic models.

In the context of land-use/cover change (LUCC) research that focuses on explanation and understanding land-related decision-making processes and their implications, ABMs serve as virtual →**land-use systems**, or land use systems *in silica*, within which such land-related processes can be evaluated. Such studies build on the tradition of complex systems research that has been referred to as "generative social science," an approach elaborated by Joshua Epstein. The goal of such studies is to understand the emergence of patterns, trends, or other characteristics observable in society or geography by generating them from models built to represent the actors but not, explicitly, the collective outcomes.

For example, it is a common goal of LUCC studies to characterize the spatial patterns of landscapes through the calculation of metrics that describe the composition and configuration of land cover in a particular area. Such →**pattern metrics** might be used to observe that forests are becoming more fragmented, that urban settlements are becoming more dispersed, or that grasslands are declining. A goal of agent-based modeling in these contexts could be to program plausible agent-level behaviors and interactions that, when run in a simulation, produce similar trends in the patterns as those observed through analysis of maps. Such a model represents a candidate explanation for the emergence of the observed patterns. The challenges in such applications are (a) ensuring that the agent-level representations are plausible and (b) testing alternative models to identify the range of micro-behaviors that can produce a given macro-behavior. If that range is large, then the set of candidate explanations is too.

Prognostic uses of ABMs in LUCC are most likely to be successful in the context of scenario development and policy analysis. In such applications, an ABM whose mechanisms are well supported by empirical understanding and data can be used to evaluate the possible effects on model outcomes of changes in →**initial conditions**, in information available to agents, in constraints or incentives to agent behavior, etc. Such analyses can be used to understand how system dynamics and outcomes can be affected by changing contexts. An obvious application of such approaches is to evaluate the effects of system changes over which some agency or group has some control (e.g., for policy analysis), to evaluate the range of system effects a particular change in policy might be expected to have.

Positive feedbacks, common in many such systems, can create system behavior referred to as →**path dependence**, in which the path a process takes is very sensitive to both initial conditions and small variations in stochastic processes. For this reason, predictive modeling with agent-based models, or any models of systems that contain positive feedbacks, can be challenging. Some land-use processes do contain feedbacks. A classic example is the development of certain industrial regions (e.g., Silicon Valley, USA) in which the returns to locating in a region increase as more and more facilities are located there. Because the location of such a large agglomeration of industries of a given type can be very sensitive to some early decisions that are hard to predict, the land-use system itself can be hard to predict. The impulse to calibrate a model to fit existing data can result in an overfit model, in which the model is overly constrained so that it fits the data but is insufficiently general to represent the full range of possible outcomes from the system.

BUILDING AGENT-BASED MODELS

The process of building ABMs starts with *conceptual modeling*, in which the basic questions or goals, the elements of the system (i.e., agents and other features), the relevant behaviors of the agents, and the measurable outcomes of interest are identified. Depending on the goals of a particular model application, a model may involve the use of designed or empirically grounded agents. If designed, the agents are endowed with characteristics and behaviors that represent, often simplified, conditions for testing specific hypotheses about general cases. If empirically grounded, the agents are intended to represent real individuals or organizations for which data about decision making processes are available. Similarly, the environment within which the agents act can be designed, i.e., given characteristics that are simplified to focus on specific attributes, or empirically grounded to represent specific places. In practice, a study may start with simple models, often with designed agents and environments, to evaluate and understand the dynamics of the system. Where such understanding is more mature, and where a project has prognostic goals related to specific cases, a study would use empirically grounded agents and environments.

Following conceptual modeling, the computer programs must be *designed* in ways that represent the conceptual model with fidelity and that are computationally efficient. The design of an ABM program follows the framework of object-oriented programming. As such classes of objects are defined, some of which are agents and some of which can include other computational objects, e.g., an object that observes and reports the outcomes of the models. The defining characteristics of object-orientation include encapsulation, in which the attributes and methods of objects are bundled together so that the actions of objects and agents can be run independently, and inheritance, in which a sub-class of objects (or agents) inherits the attributes and methods of its super-class. Inheritance makes it possible to efficiently define all agents, as a class with a set of attributes (e.g., type and color), and types of agents, as a sub-class with different sets of additional attributes and methods for each type.

Once a program has been designed, the *software* must be written. ABMs use object-oriented programming languages, like C++, Java, or Objective-C. There are several tools available to assist in building ABMs. Many ABMs are written using sets of software libraries that provide pre-defined routines specifically designed for use in agent-based modeling. Perhaps the most prominent and early of these libraries was SWARM, originally developed at the Santa Fe Institute and programmed with Objective-C, but later supporting models written in Objective-C or Java. Recent years have witnessed the development of several more libraries, including REPAST, ASCAPE, and MASON, all written with Java. Several systems based on Smalltalk have been developed and include CORMAS, which has a natural resource focus, and SDML. The object-oriented programming paradigm of ABMs allows for the incorporation of a range of other software libraries. Importantly for LUCC research, libraries of geographic information systems (GIS), like GEOTOOLS, can enhance the functionality of ABMs, through provision of data management and spatial analytical functionality.

In addition to software libraries, several packages provide routines and software environments that simplify the programming of ABMs and the building of visualizations based on these models. Notable among these packages are STARLOGO, its derivative NETLOGO, and AGENTSHEETS. While these packages are comparatively much easier to use than the Java-based libraries, they also provide more limited functionality.

Once a model has been coded, it must be *verified*, by running the model to test that the specific programmed behaviors perform as expected. These tests can be posed as hypotheses about the behavior of the model under a range of different parameter settings. The hypotheses should be ones for which the answer is known given the conceptual model as designed. If the coded model returns different results, then the modeler should further evaluate the model for logical errors and programming mistakes. Given that complex systems can often return counter-intuitive results, however, care must be taken to determine whether the unexpected result is an error or a feature of the system. Only after the various methods programmed in the model have been tested in this way should the model be considered sufficient for scientific application.

CALIBRATION AND VALIDATION

Some of the hardest issues to deal with in building ABMs are related to →**calibration**, i.e., setting model structure and parameter values in ways that accurately reflect a real-world situation, and →**validation**, i.e., confirming that the model behaves substantially similarly to the way the target system behaves. Calibration typically requires data on the micro-level processes that the model is programmed to represent. These data can be acquired through surveys, statistical analysis of empirical data (e.g., through hedonic price analysis), or experiments specifically designed to elicit decision-making strategies and factors. Validation usually involves comparison of model outcomes, often at the macro-level, with comparable outcomes in the real world (e.g., land cover patterns). Because the models are dynamic, these comparisons often require multi-temporal and spatially explicit data to fully evaluate that a model reflects dynamics accurately. Given the path dependence that can be present in land-use systems, comparison of a model that has multiple possible outcomes with a real world that has only one outcome presents new analytical challenges for agent modelers.

See also Multiplicity; Pattern to process.

Further Readings

AGENTSHEETS [Online, July 2004]. Web Site for Agent-Based Modeling Software Package <agentsheets.com>.

ASCAPE [Online, July 2004]. Web Site for Agent-Based Modeling Java Software Framework <www.brook.edu/es/dynamics/models/ascape/default.htm>.

Brown, Daniel G., Page, Scott E., Riolo, Rick, Zellner, Moira, and William Rand. "Path dependence and the validation of agent-based spatial models of land use." *International Journal of Geographical Information Science* in press.

CORMAS [Online, July 2004]. Web Site for Multi-Agent System Software Development Environment <cormas.cirad.fr/indexeng.htm>.

Epstein, Joshua M. "Agent-based computational models and generative social science." *Complexity* 4 (1999): 41-60.

GEOTOOLS [Online, July 2004]. Web Site for Geographic Information System JAVA Software Toolkit <www.geotools.org>.

Gilbert, Nigel and Steven Bankes. "Platforms and methods for agent-based modeling." *Proc Natl Acad Sci U S A.*, 99 (Suppl. 3) (2002): 7197–98.

Gimblett, Randy H. *Integrating Geographic Information Systems and Agent-Based Modeling Techniques for Simulating Social and Ecological Processes.* New York: Oxford University Press, 2002.

Janssen, Marco A., ed. *Complexity and Ecosystem Management: The Theory and Practice of Multi-Agent Systems.* New York: Edward Elgar, 2003.

MASON [Online, July 2004]. Web Site for Agent-Based Modeling Java Software Library <cs.gmu.edu/~eclab/projects/mason>.

NETLOGO [Online, July 2004]. Web Site for Agent-Based Modeling Software Environment <ccl.northwestern.edu/netlogo>.

Parker, Dawn C., Manson, Steven M., Janssen, Marco A., Hoffman, Matthew J., and Peter Deadman. “Multi-agent systems for the simulation of land-use and land-cover change: A review.” *Annals of the Association of American Geographers*, 93 (2003): 314-37.

Parker, Dawn C., Berger, Thomas, and Manson, Steven, ed. *Agent-Based Models of Land-Use and Land-Cover Change: Report and Review of an International Workshop, Irvine, CA, USA, October 4-7*. (LUCC Report Series No. 6). Bloomington, IN: LUCC Focus 1 Office, 2002.

REPAST [Online, July 2004]. Web Site for Agent-Based Modeling Java Software Library <repast.sourceforge.net>.

SDML [Online, July 2004]. Web Site for Agent-Based Modeling Software Language and Environment <sdml.cfpm.org>.

STARLOGO [Online, July 2004]. Web Site for Agent-Based Modeling Software Environment <www.media.mit.edu/starlogo>.

SWARM [Online, July 2004]. Web Site for Agent-Based Modeling Objective-C and Java Software Library <wiki.swarm.org>.

Daniel G. Brown