

Agent-Based Modeling Position Statement

Kevin M. Johnston and David Maguire (ESRI)

This position paper is written from the perspective of ESRI as a developer of commercial-off-the-shelf GIS products that ship in volume. Our overarching goal is to build software that solves important problems in a widely applicable way. As such, solutions must be robust, reliable, generic, deliver acceptable performance on large, real world data sets, and be easy to implement.

ESRI has 25 years of experience creating spatial modeling tools for vector and raster analysis. In the last five years we have built a geoprocessing modeling environment that synthesizes our experience in spatial modeling into an easy to use, flexible framework. Within the framework hundreds of spatial modeling functions are available each as individual tools that can be accessed from dialogs, the command line, a graphical modeling environment (ModelBuilder), scripting, or from any programming language. The framework is fully object-oriented and is built on .Net standards. In using ModelBuilder a model is comprised of data that are connected to spatial and aspatial tools to create a process to perform some function. Processes can be linked to one another to create simple as well as complex models. This framework has been used extensively to create static, cartographic process models that encompass classic map analysis/algebra operations.

In the latest 9.2 release of ArcGIS we expanded the geoprocessing environment, in particular ModelBuilder, with looping capabilities and the ability to model stochastic events through a series of random number and data creation functions. With these new capabilities we have explored how to add time to models and how to create various simulation scenarios in a dynamic modeling environment. We have been particularly interested in how to develop process models (e.g., a fire growth model), process models with stochastic events (e.g., a fire growth model with spotting), error analysis through simulations (e.g., examine the effects of error in a DEM on a stream delineation model), and sensitivity analysis (e.g., how does the output change with a 5% change in the distance to roads in a housing suitability model).

With a strong interest in agent-based modeling, we worked with Argonne National Laboratories, the creators of the RePast ABM system, to build Agent Analyst. Agent Analyst is a free open source extension designed to perform ABM within the ArcGIS environment. One of the main goals for the integration of a GIS with an ABM is to provide a realistic landscape that can change through time on which the ABM runs. Agent Analyst is a mid-level integration of ArcGIS and RePast. ArcGIS provides spatial tools to prepare input data, manages the real landscape, displays the simulation results for each time step, and hundreds of spatial operations that can be used in the ABM. RePast provides the agent control and the scheduler for the ABM.

We created several ABM's to test the integration of Agent Analyst to ensure the user can perform tasks on large realistic datasets. One particular model was a cougar movement model. We worked with the USGS in Flagstaff, AZ to explore the interaction of cougars with their landscape and encounters with people. The model was performed on a large real landscape dataset (10 GB) and the rules were derived from scientists and data collected from collared cougars. We are aware of several other ABM models that have been implemented within this framework.

We found the software performed reasonably well. However the software is quite difficult to use especially for non-programmers. There is no easy way to create the rules for an agent other than through code. Looking through the future, we are interested to see if it is possible to develop a generic ontology that could be used to define agent rules. From this ontology a graphical user interface might be created which would greatly widen the audience for ABMs. Second, agents are still moved in x, y space independent of the GIS and then linked back to ArcGIS by feature data updates. Third, it is somewhat difficult to access the Java ArcObjects from within the ABM. This has limitations for future extensions of the framework and for creating more specific models.

Through creating and using Agent Analyst we discovered that GIS and ABM integration is more synergistic and bi-directional than we anticipated. The GIS provides input for creating agents (e.g., a shapefile), creates changing landscapes, and provides spatial modeling tools that are used each time step of the simulations. The ABM models and tracks the state changes and the decision making of the agents on this changing landscape and utilizes spatial functions (e.g., how far am I from something) in the decision making process for each time step.

Our work on Agent Analyst and dynamic modeling has revealed a number of important software issues:

- how to store and manage many outputs from a simulations and scenarios
- how time is explicitly handled
- synchronizing input time series data with different time intervals
- tools to analyze the simulation results
- metrics to compare and evaluate different scenarios

We are increasingly coming to the view that the distinction between dynamic modeling and ABM is fuzzy. Issues that promote the fuzziness include, if in the dynamic modeling framework the modeler can change the feature attributes like in the Schelling model is this ABM? With iteration and neighborhood notation in Spatial Analyst cellular automata models can be created; is this ABM? If Spatial Analyst changes from a passive mode (e.g., can only change your own value) to an active mode (e.g., allow the processor change other cell values) a modeler can then model the movement of oil from the perspective of the spill exchanging concentrations based on a series of neighborhood locations simultaneously and can change cell values beyond the immediate neighborhood in a CA model; is this ABM? If features and rasters are processed randomly instead of sequentially then situations in which each object changes the decisions of other objects can be addressed since the object making the first move is randomly determined. By processing a raster randomly the modeler can also perform dynamic allocations such as siting 50 bird nests in the most suitable locations assigning restrictions such as no three nests can be within 100 meters of one another; is this ABM? Changing states of features and rasters, moving agents based on the characteristics at each location on the landscape (grazing models), and cellular automata are relatively easy to do in a dynamic modeling environment. To address ABMs where each discrete object is aware of how to make complex decisions and changes the decision making process through mechanisms such as learning and memory poses a greater challenge for a general dynamic modeling environment.

Is ABM a formal modeling procedure or a state of mind derived from the object-oriented coding constructs? Castle and Crooks (2007) summarizes several studies defining the functionality of an ABM as containing autonomy, heterogeneity, active, pro-active / goal-directed, reactive / perceptive, bounded rationality, interactive / communicative, mobility, adaptation / learning. ESRI wants to understand what is the necessary functionality to meet these requirements. We are interested in understanding how to implement non-linear interactions, stochastic events, and how modeling objects make trade offs between multiple criteria in decision making.

By formally defining ABM the requirements for creating ABM models will become more obvious. Clearly there is great overlap between dynamic modeling and ABM. What we at ESRI are most interested in is the essence of the functionality that is being asked for by both dynamic models and ABM. If we can define the essence of the functionality, we would like to create an integrated dynamic modeling environment that can hopefully encompass ABM.