Geographic Automata Systems: A New Paradigm for Integrating GIS and Geographic Simulation

Itzhak Benenson[1], Paul M. Torrens[2]

[1]Department of Geography and Human Environment and Environment Simulation
Laboratory, Tel Aviv University, Israel,
bennya@post.tau.ac.il

[2]Department of Geography and Centre for Advanced Spatial Analysis,
University College London, UK,
torrens@casa.ucl.ac.uk

A new wave of urban simulation models has come to the fore in recent years, supported by an array of advances both in the geographical sciences and in fields outside geography. This new wave can be characterized by a distinctly innovative approach to modeling—the geosimulation approach. Geosimulation is concerned with automata-based methodologies for simulating discrete, dynamic, and action-oriented spatial systems, combining cellular automata and multi-agent systems in a spatial context. In geosimulation-style models, urban phenomena as a whole are considered as the outcome of the collective dynamics of multiple animate and inanimate urban objects.

GIS have an integral role in the development of geosimulation models, providing the dataware for simulations. Yet, there remains much potential for coupling both GIS and geosimulation, fusing the two into full-blown, symbiotic systems. In this paper, we propose a paradigm for integrating GIS and geosimulation into what we term *Geographic Automata Systems* (GAS). With the aid of the *Object-Based Environment for Urban Simulations* (OBEUS) software developed at the Environment Simulation Laboratory of the University of Tel Aviv, we will demonstrate the use of this new paradigm, re-working traditional urban geographic models as Geographic Automata Systems and presenting examples of new models.

Many questions arise when applying automata-based simulation approach for spatial purposes. How can a variety of topologies be incorporated into the models, e.g., regular tessellations, irregular tessellations, networks, graphs, dynamic polygons, etc.? How should fuzzy entities be represented spatially in the models? How can theoretical ideas relating to spatial mobility—way-finding, spatial cognition, action-at-a-distance, etc.—be incorporated into the behavioral rules of mobile agents or immobile cells and parcels and further translated into automation rules? How can the global patterns generated in simulations be recognized and validated?

Of course, researchers in Geographic Information Science have been grappling with many of these issues for years. To a certain extent, contemporary GIS can provide much support for the development of geosimulation models by storing and retrieving the states and location of non-modifiable spatial objects, just as they store and retrieve static information regarding the real world. In this sense, GIS become loosely-coupled to geosimulation models.

However, the possibility for a tightly-coupled relationship between GIS and automata-based models exists. The power of GIS to register spatial actions can be used to support geosimulation models and, reciprocally, geosimulation models can be used to encode simulation functionality in GIS. The fusion of GIS and geosimulation that we propose offers the opportunity for developing a new paradigm centered on GIS as a simulation framework, built around an automata-based core.

Geosimulation models and modern GIS are object-based in their design; both deal with discrete spatial objects (entities and features), which customarily represent the real world at “microscopic” scales. The geosimulation approach is noteworthy in its ability to model the city as a *collective* of objects. To achieve this, traditional object-based concepts are extended and spatial entities are considered as action-oriented *geographic automata*. As is the case with general automata, the rules that govern the change of *state* of automata objects in the system are defined. In addition, the *location* of automata and rules of transition between locations are also specified. An obvious approach to achieving this functionality is to extend the *entity-relationship* data model for GIS toward object-oriented modeling; in this framework, automata rules may be reformulated as class methods.

Geographic Automata Systems consider physical and social components of geographic systems simultaneously, while focusing on *location* and *spatial relations* between automata. *Unitary* geographic automata can be either *immobile, e.g.*, cellular automata cells, households, road segments, land parcels; or *mobile*, e.g., migrating householders, vehicles in traffic queues, pedestrians in shopping districts. Immobile objects are situated in space in a standard GIS fashion, by coordinates, while mobile objects are situated in space by *pointing to* immobile entities. Besides non-modifiable spatial GIS units, geosimulation considers assemblies of “atomic” automata self-organizing from their dynamics, e.g., residential communities, commercial areas, congested road links. To represent these emerging entities at higher levels of urban hierarchy we consider *domains*—conglomerations of unitary geographic automata satisfying the criteria given by set of predicates.

Formally, a Geographic Automata System (GAS), **G**, may be defined as consisting of seven components:

$$\mathbf{G \sim (K; S, T_S; L, M_L; R, N_R)}$$

Here, **K** denotes a set of *types* of automata participating in the GAS and each of three pairs represents a specific component and the rules that determine their dynamics.

The first pair denotes a set of *states* **S**, associated with the GAS, **G** (consisting of subsets of states $\mathbf{S}^{\mathbf{k}}$ of automata of each type **k Î K**), and a set of state transition rules $\mathbf{T}_{\mathbf{S}}$, used to determine how automata states should change over time. The second pair represents location information. **L** denotes the geo-referencing conventions that dictate the location of automata in the system and $\mathbf{M}_{\mathbf{L}}$ denotes the movement rules for automata, governing changes in their location. State transitions and changes in location for geographic automata depend on the state of automata themselves and the states of neighbors, the latter specified by the third pair. **R** represents there the neighbors of the automata and $\mathbf{N}_{\mathbf{R}}$ represents the neighborhood rules that govern how automata relate to the other automata in their vicinity.

If the state of a geographic automaton, **g**, at time **t** is $\mathbf{s_t} \in \mathbf{S}$ and the automaton is located at $\mathbf{l_t}$, then its neighbors are $\mathbf{r_t}$. The state transition, movement, and neighborhood rules, $\mathbf{T_S}$, $\mathbf{M_L}$, and $\mathbf{N_R}$ define **g**'s state, location and neighbors at time **t + 1**. To animate, or spatially enable the GAS, these rules are applied to each of its automata:

$$\mathbf{T_S : (s_t, l_t, r_t)} \ \circledR \ \mathbf{s_{t+1}}$$
$$\mathbf{M_L : (s_t, l_t, r_t)} \ \circledR \ \mathbf{l_{t+1}}$$
$$\mathbf{N_R : (s_t, l_t, r_t)} \ \circledR \ \mathbf{l_{t+1}}$$

Exploration with GAS then becomes an issue of qualitative and quantitative investigation of the spatial and temporal behavior of **G**, given all of the components defined above. In this way, GAS models offer a framework for considering the *spatially enabled* interactive behavior of elementary geographic objects in a dynamic system.

The Geographic Automata System paradigm extends GIS, and extends its limits. However, this paradigm has its own and definite limitations. As with any automata, geographic automata are supposed to be spatially non-modifiable. The transition rules $\mathbf{T_S}$, $\mathbf{M_L}$, and $\mathbf{N_R}$ (no matter how multiple they may be), which determine the change of states and location of geographic automata, as well as predicates that define domains, should be defined *a priori*.

The automata constraints make the Geographic Automata Systems an extension of the object-oriented GIS. Automata of each type determine the make-up of a database, which stores their properties; in the same manner, the set of predicates determining domains in the model defines the domain tables. The relations between unitary objects and unitary objects and domains, necessary for applying transition rules, are defined in relationship tables. The separation between automata and relationships in GAS implementations is indicative of problems of concurrency when updating one-to-many and many-to-many relationships between geographic objects. Resolving the problem depends on the semantics of relationships and a trade-off exists between the limitations we impose on relationships and the universality of the simulation environment.

Based on the differentiation between mobile and immobile automata and the location of mobile automata by pointing to immobile entities, a minimal GAS software implementation of GAS for urban modeling - *Object-Based Environment for Urban Simulations* (OBEUS) is currently in development at the Environment Simulation Laboratory of the University of Tel Aviv. This implementation includes all the basic characteristics of GAS, but specifies them in the simplest possible manner.

The Geographic Automata Systems framework will be demonstrated in two ways. First, we will demonstrate how available urban models are all "translated" into Geographic Automata Systems by means of explicit types of automata and transition rules; minimal Geographic Automata Systems framework is sufficient for this translation. Second, an operational implementation will be demonstrated in the context of GAS-based simulation software OBEUS. We will present and discuss a number of abstract and real-world Geographic Automata Systems models developed within this environment.