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

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

[1]Environment Simulation Laboratory Porter School of Environmental Studies, and Department of Geography and Human Environment Tel Aviv University, Israel,
[2]Department of Geography, University of Utah, United States,
bennya@post.tau.ac.il, torrens@geosimulation.com

Abstract

Geographic simulation 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 this paper, we propose a paradigm for integrating GIS and geosimulation into what we term *Geographic Automata Systems* (GAS), the latter fusing the two into full-blown, symbiotic systems.

1. Introduction

New forms of simulation have come into popular use in geography and social science in recent years, supported by an array of advances both in the geographical sciences and in fields outside geography. These models and simulations 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 (Benenson & Torrens 2004a, b). 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.

Geosimulation is based on Cellular Automata (CA) and Multi-Agent Systems (MAS). Applied in isolation, CA and MAS approaches have been used to simulate a wide variety of urban phenomena (Torrens 2002a) and there is a natural imperative to combine these frameworks and open numerous avenues for exploratory and applied simulation in urban geography (Torrens 2002a, 2003a). Nevertheless, the direct amalgamation of CA and MAS eventually suffers from awkward compromises, entailed by the necessity for CA partition of urban space into cells. Given cell partition, no matter whether units are identical or vary in size and shape, cells are either granted some degree of 'agency' and are simply reinterpreted as artificial agents (Box 2001) or MAS are imposed on top of CA and simulated agents respond to averaged cell conditions (Benenson 1998; Polhill *et al.* 2001). These frameworks are certainly useful, but are a function of the limitations of the available tools rather than laws of real urban systems behavior.

The dataware for simulations is often provided by GIS, which have an integral role in the development of geosimulation models. Dramatic changes in geographic databases during the last decades of the Twentieth Century have ushered in a new wave of urban modeling (Benenson *et al.* 2002). Automated procedures for data collection—remote sensing by spectrometer, aerial photography, video, etc.—have provided new information sources at fine resolutions, both spatial and temporal (Torrens 2004). New methodologies for manipulating and interpreting spatial data developed by Geographic Information Science and implemented in GIS have created added-value for these data. Information collection is much more pervasive than before (Brown & Duguid 2000), and high-resolution databases for urban land-use, population, real estate, and transport are now more widespread.

Yet, there remains much potential for fusing GIS and geosimulation into full-blown, symbiotic systems. In this paper, we propose a paradigm for integrating GIS, CA and MAS into what we term *Geographic Automata Systems*. This approach is intended to be 'down to earth', and we will demonstrate the implementation of the approach, in this paper, with reference to GAS-based software developed at the Environment Simulation Laboratory of the Porter School of Environmental Studies at the University of Tel Aviv—the *Object-Based Environment for Urban Simulations*. This paper builds on discussions of urban geocomputation (Benenson & Omer 2000; Torrens & O'Sullivan 2000) and software environments for geocomputational research in urban contexts (Benenson *et al.* 2001), which were presented at previous Geocomputation meetings.

2. The basic automata framework

Put very simply, an automaton is a processing mechanism. An automaton is a discrete entity, which has some form of input and internal states. It changes states over time according to a set of rules that take information from an automaton's own state and various inputs from outside the automaton to determine a new state in a subsequent time step. Formally, an automaton, **A**, can be represented by means of a set of *states* **S** and a set of *transition rules* **T**.

$$\mathbf{A} \sim (\mathbf{S}, \mathbf{T}) \qquad (1)$$

Transition rules define an automaton's state, $S_{t+1,}$ at time step t + 1 depending on its state, S_t (S_t, $S_{t+1} \in$ S), and *input*, I_t, at time step t:

$$\mathbf{T}: (S_t, I_t) \rightarrow S_{t+1} \qquad (2)$$

Automata are discrete with respect to time and have the ability to change according to predetermined rules based on internal (**S**) and external (**I**) information.

In terms of urban applications, automata lend themselves to specification as city simulations with myriad states and transition rules. However, to make sense, an individual automaton should be as simple as possible in terms of states, transition rules, and internal information (Torrens & O'Sullivan 2001). Simplicity is a characteristic of the most popular automata tools in urban geography, Cellular Automata (CA)—a system of spatially located and inter-connected automata.

3. From General to Cellular Automata

CA are arrangements of individual automata in a partitioned space, where each unit (cell) is considered as an automaton **A**, for which input information **I** necessary for the application of transition rules **T** is drawn from **A**'s *neighborhood* **R**. In urban applications, cells are most commonly used to represent land units with state representing possible land-uses (White *et al.* 1997). Usually, CA lattices are partitioned as a regular square or hexagonal grid. We can specify (1) – (2) to specify an automaton, **A**, belonging to a CA lattice as follows:

$$\mathbf{A} \sim (\mathbf{S}, \mathbf{T}, \mathbf{R}) \qquad (3)$$

where **R** denotes automata neighboring **A**.

Although there are direct analogies between land parcels and cells on the one hand and land-uses and cell states on the other, there were geographical applications of CA models until the 1990s. There were a few examples published in the 1970s (Chapin & Weiss 1968; Nakajima 1977; Tobler 1970, 1979), but the field was largely ignored in terms of research until interest was revived in the 1980s (Couclelis 1985; Phipps 1989). Beginning in the 1990s CA modeling became a popular research activity in geography, with pioneering applications in urban geography (Batty *et al.* 1997; O'Sullivan & Torrens 2000).

In terms of space, neighborhood relationships are important for making CA *spatial* systems. In basic CA, neighborhoods have identical form for each automaton, e.g., Moore or von Neumann (Figure 1a). In the last decade it became clear, however, that reliance on regular partitions of space is largely superficial in urban contexts (Torrens & O'Sullivan 2001). Consequently, CA have been implemented on irregular networks (Figure 1b), or partitions given by GIS-based coverage of land parcels or Voronoi tessellations (Figure 1c) (Benenson *et al.* 2002; O'Sullivan 2001; Semboloni 2000; Shi & Pang 2000). An assortment of definitions of neighborhoods, based on connectivity, adjacency, or distance can be applied to these generalized CA, where the form of the neighborhood and the number of neighbors varies between automata.

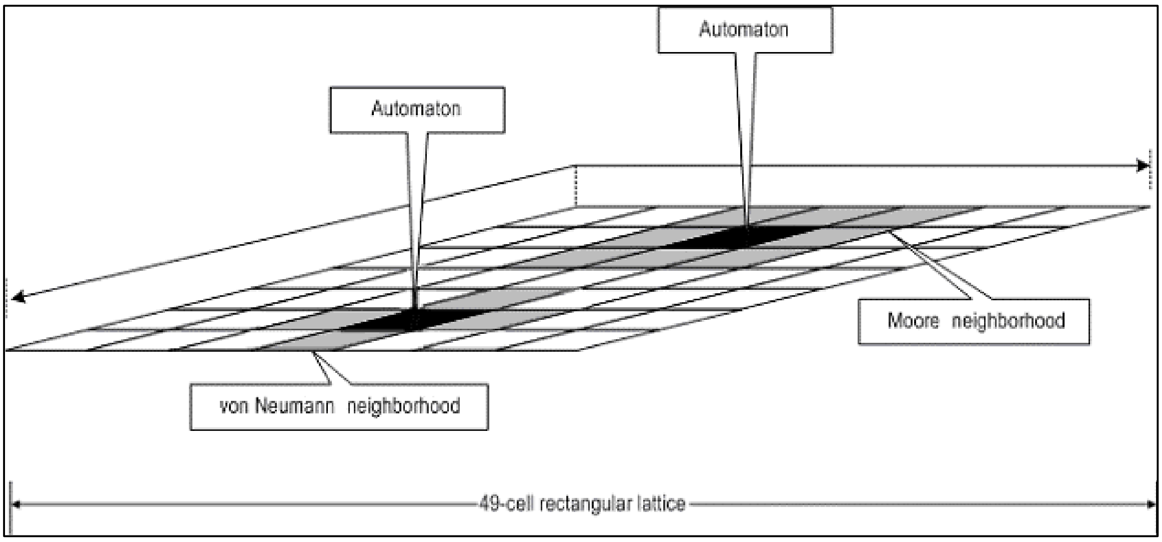

Figure 1a. Automata defined on a two-dimensional regular lattice, with von Neumann and Moore neighborhoods represented.

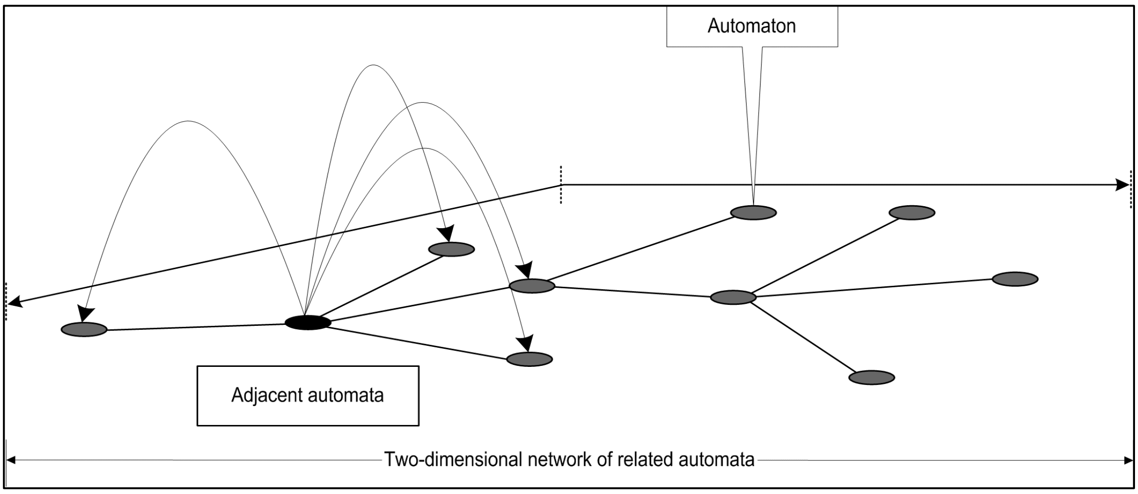

Figure 1b. Automata defined on a two-dimensional network.

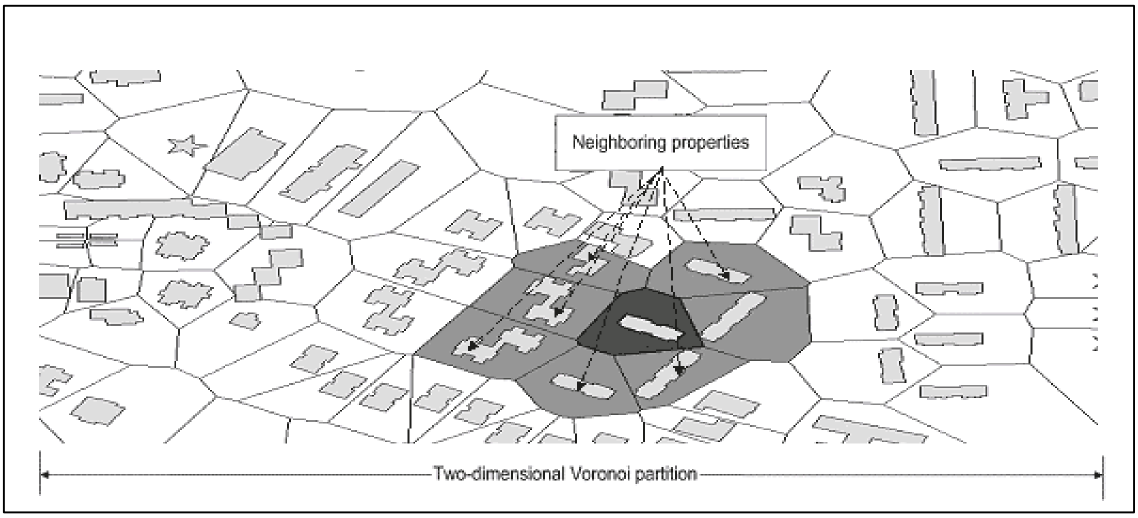

Figure 1c. Automata defined on a Voronoi partition of two-dimensional urban space, based on property coverage.

Another, more important weakness of the CA approach is the inability of automata cells to move within the lattice in which they reside. Despite repeated attempts to mimic units' mobility (Benenson 1998; Schofisch & Hadeler 1996; Wahle *et al.* 2001), the genuine inability to allow for automata movement in the CA framework catalyzed geographers' recent interest in MAS. This tendency is especially strong in urban geography, where the CA framework is regarded as insufficient in dealing with mobile objects such as pedestrians, migrating households, or relocating firms.

4. From General to Agent automata

Fundamentally, a*gents* are automata and, thus, incorporate all of the features of basic automata that have just been discussed. However, there are some important distinctions between general and agent automata, largely owing to the fact that the latter are generally interpreted for representation of autonomous *decision-makers* (Epstein 1999;

Kohler 2000). In urban studies, the states **S** of agent automata in Multi-Agent Systems (MAS) are usually designed to represent socioeconomic characteristics; transition rules are formulated as the rules of agent decision-making, and correspond to human-like *behaviors*.

In the social sciences outside geography, work in agent-based simulation is usually *non-spatial*; many of the decisions and behaviors of geographic *agents* are, however, spatial in nature. Consequently, the states **S** of *geographic* MAS should include agents' *location*, and transition rules **T** should reflect, thus, the ability of agents to relocate.

Human-based interpretations of MAS have their foundation in the work of Schelling and Sakoda (Sakoda 1971; Schelling 1969, 1971, 1974, 1978). Just as with CA, the tool began to feature prominently in the geographical literature only in the mid-1990s (Benenson 1999; Dijkstra *et al.* 2000; Portugali *et al.* 1997; Sanders *et al.* 1997). Until recently, the main thrust of MAS research in geography involved populating regular CA with agents of one or several kinds, which could *diffuse* between CA cells. Often, it is assumed that agents' migration behavior depends on the properties of neighboring cells and neighbors (Epstein & Axtell 1996; Portugali 2000). Very recently, agent-based models have been designed, which locate agents in relation to real-world geographic features, such as houses or roads, the latter stored as GIS layers (Benenson *et al.* 2002) or landscape units—pathways and view points (Gimblett 2002).

Despite the widely acknowledged suitability of automata tools for geographic modeling (Gimblett 2002), there has been relatively little exploration into addressing these limitations and developing patently *spatial* automata tools for urban simulation. In the framework described in this paper, spatial abilities are treated with paramount importance and we define a class of automata that is capable of supporting explicit expression of processes as *geography* comprehends them.

5. What geography needs from automata systems

Focusing on space, we might identify three internal geographic mechanisms that are essential to an urban automata system:

- A typology of entities regarding their use of space in which they are situated;
- The spatial relationships between entities;
- The processes governing the changes of their location in space.

Simulating spatial systems, then, involves explicit formulation of these three components and neither CA nor MAS fully provide the necessary framework. The geography of the CA framework is problematic for urban simulation because CA are incapable of representing autonomously mobile entities. At the same time, MAS are weak as a single tool because of common underestimation of the importance of space and movement behavior.

It is evident, then, that there is a need for uniting CA and MAS formalisms in such a way as to directly reflect a geographic and object-based view of urban systems. The Geographic Automata System framework that we propose attempts to do that.

6. An idea: Urban object = Geographic Automaton

As a spatial science, geography concerns itself with the behavior and distribution of *objects in space*. In urban geography, these are urban agents—householders, pedestrians, vehicles—and urban features—land parcels, shops, roads, sidewalks, etc. In dynamic spatial systems, all these objects change their properties and/or location; the goal of a geographic model is to simulate these activities and their consequences, often at multiple scales.

In developing Geographic Automata Systems, our aim is to infuse spatial properties into automata tools and adopt an *object-based* view of urban systems. Objects are conceptualized as *geographic automata*, with focus on their *spatial properties and behaviors*. Under this framework, a city system can be modeled as an ensemble of geographic automata, i.e. a Geographic Automata System.

7. Formal definition of Geographic Automata Systems

Geographic Automata Systems consist of interacting *geographic automata* of various types. In general, automata are characterized by states **S**, description of the information input **I**, and transition rules **T**. In the case of geographic automata, we re-interpret these components to enable the explicit consideration of *space* and *spatial behavior*. We also add additional components.

Specification of state set **S**:

- In addition to non-spatial *states*, geographic automata are also characterized by their *locations.*
- Instead of the fixed location of CA automata, we introduce a set of *geo-referencing rules* for situating geographic automata in space.

Specification of input information **I**:

- Instead of relying on fixed neighborhood patterns that are incapable of being varied in space or time, we define *neighborhood relations* that can change in time and determine the automata providing input information for a given one.
- Neighborhood relations can thus change; these changes are governed by *neighborhood rules*.

Specification of transition rules **T**:

- *State transition rules* specify the changes of non-spatial states.
- The ability of changing location is given by a set of *movement rules* that allow for navigation of geographic automata in their simulated environments.

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

$$\mathbf{G} \sim (\mathbf{K};\, \mathbf{S},\, \mathbf{T_S};\, \mathbf{L},\, \mathbf{M_L};\, \mathbf{R},\, \mathbf{N_R}) \qquad (4)$$

Here **K** denotes a set of *types* of automata featured in the GAS and three pairs of symbols denote the rest of the components noted above, each representing a specific property and the rules that determine its dynamics.

The first pair denotes a set of *states* **S**, associated with the GAS, **G** (consisting of subsets of states $\mathbf{S}^k$ of automata of each type $k \in \mathbf{K}$), and a set of state transition rules $\mathbf{T_S}$, 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 $\mathbf{M_L}$ denotes the movement rules for automata, governing changes in their location.

According to general definition (1) – (2), state transitions and changes in location for geographic automata depend on automata themselves and on input (**I**), given by the states of neighbors. The third pair in (4) specifies this condition. **R** represents neighbors of the automata and $\mathbf{N_R}$ represents the neighborhood transition rules that govern how automata relate to the other automata in their vicinity.

Let the state of geographic automaton G at time t is S_t, it is located at L_t, and the external input, I_t, is defined by its neighbors R_t. The state transition, movement, and neighborhood rules—$\mathbf{T_S}$, $\mathbf{M_L}$, and $\mathbf{N_R}$—define state, location and neighbors of a given automaton G at time t + 1 as:

$$\begin{aligned} \mathbf{T_S}&: (S_t, L_t, R_t) \rightarrow S_{t+1} \\ \mathbf{M_L}&: (S_t, L_t, R_t) \rightarrow L_{t+1} \\ \mathbf{N_R}&: (S_t, L_t, R_t) \rightarrow R_{t+1} \end{aligned} \qquad (5)$$

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 system.

8. Tight-coupling between GAS and GIS

Many questions arise when applying automata-based simulation approaches 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? Answering these questions often requires a tight-coupling between GIS and automata-based models.

Of course, the nature of GIS, as a special kind of Database Management System (DBMS), provides much support for GAS models. First and foremost, this is the standard ability of GIS to store and retrieve the location and states of spatial objects and to register spatial actions. The next step toward real fusion of GIS and geosimulation focuses on *object-based* views of urban reality.

Indeed, both GAS and GIS are object-based in their design; both deal with discrete spatial objects, which customarily represent the real world at "microscopic" scales. A geosimulation approach considers the city as a *dynamic collective* of action-oriented objects. GIS deals mostly with static objects and employs an *entity-relationship model* (ERM) for their representation. We merge GAS and GIS functionality by implementing

GAS as a specialized object-oriented database of geographic automata. GAS transition rules are thus reformulated as methods of automata classes. Let us specify GAS definition (4) – (5) in terms of abstract OODBMS.

8.1. Geographic automata types, K

At an abstract level, we distinguish between *fixed* and *non-fixed* geographic automata. Fixed geographic automata represent objects that do not change their location over time and thus have close analogies with CA cells. In the context of urban systems, these are objects such as road links, building footprints, parks, etc. Fixed geographic automata may be subject to state and neighborhood transition rules $\mathbf{T_S}$ and $\mathbf{N_R}$, but not of rules of motion, $\mathbf{M_L}$.

Non-fixed geographic automata symbolize entities that change their location over time, for example: pedestrians, vehicles, and households. The full range of rules for GAS can be applied to non-fixed geographic automata.

8.2. Geographic automata states and state transition rules, S and $\mathbf{T_S}$

State transition rules $\mathbf{T_S}$ are based on geographic automata of *all* types from **K**. In contrast to CA, the states **S** of urban fixed infrastructure objects depend on the neighboring objects of the infrastructure, but are also driven by non-fixed geographic automata—agents—that may be responsible for governing object states such as land-use, land value, etc. In this way, urban objects do not simply mutate like bacteria (O'Sullivan & Torrens 2000); rather, state transition is governed by other objects, the latter crucial for simulating *human-driven* urban systems, in which people are affected by their environments and also change them.

8.3. Geo-referencing conventions, L and movement rules $\mathbf{M_L}$

Geo-referencing conventions are crucial for coupling GAS and GIS. On the one hand, they should be sufficiently flexible to enable translation of a geographic view of locating real world objects, both fixed and non-fixed; on the other, they should satisfy limitations of entity-relationship models, just to be convenient for GIS management.

The GAS framework resolves these limitations by introducing two forms of geo-referencing—*direct* and *by pointing*. Direct methods of geo-referencing follow a vector GIS approach, using coordinate lists. Such a list indicates all spatial details necessary to represent an object—automata boundaries, centroids, nodes' location, etc. Fixed geographic automata are usually located by means of direct geo-referencing. The details of the particular rules employed depend on the automata used in a modeling exercise. For typical urban objects such as buildings or street segments, 2D basement polygons or 3D prisms may be used.

Non-fixed geographic automata may move; updating position coordinates might cause difficulties, which, when the automata are multiple and their shape is complex, are irresolvable in the framework of an entity-relationship model. We address these difficulties with a second method of geo-referencing—by *pointing* to other automata (Figure 2).

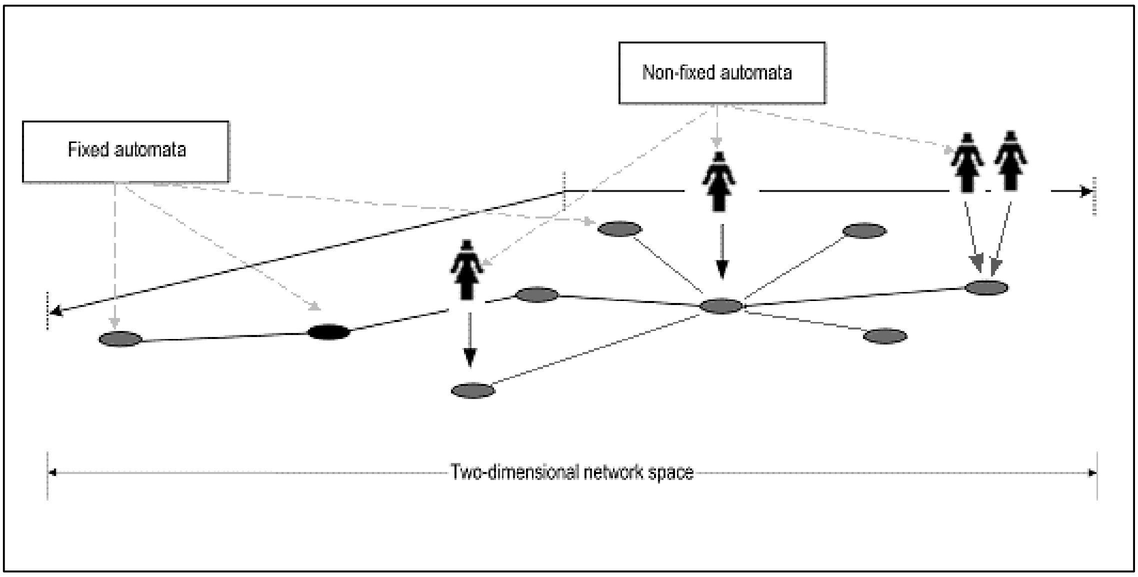

Figure 2. Direct and indirect geo-referencing of fixed and non-fixed GA

Let us consider a typical example. In the case of property dynamics, householders can be geo-referenced by address. Landlords provide a more complex case: they can be geo-referenced by pointing to the properties that they own. Indirect referencing can also be used for fixed geographic automata, e.g., for apartments in a house (Torrens 2001). Referencing by pointing is dynamic-enabled both in space and in time and compatible with the ERM database model.

GAS-based research into different formulations of $\mathbf{M_L}$ offers great potential for geographers. Realistic rules, $\mathbf{M_L}$, for encoding automata movement, based on repel-attract-synchronize interactions between close neighbors, are being developed, for example, in Animat research (Meyer et al. 2000) and in the gaming industry (Reynolds 1999). There is much opportunity for geographers to contribute to this line of research. Traditionally, attention has focused on the generation of realistic choreographies for automata, particularly in traffic models, through the specification of rules for collision avoidance, obstacle negotiation, lane-changing, flocking, behavior at junctions, etc. (Torrens 2003b). However, there remain many relatively neglected areas of inquiry: spatial cognition, migration, way finding, navigation, etc.

8.4. Neighbors and neighborhood rules, R and $\mathbf{N_R}$

The set of neighbors of automata, **R**, is necessary for determining input information **I**. In contrast to the static and symmetrical neighborhoods usually employed in CA models (Figure 1a), spatial relationships between geographic automata can vary in *space and time*, and, thus, rules for determining the neighborhood relationships $\mathbf{N_R}$ is thus necessary. Neighborhood rules for fixed geographic objects are relatively easy to define (simply because the objects are static in space). There are a variety of *geographical* ways in which neighborhood rules can be expressed for them—via adjacency of the units in regular or irregular tessellations, connectivity of network nodes, proximity, etc., (Figure 1b, 1c). Spatial notions related to the incorporation of human-like automata into GAS, such as accessibility, visibility, and mental maps can be formally encoded as $\mathbf{N_R}$ rules.

Non-fixed geographic automata pose more of a challenge when specifying neighborhood rules, because the objects—and hence their neighborhood relations—are dynamic in space and time. It can be done straightforwardly, via distance and nearest-neighbor relations, as used in Boids models (Reynolds 1987), but just as movement rules $\mathbf{M_L}$, can become very heavy computationally when complex definitions of, say, visibility or accessibility are involved. It this case it is more appropriate to base neighborhood rules on indirect location, as defined in the sections above, and consider two indirectly located automata as neighbors when *the automata they point to are neighbors*.

For example, two householder agents can be established as neighbors by assessing the neighborhood relationship between the houses in which they reside. Even when these agents are physically separated in the simulation—when they go shopping or go to work, for example—they remain 'neighbors' by virtue of the (fixed) relationship between their properties.

9. Temporal dimension of GAS

A basis for fixed and non-fixed objects, and direct and by-pointing locating, implies specification of Geographic Automata Systems within Object-Oriented GIS. However, the dynamic nature of GAS also implicates *temporal* dimensions of GIS databases, and, thus entails its own limitations. Given these considerations, transition rules $\mathbf{T_S}$, $\mathbf{M_L}$, and $\mathbf{N_R}$, should be defined in a way that avoids conflicts when states, locations, or neighborhood relations are created, updated, or destroyed.

According to (5), the triplet of transition rules determines the states **S**, locations **L** and neighbors **R** of automata at time t + 1 based on their values at time t. It is very well known that different interpretations of the 'hidden'—time—variable in a discrete system can critically influence model formulation and resulting dynamics (Liu & Andersson 2004). There are several ways to implement time in a dynamic system. On the one hand, we consider time as governed by an *external* clock, which commands simultaneous application of rules (5) to each automaton and at each tick. On the other, each automaton can have its own *internal* clock and, thus, the units of time in (5) can have different meaning for different automata. Formally, these approaches are expressed as *Synchronous* or *Asynchronous* modes of updating of automata states. System dynamics strongly depend on the details of the mode employed (Berec 2002).

10. What is necessary for software implementation of GAS

A software implementation of GAS for urban modeling—Object-Based Environment for Urban Simulation (OBEUS)—is currently in development at the Environment Simulation Laboratory of the University of Tel Aviv, and an initial version was presented at the previous GeoComputation conference (Benenson *et al.* 2001). Recently, OBEUS was modified to include all the basic characteristics of GAS; that was intentionally done in the simplest possible manner. Based on recent OBEUS experience, we specify—below—the main components that are necessary to implement the GAS approach as a software environment.

11. Abstract classes of OBEUS

The basic components of GAS are defined in OBEUS with respect to automata types k $\in$ **K**, its states $\mathbf{S}_k$, location **L**, and neighborhood relations **R** to other objects. These are implemented by means of three abstract root classes (Figure 3):

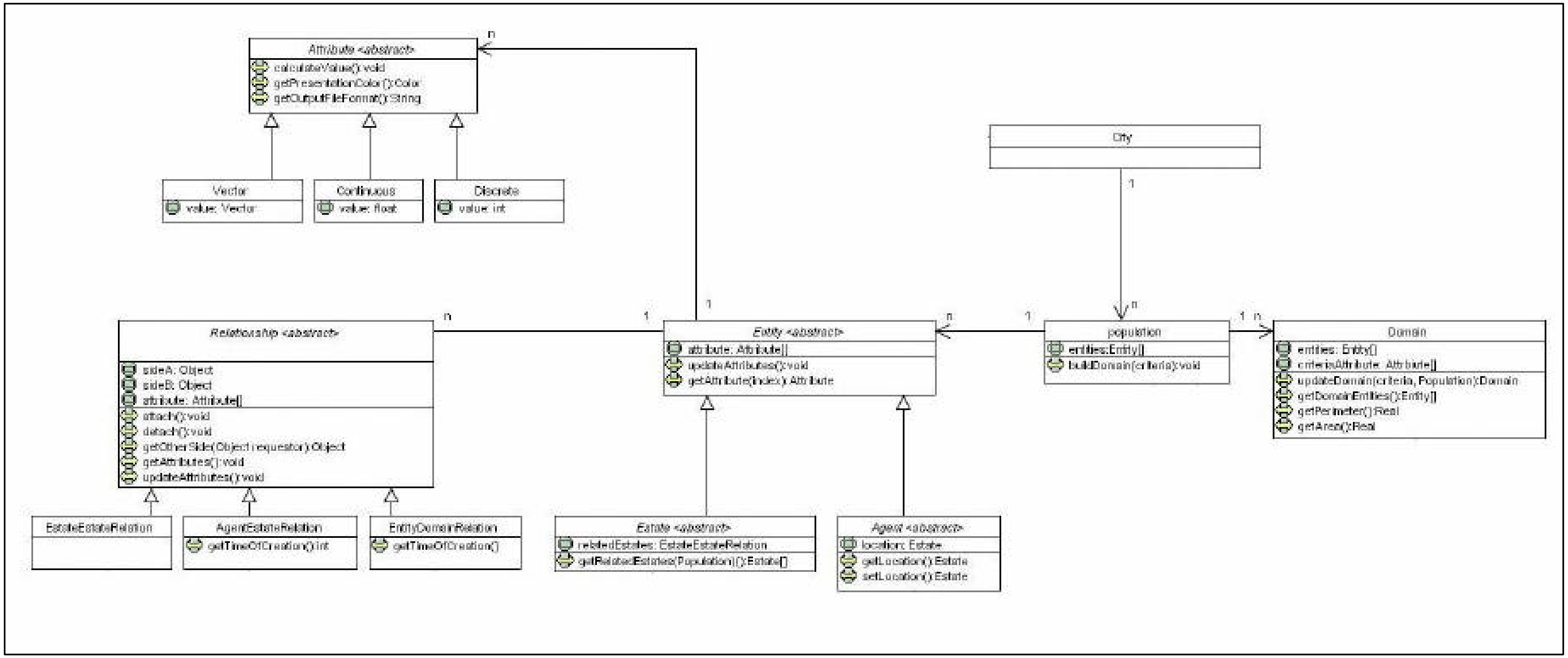

Figure 3. A UML scheme illustrating the abstract-level classes of OBEUS and the example of model-level classes for a residential dynamics simulation

Population, which contains information regarding the population of objects of given type k as a whole; **GeoAutomata** acting as a container for geographic automata of specific type; and **GeoRelationship**, which facilitates specification of (spatial, but not necessarily) relationships between geographic automata. This functionality is available regardless of the degree of neighborhood and other relationships between automata, whether they are one-to-one, one-to-many, or many-to-many. The *location* information for geographic automata essentially depends on whether the object we consider is fixed or non-fixed. This dichotomy is handled using abstract classes, **Estate** and **Agent**. The **Estate** class is used to represent fixed geographic automata (land parcels and properties in a residential context). The **Agent** class represents non-fixed geographic automata (householders and landlords in a residential context). Following from this, three abstract relationship classes can be specified: **EstateEstate**, **AgentEstate**, and **AgentAgent**. The latter is not implemented because the only way of locating non-fixed agents modeled in OBEUS is by pointing to fixed estates; consequently, direct relationships between non-fixed objects are not allowed.

12. Management of time in OBEUS

The OBEUS architecture utilizes both *Synchronous* and *Asynchronous* modes of updating. In *Synchronous* mode, all automata are assumed to change simultaneously and conflicts can arise when agents compete over limited resources, as in the case of two householders trying to occupy the same apartment. Resolution of these conflicts depends on the model's context, a decision OBEUS leaves to the modeler. In *Asynchronous* mode, automata change in sequence, with each observing a geographic reality left by the previous automata. Conflicts between automata are thereby resolved; but the order of updating is critical as it may influence results. OBEUS demands that the modeler sets up an order of automata updating according to a template: randomly, sequence in order of some characteristic, and object-driven approaches are currently being implemented.

13. Management of relationships in OBEUS

Relationships in GAS models can change in time and this might cause conflicts, when, in housing applications, for example, a landlord wants to sell his property, while the tenant does not want to leave the apartment. This example represents the general *problem of consistency* in managing relationships. It has no single general solution; there are plenty of complex examples discussed in the computer science literature (Peckham *et al.* 1995). In OBEUS, we follow the development pattern proposed by Noble (Noble 2000). To maintain a consistency in relationships, an object on one side, termed the *leader*, is responsible for managing the relationship. The other side, the *follower*, is comprised of passive objects. The leader provides an interface for managing the relationship, and invokes the followers when necessary. There is no need to establish leader or follower ‘roles’ in a relationship between fixed objects once the relationship is established, while in relationships between a non-fixed and a fixed object, the non-fixed object is always the leader and is responsible for creating and updating the relationship. For instance, in a relationship between a landlord and her property (when ownership cannot be shared), the landlord initiates the relationship and is able to change it. We do not have evidence that the majority of real-world situations can be imitated using a *leader-follower* pattern, but we are also unaware of cases—in urban contexts—where this pattern would be insufficient.

14. Implementing system theory demands within OBEUS

Systems theory suggests another challenge for automata modeling in which the usefulness of the GAS-OBEUS approach appears to offer advantages. In a systems context, *many* interacting automata are often necessary for capturing the nuances of geographic reality. It is very well known that if system rules are non-linear and the system is open, then emergence and self-maintenance of entities at above-automata levels become feasible. Gentrified areas and commuting zones are examples.

The idea of self-organization is external to GAS and it is not necessary to incorporate it into software implementation. Nonetheless, self-organization is too often important for studying urban systems to be ignored, even at the first step of GAS software implementation. Emerging *spatial ensembles* of geographic automata are supported in OBEUS by means of the abstract class **GeoDomain**. The simplest approach to emergence, determined by the set of *a priori* given predicates defined on geographic automata is implemented; domains are thus limited to capturing 'foreseeable' self-organization of specific types.

As with non-fixed agents, domains are always leaders in their relationships with the automata within them. These relationships can capture properties such as distance between fixed automata and the domain; several definitions of distance based on objects' and domains' centroids, boundaries, etc. can be applied.

15. Conclusions

We have introduced a Geographic Automata Systems framework as a unified scheme for representing discrete geographic systems. Technically, the framework is designed to merge two popular tools used in urban simulation—Cellular Automata and Multi-Agent Systems—and specify them in a patently spatial manner. Conceptually, our assertion is that GAS forms the kernel of the system, as far as the system is spatially driven.

The minimal GAS skeleton allows for a degree of standardization between automata models and GIS. It also provides a mechanism for *transferability*. Until now, the majority of spatial simulations can be investigated only by their developers. The development of GAS software breaches this barrier, offering opportunities to turn urban modeling from art into engineering.

A few additional steps are necessary for full implementation of the GAS framework; none, we think, requires decades of research to realize. The first requirement we have identified relates to transforming the GAS framework into software. As demonstrated with reference to OBEUS, we have advanced along that line of research inquiry, in urban contexts. Development of a (preferably geography-specific) simulation language based on GAS is a second requirement that we consider. The intent, in that context, is to enable the formulation of simulation rules in terms of objects' spatial behavior. We believe that the continued development of simulation languages (Schumacher 2001) that has gathered steam in the last decade, coupled with advances in GI Science and spatial ontology, could answer this requirement in the near future. The third requirement is development of GAS applications. We have developed GAS-based models of housing dynamics (Benenson *et al.* 2002) and urban growth (Torrens 2002b) thus far, and the results are promising.

16. References

Batty, M., H. Couclelis, and M. Eichen. 1997. Special issue: urban systems as cellular automata. *Environment and Planning B.* 24 (2).

Benenson, I. 1998. Multi-agent simulations of residential dynamics in the city. *Computers, Environment and Urban Systems.* 22 (1):25-42.

———. 1999. Modelling population dynamics in the city: from a regional to a multi-agent approach. *Discrete Dynamics in Nature and Society.* 3:149-170.

Benenson, I., S. Aronovich, and S. Noam. 2001. OBEUS: Object-Based Environment for Urban Simulations. In *Proceedings of the Sixth International Conference on GeoComputation*, edited by D. V. Pullar. Brisbane: University of Queensland, GeoComputation CD-ROM. (Available at http://www.geocomputation.org/2001/papers/benenson.pdf.)

Benenson, I., and I. Omer. 2000. Multi-scale approach to measuring residential segregation and the case of Yaffo, Tel-Aviv. In *Proceedings of the Fifth Annual Conference on GeoComputation*, edited by R. Abrahart. Manchester: GeoComputation CD-ROM. (Available at http://www.geocomputation.org/2000/GC047/Gc047.htm.)

Benenson, I., I. Omer, and E. Hatna. 2002. Entity-based modeling of urban residential dynamics: the case of Yaffo, Tel Aviv. *Environment and Planning B: Planning and Design.* 29:491- 512.

Benenson, I., and P.M. Torrens. 2004a. Geosimulation: object-based modeling of urban phenomena. *Computers, Environment and Urban Systems.* 28 (1/2): forthcoming.

———. 2004b. Special Issue: Geosimulation: object-based modeling of urban phenomena. *Computers, Environment and Urban Systems.* 28 (1/2): forthcoming.

Berec, L. 2002. Techniques of spatially explicit individual-based models: construction, simulation, and mean-field analysis. *Ecological Modelling.* 150:55-81.

Box, P. 2001. Spatial Units as Agents: Making the Landscape an Equal Player in Agent-Based Simulations. In *Integrating Geographic Information Systems and Agent-Based Modeling Techniques for Simulating Social and Ecological Processes,* edited by H. R. Gimblett. Oxford: Oxford University Press 59-83.

Brown, J.S., and P. Duguid. 2000. *The Social Life of Information*. Boston: Harvard Business School Press.

Chapin, F.S., and S.F. Weiss. 1968. A Probabilistic Model for Residential Growth. *Transportation Research.* 2:375-390.

Couclelis, H. 1985. Cellular worlds: A framework for modeling micro-macro dynamics. *Environment and Planning A.* 17:585-596.

Dijkstra, J., H.J.P. Timmermans, and A.J. Jessurun. 2000. A multi-agent cellular automata system for visualising simulated pedestrian activity. In *Theoretical and Practical Issues on Cellular Automata,* edited by S. Bandini and T. Worsch. London: Springer-Verlag 29-36.

Epstein, J.M. 1999. Agent-based computational models and generative social science. *Complexity.* 4 (5):41-60.

Epstein, J.M., and R. Axtell. 1996. *Growing Artificial Societies from the Bottom Up*. Washington D.C.: Brookings Institution.

Gimblett, H.R., ed. 2002. *Integrating Geographic Information Systems and Agent-Based Modeling Techniques for Simulating Social and Ecological Processes*, *Santa Fe Institute Studies in the Sciences of Complexity*. Oxford: Oxford University Press.

Kohler, T.A. 2000. Putting social sciences together again: an introduction to the volume. In *Dynamics in Human and Primate Societies*, edited by T. A. Kohler and G. Gumerman. New York: Oxford University Press 1-18.

Liu, X.-H., and C. Andersson. 2004. Assessing the impact of temporal dynamics on land-use change modeling. *Computers, Environment and Urban Systems.* 28 (1/2): forthcoming.

Meyer, J.-A., A. Berthoz, D. Floreano, H.L. Roitblat, and S.W. Wilson, eds. 2000. *From Animals to Animats 6: Proceedings of the Sixth International Conference on Simulation of Adaptive Behavior*. Cambridge, MA: MIT Press.

Nakajima, T. 1977. Application de la théorie de l'automate à la simulation de l'évolution de l'espace urbain. In *Congrès Sur La Méthodologie De L'Aménagement Et Du Dévelopment*. Montreal: Association Canadienne-Française Pour L'Avancement Des Sciences et Comité De Coordination Des Centres De Recherches En Aménagement, Développement Et Planification (CRADEP) 154-160.

Noble, J. 2000. Basic relationship patterns. In *Pattern Languages of Program Design 4*, edited by N. Harrison, B. Foote and H. Rohnert. New York: Addison-Wesley.

O'Sullivan, D. 2001. Exploring spatial process dynamics using irregular cellular automaton models. *Geographical Analysis.* 33 (1):1-18.

O'Sullivan, D., and P.M. Torrens. 2000. Cellular models of urban systems. In *Theoretical and Practical Issues on Cellular Automata*, edited by S. Bandini and T. Worsch. London: Springer-Verlag 108-117.

Peckham, J., B. MacKellar, and M. Doherty. 1995. Data models for extensible support of explicit relationships in design databases. *VLDB Journal.* 4:157-191.

Phipps, M. 1989. Dynamic behavior of cellular automata under the constraint of neighborhood coherence. *Geographical Analysis.* 21:197-215.

Polhill, J.G., N.M. Gotts, and A.N.R. Law. 2001. Imitative and non-imitative strategies in a land use simulation. *Cybernetics and Systems.* 32:285-307.

Portugali, J. 2000. *Self-Organization and the City*. Berlin: Springer-Verlag.

Portugali, J., I. Benenson, and I. Omer. 1997. Spatial cognitive dissonance and sociospatial emergence in a self-organizing city. *Environment and Planning B.* 24:263-285.

Reynolds, C. 1987. Flocks, herds, and schools: A distributed behavioral model. *Computer Graphics.* 21 (4):25-34.

———. 1999. Steering behaviors for autonomous characters. Paper read at Game Developers Conference, San Jose, CA. (Available at http://www.red3d.com/cwr/papers/1999/gdc99steer.pdf.)

Sakoda, J.M. 1971. The checkerboard model of social interaction. *Journal of Mathematical Sociology.* 1:119-132.

Sanders, L., D. Pumain, H. Mathian, F. Guérin-Pace, and S. Bura. 1997. SIMPOP: A multiagent system for the study of urbanism. *Environment and Planning B.* 24:287-305.

Schelling, T.C. 1969. Models of segregation. *American Economic Review.* 59 (2):488-493.

———. 1971. Dynamic models of segregation. *Journal of Mathematical Sociology.* 1:143-186.

———. 1974. On the ecology of micro-motives. In *The Corporate Society*, edited by R. Marris. London: Macmillan 19-55.

———. 1978. *Micromotives and Macrobehavior*. New York: WW Norton and Company.

Schofisch, B., and K.P. Hadeler. 1996. Dimer automata and cellular automata. *Physica D.* 94:188-204.

Schumacher, M. 2001. *Objective Coordination in Multi-Agent System Engineering*. Berlin: Springer.

Semboloni, F. 2000. The growth of an urban cluster into a dynamic self-modifying spatial pattern. *Environment and Planning B: Planning & Design.* 27 (4):549-564.

Shi, W., and M.Y.C. Pang. 2000. Development of Voronoi-based cellular automata--an integrated dynamic model for Geographical Information Systems. *International Journal of Geographical Information Science.* 14 (5):455-474.

Tobler, W. 1970. A computer movie simulating urban growth in the Detroit region. *Economic Geography.* 46 (2):234-240.

———. 1979. Cellular Geography. In *Philosophy in Geography*, edited by S. Gale and G. Ollson. Dordrecht: Kluwer 379-386.

Torrens, P.M. 2001. *New tools for simulating housing choices.* Program on Housing and Urban Policy Conference Paper Series C01-006. Berkeley, CA: University of California Institute of Business and Economic Research and Fisher Center for Real Estate and Urban Economics. (Available at http://urbanpolicy.berkeley.edu/pdf/torrens.pdf.)

———. 2002a. Cellular automata and multi-agent systems as planning support tools. In *Planning Support Systems in Practice*, edited by S. Geertman and J. Stillwell. London: Springer-Verlag 205-222.

———. 2002b. SprawlSim: modeling sprawling urban growth using automata-based models. In *Agent-Based Models of Land-Use/Land-Cover Change*, edited by D. C. Parker, T. Berger, S. M. Manson and W. J. McConnell. Louvain-la-Neuve, Belgium: LUCC International Project Office 69-76.

———. 2003a. Automata-based models of urban systems. In *Advanced Spatial Analysis*, edited by P. A. Longley and M. Batty. Redlands, CA: ESRI Press.

———. 2003b. Geosimulation approaches to travel modeling. In *Transport Geography and Spatial Systems*, edited by D. Hensher. London: Pergamon.

———. 2004. Looking forward: remote sensing as dataware for human settlement simulation. In *Remote Sensing of Human Settlements*, edited by M. Ridd. New York: John Wiley and Sons, forthcoming.

Torrens, P.M., and D. O'Sullivan. 2000. Cities, cells, and cellular automata: Developing a research agenda for urban geocomputation. In *Proceedings of the Fifth Annual Conference on GeoComputation*, edited by R. Abrahart. Manchester: GeoComputation CD-ROM. (Available at http://www.geocomputation.org/2000/GC044/Gc044.htm.)

———. 2001. Cellular automata and urban simulation: where do we go from here? *Environment and Planning B.* 28 (2):163-168.

Wahle, J., L. Neubert, J. Esser, and M. Schreckenberg. 2001. A cellular automaton traffic flow model for online simulation of traffic. *Parallel Computing.* 27:719-735.

White, R., G. Engelen, and I. Uljee. 1997. The use of constrained cellular automata for high-resolution modelling of urban land use dynamics. *Environment and Planning B.* 24:323-343.