International Journal of Geographical Information Science
Vol. 19, No. 4, April 2005, 385–412

Research Article

Geographic Automata Systems

PAUL M. TORRENS†* and ITZHAK BENENSON‡
†Department of Geography, University of Utah, USA
‡Department of Geography and Human Environment, and Environment Simulation Laboratory, Porter School of Environmental Studies, University Tel Aviv

(*Received 22 September 2003; Accepted 8 July 2004*)

A novel approach to automata-based modeling for spatial systems is described: geographic automata and Geographic Automata Systems. We detail a framework that takes advantage of the formalism of automata theory and GI Science to unite cellular automata and multi-agent systems techniques, and provides a spatial approach to bottom-up modeling of complex geographic systems that are comprised of infrastructure and human objects. The suitability of the framework is also discussed with reference to existing cellular automata and multi-agent systems models used in urban studies. Practical implementation of the framework is illustrated with reference to an object-based urban simulation environment and implementation of a popular socio-spatial segregation model.

Keywords: Geographic Automata Systems; cellular automata; multi-agent systems; Geographic Information Science; geosimulation; Geographic Information Systems

1. Introduction

A new class of simulation models has come to the fore in recent years, supported by an array of advances and developments both in the geographical sciences and in fields traditionally understood to be relatively external to geographic inquiry (Benenson and Torrens 2004a). These models are more commonly based on cellular automata (CA) or multi-agent system (MAS) formalisms and are often applied to the simulation of spatial systems in dynamic and high-resolution contexts.

Applied in isolation, CA and MAS approaches have been used to simulate a wide variety of phenomena, particularly urban phenomena (Benenson and Torrens 2004a, Benenson and Torrens 2004b). When used in a combined framework, CA and MAS open numerous avenues for exploratory and applied simulation in geography. Nevertheless, the amalgamation of CA and MAS tools for spatial simulation necessitates certain awkward methodological compromises and most combined CA-MAS computer environments and applications exploit a strict CA view of the geographic systems that they model. While these frameworks are certainly useful, they are based on pragmatic rather than theoretical considerations: the combination of CA and MAS in this manner is a function of the limitations of the available tools rather than being informed by knowledge or theory regarding how real systems function in space.

In what follows, we offer a genuinely *spatial* framework for modeling geographic systems, one formulated on the basis of *objects located in space*. We specify

*Corresponding author. Email: torrens@geosimulation.com

International Journal of Geographical Information Science
ISSN 1365-8816 print/ISSN 1362-3087 online © 2005 Taylor & Francis Group Ltd
http://www.tandf.co.uk/journals
DOI: 10.1080/13658810512331325139

simulated geographic objects as *geographic automata* that combine CA and MAS concepts in unique ways, by considering collections of interacting geographic objects as *Geographic Automata Systems* (GAS). In this framework, geographic phenomena as a whole are considered as the outcomes of the collective dynamics of multiple animate and inanimate geographic automata. We argue that the GAS framework serves as an intuitive basis for merging CA and MAS (which are popularly confused in the geographic literature), with *geography* as the binding force. In the framework, automata become uniquely geographical, fusing CA and MAS but extending the concept to incorporate notions from GIS and GI Science. We demonstrate the operational use of the framework with reference to the development of software for simulating the space-time dynamics of urban systems and implementation of Schelling's classic model of urban segregation.

2. Cellular automata and multi-agent systems as automata systems

2.1 *The basic automata framework*

Automata have many uses, among them the imitation of living organisms and lifeless elements of the environment. There are various kinds of automata, each with its own formal definition. Simply stated, an automaton is a discrete processing mechanism, characterized by internal states. An automaton 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. In this way, automata have the capacity to process information from their surroundings and to alter their characteristics accordingly. They are flexible and efficient abstractions that enable the construction of detailed, complex, dynamic models; they are also well suited to handling geographic phenomena.

Formally, an automaton, A, can be represented by means of a set of *states* $\mathbf{S}$ and a set of *transition rules* $\mathbf{T}$.

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

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

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

This basic formulation does not define the nature of the states $S \in \mathbf{S}$, or possible inputs $I \in \mathbf{I}$. The essence of the automata approach is in temporal discreteness and the ability of an automaton to change according to predetermined rules based on internal ($\mathbf{S}$) and external ($\mathbf{I}$) information.

Regarding urban applications, nothing prevents us from considering the entire city as an automaton with myriad states and transition rules (Torrens 2002). However, to make sense, an individual automaton should be as simple as possible in terms of states, transition rules, and input information (Torrens and O'Sullivan 2001). Simple structure is characteristic of the automata systems applicable in urban geography, Cellular Automata (CA) and Multi-Agent Systems (MAS).

2.2 *Cellular Automata*

CA are arrangements of individual automata over tessellated space, where an individual automaton is influenced by automata in neighboring cells. Usually, partitioning is regular, with cells defined by a square or hexagonal grid and 'neighboring' refers to adjacency. We can refine general definition (1)–(2) to specify an automaton, A, belonging to a CA lattice as follows:

$$\mathrm{A} \sim (\mathbf{S}, \mathbf{T}, \mathbf{N}) \tag{3}$$

where **N** denotes automata neighboring A and defines the set of cells for drawing input information **I**, which is necessary for the application of transition rules **T**.

In basic CA, *neighborhoods* have identical form for each automaton, e.g., Moore or von Neumann; it is also supposed that input information is gleaned only from an automaton's neighborhood (figure 1). The assumption of regularity is largely

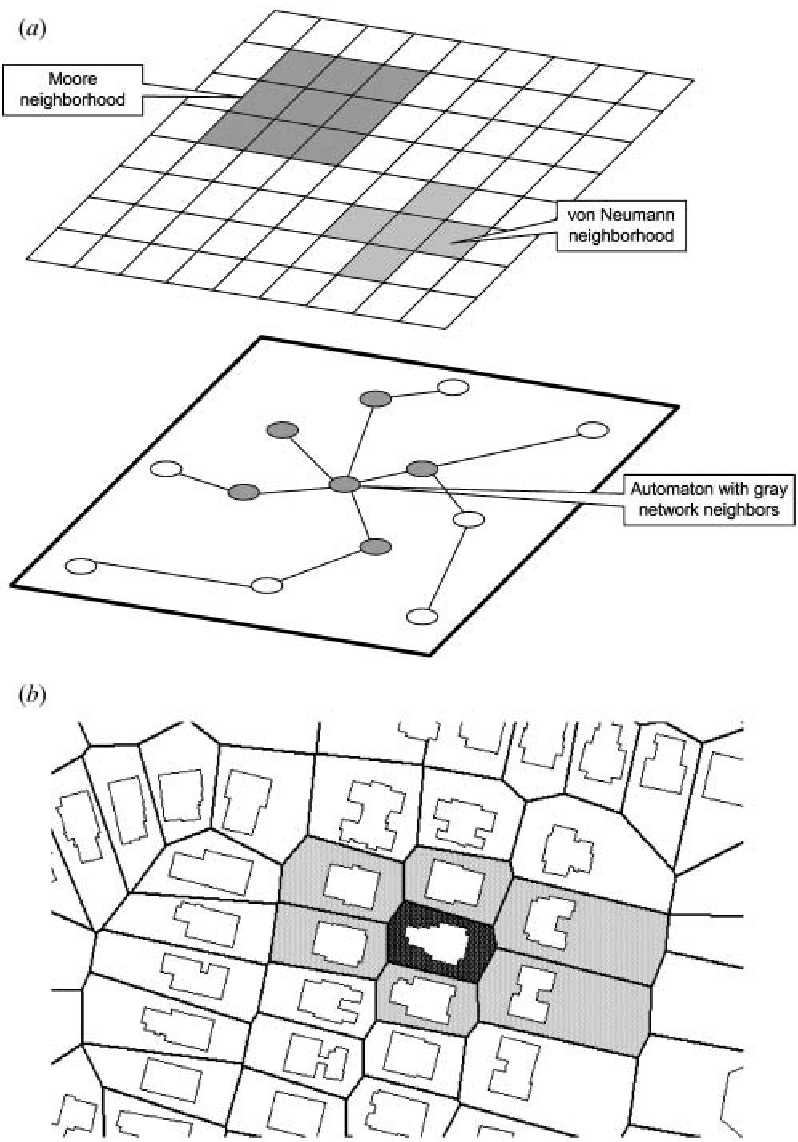


Figure 1. (*a*) Grid and network neighborhoods. (*b*) Voronoi neighborhood (gray), based on property coverage.

superficial, however, and CA have been implemented on a variety of non-regular tessellations: arbitrary networks, irregular partitions given by GIS-based coverage of land parcels, and Voronoi tessellations (Benenson *et al.* 2002, O'Sullivan 2001, Semboloni 2000, Shi and Pang 2000). In this case, the form of the neighborhood and the number of neighbors varies between automata of the CA. An assortment of definitions of neighborhoods, based on connectivity, adjacency, or distance can be applied to these generalized CA.

The weakness of the CA approach is the inability of automata cells to move. Despite repeated attempts to interpret units' mobility (Benenson 1998, Schofisch and 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.

2.3 *Multi-agent systems*

Agents are also automata and, thus, incorporate all of the features of basic automata that have just been discussed. However, there are some important distinctions between CA and MAS, particularly when agents are specified with mobility, which is the common interpretation in geographic models.

According to (3), CA are capable of diffusing state information to neighboring automata; however, the individual cells of CA remain fixed in their simulated spaces; they cannot change location. Mobile agents transmit information by themselves, moving to another location, which can be at any distance from an agent's current position. Agents' spatial behavior can manifest more complex forms than simple relocation. For example, landlord agents might perform spatially-mediated sale and purchasing of real estate; the spatial behavior of agents designed to represent car drivers could include the choice of links and turning opportunities at junctions.

Generally, agent automata employed in social science research (Epstein 1999, Kohler 2000) are used to represent individual decision-makers (or, sometimes, groups of decision-makers). Consequently, the states $\mathbf{S}$ that are attributed to social-science agents are usually designed to represent socioeconomic characteristics, and agent transition rules $\mathbf{T}$ commonly correspond to human-like *behaviors*. For the most part, however, work in agent-based simulation in the social sciences outside geography is *non-spatial* in nature, as are the tools that are used. Many of the decisions and behaviors of *geographic agents* are spatial in nature, and this distinguishes agent tools necessary for geographic applications.

3. A rationale for *Geographic* automata systems

Despite their potential for urban simulation, CA and MAS are limited in their geographic functionality when considered in isolation. The limitations stem from the disjunction between the tools and our understanding of the spatial dynamics of the systems they are used to simulate. In many cases, the simulated entities represented by CA and MAS models do not behave as we understand they should, largely because the modeling framework will not permit them to.

General automata are characterized by states **S** and transition rules **T** that govern the change of states. Intuitively, we might identify several internal characteristics that are essential to geographic automata:

- A typology or ontology of automata based on the ability to relocate;
- The space in which they are situated;
- The spatial relationships between automata;
- The processes governing the changes of automata location in space.

Simulating geographical systems, then, involves explicit formulation of all these four components.

Neither CA nor MAS can fully provide these requirements in isolation. The geography of the CA framework is evidently restricted because CA are incapable of representing autonomously mobile entities. Individual automata in CA can *diffuse* the information encapsulated in their cell states over automata neighborhoods, but the individual cells themselves are not free to move (Torrens 2004a). At the same time, MAS methodologies and existing tools are over-general and underestimate, if not ignore, the importance of space and spatial behavior.

A truly geographic framework for automata models remains sought after. Despite the widely acknowledged suitability of automata tools for geographic modeling (Gimblett 2002), there has been relatively little exploration into the development of patently *spatial* automata tools for urban simulation. This is partly due to the history of automata development; automata were pioneered as computing media (Ifrah 2000, Wiener 1961), originally used for the description of networks of units influencing each other by means of signals transferred along links. These networks were used as an abstraction of several phenomena: universal computational devices, neural networks, the human brain, cellular tissue, ecological webs, etc. Interest in CA remained obscure for two decades after these initial developments, until revived by the popularity of John Horton Conway's Game of Life (Gardner 1970, 1971), as well as many applications in physics, chemistry, biology, and ecology (Wolfram 2002).

However, the introduction of automata tools in geography is a relatively recent phenomenon. It took geography, as a discipline, a further twenty years to adopt the concept. Despite direct analogies between land parcels and cells on the one hand and land-uses and cell states on the other, geographical applications of CA models were few and far between (Benenson and Torrens 2004a). A few sound examples published in the 1970s (Chapin and Weiss 1968, Nakajima 1977, Tobler 1979) were nonetheless ignored, before interest was revived in the 1980s (Couclelis 1985, Phipps 1989). However, it was not until the 1990s that CA modeling became a widespread research activity in geography, popularized by applications in urban geography (Batty *et al.* 1997).

The study of MAS has taken place much more recently than that of CA. 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 late-1990s (Benenson 1999, Dijkstra *et al.* 2000, Portugali *et al.* 1997, Sanders *et al.* 1997), following its introduction in ecology and economics (De Angelis and Gross 1992, Tesfatsion 1997). Until recently, the mainstream of MAS research in geography involved populating regular CA with agents of one or several kinds, which could migrate between cells, or simply *reinterpreting* CA as agent-based models, by attributing anthropomorphic state variables to cells. Often, it is assumed that agents' migration behavior depends on the properties of neighboring cells and neighbors (Epstein and Axtell 1996, Portugali 2000). Very recent explicit agent-based models 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). These models clearly demonstrate the potential of MAS for modeling the intricacies of human spatial behavior. But, we argue, the work can be taken much further.

As a spatial science, geography concerns itself with the behavior and distribution of *objects in space*. In dynamic spatial systems, many of these objects change their properties and/or location; the goal of a geographic model is to mimic these activities and their consequences, often at multiple scales. In what follows, we present a framework that aims to infuse spatial properties into automata tools, based on the assumption that geographic objects—agents and features—are all individual automata and, as characteristic of automata, their rules of behavior can be defined *a priori*, with focus on their spatial properties and behaviors. A geographic system can be thus modeled as a collection of geographic automata, as a *Geographic Automata System*.

4. Geographic Automata Systems

A Geographic Automata System (GAS) consists of geographic automata of various types or ontology. In general, automata are characterized by states and transition rules. In the case of geographic automata, we introduce functionality to enable the explicit consideration of *space* and *spatial behavior*. In CA, pre-defined partitions are often used as a *proxy* for geography. The approach that we adopt with GAS differs; instead of pre-defined partitions, we introduce a set of *geo-referencing rules* for situating geographic automata in space. Likewise, we define *neighborhood rules*, rather than relying on fixed neighborhood patterns that are incapable of being varied in space or time once delineated. Considering the mobility introduced by the agent-based paradigm, we define *movement rules* that allow for the navigation of geographic automata. Formally, a Geographic Automata System, G, consists of seven components:

$$\mathrm{G} \sim (\mathbf{K};\ \mathbf{S}, \mathbf{T_S};\ \mathbf{L}, \mathbf{M_L};\ \mathbf{N}, \mathbf{R_N}) \tag{4}$$

Here, $\mathbf{K}$ denotes a set of *types* of automata featured in the GAS and three pairs of symbols denote a specific spatial mechanism and the rules that determine its dynamics.

The first pair denotes general automata features. It represents a set of *states* $\mathbf{S}$, associated with G, and a set of state transition rules $\mathbf{T_S}$, used to determine how automata states should change over time. $\mathbf{S}$ consists of subsets of states $\mathbf{S^k}$, characteristic of automata of each type $\mathbf{k} \in \mathbf{K}$. The second pair represents location information. $\mathbf{L}$ 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 in time. According to general definition (1)–(2), state and location transitions depend on automata themselves and on input (I), given by the states of neighbors. The third pair in (4) specifies this condition. $\mathbf{N}$ represents the neighbors of the automata and $\mathbf{R_N}$ represents the rules that govern changes of automata relations to the other automata.

The general automation process (2) can be specified in terms of the framework laid out above. Let us consider a geographic automaton G at time t with state S_t; it is located at L_t, and its external input, I_t, is defined by its neighbors N_t. To animate, or spatially enable GAS, state transition, movement, and neighborhood rules—$\mathbf{T_S}$, $\mathbf{M_L}$, and $\mathbf{R_N}$—should be applied to each G, and this results in a new triplet: S_{t+1}, L_{t+1}, and N_{t+1}:

$$\begin{aligned} \mathbf{T_S}&: (S_t, L_t, N_t) \rightarrow S_{t+1} \\ \mathbf{M_L}&: (S_t, L_t, N_t) \rightarrow L_{t+1} \\ \mathbf{R_N}&: (S_t, L_t, N_t) \rightarrow N_{t+1} \end{aligned} \tag{5}$$

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 representing *spatially enabled* interactive behavior of elementary geographic objects in a system. Let us specify GAS components further.

4.1 *Geographic automata types,* $\boldsymbol{K}$

As mentioned, GAS may be composed of automata of different types. At an abstract level, we can 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. For example, in the context of urban systems, a variety of infrastructure objects may be specified as fixed geographic automata: road links, building footprints, parks, households, etc. Fixed geographic automata may be subject to any of the transition rules outlined already, except rules of motion $\mathbf{M_L}$.

Non-fixed geographic automata symbolize entities that change their location over time. The full range of rules for GAS can be applied to non-fixed geographic automata, including movement rules. Typical examples of non-fixed urban automata include pedestrians, vehicles, householders, landlords, etc.

A geographic system usually contains objects of both fixed and non-fixed types. For example, in a model of housing dynamics, apartments and houses might be represented by fixed geographic automata, with state variables describing their characteristics that are important for residential choice: number of rooms, floor level, value, architectural style, the year of establishment, presence of elevators, etc. Non-fixed geographic automata in a housing context might represent householders, with state variables including economic status of a family, mean age of parents, and number of children. In the case of cars as non-fixed automata, state variables of relevance to the movement rules of the GAS may include heading, speed, progress toward destination, etc. (Torrens 2004a).

4.2 *Geographic automata states and state transition rules,* $\boldsymbol{S}$ *and* $\boldsymbol{T_S}$

The characteristics of fixed and non-fixed automata often depend on each other. For example, the value of an apartment depends on real estate in the property and property's neighborhood and on the structure and neighborhood population. Consequently, a transition rule, $T_{\text{propertyvalue\_update}} \in \mathbf{T_S}$ that describes the change in value of real estate should depend on the states of the fixed automata representing real estate and, importantly, on the states of non-fixed automata representing householders that occupy them. In the same manner, a transition rule $T_{\text{househodereconomicstate\_update}} \in \mathbf{T_S}$, describing changes in the economic status of householders, a rule $M_{\text{householder\_relocate}} \in \mathbf{M_L}$, describing the way householders choose new residence, and a rule $R_{\text{householderneighbors\_update}} \in \mathbf{R_N}$, describing how householders' neighbors are determined depend on the states of the real estate and on attributes of the householders that occupy it.

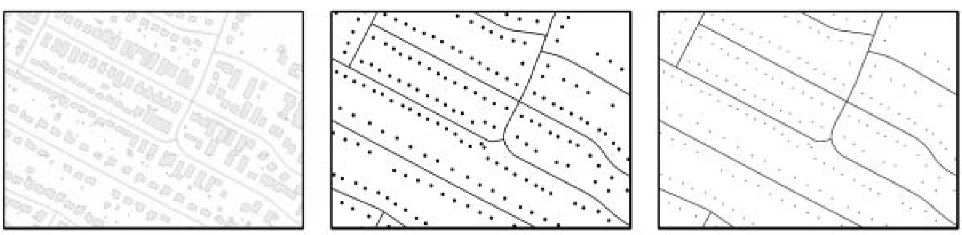
It is worth mentioning that, in the context of the GAS framework, CA are artificially closed, simply because cell-state transition rules are generally driven only by cells. In GAS, infrastructure objects' transition is governed by other relevant objects acting upon and within them. This is a crucial concept for simulating *human-driven* systems, in which people interact and are affected by their environments.

4.3 *Geo-referencing conventions and migration rules, $\mathbf{L}$ and $\mathbf{M_L}$*

Geo-referencing conventions $\mathbf{L}$ govern how geographic automata are registered in space. Fixed geographic automata can be geo-referenced by recording their position coordinates, which do not change in time. Non-fixed geographic automata may move and this demands specific conventions regarding $\mathbf{L}$. It is also worth noting that there are instances in which geo-referencing is dynamic for the geographic automata that represent infrastructure objects, for example, when land parcel objects are sub-divided during simulation.

Formally, we say that automata in a GAS can be geo-referenced to simulated spaces *directly* and *indirectly*.

Direct geo-referencing follows a vector GIS approach, using coordinate lists. Such a list indicates all spatial details necessary to represent automata as a spatial object: boundaries, centroids, node's location, etc. Fixed geographic automata are located by means of direct geo-referencing. The details of the particular rules depend on the automata employed in a modeling exercise. For typical urban objects such as street segments or buildings, polylines, 2D basement polygons, or 3D prisms may be used to register objects in space. Varying resolutions may be employed, depending on the model application. For example, when modeling housing dynamics at a 'micro-scopic' scale, building footprints, outer borders, and road segments may be required (figure 2a). However, in other cases, this amount of detail may not be needed and the centroid of a building and centerlines of road segments may be enough to register



(a) (b) (c)

Figure 2. Direct geo-referencing. (a) Buildings are represented by means of foundation contours; road segments by means of road boundaries, (b) Buildings are represented by means of foundation centroids; road segments by means of a road segment centreline, (c) Building centroids and roads are represented by cells.

automata in the model (Figure 2b). In abstract models, cell-based approximations may suffice (Figure 2c).

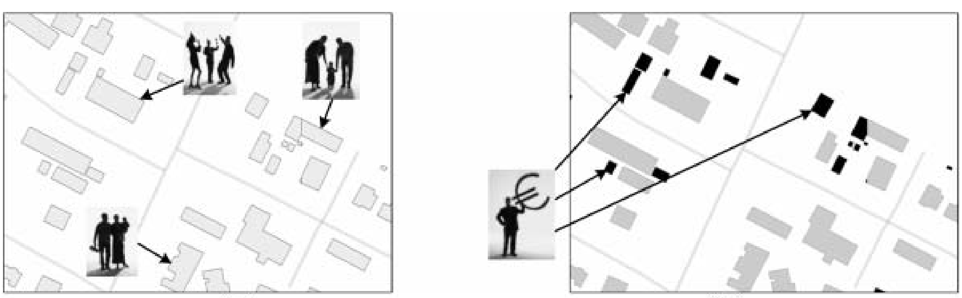
The second method by which geographic automata might be geo-referenced, indirectly, is by *pointing* to other automata. For example, in the instance of a model of property dynamics we can geo-reference householders by address (Figure 3a). Landlords provide a more complex example: they can be geo-referenced by their home addresses, while pointing to the properties that they own might be more important for modeling urban dynamics (Figure 3b). Indirect referencing is mostly relevant for non-fixed geographic automata, but can also be used for fixed geographic automata, e.g., for apartments in a building. It is convenient for dynamic modeling, with references varying as a simulation evolves.

Different formulations of $\mathbf{M_L}$ offer great potential for encoding the motion of traveling objects—vehicles, pedestrians, householders, institutions, etc. They can be based on repel-attract-synchronize interactions between neighbors, as developed, for example, in Animat research (Meyer *et al.* 2000) and in the gaming industry (Reynolds 1999) or, in traffic models, through the specification of rules for collision avoidance, obstacle negotiation, lane-changing, flocking, behavior at junctions, etc. (Torrens 2004a). At the same time, $\mathbf{M_L}$ can code the changes in relationships to locations, such as ownership following spatially-mediated sale and purchasing of real estate.

4.4 *Neighbors and neighborhood rules, $\mathbf{N}$ and $\mathbf{R_N}$*

Neighbors $\mathbf{N}$ for fixed geographic objects are relatively easy to define, simply because the objects are static in space. There is a variety of *geographical* ways to do that—via adjacency of the units in regular or irregular tessellations, connectivity of network nodes, proximity, or via human-like measures such as accessibility or visibility.

Neighborhood rules $\mathbf{R_N}$ account for variation of spatial relationships between geographic automata in *time*. In an urban context, it might be important for fixed objects, when, for example, the neighborhood relationship between buildings is based on accessibility, and could be established, for example, if a path opened up between two buildings, or connected an existing building to a new construction. However, non-fixed automata pose a real challenge, because their neighborhood relations are *usually* dynamic in space and time.



(a) (b)

Figure 3. Indirect geo-referencing by pointing. (a) Locating households by pointing to the houses they occupy, (b) Locating a landowner by pointing to its properties.

Straightforward definitions, via distance and nearest-neighbor relations, as used in Boids models (Reynolds 1987), can be evidently employed, but can become very heavy computationally for GAS containing many automata or when more complex measures of proximity, such as visibility or accessibility, are involved. In this case, when geo-referencing rules **L** are based on indirect location, GAS provide a general solution—two indirectly located automata can be considered as neighbors, when *the automata they point to are neighbors*. For example, in figure 4, two households are established as neighbors by assessing the neighborhood relationship between the houses in which they reside.

The idea of GAS is not a panacea, but a conceptual framework aimed at unified bottom-up description of geographic reality as we see it. We have reviewed the available urban modeling literature and our review demonstrates that each of the examined models can be easily reformulated as a GAS, with transition rules formulated in the form of (4)–(5), and we take this as a promising sign (see Table 2 at the end of the paper). The predominant absence of non-fixed automata, and general lack of location rules in the models documented in Table 2 is worth noting.

5. Geographic Automata Systems as a dynamic extension of Geographic Information Systems

5.1 *Geographic Automata Systems and vector and raster models*

Geographic automata models are tightly bound to vector GIS. First, geographic automata of many types correspond to GIS features, which can be used to derive automata location, and for fixed geographic automata, it is done directly, by using

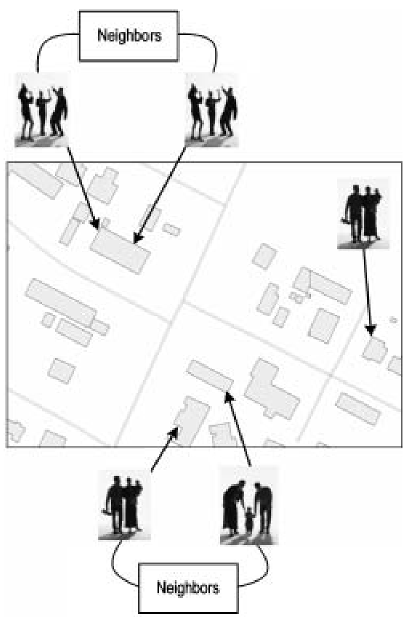


Figure 4. Neighbor relationships for indirectly located geographic automata. Two households are neighbors if they are located in the same property or in neighboring properties.

the coordinate representation of a corresponding GIS feature. Second, the majority of relationships between geographic automata can be naturally evaluated within vector GIS: standard overlay operators such as point-in-polygon, buffering, intersection, etc., make it possible to determine how automata are situated in relation to other automata. More specifically, neighborhood rules are readily available for evaluating adjacency, contiguity, continuity, distance, accessibility, visibility, and so on.

In many ways, GAS move far beyond vector GIS, just as CA models go far beyond raster GIS. First, this relates to GAS object types—GIS features do not cover all the variety of geographic automata types. Landlords, as part of housing GAS, are a typical example of this kind: their own location might not be important for the model, while the location of the real estate that they possess really is. Consequently, the location of landlords in such a housing model should be implemented by pointing to real estate holdings. Second, GAS are dynamic, while GIS are not. The essence of GAS is in the rules of state, location, and neighborhood transitions—$\mathbf{T_S}$, $\mathbf{M_L}$, $\mathbf{R_N}$—which do not have analogies in GIS. GIS would benefit from the introduction of automata-like functionality.

While GAS have obvious affiliations with vector GIS; they are also functionally connected to raster GIS. Each pixel of a raster layer can be regarded, at least morphologically, as an automata cell, geo-referenced by column and row positions within a GIS scene. Based on these coordinates, one can easily consider a point or square representing the cell as a feature of vector GIS, and the latter fully enable vector GIS functionality, including estimation of relations between objects. Nonetheless, the conceptual difference between features originating from cells of raster and features of vector GIS layers still remains; the latter are normally chosen to represent real-world objects, while the former are not. The choice of a raster or vector view is beyond the GAS scheme and evidently depends on the goal of a model. Irregular tessellations based on land partition fit naturally into real-world simulations, while raster representations essentially simplify neighborhood definitions and might be chosen for abstract modeling.

5.2 *Interfacing Geographic Automata Systems with GIS and GIS data*

Information collection is now much more pervasive than before (Brown and Duguid 2000), and high-resolution spatial databases for land-use, population, real estate, and transport are now widespread (Torrens 2004b). Automated procedures for data collection—multi-channel remote sensing, aerial photography, etc.—have provided new information at fine resolutions, both spatial and temporal with added-value created by interpreting these data. New databases and the abilities of GIS to register data spatially, to use spatial analysis to shape data as layers of objects, and to estimate relationships between them (Benenson and Torrens 2004b), provide an extraordinary foundation for GAS modeling framework.

Recent advances in GIS technology also guarantee a functional GIS background for potential GAS computational environments. A number of GIS libraries can be interfaced with other software through the Component Object Model (COM) (Microsoft Corporation and Digital Equipment Corporation 1995, Ungerer and Goodchild 2002) or technologies such as JavaBeans™ (Sun Microsystems 2002). Indeed there are opportunities to extend regular vector GIS, especially open source GIS (Baylor University 2002, Centre for Computational Geography 2002) toward GAS; recent raster GIS extensions toward CA provide a proof-of-concept for such development (Clark Labs 2002). Finally, GIS are an excellent tool for visualizing, and querying, the outcomes of GAS simulations.

6. From GAS to Dynamic Spatial Systems

So far, we have focused on two notions merged within the GAS abbreviation, namely on 'Geographic' and 'Automata.' The third one—'System'—comes into focus when *dynamics* of a given GAS model are considered. The GAS approach provides specification of geographic systems, but does not impose limitations on the ways the update rules $\mathbf{T_S}$, $\mathbf{M_L}$, $\mathbf{R_N}$ are formulated, or how the dynamics of the GAS as a complex system are represented, investigated, and understood. System theory provides a basis for studying GAS models, the latter should unambiguously outline the system concepts it assimilates, just as each CA, MAS, or any other dynamic model should. Let us point to two aspects of system theory that are especially important in GAS application, characterized by many interacting decision makers—management of time and self-organization.

6.1 *Management of time in Geographic Automata Systems*

According to (5), the triplet of transition rules $\mathbf{T_S}$, $\mathbf{M_L}$, $\mathbf{R_N}$, determines the states $\mathbf{S}$, locations $\mathbf{L}$, and neighbors $\mathbf{N}$ of automata at time $t + 1$ based on their values at time t. It is very well known that different interpretations of the 'hidden' variable—time—in a discrete system can critically influence model formulation and resulting dynamics (Liu and Andersson 2004). On the one hand, one can 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), and the ability to set it up is implemented in the GAS-based modeling environment, as we discuss shortly.

6.2 *Emergence and self-organization in GAS*

From the point of view of system theory, to straightforwardly reflect geographic reality GAS employs *many* interacting automata. 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. Ghettos and market areas in urban contexts are an example.

The interactions of GAS automata are 'hidden' within transition rules $\mathbf{T_S}$, $\mathbf{M_L}$, and $\mathbf{R_N}$, and superficial analysis is sufficient to reveal that non-linearity is characteristic for GAS models. GAS can self-organize in two ways, thus reflecting the afore-mentioned dichotomy of fixed and non-fixed objects. First, fixed automata can change their properties in a way that entails emergence of assembled spatial units; models of voting are an obvious example (Stauffer 2001). Second, the same can happen when non-fixed elements change their locations, as in the examples explored through socio-spatial modeling (Sakoda 1971, Schelling 1969, 1971, 1974, 1978). (We will demonstrate a generalized implementation of Schelling's model in section 7.3.)

The study of emergence and self-organization as well as the possibility of abrupt bifurcations between different dynamic regimes is very often the very goal of a geographic model (urban examples are discussed in Portugali 2000). The phenomena might be especially important for GAS including human individuals, whose behavior is essentially based on the ability to recognize emergence or disintegration of ensembles of objects, as, for example, concentration of householders in particular groups. Any implementation of GAS will be, thus, incomplete, until it incorporates tools for delineating self-organization.

The GAS-based modeling environment OBEUS (Benenson *et al.* 2004) we discuss shortly can be considered as a demonstration of the constructiveness and usefulness of this approach. In addition to demonstrating proof-of-concept, it sheds light on the issue of translating the GAS framework into a working tool, and offers simple, but efficient approaches to the issue, including simple algorithms aimed at capturing spatial emergence in geographic systems.

7. Implementing GAS framework in an urban context

We will now demonstrate the operational implementation of the GAS framework as spatial simulation software designed to support modeling of complex urban systems, and with application to a generalized segregation model. The software—Object-Based Environment for Urban Systems (OBEUS)—has been developed at the Environment Simulation Laboratory at Tel Aviv University as a software package based on a GAS core. OBEUS itself is best considered as a foundation for developing geographic simulations in the style of other popular social science computing libraries such as Swarm (Minar *et al.* 1996), RePast (University of Chicago 2004), and the Multi-Agent Modeling Language (MAML) (Gulyás *et al.* 1999), but differentiated by its emphasis on *geography* and *relationships*. Users extend the base OBEUS package to suit their particular modeling requirements. It thus shields the user from much of the overhead required when building a spatial simulation, focusing her attention on specifying the automata objects included in the model and rules of their (spatial) behavior. The software itself is detailed elsewhere (Benenson *et al.* 2004). In this section, we demonstrate the implementation of the GAS framework in OBEUS, generally with reference to the modeling of urban dynamics, and specifically in the context of Schelling's segregation model.

7.1 *Implementing GAS demands in software*

The computational background of GAS should provide functionality for the representation of all the components defined by (4)–(5): to characterize and locate geographic automata; to determine neighborhood and other spatial relationships between them; and to formulate state transition, migration, and neighborhood rules. All this inherently fits to an *Object Oriented Programming* (OOP) paradigm and we describe the software designs in OOP-fashion in the proceeding sections.

7.1.1 Universal and User classes. Currently, there are two class levels in the OBEUS scheme: *Universal* and *User-defined*. The abstract classes of the Universal category are considered as those that are necessary for simulating *any* Geographic Automata System. User classes inherit abstract classes, and implement automata and transition rules characteristic of specific phenomena. In what follows, an abstract model of housing dynamics is considered as an example of such a phenomenon. User-defined classes reflect specificity in users' models and might be constructed anew, or acquired from previously developed applications and reused.

7.1.2 GAS software as Object-Oriented Database Management Systems. Data storage, updating, querying, and computation within OBEUS environments follows an *Entity-Relationship Model* (ERM) (Peckham *et al.* 1995). There are two rationales for this. First, as we noted before, the GAS approach is tightly coupled to vector GIS, which, in turn, is an extension of a relational database. Second, GAS separates automata types $\mathbf{K}$, states $\mathbf{S}$, locations $\mathbf{L}$, and neighbors $\mathbf{N}$. While type and state can be encapsulated within automata themselves, the information on relations between automata and neighbors and, in the case of indirect geo-referencing, location information is attributed to the pair of automata, that is, to *relationships* between them. Implementation of GAS can evidently benefit if relationships are implemented as separate software objects, as with ERM. To merge an ERM scheme and automata approach, OBEUS is developed as an Object-Oriented Database Management System (OODBMS).

7.1.3 Universal abstract classes. The basic components of GAS are defined in OBEUS with respect to automata types $\mathbf{k} \in \mathbf{K}$, its states $\mathbf{S_k}$, location $\mathbf{L}$, and neighborhood relations $\mathbf{N}$ to other objects. These are implemented by means of three abstract root classes (Figure 5): *Population*, which contains information regarding the population of objects of given type $\mathbf{k}$ 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 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 (figure 5).

7.1.4 The problem of managing relationships. Relationships in GAS models can change in time and an application of the $\mathbf{R_N}$ rules that describe these changes 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. Who has the right to destroy the relationship between the tenant and the property, then? This example represents the general *problem of consistency* in managing relationships. It has no one general solution; one can refer to the computer science literature for complex situations (Peckham *et al.* 1995). In OBEUS, we follow the development pattern proposed by Noble (2000). To retain consistency of 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

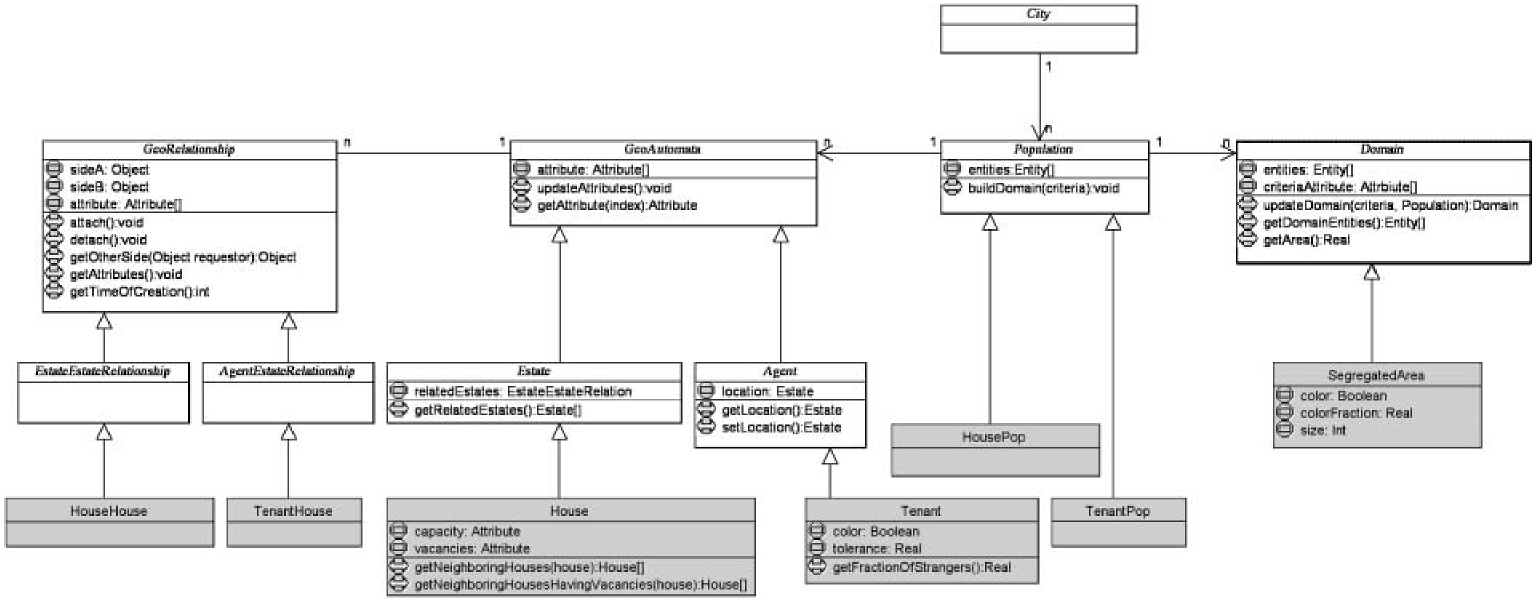


Figure 5. A UML scheme illustrating the abstract-level classes of OBEUS and the example of model-level classes for the Schelling simulation.

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. Accounting for limitations of OBEUS (direct relationships between agents are not allowed) the way to force the tenant to leave the property is to raise their payment. The tenant (the leader in tenant-property relationship) will likely end the relationship by herself in that case. There is no proof that the majority of real-world situations can be imitated by the *leader-follower* pattern, although we are not aware of any natural instance where this pattern is insufficient.

7.2 *Implementing system theory demands within OBEUS*

7.2.1 Management of time. We mentioned the importance of the synchronization mode for dynamic models; 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. It is worth noting that the logic of synchronous updating often passes conflicts further in time. If two mutually-avoiding agents occupy adjacent locations and simultaneously leave this unfriendly neighborhood, at a given time-step, then in synchronous mode nothing can prevent occupation of these locations by another pair of avoiding agents.

In *Asynchronous* mode, automata change in turn, 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.

7.2.2 Management of self-organizing spatial ensembles. We have also discussed the importance of tools aimed at recognition of the emerging units in software for modeling complex dynamic systems. In urban GAS models, typical examples of self-organizing spatial units are suburbs populated by members of a particular socioeconomic group or areas exhibiting similar land-use. These emerging *spatial ensembles* of geographic automata are supported in OBEUS by means of the abstract class *GeoDomain* (figure 5). 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.

Stated formally, for a given set of predicates C, the set of geographic automata D_C form a domain in OBEUS, if

1. For each $G \in D_C$, a sufficient number of G's neighbors (but not necessarily G itself) satisfy criteria C;
2. D_C contains sufficient number of geographic automata G;
3. D_C is 'practically' continuous.

This scheme is detailed elsewhere (Benenson *et al.* 2004).

Finally, a ***City*** class contains parameters (variables) and methods (mechanisms) relating to a city as a whole (Figure 5).

Inheriting and extending universal classes of OBEUS, we can build an urban geographic system, i.e., specify software objects representing fixed and non-fixed automata, and the geographic mechanisms of their location and relations to other automata. The rules $\mathbf{T_S}$, $\mathbf{M_L}$, and $\mathbf{R_N}$ that eventually drive system evolution become the methods of the classes at the User level.

The next section presents an example of these classes for a Schelling-like simulation (Schelling 1969) of socio-spatial segregation dynamics in a city comprising individual householders and households. It is worth noting that an abstract raster-based version and a version of the model based on a layer of real-world houses have identical formulation in OBEUS; the only difference between the versions is in the definition of neighboring relations between fixed house-automata, encapsulated within the *EstateEstate* class.

7.3 *A generalized Schelling model as an example of implementing a Geographic Automata Systems approach*

In order to demonstrate the usefulness of the GAS concept in modeling urban dynamics, we have formulated Schelling's popular segregation model (Schelling 1969, 1971, 1974, 1978), as a GAS, and in this section we present it in terms of the OBEUS software already discussed.

Schelling's original model was realized using black (B) and white (W) checkers on a chess board. If the fraction of agents of different type (say of W-type for a B-agent) within the neighborhood of a target agent's location is above the *tolerance* threshold of this agent, then the agent will attempt to relocate to a nearest unoccupied location, where the fraction of different agents is below that threshold. The Schelling model is asynchronous in its handling of time; each agent observes the state of the system as left by a previously-considered agent.

In formulating the model as a Geographic Automata System and implementing it in OBEUS, we assume that agents of two types are located in houses. Houses are characterized by their capacity. Agents differ in their tolerance to agents of an opposite type and react to a fraction of opposite agents in the neighborhood of those houses.

We thus define two types of objects (K=2), as denoted in table 1. The implementation accommodates these objects, their relationships, the location agreements, one state transition rule and one movement rule (table 1). According to the scheme (figure 5), abstract classes *Agent* and *Estate* are inherited by Tenant and House. Abstract relationship classes *EstateEstateRelationship* and *AgentEstateRelationship* are inherited by HouseHouse and TenantHouse, respectively. It is worth noting that objects of the HouseHouse class, when initiated, define neighborhood relationships between houses.

Objects belonging to the class House are endowed with two properties: Capacity and Vacancies; both are integers. Its methods include getNeighboring-Houses(house), which returns the list of neighboring houses, and getNeighboringHousesHavingVacancies(house), which returns a list of houses with vacancy.

Another class, Tenant, features with two properties of its own: Color, which is Boolean and has values B and W, and the Tolerance threshold, which is real. Its methods include getFractionOfStrangers(tenant, house), which returns a fraction

Table 1. Parameterization of the Schelling model implementation.

Object Type	*Fixed*	*Non-fixed*
Object's name and notation	House, H	Tenant, D
States S	Capacity P_H; Number of vacancies V_H	Color C_D, Tolerance threshold T_D
State transition rules T_S	P_H does not change in time; V_H equals P_H minus current number of tenants in H	–
Location agreements L	Shape of the footprint polygon	By pointing to a house H tenant D is located in: $D \rightarrow H$

Location transition rules M_L	–	Calculate fraction f_D of strangers among the neighbors of D
		If ($f_D < T_D$) do nothing;
		If ($f_D \geqslant T_D$) relocate to one of the neighboring houses satisfying $f_D < T_D$. If there are no vacancies in the neighboring houses, stay in a current house

Neighbors N	Can vary; for example, houses, which Voronoi polygons have common boundary with H	Tenants in the houses, neighboring to H
Neighborhood transition rules R_N	–	–

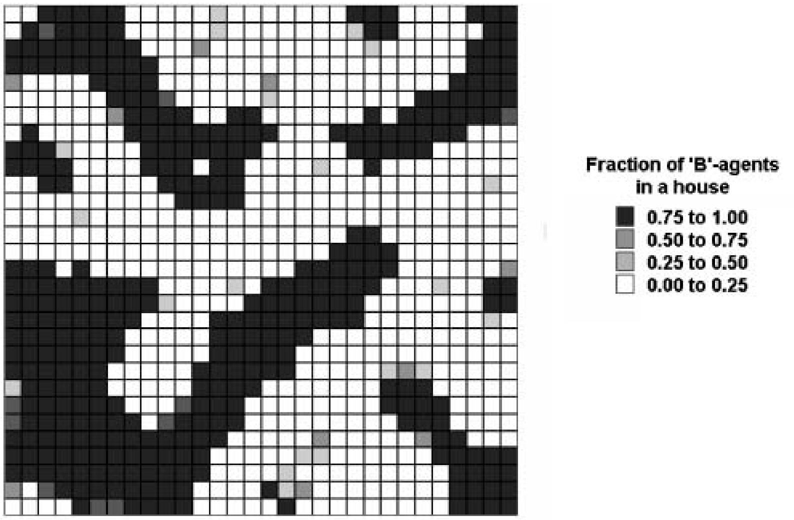
of neighbors of color opposite to that of the tenant *tenant*, if located in a house house.

To study residential segregation, emerging clusters populated mostly or exclusively by B- or W-agents should be identified in the simulation. If such clusters emerge and are recognized, they become objects of the SegregatedArea class, which inherits an abstract class *GeoDomain*. It is defined with reference to house objects and has three properties: Color, which defines the color of the segregated tenants; ColorFraction, which defines the fraction of segregated tenants of color Color over the neighborhood of each house belonging to SegregatedArea houses; and Size, which defines the number of houses in a segregated area.

This facilitates experimentation in simulation. Maps output from the Schelling simulation, implemented on abstract and real-world spaces, are illustrated in Figure 6.

8. Conclusions

We have introduced a Geographic Automata Systems framework as a unified scheme for representing discrete geographic systems. Technically, the framework is



(a)

(b)

Figure 6. Visual output of the Schelling model, implemented in (a) abstract and (b) real-world spaces.

designed to merge two popular tools used in urban simulation—Cellular Automata and Multi-Agent Systems—and specify them in a patently spatial manner. We have demonstrated the operational implementation of the GAS framework with reference to a general-purpose software tool for urban geographic simulation and its application to an urban segregation model. Conceptually, our assertion is that GAS forms the kernel of the system, as far as the system is spatially driven.

Let us consider a logical chain between geographic systems and GAS representations:

Geographic system→Priority of location information and spatial relations between elements→Collective dynamics of geographic automata in space→GAS

Indeed, with focus on geography, the system elements—geographic automata—are located and behave in space—urban space—in all the examples we have considered. The depiction of space necessitates *location conventions*, which differentiate between fixed (houses, land parcels, road segments) and moving spatial objects (householders, pedestrians, cars). A minimal realization of GAS borrows location conventions from vector GIS regarding fixed objects, and utilizes the latter as anchors for moving, non-fixed, objects. This has close analogies with the ways in which we understand real spatial entities to move within fixed geographic infrastructure, such as the case of a pedestrian shopper moving from Calvin Klein by foot, on toward Diesel by taxicab, and to Versace by limousine. These minimal location conventions are also sufficient for representing neighborhood and other spatial relationships. Thinking empirically, a tautological statement such as, "householders living in nearby houses are my neighbors" are those represented in a GAS context, involving little more than the expression of neighborhood relations between non-fixed householders via fixed real estate units. Formally, we use the notions of direct and indirect location to encapsulate these sorts of expressions in a simulation framework; informally, we claim that, for the majority of situations, this is just the way *humans describe space and spatial behavior*. It is not surprising, therefore, that a minimal GAS environment is sufficient for interpreting most urban CA and MAS simulations we know, as we present in table 2.

The priority of geography in the framework reinforces the strengths of the simulation environment and the basis on geographic relationships makes possible some simple, but important, steps toward including the ideas of self-organization and time-management in complex systems. Once we think of the main elements of a system in non-spatial terms, e.g., broker agents in a stock exchange model (Luna and Stefansson 2000), or Internet users on an Information Autobahn (Leonard 1997), the problem of selecting the relationships that are important for the model arises immediately. The common-sense geographic background for studying complex urban phenomena fades away in non-spatial systems.

The minimal GAS skeleton allows for a degree of standardization between automata models and other systems, not least of which are GIS. It also provides a mechanism for *transferability*. Until now, the majority of—if not all—spatial simulations could be investigated only by their developers. The development of GAS software breaches this barrier, offering opportunities to turn urban modeling from art into engineering.

Two additional steps are necessary for full implementation of the GAS framework; none, we think, demand decades of development. The first requirement is that the GAS framework should be transformed into a software environment. In an urban context, OBEUS is the first experiment in this direction; similar approaches, also based on object-based views of environmental processes, have recently been developed in ecology (Ginot *et al.* 2002). The second requirement is that a high-level, preferably geography-specific, *simulation language* based on the GAS approach should be developed. The goal is to enable the formulation of simulation rules in terms of objects' spatial behavior. We believe that the continued

Table 2. Existing urban cellular automata and multi-agent systems models expressed as Geographic Automata Systems.

Source	Form of geographic automata	Characterization of states	Location of objects: Fixed	Location of objects: Non-fixed	Neighborhood rule(s)
(Chapin and Weiss 1962, 1965, 1968, Donnelly *et al.* 1964)	Identical square land cells	Discrete ordinal variable denoting fraction of urban land-use	Rectangular grid	–	3×3 Moore neighborhood
(Engelen *et al.* 1995; White and Engelen 1993, 1994, 1997, White *et al.* 1997)	Identical square land cells	Nominal variable representing four land-uses: vacant, housing, industry, commerce.	Rectangular grid	–	Cells at a distance less than 7 cell-units
(Allen and Sanglier 1979, Bura *et al.* 1996, Sanders *et al.* 1997)	Identical hexagonal land cells	Non-urban and urban cells, the latter characterized by continuous variables representing population, production, services, etc.	Hexagonal grid	–	Potentially each cell influences the other, but the influence decays with distance and when there is a physical barrier (always associated with cell boundary) between cells
(Batty 1998, Batty and Xie 1994, 1997, Xie 1996)	Identical square land cells	Nominal variable, representing land-use. Often, only two uses—urban and non-urban—are considered	Rectangular grid	–	Three neighborhoods of increasing radius with the cell in vicinity: the local one is included into the neighborhood of 'interactions', the latter included into the neighborhood of 'constraints'
(Wu 1996)	Identical square land cells	Nominal variable representing four land-uses: cultivated, wood, urban, transport, water	Rectangular grid	–	5×5 Moore neighborhood

Table 2. (*Continued*).

Source	Form of geographic automata	Characterization of states	Location of objects: Fixed	Location of objects: Non-fixed	Neighborhood rule(s)
(Wu 1998, Wu and Webster 1998)	Identical square land cells	Cell potential represented by a vector of continuous economic characteristics and a binary variable denoting urban/non-urban usage	Rectangular grid	–	Decay of cell influence with distance, 3 × 3 Moore neighborhood
(Batty 1998, Wu 1998)	Identical square land cells	Continuous variable representing cell potential and a binary variable denoting urban/non-urban usage	Rectangular grid	–	3 × 3 Moore neighborhood
(Besussi *et al.* 1998)	Identical square land cells	Nominal variable, which denotes 18 population/land use states	Rectangular grid	–	3 × 3 Moore neighborhood
(Li and Yeh 2000, Yeh and Li 2000, 2001, 2002)	Identical square land cells	Nominal variable representing four land-uses: vacant, housing, industry, commerce. Continuous variable representing cell potential	GIS-based raster coverage	–	Two kinds of neighborhoods: neighbors within a circular radius of two cells; and a larger neighborhood with radius defined by the distance between urban centers (the latter established in advance)
(Erickson and Lloyd-Jones 1997)	Street segments and buildings, both represented by sets of cell	Ordinal variable representing street type (five grades). Ordinal variable representing building type (five grades)	Expanding irregular network of cells with constraints imposed on the distance between cells and their relative orientation	–	Adjacent units are neighbors

Table 2. (*Continued*).

Source	Form of geographic automata	Characterization of states	Location of objects: Fixed	Location of objects: Non-fixed	Neighborhood rule(s)
(Semboloni 2000)	Road links, land parcels	None for roads, states for land units include: unbuilt, housing, unoccupied, services for land unit	Voronoi tessellation of land units	–	Road links are neighbors if connected. Land units are neighbors if they share a common boundary
(Candau *et al.* 2000, Clarke and Gaydos 1998, Clarke *et al.* 1997)	Street cells and urban cells	Urban/non-urban	Rectangular grid	–	3×3 Moore neighborhood
(Semboloni 1997)	Identical square land cells	Nominal variable, which denotes population and land-use: characteristics include white-collar population, blue-collar population, services, base activities, empty sites	Rectangular grid	–	Two kinds of neighborhood: 3×3 Moore, and bigger neighborhoods, with a variable radius defined by the travel distance from the vicinity
(Benati 1997)	Identical square land cells, migrating firms	Cell: binary variable representing presence/absence of customers; firm: no parameter (can serve unlimited number of customers)	Rectangular grid	By pointing to grid cells	5×5 Moore neighborhood
(Benenson 1998, 1999)	Houses, migrating householders	Houses are represented by a continuous variable 'value', householder agents by two continuous variables, 'status' and 'ethnicity'	Rectangular grid	By pointing to grid cells	5×5 Moore neighborhood
(Benenson *et al.* 2002)	Houses, migrating householders	Ordinal variable representing house architectural style, nominal variable representing ethnic identity of householder agents	Voronoi coverage built on the base of house centroids	By pointing to houses	Houses are neighbors if their Voronoi polygons share a common boundary

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.

References

ALLEN, P.M. and SANGLIER, M., 1979, A dynamic model of growth in a central place system. *Geographical Analysis*, **11**(3), pp. 256–272.

BATTY, M., 1998, Urban evolution on the desktop: simulation with the use of extended cellular automata. *Environment and Planning A*, **30**, pp. 1943–1967.

BATTY, M., COUCLELIS, H. and EICHEN, M., 1997, Special issue: urban systems as cellular automata. *Environment and Planning B*, **24**(2).

BATTY, M. and XIE, Y., 1994, From cells to cities. *Environment and Planning B*, **21**, pp. s31–s48.

BATTY, M. and XIE, Y., 1997, Possible urban automata. *Environment and Planning B*, **24**, pp. 175–192.

BAYLOR UNIVERSITY, 2002, GRASS GIS (Geographic Resources Analysis Support System) 5.0 (Software). Center for Applied Geographic and Spatial Research, Baylor University, Waco, TX.

BENATI, S., 1997, A cellular automaton for the simulation of competitive location. *Environment and Planning B*, **24**, pp. 175–192.

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

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

BENENSON, I., ARONOVICH, S. and NOAM, S., 2004, Let's talk objects: generic methodology for urban high-resolution simulation. *Computers, Environment and Urban Systems*, in press.

BENENSON, I., OMER, I. and HATNA, E., 2002, Entity-based modeling of urban residential dynamics: the case of Yaffo, Tel Aviv. *Environment and Planning B: Planning and Design*, **29**, pp. 491–512.

BENENSON, I. and TORRENS, P.M., 2004a, *Geosimulation: Automata-Based Modeling of Urban Phenomena* (London: John Wiley & Sons).

BENENSON, I. and TORRENS, P.M., 2004b, Geosimulation: object-based modeling of urban phenomena. *Computers, Environment and Urban Systems*, **28**(1/2), pp. 1–8.

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

BESUSSI, E., CECCHINI, A. and RINALDI, E., 1998, The diffused city of the Italian North-East: identification of urban dyanmics using cellular automata urban models. *Computers, Environment and Urban Systems*, **22**(5), pp. 497–523.

BROWN, J.S. and DUGUID, P., 2000, *The Social Life of Information* (Boston, MA: Harvard Business School Press).

BURA, S., GUÉRIN-PACE, F., MATHIAN, H., PUMAIN, D. and SANDERS, L., 1996, Multiagent systems and the dynamics of a settlement system. *Geographical Analysis*, **28**(2), pp. 161–178.

CANDAU, J.T., RASMUSSEN, S. and CLARKE, K.C., 2000, A coupled cellular automata model for land use/land cover change dynamics. Paper read at 4th International Conference on Integrating GIS and Environmental Modeling (GIS/EM4): Problems, Prospects and Research Needs, September 2–8, 2000, at Banff, Alberta, Canada.

CENTRE FOR COMPUTATIONAL GEOGRAPHY, 2002, GeoTools 2.0. University of Leeds, School of Geography, Leeds.

CHAPIN, F.S. and WEISS, S.F., 1962, *Factors Influencing Land Development. An Urban Studies Research Monograph* (Chapel Hill: Center for Urban and Regional Studies, Institute for Research in Social Science, University of North Carolina).

CHAPIN, F.S. and WEISS, S.F., 1965, *Some Input Refinements for a Residential model. An Urban Studies Research Monograph* (Chapel Hill, NC: Center for Urban and Regional Studies, Institute for Research in Social Science, University of North Carolina).

CHAPIN, F.S. and WEISS, S.F., 1968, A probabilistic model for residential growth. *Transportation Research*, **2**, pp. 375–390.

CLARK LABS, 2002, IDRISI 32 (Software). Clark Labs, Worcester, MA.

CLARKE, K.C. and GAYDOS, L., 1998, Loose coupling a cellular automaton model and GIS: long-term growth prediction for San Francisco and Washington/Baltimore. *International Journal of Geographical Information Science*, **12**(7), pp. 699–714.

CLARKE, K.C., HOPPEN, S. and GAYDOS, L., 1997, A self-modifying cellular automaton model of historical urbanization in the San Francisco Bay area. *Environment and Planning B*, **24**, pp. 247–261.

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

DE ANGELIS, D. and GROSS, L.J., 1992, *Individual-Based Models and Approaches in Ecology: Populations, Communities, and Ecosystems* (New York: Chapman and Hall).

DIJKSTRA, J., TIMMERMANS, H.J.P. and JESSURUN, A.J., 2000, A multi-agent cellular automata system for visualising simulated pedestrian activity. In *Theoretical and Practical Issues on Cellular Automata*, S. Bandini and T. Worsch (Eds), pp. 29–36 (London: Springer-Verlag).

DONNELLY, T.G., CHAPIN, F.S. and WEISS, S.F., 1964, *A Probabilistic Model for Residential Growth. An Urban Studies Research Monograph* (Chapel Hill, NC: Center for Urban and Regional Studies, Institute for Research in Social Science, University of North Carolina).

ENGELEN, G., WHITE, R., ULJEE, I. and DRAZAN, P., 1995, Using cellular automata for integrated modelling of socio-environmental systems. *Environmental Monitoring and Assessment*, **30**, pp. 203–214.

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

EPSTEIN, J.M. and AXTELL, R., 1996, *Growing Artificial Societies from the Bottom Up* (Washington, D.C.: Brookings Institution).

ERICKSON, B. and LLOYD-JONES, T., 1997, Experiments with settlement aggregation models. *Environment and Planning B: Planning and Design*, **24.** pp. 903–928.

GARDNER, M., 1970, The fantastic combinations of John Conway's new solitaire game "Life". *Scientific American*, **223**(4), pp. 120–123.

GARDNER, M., 1971, Mathematical games: on cellular automata, self-reproduction, the Garden of Eden, and the game 'life'. *Scientific American*, **224**(2), pp. 112–117.

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).

GINOT, V., LE PAGE, C. and SOUISSI, S., 2002, A multi-agents architecture to enhance end-user individual-based modeling. *Ecological Modelling*, **157**, pp. 23–41.

GULYÁS, L., KOZSIK, T. and CORLISS, J.B., 1999, The multi-agent modelling language and the model design interface. *Journal of Artificial Societies and Social Simulation*, **2**(3).

IFRAH, G., 2000, *The Computer and the Information Revolution: The Universal History of Numbers III*, Translated by E.F. Harding (London: The Harvill Press). Original edition, Histoire Universelle des Chiffres, 1994.

KOHLER, T.A., 2000, Putting social sciences together again: an introduction to the volume. In *Dynamics in Human and Primate Societies*, T.A. Kohler and G. Gumerman (Eds), pp. 1–18 (New York: Oxford University Press).

LEONARD, A., 1997, *Bots: The Origin of a New Species* (San Francisco, CA: Hardwired).

LI, X. and YEH, A.G.-O., 2000, Modelling sustainable urban development by the integration of constrained cellular automata and GIS. *International Journal of Geographical Information Science*, **14**(2), pp. 131–152.

LIU, X.-H. and ANDERSSON, C., 2004, Assessing the impact of temporal dynamics on land-use change modeling. *Computers, Environment and Urban Systems*, **28**(1/2), pp. 107–124.

LUNA, F. and STEFANSSON, B., eds, 2000, *Economic Simulation in Swarm: Agent-based Modelling and Object Oriented Programming* (Dordrecht: Kluwer).

MEYER, J.-A., BERTHOZ, A., FLOREANO, D., ROITBLAT, H.L. and WILSON, S.W., eds, 2000, *From Animals to Animats 6: Proceedings of the Sixth International Conference on Simulation of Adaptive Behavior* (Cambridge, MA: MIT Press).

MICROSOFT CORPORATION AND DIGITAL EQUIPMENT CORPORATION, 1995, The Component Object Model Specification 0.9. Microsoft Corporation and Digital Equipment Corporation, Redmond, WA.

MINAR, N., BURKHART, R., LANGTON, C. and ASKENAZI, M., 1996, *The Swarm Simulation System: A Toolkit for Building Multi-agent Simulations* (Santa Fe: Santa Fe Institute. Working paper).

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), pp. 154–160.

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

O'SULLIVAN, D., 2001, Exploring spatial process dynamics using irregular cellular automaton models. *Geographical Analysis*, **33**(1), pp. 1–18.

PECKHAM, J., MACKELLAR, B. and DOHERTY, M., 1995, Data models for extensible support of explicit relationships in design databases. *VLDB Journal*, **4**, pp. 157–191.

PHIPPS, M., 1989, Dynamic behavior of cellular automata under the constraint of neighborhood coherence. *Geographical Analysis*, **21**, pp. 197–215.

PORTUGALI, J., 2000, *Self-Organization and the City* (Berlin: Springer-Verlag).

PORTUGALI, J., BENENSON, I. and OMER, I., 1997, Spatial cognitive dissonance and sociospatial emergence in a self-organizing city. *Environment and Planning B*, **24**, pp. 263–285.

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

REYNOLDS, C., 1999, Steering behaviors for autonomous characters. Paper read at Game Developers Conference, at San Jose, CA.

SAKODA, J.M., 1971, The checkerboard model of social interaction. *Journal of Mathematical Sociology*, **1**, pp. 119–132.

SANDERS, L., PUMAIN, D., MATHIAN, H., GUÉRIN-PACE, F. and BURA, S., 1997, SIMPOP: A multiagent system for the study of urbanism. *Environment and Planning B*, **24**, pp. 287–305.

SCHELLING, T.C., 1969, Models of segregation. *American Economic Review*, **59**(2), pp. 488–493.

SCHELLING, T.C., 1971, Dynamic models of segregation. *Journal of Mathematical Sociology*, **1**, pp. 143–186.

SCHELLING, T.C., 1974, On the ecology of micro-motives. In *The Corporate Society*, R. Marris (Ed.), pp. 19–55 (London: Macmillan).

SCHELLING, T.C., 1978, *Micromotives and Macrobehavior* (New York: WW Norton and Company).

SCHOFISCH, B. and HADELER, K.P., 1996, Dimer automata and cellular automata. *Physica D*, **94**, pp. 188–204.

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

SEMBOLONI, F., 1997, An urban and regional model based on cellular automata. *Environment and Planning B*, **24**, pp. 589–612.

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

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

STAUFFER, D., 2001, Monte Carlo simulations of Sznajd models. *Journal of Artificial Societies and Social Simulation*, **5**(1).

SUN MICROSYSTEMS, 2002, Java Development Kit 1.4.1. Sun Microsystems, Mountainview, CA.

TESFATSION, L., 1997, How economists can get alife. In *The Economy as an Evolving Complex System II*, B.W. Arthur, S. Durlaf and D. Lane (Eds), pp. 533–564 (Reading, MA: Addison-Wesley).

TOBLER, W., 1979, Cellular Geography. In *Philosophy in Geography*, S. Gale and G. Ollson (Eds), pp. 379–386 (Dordrecht: Kluwer).

TORRENS, P.M., 2002, Cellular automata and multi-agent systems as planning support tools. In *Planning Support Systems in Practice*, S. Geertman and J. Stillwell (Eds), pp. 205–222 (London: Springer-Verlag).

TORRENS, P.M., 2004a, Geosimulation approaches to traffic modeling. In *Transport Geography and Spatial Systems*, P. Stopher, K. Button, K. Haynes and D. Hensher (Eds), (London: Pergamon, in press).

TORRENS, P.M., 2004b, Looking forward: remote sensing as dataware for human settlement simulation. In *Remote Sensing of Human Settlements*, M. Ridd (Ed.) (New York: John Wiley and Sons), in press.

TORRENS, P.M. and O'SULLIVAN, D., 2001, Cellular automata and urban simulation: where do we go from here? *Environment and Planning B*, **28**(2), pp. 163–168.

UNGERER, M.J. and GOODCHILD, M.F., 2002, Integrating spatial data analysis and GIS: a new implementation using the Component Object Model (COM). *International Journal of Geographical Information Science*, **16**(1), pp. 41–53.

UNIVERSITY OF CHICAGO, 2004, RePast 3.0 (Software). Social Science Research Computing Program, Chicago.

WAHLE, J., NEUBERT, L., ESSER, J. and SCHRECKENBERG, M., 2001, A cellular automaton traffic flow model for online simulation of traffic. *Parallel Computing*, **27**, pp. 719–735.

WHITE, R. and ENGELEN, G., 1993, Cellular automata and fractal urban form. *Environment and Planning A*, **25**, pp. 1175–1199.

WHITE, R. and ENGELEN, G., 1994, Urban systems dynamics and cellular automata: fractal structures between order and chaos. *Chaos, Solitions, and Fractals*.

WHITE, R. and ENGELEN, G., 1997, Cellular automata as the basis of integrated dynamic regional modelling. *Environment and Planning B*, **24**, pp. 235–246.

WHITE, R., ENGELEN, G. and ULJEE, I., 1997, The use of constrained cellular automata for high-resolution modelling of urban land use dynamics. *Environment and Planning B*, **24**, pp. 323–343.

WIENER, N., 1961, *Cybernetics: or Control and Communication in the Animal and the Machine* (Cambridge, MA: MIT Press).

WOLFRAM, S., 2002, *A New Kind of Science* (Champaign, IL: Wolfram Media, Inc).

WU, F., 1996, A linguistic cellular automata simulation approach for sustainable land development in a fast growing region. *Computers, Environment and Urban Systems*, **20**, pp. 367–387.

WU, F., 1998, SimLand: a prototype to simulate land conversion through the integrated GIS and CA with AHP-Derived Transition Rules. *International Journal of Geographical Information Science*, **12**(1), pp. 63–82.

WU, F. and WEBSTER, C.J., 1998, Simulation of land development through the integration of cellular automata and multicriteria evaluation. *Environment and Planning B*, **25**, pp. 103–126.

XIE, Y., 1996, A generalized model for cellular urban dynamics. *Geographical Analysis*, **28**(4), pp. 350–373.

YEH, A.G.-O. and LI, X., 2000, Simulation of compact cities based on the integration of cellular automata and GIS. In *Theoretical and Practical Issues on Cellular Automata*, edited by S. Bandini and T. Worsch (London: Springer-Verlag), pp. 170–178.

YEH, A.G.-O. and LI, X., 2001, A constrained CA model for the simulation and planning of sustainable urban forms by using GIS. *Environment and Planning B: Planning & Design*, **28**(5), pp. 733–753.

YEH, A.G.-O. and LI, X., 2002, A cellular automata model to simulate development density for urban planning. *Environment and Planning B: Planning & Design*, **29**(3), pp. 431–450.