Torrens, P.M. (2003) “Automata-based models of urban systems”. In Advanced Spatial Analysis, P. Longley & M. Batty (Eds.), Redlands, ESRI Press, pp. 61-79

Chapter

4

Automata-based models of urban systems

Paul M. Torrens

Simulation models of urban-activity location are currently undergoing a transition from large scale, aggregate spatial representation in a static equilibrium to much finer scale disaggregate forms where dynamic processes are the prime focus of the simulation. This chapter reviews the developments where automata and agents feature centrally and suggests a framework for geographical automata systems that combines the key features of cellular automata approaches to urban development with multi-agent modelling. The framework is developed and applied to residential simulation in the Singapore model, pointing the way to new ways of integrating demand supply sides through agent-based modelling.

1 History of automata

Automata were first conceived of in the 1930s by the British mathematician Alan Turing. Since then, the idea has been expanded and used for a variety of purposes: automata form the basic principle on which the digital computer is based, they are the mainstay of artificial intelligence (AI) and artificial life (ALife), and authors have suggested that the universe may even be regarded as an automaton itself (Wolfram 2002). Recently, automata have seen application in the realm of model development, where they are used as building blocks for the computer simulation of complex systems. Researchers in geography and urban studies have also begun to use automata to develop models of urban systems (Batty, Couclelis and Eichen 1997; Benenson and Torrens 2003 O'Sullivan and Torrens 2000; Torrens 2002).

Simply stated, an automaton is a processing unit, which itself can be characterized using variables of any description. In addition, an automaton is endowed with the ability to process information input to it from external sources, generally understood to be the information contained in other neighbouring automata (Figures 1 and 2). Various rules can be designed to determine how an automaton processes the information contained in its own characteristics, as well as that which it receives as input from neighbouring automata. These rules are time-dependent and can be considered as transition rules governing how automata should adapt and change over time in reaction to information in their surroundings. Herein lies the power of the automata concept: any mechanism, process or action that can be expressed in computable terms can be used to process information in an automaton. Practically speaking, this means that automata can be used to mimic just about any process. They are, therefore, very powerful tools for simulation.

Geographers became interested in automata in the 1980s, with earlier contributions from researchers in urban studies. One class of automata in particular, cellular automata (CA), have become especially popular for urban simulation in recent years. Also, another form of automata-multi-agent systems (MAS) are beginning to be adopted for use in urban modelling (Benenson and Torrens 2003; Benenson 1998; Benenson, Omer and Hatna 2002). However, automata tools do not simply 'port' across to geographical applications from their origins in mathematics and computer science, partially because investigation with these tools has not usually focused on the spatial properties, or the spatial applications, of the tools. Nevertheless, automata offer significant advantages for representing space and space-time dynamics, and research into the modification of automata for geographical use, application to spatial systems, and development of spatially-explicit automata-based software is active (Torrens and O'Sullivan 2001).

This chapter presents an overview of research projects in the field of automata-based modelling of urban systems that the author is engaged in. Section 2 describes basic automata, cellular automata, and multi-agent systems and section 3 generalizes these concepts to the application of urban systems. Section 4 describes ongoing research in automata-based modelling of urban systems, discussing the derivation of spatially-explicit automata tools and their application to urban systems. Section 5 concludes the chapter with a discussion of remaining limitations and opportunity for work in the field.

2 Automata, cellular automata and multi-agent systems

Basic automata, such as in Turing machines, are simple processing mechanisms, albeit with surprising power and functionality despite their simple specification. Basic automata are composed of a few components: states, an input stream, rules and a 'clock'. States describe internal attributes of an automaton: on, off, 1, 0, road, rail, etc. The input stream to a given automaton consists of information gleaned from outside the automaton, which the automaton will then process using its rule-set. Input can take any form, although it is generally formulated as information derived from the states of neighbouring automata (for example, the automaton to the left is 'on' and the automaton to the right is 'off'). Rules are conditional statements (which may also take the form of mathematical operators) that determine how an automaton should react to the information in its input stream. Generally, these rules are linked to an automaton's clock, governing how an automaton should alter its own internal states between time-steps based on information delivered via an input stream.

CA operate in much the same way as the basic automata described above. In the CA approach, however, individual automata are understood to be bounded by a cellular structure, for example, a grid, a hexagon or an irregular polygon. The cell represents the discrete confines of an automaton. Collections of CA can thus be understood to form a lattice structure. Also, the neighbourhoods from which an individual cellular automaton draws input can be defined as a structure of adjacent cells in some arrangement around an automaton (Figure 1).

MAS operate on the same principles as basic automata. Individual automata in a MAS are understood to be autonomous agent-automata, and their state descriptors generally reflect some agent-based characteristics. Likewise, transition

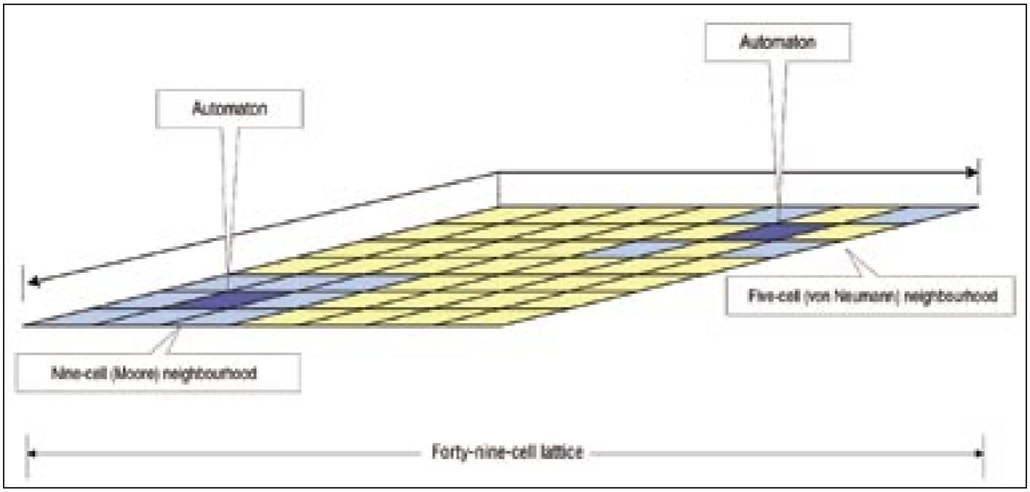


Figure 1 Two-dimensional cellular automata arranged in a regular lattice tessellation

rules for MAS are generally formulated in such a way as to represent behavioural characteristics or in some instances to simulate intelligence. Individual agent-automata may also be bounded by a discrete cellular space, although this is not a requirement.

MAS differ from CA in one important respect: individual automata are free to move within the spaces that they ‘inhabit’. With CA, information moves between cells, propagating by diffusion through neighbourhoods as input streams between automata. However, individual cells themselves remain fixed in the CA lattice. In contrast, with MAS, automata are mobile. This has obvious consequences for the representation of spatial systems, and this is a topic that will be explored later in the chapter. The movement of agent-automata has implications for other components of the automata system, however. Neighbourhood relationships in agent-automata are dynamic: when individual agents themselves alter their location in the simulated space, the measures of adjacency between them also change (Figure 2).

3 Automata as urban simulation tools

Assembling artificial cities from automata building-blocks

The components of automata listed above have close analogies with cities. Most urban entities, phenomena and systems can be specified as automata. State variables can be used to encode a wide variety of properties of urban systems into an

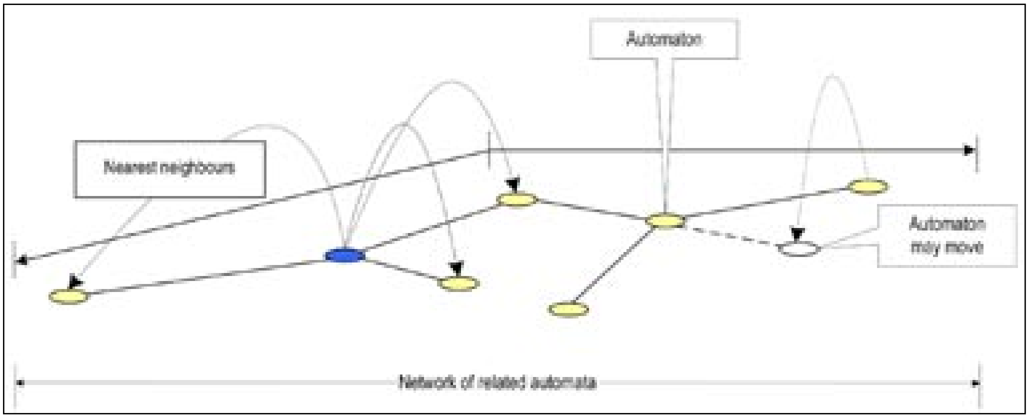


Figure 2 Two-dimensional multi-agent systems related by nearest neighbours

automaton model: land cover, land use, population density, etc. Similarly, a variety of urban objects can be represented as cells: land parcels, vehicles, road links, etc. Others can be represented without cellular boundaries: property centroids, trip waypoints, transport nodes, etc. Neighbourhood functions can be specified based on well-understood geographical relations such as market catchment areas, commuting watersheds, walking distance buffers, etc. Transition rules may be specified in such a way that they incorporate any geographic theory or methodology, for example, bid-rent theory, spatial cognition, space-time budgets, etc. Likewise, a variety of spatially-explicit movement rules can be designed for non-fixed automata, including collision avoidance, lane-changing and navigation. Also, an assortment of automata clocks can be designed to mimic the temporal attributes of real-world urban systems.

The advantages of automata as simulation tools

Automata models offer significant advantages for simulation construction in general and spatial simulation in particular. First, automata systems are decentralized in nature, with individual automata retaining autonomy within the system. In this sense, automata can be designed to work, collectively, 'from the bottom up' to accomplish tasks. Decentralisation is beneficial for a number of reasons. Centralisation in urban models is often an artefact of the methodology being used rather than an intuitive attribute. Most systems of interest in social science are understood to operate in a decentralised fashion, and automata offer the flexibility to represent them in a simulation environment. Also, because automata act as independent processors, automata models can be designed as massively parallel systems, with associated advantages for computational efficiency, especially when simulation processing is distributed across machines or processors.

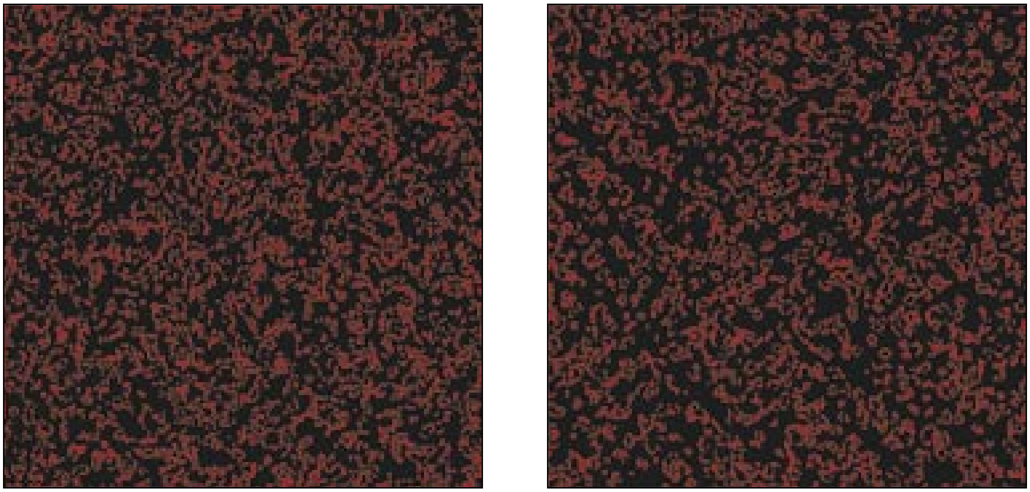
Automata models can be designed to represent simulated objects at very high resolutions. They can be thought of as microscopic or atomic models, with objects represented at entity-level resolutions: pedestrians, households, vehicles, houses, etc. (Benenson and Torrens 2003). This has obvious advantages for representing human systems: it allows model developers to part with notions of average individuals and to avoid problems of ecological fallacy. In terms of representing spatial systems, it also offers opportunities for circumnavigating modifiable areal unit problems. If coarser resolutions are required, they can be aggregated on an intuitive basis from collections of objects at finer resolutions (Benenson and Torrens 2003).

The emphasis on interaction in automata models is another important property for simulation, particularly because the autonomous treatment of individual automata permits the simulation of entity-level interaction and any emergent behaviour that may result. Previous popular methodologies for urban simulation represented interaction in terms of aggregate flows; the spatial interaction model is an example.

The dynamic nature of automata is another attractive property for simulation-building. As mentioned, individual automata can be designed with internal clocks. Time in automata moves in discrete steps, and these steps can be specified at resolutions that approximate real-time. State transition rules can be tied to simulated time-clocks so that processes may be simulated along realistic time-scales. For example, a model of traffic on a highway could be made to simulate real-time traffic movement, as well as diurnal periods of peak congestion, and even more long-term patterns such as holiday-influenced volume surges.

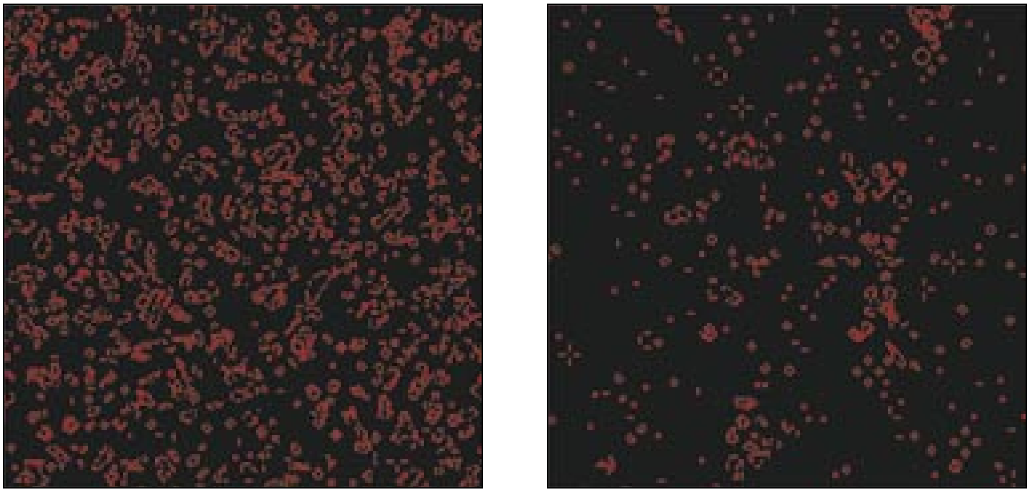
Simplicity is one of the often-recommended advantages of automata as a simulation tool. As with the Game of Life and Boids, automata simulations designed with simple specifications and few rules are often capable of yielding complex behaviours (Figures 3 and 4). The decentralised and autonomous characteristics of automata allow for a variety of complex characteristics to be generated from simple conditions: chaos, emergence, self-organisation, non-linearity, phase transition, etc. This is appropriate for many urban systems where almost chaotic end-states are understood to result from simple initial conditions.

Of course, one of the motivations behind using automata as an urban simulation tool is the advantage that they offer for spatial modelling. In the context of CA, cellular boundaries can be designed to mimic various geographies: standard



A Initial seed conditions

B After 10 time-steps



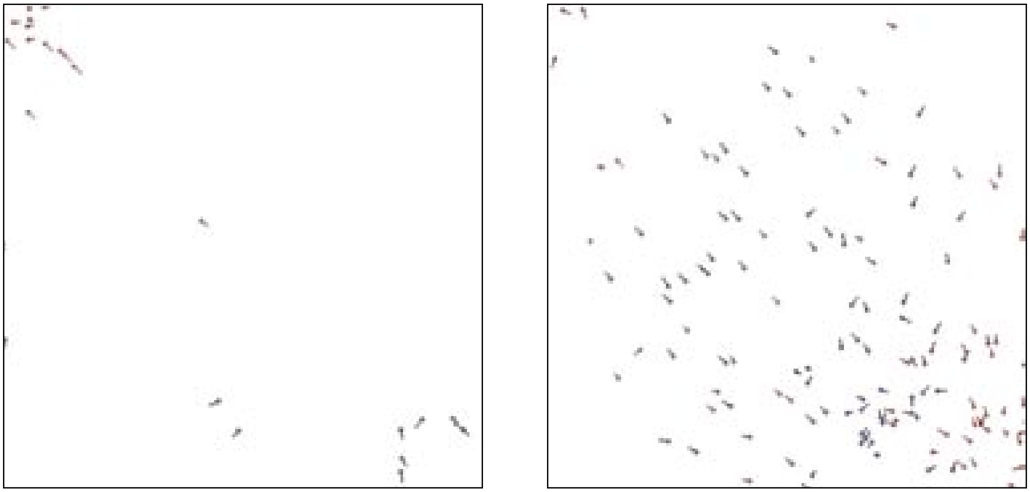
C After 25 time-steps

D After 500 time-steps

Figure 3 Game of Life at several stages of evolution.

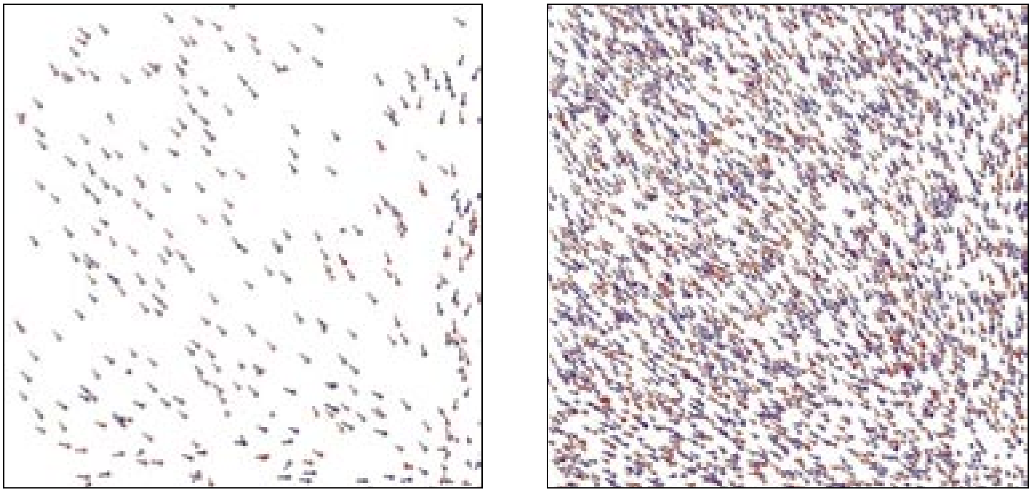
The model was developed using the RePast Java libraries (University of Chicago 2002). The model was defined as a 100 by 100 toroid space, with a cell size of two pixels. The initial seed conditions were based on the string 20010204593, with the space set to 50% fill capacity at the initial time-step

University of Chicago 2002, is it in bibliography?



A Fifty Boids

B Two hundred Boids



C Five hundred Boids

D Five thousand Boids

Figure 4 Multi-agent systems specified with a varying number of Boids.

The transition rules for the Boids model are detailed in Reynolds, 1987

Source: Thanks to Steve Coast, Department of Physics, University College London for the base Java code used to generate these images polygons, Delauny triangulations, Voronoi polygons, etc. Various spatial analysis techniques from geographic information science (GISc) are available for generating such tessellations. Likewise, neighbourhood expressions can be used to encode spatial topologies and structures into automata models: networks, lattices, graphs, etc. Transition rules allow for the infusion of urban and geographic theory directly into model designs. In particular, transition rules permit the expression of form and function in a symbiotic relationship: the processes that drive systems can be expressed with the patterns that they generate. Again, there are formal geographic expressions that can be used to articulate these relationships: GIS operators, geo-algebra, etc.

Application of automata tools to urban systems

Automata tools have been used to build urban simulations for an array of applications, both as pedagogic models and planning support systems (Torrens 2002). CA, in particular, have enjoyed widespread use as tools for urban development modelling, land-use simulation, and land-use/land-cover change modelling. MAS models have been less popular in urban simulation, although there seems no reason why this might be the case. Nevertheless, MAS models of urban systems have been constructed for a variety of purposes: simulating residential location dynamics (Torrens 2001; Benenson, Omer, and Hatna 2002), traffic systems (Barrett et al 2001), and urban population dynamics (Benenson and Torrens 2003). However, in most cases, these models have actually been formulated as CA and simply reinterpreted as MAS.

4 Using automata-based tools to advance urban simulation research

Despite the numerous advantages of automata for urban simulation and the volume of applications to urban systems, the field is very much in its infancy as an area of research. In papers published elsewhere the author has outlined potential research threads for automata-based modelling of urban systems (O'Sullivan and Torrens 2000; Torrens and O'Sullivan 2001). The following sections will detail some of the work that the author has been pursuing at CASA along these lines of inquiry. The next section deals with the design of spatial automata systems. This is followed by sections that outline ongoing work of applying this framework to urban systems.

Geographic Automata Systems as an expressly spatial simulation technology

There exists some justification for developing new and patently spatial automata-based simulation technologies for urban simulation. There is somewhat of a disjoint in the current literature between CA and MAS models in urban applications. CA are often used to model processes that would be better modelled using MAS and vice-versa. In many cases, MAS models are specified as CA models, with cells being paraphrased as agents. There is nothing particularly wrong with this approach, but it may be more useful to use the right tool for the right job. To some extent, the confusion between CA and MAS can be cleared by looking at their spatial attributes. CA are fixed and MAS can move. CA transmit information through their neighbourhoods, and the propagation of information is constrained by this function; in MAS, information can be exchanged through agent-neighbourhoods, but information can also travel with the agent in which it is housed and so may be less constrained than in CA (Figure 2). This makes CA an appropriate tool for fixed entities that influence their environments by diffusive processes; MAS are more appropriate for representing non-fixed objects that affect their surroundings by exchange mechanisms such as communication, memes and stigmergy. However, both properties are complementary, and an explicitly spatial framework for automata model-building could reconcile the functionality of CA and MAS in a seamless fashion.

Often, automata-based tools are borrowed from origins in other disciplines outside geography: computer science, mathematics, physical sciences, etc. For the most part, the tools were originally developed for non-spatial uses and their application to geographical systems necessitates modification. Geographers have long been uncomfortable with modifying automata, uneasy about departing from strict formalisms (Torrens and O'Sullivan 2001). Nevertheless, by modifying automata-based tools along spatial lines, there is an opportunity to learn a great deal about the real-world systems that they are abstracting, as well as developing useful simulation environments for exploring geographic ideas and hypotheses. To date, most of the modification to automata-based tools has been rather ad hoc in nature, and there is a need for a formal framework to develop explicitly spatial automata models.

A number of authors have hinted at the rich potential for integrating automata and GIS. For the most part, current applications constitute a loose coupling of the two. The possibility of more tightly coupled models has received relatively little attention. In particular, there remains much opportunity for connecting automata with principles from GISc for manipulating data for use in automata models, tessellating automata spaces, formulating neighbourhood functions, and expressing other spatial relationships between automata.

Finally, there is a strong need for geographically-rooted software libraries for developing automata-based models. A number of libraries enjoy widespread use, including Swarm (Swarm Development Group 2001), RePast (University of Chicago 2002), and Ascape (Brookings Institution 2001), however, they were not designed explicitly for spatial applications and their representation of space is often weak. This is unfortunate, particularly considering the symmetry between object-oriented programming paradigms, urban models, and GIS.

University of Chicago 2002, is it in bibliography?

With these ideas in mind, the author has been collaborating with the Environmental Simulation Laboratory at Tel Aviv University's Department of Geography and Human Environment to build a patently spatial automata-based simulation framework for urban modelling: geographic automata systems (GAS). Briefly put, GAS offer much of the functionality of basic automata, CA and MAS, but are specified in a spatially-specific fashion. GAS models are constructed from individual geographic automata (GA) building blocks and GAS are defined with geographic components:

- a typology of automata: GAS may comprise GA of different types, for example, spatially fixed and non-fixed.
- automata states: GA can be characterized with state variables, as is the case with all automata. Variables of uniquely geographic significance such as heading, velocity, progress from an origin, etc., may be introduced.
- general state transition rules: GA state dynamics are driven by general state transition rules, akin to those of the other automata discussed previously. However, in GAS complex relationships between rule-sets for different automata types can be specified, for example, stigmergetic associations between fixed and non-fixed automata.
- geo-referencing rules: these rules determine the placement of automata in simulated spaces. Automata can be georeferenced, following GIS approaches, directly using coordinate arrays. GA can also be georeferenced using indirect rules that point to automata in a variety of ways, even when they operate dynamically in space and time.
- movement rules: a dedicated rule-set for controlling the movement of automata is introduced in GAS, allowing for a plethora of fluid-like and migratory motions.

- neighbourhood rules: rather than relying on pre-defined neighbourhood topologies, GAS use a neighbourhood rule-set for determining adjacency between automata. This allows for the dynamic specification of neighbourhoods in space and time, as well as the linking of neighbourhoods to other properties of the model.

It is hoped that the development of GAS tools and software will advance geographic research in automata-based modelling. The following sections outline examples of GAS models applied to urban systems.

Hybrid automata for simulating urban development

The previous section offered a rationale for developing hybrid CA-MAS models in a GAS framework. However, there are also compelling reasons for constructing hybridisations between automata and non-automata models, such as traditional large-scale land-use and transport models. No single technique can capture the richness of an urban system, and it makes sense to use a diversity of tools to build virtual environments particularly when they are to be employed as planning support tools (Torrens 2002). Despite the flexibility of the automata approach, limitations remain.

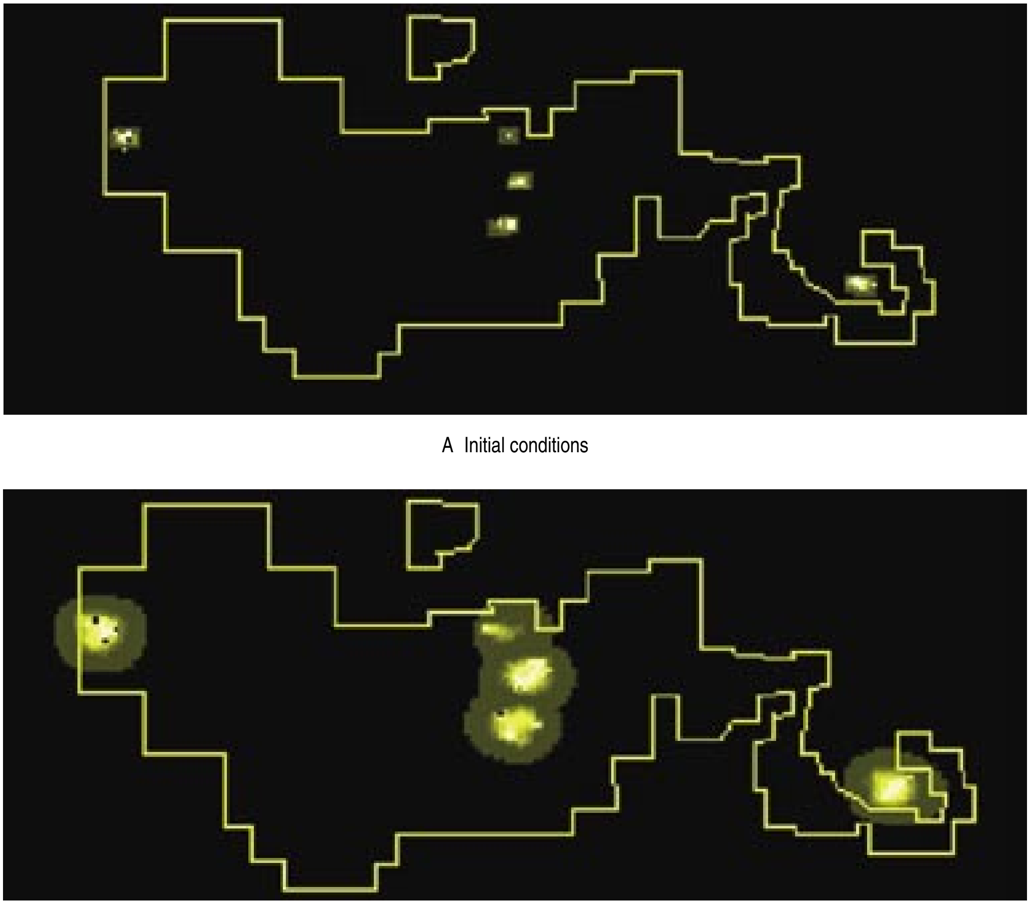
Automata are not particularly useful for simulating top-down processes, planning regimes being an obvious example. Also, automata-based models are closed systems. Closed systems are generally easier to model than open systems; there are simply less unknowns to worry about (Batty and Torrens 2001). However, system closure is an inappropriate assumption for most urban systems, which may be sensitive to a host of exogenous influences: national boom and bust cycles, regional inequalities, meteorological phenomena, etc.

The GAS approach goes some way toward resolving these problems, but in many instances it is necessary to interface automata models with exogenous simulations. Several authors have built models in this fashion, most commonly by introducing exogenously-specified growth rates, and in some instances tying automata models to a complicated chain of related simulation modules (Barrett et al 2001).

The author has been researching a general hybrid framework for modelling urban systems. The framework is hybrid in a number of ways. First, it is demarcated spatially with macro-scale elements of the simulated system relegated to exogenously-specified models. Second, this demarcation also serves to segment the simulation in terms of the direction of modelled processes: top-down events are handled largely exogenously, while bottom-up dynamics are simulated endogenously with automata-based tools. Third, the automata framework is itself a hybrid, along the lines of the GAS principles previously discussed. CA and MAS are fused in a unified and symbiotic manner. The next section discusses the specification of a pedagogic model designed to simulate general urban growth.

An application to urban growth

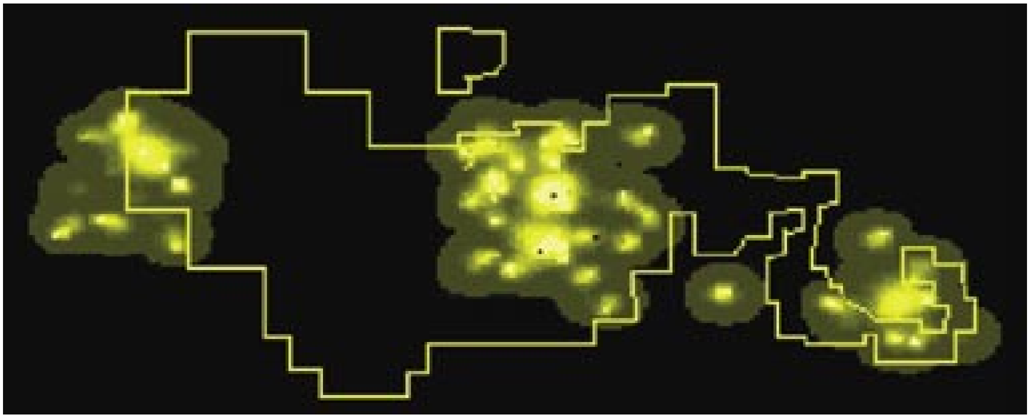
This section discusses the implementation of a hybrid automata framework applied to the simulation of general urban growth in a pedagogic environment. The model simulates the spatial evolution of an urban system over time with emphasis on the patterns of urbanisation that it generates and the pace with which the landscape



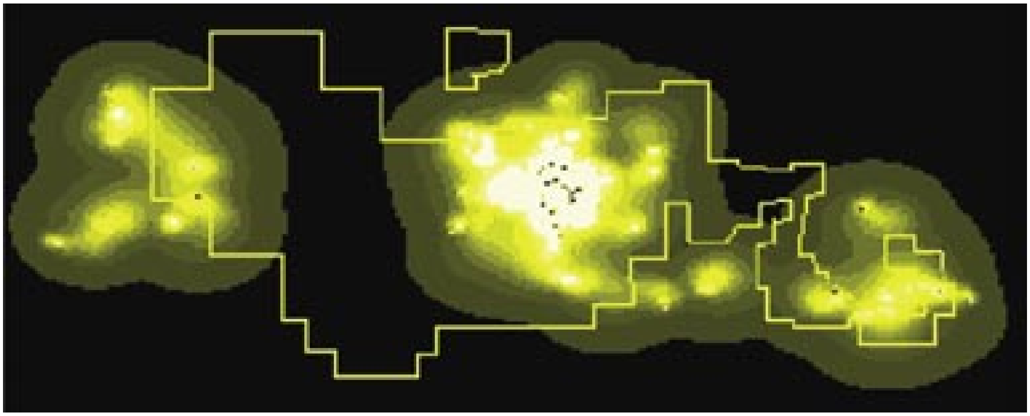
A Initial conditions

B After 50 iterations

Figure 5 The evolution of a hybrid model of urban growth from initial seed conditions



C After 100 iterations



D After 250 iterations

Figure 5 cont. The evolution of a hybrid model of urban growth from initial seed conditions

is urbanised. In the simulation, a city-system evolves from initial seed settlements, going through processes of compaction, poly-nucleation, infill, peripheral sprawl and densification of the central city (Figure 5).

The structure of the model is outlined in Figure 6. The model is divided into separate modules that simulate various components of urban growth at three levels of geography. At a macro-level, the simulation is supplied from the top-down with exogenously-defined growth rates. These serve as the general 'metabolism' for the subsequent simulation. Generally, growth is defined in terms of population, with population being delivered through various gateways that are coded into the model as state variables. Because the example illustrated in Figure 5 is pedagogic in nature, growth rates are defined with arbitrary values. However, in other implementations those rates have been linked to census data (Torrens 2002).

At a meso-scale, constraint data is introduced to confine the simulation dynamics within reasonable bounds. This is particularly relevant to location in the model, as automata simulations tend to be quite sensitive to initial conditions. In the example illustrated in Figure 5, the aforementioned gateways serve as constraints. The micro-level module then takes over, simulating the spatial distribution of this population locally around these gateways.

Micro-scale dynamics are simulated using hybrid CA-MAS. Agents, supplied to the micro-system from higher-level modules, are granted life-like functionality to mimic the location behaviour of developers and settlers in real urban systems. The micro-simulation framework differs from standard CA and MAS implemen-

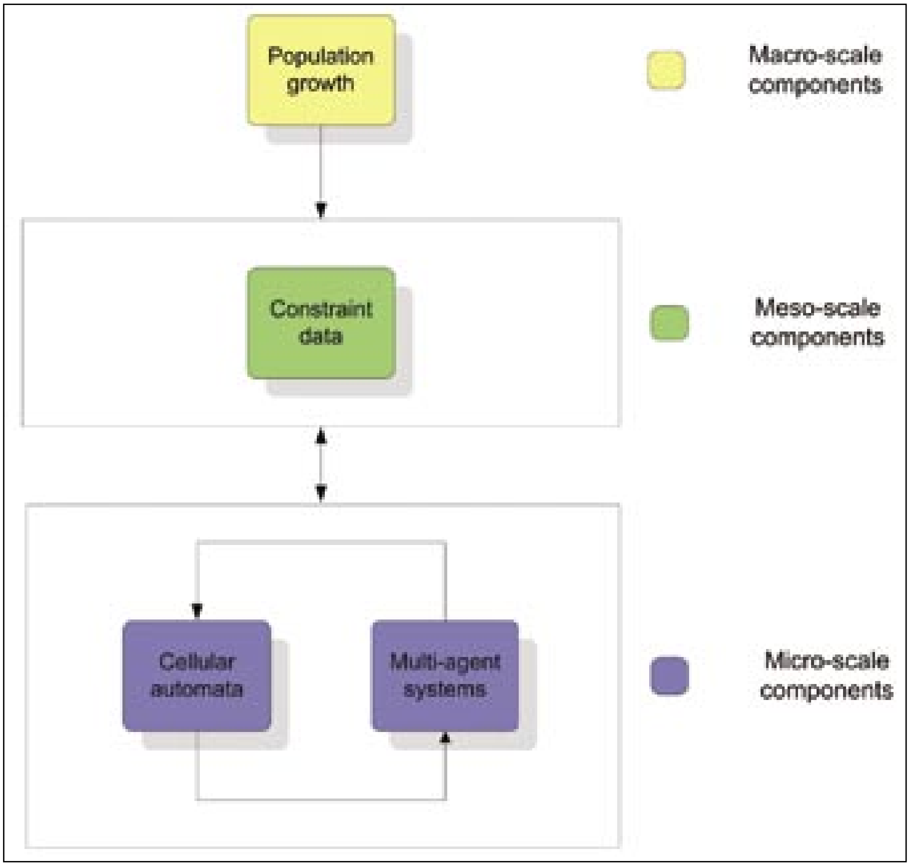


Figure 6 The structure of a hybrid urban growth model

tations. It is specified as a GAS, but retains the individual properties of CA and MAS, as well as offering new functionality. Essentially, agents are laid on top of a CA layer but granted functionality that enables them to initiate state transitions in CA directly.

Two types of automata are considered: mobile automata (developer-settler agents) and infrastructure automata (representing the landscape and built environment). These automata have a set of states that describe their characteristics in relation to the simulation. Mobile automata have state descriptors that represent their movement (moving, static); infrastructure automata have states describing their development (occupied, density of occupation). A series of georeferencing conventions have been introduced to track automata in the simulated space. Automata have a set of coordinates that register their position in the lattice; mobile automata are also aware of their distance from the seed gateway from which they originated.

For fixed automata, neighborhoods are defined as a static nine-cell Moore neighborhood. Mobile automata are free to roam the system, constrained by user-defined weights applied to the distance with which the movement rules are exercised.

A number of transition rules are used to mimic the spatial patterns of development and settlement that govern the dynamics of growth in an urban system. The movement of mobile developer-settler agents in the simulated space is exercised subject to several rules of movement. Mobile agents exercise a given movement rule, based on pre-defined probabilities that can be defined elastically to weight

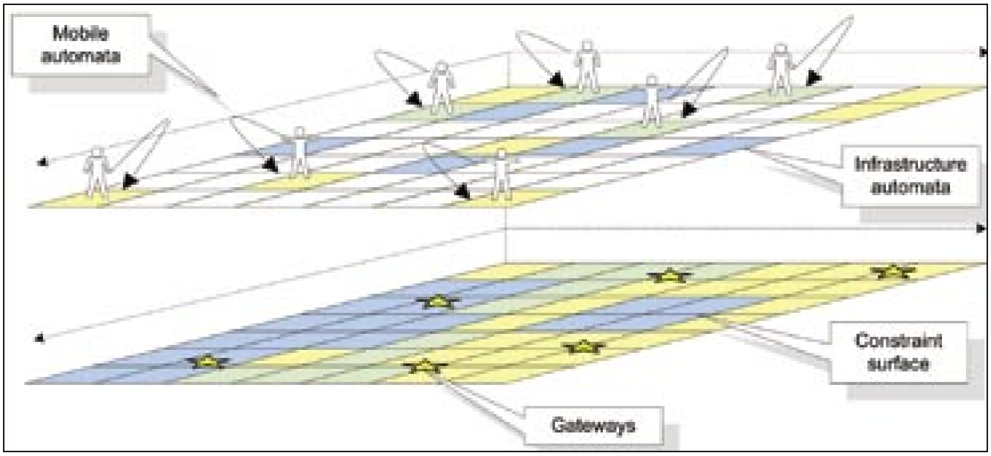


Figure 7 Cellular automata and multi-agent systems as information layers

certain patterns of behaviour over others. Once a movement has been exercised and an agent comes to a standstill, it develops and settles the infrastructure beneath it, that is, it exercises a single transition rule that allows it to access the state variables of the static infrastructure automata underneath, affixing a population value to the set of states that describe that cell. There are two local movement rules: one for development and settlement activity that occurs in the immediate vicinity of a given cell and another for that which takes place in a slightly extended area. Movement by leap-frogging is modelled, representing staggered, speculative and piecemeal settlement and development behaviours. Irregular movement is introduced to correspond to organic development and settlement in irregular linear patterns. A road-influenced movement rule assumes the gradual development of infrastructure to link clusters; accordingly, movement under this rule takes place in a linear fashion between nodes.

In addition to the rule that allows mobile automata to settle static automata cells, two other general state transition rules are included. A single state transition rule is available to static automata in the model. This simply diffuses population values between static cells in a nine-cell neighbourhood, thereby smoothing the distribution of settlement in the simulation. A random function that causes the population of given cells to decline in a given time step is also applied, related to the volume of development in the model overall. The larger a city grows, the greater the probability of population decline at a local scale. This function is designed to mimic decline due to overcrowding and associated problems of urban blight, etc.

Using these simple specifications, a realistic pattern of urban evolution can be simulated. The city-system starts from initial seed conditions, slowly extending and forming isolated clusters on the periphery which then coalesce over time and spawn additional clusters. As the simulation progresses, the original central seed sites, and their immediate periphery, grow denser and more compact while peripheral areas sprawl at lower densities and in a more fragmented pattern (Figure 5).

There is at least one obvious limitation to this approach, however. The rule-sets driving model dynamics generate realistic patterns of urbanisation, but their representation of the processes operating within the system are somewhat unrealistic. The rules mimic the end results of system processes—the spatial manifestation of the phenomena—rather than the root behaviour of the system. Certainly developers and relocating households build and settle in a fragmented manner, agglomerate around existing settlements, and so on. It is appropriate to represent the system in this fashion when simulating the spatial dynamics of urban growth in a general sense. However, there are fundamental behaviours that motivate these actions: economic impulses leading to agglomeration, lifecycle motivations for moving to the urban fringe, etc.

In light of this consideration, the author is also working to build microscopic behaviour modules at an atomic scale that will simulate the sort of principal behavioural components characteristic of real-world systems. Currently, a small-scale residential location (demand) model has been built and a complementary development decision (supply) model is also planned. The specification and application of these models are beyond the scope of this chapter, but are reported elsewhere (Torrens 2001, 2002).

5 Caveats for the future

Thus far, the discussion about the use of automata tools in urban simulation has been reasonably sanguine, however, the optimism comes equipped with some caveats. A major issue of concern with models of such complexity and resolution is one of validation. Most urban simulations, whether developed as pedagogic tools or not, are built in applied contexts. A model must be applied in practice in order to test its validity. For previous generations of aggregate urban models, well-understood statistical tools are available for assigning predictability. Automata models cannot be thought of as predictive tools in the same sense simply because they are so non-linear in nature (Batty and Torrens 2001). They are better thought of as game-playing environments or 'tools to think with'. That is not to say that they are incapable of validation—a lot of research effort is being expended on developing validation techniques (Benenson and Torrens 2003). One approach is to validate the patterns that the simulations generate. However, more process-oriented mechanisms are needed.

The next obvious caveat concerns data. High-resolution models require fine-scale data. However, data on individual-level urban objects are not always available. Other authors have developed synthetic population generation routines for automata models (Barrett et al 2001), but data availability is still a concern. Privacy issues associated with the use of micro-level data should be considered. From a computational standpoint, processing is another concern. Ironically, advances in computer software and hardware catalysed the popularity of automata for simulation, but processing power and bandwidth remain problematic when compiling and running simulations. The experiments described in this chapter have a modest volume of simulated entities (a few thousand). To be useful as planning support tools, models with millions of interacting entities may need to be built. At these scales, most desktop machines begin to smoke around the exhaust fan. Elsewhere, authors are experimenting with distributed processing over networks of machines or Beowulf clusters, but the techniques have yet to trickle down to popular use.

Finally, the issue of practical application must be raised. If automata models are so good, why are they not widely used? The answer to that question is that they are, for the most part, still academic in nature. The field is in its infancy and the technology lags behind its practical application. Nevertheless, automata models of this nature have seen successful application in pilot studies (Barrett et al 2001) and their future as planning support tools looks promising.

Despite these caveats, automata modelling has the potential to contribute to spatial analysis, urban studies and urban planning in significant ways, both in terms of new tools for model development, new understanding of spatial systems and the virtual evaluation of hypotheses and public policy.[1]

Footnotes

1 The GAS idea was developed, collaboratively, with Prof. Itzhak Benenson at Tel Aviv University. He deserves at least 50% of any complaints that its description prompts.