A Generic Spatial Model Based on Vector-Agents

Yasser Hammam, Antoni Moore, Peter Whigham, and *Claire Freeman

Spatial Information Research Centre
Information Science Department
University of Otago, Dunedin, New Zealand
Tel: +64 3-479-8301
FAX: +64 3-479-8311
Email: {yhammam}, {amoore}, {pwhigham}@infoscience.otago.ac.nz

*Geography Department
University of Otago, Dunedin, New Zealand
Tel: +64 3-479-8785
FAX: +64 3-479-9037
Email: cf@geography.otago.ac.nz

Abstract
The work reported here has been motivated by the need for a generic spatial model to overcome the limitations of Cellular Automata (CA) regarding the rigid square-cell structure and limited neighbourhood configurations. A novel approach for spatial modelling technique is developed: the "vector-agent" in which the individual entity is represented by their real geometric boundaries (which can change over time) beneath an agent modelling structure. We show in this paper how the theory behind CA and agents can be combined to produce a generic and dynamic agent based on the vector data structure. This new paradigm has extended capabilities over the Geographic Automata (Torrens and Benenson, 2005) in terms of CA disunity and the abstraction of non-fixed-objects. Through computer simulation, different techniques and algorithms have been derived achieving a high degree of representational realism for a variety of phenomena.

1. Introduction
Long debates have been articulated about the value of using Cellular Automata (CA) for spatial modelling, especially for a complex spatial phenomena such as the city (Batty, 2000, 2001; Torrens and O'Sullivan, 2000a). In particular, the representation of space as a collection of regular square cells is regarded as a limited assumption in a spatial simulation domain (Benenson and Torrens, 2004a; Benenson and Torrens, 2004b). Initial research towards an irregular CA have used Voronoi diagrams instead of the fixed and regular neighbourhoods of CAs (Shi and Pang, 2000). However, these limited configuration ignore any distance function, which is a part of any dynamic spatial process (White and Engelen, 2000). Various alternative approaches have been investigated, including Delaunay triangulation (Semboloni, 2000) and planar graphs (O'Sullivan, 2000, 2001) as the basis for neighbourhoods. A question arose from the consequences of this large number of modifications to the basic CA framework in the phenomena being modelled, "Do these modifications lead the spatial model towards

the needed degree of realism?". It should be made clear that these extensive modifications may move the attention of model developers away from exploring the idea behind the phenomena being processed and how the systems function, and lead to a more chaotic model structure (Torrens and O'Sullivan, 2000b). CA, even modified-CA, cannot therefore truly model real world entities. A more flexible and dynamic spatial model with no supplements or modifications is crucial.

The notion of spatial-agents has been precisely investigated (Rodrigues, 1999), where agents interact spatially in coordinate-space. Such agents claim the capability to position and react to stimuli in a spatial domain. However, most of the research have combined Multi Agent Systems (MAS) with CA as a model framework (Barros, 2003; Batty *et al.*, 2003; Haklay *et al.*, 2001). Although, such integration has advanced the model mobility and flexibility, this model framework still exploits a strict CA regular configurations (Torrens and Benenson, 2005).
More recently the term automata has emerged from the CA model and employed independently as an autonomous object in the spatial simulation domain (Benenson and Torrens, 2004b). The most prominent research in this area is the Object-Based Environment for Urban Simulation (OBEUS) (Torrens and Benenson, 2005). The notion of this new automata is based on a set of spatial-referencing rules for situating automata in space with more flexibilities in defining the neighbourhood rules instead of a fixed neighbourhood as in CA. This has been done in space formed by square-cells as a testing environment and justified in existing geo-spatial database utilising two type of agents; fixed-agent (i.e. land parcel) and non-fixed agent (social actors, i.e. householders, landlord…etc.).
However, the question still remains; since many phenomenon objects are subject to irregular change in nature (like-organisms in biology or urban pattern in city), "how do agent systems reveal these dynamic objects in a more realistic fashion and represent the interaction among these types of non-fixed agents"? This cannot be conceived using CA-agents. The regular partition of space as a conceptual basis should be substituted by another approach.

This paper introduces a generic spatial modelling technique: the "vector-agent", which is based on irregular vector data structure and influenced by the agent oriented paradigm. The vector agent provides more realism for representing individual object by their real geometric boundaries (which can change over time). The merit of such an approach is that the vector-agent can be a direct abstraction of real world entity allocating itself in the spatial domain, not virtually trying to disperse its entity to a fixed-object (Torrens and Benenson, 2005).

The paper outlines an advanced development to the model structure based on an early hypothetical test of vector-agents applied to von Thunen's theory of agricultural land use for model verification (Hammam *et al.*, 2004). The novel characteristics of vector-agent and how these can be supported by the notion of an agent-system will be discussed in section 2. Section 3 illustrates a review of topological relations among agents and algorithms for constructing geometry. Model implementation and experimental results are presented in section 4. Finally, section 5 draws some conclusions and describes future directions for this on going research.

2. Agent-Based Simulation and Vector-Agent Paradigm

The thesis of this paper is that the use of the agent-oriented paradigm is to represent dynamic individuals embedded in space, and who are able to interact spatially derived by goal-oriented behaviour. This not only support the views presented later (especially in Luck *et al.*, 2003), but it provides additiona l advantages which can exist with the vector-agents paradigm. These can be summarised as follows:

- *Representing any phenomenal entity by irregular vector data structure*: the agent has therefore an advantage of being more realistic, flexible for representing real world features, such as buildings, roads ….etc., not generalised square-cells. The agent has also advanced interaction capabilities with variant topological relations.
- *The entity is abstracted so that it is able to define its own location in space with dynamic rules*: this can overcome the limitations of a restricted neighbourhood exhibited in CA. The agent can allocate itself randomly or regarding attraction and repulsion forces generated from the surrounding environment.
- *The agent is born with a nondeterministic shape boundary*: this advances the capability of agent to construct a rule-based shape with increasing structural complexity, rather than assign a new entity to an object with fixed boundary (Torrens and Benenson, 2005), or allocates itself in space with static shape to just interact spatially with other agents (Rodrigues, 1999). This can be achieved using different operators for assigning and changing the object boundary such as midpoint-displacement, line-displacement,…..etc. The vector-agent provides a flexible mechanism for meeting a certain threshold and satisfying the agent's goal using more complex operations such as, a generalisation technique (for achieving a desired fractal dimension), or split (to meet a certain size).
- *Since the agent is an abstraction of a real-world entity in the simulation domain, the agent's goals are consequently abstraction of the entity's properties*: the agent can therefore maintain its identity derived by that entity's interactive behaviour in a spatial domain.

Here, it is worth noting the differences between the agent-oriented and object-oriented paradigms, and the notion of an agent system, in order to justify the above characteristics of vector-agents. Objects inhibit the behaviour (methods) and fixed roles required to implement the functions needed and do not usually change roles once the application has been deployed. The interactions between objects are explicitly defined. Objects are used by other objects to perform actions, which in turn do not initiate actions of their own choice. Agent systems make use of concurrency, both inside individual agents and certainly among different agents. Sequencing control through a set of agents cannot provide a truly agent-oriented system. Therefore, the collection of agents have to be running simultaneously (via multithreading), not sequentially. Each agent has internal goals as well as roles. However, agents can change roles dynamically as the application runs, which is not a property of a standard object-oriented system (Russell and Norvig, 2003).

The definition of agent may be considered as follows:

"*An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors*" (Russell and Norvig, 1995).

"*Intelligent agents continuously perform three functions: perception of dynamic conditions in the environment; action to affect conditions in the environment; and reasoning to interpret perceptions, solve problems, draw interfaces and determine actions*" (Hayes-Roth, 1995).

"*An intelligent agent is generally regarded as an autonomous decision-making system, which senses and acts in some environment*" (Wooldridge, 1997).

Considering the previous definitions, we argue here that three common properties can be observed; the agents must sense: the environment that surrounds the agents; agents operate without the direct intervention of humans; and agents interact with other agents, which exhibit a goal-directed behaviour to have some kind of control over their action. This argument is supported by (Luck *et al.*, 2003), who provides a rigorous framework in defining agent in terms of "entities' hierarchy". They propose a four-tiered hierarchy comprising entities, objects, agents, and autonomous agents (Figure 1). In their essence, entities simply provide a way to denote components in the world before they can be at any recognisable structure. Objects can then be defined to be things that have abilities and attributes. Similarly, agents are just objects that are useful, where this usefulness is defined in terms of satisfying some goals. In other words, an agent is an object with an associated set of goals. Lastly, autonomous agents are just agents that can generate or adopt their goals in response to current environmental conditions. They concluded the framework as "*…if there are attributes and capabilities, but no goals, then the entity is an object. If there are goals but no motivations, then the entity is an agent. Finally if neither the motivation nor goal sets are empty, then the entity is an autonomous agent*".

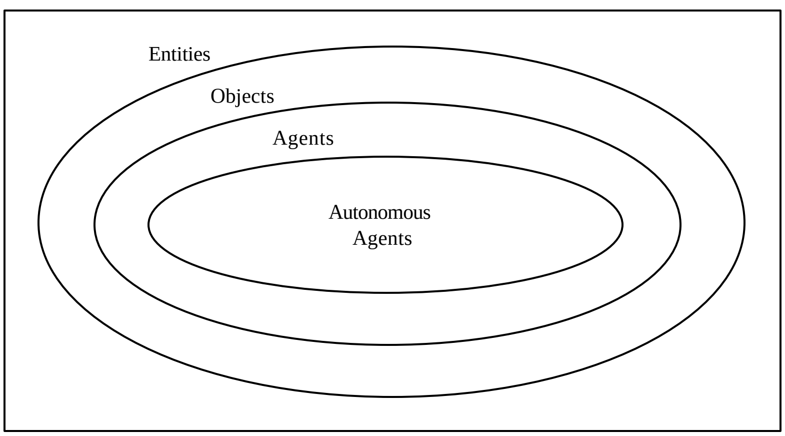


Figure 1. Entity Hierarchy overview (source: Luck *et al.*, 2003)

In terms of spatial simulation, most types of simulation involve some spatial context. A typical example is the fact that generally a simulation is representing an environment that exists somewhere in space. The objects embedded in the environment will be spatially located (Rasmussen and Barrett, 1995). The concept of a Spatial-agent has been defined concerning the framework provided by (Russell and

Norvig, 1995) as a template with spatially-aware properties through reinforcement learning methods (Rodrigues *et al.*, 1998; Rodrigues, 1999).

Bearing in mind the above characteristics; the agent properties with the conclusion drawn by (Luck *et al.*, 2003); and the notion of spatial agent (Rodrigues, 1999), we state that a "vector-agent is defined by goal-oriented Euclidian geometry who is able to evolve and change it's own shape keeping the entity properties while interacting with other agents using a set of rules in the Euclidean plane".

3. Irregular Shapes and Randomness in Fractal Construction

As mentioned, the main objective is to construct an irregular geometry class agent. Before considering the implementation of such an agent class (section 4), it is necessary to know the specific characteristics of the geometry relationship when inserting shape agents in a spatial environment, and the mechanisms for constructing such geometry in a spatial simulation domain.

3.1. Space and topological relationships

Topology is a branch of geometry, concerned with the set of geometric properties that remain invariant under scale transformation (Laurini and Thompson, 1992; Worboys and Duckham, 2004).

All familiar topological properties can be defined: meet, joint, disjoint, intersect, overlap…. etc., which define the topological spatial relations to polygonal areas in the plane (Egenhofer and Franzosa, 1991). These spatial relation properties will derive the relative neighbourhood relationships of geometries positioned in space. Subsequently, by increasing the model complexity, other topological relations may be defined, not with respect to sets of adjacent objects or entities, but as a region of space (i.e. spatial proximity) based on a certain distance from the object under question More variant relationships with respect to the exterior element direction can also be defined (Egenhofer and Franzosa, 1994).

3.2. Shape construction and evolution in a system environment

In many cases world phenomena carries some fractal characteristics. For example, urban patterns, landscape features, coastline…etc. obeys some fractal laws. Self-similarity seems to be one of the fundamental fractal geometric attributes (Peitgen *et al.*, 1992). However, some man-made objects such as building blocks have no apparent self-similarity, but still the objects carry some statistical fractional sense when magnified. Generally, many natural shapes possess the property that they are irregular in their boundary. Methods for generating models of shapes with prescribed fractal dimension with exact self-similarity, are not perceived as realistic models. The reason lies in their lack of randomness (Peitgen *et al.*, 1992). Therefore, one of the consequences is that it is impossible to assign only the self-similarity fractional law for abstracting objects which have such irregularity in nature. We argue here that in order to create more natural shapes, a randomization process must be utilized that is closer to stochastic shape evolution. One such dynamical process is Brownian Motion named after Robert Brown, regarding his work on the random movement of microscopic particles (Rucc, 1994). Pure Brownian motion (Bm) can be defined as tracing out the total random walk distance travelled by a point in the plane in appropriate units of time resulting in a *Gaussian or bell-shaped* distribution from the initial location The most popular way to produce Brownian motion is called random midpoint displacement (Kenkel and Walker, 1996). This method has received interest from those concerned with computer graphic simulation and many other disciplines especially in simulating natural fractal shapes such as stochastic 3D modelling of terrain (Goodchild and Mark, 1987). Simple Bm has been generalised to derive the fractal dimension for varying phenomena by introducing Hurst exponent *h* as fractal parameter specifying the roughness of an object (Voss, 1988). It has been proved that $D = 2-h$, where *D* is the fractal dimension. With this relation the fractal dimension *D* of regular Brownian motion ($h = 0.5$) would be *1.5*. When $h < 0.5$ the shape is rough and when $h > 0.5$ the shape is smoother.

In practice, the Bm can be achieved in Euclidean space by considering a line segment as an initiator with repetition of recursive subdivision by midpoint displacement. This interpolation has two forms; displacement of the middle point along the axis of the line segment or displacement of the middle point along the segment perpendicular bisector (the y-axis), (see Figure 2. c, d, e). A generalised algorithm is given by the formula in Equation 1:

$$\mathbf{Y_{new} = 0.5\,(y_1 + y_2) + \mu o_0 2^{-lh}} \quad (1)$$

where ($\mathbf{y_1}$, $\mathbf{y_2}$) are the start and end points of the line segment being subdivided along the y axis, $\boldsymbol{\mu}$ stands for a random number from Gaussian (normal curve) distribution, $\mathbf{o_0}$ is the standard deviation of Gaussian curve which is equal to 1, **l** is the level of recursivity, and **h** is the fractal parameter mentioned above specifying the roughness of an object (Laurini and Thompson, 1992).

In summary, the Brownian motion will be extended as a base for constructing the irregular geometry in our vector-agent model. However, adaptations for giving the geometric shape more freedom to evolve stochastically have been performed and will be illustrated in the following section for model implementation.

4. Model Implementation and Experimental Results

This section describes a testing environment for the methodology discussed in sections 2 and 3 The main concept is to create a simulation, which involves a spatial environment with elements positioned in it. These elements are agents with adaptive capabilities that enable them to interact and take actions with spatial implications.

4.1. Model elements

The main focus of our simulation model in this stage is to demonstrate that the vector-agent is capable of initiating its own shape, which can change over time with a number of modified parameters using different operators associated with variant system probabilities.

The simulation is therefore composed of the following:

- Continuous vector-space (coordinate-space) with predefined *x*, *y* coordinates. This is a passive or static object that will never change its state.
- The shape class, an agent, searching for unoccupied space that fulfils its preferences in interacting with other agents.
- The neighbourhood, a rule-based class, which the shape agents use to extract the current interaction rule with other agents. This governs the previous topological relations addressed in section 3.1.

- The shape-behaviour class, containing three different algorithms to be conducted by the shape-agent in each execution of the simulation. These can be summarised as follows:
    - Splitting the shape edge with new point generated by Brownian motion algorithm. The point is being displaced assigned randomly along the edge (Figure 2. e, f). The displacement of this point is thereby allocated into the new position along the segment bisector with random angle (*0 < ? < 180*).
    - Moving the whole edge on new coordinates with a certain distance (Figure 2. g, h).
    - Moving one of the shape vertices outside the shape-boundary with random distance into a new position (i.e. new *x, y*) (Figure 2. i, j).

The last two rules have been suggested to produce different shape with different sizes, rather than employing Bm algorithm solely, and to provide flexibilities for more realistic shape evolution. Providing evidence for generating unrestricted shape boundary on current stage of model implementation is crucial. In a generic sense, the system must be able to deal with any phenomena in the simulation domain. Therefore, the model was elaborated in such a way that the shape change algorithms are as simple and complete as possible.

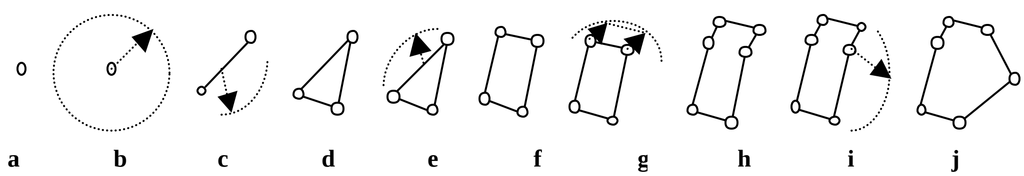


Figure 2. How shape starts and evolves in spatial simulation domain: (a) starting by random point, (b) allocating second point by random x, and Bm around the y axis, (c, d) applying the random midpoint displacement and accomplishing closed polygon, (e, f) choosing any edge randomly and applying the midpoint displacement, (g, h) edge displacement, (i, j) vertex displacement.

4.2. Simulation results

The simulation starts by executing the desired number of agents. Every agent starts by exploring any available empty space and allocating itself randomly (i.e. random *x, y*). Subsequent points are generated based on Bm algorithm to accomplish a closed geometry (polygonal shape) (Figure 2. a, b, c, d). Thus, the shape-agent begins to evolve conducting one of the previous algorithmic operators in the shape-behaviour class (Figure 4). Here it should be made clear that the agent is born with equal opportunity for observing these operators (i.e. 1/3 probability for midpoint displacement…..etc.). This is the case in the primary stage in the simulation runs. According to the neighbourhood role set up before execution, any agent has consequently a chance to change the opportunity for observing one of the shape evolution techniques based on the current situation with other agents. This may be the evidence for changing interactive behaviours regarding the agents' perception in the environment, and decide whether to move in relation to other agents.

Figure 3 shows a typical simulation run for 600 time steps. Four agents utilise an "*overlap*" topological relationship with no restriction of shape size. At 0 time step the agent is observing the space to allocate its new point with a subsequent of other two points (t_2). After 3 time steps where the agent achieved the third point and closed polygon, it is the time for the agent to start evolving. As the simulation progress, the agent is conducting one of the operators described above utilising the neighbourhood relationship. By the time 200 and 400 a hole has formed in the latest polygons, which can reminiscently anticipated many real-world geographic phenomena, such as cities.

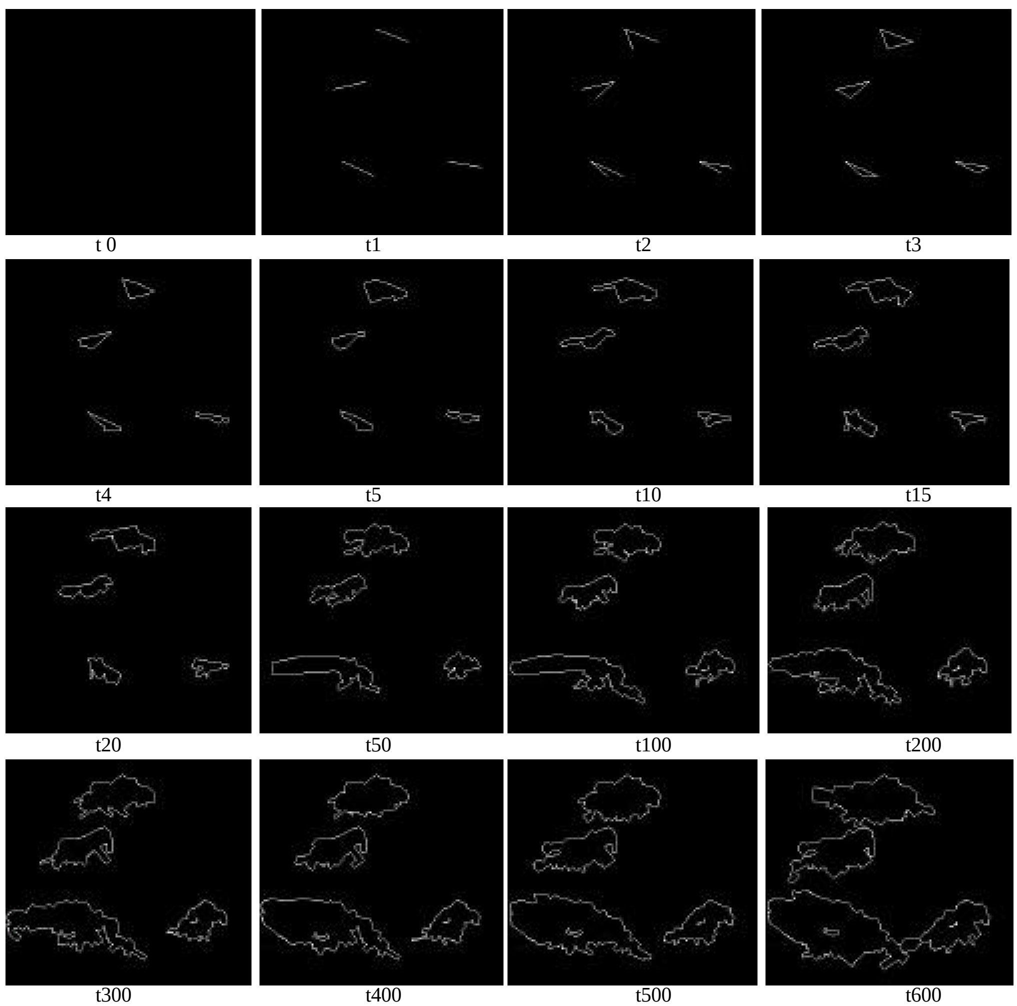


Figure 3. Simulation result for first 600 time steps
with *overlap* topological relationship

Twelve different combinations of algorithmic operators have been tested associated with a variant of system's probability (p). Fractal analysis in different combinations has been done to prove that vector-agents can successfully produce any desired shape regarding a system demand. According to insignificant change in shape size after

1000 time steps observed from a number of different simulation executions, such time has been set up for investigating all operators behaviours. Table 1 Figure 5 illustrates the fractal dimension (D) for a single agent in different time steps. As the agent starts to evolve after accomplishing a closed polygon, t_4 indicates the starting evolution time. Over the simulation time, the agent varies the D parameter regarding the operator/operators being used. By using only the midpoint displacement with initial D parameter generated from Equation 1 (in this test 1.5 was the initial parameter, i.e. h = 0.5), the fractal dimension does not exceed 1.512. This is because the Bm algorithm controls the degree of roughness during the simulation. However, this type of restriction has not successfully controlled the simulation when deploying with other operators. One possible cause for this might be the altering of the other two operators' parameters.

Obviously, the fractal dimension increases sharply to more than 1.668 at 100 time steps while the system deploys the edge displacement operator with a higher degree of probability (Figure 5.b, g, k). A reason for such occurrence is due to the shape, which expands its boundary with two new vertices in each time steps. This operator is distinct from the other two operators where only one new verte x is claimed.

As mentioned, the model should be able to produce different types of shape with various sizes in order to easily behave like any phenomena abstraction. Figure 5 describes the trends over time of shape size associated with different combinations of algorithmic operators. While deploying an edge displacement solely or at any combinations, dramatic change can be noticed in shape size (Figure 5. b, g, k). Alternatively, applying the midpoint or vertex displacement individually or with a higher degree of probability, a slight change can be generated (Figure 5. a, c, d, h, i).

Table 1. Fractal dimension associated with different operators' probabilities as simulation result for first 1000 time steps

Operators	**Initial fractal dimension (D)**	t_4	t_{100}	t_{200}	t_{300}	t_{400}	t_{500}	t_{600}	t_{700}	t_{800}	t_{900}	t_{1000}
$M_{p=1}$	1.5	1.325	1.358	1.504	1.461	1.434	1.478	1.498	1.491	1.512	1.459	1.475
$E_{p=1}$	Null	1.315	1.668	1.657	1.626	1.642	1.637	1.643	1.652	1.654	1.652	1.657
$V_{p=1}$	Null	1.311	1.439	1.422	1.407	1.406	1.401	1.422	1.429	1.432	1.447	1.445
$M_{p=0.33}$, $E_{p=0.33}$, $V_{p=0.33}$	1.5	1.327	1.581	1.663	1.766	1.798	1.781	1.791	1.785	1.774	1.795	1.782
$E_{p=0.5}$, $M_{p=0.5}$	1.5	1.319	1.715	1.726	1.741	1.737	1.726	1.731	1.731	1.729	1.729	1.725
$M_{p=0.5}$, $V_{p=0.5}$	1.5	1.274	1.634	1.675	1.694	1.694	1.693	1.691	1.691	1.693	1.686	1.681
$E_{p=0.5}$, $V_{p=0.5}$	Null	1.316	1.662	1.644	1.661	1.679	1.698	1.711	1.718	1.727	1.761	1.753
$V_{p=0.9}$, $E_{p=0.1}$	Null	1.223	1.583	1.571	1.591	1.579	1.606	1.659	1.660	1.681	1.701	1.723
$V_{p=0.9}$, $M_{p=01}$	1.5	1.323	1.483	1.471	1.491	1.479	1.506	1.559	1.595	1.581	1.601	1.62
$M_{p=0.8}$, $E_{p=0.1}$, $V_{p=0.1}$	1.5	1.244	1.636	1.689	1.707	1.688	1.711	1.744	1.731	1.734	1.746	1.748
$E_{p=0.8}$, $M_{p=0.1}$, $V_{p=0.1}$	1.5	1.292	1.668	1.668	1.706	1.731	1.747	1.751	1.721	1.789	1.796	1.806
$V_{p=0.8}$, $E_{p=0.1}$, $M_{p=0.1}$	1.5	1.314	1.559	1.621	1.641	1.644	1.654	1.649	1.722	1.783	1.791	1.831

M = Midpoint displacement, E = Edge displacement, V = Vertex displacement, $_p$ = the system probability

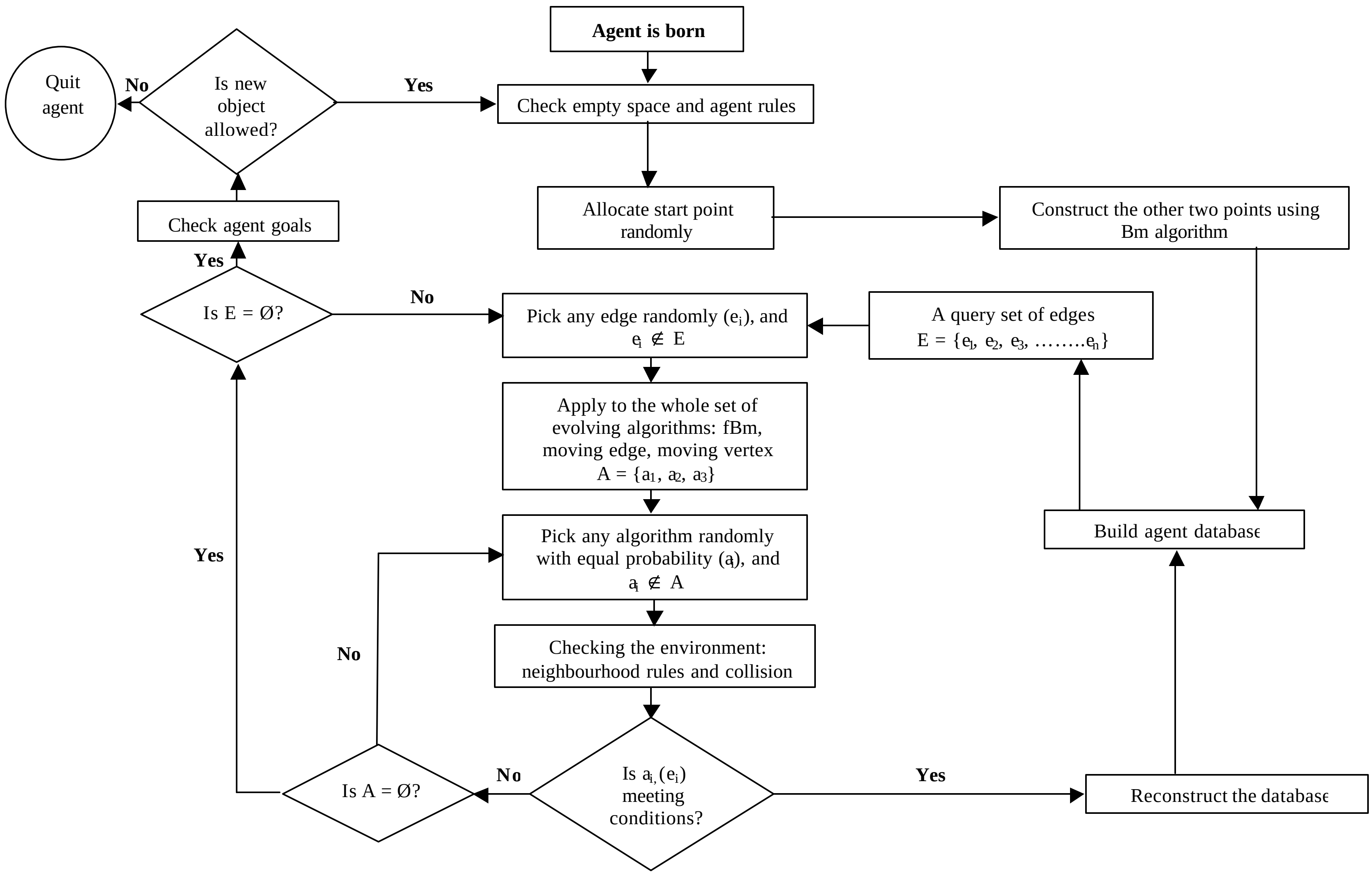
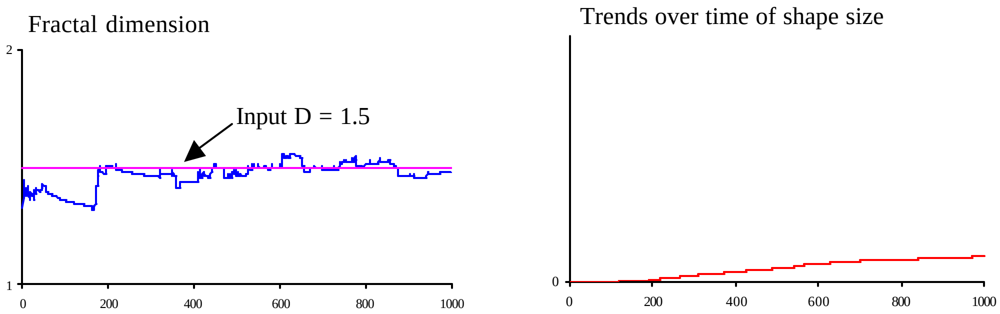
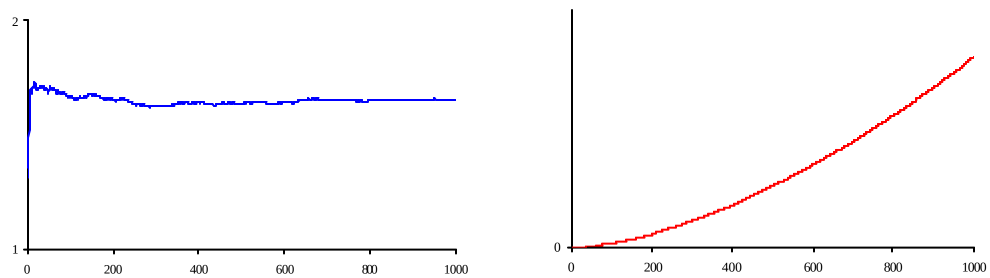


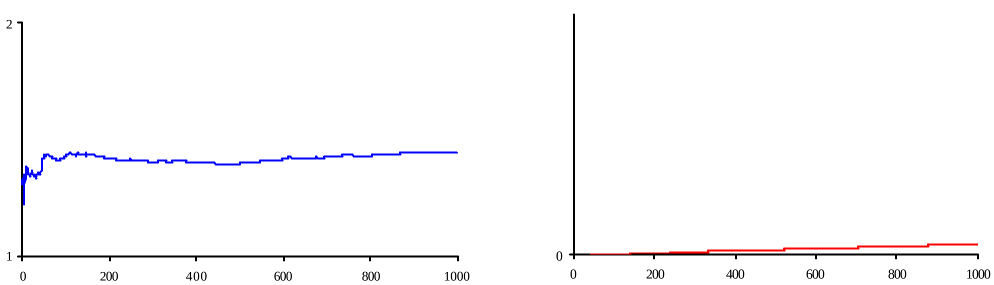
Figure 4. Schematic representation of agent behaviour algorithm



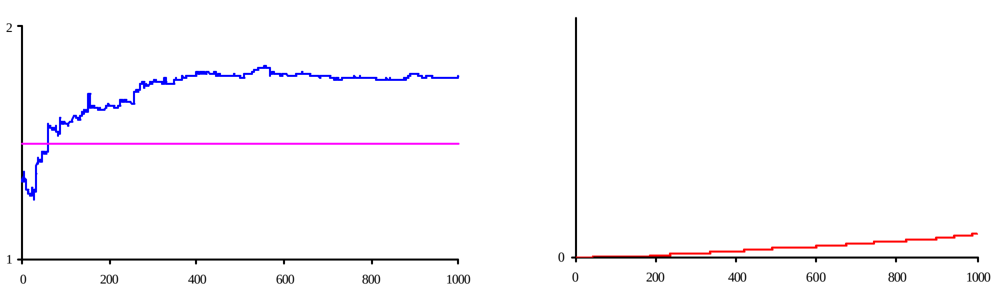
a. Midpoint displacement p = 1



b. Edge displacement p = 1

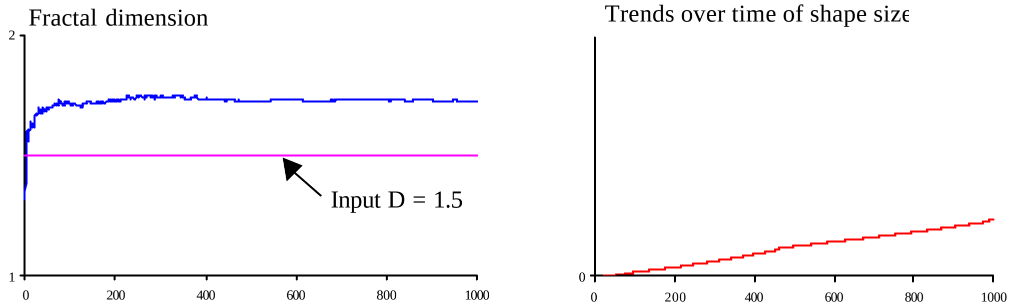


c. Vertex displacement p = 1

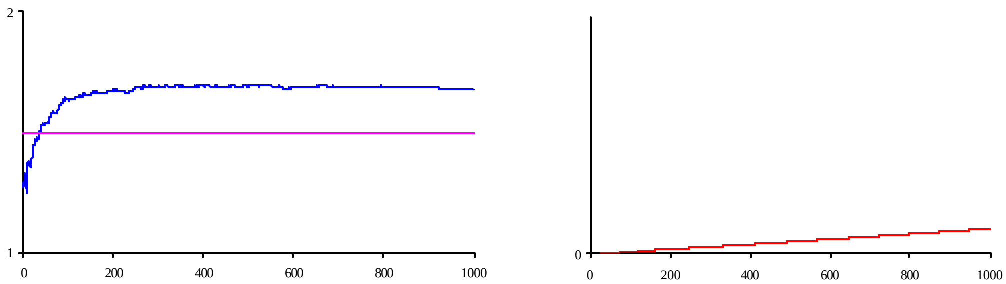


d. Midpoint displacement p = 0.33, Edge displacement p = 0.33, Vertex displacement p = 0.33

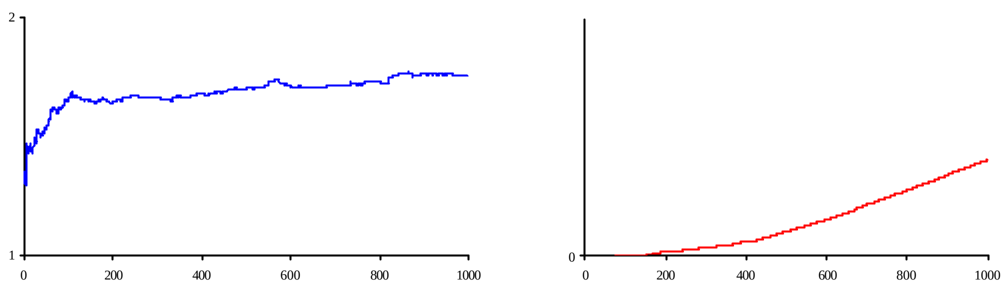
Figure 5. Fractal dimension and the trends over time of shape size generated by applying different operations with different probability (p) as simulation result for first 1000 time steps



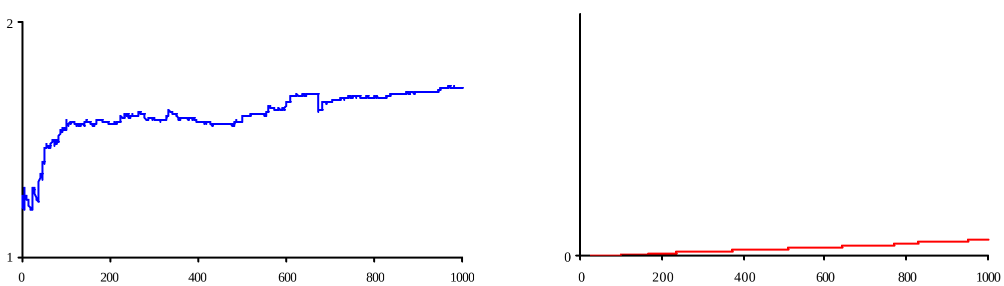
e. Edge displacement p = 0.5, Midpoint displacement p = 0.5



f. Midpoint displacement p = 0.5, Vertex displacement p = 0.5

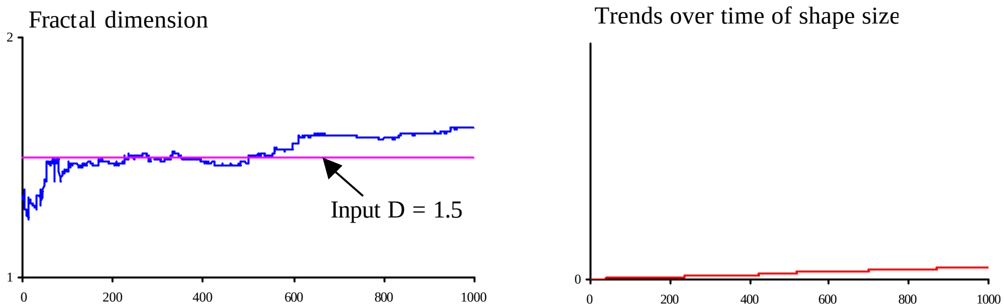


g. Edge displacement p = 0.5, Vertex displacement p = 0.5

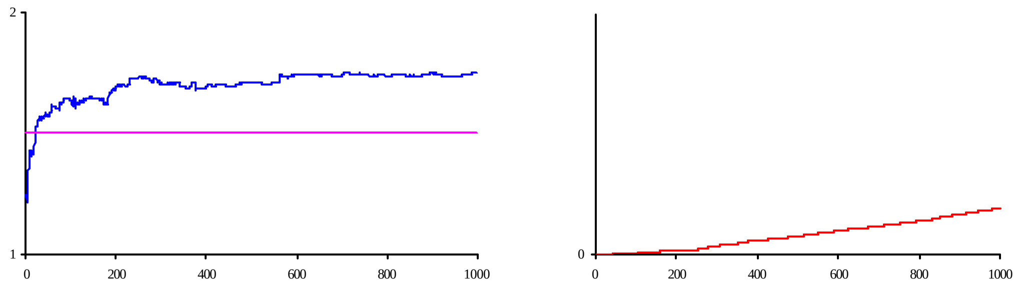


h. Vertex displacement p = 0.9, Edge displacement p = 0.1

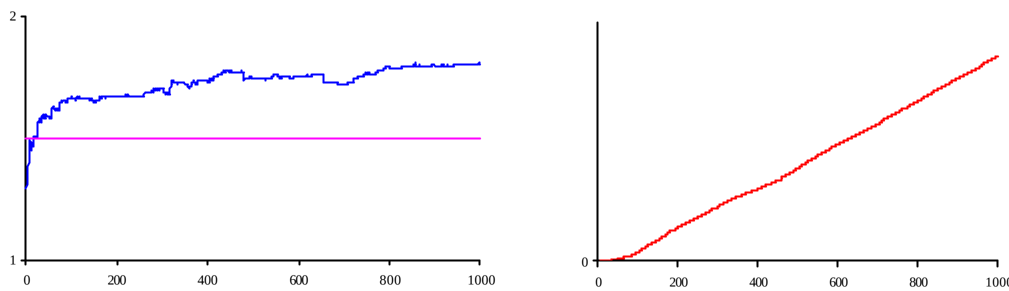
Continue Figure 5.



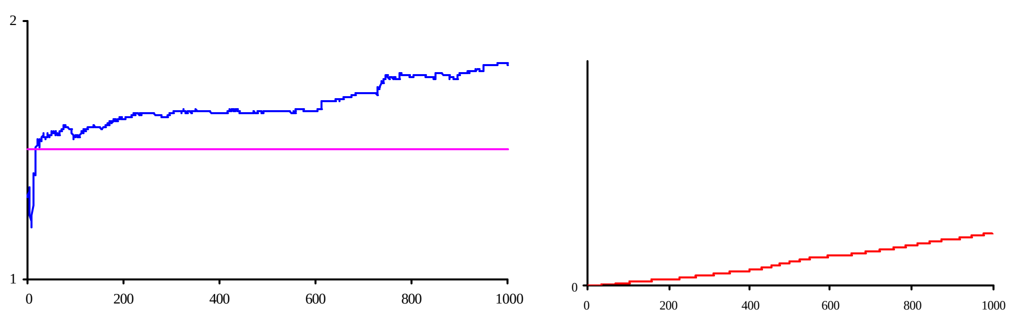
i. Vertex displacement p = 0.9, Midpoint displacement p = 0.1



j. Midpoint displacement p = 0.8, Edge displacement p = 0.1, Vertex displacement p = 0.1



k. Edge displacement p = 0.8, Midpoint displacement p = 0.1, Vertex displacement p = 0.1



l. Vertex displacement p = 0.8, Edge displacement 0.1, Midpoint displacement p = 0.1

Continue Figure 5.

We can summarise our findings in the way that city as a complex phenomena is likely to be the most prominent approach for our model justification. The similarity between our model output and the land parcel extracted from real-world data can be visually proved (Figure 6). Where more evidence can be claimed, the shape generated from model simulation is noticeably similar to a part of the model output generated from CA for simulating an urban growth pattern (Figure 7). These primary comparisons and observation can ground some of the vector-agents hypothesis, which is the ability to simulate irregular objects and manipulate their geometric boundary. These in turn have resulted in generating different shapes that can easily interpret different types of complex phenomena as in city.

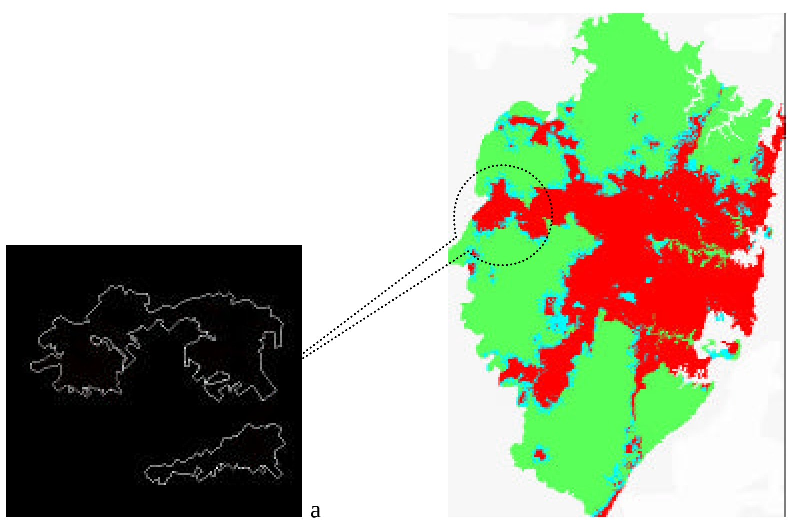
It is important here to state that carrying out these comparisons are just to proving some visualisation similarities, nothing more than that. No entity states, transition rules, or time scale, have been set up yet, which it will be the next stage in the model implementation while is being calibrated with a real-word data.

A sample of land use parcels for Swindon, south central England, with average fractal dimension 1.570 (source: Batty and Longley, 1994)

Disjoint *overlap* *overlap* *meet*

Vector-agents in different simulation output with various topological relationships and average fractal dimension 1.581

Figure 6. Comparison between vector-agents simulation output and land use parcels



a b

Figure 7. Comparison between vector-agent simulation output with non-restricted shape size and *overlap* topological relationship (a), and simulated urban scenario of Sydney for urban development generated by CA (b) (source: Liu, 2001)

5. Conclusion and Future Research Direction

The aims of this paper have been to explore a generic spatial model using vector-agents. Current spatial modelling techniques are limited due to the constraints of Cellular Automata. These limitations and the lack of achieving satisfactory model output led to introducing a new class of geographic automata (Torrens and Benenson, 2005). The general concept of this new automata has been derived by merging CA with multi-agent systems. While the agents are deployed with a fixed-object, and no interaction between them, limitations still remain in this new automata class. To fill this gap our generic model has been implemented with the power of the notion of the agent-oriented paradigm (Luck *et al.*, 2003) and the concept of spatial-agents (Rodrigues, 1999). The notion of a vector-agent is capable of being born, evolve, and interact spatially as an irregular geometry in coordinative space. Whereas, the geometric shape is constructed based on the Bm for controlling the degree of roughness in the simulation domain with other developed algorithms. These procedures for simulating irregular objects and manipulating their geometric boundary get the objects into a higher degree of reality as an abstraction of real-world entities.

It is noteworthy that the focus of this agent is on the importance of the emergent pattern from the behaviours of each individual and its interactions in the spatial simulation domain. Here only the abstraction of an entity into the spatial simulation domain and how it can be evolved is represented, i.e. the mechanisms for constructing the geometry and how the boundary changes over time.

Further research is on going for providing the relation of agents with other attraction or repulsion constraints. By introducing isotropic features and increasing the model complexity, agents can claim the desired fractal dimension, size and evolution of shape. The primary simulation test which has been carried out for measuring the shape roughness and size suggests that the model can be successfully calibrated in the area of land use and urban context.

6. Acknowledgements

This work would not be possible without funding from Otago University Postgraduate Research Scholarship.

7. References

Barros, J. (2003), "Simulating Urban Dynamics in Latin American Cities", Proceedings of the 7th International Conference on Geocomputation, University of Southampton, UK.

Batty, M. (2000), "Geocomputation Using Cellular Automata", In Openshaw, I. S. and Abrahart, R. J. (Eds.) Geocomputation, Taylor & Francis, London, UK, pp. 95-126.

Batty, M. (2001), "Cellular Dynamics: Modelling Urban Growth as a Spatial Epidemic", In Fischer, I. M. M. and Leung, Y. (Eds.) Geocomputational Modelling: Techniques and Applications, Advances in Spatial Science, Springer, Berlin, pp. 109-141.

Batty, M., Desyllas, J. and Duxbury, E. (2003), "The Discrete Dynamics of Small-Scale Spatial Events: Agent-Based Models of Mobility in Carnivals and Street Parades", International Journal of Geographic Information Science, Vol. 17(7), pp. 673-697.

Batty, M. and Longley, P. (1994), "Fractal Cities", Academic press, London.

Benenson, I. and Torrens, P. (2004a), "Geosimulation Object-Based Modelling of Urban Phenomena, Editorial", Computers, Environment and Urban Systems, Vol. 28, pp. 1-8.

Benenson, I. and Torrens, P. M. (2004b), "Geosimulation: Automata-Based Modelling of Urban Phenomena", Wiley, England.

Egenhofer, M. J. and Franzosa, R. D. (1991), "Point-Set Toplogical Spatial Relations", International Journal of Geographic Information Science, Vol. 5(2), pp. 161-174.

Egenhofer, M. J. and Franzosa, R. D. (1994), "On the Equivalence of Topological Relations", International Journal of Geographic Information Science, Vol. 8(6), pp. 133-152.

Goodchild, M. F. and Mark, D. M. (1987), "The Fractal Nature of Geographic Phenomena", Annals of the Association of American Geographers, Vol. 77, pp. 265-278.

Haklay, M., O'Sullivan, D. and Thurstain-Goodwin, M. (2001), "So Go Downtown: Simulating Pedestrian Movement in Town Centres", Environment and Planning B: Planning and Design, Vol. 28, pp. 343-359.

Hammam, Y., Moore, A., Whigham, P. and Freeman, C. (2004), "Irregular Vector-Agent Based Simulation for Land-Use Modelling", Proceedings of the 16th Annual Colloquium of the Spatial Information Research Centre, University of Otago, Dunedin, New Zealand.

Hayes-Roth, B. (1995), "An Architecture for Adaptive Intelligent Systems", Artificial Intelligence, Vol. 72, pp. 329-365.

Kenkel, N. C. and Walker, D. J. (1996), "Fractals in the Biological Sciences", COENOSES, Vol. 11, pp. 77-100.

Laurini, R. and Thompson, D. (1992), "Fundamentals of Spatial Information Systems", Academic Press, London.

Liu, Y. (2001), "Modelling Urban Development with Geographical Information Systems and Cellular Automata: A Case Study of Sydney since 1971", Unpblished PhD Thesis, University of Queensland, Australia.

Luck, M., D'inverno, M. and Munroe, S. (2003), "Autonomy: Variable and Generative", In Hexmoor, H., Castelfranchi, C. and Falcone, R. (Eds.) Agent Autonomy, Kluwer Academic Publishers, USA.

O'Sullivan, D. (2000), "Graph-Based Cellular Automata Models of Urban Spatial Processes", Unpublished PhD Thesis, Barlett School of Architecture and Planning, University College London, University of London, London.

O'Sullivan, D. (2001), "Graph-Cellular Automata: A Generalised Discrete Urban and Regional Model", Environment and Planning B: Planning and Design, Vol. 28, pp. 687-707.

Peitgen, H., Jurgens, H. and Saupe, D. (1992), "Chaos and Fractals: New Frontiers of Science", Springer-Verlag, New York.

Rasmussen, S. and Barrett, C. L. (1995), "Elements of a Theory of Simulation", proceedings of ECAL'95, Lecture Notes in Computer Science, Springer-Verlag, Berlin.

Rodrigues, A., Grueau, C., Raper, J. and Neves, N. (1998), "Environmental Planning Using Spatial Agents", In Carver, S. (Ed.) Innovations in GIS 5, Selected Papers from the 5th National Conference on GIS Research UK (GISUK), Taylor & Francis Ltd., London.

Rodrigues, M. A. S. (1999), "The Development of Spatial Intelligent Agents with Geographic Information Systems", Unpublished PhD Thesis, City University, UK.

Rucc, J. C. (1994), "Fractal Surfaces", Plenum Press, New York.

Russell, S. and Norvig, P. (1995), "Artificial Intelligence: A Modern Approach", Prentice Hall International Inc., NJ, USA.

Russell, S. and Norvig, P. (2003), "Artificial Intelligence: A Modern Approach, Second Edition", Prentice Hall Series in Artificial Intelligence, New Jersey, USA.

Semboloni, F. (2000), "The Growth of an Urban Cluster into a Dynamic Self-Modifying Spatial Pattern", Environment and Planning B: Planning and Design, Vol. 27, 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, Vol. 14(5), pp. 455-474.

Torrens, P. and O'Sullivan, D. (2000a), "Cities, Cells, and Complexity: Developing a Research Agenda for Urban Geocomputation", Proceedings of the 5th International Conference on Geocomputation, University of Greenwich, London.

Torrens, P. M. and Benenson, I. (2005), "Geographic Automata Systems", International Journal of Geographic Information Sciences, Vol. 10(4), pp. 385-412.

Torrens, P. M. and O'Sullivan, D. (2000b), "Cellular Models of Urban Systems", In (CASA) The Centre for Advanced Spatial Analysis, University College London, UK., Paper No. 22.

Voss, R. F. (1988), "Fractals in Nature: From Characterization to Simulation", In Peitgen, H. O. and Saupe, D. (Eds.) The Science of Fractal Images, Springer-Verlag, New York.

White, R. and Engelen, G. (2000), "High Resolution Integrated Modelling of the Spatial Dynamics of Urban and Regional Systems", In Computers, Environment and Urban Systems, Vol. 24, pp. 383-400.

Wooldridge, M. (1997), "Agent-Based Software Engineering", Proceedings on Software Engineering, Vol. 144(1), pp. 26-37.

Worboys, M. and Duckham, M. (2004), "GIS : A Computing Perspective", CRC Press New York.