A simple agricultural land use model implemented using Geographical Vector Agents (GVA)

A. B. Moore[1], Y. Hammam[2], P. A. Whigham[2]

[1]School of Surveying / [2]Department of Information Science,
University of Otago, PO Box 56,
Dunedin, New Zealand
Telephone: +64 3 4797589
Fax: +64 3 4797586
Email: tony.moore@otago.ac.nz

1. Introduction

In their book on Geosimulation, Benenson and Torrens (2004) have pointed out, "if modelled phenomena are an abstraction of real-world phenomena, why should modelled objects differ from their counterparts in the real world?" (p.4). With this in mind, Geographical Vector Agents (GVA) have been introduced as a generic spatial modelling framework populated by objects as computational agents (Hammam et al, 2007). The GVA fits within the geometric element of the Geographic Automata System (GAS) framework, developed by Torrens and Benenson (2005) to provide a dynamic vector-based modelling structure as an alternative to the prevailing and less spatially realistic Cellular Automata (CA). Before this, there had been attempts to use irregular alternatives to a cell-based structure, including Voronoi polygons (Shi and Pang, 2000), and more complex neighbourhoods through Delaunay triangle links (Semboloni et al, 2000) and planar graphs (O'Sullivan, 2001). GVA was initially developed on a scheme centred on the fractal manipulations of polygons. Algorithms that controlled object size and boundary complexity were implemented and tested visually with selected real-world instances (Hammam et al, 2007).

This paper reports on work that implements vector agents in a theoretical spatial model context, a step towards the calibration of vector agents with real world processes. The generic GVA model is outlined in terms of its implementation within the agent modelling shell Repast Simphony and briefly, how the dynamic geometry is driven with additional exposition of newer GVA developments: neighbourhoods, states and their associated rules (in line with the GAS framework). The theoretical spatial model used is von Thünen's model of agricultural land use (Hall, 1966).

2. The Geographical Vector Agent

The GVA is a spatial agent that is physically and explicitly defined by a Euclidean geometry, able to change its own geometric characteristics while interacting with other agents in its neighbourhood using a set of rules (Hammam et al, 2007).
Its properties include:

- a dynamic and self-directed nature that manifests itself through location and subsequent boundary manipulation – in this initial implementation, three simple algorithms are used for non-deterministic geometric change: the fractal-like midpoint displacement, edge displacement and vertex displacement, applied probabilistically (fig. 1)
- an ability to represent multifarious discrete geographic phenomena realistically through an irregular (or regular) vector data structure and flexible geometry rules, neighbourhood, time, states and transition rules.

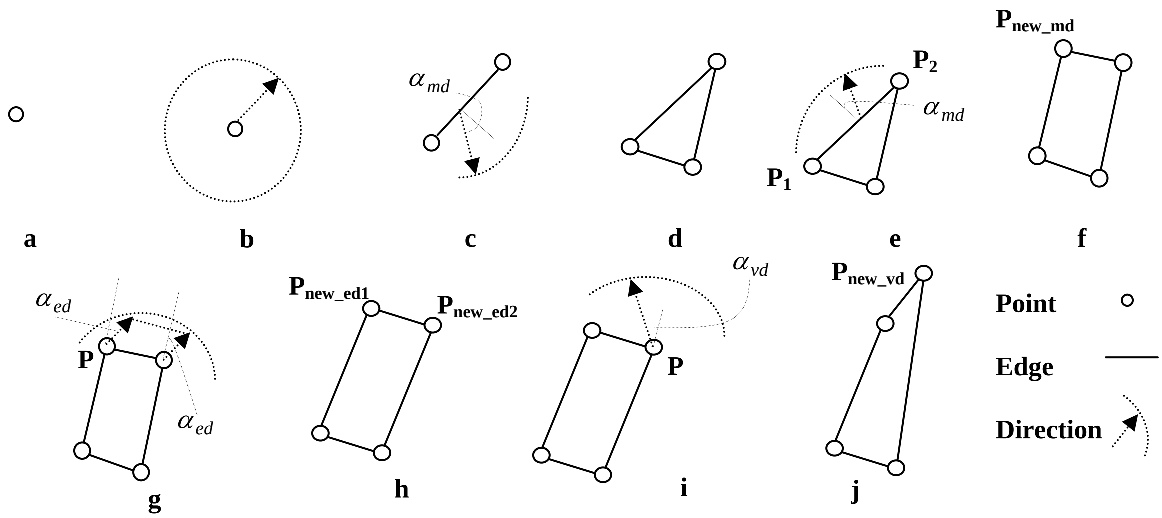

Figure 1. How a vector agent shape is born and evolves in the spatial simulation domain: (a) initialising by random point, (b) allocating second point by random displacement, (c, d) applying the random new point displacement and accomplishing closed polygon, (e, f) choosing any edge randomly and applying midpoint displacement, (g, h) edge displacement, (i, j) vertex displacement. All displacements are randomly defined by algorithms outlined in detail in Hammam et al (2007).

3. Generic Model Implementation in RePast

A Geographic Vector Agent application for the von Thünen scenario was developed in the Repast (Recursive Porous Agent Simulation Toolkit) framework, in Java (Howe et al, 2006). It was decided to use Repast Simphony for this implementation, as since the original GVA model development (developed in Java; Hammam et al 2007) this toolkit had added support for geographic vector data that included enablement of geometric change (i.e. modelling non-deterministic GVA objects) within the toolkit environment.

For the implementation described in this paper (see fig. 2), there is one context (a context in Repast is a container for agents that are semantically similar, or localized), VecContext, to which all agents belong. There are three major agent types for the simulation: the VecAgent (i.e. the embodiment of the GVA) is a dynamic geometry agent that behaves physically in the way described in fig. 1. The MakerAgent (which has a simple static geometry) creates VecAgents in the first place, putting them into the context. The IndividualAgent represents a human decision maker and does not have a geometry in itself; rather it is link to one of the aforementioned geometric agents. These abstract agent classes are generic and built on with specific scenarios. The application-defined agent classes holding von Thünen-specific members and methods are CityAgent, FarmAgent and EconomicAgent, which extend the MakerAgent, VecAgent and IndividualAgent respectively.

Three projections (which represent the spatial environment that links the agents together, geographic or abstracted) exist within the context, one geographic to hold the geometry of the GVA and two network representations to model projections for:

a) a vector network holding linear links to neighbouring vector objects (i.e. the VecAgents, MakerAgent) according to a Delaunay triangular network for a topological representation to model agent neighbourhood
b) a network that includes the links from IndividualAgents to the VecAgent or MakerAgent geometry.

In support of the agent classes, extended geometry classes implement the GVA specific geometric operations and the Delaunay neighbourhood (based on the polygon centroids) used to check for and prevent overlap of neighbourhood objects.

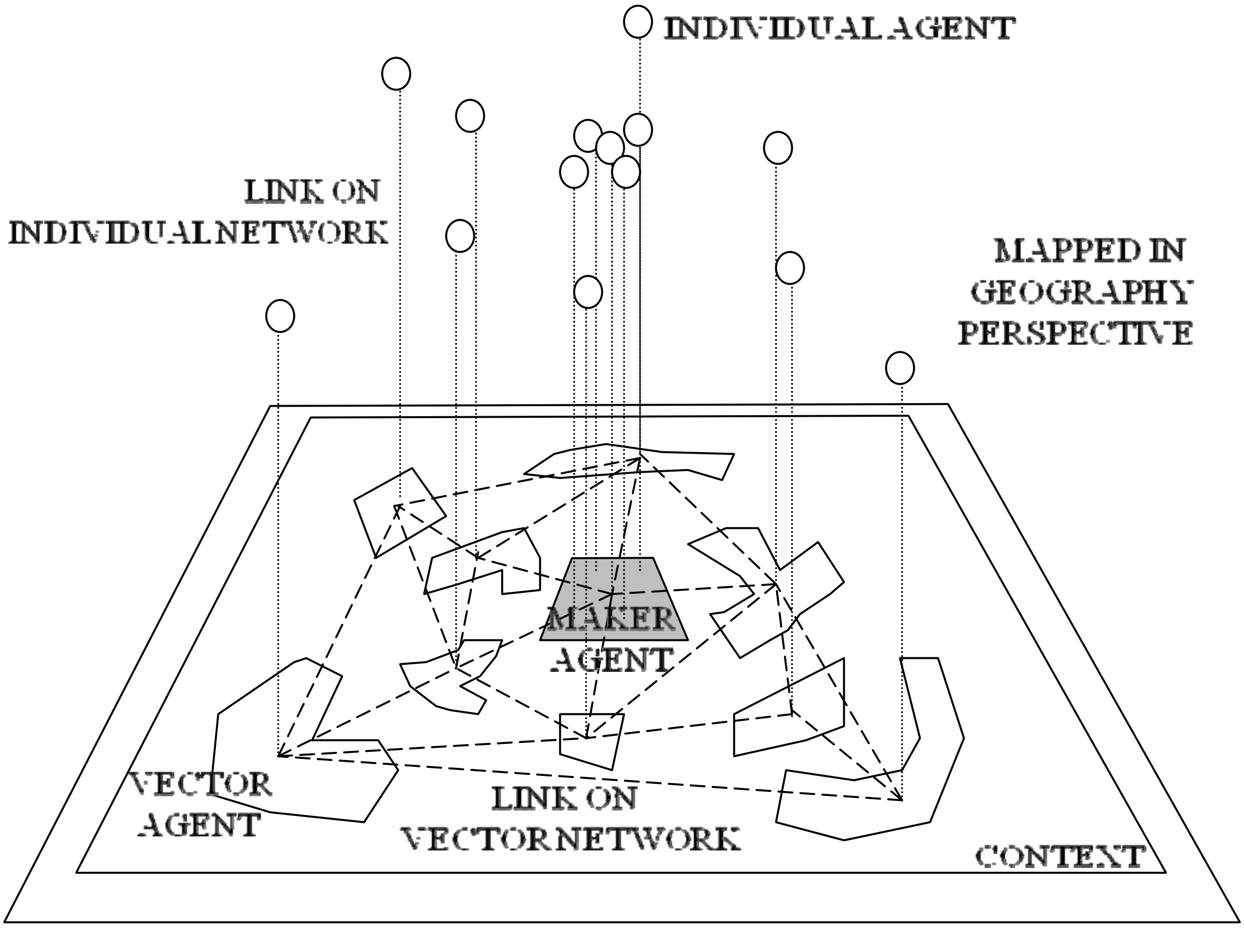

Figure 2. How the various agent types are related to their context and embedded projections (one geographic, two network-based) in Repast.

4. A Geographical Vector Agent Model for Von Thünen's Agricultural Land Use Theory

Building on the generic GVA model outlined in section 3, domain specific rules and states were added in order to implement von Thünen's theoretical model of agricultural land use. The model held that the relative costs of transporting different agricultural products to a central market determined the agricultural land use around that market (Hall, 1966). The most productive activities would compete for the closest land, pushing out the less productive activities. These land uses manifest themselves as concentric rings. Since the late 1950s there have been numerous attempts to quantify von Thünen's theory as a process (e.g. Stevens, 1968; Okabe and Kume, 1983). But in particular, it is the agent-based approach on a CA grid implemented by Sasaki and Box (2003) that will most inform this GVA model.

The aim of the simulation is to demonstrate the use and effect of irregular and dynamic GVA on a simple theoretical scenario. The hypothesis is that using start-up parameters adapted for the vector data model from Sasaki and Box (2003) the end result will be a concentric ring pattern around the market or city, the rings in order of distance from market, being intensive agriculture, forest resources, grain farming and livestock farming.

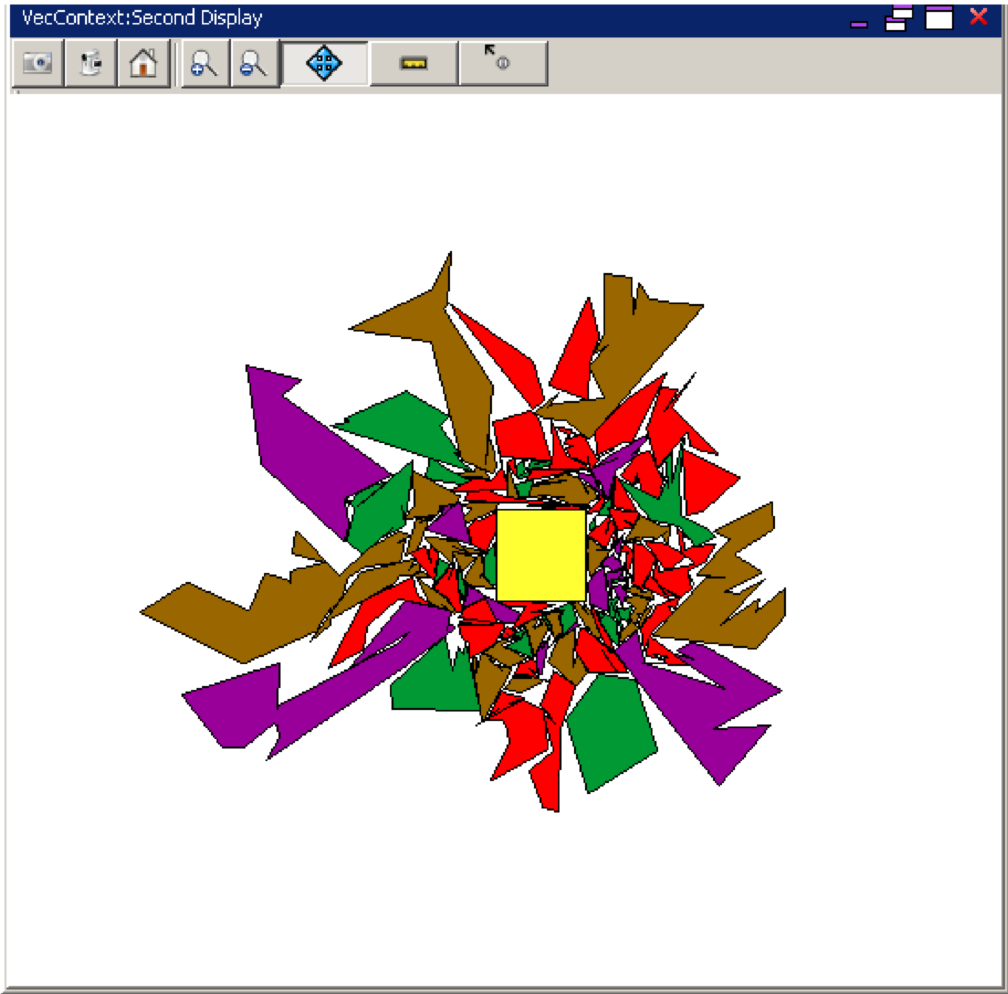

Figure 3. Output from the Repast GVA implementation. The City / Maker agent is the yellow square in the centre, with Farm / Vector agents surrounding it (red = intensive; green = forestry; purple = grain; brown = livestock).

This paper will show the latest progress in this Repast GVA implementation (a screenshot is shown in fig. 3). In summary, GVAs are vector geometry led agents, which have the advantage of giving real world object equivalents their own single identity, but it comes at the expense of computing complexity and cost. What is seen here is one realisation of GVA, with specific simple algorithms to govern object polygon shape and applied to a specific domain. Other algorithms or applications could be substituted with ease, building on the generic GVA structure. There is also a separation of geometry and attributes / states (i.e. linking geometry to economic success in the case of the von Thünen farmer) which needs to be addressed.

5. References

Benenson, I and Torrens, P. 2004. *Geosimulation: Automata-Based Modelling of Urban Phenomena*, Wiley, England.

Hall, P. 1966. *Isolated State: an English edition of der Isolierte Staat*. Pergamon Press.

Hammam, Y, Moore, A and Whigham, P. 2007. The dynamic geometry of Geographical Vector Agents. *Computers, Environment and Urban Systems*, 31, 5, 502-519.

Howe, T, Collier, N, North, M, Parker, M and Vos, J. 2006. Containing Agents: Contexts, Projections and Agents. In: *Proceedings of the Agent 2006 Conference on Social Agents: Results and Prospects.* Argonne National Laboratory.

Laurini, R and Thompson, D. 1992. *Fundamentals of Spatial Information Systems.* Academic Press.

Okabe, A and Kume, Y. 1983. A Dynamic von Thünen model with a demand function. *Journal of Urban Economics*, 14, 3, 355-369.

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.

Sasaki, Y and Box, P. 2003. Agent-Based Verification of von Thünen's Location Theory, *Journal of Artificial Societies and Social Simulation*, Vol. 6(2).

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(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*, Vol. 14(5), pp. 455-474.

Stevens, B. 1968. Location theory and Programming models: the von Thünen case. *Papers of the Regional Science Association*, 21, 19-34.

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