RBSim3: Agent-based simulations of human behaviour in GIS environments using hierarchical spatial reasoning

Robert M. Itami

GeoDimensions Pty Ltd
22 Dunstan Avenue
Brunswick 3056 Australia
Bob.Itami@geodimensions.com.au
Phone: +61 3 9386 0280

Abstract: This paper describes recent advancements in the development of an agent based simulation model called Recreation Behaviour Simulator (RBSim). RBSim allows recreation managers to investigate alternative management scenarios for linear recreation systems such as roads, trails, and rivers. The simulation program integrates Geographic Information Systems to represent natural environments with network algorithms and agent based simulations to model human behaviour. As a management tool, RBSim allows managers to alter characteristics of the environment through standard GIS techniques, as well as altering arrival schedules of visitors, global events such as weather, and the size and location of facilities such as parking areas for vehicles, campgrounds, and viewing platforms. Statistical output from the simulator may then be analyzed to reveal bottlenecks, crowding conditions, excessive queuing times for facilities, and the number of visual encounters. The architecture of this model has been described in detail in other papers (Itami et al 2002, Itami and Gimblett, 2001). This paper will focus on the spatial hierarchy used to simulate trips, and the use of hierarchical probabilistic rules to define complex spatial reasoning and context specific decision making in recreation agents. These concepts are demonstrated in a day use simulation for Port Campbell National Park in Victoria, Australia and compared to visitor use data collected for the study area.

Keywords: *Intelligent Agents, Recreation Behaviour, Recreation Simulation, RBSim, Recreation Management*

1. RBSIM: SIMULATING HUMAN BEHAVIOUR IN RECREATION ENVIRONMENTS

RBSim is a computer simulation tool first developed in 1997 in collaboration with Dr. Randy Gimblett of the University of Arizona (Gimblett, 1998, Itami and Gimblett 1997). It is designed as a general recreation management evaluation tool for linear recreation systems. This capability is achieved by importing park information from a geographic information system. Once the geographic data is imported into RBSim, other variables in the management model can be defined. Some of the factors the manager can change include the number and kind of vehicles, the number of visitors, and facilities such as the number of parking spaces, road and trail speed limits and the total capacity of facilities

RBSim allows park management to explore the consequences of change to any one or more variables so that the impact on facilities and the quality of visitor experience is measured. Statistical measures of facility use and visitor experience are generated by the simulation model to document the performance of any given management scenario. Management scenarios are saved in a database so they can be reviewed and revised. In addition, the results of a simulation are stored for further statistical analysis. The software provides tables and graphs from the simulation data so park managers can identify points of over-crowding, bottlenecks in circulation systems, and conflicts between different user groups.

RBSim uses concepts from recreation research and artificial intelligence (AI) and combines them in a GIS to produce an integrated system for exploring the complex interactions between humans (recreation groups) and the environment (geographic space). RBSim joins two computer technologies:

- Geographic Information Systems to represent the environment
- Autonomous human agents to simulate human behavior within geographic space.

An autonomous agent is a computer simulation that is based on concepts from Artificial Life research and Complex Adaptive Systems

(Railsback, 2001). Agent simulations are built using object oriented programming technology. The agents are autonomous because once they are programmed they can move about their environment, gathering information and using it to make decisions and alter their behavior according to specific environmental circumstances generated by the simulation. Each individual agent has it's own physical mobility, sensory, and cognitive capabilities. This results in actions that echo the behavior of real animals (in this case, human) in the environment.

What is compelling about this type of simulation is that it is impossible to predict the behavior of any single agent in the simulation and by observing the interactions between agents it is possible to draw conclusions that are impossible using any other analytical process.

The specifics of the object model and architecture of RBSim is described in Itami et al. (2002). This paper focuses on recent enhancements in relationship to the implementation of hierarchical probabilistic rules. The paper will therefore focus only on those aspects of RBSim that impact the way-finding behaviour of agents and specifically, the improvements in rule processing and increased flexibility in defining agent behaviour.

2. MODELLING THE VIRTUAL TOURIST

How does one model the behaviour of the traveling tourist? Agents have a number of standard attributes including fitness level, travel mode, travel speed, and preferences. Preferences are a list of values that correspond to site qualities that are attributes of network nodes. Values are weighted using Saaty's (1995) Analytical Hierarchy Process[1]. Preferences are used in the agent's logic for way-finding in locales which is described later in this paper.

Agents also have visual abilities. They can do line of site calculations to other agents at run time to count the number of other agents visible within a user defined radius.

2.1. The Travel Network

RBSim is currently constrained to recreation travel on linear networks. This includes roads, trails, rivers and streams, and flight paths. The network is comprised of links and nodes. Links are simple or compound lines defined by a set of vertices. The endpoints of a link are defined by nodes. Nodes are points that mark the endpoint of a link, the intersection of two or more links or marks a destination on a link. Links attributes include: maximum travel speed, access restrictions to specific travel modes and a link category (eg. divided highway, major arterial, paved road, unpaved road, high use trail, wilderness trail, etc.).

Node Attributes include a node label, a list of facilities and capacities, a list of site qualities and ratings for each site quality on a scale of 1 to 10 and optionally, membership in a locale. An example of a facility is a car park, with a capacity of 100 cars. An example of a site quality is historic value with a rating of 7 out of 10.

A locale is a collection of one or more *nodes* with associated facilities that have a shared identity and can be grouped based on proximity to each other and common access. Locales must have one or more entry nodes. An example of a locale might be a campground with links being comprised of the entry road and nodes being defined by campsites, toilet blocks and trailheads. The concept of a locale is important in the way-finding logic of agents, because it allows the agent to have rules that are sensitive to spatial context. That is, rules can be generated that will only evaluate for a given locale.

RBSim represents the terrain as a digital elevation model in the form of an evenly spaced grid of elevations which has been registered to the same coordinate system as the travel network. This allows uphill and downhill slope to be factored into travel speeds of an agent.

2.2. Typical Trips

The concept of a typical trip is central to the trip planning behaviour of a recreation agent. A typical trip is described by an entry node, an exit node to the network, an arrival curve, and probability distribution of agent types, a list of destinations (locales), and a trip duration. The concept of a typical trip is based on the premise that visitors have common patterns of use. For example, day use visitors arriving during weekdays will have a different arrival pattern, a different duration of stay, and perhaps a different pattern of destinations than a traveller arriving on a weekend or an overnight visitor. Typical trips can be derived from field data or based on the experience and expertise of managers on-site.

A typical trip always has a *global trip plan*, which specifies an intended list of destinations and durations at these destinations that the agent wishes to visit in a given sequence.

A *global trip plan* consists of:

- Total trip duration
- An entry node

[1] For a more detailed explanation of the use of AHP in agent reasoning see Itami and Gimblett, 2001

- An arrival time
- An exit node
- A sequence of intermediate nodes between the entry and exit nodes. Each destination node has a visit duration.

On execution, the agent moves from one destination to the next across the travel network using shortest travel time between the two points.

Hard Wired Trips vs Smart Trips

If data exists for the exact itinerary of all agents in a simulation, each trip can be explicitly defined in the Global Trip. These trips are referred to as "Hard Wired Trips". It is more common however that an individual trip will vary in ways that are difficult to measure and impossible to predict. It is for these kinds of trips that agent modeling is most useful. For these trips, RBSim provides agents with a way-finding logic and intelligence through a set of user definable rules that allow the agent to make decisions about what destination to visit next. These trips are referred to as "Smart Trips"

How Smart Trips are specified

If Smart Trips are to be defined, the network nodes must be clustered into Locales. Once locales and locale entries have been defined for the travel network, Smart Trips are defined in almost an identical fashion to Hard Wired trips, except that destination nodes are defined as a sequence of locale entry nodes and the duration is defined as the total duration of stay at the locale. If the agent is to stay overnight at a locale, the duration is defined by setting the overnight attribute for the locale and then an early and late departure time for the next day is set.

As the agent executes this plan, it follows the global trip, but once it arrives at a locale entry it checks its rules (described in more detail later) to see if the locale has facilities or attractions that it is interested in. If it does, the agent enters the locale and uses its internal way-finding logic to navigate through the locale. This logic is defined in more detail in the following section.

2.3. Way-finding logic of agents.

The way-finding reasoning of an agent is influenced by the following factors:

- Available time (defined by time elapsed subtracted from total trip duration)
- Travel mode as it affects travel time
- Agent preferences
- List of rules and their order
- Currently executing rules
- Internal state of the agent
- Current location of the agent
- Condition of the network including availability of facilities, access restrictions, and travel time to destinations on the network
- Previously visited destinations.

Once an agent reaches a locale, it must use its internal way-finding logic to find destinations, generate a path that links these destinations, and simultaneously take into account the factors in the above list.

When an agent arrives at a locale, it checks to see if there is a duration set for this locale in the global trip itinerary. If the duration is >0 then the agent checks to see if there is enough time left in its total trip duration by subtracting the time elapsed since the beginning of the trip and the time to travel to the exit node. If the remainder is positive and greater or equal to the duration set for this locale, the agent enters the locale and performs the following initialization procedures:

1. Loads the subset locale network for the agent's current travel mode
2. Generates weights for the site qualities for each node in the locale by multiplying the node site quality with the corresponding personality preference value (unique to the agent)
3. Marks any nodes that have already been visited as "visited" and sets their site qualities to zero
4. Sets its internal state to "entering locale"
5. Loads its rule list
6. Generates a locale trip plan
7. Executes its move behaviour for the locale.

The way-finding logic is encapsulated in step 6, generating the locale trip plan. Once the locale network has been initialised, the agent then evaluates all possible combinations of destinations from its current node location. These paths were pre-calculated when the simulation was initialised to enhance performance. The agent then evaluates each path and rejects any path that exceeds the available locale visit duration. The remaining paths are then ranked to maximize the site preferences and contain facilities that are on the agent's current rule list. A gravity model is used to weight the paths so paths with high priority facilities are ranked higher for facilities close to the agent's current location.

Once the preferred path is selected, the agent loads it as its current trip itinerary. The agent then traverses this itinerary as far as it can in the current time step. If the agent encounters a node that contains facilities that are on its current rule list, the agent changes its internal state to "visiting facility" and generates a visit duration for that facility. If the facility at the node has no available capacity (e.g. the parking lot is full), the agent "looks ahead" on its itinerary to see if a facility of the same class is available, if there is, the agent then continues its trip toward that node. If there is no other facility of the same class, the agent will then change its state to "queuing" and waits until the facility becomes available.

At each iteration of the simulation the agent must check its available trip time, its current travel mode, its current rule list, and its current state. Any of these can trigger a change in behaviour. The agent may abandon its current trip and calculate a path back to its arrival vehicle, or to the exit. If the conditions have not changed, then the agent continues to execute its current behaviour.

Though there are more details to this behaviour, the above reflects the overall logic behind the agent's way-finding decisions. When implemented, the logic produces behaviour that appears "smart" in that the agents generate logical paths and exhibit behaviour that is human-like.

2.4. Agent Rules

Agent rules are a set of user defined behaviours that are defined using a stimulus/response or event/action framework. RBSim exposes runtime properties of the network, agent, and global events. Each of these properties will have a state or value that can be defined as a stimulus or event. Boolean logic can be used to combine two or more stimuli to create complex conditions for behaviour.

Behaviour is defined as a directive to search for a location or facility. An example of a complex rule is:

> If (TravelMode = 'Car' AND
> Locale='12 Apostles' AND
> LocaleEntry = True) THEN Find
> Carpark

This rule is an example of a rule that will fire only within the context of a locale (12 Apostles). It defines the travel mode of the agent (car) and the location of the agent (locale entry).

The user can specify the order in which the agent considers rules for execution. For instance, an agent should always park a car before going to a visitor centre.

2.5. Calibrating agents to field data using probabilistic rules

When field data is collected on visitor movements using mechanisms like trail counters, infrared sensors, GPS, or field observation, it is often possible to surmise the reason why a tourist will select one path as opposed to another and then develop a rule for an agent that will replicate this choice behaviour. However many times it is impossible to determine the reason for a choice an agent has made, but there is a clear and consistent pattern to the movement. For example data collected may indicate that 25% of the visitors will turn left and 75% of visitors will turn right at a trail intersection, however there is no apparent reason for this choice. Yet, it is desirable to be able to accurately simulate this pattern of behaviour.

Using the simple rules described in the previous section, it is impossible to replicate probabilistic behaviour in agents. However by making some relatively simple changes to the structure of the rules it is possible to implement probabilistic hierarchical rules.

2.6. Probabilistic, Hierarchical Rules

Probabilistic rules are rules that execute based on a randomly generated number at run time. When a probability is assigned to a rule, the rule will only execute if the randomly generated number is less than or equal to the probability.

Hierarchy is implemented using two mechanisms, first is the order the rules are assigned to the agent. Rules will always execute in the order they are assigned to the agent. However the user may also make a rule imperative. That is, the rule MUST execute before any subsequent rules will execute. This concept enforces a simple two level hierarchy for rules.

Probability and imperativeness are assigned as properties to a rule. The default state of these properties is a probability of 1 (rule will always fire), and rule is not imperative.

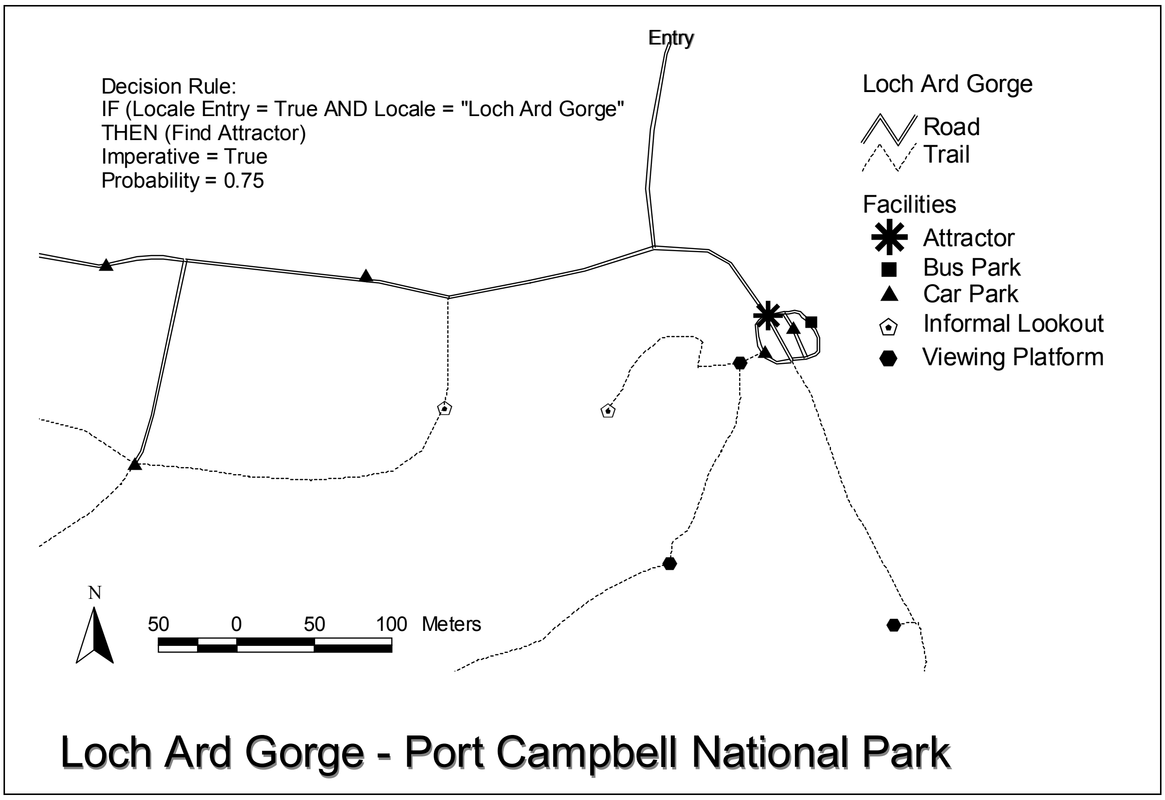

Figure 1. Part of Loch Ard Gorge locale showing entry point, road and trail network and facilities. Note attractor facility at entrance to east parking area.

3. A SIMPLE IMPLEMENTATION

Figure 1 shows a map of the entry sequence to Loch Ard Gorge, a locale in Port Campbell National Park in Victoria, Australia. The entry to the locale is off the Great Ocean Road to the north of the site. When a visitor turns into the entry, a 'T' intersection is reached. This is a blind intersection heavily vegetated, so there are no clear views of the facilities or scenery within the locale. Traffic count data taken at the site shows 37% of traffic turning west and 63% turning east.

When RBSim runs the simulation without hierarchical probabilitic rules the agents in the simulation are assigned the following rules:

1. If Arriving at a locale in a Car THEN find a Car Park
2. IF at any Locale THEN find Viewing Platform (repeatedly)
3. IF at any Locale THEN find Informal Lookout (repeatedly)

The resulting behaviour is that all agents will turn west since the nearest car park is to the west (agents by default select paths with the shortest travel time to the facility of the currently executing rule) until all car parking is full, then new arrivals will turn east. However as soon as parking becomes available in the closest parking areas to the west, new arrivals will always prefer the westward turn at the intersection.

To modify this behaviour to correspond to the data captured in traffic counts, a new imperative rule is created with a new facility called an "Attractor" that is placed close to the entry of the East parking area. The new rules for the agents are as follows:

1. IF Entering Loch Ard Gorge THEN go to Attractor
2. If Arriving at a locale in a Car THEN find a Car Park
3. IF at any Locale THEN find Viewing Platform (repeatedly)
4. IF at any Locale THEN find Informal Lookout (repeatedly).

The first rule is assigned a probability of 0.63 and is made imperative. When the agent arrives at Loch Ard Gorge, the simulator generates a random number. If the number is less than or equal to 0.63 the agent immediately executes rule 1 and turns east to the attractor facility. If the number is greater than 0.63 the agent does not execute rule 1 and then executes rule 2 which, as in the first simulation will cause the agent to select the path to the nearest parking area to the west.

If rule 1 fires, the agent will then execute rule 2. Since the attractor facility is close to the car park, the agent will always proceed to the nearest parking space. If all spaces are full, the agent will then drive to the west parking areas. If all spaces are full, the agent will queue until a space becomes available.

When this scheme is implemented, the pattern of behaviour at the entry intersection derived from traffic count data is replicated.

If no attractor facility is found in the locale, rule 1 will fail to fire, since conditions of the rule are not met. Since the rule is imperative and first in the list of the rules, all other rules will not fire. This will cause the trip to fail, and the agent will not enter the locale.

4. CONCLUSIONS AND RECOMMENDATIONS

The inclusion of hierarchical probabilistic rules to RBSim adds considerable flexibility in defining the way-finding reasoning of recreation agents. This expands the flexibility of RBSim in that completely probabilistic trips derived from automatic traffic count data can be generated provided direction of travel can be determined. With improvement in monitoring visitor movement (see Arnberger et al., 2002) and O'Connor et al., 2003) the accuracy of simulations such as RBSim will improve, this will increase the utility and reliability of the results of these simulations. As more knowledge and information about human movement in recreation environments is gained, agent simulations will need to have a richer set of mechanisms for expressing the human decision making processes and behaviour. Probabilistic hierarchical rules along with spatial hierarchies provided by locales in RBSim allow for a richness of behaviour that anticipates future demands for more accurate and detailed specification of agent intelligence.

On the basis of the promise of the initial implementation of a simple two level hierarchy for agent rules, the next logical step is to expand the rule engine so it handles a complete tree hierarchy of rules. In concert with these efforts, more simulation states will be exposed to the rule engine, as well as the specification of more agent behaviours, hence providing a wider range of rule driven behaviours.

5. ACKNOWLEDGMENTS

Dino Zanon, Parks Victoria managed the Port Campbell and Bay of Islands National Park project and provided the visitor data for the simulation.

6. REFERENCES

Arnberger, A., C. Brandenburg and A. Muhar (2002) Monitoring and Management of Visitor Flows in Recreational and Protected Areas. Conference Proceedings, Vienna, Austria: 143-149

Gimblett, H.R.; Itami, R. M. (1997). Modeling the Spatial Dynamics and Social Interaction of Human Recreationists' Using GIS and Intelligent Agents. *MODSIM 97 - International Congress on Modeling and Simulation*. Hobart, Tasmania. December 8-11.

Gimblett, H.R. (1998). Simulating Recreation Behavior in Complex Wilderness Landscapes Using Spatially-Explicit Autonomous Agents. Unpublished Ph.D. dissertation. University of Melbourne. Parkville, Victoria, 3052 Australia.

Itami, R., R., Raulings, G. MacLaren, K. Hirst, R. Gimblett, D. Zanon, and P. Chladek (2002). RBSim 2: Simulating the complex interactions between human movement and the outdoor recreation environment, *Monitoring and Management of Visitor Flows in Recreational and Protected Areas, Institute for Landscape Architecture and Landscape Management Proceedings, University of Agricultural Sciences, Vienna, Austria.*

Itami, R.M. and H.R. Gimblett (2001) Intelligent recreation agents in a virtual GIS world. *Complexity International*, Vol. 8. http://ww.csu.edu.au/ci/vol08/itami01/itami01.pdf

O'Connor, Zerger, A. and Itami, R.M. (2003) Building better agents: Geotemporal tracking and analysis of tourist behaviour MODSIM 2003, Townsville, Queensland.

Railsback, Steven F. (2001) Concepts from complex adaptive systems as a framework for individual-based modelling *Ecological Modelling* 139 47–62

Saaty, Thomas L. (1995) Decision making for leaders: the analytical hierarchy process for decisions in a complex world. Published Pittsburgh, Pa. RWS Publications.