Distributed Individual-Based Environmental Simulation

Mark Scahill,
Computing Laboratory, University of Kent at Canterbury.
The University, Canterbury, Kent, CT2 7NF, UK.
Telephone: +44 1227 823979. Fax: +44 1227 762811
E-mail: M.Scahill@ukc.ac.uk

Abstract

This paper describes the implementation of a distributed modelling environment which allows the simulation of a large number of individuals. Particular attention will be paid to modelling an individual's behaviour, communication and interaction with the population and a shared environment. Individual based modelling is not a new concept, nor is the idea of distributed simulations, the system detailed here offers a means of combining these two paradigms into one large-scale modelling environment.

A key concept in this system is that each individual being modelled is implemented as a separate entity. This atomisation of the model allows the simulation a greater flexibility, individuals can be rapidly developed and the simulation can be spread over a number of machines of varying architectures.

In an attempt to produce a flexible, extensible, individual-based model of a large number of individual subjects the client-server paradigm has been employed. Combining the individual-based modelling techniques with a client-server network architecture has been found to be straightforward with the added bonus of having communication between individuals included for free. The idea of considering the problem as one of interaction between an individual and the environment means that the problems normally associated with distributed simulations, those of continuity of world-views for different clients and of communication between clients, are easily solved.

Although this system has been developed originally to allow simulations of the mountain gorilla population, the modelling methods employed have meant that almost any entity can be simulated with very little change to the basic simulation processes described here. As such details of how the gorilla behaviour or learning will be implemented are not covered except for how they are facilitated by the system See Scahill (1996, 1997) for details of the system not covered in this paper.

Keywords

Distributed, Individual-based, Population Modelling.

Environmental Software Systems Vol. 2 R. Denzer, D.A. Swayne & G. Schimak (Eds.)
Published by Chapman & Hall

1. INTRODUCTION

This paper describes the construction of a distributed individual-based model allowing the simulation of a large number of individuals. The system will be used to model the entire Mountain Gorilla (*Gorilla Gorilla Beringe*) population but has been designed to be as general as possible. Areas which will be studied will include: gorilla behaviour, population dynamics and environmental dependencies.

Mountain gorillas were chosen for this case study for a number of reasons: They form stable social groups, leading to relatively simple rules of interaction. (Schaller, 1963); The environment they live in has been documented in great detail over the past thirty years. (Schaller, 1963; Fossey, 1983; Harcourt, 1981); The population currently numbers less than seven hundred individuals (Harcourt, 1995); Their habitat is a closed geographical area; and they are officially an endangered species, so any information about the future survival of the species could help present preservation planning. (Harcourt, 1995)

Each gorilla will be modelled as a separate individual with information being passed between a gorilla and an environment which is shared between all the gorillas being modelled. This sort of communication between individuals and the environment has a number of advantages: The model can be implemented over a network of computers which can prevent any one computer from becoming too heavily loaded; Each gorilla can have their own view of the world, independent from other gorillas, yet appear in other gorillas' views; Gorillas may learn and act on an individual basis whilst being aware of the actions of other gorillas in their neighbourhood; Interactions between gorillas can be easily implemented.

The environment is modelled as a rectangular grid containing data on the height of each grid-point, the vegetation type and any activity in the area. The environmental process accepts actions from all gorillas and then sends back to individuals the results of their actions and the new, updated, view that the individual may have of its own locality.

Although the simulator has been developed to be independent of the species being studied it is convenient to refer to the mountain gorilla model throughout this document.

2. INTERACTION

In any investigation of the behaviour of a number of individuals, it is important to allow those individuals an ability to interact with each other. The use of a common environment in the system provides a reliable means of ensuring all interactions between individuals can be experienced by other individuals in the locality as well as those directly involved in the interaction.

Communication between gorillas is based on visual, audio and olfactory senses. The simulation environment must allow for these various interaction media and be able to broadcast actions over a varying range depending upon the nature of the action. Figure 1 shows the range of these senses. A given individual may see the events occurring in the cell they currently reside in and each of their immediately neighbouring cells. They may hear any sounds from these cells and cells one step further out. Finally (depending upon wind speed and direction) they can detect odours up to three cells out from their present location. The nearer a sound or smell originates, the stronger the experience.

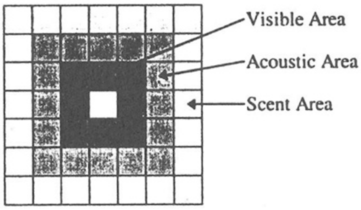

Figure 1: Gorilla sensory zones

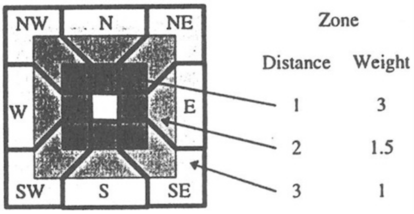

Figure 2: Gorilla movement zones

The remaining form of interaction, that of physical contact, can only be experienced by gorillas located in the same cell at the same time. These interactions are directly experienced by one other gorilla, for example the gorilla being groomed, but may be observed by the neighbouring gorillas.

Schaller (1963) described the bonds which kept gorilla groups together yet limited their size to between five and thirty individuals. He also described the rules forming the basis of social organisation within the population.

These factors include: the relationships between parents and offspring, which is more pronounced in females; the protection given to young gorillas by more senior members of the group; the loyalty or respect gorillas have for their senior members; the rivalry between gorillas of similar stature and the intimidation subordinate gorillas may feel about larger more aggressive individuals.

In general, the forces acting between gorillas will be a combination of these factors. The attractive forces acting to keep groups together and the repulsive forces helping to stop all the gorillas from trying to be in the same group.

These inter-gorilla relationships are planned to be modelled as a set of ***springs*** combining the attractive and repulsive forces listed above. A gorilla will be governed by its desires to be near some gorillas and away from others, in a certain vegetation type or doing a certain activity. This conflict of interests will be the basis for group formation and stability.

3. THE SIMULATION

In this system, each individual is implemented as a separate process and the environment as another process. This atomisation of the model enables rapid development of the system whilst giving individuals the ability to easily modify their own world views and behaviour. The alternative method of modelling large numbers of individuals on this scale would be to have one large monolithic system with all control being handled internally. The complications of ensuring separate world views for each gorilla, and allowing individual's the ability to develop their own behaviour, in this kind of system would create a large unwieldy program. By dividing the model into many smaller processes, each individual process requires less resources from the host computer and, as mentioned, allows gorillas to be distributed over a network of computers reducing the demand (in terms of processor time and system memory) on any one machine. DeAngelis (1992) detailed many different approaches to individual-based modelling, including parallel methods.

The simulation is performed through a series of discrete time steps. During any given time step each individual will decide upon its action and obtain the results of those actions. Possible actions include movement, feeding, interacting with other individual and sleeping.

Each individual remembers areas of the environment which it has experienced. When deciding upon a given movement step an individual will consider their immediate surroundings and where it remembers a given resource (such as a food-type) to be obtainable. An individuals movement is based on the 49 grid cells surrounding their current location (3 cells in each direction as indicated in Figure 1). Decisions about which of the immediately adjoining cells to move to will be made based upon what they can sense in those cells, their memory of the location/area if they have been there before and any environmental feedback from the surroundings.

Environmental feedback may well come from up to three squares away (in the case of smells) but the individual experiencing the smell can have no idea how far away the smell originated, only the direction can be determined. A similar argument applies to sounds so the sensory feedback is going to appear to have come from the nine cells adjacent to the individual's present location. These nine cells, the currently occupied cell and its eight adjoining cells are all that need to be passed back from the environment to a gorilla.

Based on its senses, a gorilla will decide to move in a given direction. There are nine movement directions: N, NE, E, etc. and stationary. The stationary zone is the individual's current location, the other zones comprise an area of six grid squares each. Figure 2 shows how these zones are arranged. Each zone contains the same number of cells, half cells being shared equally between the two neighbouring zones, and the same number of cells (and shared cells) at each distance. In considering which direction to move, an individual considers the contents of each cell in the zone, weighted depending upon how far away the cell is.

Also to be considered are land heights and vegetation types. Heights are important since different vegetation types tend to be found at certain altitudes. As gorillas learn more about their environment they will learn to associate sources of food with different altitudes. Each reference point in the landscape contains information on the altitude and vegetation type at that point. This information is available for the whole of the Virungan Volcano region where the mountain gorillas were first found. The data for the landscape is mostly obtained from the Grant F. Walton Center for Remote Sensing and Spatial Analysis, where the elevation map was generated from Belgium maps and the vegetation from a mixture of Satellite photographs and ground based observations.

For more information see http://deathstar.rutgers.edu/projects/gorilla/gorilla.html. Although the actual landscape being modelled is not as regular as this grid representation implies, the abstraction makes the model more simple to construct without being too artificial.

4. COMMUNICATION

Communication in this simulation is performed through the use of Unix TCP sockets. This means that two processes can set up a private link between each other and pass data along this link in both directions. These links can be between processes on the same physical machine, or between processes on different machines. As far as the system is concerned, all the complicated message passing routines are hidden by calls to the operating system which send information to, and receive information from, a message channel (Tanenbaum, 1989; Bacon, 1993).

Early on in the development of this modelling system, it became apparent that modelling over three hundred individuals on the same computer was not going to be feasible. A limit on the number of channels that could be open at one time meant that it was not possible to model more than about forty gorillas simultaneously on any one computer.

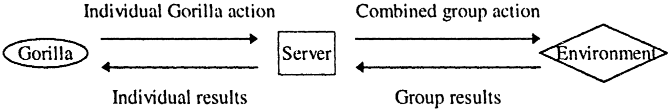

Figure 3: Gorilla to environment communication.

Since the limiting factor was the number of channels, spreading the gorillas over a number of machines would not have solved this problem as each individual would still have to communicate with one single environment. This problem was solved by giving each machine a server, which acted as an intermediate service between the gorilla and the environment, as shown in Figure 3. Using this method, the environment would only need to communicate over the network with the servers and each server would communicate with the local gorillas and one link to the environment. Therefore a simulation of the entire mountain gorilla population of just over 600 is possible using the servers as an intermediary between the environment and the individual. Each machine running gorilla processes would require only one server and thus only one link to the central environment.

By spreading the simulation over a number of machines there is also the bonus of allowing individuals to process the results of their actions in parallel. This allows a substantial increase in the number of actions which can be performed in a given time over the single modelling process normally used for individual-based simulation. Actions are passed to the group server which in turn passes them on to the environment. The environment processes these actions and sends the results back to the gorilla, via the group server. Figure 4 shows the environment's view of a system of thirty individuals on six separate machines (seven including the environment). The environment cycles though the servers receiving all the actions for each

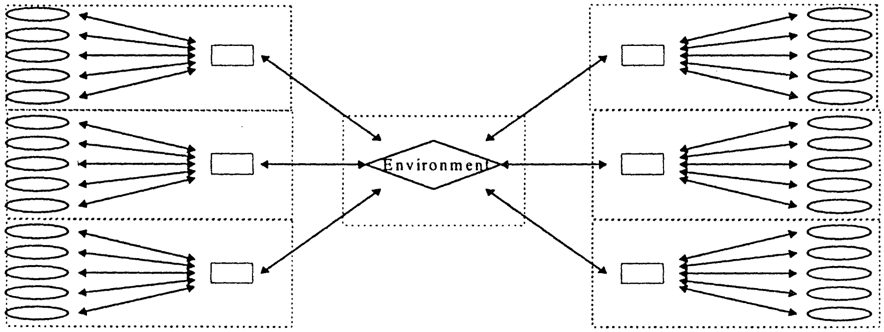

Figure 4: Environment Communication

collection of individuals, then processes all the actions and sends results back to each server in turn. The environment, once it has received all the servers' actions, will process gorilla actions on an individual basis irrespective of the machine they are running on.

The communications system allows each gorilla to share a common environment, communicate with other gorillas in their vicinity and allow individual gorillas to form their own view of this environment. Being able to spread the gorillas out among other machines prevents any one machine from becoming too loaded. The servers, acting as multiplexors, reduce the number of network connections and the network lag which would have been present had each gorilla communicated directly with the environment. These factors combined result in the system performing much faster with a larger number of gorillas than it did before the servers were introduced and the gorillas distributed over a number of networked machines.

5. COMMUNICATION PROTOCOLS

Since the most important part of this system is the communications between processes it is important that enough information is shared between individuals and the environment to allow individuals to make valid decisions on their actions and to interact as naturally as possible with each other.

0	Ape ID		Server ID	Size	
4	X-coord		Y-coord		
8	Action	Intent	To Ape		12

Figure 5: Gorilla to server packet

A typical action is sent as a series of twelve bytes, as shown in Figure 5. These packets contain details of the individuals unique ID number, the group-server the gorilla belongs to (for authentication purposes) the size of the individual and their location. It then contains their action (moving, eating, grooming, etc.), the intention behind the action (anger, peaceful etc.) and the ID of the individual the action is aimed at. The server combines all these actions together into a larger packet, adds the server id (for authentication) and the number of gorillas which the server is responsible for serving. For a group serving thirty individuals this packet will be 364 bytes long.

Upon receiving packets from all the servers the environment processes the actions for each

0	Ape ID
2	Vegetation
11	Heights
20	Sounds + smells
29	Number of local apes
31	Neighbour 1 Action
39	Neighbour 2 Action
47	Neighbour 3 Action
55	

Figure 6: Server to Gorilla Packet

gorilla and calculates their results. These results are then compiled into a packet containing the results for all the individuals managed by a given server and sent to that server. The environment to server packet will be these results packets together with the server's ID number and the number of gorillas being served, which may be more than the group previously served through births or deaths. The packet sent by the server back to an individual, shown in Figure 6, contains the individuals ID number followed by the vegetation heights, sounds and smells of the adjacent nine grid locations. The number of visible gorillas then follows along with the observed actions of each neighbour: the neighbours ID, their size, their location relative to the destination gorilla, their action, their intent and which gorilla the action appears to be aimed at. For a gorilla with no neighbours, the results packet will be 31 bytes long whilst a gorilla with ten neighbours will receive a packet 111 bytes long.

For a server with eleven distributed individuals, the result packet received from the environment will be 345 bytes, whilst a group with eleven gorillas within close proximity to each other will contain 1225 bytes (each gorilla being informed of its ten neighbours actions). Table 1 shows the volume of network traffic created for one server when serving thirty individuals in close proximity to each other.

Conversation	Packet size		total bytes sent	Sent Over
Gorilla to Server	12	per gorilla	360	Local
Server to Environment	364	per server	364	Network
Environment to Server	7894	per server	7894	Network
Server to Gorilla	263	per gorilla	7890	Local

Table 1: Packet sizes (in bytes) for thirty gorillas in close proximity

With a simulation of 300 gorillas spread over ten machines, this means that about 80kb of information may be sent over the network for each time step (not including any protocol headers added by the system).

Number of servers	number of clients					
	1	3	20	42	60	100
1	4137	9350	22 857	44 017	-	-
3	-	8470	43 636	68 200	65 753	
4	-	-	50 526	69 041	87 000	61 224

Table 2: Average number of gorilla actions processed each minute

Table 2 shows the number of gorilla actions served each minute for varying numbers of gorillas and server. In these tests, the gorillas were distributed around the environment and made no active movements. These tests were carried out on a network of Sun Sparc 10's connected through 10Mb/s Ethernet. Although these machines were being used by other people at the time of these tests, the load remained fairly constant during the testing period. Each test was carried out for a period of 100 000 gorilla actions and the number of actions processed by the environment in a minute was averaged over the test. The figures show that using servers, to aid distribution of the processes, produces a very large increase in the number of actions processed by the environment in a given time period. When testing 20 gorillas on the same machine as the environment the load increases dramatically and the number of actions drops to around 9300 actions per minute.

6. CONCLUSION

This paper has described the construction of a distributed simulation environment. The system has been designed with the intention of modelling an entire species numbering some six hundred individuals. Although much of the implementation has been based on the mountain gorilla population it has always been the intention that the underlying simulation methodology should be adaptable to model any species meeting the criteria of living on a closed environment, having a stable social structure and well documented patterns of interaction.

The system makes extensive use of the Unix network facilities and has been successfully implemented on a number of systems including SunOS, Ultrix and Linux. It should be easily portable to any system offering TCP networking facilities. As an experiment, a single gorilla has also been implemented in Java - and though slower than compiled code, shows the true flexibility of the system, being network and architecture independent.

The ability to model a large number of individuals as separate processes allows a very flexible approach to be taken in the development of this system. Other species can easily be added to the gorilla model or the system can be readily adapted to modelling completely different scenarios.

7. REFERENCES

DeAngelis, D.L. and Gross, L.J. (1992) *Individual Based Models and Approaches in Ecology*, Chapman and Hall.

Fossey, Dian (1983) *Gorillas in the Mist*, Houghton Mifflin Company.

Harcourt, A.H and Fossey, D. and Sabater-Pi, J. (1981) Demography of Gorilla gorilla, *Journal of Zoology*, London, **195** pages 215-233.

Harcourt, A.H. (1995) Is the gorilla a threatened species? How should we judge?, *Biological Conservation*, **75** pages 165-176.

Schaller, G.B. (1963) *The Mountain Gorilla*, Chicago U.P.

Scahill, M. (1996) *Modelling Mountain Gorillas*. UKC Technical Report, Computing Laboratory, Canterbury, Kent.

Scahill, M. (1997) *Distributed Environmental Simulation*. UKC Technical Report, Computing Laboratory, Canterbury, Kent.

Tanenbaum, A.S., (1989) *Computer Networks*, Prentice Hall Inc.

8. BIOGRAPHY

Mark Scahill has been a Computing Fellow at the University of Kent at Canterbury for nearly six years. When he hasn't been teaching or looking after first year exams he has been investigating the ability to model the behaviour of the Mountain Gorillas on a species wide basis. With a BSc. in Astrophysics and an MSc. in Computer Science it was either that or write space games. He has a Victorian house that still needs decorating after two years of hard work and a garden where plants die and attracts more cats than wildlife.