Chapter 1
GENESIS Social Simulation

Andy Turner

Abstract GENESIS is a research project funded by the UK Economic and Social Research Council through the National Centre for e-Social Science research node program. Details of GENESIS can be found on-line via the following URL:

http://www.geog.leeds.ac.uk/people/a.turner/projects/GENESIS/

One strand of GENESIS work aims to develop simulation models that represent individual humans and their organisations and how they change their location and influence over time. This chapter describes the development of two types of model that operate at different temporal resolutions over different time scales: Traffic Models work with time steps of a second; and Demographic Models work with time steps of a day. Both are computationally demanding and the chapter describes not just the development of the models, but also the work done to scale up from village scale with a thousand individuals, to produce simulation results at city scale with a million individuals.

1.1 Introduction

GENESIS is a research project funded by the UK Economic and Social Research Council through the National Centre for e-Social Science research node program[1]. Details of GENESIS can be found on-line via the following URL:

http://www.geog.leeds.ac.uk/people/a.turner/projects/GENESIS/

One strand of GENESIS work aims to develop simulation models that represent individual humans and their organisations and how they change their location and influence over time. For this strand, the author is developing two types of model that operate at different temporal resolutions over different time scales. These models

Andy Turner
University of Leeds, Leeds, LS2 9JT, UK e-mail: A.G.D.Turner@leeds.ac.uk

are implemented in the Java programming language and share a common code base (packages, classes and methods) which is released as open source on-line via the following URL:

http://www.geog.leeds.ac.uk/people/a.turner/src/andyt/java/projects/GENESIS/

The types of models are: traffic models that work with time steps of a second and are aimed to be run for simulated time scales of days; and demographic models that work with time steps of a day and are aimed to be run for simulated time scales of years. Both are computationally demanding and much effort has gone into organising data and processing so as to scale up from village scale models with a thousand inhabitants residing in a region with a radius of a few kilometres, to city scale models with a million inhabitants residing in a much larger region. A long term aim is to scale up further and run larger regional, national and even global scale simulations. Some background to this modelling work is provided in Section 1.2. More detailed descriptions of the traffic and demographic models and their development is provided in Section 1.3 and Section 1.4 respectively. Section 1.5 focusses on computationally scaling up. Section 1.6 considers further work. Section 1.7 provides a short summary and concluding remarks.

1.2 Background

The dynamic simulation models described in Section 1.3 and Section 1.4 were developed from first principles, conceptually aiming to be as basic as possible in the first instance then develop incrementally by: factoring in constraints; incorporating less basic concepts; computational enhancements (i.e. improving scalability); enabling simulation control (stop and restart); and interfacing with some form of archive.

The first phase of development involved a sort of prototyping that evolved basic visual outputs from the two types of models. The second phase has focussed on scaling up so that simulations of the models would run for larger regions at higher spatial resolution and for larger populations of individually represented people. For the first phase, a basic demographic model was reasonably straightforward to develop, whereas a basic traffic model was more of a challenge and was developed in several steps.

As part of the EUAsiaGrid project [2] (funded via the European Commission), Alex Voss [3] implement a basic demographic model in Repast [4] based on a basic GENESIS demographic model. This was presented at specially arranged meetings at the Academia Sinica Centre for Survey Research during the International Symposium for Grid Computing 2009 [5] aiming to encourage EU-AsiaGrid collaboration and raise awareness of the GENESIS social simulation efforts. This pioneering work was progressed by Alex who with David Fergusson [6] developed materials for and organised a Social Simulation Tutorial which we ran together at the International Conference on e-Social Science 2009 [7]. An updated Social Simulation Tutorial developed by Alex, Rob Procter [8] and myself was run at the International Symposium for Grid Computing 2010 [9].

To begin to address the computational challenge of developing large scale social simulation models, the GENESIS models were evolved to take greater advantage of the usually relatively large persistent disk based memory of contemporary computer hardware to store data about a simulation. Methods were developed to control the swapping of simulation data from this large disk memory to and from the smaller, more volatile yet faster memory. The author had experience of implementing such memory management in the development of a raster data handling library called Grids [10] about which he made a presentation at FOSS4G2006 [11]. The development of Grids has been a singular effort funded indirectly under various research grants from the European Commission and UK research councils. It is a core library used in the traffic models which needed adapting to work in this context. The general solution for memory handling in Grids could be readily applied memory handling for the agent data for the GENESIS models. (This agent data currently representing individual humans in the models.) The generic code used in Grids and the GENESIS software was abstracted to another library called Generic [12].

Along side the GENESIS project, the author is working on the NeISS project [13] (funded via the UK Joint Information Systems Committee) to develop a national e-Infrastructure to support social simulation. NeISS aims to support GENESIS models and the hope is that a community of users will be established for these models. The e-Infrastructure should make the models more easy to use by allowing for Grid and Cloud based computational resources to be tapped and the results of running the models easier to archive and reproduce. Key to the NeISS work is making it easy for others to modify and run the simulation models and produce outputs they are interested in.

1.3 Developing a Traffic Model

To model people movements on an individual level requires a way to store the location of each individual. A first model positioned people agents in a bounded region of a Euclidean 2D plane and moved these around randomly by repositioning them at each time tick. Movement was then constrained so these agents could only move a set distance at each time tick. A basic visualisation was developed which depicted agent movements as lines on an image. Next the concept of destinations was developed, so that rather than necessarily having a different destination at each time tick, an agent might be assigned a destination beyond its maximum range for movement in a time tick which it would gradually move to over successive ticks.

Models were generated with agent origins and destinations initialised in various ways and the output images were studied. None of the resulting images of these basic origin-destination models looked as 'road like' as random movement models. The agent routes were too direct and unique to appear collectively like a road network and so were somewhat unrealistic. Some form of agent-environment and/or agent-agent interaction can be used to encourage agents to share routes. Models were conceived with a small benefit to each agent of using an existing route or sharing journeys with other agents. In a way this implementing these models would be like modelling the emergence of a road network. Rather than do this, focus switched to route agents via an existing road network.

So, adding to the typology of models let us distinguish those that are entirely synthetically made up, and those which are seeded from available data. Also for those models seeded with available data there can be a distinction of those seeded from open use and publicly available data, and those seeded (at least in part) from data which is not.

Data about commuting journeys in the UK is captured by its Human Population Census that has been taking place every ten years. From the 2001 Census, Special Transport Statistics (STS) datasets are available for registered users via the Centre for Interaction Data Estimation and Research (CIDER) [14]. These provide a snapshot about where people lived and worked in the UK in 2001 and are available at a range of levels of spatial detail down to Output Areas which typically contain around 300 people. The STS data contain some details of commuting journey flows (breakdowns by mode of transport and job type variables), but there is no direct linkage to data about the usual times of work of the people represented in the flows. Although the STS data is incomplete and lacking many of the theoretically desirable details, it offers an opportunity to seed some UK traffic models which might be assumed to make them more realistic.

Commuting models for individual UK Local Authority Districts (LAD) were developed based on the STS data. These represented people moving from home to work locations and back again in a daily cycle. Initially, to keep the model simple, all people agents were given the same times (shift) to be at the work location and only the commuting journeys that both started and ended in any of the LAD were considered. As the distance to the work locations was variable, each agents journey would not necessarily start or end at the same time.

Let us consider a commute to work and the need to have an estimate of the time needed for the journey and how this might be calculated. An attempt can be made to compute this using data (such as the distance to be travelled) and it can be learned by timing the undertaken journey. In a basic model without constraints on the amount of traffic located in any place or the amount of traffic moving from one place to another, learning can be done in a single journey and agents can arrive at work on time at the start of their shift and arrive back at home at a predictable time. The unconstrained calculation (although perhaps idealistic and certainly unrealistic) is still of use as it indicates the minimum travel time if no agent got in another agents way. Anyway, despite being based on real-world data the visualisations of these models seemed as unrealistic as any made previously. This was mainly due to the aforementioned observation that agents were rarely sharing routes, but also because no capacity like constraints were being imposed to constrain movement. Additionally there was no other (non-commuter) traffic active which is likely to effect commuter traffic...

To constrain agent movements, two things were considered:

1. Restricting agent movements to a high resolution regular network

2. Routing agents along known transport infrastructure

Code was developed to restricting movements to a high resolution regular network. There are still plans to develop models for simulating the evolution of transport infrastructure through use with this, but little progress has been made...

Effort was focussed on constraining movement to existing transport infrastructure. For this two data options were considered:

1. OpenStreetMap [15] data which is publicly available for the entire world
2. UK Ordnance Survey [16] data available under an academic license via Edina [17]

The first option was made more attractive by the existence of the TravelingSalesman routing API [18] [19]. TravelingSalesman provided a means to route Agents via the OpenStreetMap road network. Code was developed to route an agent using this, but a problem was encountered when trying to route many agents. The issue was that agents needed reference to the route they had planned for a journey and this was a reasonably large amount of data compared with simply knowing a destination location. Also the road network had to be loaded into the fast access memory for the routes to be obtained meaning there was less available memory for initialising agents. To scale up to something city size, a computational resource with a few hundred Gigabyte of fast access memory or some better way to handle the data was needed.

1.4 Developing a Demographic Model

For GENESIS, Belinda Wu [20] and Mark Birkin [21] are developing dynamic demographic models that work with time steps of a year [22] whilst the author is developing similar models with time steps of a day. Many things happen day to day, most importantly in terms of a demographic model, people conceive, give birth and die on specific days. Variations in rates of these within a year could be important. Indeed, the timings of conceptions, births and deaths are more fine grained than days and this might also be important, but in terms of demographic modelling, it can be argued that a resolution of a day is perhaps best. Arguably it will depend on what the model is to be used for and at this stage GENESIS model development is not use case driven, so a day time step keeps options more open than any time step longer than a day. Also, people move house, get married and organise other activities on specific days; and, in many contemporary societies there are holidays when people generally may have very different activity patterns compared with other days. It is known that many activities are seasonal and relate to holidays and this is sure to have demographic consequences.

Explicit linkages between traffic and demographic models is also easier with a demographic model time step of a day. An example link between the models could be the journeys that many people make to a hospital for a birth, the timing of the birth can be given by the demographic model simulation, and the journey for the individuals can be simulated in the traffic model. A more complex feedback example, from the traffic model simulation to the demographic simulation model is to do with conception and the general need for partners to be together for this to happen.

One further reason for a daily time step is computational. There are a similar number of time steps running a model that works with time steps of a day and runs for years as there are for one that works with time steps of a second and runs for days. Even with a time step of a day, demographic models are probably less computationally demanding than traffic models with the same number of agents (depending on the level and complexity of agent movement).

Development of a basic demographic model focusses on the processes of death and birth. A more sophisticated model might include the notion of a home and migration. In the authors GENESIS code there are Male and Female classes extended from the People class representing males and females respectively. For simulating death at each time step, age and gender specific mortality probabilities are used. The next number in a pseudo-random sequence gets compared with the probability for each individual to determine if they become dead. If they become dead, they are no longer active in a basic model simulation and it is sensible to store the agents data on disk and delete it from the volatile fast access memory so this storage can be used to store other data.

For Female agents additional processing is done each time step. Firstly pregnancy (conception) is determined. For each Female that is not yet pregnant (and is of fertile age), their age specific fertility probability is obtained and compared with the next number in a pseudo-random sequence. Those that become pregnant are set a due date in 266 days time. Secondly, all pregnant Female agents are iterated over to determine if any miscarry. For a basic model the miscarriage probability is constant. Thirdly, birth is simulated. In a basic model, the due date is determined at conception and the birth occurs on this date unless the female dies or there is a miscarriage.

In reality conception is much more complex and fertility depends on other characteristics of the female (such as, day in a menstruation cycle, number of existing children, whether using birth control, whether they are in good health) and the fertility and availability of a male. Miscarriage is also variable over the stages of pregnancy and is also likely to be effected by environmental circumstance. The time of birth is also variable and some babies are born prematurely and some are born over term. The calculation of a due date is commonly used to induce birth in some societies. All these things may be taken into account in more complex models.

The basic model allows for the generation of population visualisations, such as Age Gender Plots showing counts of Male and Female Agents of different ages or with different birth dates within each year. Such plots are interesting graphical outputs which demographers commonly study. Automated procedures for generating these types of output using the JFreeChart library [23] were developed. These graphical outputs can be studied to see how a simulation settles into a steady state and seeded and compared with population data.

Much can be done to make basic models more realistic, and more fitting in terms of a real population being simulated. For instance, a contemporary demographic model for the population in mainland China where a one child policy is in place [24] should use a very different Female fertility, one that is more conditional such that if a Female already has one living child, they become much less likely to have another. Furthermore, the probability may depend upon the gender of the existing child. All probabilities used in the demographic models can be modified and different random number sequences can be used in the simulation resulting in different Age Gender Plots. The variation, tends and extremes in multiple simulation runs can then help to reveal some of the uncertainties in population forecasts based on current trends.

Probabilities for a basic model were adjusted so that for a given seed population and pseudo-random sequence, a gradually increasing population was obtained. This was then allowed to run for thousands of years. Providing a way to seed the population from any age and gender distribution allowed city and indeed national population scale simulations on a modest desktop computer.

Examining the distribution of births within a year revealed a major cohort effect as the basic models did not initialise pregnancies prior to the first tick of the simulation. The consequence is that a large number of pregnancies are simulated in the first few days and there is a high level of births 266 days or so after the model starts. It can take many iterations, indeed (depending on age based fertility rates) generations before the rate of births per day begins to average out. To begin a simulation with a more realistic pregnancy rate, one solution is to initialise pregnancies by running the simulation forward but only simulating pregnancy and miscarriage (not birth and death) for a year and then seed the pregnancy and due date back. This can be repeated and should gradually smooth out pregnancy and birth rate and result in start situation for the simulation in which Females may be simulated to give birth in the first iteration.

1.5 Scaling Up

Scaling up to run larger simulations with more agents and input data is partly an issue to do with accessing bigger and better computational resources with a greater amount of memory and processing power. It can also be achieved by using computational resources more effectively and improving efficiencies in the computation steps. It is probably not something that is best done by attempting translation of some part or all of the implementation into what may be perceived as a more efficient language.

For the initial basic GENESIS models it was evident that more use could be made of the slower access persistent disk memory of the computational resources being used, especially when the faster access volatile memory of a processing unit was becoming fully used and the programs failed with out of memory type error. Some form of this data handling is probably needed to run large scale traffic simulations even on the contemporary computers with the largest amounts of fast access memory. Access to such computer resources is also not readily available.

The key to good data handling and memory management for simulation modelling is knowing; what data is needed and in what order, and the approximate amounts of allocatable memory that can be used. If allocatable memory is about to run out, it is best that the model either takes steps to cope before there is a problem, or it fails or copes gracefully when one is encountered (so that if more resources become available, the simulation can be restarted from the point of failure rather than having to be re-run from a much earlier stage).

To phrase this another way: Suppose the fast access memory is running low during a simulation and a reasonably large amount of data is to be created or loaded from persistent memory (disk). Further assume that if nothing else is done, when trying to create or load the data an out of memory type error will be encountered. Furthermore, assume there is plenty of available disk. Now, this is the problem of memory management that I have addressed in GENESIS model development. There are two ways to proceed: One way is to calculate in advance and make more memory available in advance to prevent the error, the other is to try to create or load the data and catch the error when it happens and then swap data around and try again repeatedly. These two approaches are complimentary. There is an overhead in calculating in advance if memory is low and there is an overhead in the try and catch error handling. It is possible to have a concurrent thread keeping track of memory allocation so that a method does not have to wait long to know there is enough memory to continue, but still there is some overhead to this.

In the basic demographic models outlined in Section 1.4, the only data are agent data; whereas in the traffic models outlined in Section 1.3, there are agent, raster, vector and other data. So developing memory handling code for the demographic models was an easier task as well as being a step towards developing memory handling for the traffic models.

Swapping individual agents to disk and back again was found to be expensive computationally due to the overhead of opening and closing files. So, agents were organised into collections and it is these collections became the entities swapped to and from disk. This is similar to how Grids data is organised where data is swapped to and from disk in chunks. Anyway, an agent file store as well as an agent collection file store was wanted. The agent store was to store all data for agents by the end of the simulation and the agent collection store was like a cache extending the memory for a running simulation.

The file structure for agent and agent collection files is such that given a numerical identifier, and the location of the top level directory, the file location for storing the serialised Object and other data can be calculated. In other words the location of the files are implicit given some configuration parameters and an simple numeric identifier (ID). With most operating systems there is a maximum number of files it is sensible to store in a directory. Also, there is a maximum length of file name (including the directory path) it can cope with. Setting the maximum number of files to be stored per directory to be 100 is a reasonable compromise and it makes it relatively easy for a human to find a particular file. A million million files can be stored in directory structure with a depth of six directories branching by 100 each time. Consider this as follows:

- 1
- 100 = 100 * 1
- 10000 = 100 * 100
- 1000000 = 100 * 10000
- 100000000 = 100 * 1000000
- 10000000000 = 100 * 100000000
- 1000000000000 = 100 * 10000000000

So, files for an agent with ID equal to zero would be stored in the directory that can be addressed as follows:

- Agents/
- 000000000000-199999999999/
- 0000000000-0199999999/
- 00000000-01999999/
- 000000-019999/
- 0000-0099/
- 0/

Similarly, files for an agent collection with ID 123456789 would be stored in the directory that can be addressed as follows:

- AgentCollections/
- 000000000000-199999999999/
- 0100000000-0199999999/
- 23000000-23999999/
- 450000-459999/
- 6700-6799/
- 89/

In attempting to create or retrieve data from disk, care is needed so memory handling operations do not get stuck in loops swapping to file the data that is required. Indeed, memory handling needs various sophistications to work well. For each of the library the author has developed that attempt memory handling, if that instance cannot cope with the memory handling using methods internal to the library, it throws the error or an exception to a higher level manager. Rather than detail any memory handling code here, readers that are interested are referred to the latest releases of the authors code which is openly available on-line [25]. Much optimisation has been implemented in this memory handling, but this is perhaps only a start to the work needed to generate spectacular city or larger scale simulation results. The next section outlines what is planned in terms of computational enhancement to the GENESIS modelling effort.

1.6 Further Work

GENESIS models can be developed in lots of ways, some have been suggested in the preceding sections. Simulations can be run for synthetic data under different scenarios for different parts of the world seeded with data about the local population and environment. The long term goal which is to integrate the traffic and demographic models requires a number of refinements in each model first.

The next steps for enhancing traffic models is to develop code to constrain traffic to specific densities on the networks and allow for queueing of traffic. With this implemented, a set of simulation results should be produced and examined. Some outputs that help identify where network congestion occurs in the models might be wanted and some comparison of the simulated results with what is known about traffic in reality might be of general interest. Next, for commute journeys code to allow agents to try alternative routes and also allow for dynamic re-routing around queues is wanted. For the UK models, more journeys including journeys to school and to places of entertainment might be modelled. For some of this work in the UK I hope to make use of the linked data being made available via the data.gov.uk initiative [26].

The next step for enhancing demographic models is to develop code to explicitly model male parentage and develop a process for partnering parents. People agents will hold references to collections of other agents that they have been close to and these will more likely be those that are partnered. A crucial next step is to evolve spatially explicit demographic models and begin to model the processes of migration. Many other things can then be considered, such as explicitly modelling change in health status, but there is much to do before this!

In terms of computation, one of the key challenges is to parallelise the models to exploit distributed resources. This is non-trivial and there are various options that need to be explored for partitioning the models. Similarly, with the multi-core nature of many compute resources, threading the processing should be considered not least in terms of memory management. Various technology (e.g. Virtual File System, database software and persistence frameworks) can be investigated in this respect.

Perhaps key to all future work is to build up a community of users to test code and get involved in developing and using models.

1.7 Summary and Concluding Remarks

Development of two different types of simulation models has been outlined: traffic models that operate on the scale of days with a second time step, and a demographic models that operate over the time scale of a number of years with a day time step. There is a long term goal to somehow integrate these into general geographical social simulation models and use these to study various socio-economic and environmental interactions. Basic traffic and demographic models have been developed separately and work has bee done to produce simulation outputs at a city scale of up to a million individual agents. The models share the same code base and less basic models are being worked on to produce more realistic outputs.

There are significant computational challenges in social simulation work. Running a model for the contemporary UK involves modelling and recording changes in the locations and other aspects of the state of agents representing approximately 60 million real people. The entire history of an agent in terms of the interactions it has with other agents and the environment could be massive and there is a challenge to resources the storage of these data if national level simulation models are wanted.

A generic data storage solution has been outlined and its use in memory handling for scaling simulation models described. The generic data storage solution is for storing data in files using a simple branching directory structure which allows for data to be stored in file locations that are implicit from a number used as the identifier for the data.

Acknowledgements GENESIS is funded by the ESRC with research grant RES-149-25-1078. This work is the result of collaboration and the author would like to mention in particular the help and encouragement of the e-Research community; developers of open source software and Java; the University of Leeds and its School of Geography and Centre for Computational Geography. Paul Townend provided a method to get the total free memory available to the Java Virtual Machine without which, methods that try to ensure there is enough memory to continue could not be developed.— [27] Alex Voss for taking an interest in social simulation and collaborating with me as a geographer.— [3] Thanks to the reviewers of this work. An initial review in the context of the target publication lead to a number of changes to this work which have made it more appropriate for a general audience. Some suggestions in the review have also given impetus for further publication based on this work to address its various shortcomings.—

References

1. Andy Turner's GENESIS Project Web Page
http://www.geog.leeds.ac.uk/people/a.turner/projects/GENESIS. Cited 8 June 2010
2. The EUAsiaGrid Website
http://www.euasiagrid.org/. Cited 8 June 2010
3. Alex Voss' University of St Andrews Web Page
http://www.cs.st-andrews.ac.uk/%7Eavoss. Cited 8 June 2010
4. Repast Sourceforge Web Page
http://repast.sourceforge.net/. Cited 8 June 2010
5. Andy Turner's NCeSS Sakai Portal: My Workspace Worksite: Wiki: ISGC 2009 Page
http://portal.ncess.ac.uk/access/wiki/site/%7Ea.g.d.turner%40leeds.ac.uk/isgc%202009.html. Cited 8 June 2010
6. David Fergusson's NeSC Web Page
http://www.nesc.ac.uk/nesc/staff/dfergusson.html. Cited 8 June 2010
7. NCeSS Sakai Portal: SocSim Tutorial Worksite: Wiki: Social Simulation Tutorial at 5th International e-Social Science Conference 2009 Page
http://portal.ncess.ac.uk/access/wiki/site/socsim/social simulation tutorial at 5th international e-social science conference 2009.html. Cited 8 June 2010
8. Rob Procter's MeRC Web Page
http://www.merc.ac.uk/?q=Rob. Cited 8 June 2010

9. Alex Voss' ISGC 2010 Social Simulation Tutorial Wiki Page
https://e-research.cs.st-andrews.ac.uk/site/bin/view/SocialSimulation/ISGC2010/. Cited 8 June 2010
10. Andy Turner's Grids Java Library Web Page
http://www.geog.leeds.ac.uk/people/a.turner/src/andyt/java/grids/. Cited 8 June 2010
11. FOSS4G2006 Conference Website
http://www.foss4g2006.org. Cited 8 June 2010
12. Andy Turner's Generic Java Library Web Page
http://www.geog.leeds.ac.uk/people/a.turner/src/andyt/java/generic/. Cited 8 June 2010
13. Andy Turner's NeISS Project Web Page
http://www.geog.leeds.ac.uk/people/a.turner/projects/e-ISS. Cited 8 June 2010
14. CIDER Website
http://cider.census.ac.uk/. Cited 8 June 2010
15. OpenStreetMap Website
http://www.openstreetmap.org/. Cited 8 June 2010
16. Ordnance Survey Website
http://www.ordnancesurvey.co.uk/. Cited 8 June 2010
17. Edina Website
http://www.edina.ac.uk. Cited 8 June 2010
18. Sourceforge Traveling Salesman Wiki Main Page
http://sourceforge.net/apps/mediawiki/travelingsales/. Cited 8 June 2010
19. Sourceforge Traveling Salesman Project Web Page
http://sourceforge.net/projects/travelingsales/. Cited 8 June 2010
20. Belinda Wu's University of Leeds Web Page
http://www.geog.leeds.ac.uk/people/b.wu. Cited 8 June 2010
21. Mark Birkin's University of Leeds Web Page
http://www.geog.leeds.ac.uk/people/m.birkin. Cited 8 June 2010
22. Wu, B.M., Birkin, M.H. and Rees, P.H.: A spatial microsimulation model with student agents, Journal of Computers, Environment and Urban Systems (32) pp. 440-453 (2008) doi: 10.1016/j.compenvurbsys. 2008.09.013
23. JFreeChart Web Page
http://www.jfree.org/jfreechart/. Cited 8 June 2010
24. Wikipedia One-child_policy Page
http://en.wikipedia.org/wiki/One-child_policy. Cited 8 June 2010
25. Andy Turner's Open Source Java Web Page
http://www.geog.leeds.ac.uk/people/a.turner/src/andyt/java. Cited 8 June 2010
26. data.gov.uk Website
https://www.data.gov.uk/. Cited 8 June 2010
27. Paul Townend's University of Leeds Web Page
https://www.comp.leeds.ac.uk/people/staff/scspmt.html. Cited 8 June 2010