A NeISS Collaboration to Develop and Use e-Infrastructure for Large-scale Social Simulation

Tom Doherty[1], Sam Skipsey[2], Andy Turner[3], and John Watt[1]

[1]National e-Science Centre, University of Glasgow
[2]School of Physics and Astronomy, University of Glasgow
[3]Centre for Computational Geography, University of Leeds

The National e-Infrastructure for Social Simulation (NeISS) project is focused on developing e-Infrastructure to support social simulation research. Part of NeISS aims to provide an interface for running contemporary dynamic demographic social simulation models as developed in the GENESIS project. These GENESIS models operate at the individual person level and are stochastic. This paper focuses on support for a simplistic demographic change model that has a daily time steps, and is typically run for a number of years.

A portal based Graphical User Interface (GUI) has been developed as a set of standard portlets. One portlet is for specifying model parameters and setting a simulation running. Another is for comparing the results of different simulation runs. Other portlets are for monitoring submitted jobs and for interfacing with an archive of results. A layer of programs enacted by the portlets stage data in and submit jobs to a Grid computer which then runs a specific GENESIS model program executable. Once a job is submitted, some details are communicated back to a job monitoring portlet. Once the job is completed, results are stored and made available for download and further processing. Collectively we call the system the Genesis Simulator.

Progress in the development of the Genesis Simulator was presented at the UK e-Science All Hands Meeting in September 2011 by way of a video based demonstration of the GUI, and an oral presentation of a working paper. Since then, an automated framework has been developed to run simulations for a number of years in yearly time steps. The demographic models have also been improved in a number of ways. This paper summarises the work to date, presents some of the latest results and considers the next steps we are planning in this work.

***Key words*: social simulation, demographic, stochastic, model, grid, e-Infrastructure, automate, framework, NeISS, GENESIS.**

Proc. R. Soc. A 1–15; doi: 10.1098/rspa.00000000
January 18, 2012

1. Introduction

Social simulation is an attempt to model societies in a dynamical way in a digital computer (Gilbert and Troitzsch, 2005).

> "e-Infrastructure consists of social and technical arrangements around advanced, networked information and communications technologies that can enable new research practices and methods" (Voss et al., 2009).

This work is part of the National e-Infrastructure for Social Simulation (NeISS) project, which is attempting to develop e-Infrastructure to support social simulation research (NeISS, 2010).

Section 2 provides some background. Section 3 details the current e-Infrastructure and workflow for generating results. Section 4 presents some results with discussion and details the quantities of Grid resource used in terms of CPU hours, model memory requirement and the amount of storage used for results. Section 5 outlines next steps we are planning for this research collaboration. Section 6 briefly concludes.

2. Background

NeISS is a project funded for three years by the UK Joint Information Systems Committee (JISC) under its Information Environment programme (NeISS, 2010). NeISS started in April 2009 and involves partners from 8 different institutions. In April 2010, one small part of this reasonably large collaboration started to develop an interface for running GENESIS demographic daily time step social simulation models. This paper details this work to date.

A variety of contemporary focused social simulation models were developed as part of the Generative e-Social Science for Socio-Spatial Simulation (GENESIS) project. GENESIS was funded as a second phase research node of the UK National Centre for e-Social Science by the UK Economic and Social Research Council (ESRC). The funding supported the project for three years starting in October 2008 (GENESIS, 2008). One focus of GENESIS was the development of individual based dynamic demographic social simulation models.

This paper, focusses on the development of NeISS e-Infrastructure for supporting a simple GENESIS model of demographic population change. An initial simulation inputs some population summary count data along with

mortality, fertility and miscarriage probabilities. The seed of a pseudo-random number generator (RNG) is also input. This allows for the production of range of stochastically generated results. Simulations can be continued with or without resetting the RNG for subsequent time periods. Additionally, the probabilities used for mortality, and for determining if a female of a specific age will become pregnant or have a miscarriage may also be modified for the continuation simulations.

The model is simplistic not least because it does not incorporate a migration component that details the movement of people from one location to another. Yet, the model does explicitly handle multiple births and miscarriage, and in that we believe it is novel. Indeed, the model is perhaps a first attempt to model demographic change at an individual level, for a daily time steps.

The results presented in Section 4 are for Leeds Local Authority District simulations for the time period from 1991 until 2001. These simulations are based on: 1991 Census data population counts; annually produced UK National Statistics about mortality and fertility, and some general statistics about miscarriage rates. The results were produced both on a Grid Computer and on an average specification desktop machine.

All the source code for the Genesis Simulator including both the supporting e-Infrastructure programs and those for the GENESIS models are open source and available under GNU Lesser General Public License (GNU, 2007).

3. e-Infrastructure and workflow

A web portal based Graphical User Interface (GUI) has been developed as a set of standard JSR-168 portlets. One portlet is for specifying model parameters and setting a simulation running. Another portlet is for specifying simulation results to compare and setting a comparison job running. Other portlets are for monitoring submitted jobs and for interfacing with an archive of results. Currently there is only support for deleting results from the archive, but work is on-going to allow the full results to be transferred elsewhere out of the archive. The portlet and the underlying programs that have been developed for this work are made available via Tom Doherty's NeISS Source Code Web Page (Doherty, 2011).

Programs enacted by the portlets stage data in and submit jobs to a Grid computer. Once a job is submitted some details are communicated back to the user via a job monitoring service. Once the job is completed, the full result is archived and smaller parts of the results are stored and made available for download.

The portlets have been developed and tested on the DAMES Application Portal hosted at the National e-Science Centre (NeSC) in Glasgow (DAMES, 2010). Being standard JSR-168 portlets (Sun, 2003), the set of portlets can be readily migrated to work within other portals, however, the model will only run with the other required programs in place.

The model itself is implemented in Java and is made available via Andy Turner's GENESIS Source Code Web Page (Turner, 2009). The GENESIS source code forms a library of packages with dependencies on a large number of third party open source libraries. It contains code for running other types of geographical models and some of the dependencies for the library are not required for reproducing the results presented in Section 4. For the Genesis Simulator two programs were compiled from the library. One is the 'simulation model program', the other is the 'simulation model results comparison program'. These compiled programs are passed with the data in workflow execution and require Version 1.6 or later of the Java Runtime Environment (JRE) to run.

The results in Section 4 are for 10 years (1991 to 2001) of simulation for Leeds Local Authority District in the UK. The general workflow is shown in Figure 1. For each year, four simulations are run using different pseudo random number sequences that produce a range of results. These simulations were originally run in pairs, but they may as well have been run singularly. One of the results (from the two pairs of simulation results for each year), is selected for use as input for the next year to be simulated. The selection is based on a comparison program output which details and summarises differences between simulated and input mortality and fertility rates for each simulation. For the results described in Section 4, the simulation result that is calculated as most similar to an expected result and with mortality and fertility rates closest to those input is selected for continued simulation in the next year of simulation.

[Figure 1 about here.]

The e-infrastructure uses the National Grid Service (NGS, 2012) Workload Management System (WMS) (SA3, 2004) to provide resource brokering-based job scheduling across all the Grid sites that support the NeISS VO (VOMS, 2010). The NGS user interface (UI) (NGS, 2010b) provides a gateway for WMS job submission to NGS and GridPP (GridPP, 2008) nodes via a command line interface. Virtual Organisation Membership Service (VOMS) (Alfieri et al., 2003; NGS, 2009) proxy certificates allow users to submit Grid jobs using a Job Description Language (JDL) file that specifies the job parameters (Pacini, 2005). These parameters detail the type of file to execute on the worker node, abstract file paths for uploading data via the input sand box, and abstract file paths for retrieving data from the output sand box.

Currently, the job staging script is executed on a single worker node and it handles the running of the model and pushing of results into a data archive organised via a Logical File Catalog (LFC) (GridPP, 2010b). On successful submission, a job id is returned which is used to monitor the job status from the portal interface which calls the glite-wms-job-status command in the background. Once the job has run, result files are pushed automatically via the output sandbox to be stored in the archive. Summary results are also passed to a portlet for display and download.

Without the job staging script and portal based GUI, a user would have to issue the appropriate glite-wms-job-submit command to submit the job and associated JDL file to the WMS and associate a delegated VOMS proxy with it. To fetch the job output once the job completed, they would also have to issue an appropriate glite-wms-job-output command. All this command line work is now hidden from the user behind the simple and easy to use portal based GUI.

The complexity of Grid middleware coupled with grid certificates has proven to be a barrier to entry for researchers in some disciplines (Jensen et al., 2007). As portals were already being used within the NeISS project, it was fitting to develop and use portlets to facilitate job-submission, status-monitoring, output retrieval and job output comparison. For the SARoNGS (SARoNGS, 2010) implementation the OMII SPAM-GP Shibboleth module (Jiang et al., 2008) is used to login to the portal framework, and an iFrame (Raggett et al., 1999) within one of the portal pages provides a link to the NGS Credential Translation Service (NGS, 2010a) GUI from which users are to generate and download their SARoNGS certificate. This is still to be integrated as we await a certificate fix to be rolled out to GridPP sites. A Registration Portlet allows the information provided in the portal account and the generated SARoNGSs proxy to be used to contact the VO administrator to request membership. Job metadata and monitoring is recorded in a database which makes it possible to track and manage each job submitted. A Management Portlet allows for the deletion of results data stored in the database and the associated archived data. The main Job Submission Portlet is designed with a 'wizard' type flow where the user enters the necessary information for the simulation in a step by step fashion. The certificate and JDL configuration is transparent to the user.

Archive data management leverages the existing WLCG (CERN, 2010) gLite (EMI, 2010) infrastructure, in order to reduce the amount of additional work needed to support it across potential sites . Files generated by a job are stored at the local Storage Element (for the UK, most often a DPM (GridPP, 2010a) in front of a disk pool), and registered in the UK LFC at RAL (Wikipedia, 2009). Later jobs can be directed to sites holding local copies of required data, and copies of data can be replicated at other sites,

as the files are managed entirely in terms of their Globally Unique Identifier (GUID) (Wikipedia, 2003) assigned by the LFC.

The main components and workflow of the Genesis Simulator e-Infrastructure are depicted in Figure 2.

[Figure 2 about here.]

Since the UK e-Science All Hands Meeting 2011, the workflow has been automated. It became apparent after stepping through the workflow manually for each year that this process is quite cumbersome. Not so much in the case of acquiring one off results for this paper. But as we scale upwards and also plan to fine tune the simulation model itself then we know that the number of iterations required to run these models using this workflow will increase dramatically. We have therefore decided to design an automated framework to automatically run the workflow for us.

We first investigated introducing Direct Acyclic Graph (DAG) type jobs in to our JDL files (Pacini, 2005). This type of JDL file allows for the chaining of Grid jobs together and allows the output of one job to be used as an input to another. The simulation job output could then have automatically been provided as input to the comparison job and this process repeated for the year range 1991 to 2001. Unfortunately this technology has not been implemented yet for the type of Grid Computing Element (CE) called CREAM that is available to us (EGEE, 2010). We can although use another type of JDL file called Parametric (Pacini, 2005). This job type allows a set of jobs to be created and run concurrently that only differ in arguments. In our case for the simulation jobs we could create a JDL file to run all of the simulation seed jobs concurrently. The new automation framework is designed to use the job monitoring service extensively. The intelligence on how far advanced a job is in the workflow year range and what to do next depending on the job type (Simulation or Comparison) is all controlled by this service. This means a new iteration of the job submission portlet is needed that allows for details such as year range, job group name, pseudo random number seed initial value and increment, number of runs (in the AHM Leeds study case this is four) and the means for uploading all the area probability files for mortality, fertility and miscarriage for each year in one zipped file. As the comparison step will now be handled automatically and under the hood there is no need for the comparison portlet for this framework. Jobs are now grouped together in the summary portlet under the same job group name. A report is created in HTML format (Wikipedia, 2011) and made available on the summary portlet for each simulation job so that the output for all of the jobs for each seed can be grouped together and made available as before via HTML links. This framework is currently in the advanced stage of testing and will be

available immediately for the next stage of data taking as we scale to the next level.

4. Demographic Simulation Example Scenario

In this section, the demographic model is described in more detail and demographic model outputs are presented along with details about resource usage. The results presented in this section are based on a 10 year simulation for the Leeds Local Authority District in the UK from 1991 to 2001. The many simplifying assumptions of the model are introduced along with the probabilistic work that generates the probabilities used by the model. (The focus of this paper is on the e-Infrastructure developments and the simulation workflow. For this reason and for brevity, an in depth review of these simplifying assumptions and details of the probabilistic work are not presented. It is intended that these will be written up in another publication.)

To recap, the workflow for the multiple year simulation is shown in Figure 1. Each yearly step is repeated a number of times varying only the random number seed. All the outputs for the same simulation step are then compared and the best result is evaluated and set to be used as the input for the next year of simulation.

The simulation outputs include a snapshot of the individual level population, a set of aggregated statistics, and various other metadata. The statistical outputs include age by gender population summary data and images that depict the population alive at the end of yearly time steps and the population that died during it. All this data can be downloaded from the Genesis Simulator. The result can be readily recreated at another site supporting the workflow execution using metadata provided with the output.

The simulated population is initialised from age by gender count statistics from the 1991 Census of Population Small Area Statistics Table 2 Office for Population Censuses and Surveys (1991). The age categories are as follows: 0 to 4; 5 to 9; 10 to 14; 15; 16 to 17; 18 to 19; 20 to 24; 25 to 29; 30 to 34; 35 to 39; 40 to 44; 45 to 49; 50 to 54; 55 to 59; 60 to 64; 65 to 69; 70 to 74; 75 to 79; 80 to 84; 85 to 89; 90 and over. An approximately equal division of ages in single years was assumed. The population was effectively dealt out in the individual age categories so that younger ages were assigned remaining individuals in cases with not an exact division across all ages. A maximum age of 104 was assumed. An age by gender individual years of age plot for this population is shown in Figure 3.

[Figure 3 about here.]

Annual mortality rates for all years were estimated by dividing counts of deaths by the 1991 population count for each age and gender class. The UK Office for National Statistics Vital Statistics (ONS, 2010) that were made available for this work included counts of deaths by age and gender for the following age categories: 0 to 4; 5 to 9; 10 to 14; 15; 19; 20 to 24; 25 to 29; 30 to 34; 35 to 39; 40 to 44; 45 to 49; 50 to 54; 55 to 59; 60 to 64; 65 to 69; 70 to 74; 75 to 79; 80 to 84; 85 and over. Fortunately there was good correspondence between these age classes and the 1991 Census of Population Small Area Statistics Table 2 classes. Using 1991 population estimates as the denominator for mortality rate in any year other than 1991 is questionable. Arguably it would have been better to use mid-year population estimates for each of the years to calculate the annual mortality rates.

Daily mortality probabilities were estimated from the annual mortality rates (assuming an even likelihood of death on any day of the year and a fixed number of 365 days in any simulation year) as follows: Firstly, annual survival probabilities were computed from the annual mortality rates (for each specific age and gender class). This simply involved taking the annual mortality rate from one. The next step was to take the 365th root of the annual survival probability to compute the probability of survival on any day. Daily mortality probabilities were then computed by taking the probability of survival on any day from one.

As well as representing each individual in the population as a unique entity, the population initialisation also involved assigning individually represented people with a date of birth and hence a birthday within a year (when their age in years gets incremented in simulation). For this, an even spread of birthdays across the year was assumed.

Fertility data for 1991 was used to initialise pregnancies in the initial population and to estimate the probabilities that non-pregnant females would become pregnant on a daily time step during simulation. The UK Office for National Statistics Vital Statistics (ONS, 2010) that were made available for this work included counts of births (by age of mother at birth) and mid-year female population estimates for the following age categories: less than 20; 20 to 24; 25 to 29; 30 to 34; 35 to 39; 40 and over. These data were combined with four other probabilities: the probability of twins and triplets; and two miscarriage probabilities, an Early Pregnancy Loss probability which is relevant up to 42 days of pregnancy, the other, Clinical Miscarriage probability which is for the remainder of the pregnancy term. In the simple model, these probabilities were assumed not to vary dependent on the age of the mother or given any historical information about previous pregnancies.

A female assigned as pregnant in the model is also assigned a due date and the gender of the unborn babies is specified. The genders of the unborn babies

are stochastically assigned assuming equal chances of the gender being classed male or female at birth (and also assuming these are the only possibilities). The model assumes a fixed pregnancy term duration of 266 days. The probabilities results in significantly more females being only a few days pregnant than females being within a few days of giving birth although the differences are actually slight.

The population initialisation is an important part of the simulation model. It is an attempt to produce an unbiased start point and one which has the appropriate chance of simulating births and miscarriages on day one of the simulation. One further thing to note is that the population was initialised as the population at the very beginning of 1991. This section has given a brief overview of the assumptions of the model and hints at the complexity of the probabilistic work that can be involved in demographic modelling for daily time steps. It continues by presenting some results.

Figure 4 presents the theoretically estimated living population at the start of 1992. This is calculated by applying the annual mortality and fertility probabilities and ageing the population by a year. Figure 6 presents the best fitting simulated population at the start of 1992. Figure 5 and Figure 6 respectively present the theoretically expected and simulated dead populations for the year 1991. There is not much difference in the shapes in Figure 4 and Figure 6 and Figure 5 and Figure 6 which is encouraging from a demographic modelling perspective.

[Figure 4 about here.]

[Figure 5 about here.]

[Figure 6 about here.]

[Figure 7 about here.]

There is little overall variation in the shapes of the output demographics for each simulation of 1991. Four different random number seeds were used to produce a range of results in our example use. These do produce a range of results and a comparison between the theoretically expected and simulated populations forms the basis for a comparison. In addition, for each simulation, counts of miscarriages and multiple births and the number of days in pregnancy of early and late stages and the number of days lived in each age and gender group are calculated. From these annual rates can be calculated and compared with the values that are used to initially specify the model parameters. An aggregate measure forms the basis of a comparison and the best result is input for further simulation. Figure 7 depicts the best fitting simulated population at the start of 2001.

[Figure 8 about here.]

The minimum memory requirement for the model has not been calculated, but 1.2GB is set for the JVM in model execution in producing these results. As we scale to larger populations there is likely to be a larger memory footprint, but as most of the data is stored on slower access memory, it is effectively only the indexes for the data that grow in memory. Indeed, the indexes can be stored in collections also and swapped to disk if memory issues are encountered. The largest result for any yearly simulation step is 2.2GB. All results (4 for each time step and 10 time steps) for Leeds requires 53GB of storage.

In terms of resource usage to date. Just under 2160 hours (90 days) of CPU time has been used by the NeISS VO on Grid Computer resources. About 12% of this (just under 280 hours or 11 days) was for the generation of the results described above. This is the amount of resource in terms of CPU that would be needed to recreate all the result. However, less than 25% of that is required to create the results that are regarded as the best.

The comparison jobs run for about a minute or so, whereas the simulation runs for hours.

5. Discussion

The results generated to date are only for testing and developing the model and e-Infrastructure. The demographic model for which the results are presented is very basic. It does not have a migration component which represents the movement of people between different regions. Migration, is known to have a large effect on the population of the Leeds Local Authority District (Wu et al., 2008). It is also a very basic demographic model in other respects. It does not have a coupling or marriage representation and it does not represent paternal relationships. Indeed the model is so basic, that the results are not recommended for use in support of applications, however the results are useful benchmarks and the model is useful because of its relative simplicity as it is more likely that we will be able to validate it.

We are in the process of scaling up to produce results for England, enhancing the Genesis Simulator as we go to readily make use of more computational resources across multiple sites.

Computational demands for the results presented here were met by resources at the NGS/GridPP ScotGrid site (ScotGrid, 2010). In the next phase of work, we are collaborating with other GridPP sites to use their resources. GridPP have been key collaborators approving the NeISS VO and providing a scalable storage solution with 2TB of data for the Genesis Simulator at the ScotGrid site.

The e-infrastructure uses the Virtual Organisation Membership Service (VOMS) solution (Alfieri et al., 2003) to handle access to NGS and GridPP resources. The aim is to use the SARoNGS approach (SARoNGS, 2010), but while this is being organised, effectively a single certificate is being used with an appropriate accounting system in place. Hopefully the SARoNGS solution will be in place before we try to enact workflows at other sites.

In the future we aim to report a further scaling up which run the simulations for all Local Authority Districts in England. Additionally, we aim to report details about the probabilistic work involved.

6. Conclusion

The Genesis Simulator is an attempt to Grid enable some geographical models. All the source code of the model and bespoke e-Infrastructure middleware is available as open source under the GNU Lesser General Public License GNU (2007). The key to e-Infrastructure development is collaboration between the researchers and this has been working well for the NeISS work outlined in this paper. We hope that by the end of the NeISS project, the Genesis Simulator will have been used to produce some demographic model simulation results for England from 1981 until 2012 for a model that explicitly handles migration and that the results for 2012 will be in line for submission to the UK Data Archive Economic and Social Data Service (ESDS, 2012).

The paper presents an original attempt to support the development and use of demographic models that operate at the individual level at a daily resolution in a scalable way.

It is nearly always the case that in developing software for analysis or modelling that a large amount of resource is used in development. The estimated resource use given in Section 4 does not account for all the effort and computation that has gone into developing the Genesis model. The estimates given are for testing and configuring the models to run on specific Grid resources, which although a considerable effort, is only a fraction of the overall effort that has gone into model development in GENESIS.

The results planned to be generated for all of England will further test the abilities of the e-Infrastructure. The demands on computational resources are expected to be around 100 times greater when simulating all of England compared to Leeds .

As the size of the population to be simulated increases and the number of steps in the simulation increases, the model demands more and more computational resource. A UK national simulation has requirements of input and output data in the size region of terabytes, and compute times for atomic

model components that run for several days. We believe that the e-Infrastructure is capable of this, but the availability of large grid and cloud computing resources is not a certainty.

Producing larger (hopefully more impressive) simulation results should improve understanding of how well the model scales and produce results of greater interest to the demographic modelling community. We expect issues as we approach and push the boundaries of what is computationally feasible with available resources and what is implementable in terms of demographic modelling.

Acknowledgment

The Vital Statistics on births and deaths and mid-year estimates of population which underpin this work were supplied by the Office for National Statistics to Paul Norman for ESRC Research Awards RES-163-25-0032 (ESRC, 2008) and RES-189-25-0162 (ESRC, 2011). These data are Crown copyright and are reproduced with permission of the Office of Public Sector Information. We are very grateful to Paul for sharing these data with us.

This work was supported directly with funding from: JISC under the Information Environment Programme 2009-11 as NeISS; and, ESRC as part of the GENESIS project: ESRC Research Awards RES-149-25-1078

The authors would like to acknowledge the use of the UK National Grid Service in carrying out this work. On the computational side, we are also especially grateful for the support of GridPP and ScotGrid in particular that supported us with the provision of 2TB of storage for our results.

The authors are based at the University of Glasgow and the University of Leeds and are very grateful for the support of these institutions.

References

R. Alfieri, R. Cecchini, V. Ciaschini, L. dell'Agnello, A. Frohner, A. Gianoli, K. Lorentey, and F Spataro. VOMS, an Authorization System for Virtual Organizations, 2003. URL `https://twiki.cnaf.infn.it/twiki/bin/viewfile/VOMS/WebDocumentation?rev=1;filename=VOMS-Santiago.pdf`.

CERN. Worldwide Large Hadron Collider (LHC) Computing Grid (WLCG) Technical Site, 2010. URL `http://lcg.web.cern.ch/LCG/`.

DAMES. DAMES Applications Portal, 2010. URL `https://dames.nesc.gla.ac.uk/`.

T. Doherty. Tom Doherty's NeISS Source Code Web Page, 2011. URL http://ppewww.ph.gla.ac.uk/~tdoherty/NeiSSCode/NeiSS_Source_Code.htm.

EGEE. The Computing Resource Execution And Management (CREAM) Service Home Wiki Page, 2010. URL http://grid.pd.infn.it/cream/.

EMI. gLite Lightweight Middleware for Grid Computing, 2010. URL http://glite.cern.ch/.

ESDS. The Economic and Social Data Service (ESDS), 2012. URL http://www.esds.ac.uk/.

ESRC. What happens when international migrants settle? ethnic group population trends and projections for uk local areas under alternative scenarios. esrc project web page, 2008. URL http://www.esrc.ac.uk/my-esrc/grants/RES-163-25-0032/read.

ESRC. Ethnic group population trends and projections for uk local areas: dissemination of innovative data inputs, model outputs, documentation and skills. project web page, 2011. URL http://www.esrc.ac.uk/my-esrc/grants/RES-189-25-0162/read.

M. Fisher, J. Ellis, and J. Bruce. *JDBC API Tutorial and Reference*. Addison-Wesley, 2003.

GENESIS. GENESIS Project Home Web Page, 2008. URL http://www.genesis.ucl.ac.uk/.

N. Gilbert and K.G. Troitzsch. *Simulation for the Social Scientist*. 2005.

GNU. GNU Lesser General Public License, 2007. URL http://www.gnu.org/licenses/lgpl.html.

GridPP. GridPP Home Web Page, 2008. URL http://www.gridpp.ac.uk/.

GridPP. GridPP Disk Pool Manager (DPM) Wiki Page, 2010a. URL http://www.gridpp.ac.uk/wiki/Disk_Pool_Manager.

GridPP. GridPP Logical File Catalog (LFC) Wiki Page, 2010b. URL http://www.gridpp.ac.uk/wiki/LCG_File_Catalog.

Internet2. Shibboleth Framework, 2012. URL http://shibboleth.internet2.edu.

J. Jensen, D. Spence, and M. Viljoen. A Scalable PKI for a National Grid Service, 2007. URL http://middleware.internet2.edu/pki07/proceedings/11-jensen-pki-national-grid.pdf.

J. Jiang, T. Doherty, and J. Watt. Security Portlets simplifying Access to and Management of Grid Portals (SPAM-GP), 2008. URL http://www.nesc.gla.ac.uk/projects/omii-sp/index.html.

NCSA. MyProxy Credential Management Service Web Site, 2000. URL http://grid.ncsa.illinois.edu/myproxy/.

NeISS. NeISS Project Home Web Page, 2010. URL http://www.neiss.org.uk.

NGS. NGS VOMS Web Page, 2009. URL http://www.ngs.ac.uk/site-level-services/voms.

NGS. NGS Certification Autority (CA) Hierarchy Wiki Page, 2010a. URL http://wiki.ngs.ac.uk/index.php?title=NGS_CA_Hierarchy.

NGS. NGS User Interface Workload Management System Resource Broker, 2010b. URL http://www.ngs.ac.uk/ui-wms. (Unsure of the Year of Publication).

NGS. The UK National Grid Service (NGS) Home Web Page, 2012. URL http://www.ngs.ac.uk.

Office for Population Censuses and Surveys. 1991 Census: Small Area Statistics (England and Wales) [computer file]., 1991.

ONS. Office for National Statistics Vital Statistics on births and deaths and mid-year estimates of population, 2010.

F. Pacini. Job Description Language (JDL) Attributes Specification (Submission through the WMProxy Service), 2005. URL https://edms.cern.ch/document/590869/1.

D. Raggett, A. Le Hors, and I. Jacobs. HTML 4.01 Specification: Inline Frames: The IFRAME Element, 1999. URL http://www.w3.org/TR/1999/REC-html401-19991224/present/frames.html.

SA3. The gLite Workload Management System, 2004. URL http://glite.web.cern.ch/glite/wms/.

SARoNGS. Shibboleth Access to Resources on the National Grid Service (SARoNGS) JISC Project Web Page, 2010. URL http://www.jisc.ac.uk/whatwedo/programmes/einfrastructure/sarongs.aspx.

ScotGrid. ScotGrid Home Web Page, 2010. URL http://www.scotgrid.ac.uk/.

O. Sukhoroslov. A lightweight Java API and command-line interface for gLite, 2009. URL http://code.google.com/p/jlite/.

Sun. Java Specification Request 168: Portlet Specification, 2003. URL http://www.jcp.org/ja/jsr/detail?id=168.

A.G.D. Turner. Andy Turner's GENESIS Source Code Web Page, 2009. URL http://www.geog.leeds.ac.uk/people/a.turner/src/andyt/java/projects/GENESIS/.

VOMS. VOMS admin for VO: neiss.ac.uk, 2010. URL https://voms.ngs.ac.uk/voms/neiss.org.uk.

A. Voss, E. Vander Meer, and D. Fergusson. *Research in a Connected World*. 2009.

Wikipedia. Globally Unique Identifier (GUID), 2003. URL http://en.wikipedia.org/wiki/Globally_unique_identifier.

Wikipedia. Wikipedia Rutherford Appleton Laboratory (RAL) Article, 2009. URL http://en.wikipedia.org/wiki/Rutherford_Appleton_Laboratory.

Wikipedia. Wikipedia HTML article, 2011. URL http://en.wikipedia.org/wiki/HTML.

B.M. Wu, M.H. Birkin, and P.H. Rees. A spatial microsimulation model with student agents. *Computers, Environment and Urban Systems*, 32:440–453, 2008.

.

List of Figures

1 The General Simulation Workflow 17
2 The Genesis Simulator Simulation Workflow 18
3 Initialised Population in Single Years of Age for 1991 19
4 Population Theoretically Estimated Living at the start of 1992 . 20
5 Population Theoretically Estimated Dead in 1991 22
6 Population Simulated Dead in 1991 23
7 Population Simulated Living at the start of 2001 24

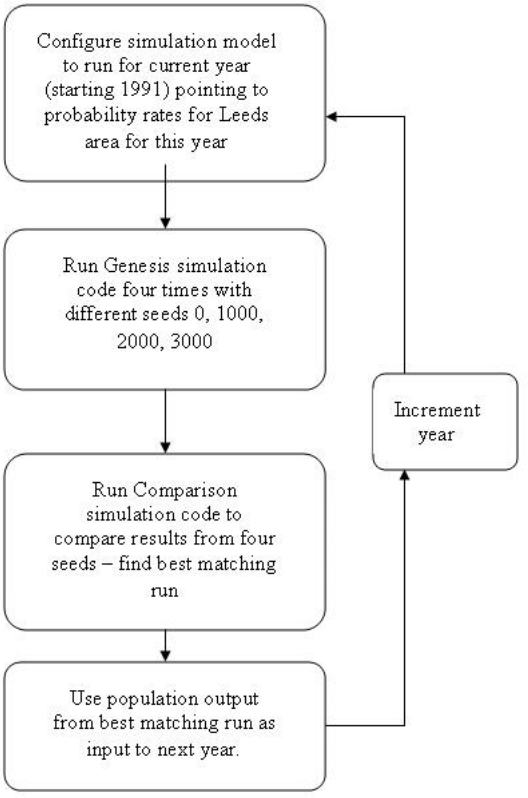


Figure 1. The General Simulation Workflow

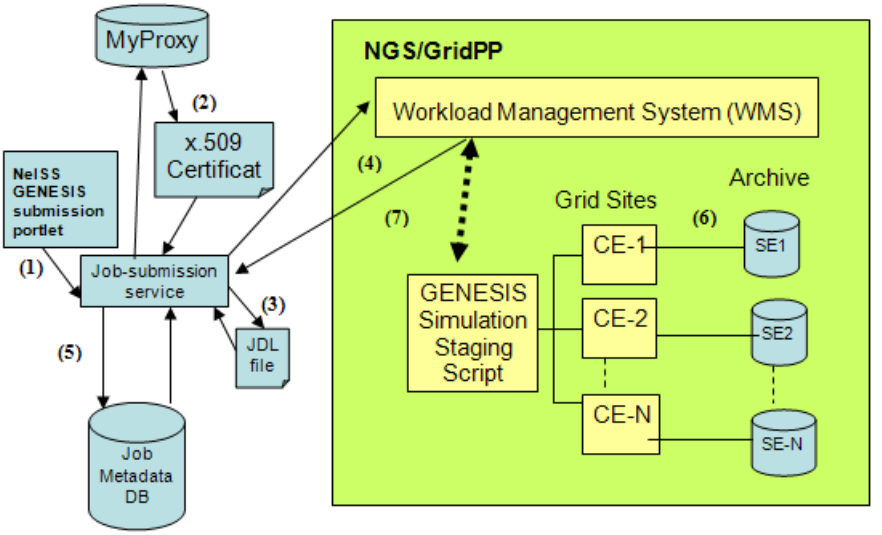


Figure 2. The Genesis Simulator Simulation Workflow

(1) After logging in and authenticating using shibboleth (Internet2, 2012) the user initiates job submission via the portlet and the portlet then invokes the job-submission service

(2) Job submission service pulls user's proxy from the MyProxy service (NCSA, 2000) and creates a VOMS proxy

(3) Job-submission service creates JDL file drawing in user input provided via the portlet interface

(4) The jLite (Sukhoroslov, 2009) API is used for Java representation of glite-WMS commands: job submitted to the WMS

(5) The Job ID and associated Job metadata are stored in the Job metadata database using JDBC API (Fisher et al., 2003)

(6) Model output saved in archive using WLCG (CERN, 2010) tools and registered in LFC (GridPP, 2010b)

(7) Model result metadata saved as output from worker node and passed back to portlet. Including GUID associated with population file (saved in archive) so that this file can be used as input to future jobs.

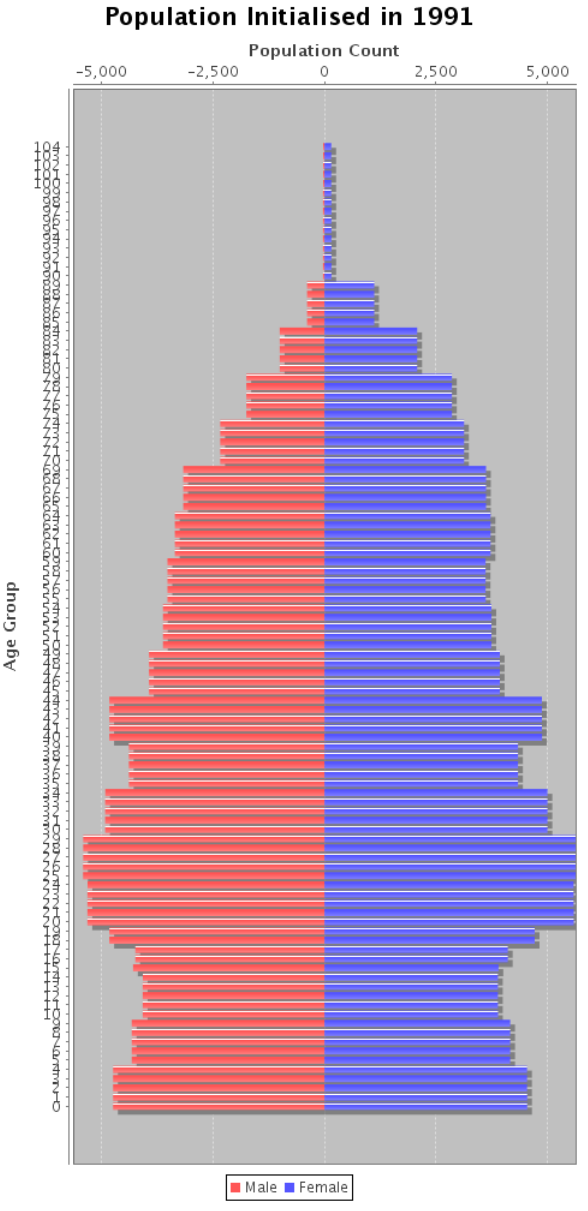


Figure 3. Initialised Population in Single Years of Age for 1991

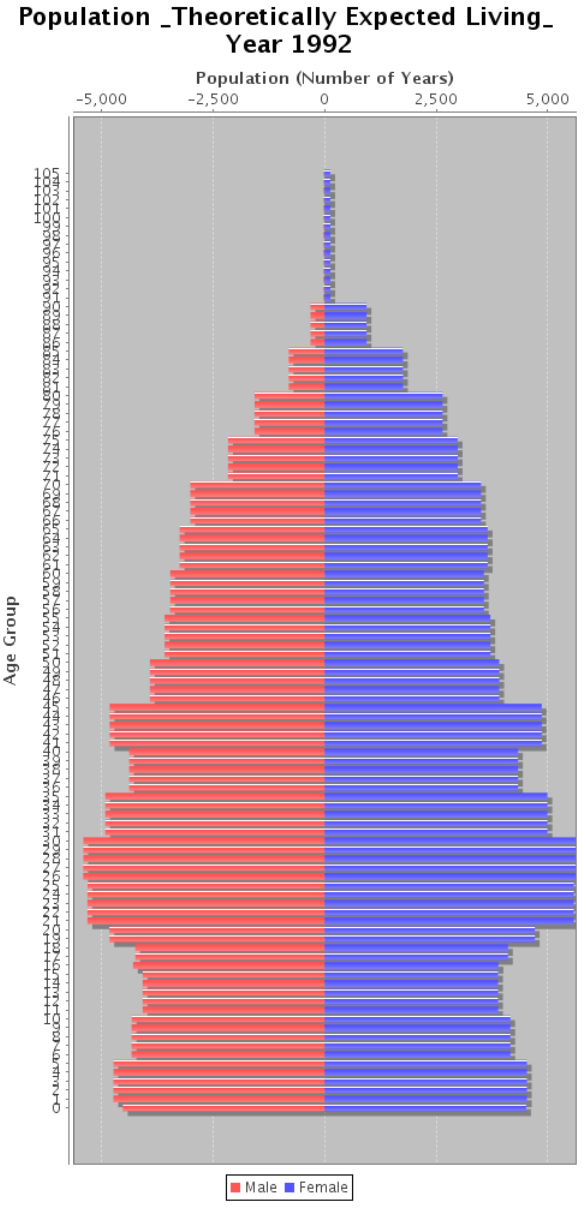
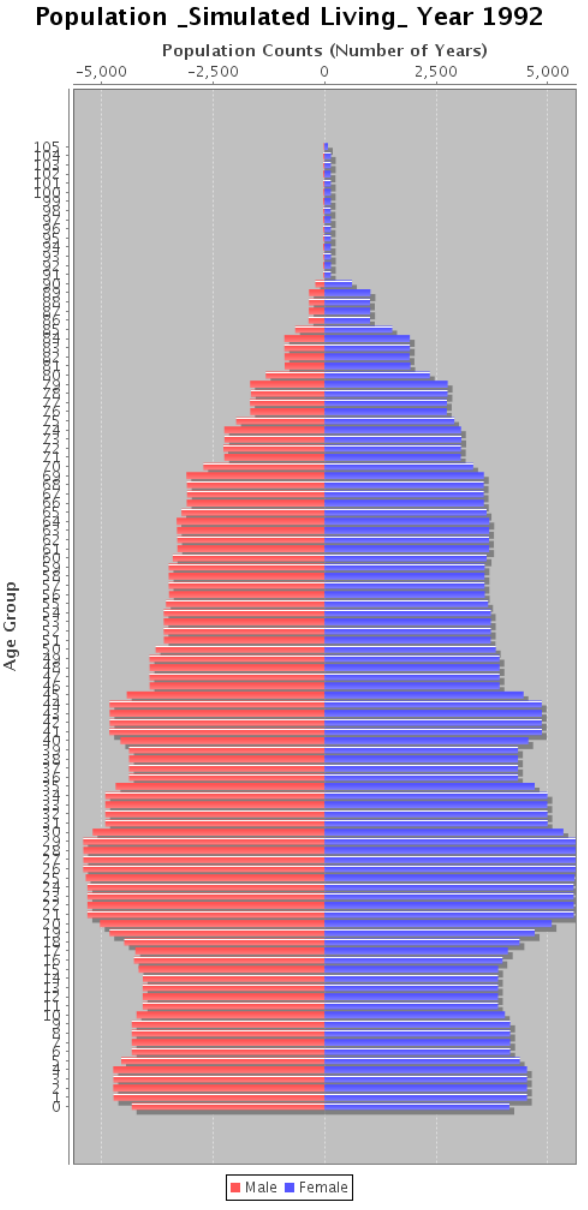


Figure 4. Population Theoretically Estimated Living at the start of 1992



aptionPopulation

Simulated Living at the start of 1992

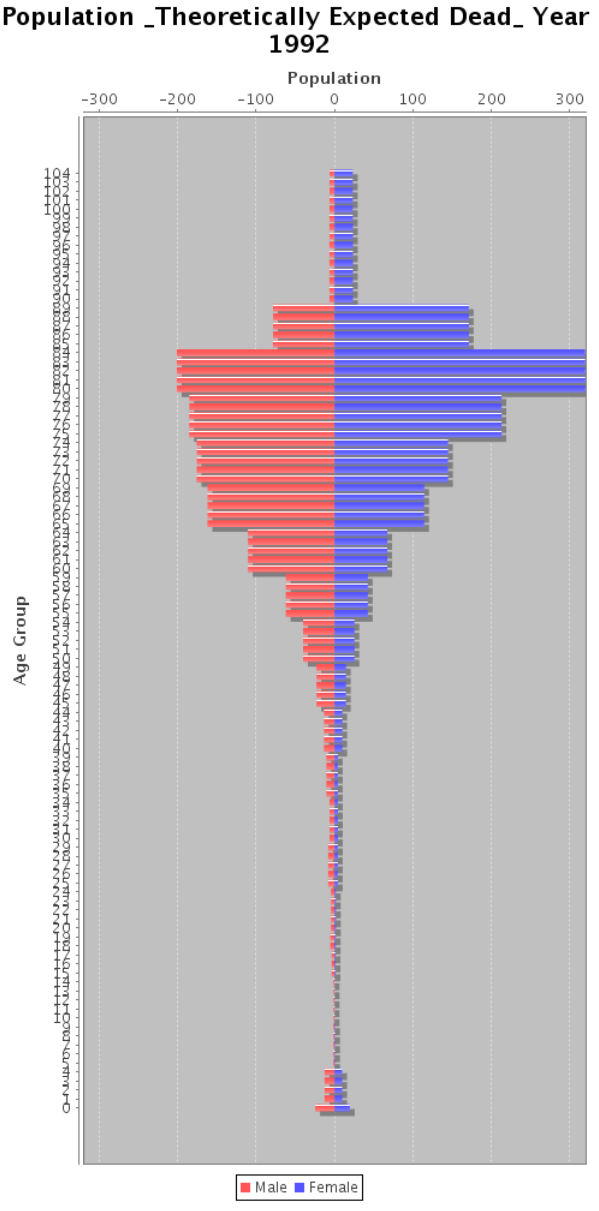


Figure 5. Population Theoretically Estimated Dead in 1991

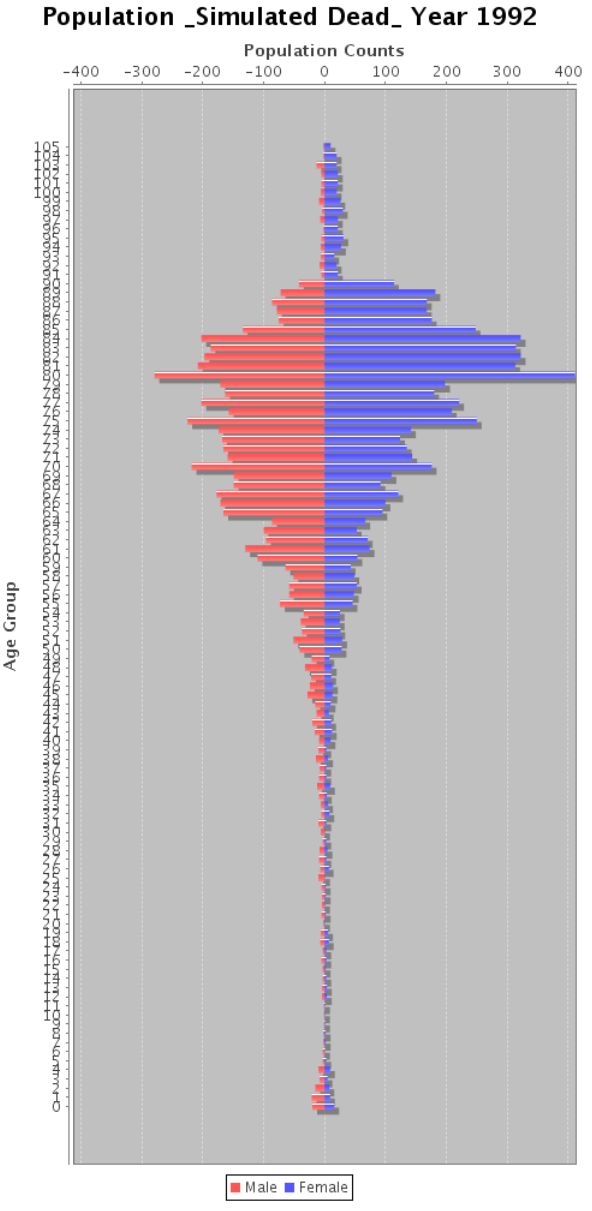


Figure 6. Population Simulated Dead in 1991

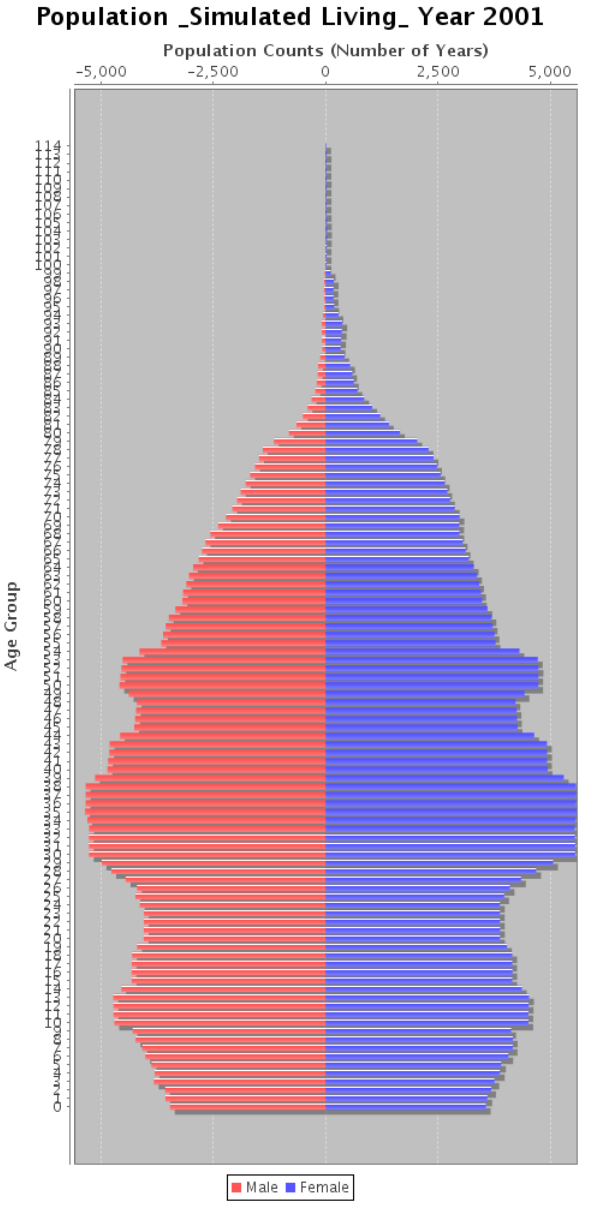


Figure 7. Population Simulated Living at the start of 2001