

Chapter 6

Integrating Agent-based Models and Geographical Information Systems for Output Visualization

6.1 Introduction	144
6.2 Using Bitmaps for Static Data Visualisation.....	144
6.3 Using Geographical Information Systems (GISs) for Modelling.....	146
6.3.1 Geographical Information Systems (GIS)	146
6.3.2 Representing Agents with Geospatial Objects and Geospatial Layers.....	148
6.3.3 Using GIS with Agent-Based Simulations	149
6.3.4 Representing the Temporal Aspect of Agent Behaviour in a GIS.....	153
6.3.5 Patterns of ABM Execution.....	154
6.4 Implementing a GIS Framework for the OWSWF Agent-based Model	155
6.4.1 Agent Information	155
6.4.2 Implementing an ABM-GIS platform	156
6.5 Using Web-Based Mapping Services for Analysis	163
6.6 Model Parameters and Their Sensitivity.....	165
6.6.1 Simulation Scenario Interface.....	165
6.6.2 Parameters and Scenarios	166
6.7 Concluding Remarks	177

6.1 Introduction

This chapter explores the approaches and challenges related to the display and analysis of the spatial output produced by the agent-based OWSWF model developed over the course of the previous chapters. The challenges related to spatial data visualisation and the approaches used to address them are not unique to agent-based modelling; however, this chapter focuses on the use of visualisation and geographical information systems with the OWSWF model. Initially, this chapter reviews the approach used to produce the high-resolution bitmap images from the simulation results (already seen in chapter five). The use of these images has several limitations, such as the lack of analytical capabilities and no provisions to combine this information with other sources. Then, I explore the use and integration of Geographical Information Systems with agent-based models for data visualisation and analysis of both input and output datasets. Different integration approaches are explored and, following on from the introduction of GIS concepts, a proof-of-concept implementation has been developed using a customised plug-in written for the open source Quantum GIS that allows for the analysis of the OWSWF model's data. In addition to the proof-of-concept implementation, the use of web-based mapping services is explored with a demonstration of a basic implementation using the Satellite mapping capabilities provided by Google Maps for creating simple visual displays. This section adds to existing agent-based modelling work by developing the ability to combine complex, large scale agent-based models with GIS tools for complex spatial analyses.

The final section in this chapter explores the use of the agent-based OWSWF model for sensitivity analysis of a selection of input parameters. This section introduces a simple user interface, which allows parameters to be easily specified and provides a brief comparison/analysis of the sensitivity scenarios against the base-case scenarios initially run in chapter five.

6.2 Using Bitmaps for Static Data Visualisation

The agent-based OWSWF model developed in chapter four and reviewed in chapter five used the cellular-automata approach to represent the spatial aspects of the OWSWF population distribution. In order to effectively represent the effects of agent's actions and outputs, this approach uses independent cells that do not actually represent agents within the system directly. Essentially, the emergent behaviour of the underlying agents (from the simulation) is mapped to a cellular automata structure. Each cell can be thought of as a conceptual container of the individual agents, and it is this cell structure that represents the spatial aspects of the agent's behaviour (in the case of the OWSWF model these cells are also used to partition processing within the model). Within each of these cells, the distribution and spatial representation of the agents along with the spatially indexed input data are considered to be homogeneous. As a result, the resolution chosen is critical for accurate representation. Figure 6.1 outlines this architecture.

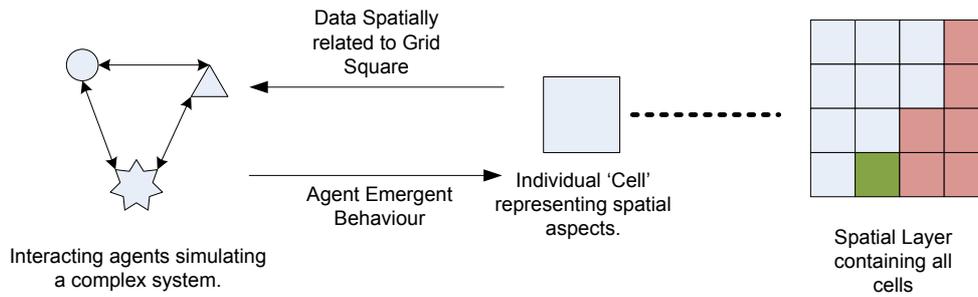


Figure 6.1 An indirect representation of agent’s behavior by spatial objects. The effects of the agents are represented by cell objects

Within chapter 5, this concept was implemented using static bitmap images. In this implementation, the entire cell-based structure is mapped to a single 4,060 x 2,880 resolution image. Each pixel in the output image represents a single cell within the simulation. The resulting population distribution maps show the population of female OWSWF within each cell of the simulation. This population, which is simply calculated from the insect agents that are in the adult life stages, is used to calculate the ‘value’ for the corresponding pixel in the output image. The pixel value (which is actually a combination of the three values – red, green and blue) is calculated using the *Hue-Saturation-Value* (HSV) colour model [91]. In this approach, the OWSWF population value for the corresponding grid square is normalised to lie between 0 and 260 using the min-max normalisation, inverted, and used as the Hue value. The Saturation and Value components are set to 1, and using the HSV colour model the red, green and blue pixel colour component are calculated [91]. The result of this process is a mapping of the value to a colour that lies between royal blue, representing 0 adult OWSWF, and bright red, representing the maximum number of OWSWF in accordance with the legend shown in figure 6.2.



Figure 6.2 The resulting colour for the female OWSWF population maps

These heat maps represent the population of the OWSWF across the simulated area at a particular point in time. In order to streamline the implementation, a base image (shown in figure 6.3.) is used that specifies the ‘Land’ grid-squares based upon the grid structure outlined in chapter four. This base image is used as a starting point so that only grid-squares that have an OWSWF population require a pixel RGB value to be calculated, making the process efficient and preserving the ‘Ocean’ grid squares. Examples of the resulting heat maps can be seen in section 5.4.3 where heat maps are used to present the output for the agent-based OWSWF model.



Figure 6.3 The base image used to produce the OWSWF population distribution maps. The ‘land’ pixels are shown in green

These bitmap-based heat maps provide a valuable tool to visualise the population distribution and allow for basic analysis of the model’s output. This approach, however, has several drawbacks that limit its effectiveness as an analysis tool:

1. The data presented in this way cannot be easily combined with other spatially referenced data, such as government or property borders and other relevant spatial information that can aid decision support tasks.
2. There are no facilities for querying, sorting or retrieving the information presented based upon the spatial attributes. This strictly limits the usefulness of the information presented.
3. In the approach used in this work, the bitmap images are produced in a one-off process and must be re-generated if the underlying spatial information changes. In most situations this is not a big issue, however if new information needs to be presented in the bit-map it must be re-generated.
4. Once generated, the bit-map images occupy storage space in addition to the storage space required for the *actual* results information. Although this is not a big issue due to relatively cheap storage space, the approach is inefficient.

These shortcomings aside, the heat map images provide a good way quickly present basic sets of results that summarise the trends evident within to policy makers and scientists.

6.3 Using Geographical Information Systems (GISs) for Modelling

6.3.1 Geographical Information Systems (GIS)

As mentioned in chapter one, *Geographical Information Systems* (GIS) are software systems that provide the tools to organise and analyse spatial data. GIS provides researchers with the tools to analyse the relationships between representations of real-world entities based upon their spatial properties. GIS refers to a software methodology rather than a specific system that provide facilities to create graphical presentations and graphical interfaces for analysis. These systems have been applied to virtually every field of study that involves the use of spatial information. Some areas that are well covered in literature include urban land

management, water and catchment management, animal or plant pest threat analysis, and forestry and logging [4, 92-94]. One of the great strengths of GIS is the ability to combine information from different sources for analysis and presentation. In a typical GIS, the different data sources are organised into sets of conceptual layers within the GIS. This organisation allows the users to combine different layers, to generate presentations for analysis and examine the relationships between objects from different sources. An example application of this ability might involve the analysis of satellite weather data and pest spread data to determine relationships between the two. Enterprise GIS software, such as ESRI ArcGIS [85], and open source systems including QGIS [86] and Saga GIS [95] all provide graphical tools for constructing spatial (and non-spatial) queries for analysing the relationships within data. The use of these off-the-shelf packages has several practical advantages over the construction of specially designed GIS software for use with modelling systems. These include being able to take advantage of existing expertise, training and support, and the savings in development and testing time.

GISs typically present two different types of data within their conceptual layers [96]. *Raster data* represents a continuous field view that represents changes over a given area using a uniform organised grid of cells. A raster data layer can be thought of as a spatially referenced image, where each pixel contains data items that represent a category, class or continuous measurement for a specific geographic area. Each pixel in a raster data layer can contain multiple data items that relate to the concept that the layer represents. A common example of raster data are maps that show the elevation over a given area. The second form of data presented by GISs is *Vector* data. Vector data is used to represent specific objects within a conceptual space. Typical GISs allow geo-spatial vector data types such as *points* that represent entities which have a single point of reference, *polygons* for representing entities that cover a defined area, and poly-lines (or multi-point lines) that denote linear features like roads and rivers to be specified, saved and presented on a vector layer. Some systems extend this basic set of types to include other more complex structures. An example of this can be seen in the PostGIS [97] database extension for PostgreSQL [98] with its multi-polygon data type, which effectively represents a collection of polygons and composite geo-data structures in ArcGIS [85]. The use of the data types (raster or vector) and the representations within them largely depends upon the concept to be modelled. For instance, linear features such as roads and boundaries are very difficult to represent accurately using raster type data due to the cellular nature of the raster data. Points of reference are similarly difficult to represent using raster-type data because the smallest that can be represented is the pixel. Vector type data is better suited for representing points of reference and linear discrete entities. On the other hand, vector data-type is not efficient in representing continuous field views of data exhibiting high levels of spatial variability such as weather or vegetation maps. In addition, raster data is more efficient for performing buffering and layering. As a result, it is common to combine layers of different data types in order to take advantage of both types of data when representing a phenomenon [96].

Many enterprise GIS software packages provide scripting and programming capabilities that can allow the development of custom mapping applications within the GIS software [85, 86, 99]. This capability has drawn

much attention in scientific literature in recent years as it allows the development of modelling software within the GIS, without the needing to use specialised programming tools.

6.3.2 Representing Agents with Geospatial Objects and Geospatial Layers

In order to represent information of an agent-based model within a GIS (whether tightly or loosely coupled), a mapping that defines how each type of agents is represented within the GIS will need to be developed.

Brown *et al.* 2004[100] touches on this topic by broadly outlining two ways in which agents and their interactions can be represented within a GIS, namely:

1. An agent can be directly represented by a geospatial object such as a point or a polygon.
2. The simulation can define interactions that agents (or collections of agents) can have on existing geospatial objects.

Agents within an ABM contain a set of data items for individual states that are updated and modified during the course of the simulation. The complexity of these items depends upon the conceptual representation and implementation within the specific model. In order to develop a truly versatile decision support tool, individual agent's data items or combinations of these items, including specific interactions between them, can be mapped to geospatial objects. To illustrate this concept, a simple example would be an agent that represents a farm in a simulation that models the spread of a contagious disease within livestock. This agent may have state information that indicates whether or not the farm is known by the authorities to have infected livestock and other information that outlines the number of livestock infected by the disease at different states. In order to represent this information in a usable decision support tool, these pieces of information could be mapped to different objects within the GIS. The information regarding the known state of the farm would best be represented by a colour coded point object. The number of infected animals on the farm could be represented by using a colour coded polygon, where the colour corresponds to the number of infected animals and the shape of the polygon corresponds to the boundaries of the farm. This would effectively represent the number of infected animals by the area within the polygon. Figure 6.4 shows an example of this scheme. By breaking down and representing the agent's data in this manner, the simulation can provide decision support for different users at different levels. In our example above, policy makers may only be concerned with the known status of farms whereas veterinarians may wish to analyse the numbers of animals affected on different farms for the sake of allocation of resources. These two different types of users can make use of the same modelling system for different decision support tasks.

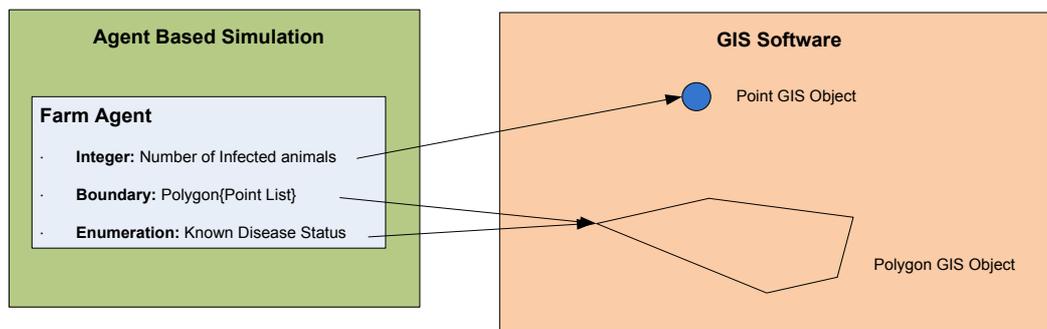


Figure 6.4 The mapping of the individual agent's data items to GIS objects within a GIS software

As mentioned in Section 6.3.1, GISs break the geospatial information down into layers of either of vector or raster data. In most situations, each layer contains data representing one aspect of the phenomena being studied. The idea of this organisation is that it allows analysts to break the spatial representations of a system down and analyse relationships between the components. In order to maintain this organisation within the GIS when used for agent-based modelling, the information mapped from different agent's data items, or combinations thereof, can be mapped to a different data layer. Carrying on the example simulation developed above, the coloured point objects that represent the perceived state of the farms could be mapped to a vector layer while the polygons representing the number of infected animals could be rendered on a separate vector layer. In practical terms, this organisation allows users to choose which layers will be displayed for analysis, enabling the generation of visual displays for combinations of information required for specific decision support roles.

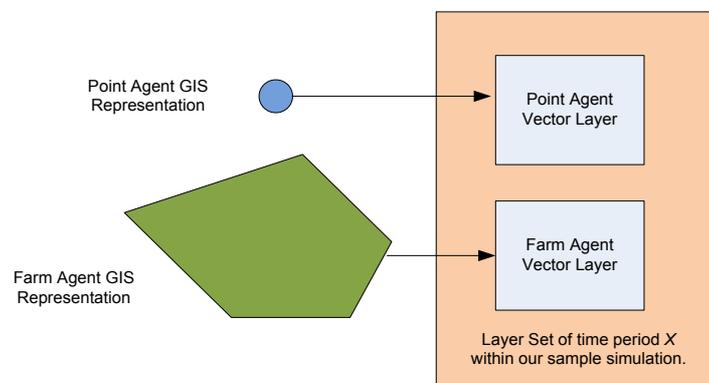


Figure 6.5 The mapping of agent representation objects to discrete spatial layers.

6.3.3 Using GIS with Agent-Based Simulations

Scientific literature regarding the coupling of GIS and simulation software (agent-based or not) has broadly focused on two types of architectures:

- *Tightly coupled* systems - where all simulation processing, geographical processing and presentation are carried out within one single physical module. This module could be constructed around a general purpose programming language or within a GIS using a scripting environment.

- *Loosely coupled* systems - where an external programming environment is used to implement the simulation logic (or part thereof) that operates alongside a GIS for geospatial processing and presentation.

Brown *et al.* [100] provides a comprehensive overview of methods for coupling agent-based models implemented in general purpose environments with GIS systems. It covers both the conceptual representations of different aspects of ABM and proposes a number of practical architectures for coupling ABMs to GIS. Furthermore, the paper defines a set of conceptual relationships between spatial objects and agents that capture the conceptual aspects of a model: *Identity* relationships, *Casual* relationships, *Temporal* relationships and *Topological* relationships. Identity relationships map agents directly to geospatial objects that are present within the GIS. For example, a vehicle might be mapped to a point type object. This provides an intuitive way to represent the agent's state within the GIS. Casual relationships capture the conceptual representation of the agent's effects by defining GIS objects that the agent can influence. For instance, a lake represented within the GIS may be influenced by the actions of agents in the simulation. This allows for an extension to the data and concepts represented within the ABM. Temporal relationships capture the time-sensitive nature of ABMs. The topological relationships define the movement of the geospatial objects in order to represent the actions of the agents. Topological relationships can also be defined directly between the geospatial objects without the linkage to agents.

The research in [100] proposes the use of a *middleware* software layer that sits between the GIS and the ABM for implementing the relationships between them. The proposed middleware layer handles the identity, casual and temporal relationships by providing an interface for the ABM to query the GIS, make changes within the GIS and extract information from it. The casual relationships are implemented within the GIS itself. This arrangement reflects the direction of information transfer between the GIS and the ABM. For example, identity relationships dictate that information flow is from the ABM to the GIS, whereas casual relationships require information passage in both directions while topological relationships require no input from the ABM. The work in Brown et al. 2004[100] does not include a detailed implementation of this middleware approach.

In Zerger and Wealands[93] the authors developed a hydro-logical flood risk management model. This model is a mathematical (i.e. not agent-based) model, however it provides good insight into loosely coupling methods for simulation systems. The model predicts the possible inundation of coastal regions in and around Cairns, Australia. In order to deal with the large quantity of information and provide an enterprise standard analysis tool for managers, the model integrates a Microsoft Access Database with the ArcGIS software and a hydrodynamic modelling system. The database stores the data related to a set of geospatial objects that represent conceptual entities including buildings, roads and other relevant geographical features that decision makers may require. The hydrological model is run to generate a set of results, from which water levels for each of the geospatial objects is calculated at time steps across the simulation time span. Once the data has been populated, a customised ArcGIS interface is used to access the populated database via an ODBC

connector. This interface incorporates the GIS functionalities that policy makers require to perform analysis. This system differs from other such as the National Model for Emerging Livestock Disease Threats (NMELDT) [4], as it does not display complete results to the user in real-time (i.e. the model completes its execution and results are then make available for analysis). The interface allows each geospatial feature to be examined at each time step set to generate detailed geospatial output that captures the temporal aspects of the simulation. The model can be run with different hydrological parameters to obtain results for different meteorological scenarios, giving decision makers a rich set of possibilities for decision support.

A common approach seen amongst geospatial modelling systems is the use of GIS libraries that are available for general purpose programming languages. From an implementation perspective, this type of systems is tightly coupled because the operations are coded within the same programming environment. The National Model for Emerging Livestock Disease Threats (NMELDT), outlined in [101], is one example. It simulates the spread of important livestock diseases, such as foot and mouth disease (FMD), on a national scale by taking into account regional and seasonal factors, different species and production sectors, and marketing systems. The simulation uses a range of inputs such as geographic data, climatic data, disease life-cycle parameters, and livestock movement data through the National Livestock Identification System (NLIS). The NLIS is Australia's system for identifying and tracking beef and dairy cattle [102]. The model makes use of the OpenMap GIS Java package [74]. This package provides an array of map display widgets, tools and functions for handling, analysing and displaying geospatial information within Java applications. The OpenMap package is largely used for displaying and analysis of results within the NMELDT in the form of point-type geospatial objects. The system uses the OpenMap display widgets to show the geospatial output of the model which is updated to reflect the state of the model after each time step as the model executes. This is analogous to watching a live television show. Currently, the NMELDT does not provide the facilities for analysis of results after the model has completed its run. Figure 6.6 below shows the interface during a simulation.

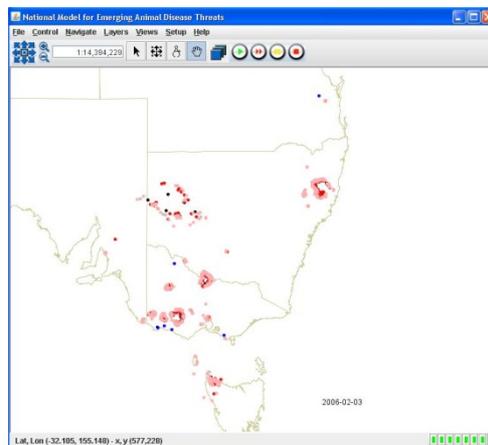


Figure 6.6 The interface of NMELDT. This interface makes of the OpenMap Java library to provide GIS display capabilities, such as layering and object display. (Source: [101])

The Repast modelling package developed in [32] offers a general purpose toolkit for creating agent-based

simulations. Repast is implemented in Java and provides a set of classes for programmers to run, display and collect data from agent-based models. Repast makes use of the OpenMap software package for geospatial modelling. The simulations are Swarm-type [29] discrete-event simulations. Events associated with agents are scheduled to occur in a specific order. Repast extends this by allowing the model to dynamically schedule events as it runs. This is achieved by allowing an agent's event to schedule other events to happen in the future, which makes possible more complex simulation. The example given in [32] proposes a model of cargo within an airport where a 'flight arriving' event schedules a 'baggage unload' event to occur. The geospatial representations of the agents involved in this simulation are displayed using the OpenMap GIS. A very good example of implementation of an agent-based model that uses the Repast-OpenMap approach is covered in [30]. This work develops an agent-based *Residential Segregation* model to simulate the urban dynamics in European cities. The model draws on some of the principles discussed in [100], and represents different classes of agents with relevant geospatial objects on their own layers within the model. The entire system is implemented using vector-type geospatial objects to represent agents and their behaviour. For example, residences are represented as static polygons and groups of individuals are represented by mobile points that move and change properties as the simulation progresses. The system also takes advantage of a general purpose programming language (Java) by including additional libraries such as the *Java Topology Suite* to efficiently carry out complex, time consuming processing tasks without the need to engineer custom functionality. This modular design demonstrates the power of coupling general purpose programming environments to GISs.

In addition to the integration of the OpenMap packages, Repast also can be integrated with the ArcGIS [85] software using the *Agent Analyst* [103] system. Agent Analyst is an extension for the ArcGIS software that allows agent-based simulations developed within the Repast ABM toolkit to be executed within the ArcGIS environment. The extension provides a full-featured graphical editor for the Repast model within ArcGIS, an interpreted language to manage the integration of the model to the GIS and a framework that tightly couples the ABM to the GIS within ArcGIS itself. In terms of comparing this system to others, it can be thought of as tightly coupled on the GIS side. In [103], the authors outline two example applications which demonstrate the system's viability for use with different types of ABMs. The work in [92] develops an ABM for simulating forest management using the Agent Analyst extension. The implementation demonstrated that the extension could be used to develop a versatile simulation system that leverages the advantages of general purpose programming environment, such as ready-made libraries and efficient processing, against the features and support of an enterprise GIS.

The work covered in [94] follows a non GIS-centred, tightly coupled approach using standard Visual C++ elements to create a geospatially referenced, cellular-automata model to simulate urban population dynamics on a city scale. The model uses raster-type geospatial data presented within a standard image widget on the interface. The geospatial processing is carried out within a C++ program to calculate updates across time intervals. This approach can be thought of as a custom built raster GIS within the general purpose C++ environment. The main disadvantages of this approach are the limited analytical functionality and the

processing efficiency of the geospatial processing. The simulation can be used to model a range of scenarios and produce detailed graphical raster representations for each time step. Due to the nature of the raster data, the analytical features of the system and its modelling capabilities are limited.

6.3.4 Representing the Temporal Aspect of Agent Behaviour in a GIS

One of the key challenges in applying GIS to simulation (not only for ABMs) is the representation of the temporal dimension captured within this type of simulation. Changes over time are a key aspect, virtually for every form of ecological modelling. In order for GIS and the agent-based modelling approach to be successfully applied to this domain, the temporal aspect needs to be successfully captured. In their most common form, agent-based models follow a discrete event, dynamically changing paradigm, where the states of the individual agents are updated through either synchronous or asynchronous update operations. GISs are designed to provide interfaces and functionality for representing and analysing the spatial information. As outlined in chapter 2, the spatial relationships presented within agent-based models are only one aspect of agent behaviour. In order for GIS to be coupled to these systems in a meaningful way, a method for representing the changes within the spatial information over time is required. Two general solutions for this problem can be seen within existing systems and the literature. The first solution simply involves updating one or more spatial layers within the GIS (whether is it loosely or tightly coupled) as the simulations progresses to reflect the state of the model at the current time. This method is analogous to a simulation executing like a live television show where results are displayed through the GIS as they are produced. This method is used in models such as NMELDT described in Welch et al.[101] and provides an effective display when the model runs. The main deficiency of this approach is that it limits the use of the GIS as an analytical tool within the simulation. This is because data for only a single day is displayed at a time and interaction is (by necessity) limited because it is changing constantly.

An improvement on this method is briefly outlined in [100] where a new layer, containing the geospatial representation of the current state, is saved for each time period of the simulation. An example of this approach might involve producing a new layer (or set of layers) containing the agent representations (GIS objects) for each day and saving this within the GIS so that it can be reviewed and analysed during and at the completion of the simulation. Each day's layer(s) is analogous to a single frame in an animation and this system would allow the simulation to be 'played' back in order to review the results of the simulation. This provides an effective way to capture the temporal aspects of the model and allows a user to apply the GIS's analytical tools to compare the set of layers at different time periods for decision making. An example application of this scheme would be a model that simulates the spread of an invasive pest within an environment. The set of layers for each time period of the simulation will capture the environment's state at each specific time period. The user can make use of the GIS's analytical tools to compare the sets of layers of different days in order to determine the overall rates of spread of the pest. These sets of layers, created for each cycle of the simulation, could be combined with static layers such as boundary, infrastructure data and vegetation maps for further analysis of the model's output.

6.3.5 Patterns of ABM Execution

Agent-based models generally follow a discrete event, time interval paradigm where a sequence of events occurred during each time interval to update the states of the agents that make up the simulation. In a typical agent-based simulation, the model executes while producing and displaying output that are either geospatial or not. Because of the stochastic nature and the emergent behaviour that they exhibit, agent-based models are usually run multiple times in order to obtain results that are statistically significant enough for analysis and use by policy makers. This requirement has fostered the need for ABMs to run in *Batch* mode where multiple runs of the model with the same parameter settings are carried out to gain the statistically significant data required. In the simplest form, this is based upon a *run-first, analyse-later* paradigm, where the simulation logic runs and produces a complete set of output data that is then available for analysis when the simulation is finished. In the batch mode, multiple runs of the simulation are completed in sequence and the results are analysed when all runs of the simulation are completed. An improvement on this approach involves the use of parallel processing technologies, including distributed computing and the use of GPU technology, to execute multiple runs of the simulation in parallel. Another form of operation within ABMs is *Interactive* mode. In this mode, the user is able to make changes to parameters during the simulation. A common implementation of this mode involves pausing the simulation after each time step, then allowing the user to view and alter values within the simulation. This ability can be a powerful decision support tool as it allows users to view the results of actions taken within a model as the events within the simulation unfold. An example of a real-world use of this mode might involve modelling the effects of the initial response to a disease outbreak in a population of animals or people. The simulation can model the epidemiology of the disease that is caused by the user modifying the parameters in the model directly via the interactions with the simulation.

Both the batch and the interactive modes of operation pose constraints upon the nature of the coupling with the GIS. Figure 6.7 illustrates the requirements for the flow of information in each mode of operation. In the Batch mode, when parallel processing techniques are used, no direct interaction with the simulation logic is required after it is initiated and all information is available at the completion of the run. This is due to the flow of information and the limited requirement for interaction, making this mode well suited to a loosely coupled approach. On the other hand, the interactive mode needs to allow the user to interact with the simulation output after the execution of each time step and for information to be passed to the simulation logic.

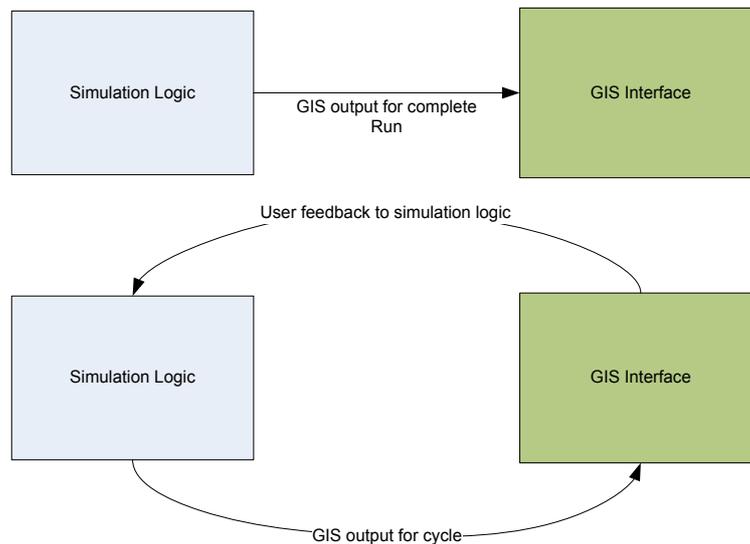


Figure 6.7 The transfer of information between the GIS and the simulation logic in Batch mode (top) and Interactive mode (bottom)

6.4 Implementing a GIS Framework for the OWSWF Agent-based Model

6.4.1 Agent Information

In this section, we will explore three different approaches for implementing the concepts outlined in the previous section with agent-based OWSWF model developed in chapter four. The approaches developed provide the ability to implement the conceptual relationships between the agents and spatial objects, as well as the temporal aspects of the OWSWF agent-based model. They also exploit architectural flexibility in terms of the structure of client and server machines and the specific GIS software used for the implementation. Throughout this section, a proof-of-concept prototype is developed to demonstrate how a GIS can be used to present and analyse data output from the OWSWF agent-based model. The GIS implementation uses a similar approach to the bit-map approach introduced in section 6.2, where a vector layer consisting of cells representing the grid-squares from the model is adopted. This approach allows each grid square to be represented within the GIS by a precise vector object that can be used in vector-based spatial queries both within a single output layer and across multiple spatial layers within the GIS. In terms of the concepts introduced in the previous sections, this amounts to what Brown *et al.* [100] describe as a *causal* relationship between the individual insect agents and the spatial object used to represent their spatial attributes. This approach is necessary due to the scale of the simulation (i.e. the number of insects) and the approach suits the architecture as the grid-square already provides a natural level of aggregation within the model's implementation.

The agent-based OWSWF model follows the run-first-analyse-later paradigm, outlined in the previous section. This means that a complete result set will be available for analysis within the GIS. The temporal aspect of the (aggregated) agent interactions are captured through the cyclic upload of data-sets at pre-defined intervals. Each of these data-sets represents a snap-shot at a specific point in time, which can be then be spatially analysed within the GIS. The grid-square level agent information presented within the vector

layer can either be individual pieces of aggregated information, such as counts of OWSWF agents, or combinations of this information. As outlined in chapter four, the output information for each grid-square that can be represented using the spatial layer of vector objects includes the following:

- Aggregated total insects for each of the 36 cohorts
- Total strikes on livestock
- Total available wounds on livestock

In addition to output information from the model, the spatially indexed input layers can easily be represented within the GIS. The process is achieved by linking each entry within the input data table with a spatial object at the database level. The use of the spatial database is outlined in the following section as it provides that basis for integrating the model with the GIS platform. The combination of the model output data layers, input data layers and external data layers along with the precise spatial representation using the vector polygon spatial objects provides a powerful platform for analysis using the GIS tools.

6.4.2 Implementing an ABM-GIS platform

Following on from the previous section which addressed the conceptual issues of applying GIS to the OWSWF model, this section outlines a *loosely coupled* implementation for integrating the agent-based model (with potentially thousands of agents) to an off-the-shelf GIS package. In the implementation, the Quantum GIS [86] software package is used. Quantum GIS is an open source GIS package that contains a full suite of analytical functions, typical of an industry standard GIS such as ESRI ArcGIS [85] and Saga GIS [95]. Critically, it also includes an interpreter for the Python scripting language which allows for the automation of data retrieval tasks and provides libraries for graphical user interfaces. These features allow users to develop complex customised scripts that are capable of loading, manipulating and analysing spatial data layers from a range of data sources. In addition to the scripting capability supplied by the Python interpreter (referred to as *PyQGIS*), Quantum GIS provides the ability to construct application extensions in the form of *Plug-ins* using the Python programming language. These extensions can include graphical user interfaces and can access the full set of functions provided for the analysis and display of geographical layers within the GIS. A built-in library of functions is available for use in the plug-ins and the scripts that provide access to geospatial databases hosted on sever machines. These custom plug-ins can be easily distributed and, once installed, these extensions are loaded automatically on start-up.

The approach chosen for coupling the model and the GIS avoids the use of the middleware option discussed in the previous section, and instead connects the general purpose GIS and the simulation logic through the use of a geospatial database system. This is implemented by making use of the scripting and plug-in capabilities to develop a customisation that connects the GIS to the database

system. The architecture for this approach is outlined in Figure 6.8. Here, we can see the core simulation module that contains the agent-based logic of the simulation that consists of the C++ implementation of the model using the NVidia CUDA GPU software. The simulation module feeds output from its set of interacting agents into a PostgreSQL database running the PostGIS geospatial extension. As mentioned in the previous section, the output data is uploaded to the database at a pre-defined interval within the logic time of the simulation, to capture the temporal aspect of the model. The default behaviour of the OWSWF model is to upload daily snapshots for each occupied grid-square to the database, providing a relatively high temporal resolution.

This database system acts as a passive middle layer that the off-the-shelf GIS software can interact with and pull geospatial agent data from. The simulation system has its own interface, constructed using Visual C# allowing the users to set parameters and control the execution of the simulation. This interface will be covered in a later section of this chapter as it is critical for supporting the use of the model as decision support tool. As the simulation executes, output is written to the database. When the simulation run is complete, the output data is made available within the database to Quantum GIS. Data is protected from being prematurely read from the database by using a flag that is set within the database against the meta-data for the simulation run.

Once the data is available to the GIS software, it is managed by a custom developed Python plug-in. This customisation provides that user with the ability to easily select the simulation run and the data from this run that is to be presented within the GIS. This is achieved by presenting the user with a user-friendly interface allowing them to select the information. Based upon the selections from the user, the plug-in generates the appropriate SQL queries and retrieves the data sets and uses the provided QGIS functions to present the output layers to the GIS interface. In this implementation, only the information for a single time period is presented within a single layer, however the plug-in can be used multiple times to generate multiple layers presented within the one GIS instance.

The user is able to analyse and compare these layers and import external layers to view alongside the simulation output. The GIS coupling with the ABM creates a powerful decision support tool. An example of this analytical capability can be seen when assessing the effect of the OWSWF on a human population with the sample modelling application. Layers containing information such as human population densities, government borders or locations of cities and landmarks can be loaded alongside simulation output and spatial queries can be carried out between these layers. The functions in the GIS provide for cross-layer analysis, allowing the users to calculate rates of spread and *buffering* queries by comparing data from separate layers. The organisation and retrieval of the ABM output is achieved using both client-side processing within the GIS's Python interpreter and

server-side database queries that return requested results to both plug-in and scripts. Accessing the PostGIS database using Quantum GIS is a relatively simple process because it has built-in functionality for accessing and loading layers directly from the database. In a distributed implementation, where the database is hosted on separate machine, processing is split between the client and host systems. In this approach, the actual analysis is fully carried out on the client machine, using the GIS capabilities. That is, the server side processing in the DBMS is only responsible for retrieving the initial data sets.

Figure 6.9 shows the user interface for the custom QGIS extension which allows the selection of various options. The interface presents the user with a three step process allowing them to select the simulation run, the date of the snap-shot to analyse and the data from this date in the simulation run to present. Because the output upload interval of results from the simulation is adjustable, the plugin does a check when the user selects the date to ensure that date is available within the simulation run for the specific date. The results of this check are displayed to the user. This plugin is still under development and additional options, such as the ability to load input data layers are still being implemented. For now, any additional layers on top of those indicated in figure 6.9 must be loaded to the GIS workspace manually.

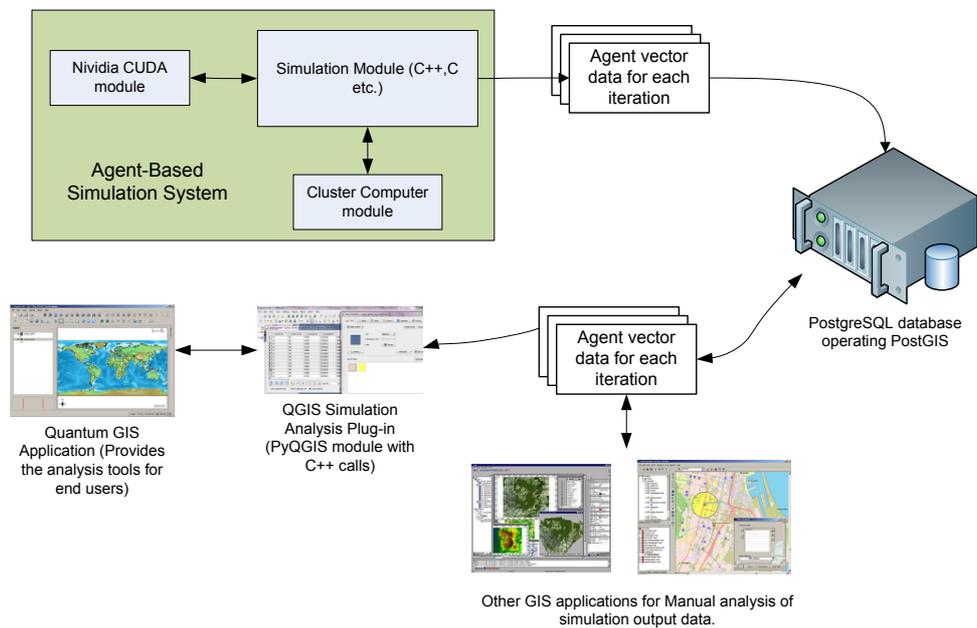


Figure 6.8 A loosely coupled approach for integrating GIS and agent-based simulation logic

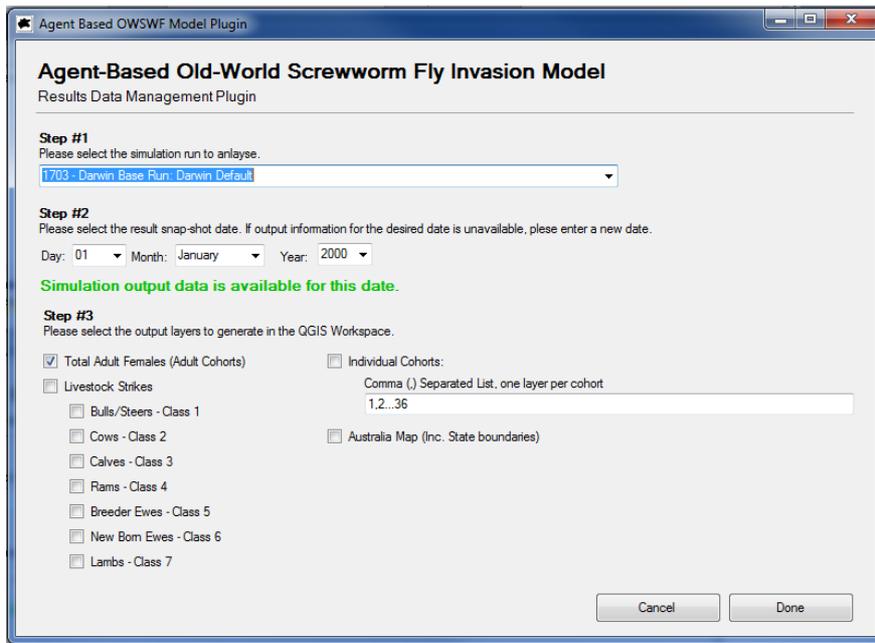


Figure 6.9 The Quantum GIS plugin developed to automatically load data layers to and from the PostgreSQL database. This interface presents a three step process to the user, allowing them to select the information to display

Using the Quantum GIS and the customized plugin, output layers from the agent-based OWSWF model can be automatically loaded into the GIS workspace for analysis and presentation. Figure 6.10 shows the output from a default scenario at 24 months presented within the GIS. The output layers, which are added using the plugin, are presented along with a vector layer containing a map of the Australian mainland (with state borders). This provides an example of the power of using of GIS for combining the simulation output with additional spatial data.

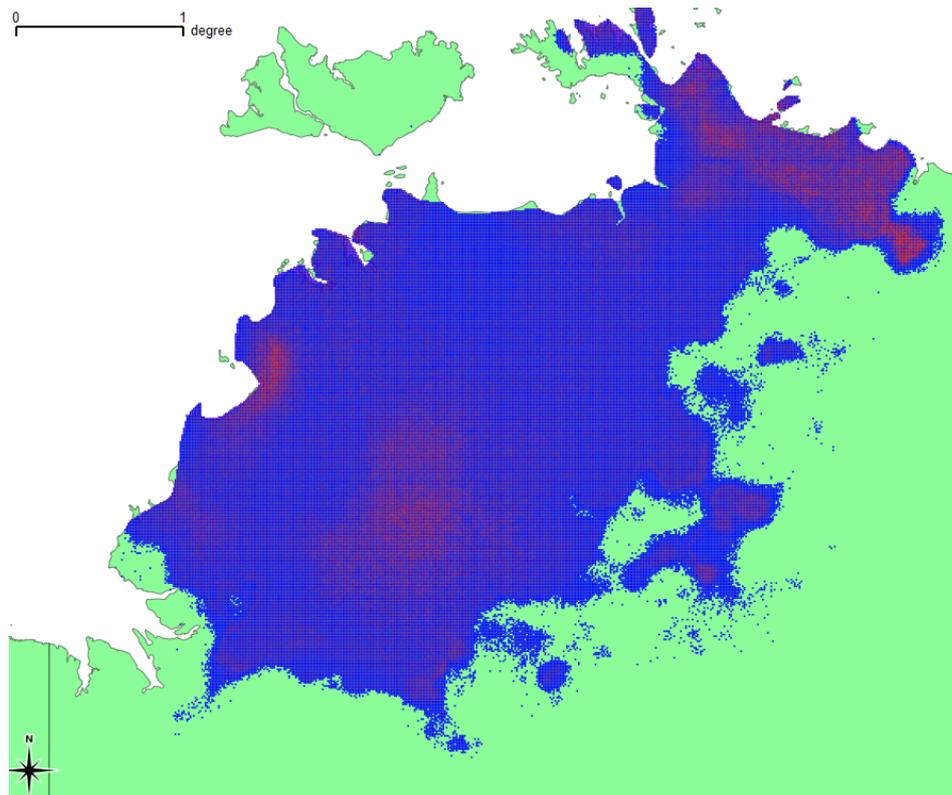


Figure 6.10 The use of Quantum GIS to present output from month number 24 in a default scenario starting in Darwin. The light green layer represents the Australian continent

This capability is further emphasized in figure 6.11, where the output from a default scenario, with the invasion starting in Sydney, is presented along with the cadastral farm boundary data for land holdings in NSW. Using spatial queries, it is possible to determine farms affected by the invasion for decision support tasks. The GIS software provides GUI facilities for constructing spatial queries. Figure 6.12 shows the spatial query constructed to find farm holdings affected by the insect invasion (pictured in 6.11). Figure 6.13 presents the affect farms with the model output layer (the affected farms are in yellow). The final figure in the section, figure 6.14 shows the affect farms (again, shown in yellow) without the model output displayed. Using the analysis tools, the agent-based behavior has been aggregated to the ‘farm’ level on the fly. These figures provide an insight into the power of spatial queries and its usefulness for decision support capabilities.

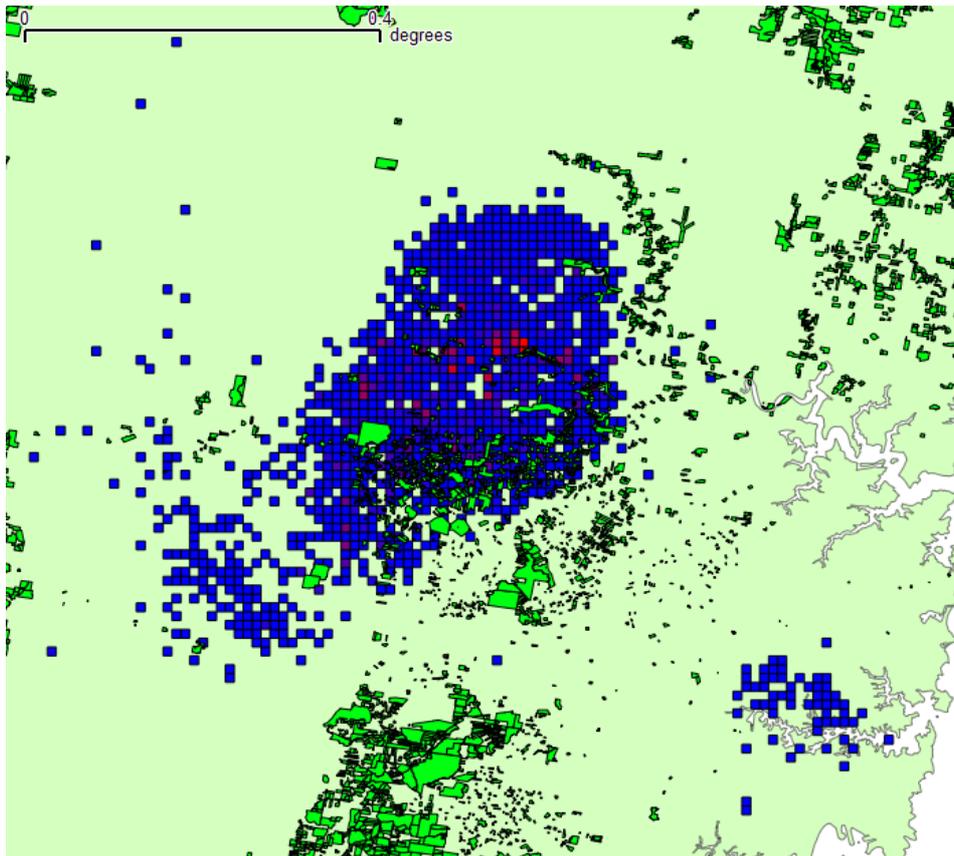


Figure 6.11 The simulation output displayed along with the cadastral farm boundary data layer

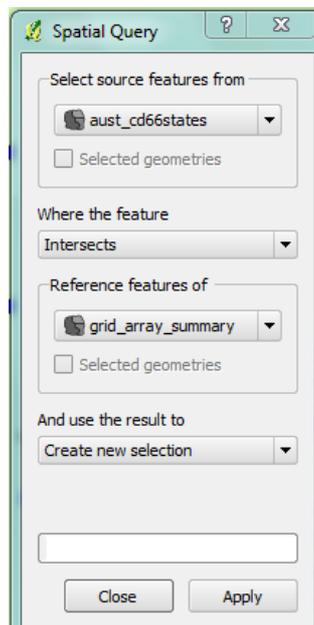


Figure 6.12 The spatial query used to determine the affected farms based upon the intersection of the farm boundary with the affected area predicted by the model

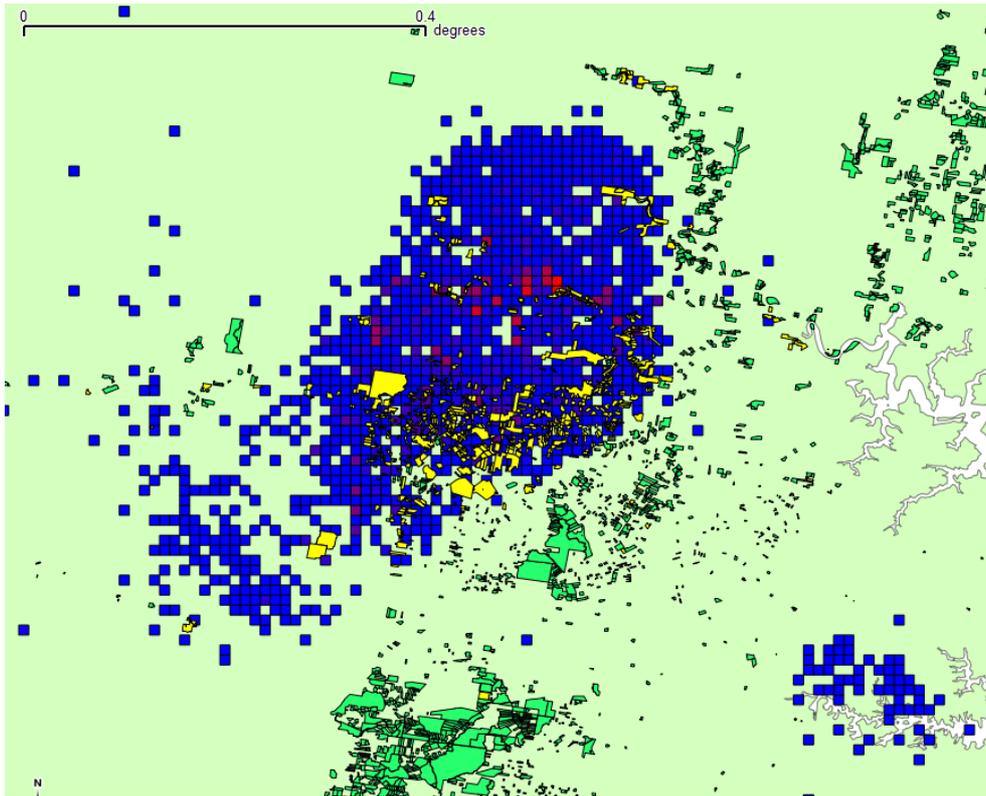


Figure 6.13 The simulation output displayed along with the cadastral farm boundary data layer with a spatial query to determine affected farms based upon the boundary information

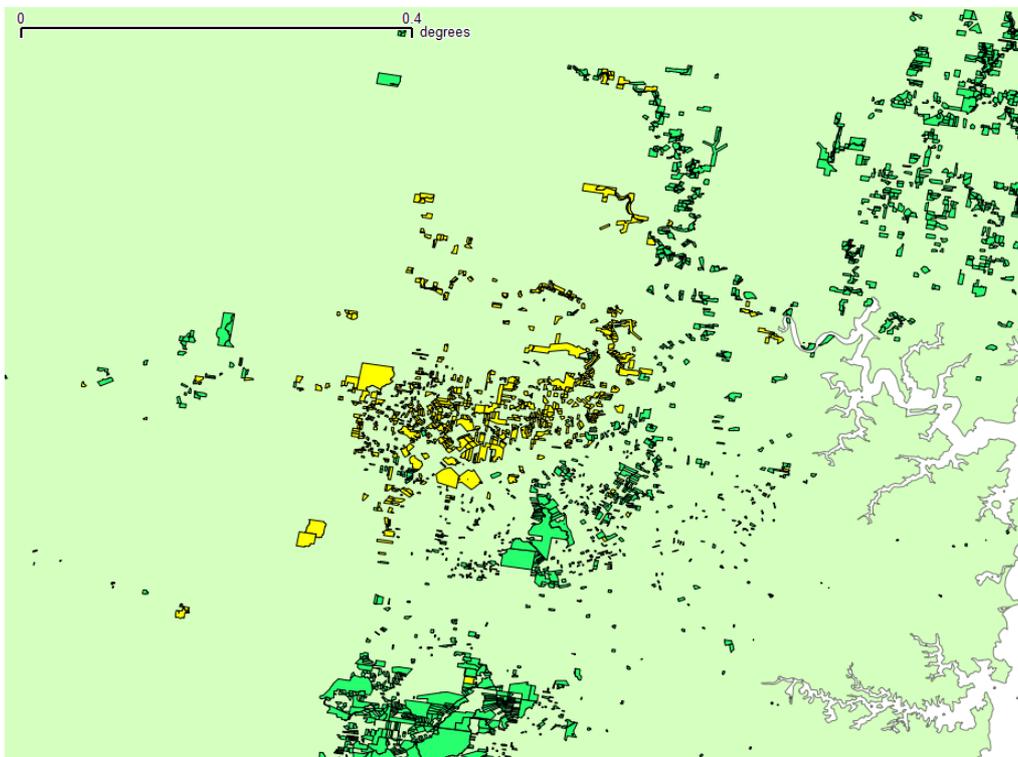


Figure 6.14 The affect farms, based upon the intersection of farm boundaries with the affected area predicted by the agent-based model

6.5 Using Web-Based Mapping Services for Analysis

In recent years, web-based mapping applications such as the Google and Bing mapping services have become an integral part of many sites, providing a range of mapping functionalities through a simple-to-use API. These applications have been applied to web sites and smart phone applications in a range of areas such as directory services, advertising and travel planning. As the usage of these applications increases, online mapping services have developed the functionality that makes them suitable for the mapping and analytical tasks related to agent-based simulation. By making use of standard web technologies, along with the web page objects and associated JavaScript API of a suitable mapping service, it is possible to create a loosely-coupled, web-based analysis and presentation interface that can operate on a client machine within a web browser for analysing model output data. The Google maps API is one example of a system that provides mapping and GIS services required for ABM analysis/presentation tasks. This API allows various mapping interfaces to be inserted directly into web-pages (as HTML objects) and manipulated by its suite of JavaScript functions. The API allows combinations of point markers, polygons and poly-lines, along with advanced iconography. The mapping service provides up-to-date layers of spatial data including street maps, terrain data and satellite imagery. Through the use of combinations of these display functions, this API provides the functionality to create complex mapping displays of ABM data. Figure 6.15 outlines a modular system for a web-based analysis tool for ABMs.

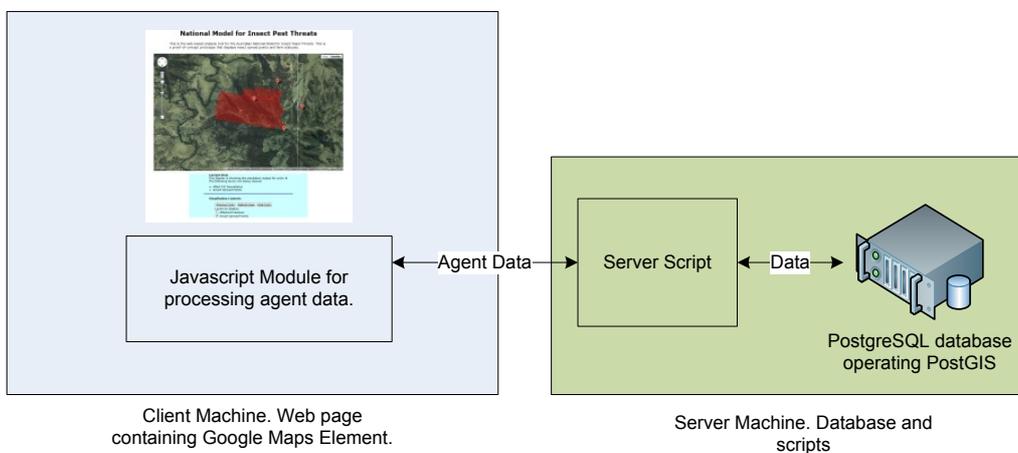


Figure 6.15 The architecture of a web-based coupling interface that makes use of the Google Maps online mapping service. A JavaScript module that is controlled by form elements on the client machine interacts with a server-side script to load and display agent data

In the prototype presented in Figure 6.15, the user is presented with a web-page that contains a Google mapping object and a series of controls for manipulating the data displayed. These controls are similar to those used in the QGIS plugin. These controls execute JavaScript functions that send *Asynchronous JavaScript and XML (AJAX)* [104] requests to a PHP script sitting on the web server. This script returns the requested data, which is then processed by the client side JavaScript on the HTML page and the mapping object is updated. This architecture allows for a distributed approach where the client and server are on physically separate machines or a single system approach where the web server and web page lie on the

same machine. This architecture is similar to the loosely-coupled system, where the simulation completes its run(s) and the results are then analysed using the GIS. As a result, this approach is suited to the batch and standard modes of operation. Due to the client-server architecture, this approach is not suitable for implementing the interactive mode of operation. The web interface is simply a tool to view stored data results that are output by a simulation run beforehand.

Using this web-based approach, all simulation data is loaded into the GIS by the client side customisation and all analytical processing carried out within the specialised software of the GIS. The key difference in the web-based approach is that the majority of the spatial analytical processing is actually done within the geospatial database system, not the client machine. Figure 6.16 shows a prototype implementation of the example agent-based simulation, showing spread points and boundaries of affected premises. In this example, functionality that is available in conventional GIS software is programmed into the logic of the page and a set of supporting server-side scripts. The user is able to use controls presented on the web page to select from a number of different views of the ABM input and output data, and then refine these views by different criteria such as by property (using a Property Identification Code), area (by specifying a containing polygon or state/region), or by agent type. These spatial views are managed using the AJAX approach described earlier, where the sever-side scripts access a spatial database system, which is responsible for the spatial queries and processing. Figure 6.17 illustrates the flow of data between the client system and the spatial DBMS.

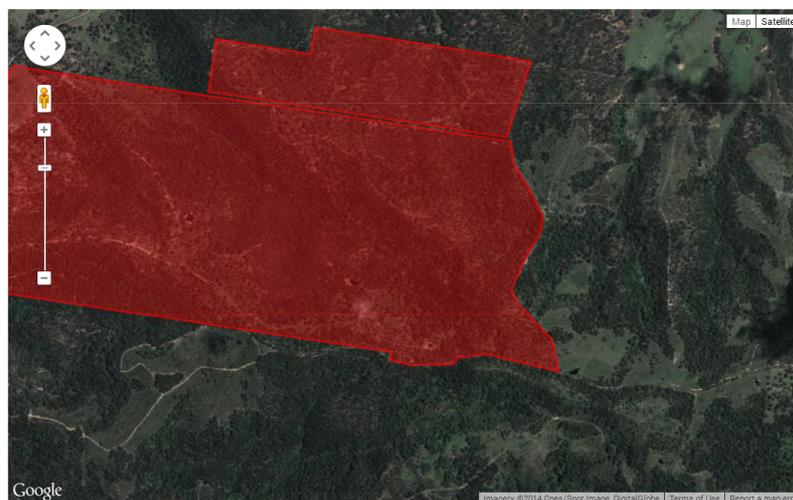


Figure 6.16 A prototype implementation of an insect spread experiment, showing the boundaries of the affected farms and points of insect spread. This implementation uses the Google Maps' JavaScript widget and makes use of its satellite imagery

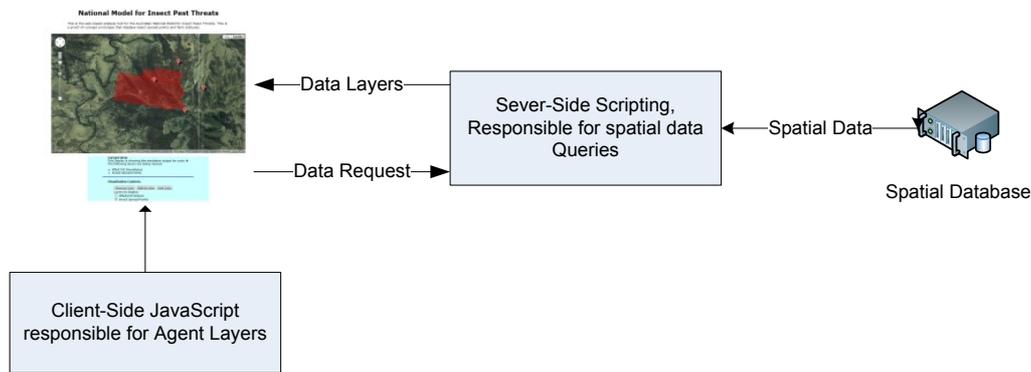


Figure 6.17 The data flow between the Spatial DBMS and the interface for the web-based analysis tool. The client side JavaScript is responsible for generating the display data within the Google Maps element on the page

One key limitation with the web-based approach is scale of the results that can be displayed through the web page. The Google Maps API is only capable of rendering a few dozen polygon objects on the map. In order to get around this issue, the grid-based output from the model is converted to a lower detailed super-polygon or other spatial information is plotted. For example, figure 6.16 show affected farms as opposed to actual grid-squares. Due to the limitation in both the analysis functionality and the performance of the mapping APIs, this web-based approach is suited more output presentation rather than analysis.

6.6 Model Parameters and Their Sensitivity

This section provides a study of a selection of key parameters that determine agent behaviour within the agent-based OWSWF simulation that was developed in chapter four. The aim of this section is to demonstrate the impact of altering the value of the various parameters within the simulation on the population distribution over time. This will provide an insight into the degree to which the selected factors and determine the overall spread. The data presented within this section was producing using a combination of the GIS functionality for spatial queries (for example for determining the affected areas) and the bit-map based approach for producing the population distribution maps for analysis of the results.

6.6.1 Simulation Scenario Interface

A simple interface has been constructed for the OWSWF model, developed in chapter four, to support experimentation with the extensive set of values within the lifecycle and dispersal algorithms that are outlined within chapter four. All values that are listed as constants within the lifecycle and dispersal equations have been parameterised so that they can be stored within the supporting PostgreSQL database within a *scenario* record that is loaded at run time, populating all of the parameters that are used through the simulation. This approach allows the users to create custom configurations of parameters (i.e. scenarios), save and run the scenarios as required. In addition to the parameter information for the dispersal and lifecycle simulations, information including the invasion start point's latitude and longitude is specified, along with the initial numbers of insects that initiate the OWSWF within each lifecycle stage, and the total start area of the invasion are also stored within a scenario. The scenario information is loaded and edited using the interface shown in figure 6.18. Within this interface, there are three tabs that contain the parameters

for the lifecycle simulation, dispersal simulation and general info respectively. Users are able to edit existing scenarios (with the exception of the ‘Default’ scenario) and create new scenarios for use within the model. The interface includes basic validation of inputs and easy selections using relevant interface controls (e.g. date-picker widgets for dates). The user interface was built using Microsoft Visual C# which connects to the PostgreSQL database using the Npsql [105] .Net data provider libraries. This library, which is based upon the libpq source used within the main simulation, provides the simple database functions required by the interface. Visual C# was chosen as the platform to develop the interface as it provided all of the GUI components needed (out of the box) to develop the interface for the Microsoft Windows environment.

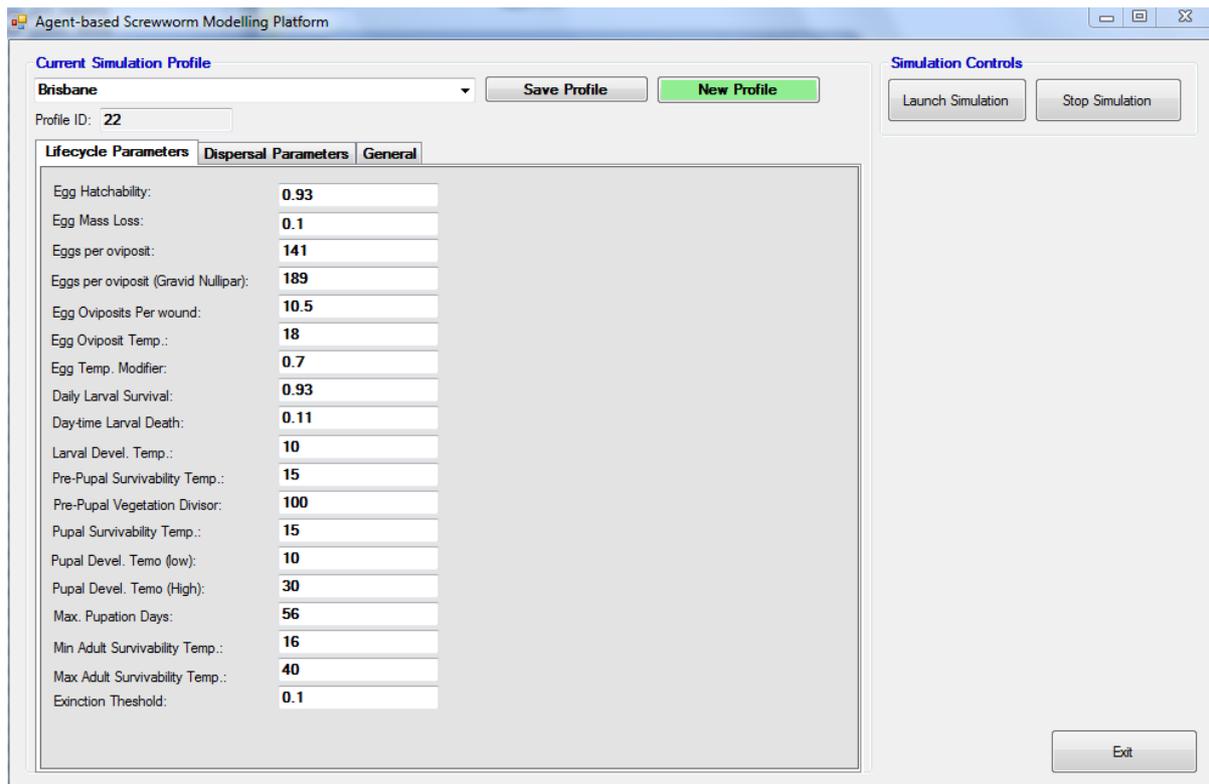


Figure 6.18 User interface for producing simulation scenarios. The three tabs contain the parameters for the lifecycle simulation, dispersal simulation and general characteristics of the model respectively

The user interface also provides the ability to launch and stop the simulation program. Once the user has edited/created a scenario using the interface, the simulation can be started using the currently selected scenario within the simulation. This is achieved by executing the main simulation process using an exec call from the simulation process, passing the scenario’s id value (which is allocated and managed by the database) through to the simulation process as a command-line argument. Upon execution, the simulation’s main process reads the id from the command line arguments and retrieves the scenario (i.e. all of the parameter values) from the database before executing the simulation.

6.6.2 Parameters and Scenarios

This section looks at a series of parameters and the effects on the rate of spread and population distribution that these parameters have on the overall results of the simulation. Because the OWSWF has never been

introduced to the Australian mainland, it is difficult to judge such aspects as rates of spread, rates of development and survivability within the Australian conditions. The *default* values for the relationships (along with many of the relationships themselves) have been based upon OWSWF populations within endemic zones. Because of the inherent uncertainty that this introduces, it is desirable to determine the sensitivity of the model in response to variations within the factors that compose the relationships which determine the survivability, development and dispersal of the OWSWF within the Australian context. This is achieved by varying the parameters (using the scenario interface) that are used within the dispersal and lifecycle models. Through this analysis, the high-level impacts of varying different factors can be established, and this knowledge can ultimately support decision making and policy relating to the control and containment. This section demonstrates the capability of the modelling system for allowing these types of *what-if* scenarios and importantly the ability to derive metrics from the output (using GIS) for analysis.

In order to determine the effects of changes to the default parameters, a base scenario is used for comparison purposes. This scenario will use the values specified throughout chapter four for the dispersal and lifecycle models and the start point for the invasion within the base case will be the city of Brisbane, Queensland (QLD). This start point represents a potential point of entry that exposes the expanding OWSWF population to the temperate climates of southern QLD and New South Wales (NSW) along with the tropical climates of northern QLD. This scenario will last for five years, allowing for the OWSWF population to establish and spread (unhindered) for a significant period of time. Figure 6.19 shows the population distribution after five years within this base case scenario.

Within this scenario it is evident that population has penetrated deeply into the cattle and sheep producing areas of NSW and QLD and has moved as far north as Townsville. Using the GIS for analysis via the integration outlined in 6.4, the graph in figure 6.20 was produced. This figure shows the total affected area (including the previously affected zones, shown in royal blue) over time. The trend shows the expected seasonal fluctuation, with the largest increases in the total area in the warm spring/summer months. The invasion occupies a total area of 625,006Km² after five years. The seasonal trend is confirmed by figure 6.21, which shows the total female OWSWF population by month for the base scenario. This figure shows population growth through the warmer months,



Figure 6.19 The OWSWF population distribution after five years within the base scenario

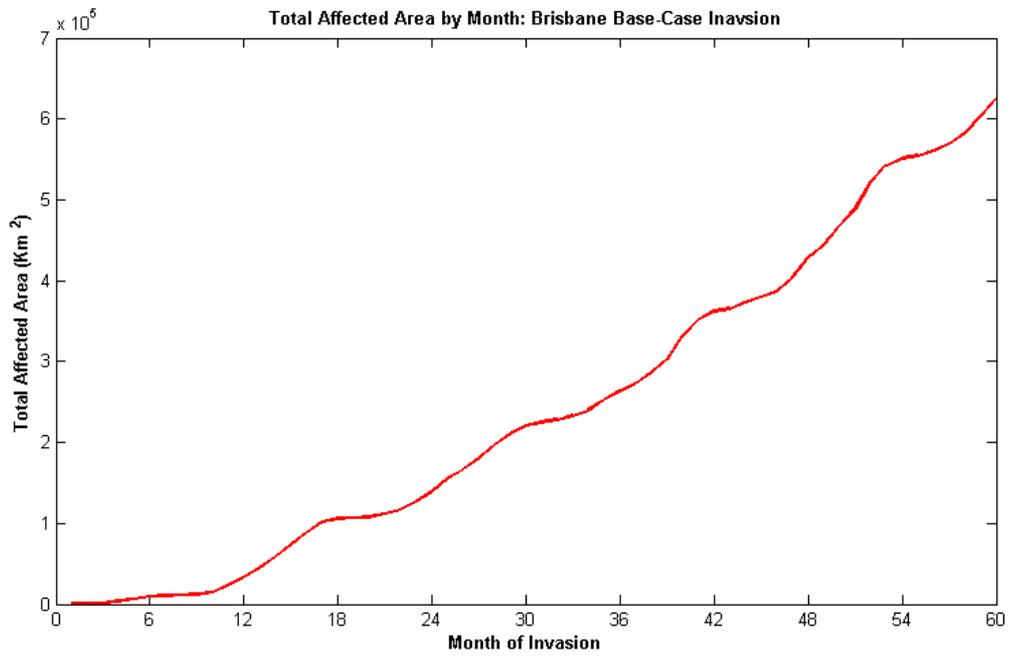


Figure 6.20 The total area affected by the OWSWF invasion for the base-case scenario

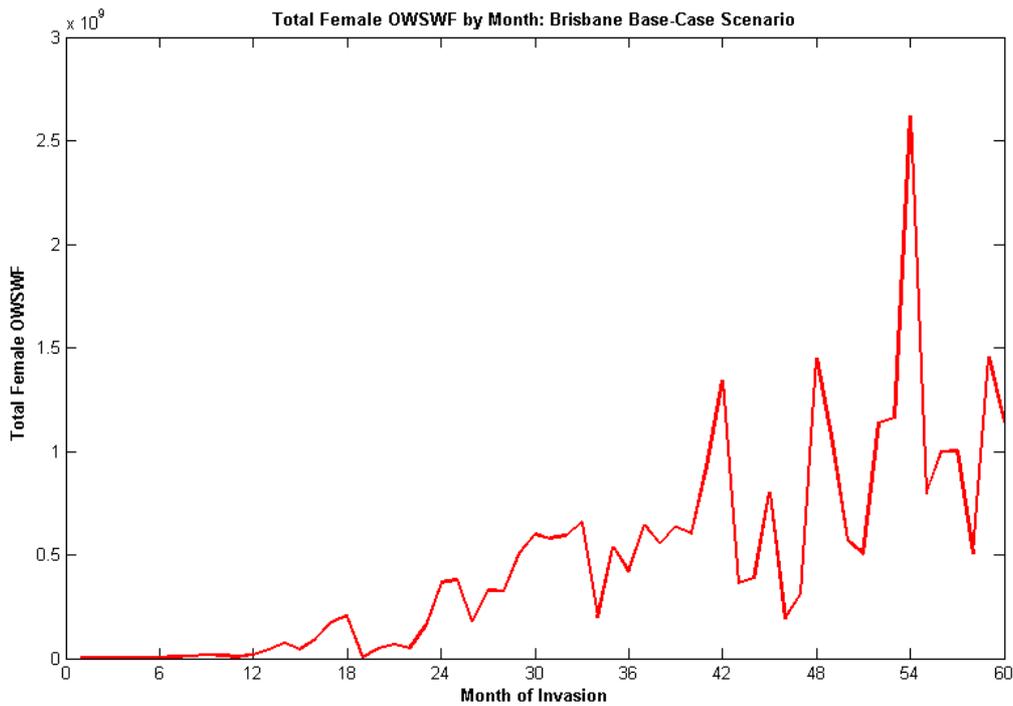


Figure 6.21 The total female OWSWF by month for the base-case scenario

with significant die off through autumn and winter. Months 18, 34 and 46 show sharp falls in the population of adult females, however due to the long 56 day pupation period, the OWSWF is able to survive the coldest periods of the year and re-population during the warmer spring and summer months. Figure 6.20 shows a fairly linear rate of increase in the affected area which implies that, the insect spreads without any hindrance besides the seasonal cold patterns. This relatively linear pattern can be seen in population's peaks at 18, 24, 42, 48 and 54, which approximately follow the temperature peaks.

The first parameter analysed in this section is the *median dispersal distance*. Recall from chapter four that by default, the median dispersal distance is determined by the ratio of gravid OWSWF to the available oviposition sites within a specific grid square. This functionality has parameterised so that a single, fixed, dispersal median can be specified within the dispersal logic. In order to analyse the impact of the median dispersal distance, two scenarios have been completed with fixed dispersal median distances of 2Km and 5Km. Figure 6.22 summarises the total affected area by month for the 2km and 5km scenarios, along with the base case. Both the 2km and 5km scenarios follow the same seasonal patterns as the base case with the 5km dispersal median resulting in a larger affected area after 24 months. After 24 months the rate of spread is slightly higher in the 5km scenario compared to the base case. The 2km scenario shows a linear increase in the affected area, but much lower expansion rate compared to the base case. This pattern is confirmed by figures 6.23 and 6.24, with insect populations and population distributions reflecting the order. This indicates that in the base scenario (with the variable median dispersal distance), after 24 months, the average dispersal distance is slightly lower than 5Km through the simulation. This is most likely due to the insects spread into intensive livestock farming areas (e.g. in Queensland) where the higher livestock numbers result in lower ratios of gravid OWSWF to oviposition sites (hence lower median dispersal distances are used). From this

analysis it is evident that the median dispersal distance has a significant impact on the population and distribution of the insect due to the variable dispersal distance.

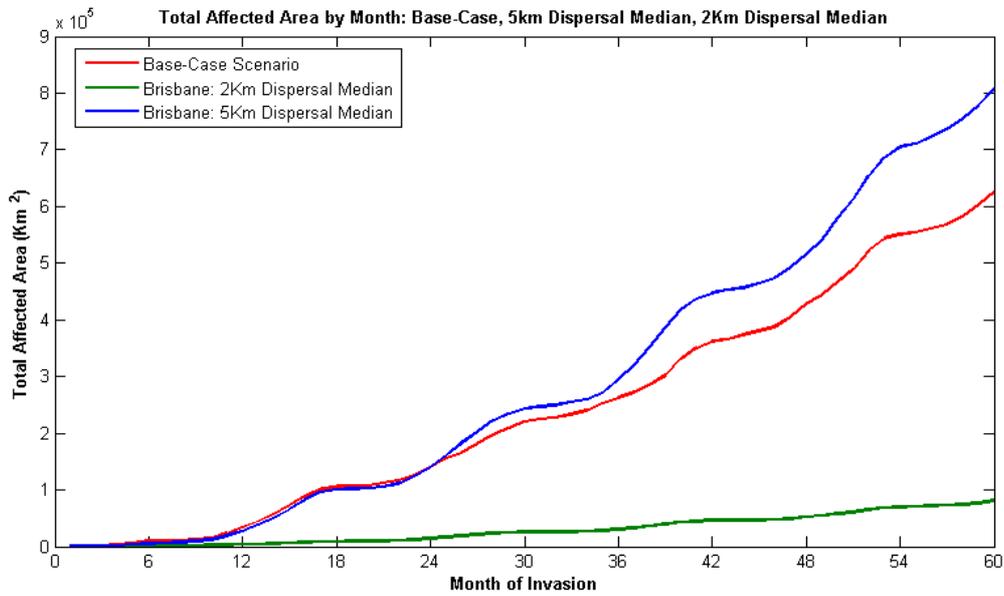


Figure 6.22 Total affected area by month for 2Km, 5Km and default dispersal median distances, with all other parameters left as default

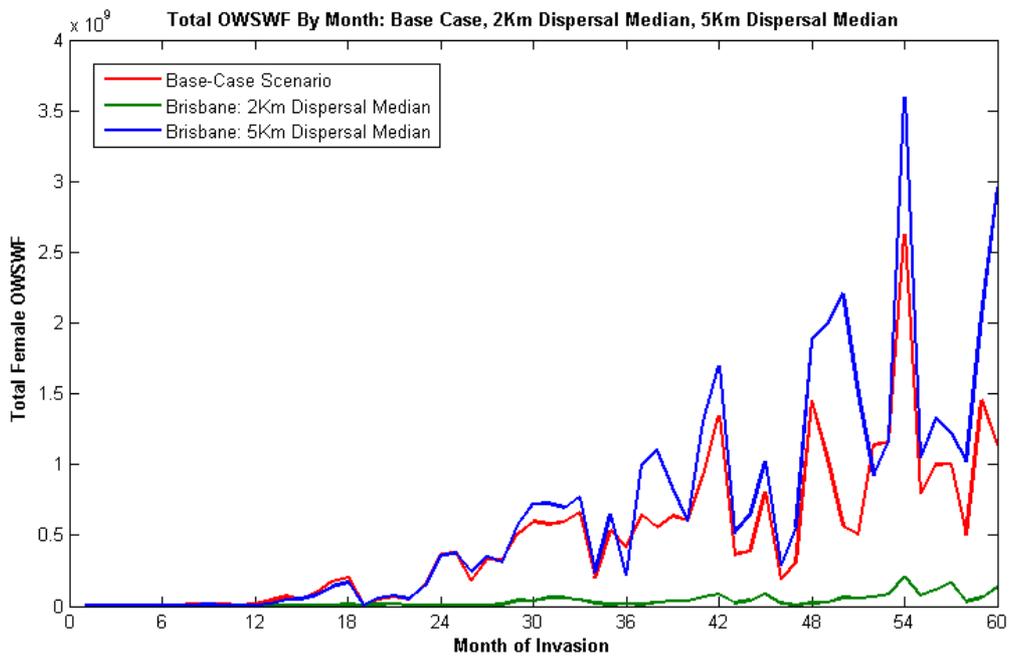


Figure 6.23 The total female OWSWF by month for the base-case, 2km and 5km median dispersal distances with all other parameters left as default

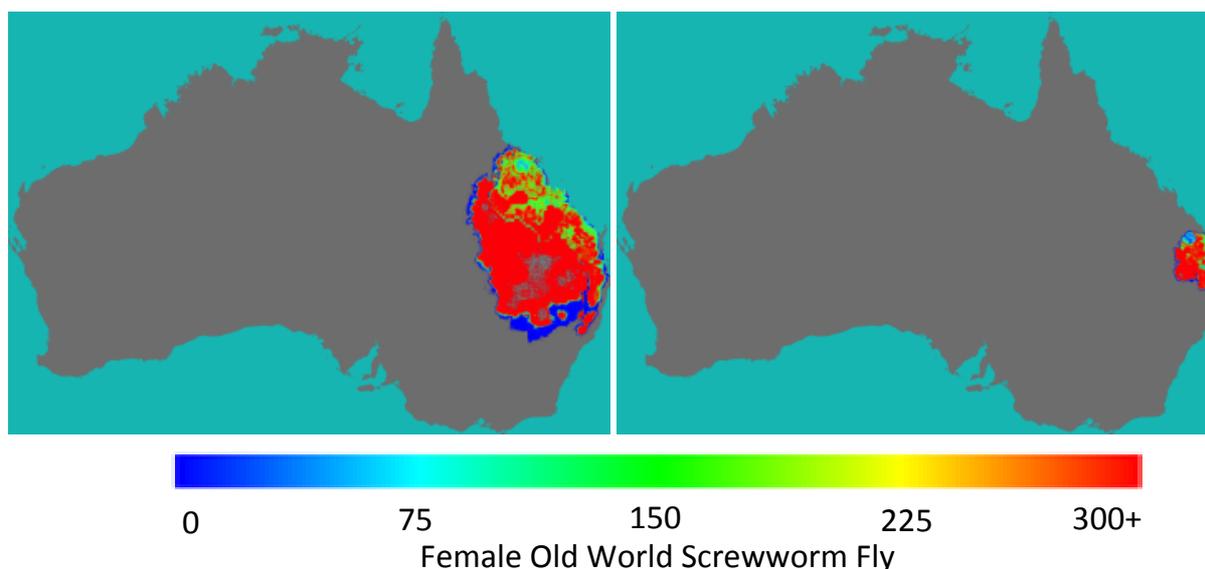


Figure 6.24 The population distribution maps for the 5Km (left side) and 2Km(right side) median dispersal distance scenarios

The second scenario involves reducing the *maximum pupation period* allowed for OWSWF pupae. In this scenario, the pupation period is reduced from 56 days as used by default in the base case, to 28 days. This scenario is based upon the work by Siddig *et al.* [80], which found that after 28 days only between 21-44% of OWSWF successfully emerged from the pupae. In order to test the effects of the findings from [80], the maximum pupation period is changed to 28 days. A more sophisticated approach would be to incorporate a stochastic component to determine pupal survival after 28 days, however the simple cut-off approach (which will result in a lower survivability than observations in [80]) provides an indication of the significance that the pupation period has on the overall invasion. Figures 6.25 and 6.26 show the affected area by month and female insects by month for the pupation period and base case scenarios. These figures indicate that the reduction of the pupation period has little effect on the insect population, and affected area has only small difference in the overall insect population is shown in the final months. The population distribution map in figure 6.27 confirms the trend, showing a similar population distribution to the base case after five years.

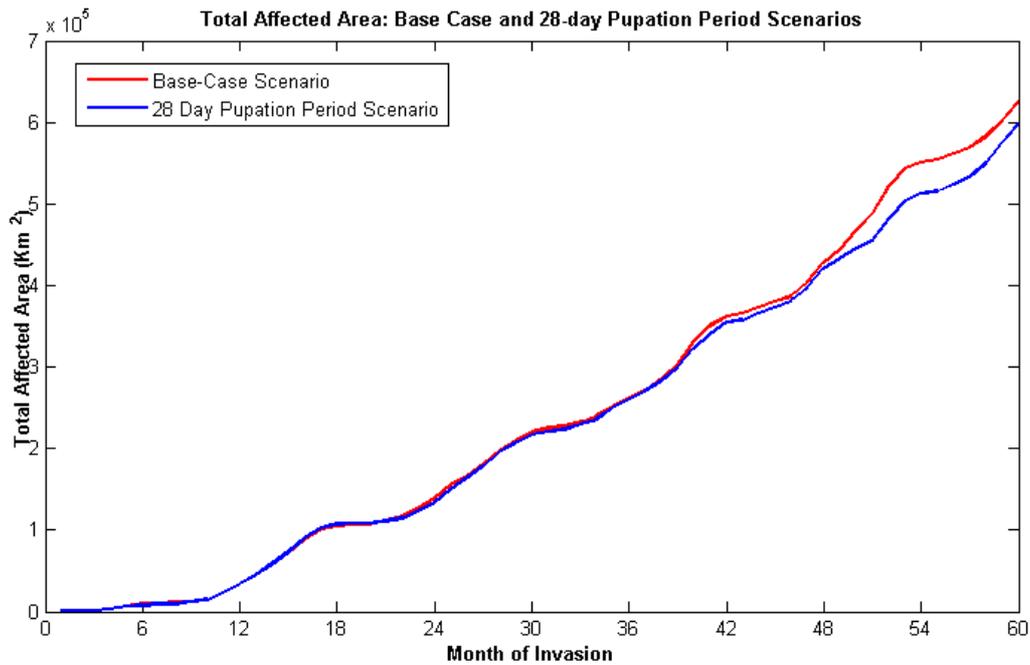


Figure 6.25 The Total affected area by month for the base case scenario and the 28-day max pupation period scenario with all other parameters left as default

Figure 6.25. The Total affected area by month for the base case scenario and the 28-day max pupation period scenario with all other parameters left as default

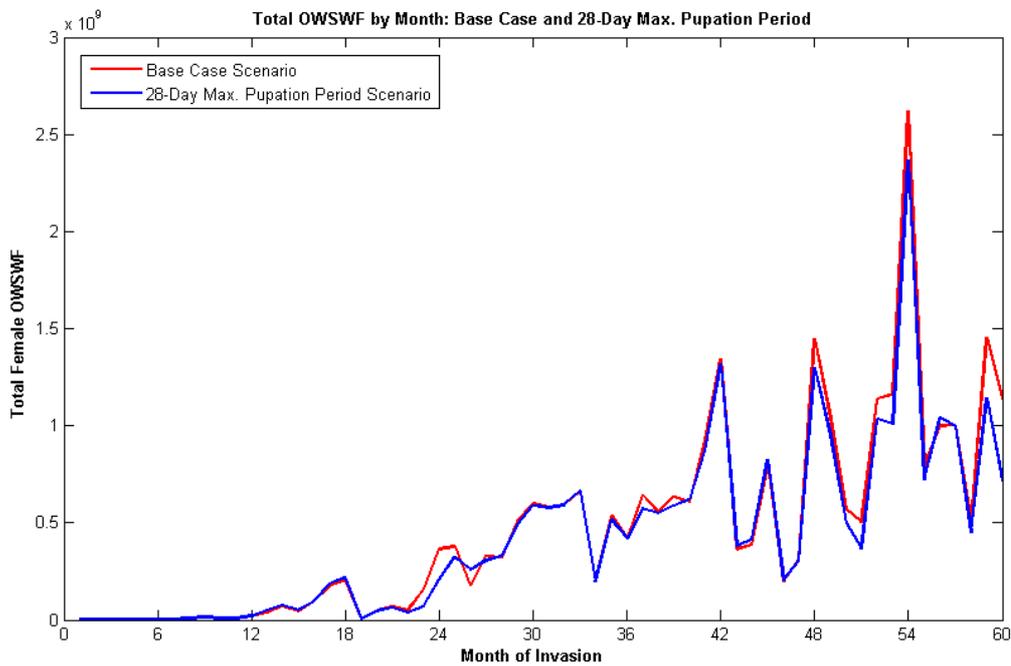


Figure 6.26 The total female OWSWF by month for the base-case and the 28-day max pupation period scenario with all other parameters left as default

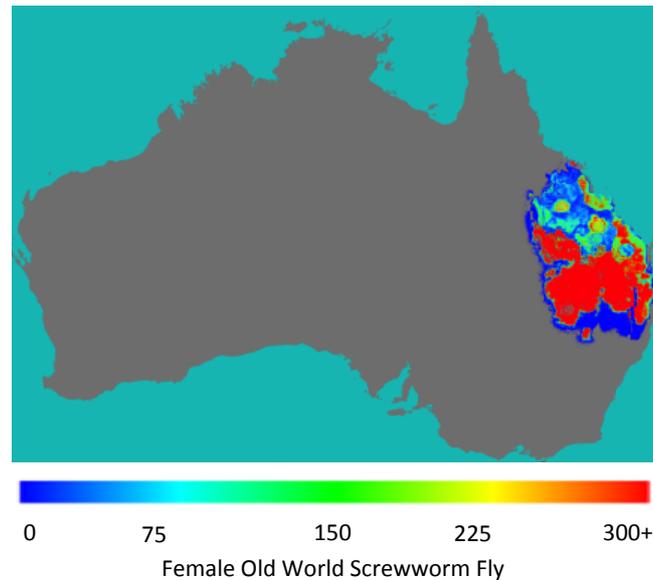


Figure 6.27 The population distribution map after five years for the 28-day max pupation period scenario with all other parameters left as default

The third scenario alters the *maximum survivability temperature* for adult OWSWF and the *minimum survivable temperature* for pupae. Like the previous scenario, the changes to these temperature parameters are based upon observations by Siddig *et al.* [80]. Siddig *et al.* [80] stated that there are significant mortality rates amongst adult OWSWF above 35°C with none surviving above 38.5°C. Within the default configuration, adult OWSWF are killed off if the temperature exceeds 40°C, a slightly higher temperature that could result in an overestimation. In addition to the temperature observations outlined in Siddig *et al.* [80], the AUSVETPLAN [19] states that pupae cannot survive frost conditions. This is not captured within the logic of the model as pupal survivability calculated at 0°C is around 80%. In order to test the effects of these findings, the maximum survivability temperature for adult OWSWF is set to 38.5°C and the minimum survivability temperature is set to 0°C, so that pupae will not survive the frosted condition in colder areas. The results from this scenario are presented in figures 6.28, 6.29 and 6.30 which show un-expected trends when compared against the base case scenario. The expected trend would be to see lower numbers of female OWSWF and a smaller affected area. Instead, both the numbers of OWSWF and the affected area are relatively unchanged by the temperature cut-offs. The small differences observed in the temperature scenarios between the affected areas and the base case can be accounted for by the stochastic design of the model. Figure 6.29 shows that the seasonal fluctuation within the female population has lower peaks in the warm months when compared to the base case along with deeper troughs in the cooler months. These trends are expected, however they do not curb the invasion significantly. Figure 6.27 shows that the population distribution is similar to the base case. Overall, altering the temperatures has limited influence over the OWSWF invasion with a start point at Brisbane.

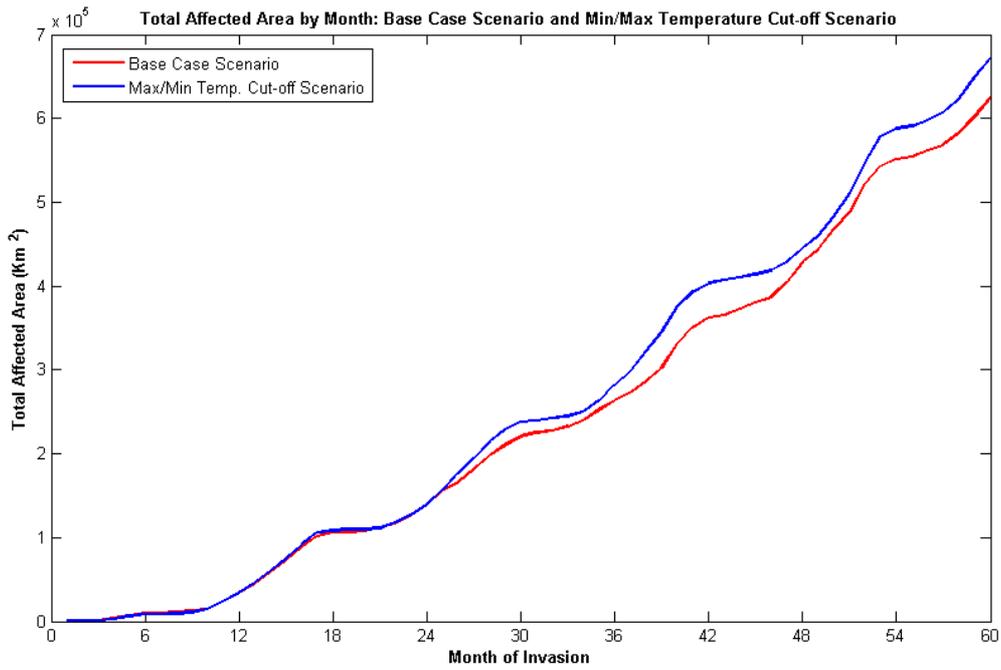


Figure 6.28 The Total affected area by month for the base case scenario and the Min/Max temperature scenario with all other parameters left as default

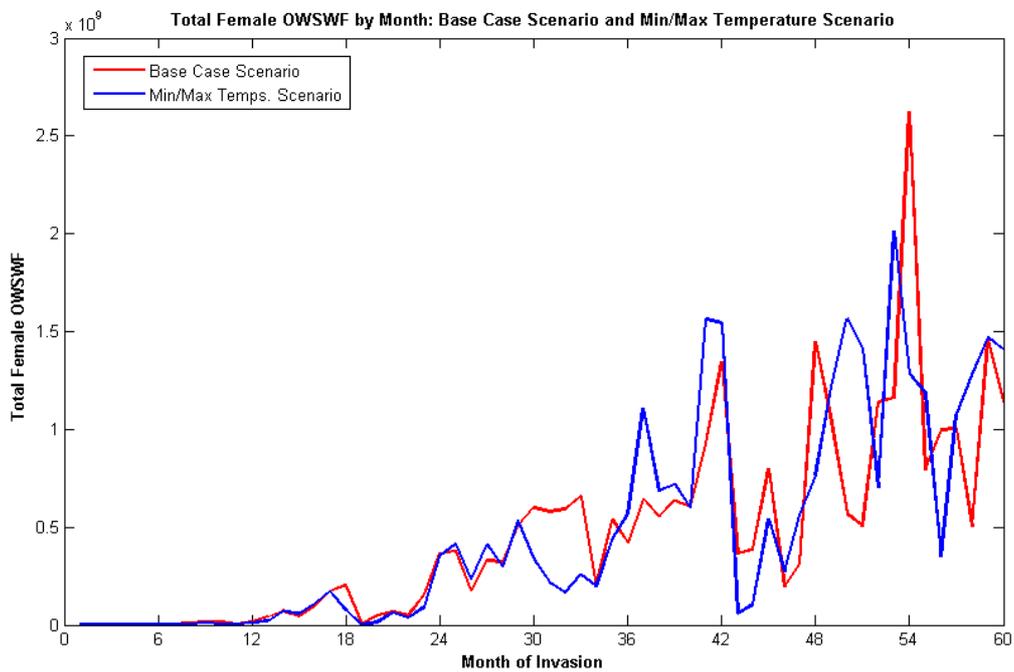


Figure 6.29 The total female OWSWF by month for the base-case and the Min/Max temperature scenario with all other parameters left as default

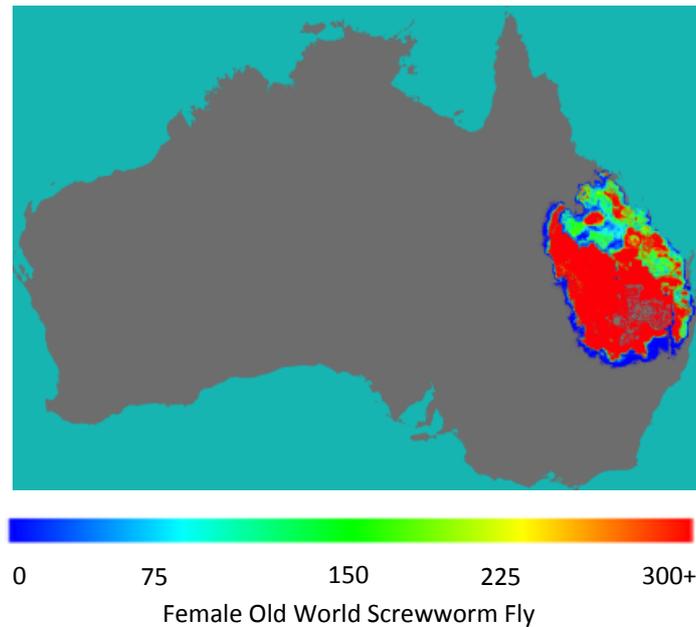


Figure 6.30 The population distribution map after five years for the Min/Max temperature scenario with all other parameters left as default

The final scenario focuses on the *number of oviposits* that are possible for each wound. For the analysis of this parameter, three scenarios were run using values of 20, 50 and 100 oviposits per available wound. This parameter is analysed as it is the most significant limiting factor of the OWSWF population given that there are a finite number of hosts. Figures 6.31 and 6.32 show the affected area and the total adult female OWSWF. These figures show that increasing the number oviposits per wound also increases both affected area and the female OWSWF population. Oddly, allowing 20 and 50 oviposits per wound results in a higher OWSWF population than 100 oviposits per wound. This is a result of the variable median dispersal distance, which is based upon the ratio of gravid OWSWF to oviposit sites. Although a larger number of oviposit sites provide the capability for the population to grow, the pressure to disperse is lowered, especially in areas of high OWSWF mortality. This results in greater OWSWF dispersal in the 20 and 50 oviposit per-wound scenarios, which provide a large number of oviposition sites, while still applying the pressure to disperse. This pressure to disperse pushes the OWSWF into new areas, increasing the overall OWSWF population, which produces the trends in OWSWF numbers and total affected area observed. The population distribution graphs confirm this trend. The seasonal fluctuation patterns in the affected areas and OWSWF population closely matches those of the base case in all scenarios, as does the population distribution, shown in figure 6.33.

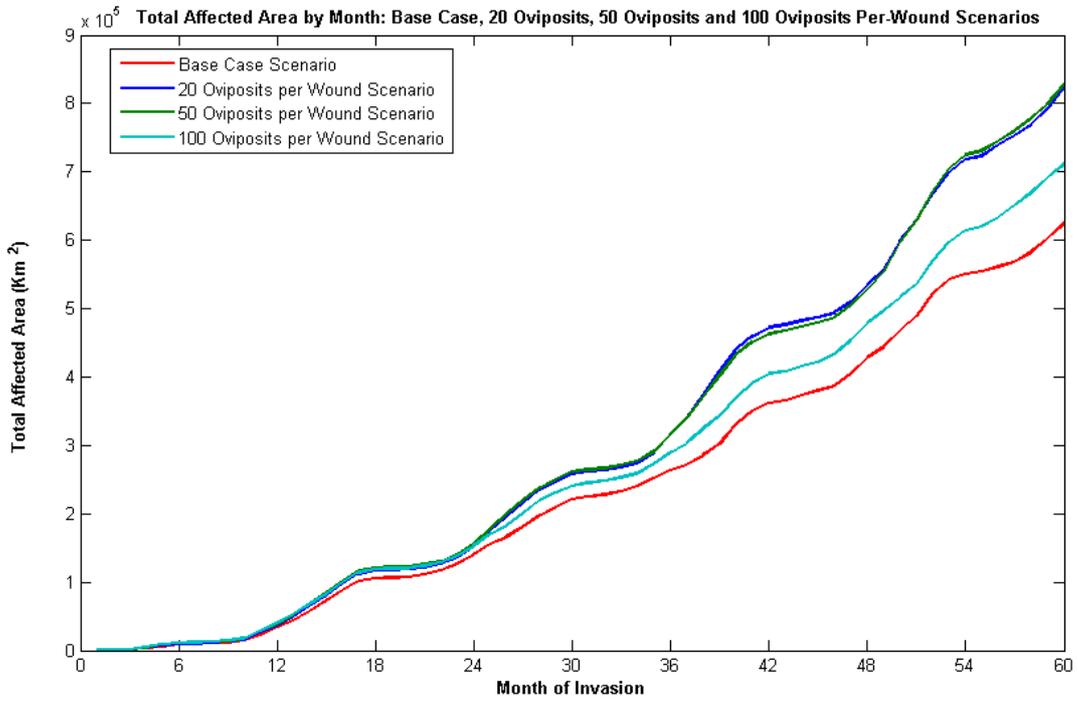


Figure 6.31 The total affected area by month for the base case, 20 Oviposit, 50 Oviposit and 100 Oviposit scenarios with all other parameters left as default

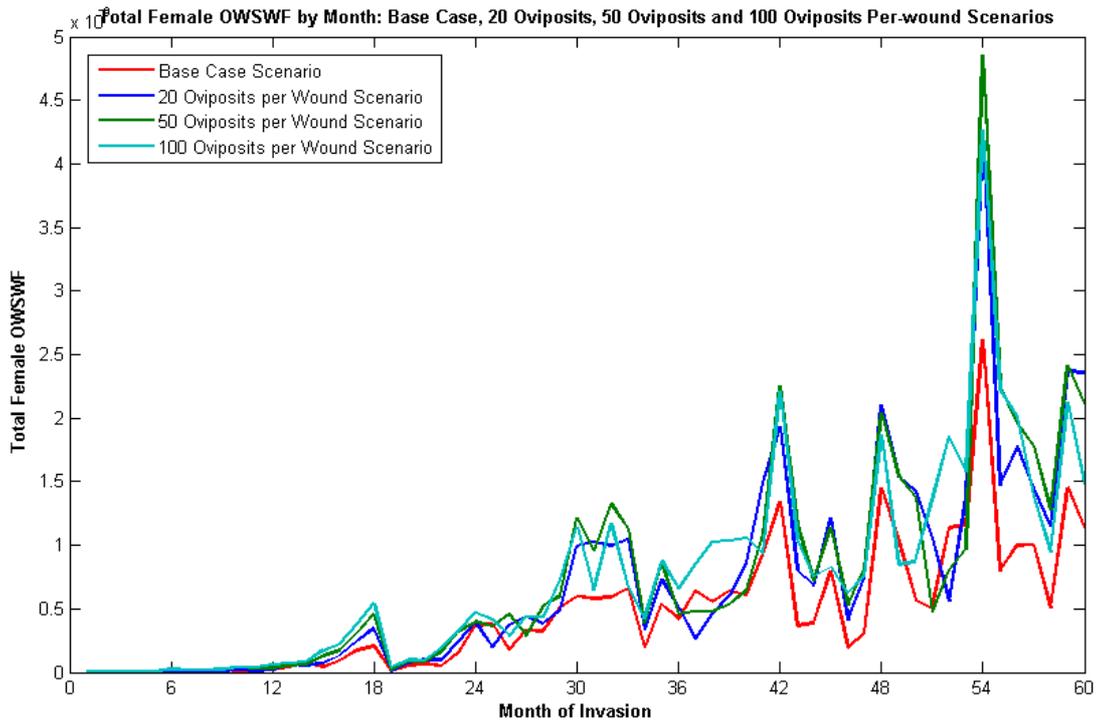


Figure 6.32 The total female OWSWF by month for the oviposit scenarios with all other parameters left as default

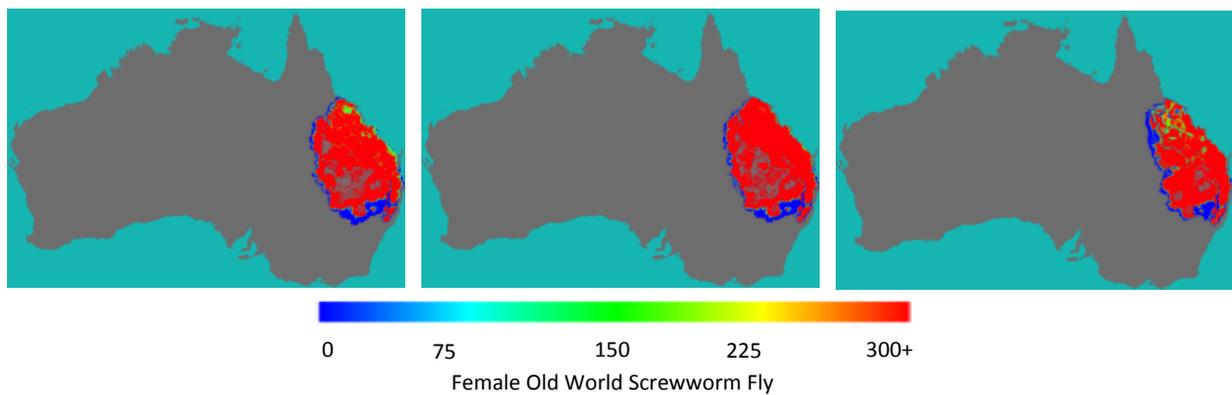


Figure 6.33 The population distribution for the 20, 50 and 100 oviposit scenarios (left to right) at five years

6.7 Concluding Remarks

This chapter has explored the use of geographical information systems with agent-based models and developed a system that allows for a loose integration between stand-alone modelling systems and off-the-shelf GIS software. This work showed how GIS add-in/plugin capabilities combined with a spatial database system can be adapted for use with the agent-based simulation for retrieving and presenting the output. This approach allows the capabilities of the GIS software to be leveraged, against the model to carry out detailed spatial analysis by combining a range of spatial data sources. The approach developed has been successfully used as part of a sensitivity analysis to determine the impact that a range of simulation parameters have on the overall scale and speed of an OWSWF invasion. Within this analysis, the dispersal distance, pupation period, maximum and minimum temperatures and number of oviposits per wound are analysed against a base case scenario that uses the default configuration, outlined in chapter four. Through these analyses, it was discovered that (of the parameters analysed) the dispersal distance and the number of oviposits per-wound have the biggest impact on the invasion's scale. The changes to the other parameters analysed had only a limited impact on the scale of the invasion. The analysis shows how large-scale agent-based modelling and GIS can be combined for a detailed analysis of the phenomena under study.

Chapter 7

Conclusion

7.1 Introduction	179
7.2 Major Contributions	180
7.2.1 Enhancing the Performance of ABMs using Data Compression Techniques and the GPU Implementation Approach	180
7.2.2 Performance Analysis and Validation	181
7.2.3 Integration with GIS	182
7.2.4 Sensitivity Analysis	182
7.3 Future Work	183
7.3.1 Further Advancing the Prototype OWSWF Model Implementation	183
7.4 Concluding Remarks	184

7.1 Introduction

This thesis was set out to explore the development of large-scale agent-based modelling systems, with the over-arching goal of developing a national-scale computational platform for simulating an invasion of the Australian mainland by the Old-world Screwworm Fly (OWSWF). The modelling approach studied builds upon existing work and develops a new individual-based lifecycle and dispersal model to simulate the population of OWSWF, within a high-resolution 1Km x 1Km grid square arrangement. The highly detailed lifecycle and dispersal simulations, coupled with the high number of OWSWF within the model and its spatial scale, requires a huge computational load that is a significant challenge of this research. It is evident from the assessment of the CLIMEX-based model, presented in chapter one, that the implementation of this high-resolution, stochastic and lifecycle-based agent-based model represents a considerable improvement, which in turn provides the ability to capture (and analyse) the effects of the relevant underlying biological processes and the inherent randomness observed within the natural system.

One of the key contributions of this thesis is the development of two original approaches for improving the run-time performance of agent-based modelling systems. Initially, the study explores the use of agent's data compression and aggregation schemes for providing the processing capability that is required for the OWSWF model. This work builds upon existing approaches by introducing a hierarchical structure that takes advantage of processing and data redundancy to improve efficiency. A proof-of-concept implementation using an existing modelling system (NMELDT) demonstrated the potential for this approach; however, due to unique and varied requirements of the OWSWF model it was deemed to be unsuitable for implementing the required platform for the OWSWF model.

The approach that was ultimately adopted for the OWSWF implementation used massively parallel NVidia's CUDA GPU platform. The key contributions of this work include a grid-based scheme for effectively partitioning agent's processing between the large numbers of executing threads and a scheme which aggregates individual agent's state information to the cohort level, while carrying out individual-based calculations. This means that, conceptually, each individual insect is processed as an individual stochastic entity, but the implementation stores the state information at the cohort level.

Following the development of the agent-based OWSWF model, a detailed performance analysis demonstrated the performance enhancement achieved through the use of the GPU approach. Both the methodology and the results provided significant insights into the use of parallel processing for agent-based model's implementation. This study showed how the efficiency of the system changes as the number of agents in the simulation increases, and the represented spatial distribution of the invasion area changes. The key contribution of this work is an understanding of how agent-based models perform in a massively parallel execution environment.

The final sections of the thesis explore the integration of geographical information systems (GIS) with agent-based modelling systems to provide advanced spatial analysis and presentation capabilities. This work is

motivated by the lack of these advanced visualisation capabilities in existing modelling systems. This research builds upon existing work by providing specific implementations that make use of web-based mapping services as well as customisations within off-the-shelf GIS packages. The use of this loosely coupled GIS integration is demonstrated in a sensitivity analysis, which examines an array of parameters that contribute to the population's distribution and spread. This study represents a significant contribution by showing that the dispersal distance has the most significant impact on the population's distribution for an invasion that spans both temperate and tropical zones.

The following sections summarise the contributions of this thesis, and the outline of future work that will be articulated on from the developments produced within this study.

7.2 Major Contributions

7.2.1 Enhancing the Performance of ABMs using Data Compression Techniques and the GPU Implementation Approach

As mentioned in the previous section, one of the key research areas addressed in this study is the implementation of performance enhancement approaches (specifically data compression techniques and GPU processing technologies) to provide the required capabilities for supporting the computationally intensive agent-based model. The implementation in the study builds upon the existing literature showing how the approach can be applied to a large scale ABM and the introduction of the hierarchy for improved performance. This contribution is significant for modelling situations where this is a large amount of redundancy within the agent's state information. By improving the efficiency of processing and memory management within such models, increasing the number of agents can be accommodated while larger scale phenomena can be modelled.

The study also highlights the limitations of this approach, showing how models (using the OWSWF agent-based model as an example) with complex agent state information and low levels of redundancy. This situation results in excessively deep information hierarchies that perform poorly in terms of the execution speed of the update processes on the model's agents set. Further development is needed for this approach in order to be fully applicable to models that fall into these unsuitable categories.

With the use of the data compression schemes shown to be un-suitable for the complex OWSWF model, the next approach studied uses graphics processing hardware and the NVidia's CUDA platform to achieve our aim. The review included in chapter two showed that the graphics processing unit technologies, including CUDA, can be successfully applied to agent-based modelling systems to produce highly detailed simulations. The work in this chapter of the thesis builds upon the existing systems by showing how a cohort aggregation, a cell-based spatial representation and agent-based processing can be combined to produce a national scale model for the population dynamics of the OWSWF spread. By using the cohort aggregation approach, alongside the agent-level processing, the model conceptually presents the individual insects as agents within the cells of a grid-based spatial representation. The study also shows how this grid-based

system can be used to partition processing within the parallel execution environment provided by the GPU platform. This allows the logic of multiple agents to be processed efficiently by a single thread of execution, thereby significantly reducing the overhead of launching a new thread for each agent that is incurred by the launch of the CUDA kernels. This approach differs from other literature where individual agents are processed by their own threads of execution.

7.2.2 Performance Analysis and Validation

In order to assess the effectiveness of the CUDA implementation of the OWSWF model, a detailed analysis of the complexity and performance was undertaken. In this process, a single threaded implementation that uses serialised execution was compared to the CUDA parallelised implementation, comparing the execution times of the different components within the simulation. The analysis included any overheads unique to the individual approaches, such as the CUDA kernel launches and the memory copy operations. The study showed that the CUDA approach achieved significantly lower execution times across all sections of the model, and the performance trends indicated that the CUDA implementation scales well as the agent (i.e. insect) population grows within the simulation.

In addition to the execution time performance analysis, the efficiency of the partitioning of the agent processing is analysed. Recall from chapter four that the agent processing is partitioned using the grid structure used to represent the spatial distribution of the insects. As mentioned in the previous section, this allows a single thread to process multiple agents' logic. The analysis (presented in chapter five) showed that as the spatial distribution of the insects increased, the processing of agents' logic was more evenly distributed across the available threads. This type of analysis is relatively novel, with most other modelling work reviewed focusing solely on the analysis of execution speeds, which gives insight into the balance of processing within the GPUs across the execution of the agent-based OWSWF implementation.

Following on from the analysis of the model's execution performance, its output was compared to that of the existing CLIMEX-based implementation. This comparison showed that agent-based implementation produced consistently more conservative population distribution. The population density patterns within the affected areas were consistent between the models, indicating that the population models possess a similar level of sensitivity to environmental factors. Analysis showed that the differences in the predicted population distributions are likely to be due to the higher resolution of the agent-based implementation coupled with statistical under-fit inherent within the dispersal sub-model used by the CLIMEX-based system. Further validation work will be important to determine if the higher resolution approach accurately represents the observed insect dispersal trends.

Another secondary but interesting contribution made by the implementation is the use of interpolation techniques, specifically bi-cubic interpolation, to upscale the low resolution spatial data-layers. This process is usually used to scale multi-channel (RGB) images; however it has been applied to the single channel data layers to generate smooth 1Km x 1Km resolution layers from the existing 20Km x 20Km layers. This process is

carried out on demand using a CUDA kernel function and allows the model to operate at the high resolutions required for using input data sets of varying resolutions.

7.2.3 Integration with GIS

The final area of study covered in this thesis involves the use of Geographical Information Systems (GIS) with agent-based modelling systems. This work focuses on the integration of off-the-shelf GIS packages with the agent-based OWSWF implementation. This work shows how customisations within off-the-shelf GIS software can be developed, which allows the output from the OWSWF model to be directly accessed for analysis and presentation. The work in this chapter of the thesis extends the conceptual understanding that is presented in the literature review by providing an implementation approach that supports the use of GIS alongside the agent-based OWSWF model. The approach adopted for the integration with the GIS is not necessarily specific to agent-based models, and the result could be applied to a range of other applications where complex spatial information is used. The key benefits of using the existing GIS software in place of a built-in presentation system include reductions in development overhead, the availability of expertise, and a reduction in the need for specialised training and testing.

The GIS integration is demonstrated through a detailed sensitivity analysis discussed in the next chapter of the thesis. Through this demonstration, spatial queries are used to determine the total affected areas and the strike rates of livestock across the affected areas. In addition to the analysis of the output from the model, the GIS integration allows external data layers to be added. This allows for the OWSWF model's output to be used in spatial queries across multiple external data sources.

7.2.4 Sensitivity Analysis

In addition to the validation against the CLIMEX-based OWSWF model, a sensitivity analysis was carried out to assess the impacts of various parameter changes on the trends within the model's output. The parameters selected for analysis within this study were chosen based upon updated knowledge regarding the biology of the OWSWF from recent literature (i.e. updates to the understanding used to develop the original lifecycle implementation) that were not updated within the model directly.

The analysis looked at outputs including the spatial distribution patterns, the total size of the affected areas and the average strike rate over time for each livestock class. The observations showed that (of the parameters tested) the median dispersal distance and the number of oviposition sites per wound have the biggest impact on the population's distribution.

This analysis provided valuable insights, showing that changes to the input parameters to reflect observations in recent overseas OWSWF outbreaks, such as those in Iraq, may have limited impact on the output overall.

7.3 Future Work

The study presented within this thesis can be broadly categorised into two themes. The core of this study develops a prototype implementation with a wide scope for future development into a full production system. The improvements and additions to the model that are discussed in this section were ultimately left out of the scope of the study in this research. This decision was made primarily based upon the availability of relevant data sets required for these additional modelling extensions and the time required for cleaning and validation. In addition to the improvement to the OWSWF model developed at the core of this research, future research directions also include the use of the techniques and approaches studied that could readily be applied to other systems and refined for further improvements within the core OWSWF model.

7.3.1 Further Advancing the Prototype OWSWF Model Implementation

The OWSWF model developed through chapters four and five consists of only the base modelling platform for capturing the core OWSWF population dynamics. In order for the system to provide the true decision support capabilities that policy makers require, the model will need to be extended to include an economic sub-model that estimates the primary economic losses based around the animal strikes, costs of prevention and containment and livestock prices within each production sector. In order to implement this additional sub-model, new data layers and additional logic will need to be included within the existing kernels used in the model. The architecture that is used in the existing implementation has been designed with this extension in mind, providing the strike rate calculations that act as the coupling point between the agent-based population model and a future economic sub-model. This future economic sub-model could be integrated into the spatial output from the model, allowing the economic impacts for individual regions to be analysed.

Closely related to modelling the economic impacts is the simulation of control and containment measures. As mentioned in chapter one, the OWSWF invasion can be countered and ultimately wiped out by a range of measures including containment/quarantine of animal movements, prophylactic drenching, and the sterile insect technique (SIT), etc. Incorporating the impacts of these measures will involve the addition of new data layers and modification of the kernels responsible for simulating the population dynamics. As mentioned in the previous paragraph, the simulation of the control and containment measures will need to be coupled with the economic sub-model to provide a complete capability for carrying out cost-benefit analysis on a national scale.

The incorporation of an agent-based (i.e. individual-based) simulation of livestock movements also has the potential to improve the overall modelling capabilities of the system. The incorporation of such a sub-model into the system would allow for a very direct modelling of spot-outbreaks of OWSWF caused by the movement of livestock that carry the OWSWF larvae from affected areas to unaffected zones. This would also tie into the simulation of containment and quarantine measures that are aimed at limiting this type of spread. Implementing this additional sub-model would require livestock movement data-sets (like those used in the NMELDT) and additional logic to manage livestock strikes on the individual animals.

The final area of improvement within the model developed in chapters four and five involves the updating of the bio-climatic datasets that were adopted from the existing CLIMEX-based modelling system. These datasets were used from the original modelling platform as they allowed for a more direct comparison between the CLIMEX-based implementation and are readily available for use. Recall from chapter four that these data sets are up-scaled from the lower 20Km x 20Km resolution used in the CLIMEX-based system. This up-scaling process could easily be removed with the use of higher resolution data-sets. The use of higher resolution data-sets will also improve the accuracy of the model's output.

7.4 Concluding Remarks

This thesis has demonstrated that the agent-based modelling approach can be successfully applied to model highly detailed simulation systems at a national scale, with the goal of supporting policy makers and scientists. The study has shown that both GPU technologies and data compression techniques can be utilised to provide a high-performance platform on a standard personal computer for executing these models, thus eliminating the need for expensive technologies and services. This work has built upon existing literature, showing how off-the-shelf GIS software can be used together with agent-based modelling systems. Finally, through the use of the aforementioned techniques and technologies, this thesis has produced a high-resolution agent-based model for an OWSWF invasion of the Australian mainland that make uses of a staged lifecycle-based sub-model for survivability and development, and an individual-based stochastic dispersal sub-model to simulate the distribution of the population.