Widening Stakeholder Involvement: Exploiting Interactive 3D Visualisation and Protocol Buffers in Geo-Computing

Christopher Andrew McCreadie

A thesis submitted in partial fulfilment of the requirements of

the University of Abertay Dundee for the degree of Doctor of Philosophy

June 2014

I certify that this thesis is the true and accurate version of the thesis approved by the examiners

Signed ______________________________

Date ______________________

(Director of Studies)

Declaration

I, Christopher McCreadie, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I can confirm that this has been indicated in the thesis.

Signed ______________________________

Date ______________________

Dedication

In loving memory of Wilma Gillen:

My teacher and my friend – You are missed everyday.

Don't walk behind me; I may not lead. Don't walk in front of me; I may not follow. Just walk beside me and be my friend.

Acknowledgements

Firstly I would like to thank my supervisors, Dr Ruth Falconer and Dr. David Blackwood. If not for both of you this thesis would not exist. Your encouragement, help and support throughout my time at Abertay has been outstanding and very much appreciated. You are both amazing supervisors and I thank you from the bottom of my heart.

Constantly working on one project can be difficult and I want to thank Dr. Ken Scott-Brown, Prof. Jim Bown, Prof. Chris Jefferies, Dr. Rebecca Wade, Dr Robin Sloan and Dr. John Isaacs – Thank you for the various projects we worked on together and recognising my need to have a change of scenery every so often! I especially enjoyed being part of the academic supply pool and I should thank Dr. Geoff Lund for giving me the opportunity to pass on some of the knowledge I acquired while at Abertay to others.

I would not have had the opportunity to do any of this or the degree beforehand without the help and (financial, mental and at times physical) support of my parents. Mum and Dad – Thank you. My grandparents deserve huge credit as well. I know you always believed in could do it and your unshakeable belief managed to convince even me, thanks Nan, thanks Pop.

To Lori and Eilidh –Perhaps the only two people to witness the highs and lows of this adventure in its fullest. I would not have survived without your constant support, encouragement and kindness. Thank you.

Abstract

Land use change has an impact on regional sustainability which can be assessed using social, economic and environmental indicators. Stakeholder engagement tools provide a platform that can demonstrate the possible future impacts land use change may have to better inform stakeholder groups of the impact of policy changes or plausible climatic variations. To date some engagement tools are difficult to use or understand and lack user interaction whilst other tools demonstrate model environments with a tightly coupled user interface, resulting in poor performance. The research and development described herein relates to the development and testing of a visualisation engine for rendering the output of an Agent Based Model (ABM) as a 3D Virtual Environment via a loosely-coupled data driven communications protocol called Protocol Buffers. The tool, named Rural Sustainability Visualisation Tool (R.S.V.T) is primarily aimed to enhance non-expert knowledge and understanding of the effects of land use change, driven by farmer decision making, on the sustainability of a region.

Communication protocols are evaluated and Protocol Buffers, a binary-based communications protocol is selected, based on speed of object serialization and data transfer, to pass message from the ABM to the 3D Virtual Environment. Early comparative testing of R.S.V.T and its 2D counterpart RepastS shows R.S.V.T and its loosely-coupled approach offers an increase in performance when rendering land use scenes. The flexibility of Protocol Buffer's and MongoDB are also shown to have positive performance implications for storing and running of loosely-coupled model simulations. A

3D graphics Application Programming Interface (API), commonly used in the development of computer games technology is selected to develop the Virtual Environment. Multiple visualisation methods, designed to enhance stakeholder engagement and understanding, are developed and tested to determine their suitability in both user preference and information retrieval.

The application of a prototype is demonstrated using a case study based in the Lunan catchment in Scotland, which has water quality and biodiversity issues due to intense agriculture. The region is modelled using three scenario storylines that broadly describe plausible futures. Business as Might Be Usual (BAMBU), Growth Applied Strategy (GRAS) and the Sustainable European Development Goal (SEDG) are the applied scenarios.

The performance of the tool is assessed and it is found that R.S.V.T can run faster than its 2D equivalent when loosely coupled with a 3D Virtual Environment. The 3D Virtual Environment and its associated visualisation methods are assessed using non-expert stakeholder groups and it is shown that 3D ABM output is generally preferred to 2D ABM output. Insights are also gained into the most appropriate visualisation techniques for agricultural landscapes. Finally, the benefit of taking a loosely-coupled approach to the visualisation of model data is demonstrated through the performance of Protocol Buffers during testing, showing it is capable of transferring large amounts of model data to a bespoke visual front-end.

Table of contents

List of figures

List of tables

Chapter 1: Introduction

1.1 Project Background and Rationale

Scotland's land use provides a range of services such as food, timber, water, energy, employment and recreation. Land use impacts on the aesthetic quality of landscapes and also plays a role in the sustainability of local, regional and national ecosystem services (Scottish Government, 2012). A major challenge of land use planning and strategy is designing future cropping regimes that are environmentally sustainable, suited to diverse and ever-changing markets and yet still appropriate for the climatic conditions found in Scotland (Scottish Government, 2011).

The implementation of land use regimes is the decision of the land owner. The study of land owner decision making is therefore required when attempting to understand Land Use Cover Change (LUCC) because local cultural influences, farmhouse demographics, past experiences, personal values and type of business all affect the drivers of behaviour for land owners (Meert et al., 2005).

Agent Based Modelling (ABM) is a modelling approach commonly used to analyse the interconnected nature of social and environmental systems (*Heckbert et al.*, 2010). This makes ABM suited for modelling LUCC, with LUCC making up one of the many systems that feedback or into local, regional and national ecosystem services. An agreed characteristic of LUCC models is that they are tools capable of communicating the potential effects of

LUCC to stakeholders through the visualisation of model output (Veldkamp & Verburg, 2004; Van Lammeren *et al.*, 2010). Typically, LUCC models visualise their output to stakeholders through a series of plots or graphs (Batty, 2003), 2D systems such as GIS (Evans & Kelley, 2004) or 2D displays designed specifically for modelling toolkits (Guillem *et al.*, 2012; Guillem *et al.*, 2013). There has been a recent push to determine the possibilities of visualising model output within 3D Virtual Environments (VE) which can be attributed to the increasing amount of 2D and 3D datasets, the abundance of detailed 3D models of rural and urban landscape components and the increase in computational power (Crooks *et al.*, 2010).

Two broad methodologies of implementing 3D output of ABMs have been determined. These are (i) close coupled models and (ii) loose coupled models. To date, the technical limitations of both methodologies have kept this emerging technology from being a serious alternative to 2D model output.

Close coupled models

Close coupling is achieved by bundling the visual output of an ABM as part of the models codebase. Many ABMs are built on platforms and toolkits which are not capable of 3D acceleration (3D acceleration is the act of using a dedicated graphics chip to perform specific numerical calculations at fast speeds). This affects the speed in which the visual output of the model can be updated and refreshed on a monitor.

Loose coupled models

Loose coupling is achieved by segregating the ABM from the visual component. The speed in which data can be transferred from the ABM to the visualisation component has proved, to date, problematic with speed results suffering from high latency, making real-time visualisation unachievable (Crooks *et al.*, 2010).

1.2 Statement of Aims and Objectives

Currently LUCC ABMs are limited to two dimensional representations of complex datasets. There is little literature available that documents proven and effective visualisation methods of land use cover change within a 3D VE and the technical difficulty of close and loose coupled models has been documented when attempting to output ABM data within a 3D environment (Vizzari, 2008; Crooks *et al.*, 2010).

This thesis evaluates and identifies gaps in existing 3D ABMs and consider what techniques from the field of computer games technology can be applied to create a 3D VE capable of visualising the realtime output of a LUCC ABM. From this evaluation a prototype was created, the Rural Sustainability Visualisation Tool (R.S.V.T) that is capable of visualising the output of a LUCC ABM in a 3D VE using a loosely coupled approach. A visualisation evaluation of the tool will be carried out with non-expert stakeholders to evaluate the effectiveness of the tool including its aesthetic quality in comparison to 2D output and the effectiveness of 3D visualisation methods used to show LUCC modelled output. The research hypothesis is defined as

“Can 3D visualisation be used to portray ABM output to non-expert stakeholders that accurately conveys LUCC in a real-time immersive environment?”

1.3 Methodology

The existing literature on visual output of 2D LUCC models was analysed to identify common problems and issues. From this literature review it is clear that there is a lack of interactivity and control over the visual output of LUCC models and ABMs. Existing tools that already have a 3D component were also identified. However these tools showed either an increase in interactivity but a decrease over the control of visual output or an increase in the control over visual output but a decrease in the interactivity exhibited by visualisation. The literature review highlights the technological limitations of 3D ABMs and can be found in **Chapter 2**.

The Lunan catchment was the chosen case study for the R.S.V.T. prototype. The Lunan catchment was selected as a case study as part of the EcoChange project (EcoChange, 2011) and the EcoChange project is contributed to, in part, by the Lunan ABM which is in concurrent development at the University of Edinburgh (Guillem, 2012). An overview of the case study is given as well as a list of indicators used within both the ABM and the subset chosen to be visualised by R.S.V.T. These indicators form the design elements of R.S.V.T. These design elements are defined as the visual parts of the R.S.V.T system that are updated by the output of the ABM. The full rationale for the subset of indicators can be found in **Chapter 3**.

The R.S.V.T prototype, which is defined as a 3D VE, uses digital elevation data to build the Lunan terrain using the freely available Shuttle Radar Topography Mission (SRTM) data provided by NASA. A custom shapefile importer was also developed to allow the inclusion of ESRI shapefiles to overlay the terrain, much like the features available in GIS packages. The shapefile used in the prototype contains the boundaries of each land parcel and these boundaries are triangulated and built into model meshes which can then be textured based on land use information supplied by the ABM output. The technological problems that surround loosely-coupled model and visualisation systems are overcome through the use of Protocol Buffers, a network communications protocol that is responsible for transferring data to and from the ABM and 3D VE. This is done through strict adherence to the M-V-P design pattern which permitted R.S.V.T's codebase and the ABM codebase to maintain their respective goals, development teams and development timespan whilst ensuring compatibility. The complete design, implementation and visualisation methods developed for R.S.V.T can be found in **Chapter 4** while the rationale for using Protocol Buffers, including a pilot test to prove feasibility, can be found in **Chapter 5**.

A visualisation evaluation forms the testing strategy that was adopted after development of the prototype R.S.V.T. A review of the different methods of visualisation evaluation concluded that four approaches are commonly used when evaluating visualisation.

1) Comparing design elements within a controlled experiment
2) Usability evaluation of the V.E
3) Comparing two or more software tools within a controlled experiment
4) Case studies of using V.E within a realistic environment

The testing strategy adopts the first three approaches by undertaking two visualisation evaluation techniques known as Visual Preference Surveys and Task Based Testing. A total of 31 non-expert participants agreed to take part in the evaluation sessions and each participant took part in both the Visual Preference Survey and Task Based Testing. By following this research methodology it is believed that the following questions can be answered.

1) Is the visualisation sufficiently effective at conveying Land Use Cover Change?
2) Can a loosely coupled model communicate data at an acceptable rate?
3) Are the methods of data capture and storage flexible and scalable to support further development?

The visualisation evaluation, which is made up of the Testing Strategy (see **Chapter 6**) and the corresponding Results (see **Chapter 7**), addresses R.S.V.T.'s effectiveness of land use cover change. The feasibility test of Protocol Buffer's performance as a loosely-coupled messaging protocol can be found in **Chapter 5** and the development of an “offline-mode” (see **chapter 4.3.5**) demonstrates a flexible method for data capture and storage.

1.4 Summary of Key Findings

The key findings of the project can be found, in full, in **Chapter 8**. The findings show that the current state of ABM toolkits has little support for integrated and intuitive 3D visualisations. The current solution of close or loose coupling of ABMs was also shown to suffer from technical limitations. The development of R.S.V.T has shown that it is possible to create a realtime, interactive 3D VE that is driven by a loosely coupled model by implementing M-V-P compliant codebase and a network communications protocol such as Protocol Buffers. Protocol Buffers proved flexible enough to integrate with MongoDB, a NoSQL database which allowed for the development of a data capture and retrieval system ensuring that model simulations could be run unencumbered by the model's resource requirements. Finally, it was shown that the V.E is effective at communicating model data through custom visualisation methods of design elements but some work on determining a wider range of acceptable visualisation methods is still required.

Chapter 2: Literature Review and Related Works

2.1 Background

Real time interactive visualisation supports stakeholder interaction and engagement (Bailey *et al.*, 2002; Bouwman, 2006). Many research areas already benefit from such interactive visualisations including medical training simulators (Stone, 2005), combat training for members of the armed forces (Prensky, 2001), urban planning (Appleton & Lovett, 2003; 2005; Isaacs *et al.*, 2011) and landscape and rural planning (Malczewski, 2004; Wergles & Muhar, 2009). The body of work contained within this thesis supports a novel approach to developing a V.E that is capable of bi-directional communication of scientific model output through a 3D and interactive visualisation in real-time. According to Thalmann (2007), 3D visualisation has potential advantages over 2D visualisations, specifically, (i) the visualisation is easier to understand when the user has a visual reference of the location, (ii) it enhances the communication of ideas, (iii) it aids in model validation whereby spotting errors in a 3D visualisation should be easier than a 2D visualisation.

Land use modelling is considered a Complex Adaptive System (CAS) (August *et al.*, 2001). CAS is a large number of sub-systems working independently of each other yet still affecting other subsystems (Holland, 2006). CA models lack the descriptive nature of social interactions which can be solved through the implementation of ABMs due to its asynchronous behaviour, allowing agents to behave in ways unfamiliar to other agents within the model (Castiglione, 2006). However, traditional ABM applications suffer from the tight coupling of two components; namely (i) the model and (ii) the visualisation of the output of the model. Tight coupling is the close integration of two components, to the point it becomes difficult or impossible to make changes to one component without making changes to the other. Traditional ABM output is 2D in nature due to the expert nature of the stakeholders using the tool and is also used by the modellers and programmers for debugging purposes. Close coupled systems are also shown to suffer from limitations of their graphical capabilities when used in conjunction with ABM toolkits (Crooks *et al.*, 2008).

The advancements of 3D rendering technology, particularly the introduction of programmable pipelines on consumer graphics cards in 2001, has made it easier for developers to create interactive 3D applications (Moller *et al.*, 2008). For simple and computationally undemanding ABMs, a modern Central Processing Unit (CPU) is fast enough to simulate ABMs, whereby agent behaviours can be described by only a few rules such as BOIDS (Reynolds, 1987). As ABMs become more complex the time spent processing each model tick increases, this is the designated unit of time within an agent based it advances before agent interactions are recalculated. Over the last 24 months, General Purpose Graphics Processing Unit (GPGPU) programmable technologies such as CUDA, OpenCL and DirectCompute, have shown a potential to speed up the processing of agent based models by 1.08% – 2.71% performance increase (Wang *et al.*, 2013) who implemented an ABM based on traffic flow. Richmond & Romano (2008) and Richmond et al. (2009) present a highly parallel GPU-enhanced ABM of BOIDS. Tools that support

GPGPU programming are still in their infancy and the uptake of visualisation applications making use of this emerging technology is slow. One reason given for the slow uptake is the difficulty in mapping high-level abstract data types (like those found in high-level programming languages) to the low-level floating-point data types of GPGPU shader languages (McDonnel & Elmqvist, 2009). Work has been done to lower the entry barrier for non-graphical programmers through a bespoke framework and compiler called hiCUDA (Han *et al*., 2011). GPGPU programming has been shown to be capable of processing and visualising large datasets relating to computational fluid dynamics (Kolb & Cuntz, 2005), CA based land use modelling (Gobron *et al*., 2011) and the creation of 3D models based on 2D image slices from medical imaging machines (Archirapatkave *et al*., 2011)..

Crooks *et al.* (2010) examine the advantages and disadvantages of both tightly coupled and loosely coupled ABMs, stating that *"loose coupling provides an attractive alternative, in the sense that we can create an agent based model using a specific programming library or use a dedicated simulation/modelling toolkit designed specifically for ABM and then visualise the outputs from the model in a 3D environment"* (Crooks *et al.*, 2010, *p15*).

This chapter introduces agent based modelling in the context of land use cover change, provides an overview of existing visualisation tools and frameworks for displaying the output of land use cover change and reviews the means to connect loosely coupled systems.

2.2 Land Use Modelling

Land underpins the economy through the provision of food and other goods in addition to its use for housing, business, transport, energy, tourism and recreation (Government Office for Science, 2010). The United Kingdom faces challenges in mitigating climate change whilst encouraging economic stability and growth to meet the needs of an increasing population all within its limited land and natural resources.

Land use modelling has evolved from the Von Thunen model that correctly predicted that different agricultural rings would appear around a city as shown in figure 2.1. Each agricultural ring represents a different land use type which is directly related to the distance from the city.

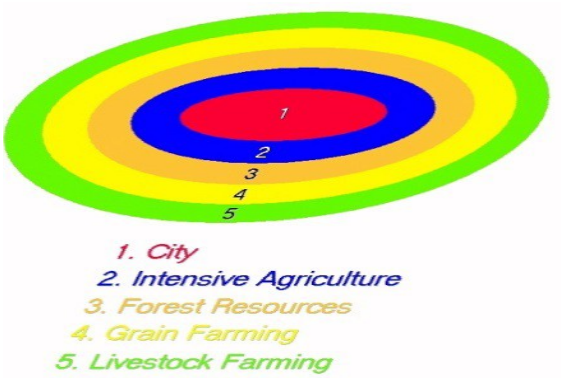


Figure 2.1: The Von Thunen agricultural model. Adapted from Sasaki & Box (2003).

This early model of land use makes some general assumptions, most notably that the central city is classed as an isolated state i.e. a state that is fully self-sufficient and is not influenced by external events (Sasaki & Box, 2003). Today, land use modelling approaches have evolved considerably taking into account global markets, societal views, environmental impact such as changes in local or regional biodiversity, production of biofuels and the effects of climate change (Rounsevell et al., 2003; 2005). Incorporating more detail such as global market and climate change models is made possible through the increasing speeds at which computers can process data and the availability of scientific models from other fields of research.

2.2.1 Complex Adaptive Systems

Complex adaptive systems have been defined as *"systems that have a large number of components, often called agents, which interact, adapt or learn"* (Holland, 2006, p.1). Mitchell (2009, p.17) defines CAS as *"[a system in which] large networks of components with no central control and simple rules of operation give rise to complex collective behaviour, sophisticated information processing and adaptation via learning or evolution."* Complex adaptive systems can be found within other complex adaptive systems.

The land use cover change of a region can be described as a complex adaptive system. Land use cover change is a broad system, encompassing other CAS that all give input to, and receive feedback from, the land use cover change and the other complex adaptive systems that can be found within land use. Examples of such subsystems include soil formation (Crawford *et al.*, 2005) and microbial growth (Falconer *et al.*, 2005; Falconer *et al*, 2008). These can all be described individually as CAS and also as part of a larger and collective system that is also complex and adaptive – behaving in non-linear and sometimes chaotic ways in such a way that emergence can be seen to form.

Nilsson & Darley (2006) compiled four key characteristics that a complex adaptive system should exhibit while Mitchell (2009) gives three similar common properties that she believes are inherent in all complex adaptive system as shown in table 2-1.

CAS Characteristics	**CAS Chatacteristics**
(Nilsson & Darley, 2006)	**(Mitchell, 2009)**
The system should consist of several agents that act in correlation with and independently of others.	All systems should consist of large networks of individual components while typically following simple rules without central control. It is the collective actions of large numbers of these components that demonstrates the complex and hard to predict patterns of behaviour.
The system should display some form of emergence that individual agents are unable to demonstrate singularly.	
The system should continually exchange information with its surrounding	All the systems produce and use information and signals from both

environments.	their internal and external environments.
The system should change its behaviour to improve its chance of survival or success through learning or evolutionary processes.	The system should have the ability to self-organise and reach a level of order between interacting systems and agents.

Table 2-1: Characteristics and properties of complex adaptive systems

CA is an example of a mathematical system constructed from many components. Each component is simple, but together forms a system capable of complex behaviour (Wolfram, 1998). Two popular examples of CA are the Game of Life (Conway, 1970) and BOIDS (Reynolds, 1987). Both the Game of Life and BOIDS follow simple rules that determine the outcome of an agent. The Game of Life is visualised on a chessboard like grid (although the grid can expand infinitely) and each cell within the grid can be one of two states, alive or dead. Should a dead cell have exactly three live neighbours (using the von Neumann neighbourhood), the dead cell becomes alive. A live cell with two or three live neighbours stays alive. In all other cases the cell will die or remain dead. The original BOIDS is based on three rules (i) Separation – the agent will never get too close to other, local agents (ii) alignment – the agent will steer towards the average heading of its local agents and (iii) cohesion – the agent will move towards the average central position of local agents. Both the Game of Life and BOIDS show that a small set of rules can be used to create fully functional CA models.

However, much of the emergent phenomena that can be witnessed within climate and weather systems, disease spread and control and biological systems remain elusive and unexplained through CA models (Mitchell, 2009). Parker (2009) points out that CA often suffers from too much emphasis on local spatial processes. In addition there is a lack of input from cell to environment whereby the environment may affect the behaviour of a cell but not vice versa.

2.2.2 Modelling Land Use Cover Change with Agent Based Models (ABMs)

Agent Based Modelling (ABM) *"allows the use of CAS approaches that can address the behaviour of each of the participants within a complex system"* (North *et al.*, 2005, p. 1196). Unlike CA, each agent in ABM has ownership over its ruleset and can communicate, receive input from and inform output to, any other agent not just neighbours in a Moore and Von Neumann neighbourhood. When agents are given multiple rulesets and less restriction over their interactions with other agents they have been shown to exhibit social behaviour similar to that found in the real world (Pavon *et al.*, 2008).

To date, much of the focus of LUCC models has been on combining crop growth models with climate change models (Pielke, 2005). The link between LUCC and climate change is well documented (Lambin *et al.*, 2001; Lambin *et al.*, 2003, Rounsevell *et al.*, 2005). Crop production is affected by meteorological variables such as increases in carbon dioxide levels and changes in regional temperature. These variables affect the type of crop that can be grown in a particular region (Parry *et al.*, 2004). Profitability and expansion of land ownership was traditionally the single largest driving factor of LUCC modelling (Rounsevell *et al.*, 2003; Abildtrup *et al.*, 2006). As LUCC modelling becomes better understood, there has been a growing concern that it lacks a description of social interaction although the environmental and economic aspects of LUCC modelling are well represented.

The importance of Societal Drivers

Understanding and forecasting future trends in relation to LUCC requires the development of models which can be used in scenario development. However, Parry & Carter (1998) argue that it is impossible to anticipate future trends in socio-economics. They claim the difficulty in predicting societal change becomes apparent when considering the human tendency to critically analyse and reflect in our surroundings and the ability to see, understand and alter our behaviour in such a way that it has an impact on the landscape. Well known systems with clear rules and easily interpretable relationships such as crop growth under particular climatic conditions have a high probability of being modelled well. In such cases it is intuitive to reduce the system to its most basic components and principles in order to more accurately predict responses. However, this does not apply to social drivers, where simplifying a home-owners perception of a landscape or a preference for a local crop type is not a mathematical system which can be deconstructed. From a modelling perspective the social aspect of LUCC models lack quantitative data and, as a result, is often ignored as having little relevance to a computer model.

Berkhout *et al.* (2002) acknowledge that the future prediction of socio-economics is far from perfect but notes that it is imperative to include socio-economic assessment. The challenge remains to build socio-economic models that are adaptable enough to model these highly uncertain drivers in the future. Audsley *et al.*, (2006) suggests that the limited number of models that take current socio-economic states into account are not reliable in the medium to long-term.

This leaves modellers and stakeholders with difficult decisions. Do they ignore socio-economic drivers because they are highly unpredictable and unquantifiable? Should they include “current day” values for social and economic drivers and accept the fact that forecasting social responses in the medium and long term is largely unknown? Should modellers and stakeholders include variable and reactive social drivers just as they do with environmental drivers? One plausible solution is to implement scenario development into LUCC models.

2.2.3 Scenario Development through ABMs

Polhill *et al.* (2001, p.34) describes a use of ABM as *“a way to provide advice to policy makers on possible land outcomes, for varying scenarios that stakeholders might want to explore. The emphasis is very much on the possibilities rather than prediction”*. Scenario development has been defined as *“one of the main tools in climate change analyses which are characterised by the assessment of future developments in complex systems, often inherently unpredictable, are insufficiently understood and have high scientific uncertainties”* (Carter *et al.*, 2001 *p.17*). The International Panel on Climate

Change (IPCC, 1994) defines scenario development as *"a coherent internally consistent and plausible description of a possible future"*. These broad definitions are explanatory of general scenario development. Borjeson (2006) linked different types of scenario "*typologies"* to different *"modes of thinking"* in an attempt to aid the planning of scenario development. These typologies can be seen in table 2-3 below.

Typology	Mode of thinking
Predictive	Knowing what will happen in the future in order to be better prepared for it. Most often this type of scenario development is achieved through well-known systems that can be described and executed through mathematical models
Eventualities	Accepting the possibility of more than one plausible outcome to a given situation. This type of scenario development is done when the processes involved in a given scenario are not well known or mathematically measured or quantified
Visionary	This is done as a collective whole. It can be a section of society, a company or organisation with the goal of making the process better for them

Table 2-2: The three typologies, as described by Dreborg (2004) and their corresponding descriptions

Following on from Dreborg's work, Borjeson *et al.* (2006) devised a guide that allows stakeholders and modellers to best develop their scenarios based upon what they want to know from the scenario and the data available. Borjeson has essentially taken Dreborg's *"modes of thinking"* and further sub-categorised them whilst using terminology and language that has become standardised within the areas of futures research. Rather than predictive, eventuality and visionary modes of thinking, Borjeson uses predictive, explorative and normative respectively to describe the different outcomes that a scenario can be developed for. Additionally, each of the three scenario typologies put forward by Borjeson is split into two sub groups as shown in figure 2.2.

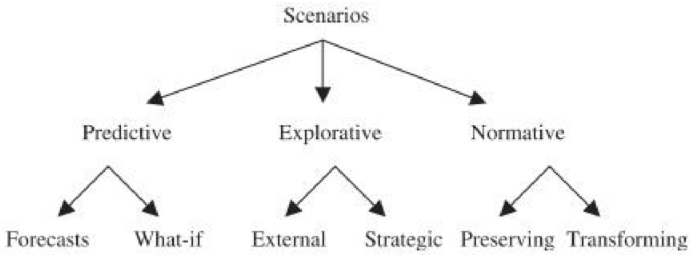


Figure 2.2: Scenario typologies with 3 categories and 6 types (adapted from Borjeson et al., 2006)

The figure above shows 3 categories (predictive, explorative and normative) with 6 types. Each is explained in more detail below.

Predictive scenario development explores the question - what will happen? This can be further sub-categorised into forecasting and what-if scenario types. Forecasting will react on what will happen with already known values - this is used for well-defined and understood systems with little or no variable input. What-if scenarios respond to the question what will happen on the condition of a specified event(s).

Explorative scenarios explore the question - what could happen? An external explorative scenario is one in which the external factors affected by the model are looked as ways of viewing plausible future outcomes. Strategic explorative scenarios respond to the question - what happens if we act in a particular way?

Finally, normative scenarios explore the question - how a particular outcome can be met? This is an if-what rather than a what-if scenario. The aim of preserving normative scenarios is to reach an optimal future target using the given starting variables and parameters. This future target can be met by altering the state of internal and/or external processes. This is contradictory to transforming normative scenarios whereby scenarios are strictly prohibited from altering any internal or external processes. The starting variables and parameters must suffice when attempting to simulate a desired future.

The type of scenario depends upon the data available and the desired output of the model. Once these aspects have been decided upon it is possible to plan how the V.E should look.

2.3 Land Use Visualisation

2.3.1 A History of Map Making

GIS evolved from the tradition of map making (Weigand, 2003). Before digital mapping was available the most popular method of map and geospatial analysis was done by creating thematic paper-based maps with changeable polyester film which was used to overlay various topographic areas of interest such as points and boundaries (GCS Research Society, 2001). These replaceable polyester sheets were known as map layers. However, traditional paper-based maps lack the strengths of digital mapping such as cost saving, improved communication and superior spatial analysis (ESRI, 2009). Modern technology, particularly the rise of the World Wide Web, has provided opportunities for the dynamic presentation and user interaction of map data, something that cannot be realised with physical maps (Kraak, 2004). Modern GIS takes the idea of map layers (made possible through the use of shapefiles) and other visualisation methods such as Triangulated Irregular Networks (TINs) and heightmaps (see **chapter 4.4**).

The first commercially available software for GIS was released in the 1985 (ESRI, 1985; James *et al.*, 2004). The availability, affordability and popularity of web-enabled GIS switched the user base of GIS tools from academic and industry experts to the general public with an Internet connection and an interest in maps (Miller, 2006). Free software including Google Earth, Google Maps, Microsoft’s Virtual Earth and NASA’s WorldWind are GIS-based and are capable of visualising land use in 2D and 3D.

2.3.2 Land Use Visualisation through GIS and ABMs

GIS has been extensively used as a visual medium to demonstrate the effects of land use cover change (Belaid, 2003; Coors, 2003; Latu, 2009). This is mainly due to the flexibility of modern GIS software packages, the ease of which digital maps can be added to or amended and the ability to overlay spatial analysis with commonly used shapefiles [see **figure 2.3**].

Shapefiles are a popular geospatial data format. ESRI, the makers of ArcView (a popular GIS package) created the original shapefile data format. File and data interoperability is an important issue surrounding GIS and the open whitepaper of the shapefile specification (ESRI, 1998) allows developers the freedom to implement shapefiles as they see fit (see **chapter 4.4.4.4**).

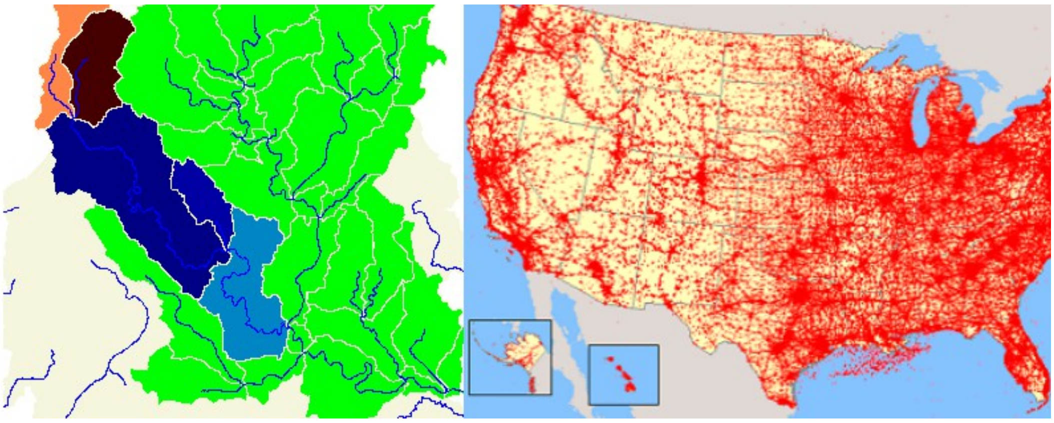


Figure 2.3: Left: A land use visualisation that utilises a shapefile overlay showing rivers in Kentucky, US. Right: A shapefile of wireless hotspots in North America.

Shapefiles prove useful when the user wishes to overlay multiple sources of data onto a map however they are not the only 2D visualisation technique used to represent land use or land use chage. As pointed out in **section 2.2.2**, cells are often used to visualise spatial relationships within a CA model. This strategy is applied to LUCC ABMs whereby cells are used to represent farms, localities, regions and larger areas as can be seen in figures 2.4 and 2.5 below.

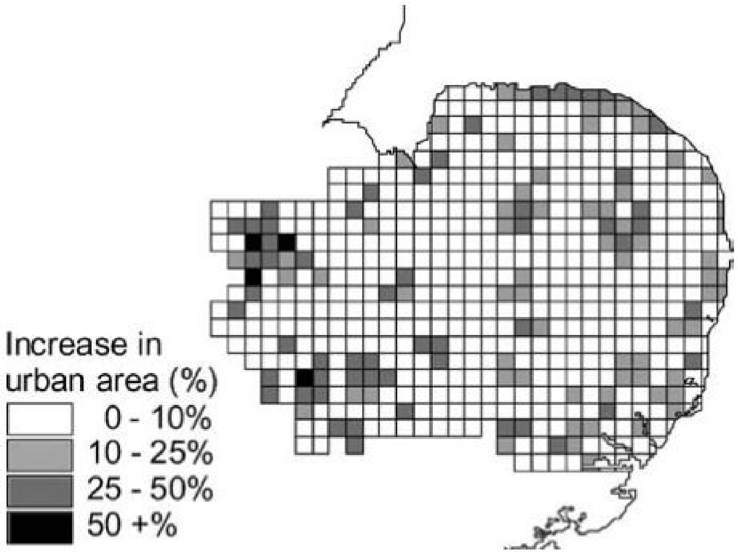


Figure 2.4: Urbanisation rate of East Anglia as a 2D cell visualisation (from Holman et al., 2005)

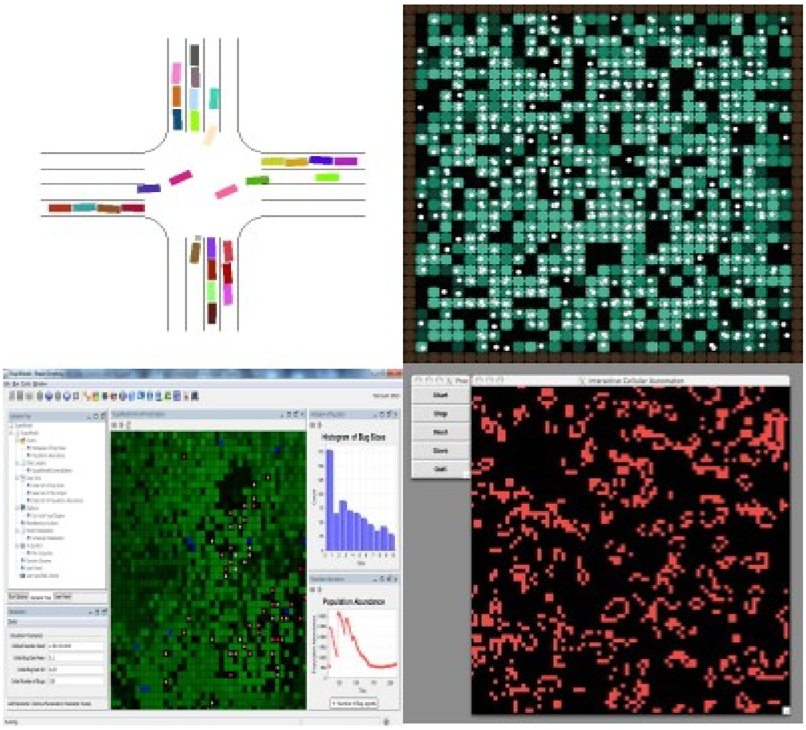


Figure 2.5 Top-left: Output of a traffic model (Wilkie, 2009). Top-right: a 2D visual representation of a hydrogen diffusion ABM (Stonedahl et al., 2010). Bottom-left: A population mapping ABM in Repast Simphony's 2D display. Bottom-Right: An example of the Game of Life

According to Crooks *et al.* (2008, p42) *"visualisation plays a key role in communicating and sharing the output of agent based model with those that it might wish to influence."* The success of an agent based model is measured, partly, by how well the visual output enables stakeholders of the system to interact with the model and participate in scenario development.

Current Technical Limitations

The 2D displays of ABM toolkits suffer from a lack of graphical acceleration (Crooks *et al.*, 2009; Macal & North, 2010). Current ABM toolkits offer limited 2D support for drawing visual output such as lines, points, polygons and 2D textures. Three of the most popular toolkits; MASON, Repast Simphony and NetLOGO all permit developers to link to external 3D graphics libraries including Java3D and OpenGL. However, to do this requires additional resources, namely graphical programmers and 3D artists, when building the model as the ABM toolkits do not implement any of the “click and drag” graphical options it offers for its 2D visualisation. Features such as loading and rendering shapefiles, representing agents as 3D models and user interactivity within the model are left to the developer to implement as they see fit.

3D Virtual Environments (VE)

The literature is split over the benefits 3D visualisation holds over traditional or 2D visualisation. There is evidence that participants of comparative 2D/3D task based identification respond quicker when identifying 3D shapes when compared to 2D shapes (Ting *et al.*, 2011) but this advantage quickly deteriorates when complexity or uncertainty is introduced such as occlusion of parts of the shape – at which point participants were seen to correctly identify 2D shapes quicker than their 3D counterparts. This confirms the findings of Sebrechts (1999) who tested the speed of participants interacting with text, 2D and 3D systems under similar conditions. There is evidence to suggest that the participants have more to do with the results of such experiments with Keehner (2004) suggesting that the spatial ability of the participant is a key driver into how effective the user can interpret and navigate within a 3D environment. Even this theory has disagreeing voices as to whether a participant with low spatial ability benefits more or less than a participant with high spatial ability (Huk, 2006). When considering the experiments carried out by Sebrechts and Ting it becomes apparent that they are using the visualisation tools as an interactive measuring tool, testing the accuracy and speed of a participant. However, not all visualisation tools are built to capture and analyse user input and the benefits of 3D visualisation over 2D visualisation can be found in other uses of 3D visualisation such as participatory planning.

Participatory planning is the planning solution that has the local/regional community at the heart of the decision making and management process. The aim is to get stakeholders, non-expert and expert, to get together and discuss how they would like to change a particular area through discussion and knowledge transfer. This type of planning is becoming increasingly popular with governments, construction companies and transport authorities as involving the local community of any landscape changes helps to identify potential problems and sources of conflict between parties before they actually happen. Examples of 3D participatory planning can be found in highway management (Bailey *et al.*, 2002), urban planning (Isaacs, 2011) and forest management (Wissen *et al.*, 2008). Bailey found the greatest benefit of 3D participatory planning tools to be the ability to quickly and easily display various possible designs to determine which designs are preferred. Wissen speaks of the benefit of being able to visually link temporal and spatial data of forested areas and move between them with ease and both Wissen and Isaacs speak of the improved communication between participants when using 3D visualisation tools within public participatory planning.

Bishop *et al.* (2009) gives an example of a 3D virtual environment that aids stakeholder decision making. The level of interactivity falls short of real-time interaction. Instead, the visualisation allows viewing a scene from a variety of pre-programmed camera angles with buttons to switch between each one as shown in figure 2.6.

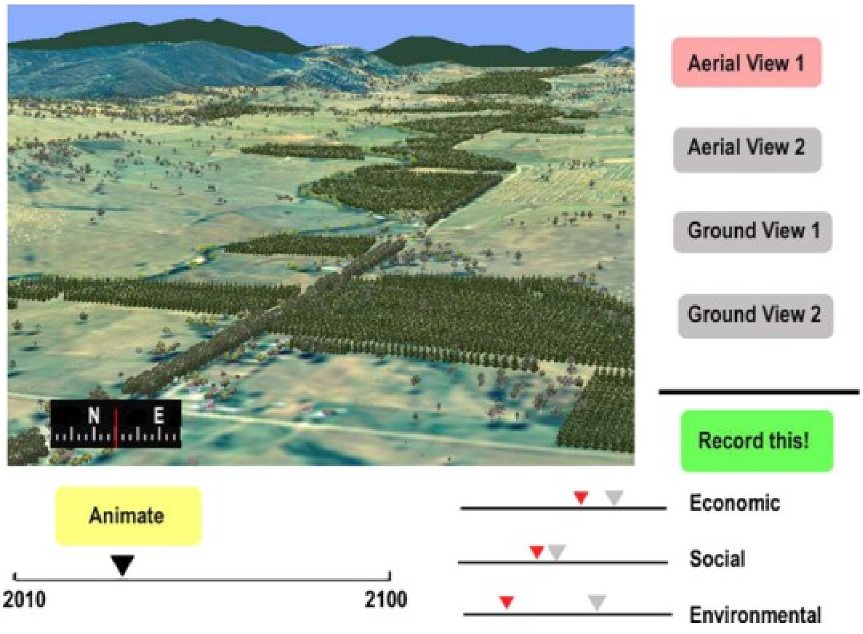


Figure 2.6: An example of a static 3D land use visualisation tool (from Bishop 2009).

Miller *et al.* (2009) present the "VisuLands" project carried out public participatory planning through the use of a Virtual Landscape Theatre (Miller *et al.*, 2009 p.319) which is a large continuous screen 5.5m wide and 2.25m tall with a screen curvature of 160 degrees allowing many participants to take part in participation sessions and visual preference surveys. One of the VisuLands case studies was of Clashindarroch forest, an area in the North-East of Scotland. Through consultation with land management stakeholders as well as the general public three plausible future scenarios for the Clashindarroch forest were created for visualisation within the Virtual Landscape Theatre. The scenarios were shown to participants as a non-interactive render with the 3D scene moving along a predetermined path. The authors conclude the presentation of their research by discussing possible future work to improve upon the visualisation tool. These areas include adapting visualisation tools to enable users to switch between scenarios with particular regard to the timescale over which landscape changes take place. The authors also state the need for transparent assessments of the effectiveness of visualisations to ensure visualisation tools "are relevant, accessible and offer meaningful information for a decision maker (Miller *et al.*, 2009 p.340).

The creation of virtual environments has been achievable for some time but concern has been raised over the insufficient representation of plants and habitats within 3D virtual environments (Paar, 2006). Software such as Visual Studio Nature can generate photo-realistic images using Digital

Elevation Models and shapefiles (Appleton *et al*., 2002). Visual Nature Studio is capable of rendering user-created 3D models that are representative of various types of vegetation such as trees, bushes and grass. There is evidence to suggest that users of Visual Nature Studio have some performance concerns due to the computational requirements of rendering many 3D models such as grass and trees (Dockerty *et al*., 2005) Recent versions of Visual Nature Studio offer some optimisations for drawing thousands of 3D models by implementing hardware instancing but these optimisation still have issues drawing tens of thousands of models (Pescarin *et al*., 2008).

Other non-interactive forms of 3D output of agent based models can be found in the film and video industry. Disney's The Lion King makes use of an agent based model for the stampede event (Reynolds, 2010). Both iRobot and The Lord of the Rings used commercial software for visualising agent based model output in 3D that was developed for large crowd related visual effects [see figure 2.7] and autonomous character animation (Massive, 2011).

Figure 2.7: Top: Multiple autonomous movements being carried out by agents in iRobot. Bottom: Large crowd simulation of multiple agents in The Lord of the Rings

The 3D output of the Massive ABMs lack two features that are important to stakeholder engagement. Firstly, the models are not supported by any scientific body of work and secondly there is no opportunity for stakeholders to interact dynamically with the rendered model as it is not rendered in real-time.

Visualisation Tools & Stakeholder Engagement

Stakeholder engagement when used in conjunction with visualisation tools can result in diverse perspectives which improve the validity of stakeholder assessment (Gibson & Hassan*,* 2005). Whitmarsh *et al.* (2009) believe the process of stakeholder engagement to be a network of social actors with divergent interests and expertise working together to develop sustainable, plausible futures through scenarios development. The use of visualisation tools as part of stakeholder engagement can aid the development of solutions to problems when no complete knowledge of the problem exists. Computer models based upon incomplete knowledge, such as land use cover change, climate change and flooding have limitations and uncertainty that can be further improved upon using knowledge and input from interested stakeholders (Bohunovsky *et al.*, 2010).

Land use change models have a broad target audience of potential stakeholders such as policy makers, social scientists, land managers and the general public. To do this requires the linking and coupling of ABM and visualisation engine.

2.4 Coupling Agent Based Modelling and Visualisation

There are two methods of connecting an ABM with a visual front-end. These methods are known as (i) tight coupling and (ii) loose coupling. Both methods are explained in detail below and examples are given of visualisation of ABM output using both methods. The problems and difficulties of each solution are also discussed.

2.4.1 Tight Coupling

Tight coupling is used to describe an agent based model that simulates and visualises on the same platform from the same code base. The model and visualisation are compiled, built and executed together. This is difficult to achieve using ABM toolkits such as Repast Simphony, NetLOGO or MASON. There is little support for advanced graphical rendering that is optimised for the GPU which forces all model computation and graphical rendering onto the CPU. The CPU pipeline bottlenecks when using even simple lines and points for ABM output of a non-trivial model (Crooks *et al.*, 2010). One solution suggested by Vizzari *et al.* (2008) is to move the agent based model away from ABM toolkits and implement them in a language or framework more suitable for 3D graphical development. Vizzari uses the C++ library of the Irrlicht Engine – an open source game engine, and ports a pedestrian crowd ABM into C++ as seen in figure 2.8.

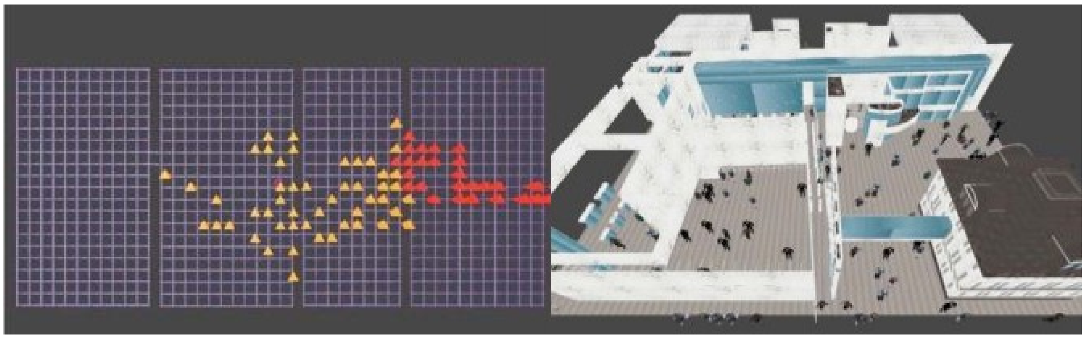


Figure 2.8: Left: A top down 2D view of a crowd evacuation model within a museum. Right: A 3D view of the same crowd evacuation model (from Vazzari et al., 2008)

The use of the Irrlicht engine allows generation of 3D scenes. The difference in visual representation of model output can be seen by comparing the images on the left and right of figure 2.8. The right image shows visual elements of physical boundaries rendering walls and the representative nature of the agents immediately identifies them as people. These visual elements are not restricted by the platform on which they are modelled like ABM toolkits - attempting to create a similar scene in Repast Simphony's Java would be difficult. The close coupling method offers a flexible approach to how the visual output can be rendered. There are disadvantages to this approach however. The crowd behaviour ABM required porting (implementation in another language/platform) from Java into C++. There is also no interaction between the user and the visualisation. The user watches the output of the model on their monitor.

3D coupled ABMs are not restricted to being created within a programming language environment such as Java or C++. Hudson-Smith (2009) and Crooks *et al.* (2009, 2010) explored the possibility of using a 3D modelling package to create tightly-coupled agent based models. Both Hudson-Smith and Crooks use 3D Studio Max, a popular modelling package used by computer artists, to create 3D output of agent based models. Hudson-Smith theorised that it was possible to create a 3D ABM using a 3D modelling package with highly realistic models and a basic in-built scripting language.

Hudson-Smith's example in figure 2.9 shows a 3D scene containing 4 pillars and 8 red blocks. The blocks are agents, following a simple move and evade model – each block moves but stops if it is going to hit another agent or part of the landscape. The ABM is implemented in MaxScript, a scripting language found in 3D Studio Max.

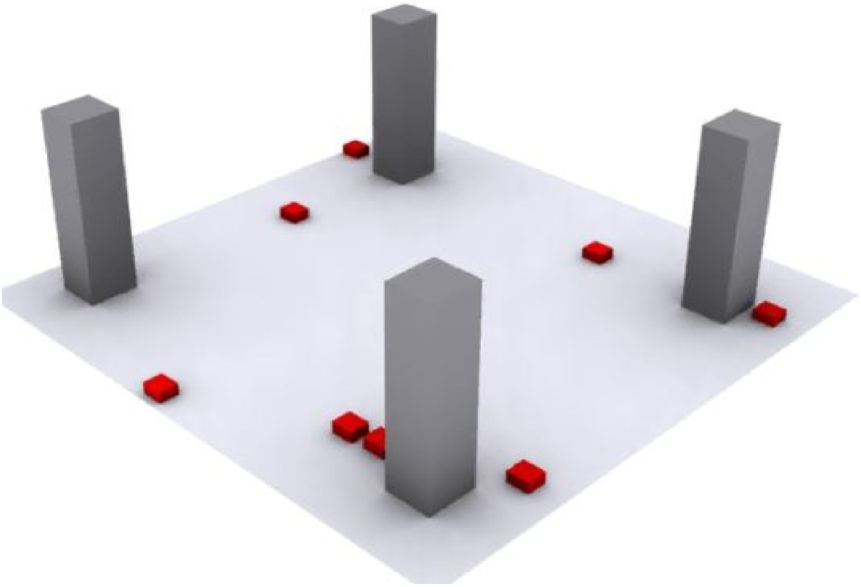


Figure 2.9: A move-and-evade simulation. The agents (red blocks) interact with its surrounding environment using basic scripting commands (from Hudson-Smith, 2009).

This method has a potential visual advantage over the previous method of using a game engine in that there are no restrictions on the number of polygons that make up a scene. This is because the output that is shown to the user is pre-rendered and has already been turned into a video file ready for viewing.

2.4.2 Loose Coupling

Loose coupling is the act of encapsulating an ABM and a visualisation engine as separate entities with the visualisation engine being controlled through a series of messages or data transfer. Crooks defines loose coupling as *“the linkage of two stand-alone systems by data transfer”* (Crooks *et al.*, 2010 *p.13*). Crooks suggest that close coupling enforces a lack of communication of the states of the ABM, calling it black boxed agent behaviour due to the lack of visibility of the internal workings of the model. Crooks provides a modified version of Wilensky’s traffic model (Wilensky, 2003), the modification being that key agent data is stored in a text file on a per-tick basis. His example also incorporates the work of Hudson-Smith’s (2009) 3D Studio Max method discussed above. The saved text file includes information on the location, speed and direction of each car within the ABM. These values are later processed by MaxScript in 3D Studio Max. MaxScript then updates the 3D models using the information initially provided by the ABM ensuring that the 3D models behave and render as informed by the ABM. This example suffers from the same lack of interactivity as the first 3D studio max example. There is no way for the user to view or change the parameters of the model while it is running and the world is a static world in which the camera does not respond to user control.

Another example of a loosely coupled model was presented by Merrick & Maher (2007). The underlying model presented by them is not agent based but is Motivated Reinforcement Learning (MRL) which is an artificial intelligence model aimed at understanding how machines can develop new skills without being explicitly taught by a human programmer. Their MRL model was used in conjunction with Second Life, a popular 3D online virtual world that allows users to customise many aspects of the world they, and others, see and interact with. The purpose of Merrick and Maher's model was to control artificial agents within Second Life that would attempt to herd sheep into a particular pen. Second Life allows users to create scripts that can access other Web Services using Extensible Markup Language Remote Procedure Calls (XML RPC) which is used as the messaging format for the virtual world. Merrick & Maher used XML RPC to create a bi-directional communication protocol that would update Second Life with model data then send Second Life data back to the model as shown in figure 2.10 below.

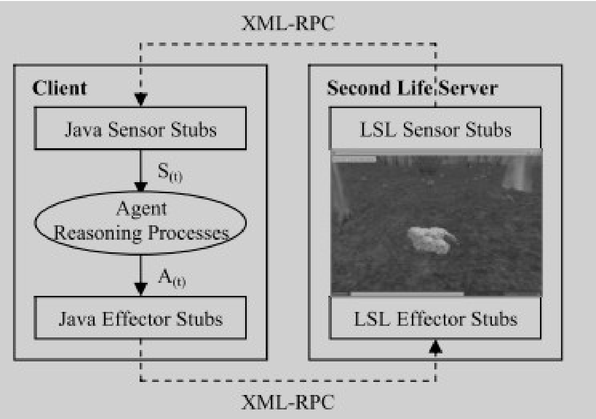


Figure 2.10: A diagram showing the input/output of XML RPC between Client (MRL model) and Server (Second Life) (from Merrick & Maher, 2007)

This implementation shows that bi-directional communication is possible between model and visualisation however there are technical problems that should be noted. Second Life is a Virtual Environment with many thousands of users, some of whom are paying subscribers. As such, there are technical limitations imposed on users to prevent abuse of the VE having a negative effect on other users within the environment. There is a message limit of 256 characters and, as XML is a human readable data format, each message is quite verbose which limits the amount of information that can be sent per tick. There is also a delay of 3 – 5 seconds per XML RPC call which slows the visualisation down considerably. If you consider each model tick to be one day a user would be required to watch an 18 minute simulation to see the model simulate one year. The authors suggest that, over time, Virtual Worlds will become more open to its users and the different uses of Virtual Worlds but it has not yet happened.

2.5 Testing the Effectiveness of Visualisation Tools

2.5.1 Defining Visualisation Evaluation

Wikj (2005) notes that a visualisation should not be considered good or an improvement over non-visual information simply because it exists. An often cited reason for developing a visualisation is that the data being visualised cannot be extracted automatically in its native format or environment or, if it can, the complexity of the data makes understanding it difficult.

The testing requirements of any visualisation application are dependent on the purpose of the visualisation (North, 2006). However a broad and agreed upon goal of visualisation is to provide insight and therefore visualisation evaluation should determine to what degree the visualisation achieves this insight (Freitas *et al.*, 2002; Saraiya *et al.*, 2005; Wijk, 2005; Chang & Ziemkiewicz, 2009) whilst allowing users of the visualisation to interact with, and understand, large and complex data (Wiss *et al.*, 1998; North, 2006).

Insight is difficult to define but North gives the following characteristics [table 2.4 below] for recognising insight.

Characteristic	**Definition**
Complex	Insight should involve large amounts of data
Deep	Insight should build over time, accumulating and often garnering further questions and discussion
Qualitative	Insight if often uncertain with subjective views and are rarely exact
Unexpected	Insight is often unpredictable
Relevant	Insight should be embedded within the data and the knowledge domain

Table 2-3: The characteristics that insight exhibits (adapted from North, 2006)

North also detailed the high number of visualisation usability studies that were simply Boolean based – answering the question of whether a user liked a visualisation or not, or if the user preferred one visualisation to another. Boolean based questions are inadequate when determining a visualisation's fitness for purpose. Additional testing strategies for the evaluation of visualisation are required.

2.5.2 Semiotics and Information Visualisation

Any communication through the medium of visual imagery whether it is a map, drawing, painting, sign, symbol, animation or 3D virtual environment takes place on a two dimensional surface such as a piece of paper, a computer monitor or a television screen (Tufte, 1991). This can be challenging for many types of data - for example, how do you visually represent multivariate data on a two dimensional screen? The science of semiotics, which is the study of meaning-making, can help form guidelines for creating useful visual imagery. When considering the design and creation of visual imagery there are three principles that should be thought about beforehand (Shanks & Darke, 1998). These are:

- What form does the symbol take with regard to shape, colour, form and words? This is known as the **syntactic semiotic**.
- What meaning does the symbol denote? This is known as the **semantic semiotics**.
- How will the symbol be used? This is known as the **pragmatic semiotics**.

For example, if a traffic light sign was examined using the three principles above the outcome would be:

Syntactic Semiotic

The sign is rectangular in shape with a height/width ratio of about 3:1. It contains 3 differently coloured circular lights containing red, amber, green from top to bottom on the rectangular sign.

The sign operates by changing colour. It starts with the red colour being lit up while the other two colours remain off – it remains in this state for an 60 seconds. It then lights up the amber light, while the red remains on – this state lasts for about one second. Next, it turns off the red and amber lights while lighting up the green light – this lasts for 30 seconds. The green light then turns off whilst the amber light turns on – this lasts for one second. Finally, the amber light is turned off and the red light is turned on – this lasts for 60 seconds and is back to the original setting. This procedure loops indefinitely.

Semantic Semiotics

The meaning of the various states a traffic light can be in is determined by laws that govern our roads (Road Traffic Act, 1988). A red light indicates that the road user should stop before the white “STOP” line. A green light indicates it is safe to continue through the traffic lights. An amber light means “prepare to stop” and, as such, a road user should not cross the “STOP” line unless they are already over the line when the light changes to amber or if doing so would likely result in a collision.

Pragmatic Semiotics

All road users have an obligation to have read and understood the Road Traffic Act. As such, the behaviour of every road user should then be the same. All road users should stop at a red light, only drive through green lights and correctly identify the few occasions when driving through an amber light is acceptable.

If all traffic lights have the same syntactic representation and the same semantic implications then why do some road users go through red lights? This is because pragmatic semiotics is about how the user behaves with the information they have taken from the sign and not everyone acts the same way with the same information. Charles Sanders Peirce, known for his early work on semiotics, categorised signs into different “modes” by examining the relationship of a signs pragmatic, semantic and syntactic semiotics. The three modes of sign, according to Peirce (1958) are:

Symbolic Signs

Symbolic signs are arbitrarily assigned or accepted as part of societal or cultural convention (Lanir, 2012). Letters of the alphabet, mathematical signs, the periodic table and national flags are all examples of symbolic signs. The relationship between the signs form (the syntactic semiotics) and the signs meaning (the semantic semiotics) must be learned before it can be of use just like children learn multiplication tables, letters of the alphabet and the colour, form and makeup of country flags. The user of a symbolic sign is required to know (or learn) the meanings behind the sign.

Iconic Signs

The syntactic semiotics of an iconic sign contains one or more characteristics or qualities of its signified object. For example, a portrait of yourself would have several characteristics that are similar to the real you. The colour of hair would be similar, as would the general form and shape of your face and facial features. Anyone looking at the portrait would know it was a representation of you. Often iconic signs require no learning as the sign should be self-evident as to its representation.

Indexical Signs

Indexical signs are signs that may not resemble the object it is trying to convey although it is still directly connected to the object in some way although sometimes inferred. For example, seeing smoke would infer that there is fire just as hearing thunder would indicate that there is lightning. Indexical signs may or may not require learning depending on the circumstances but indexical signs are considered to be easier to understand and more commonly used (Lanir, 2012).

The mode of a sign can determine who to show a particular sign to. Expert stakeholders, those with a deep understanding of their subject, may prefer symbolic signs while an average lay-person will make quicker sense and understanding out of iconic or indexical signs.

2.5.3 Strategies of Visualisation Evaluation

2.5.3.1 Comparative 2D/3D

Cockburn & McKenzie (2002) adapted a document management system called Data Mountain (Robertson *et al.*, 1991) from 2D into 3D for comparative analysis. The authors collected quantitative results by asking users of the system to complete task based experiments such as storing and retrieving documents and timing the length of time it takes to complete each task on the 2D and 3D systems. They also carried out qualitative analysis on the participants, asking them which system they preferred and which system they felt was easiest to complete each task. The authors noted that gathering the qualitative feedback from participants was invaluable as the quantitative results showed the 3D system had a higher time-to-complete than that of its 2D counterpart while the qualitative feedback was overwhelmingly in favour of the 3D system.

This type of comparative visualisation evaluation is often carried out using a Visual Preference Survey (see **chapter 6.2**) to determine which images or scenes the user prefers. According to LaGro (2011) "*statistical analysis of visual preference survey ratings can consider means, medians and variance which can help to explain preferences and difference across scenes*". These methods of analysis can be found in work carried out by Steinitz (1990), Bailey *et al.* (2001) while Ewing (2001) uses bi-nomial logit analysis and multiple regression analysis.

2.5.3.2 Naming Time

Another visualisation evaluation strategy is to use naming time (Watson & Friedman, 2000; Watson *et al.*, 2001) which has its roots in cognitive psychology. Naming time, in Watson's example, involves showing participants a series of models with various levels of polygon simplification. The participants were shown three images of each 3D model, the first with no polygon simplification, the second with 50% polygon reduction and the third with 80% polygon simplification. When the participants were shown each image they were asked to say the name of the object as soon as the recognised it and the time difference in answers are recorded.

2.5.3.3 Electronic Voting

Electronic voting or polling is becoming increasingly popular when visualisation evaluation is carried out with large groups of participants. Electronic voting can be found in a range of scientific fields including landscape management (Miller *et al.*, 2009; Southern, 2010) and highway planning (Bailey *et al.*, 2001). Electronic voting involves asking questions to a large group of participants who respond by pressing a corresponding button. Often the questions asked have multiple choice answers or are rated on a Likert scale. Electronic voting is usually carried out in conjunction with comparative analysis of 2D/2D, 2D/3D or 3D/3D scenes.

2.6 Conclusions

This chapter has **demonstrated that real-time interactive visualisation has the potential** to supports stakeholder interaction and engagement and examples of its use in urban planning and landscape and rural planning are presented. In **section 2.1** an introduction into two land use modelling paradigms are given; CA and ABM. The section concludes with a review of the different typologies of scenario development (Predictive, Eventualities & Visionary) found in CA and ABM.

The current state of land use visualisation through existing GIS and ABMs was presented in **Section 2.2**. The computational limits of graphical rendering within ABMs were shown to be CPU-bound suggesting a need to move graphical rendering to the GPU, something that ABM toolkits lack. These CPU limitations are explored in greater detail in **Chapter 5 – Evaluation of Network Communications**.

In **section 2.3** 3D ABMs are discussed in addition to the differences between tight and loose coupled ABMs. The technical issues of both tightly and loosely coupled ABMs is shown by highlighting the time consuming nature of porting existing models to another programming language; the non-interactivity that 3D modelling packages enforces and the slow response time of Internet messages within current VE all prove problematic in creating interactive 3D agent based models.

Finally, a discussion of the strategies to test the effectiveness of visualisations is presented in **section 2.4** where it is shown that both qualitative and quantitative data should be gathered to fully understand stakeholder engagement. This is further examined and discussed in **Chapter 6 – Testing Strategy**.

Chapter 3 – Case Study

This chapter introduces the R.S.V.T case study which is based on the Lunan Catchment in the North-East of Scotland and provides a summary of the models and data required to drive the design, testing and implementation of the visualisation system. The case study forms part of a larger E*coChange* project (EcoChange, 2008). The chapter presents (i) the Lunan catchment and its associated ecosystem services which are included in the EcoChange ABM, and (ii) the stakeholders and storylines used to create and explore plausible future scenarios. These informed the development (see **chapter 4** and **chapter 5**) and testing (see **chapter 6**) of R.S.V.T.

3.1 The Lunan Catchment

The Lunan Catchment is an area of land in Angus in the North-East of Scotland covering 132km$_2$ with 115 farmers and land owners managing the area (Guillem & Barnes, 2013). The Lunan Water is an easterly flowing river that runs through the area - discharging into the North Sea approximately 7 miles north of Arbroath. The Lunan catchment is rural in nature and is located close to three small towns (population 7.000 – 20,000); Arbroath to its South, Brechin to its North-West and Montrose and North-East [see figure 3.1].

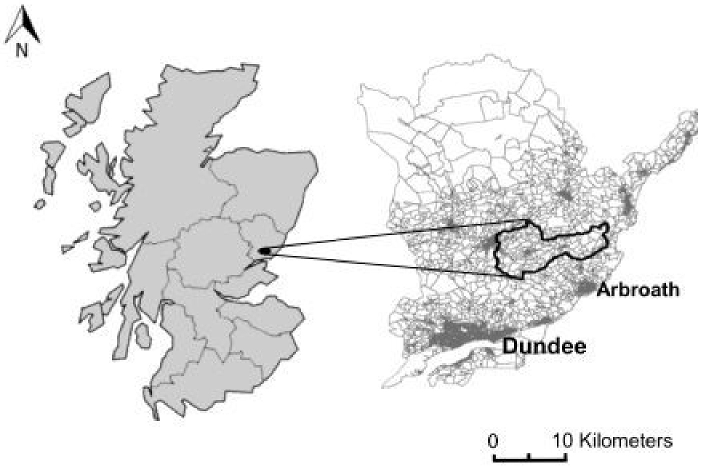


Figure 3.1: The Lunan catchment and its surrounding area in North-East Scotland (Adapted from Guillem, 2012)

The farming systems found in the Lunan catchment are general cropping (40%), mixed farming (29%) and cereals (10%) with less than 4% of the total area set aside for grazing (Guillem, 2012). This makes the Lunan catchment intensively cropped and consequently environmental issues such as water quality (SEPA, 2009; Vinton *et al.*, 2010; Guillem 2012) and regional biodiversity (RSPB, 2010, 2011; Guillem, 2012) are important drivers when considering the sustainability of the region.

3.2 The EcoChange Project

The Lunan catchment is one of the regional case studies of the EcoChange project. The EcoChange project is a European Commission funded project that uses advanced modelling approaches to assess impact of global change on biodiversity and ecosystems. The EcoChange project has 5 project objectives:

1) To improve the current data on biodiversity, climate, land use and socio-economic information
2) To improve current models and test critical model assumptions to cope with forecasting
3) Integrate prediction uncertainties into forecasting
4) To test the predictions of global change upon biodiversity and ecosystems
5) To develop a series of future projection of biodiversity, ecosystem functions and the goods and services they provide based on climate and land use scenarios both at EU and case study scales

The work presented in this thesis contributes to point 5 of the EcoChange project. R.S.V.T is designed to communicate future projections and storylines to stakeholders through the visualisation of selected indicators relating to regional biodiversity, social acceptability and economic performance.Guillem (2012) addresses points one – four of the EcoChange project by the creation of the underlying ABM that drives R.S.V.T. The task of collecting and digitizing regional data as well as creating the Lunan agent based model was part of a doctoral programme at the University of Edinburgh's GeoScience Department. This work required the definition of storylines, defining a typology for each land owner using data gathered via questionnaires and telephone interviews and the design and implementation of an ABM which includes modelling farmer decisions and skylark behaviour (Guillem, 2012). Each of these topics is discussed below.

3.3 EcoChange Storylines

Storylines are not predictions of future events rather they describe how the future might unfold (Bohunovsky *et al.*, 2010). The EcoChange projects implements three storylines – BAMBU, GRAS and SEDG. These storylines are based on the Inter-Governmental Panel on Climate Change – Special Report Emissions Scenarios (IPCC-SRES) report (Nakicenovic *et al.*, 2000). The storylines, which were originally designed for national and global agent based models, are scaled down to regional levels to fit with the spatial resolution of the regional case studies commissioned through the EcoChange project. A description of the three storylines used in the EcoChange project (and the Lunan agent based model) is given below:

BAMBU (**Business-As-Might-Be-Usual**)

This storyline defines agriculture as being specialized on a limited number of quality regional products. In the case of the Lunan catchment the products are carrots, peas, potatoes and cereals. Any additional regional needs are imported from abroad. It attempts to encapsulate the current economic, environmental and social attitudes found today and continues them into the future.

GRAS (**Growth Applied Strategy**)

The regional population is heavily influenced by personal goals of greater income, consumption and material wealth. Intensification will dominate the agriculture sector and environmental problems are only dealt with when they interfere with these goals.

SEDG (**Sustainable European Development Goal**)

This storyline assumes intense lifestyles changes. The population aims to create an environmentally stable way of life and does so at the expense of material wealth. The storyline assumes residents have a low “environmental footprint” which calls for low levels of all resource consumption. The agriculture sector will grow as will biodiversity.

3.4 Farmer Typology of the Lunan Catchment

Farmers within the Lunan catchment are assigned one of three possible types that broadly define their personality. These typologies were created by carrying out telephone interviews and questionnaires with Lunan farmers and analysing census data of the Lunan catchment (Guillem, 2012; Guillem & Barnes, 2013). From a random selection of 90 farmers within the Lunan catchment there was a 51% (46) response rate. The telephone questionnaire and the follow up questionnaires were reviewed by agricultural consultants working within the Lunan catchment and, broadly, covered four sections.

1) Socio-demographics questions
2) Attitudes towards farming and bird habitats
3) Objectives in farming
4) Management intention

Each farmer within the region is assigned a typology based on their answers. A description of each typology is given below.

Profit-Oriented

Farmers assigned this type are driven to maximize profits with no strong values of environmental or social issues.

Multi-functionalist

Farmers classified as multi-functionalist have good awareness of environmental issues surrounding the area. They were the only group to mention the decline of bird numbers in the region. They are more likely to provide suitable habitat for biodiversity.

Traditionalist

Traditionalist farmer types have an awareness of the environmental values but are more concerned with the social impact of their decisions. Farmers who fall under the traditionalist typology are more concerned with the aesthetic value of the landscape – this does not always have a positive environmental impact on the regional biodiversity.

The three storylines mentioned in **section 3.3** determine how the farmer typology of the region will be allocated. Under the **SEDG** scenario, users of the system could expect to see an increase in the number of multi-functionalist and traditionalist farmers whilst seeing a decrease in the number of profit-oriented farmers. As a result the model (and subsequently R.S.V.T) would render a scene that is likely to contain a large number of spring-based crops and have a later scheduled cutting of set-asides due to the storyline generating farmer typologies with a greater than average weighting towards environmental sustainability. Conversely, running a model simulation through the **GRAS** scenario would allocate the profit-oriented typology to a larger amount of land owners. As these types of land owners are only interested in maximising profit there is little weighting for environmental protection meaning the user of the visualisation would see a scene with higher than average numbers of winter-grown crops, which make land owners more money but has an impact on the local skylark population due to the time of year the crops are planted. While the ABM comes with these three pre-defined scenarios, each scenario can be further customised to better fit any collaborative scenario development that may arise from stakeholder meetings such as local and regional policy implications or a customised make-up of farmer typologies.

3.5 Lunan ABM BioDiversity and EcoSystem Services

The Lunan agent based model contains 3 sub-models that, when combined, calculates a number of social, economic and environmental indicators.

These 3 sub-models are:

1) A skylark Individual Based Model

2) A farmer decision making model named Aporia (Murray-Rust *et al.*, 2011; Guillem *et al.*, 2011)

3) A sub model that determines how much energy is produced by food and biofuel crops.

3.5.1 Skylark Population

The decline of the UK's farmland birds over the last half decade has been well documented (Fuller *et al.*, 1995; RSPB, 2010). There is also evidence to suggest that other habitats are not as harshly affected as farmland birds. The sharp decrease in farmland birds coincides with many of the changes to UK agricultural practices and increased intensification (Fuller, 2000, Donal *et al.*, 2001). Many farmland bird species have been declining but skylarks (*alauda arvensis*) have suffered a 55% drop in population between 1975 and 1994 (Browne *et al.*, 2000) and a recorded 61% population drop between 1967 and 2008 (BTO, 2010) and as such have been given red status to represent they are globally threatened. While it is difficult to say with complete certainty what caused this sharp decline there has been evidence that a switch from spring to autumn cereal crops (Browne *et al.*, 2000; RSPB, 2008; Taylor *et al.*, 2010) as well as intensification of grassland (RSPB, 2008) has been at the heart of the decline. The individual skylark model is designed to represent the breeding behaviours of skylarks found within the Lunan catchment. Generally, skylarks nest from early April until the end of July and can breed as much as four times in one year. Skylarks nest within crops roughly 20-50cm high. The availability and diversity of crops is generated by the farmer's typology and the scenario storyline being run (see **section 3.3** and **section 3.4** respectively). Upon entering a breeding period, the male skylark will scan approximately 500 metres of diameter and assess the surrounding landscape for nesting possibilities. If a suitable nesting site is identified by the male skylark a nest is settled at the location and the male skylark attempts to find a mate. The nest remains in place until either the skylark or his mate dies or the land use changes to an unsuitable nesting area at which point both skylarks look for another site or become “floaters” which are defined as non-reproductive flock of birds (Guillem, 2012).

3.5.2 Farmer Decision Making

The decision farmers make with regard to their land has a direct impact on the skylark population within the Lunan catchment. Skylarks, like only a handful of birds species, nest on the ground in vegetation that is 20-50cm high and will only do so between April and August. This makes most spring planted crops ideal especially cereals, however oilseed rape is unsuitable as it grows too fast. The move to autumn planted cereals, which are harvested early spring results in stubble from recently harvested crops which does not provide enough cover for skylarks looking to nest. The lack of suitable nesting areas is not the only factor that has contributed to the decline in skylark numbers. Advances in farming machinery when planting seeds means less waste from seeds scattering but has an impact on the skylark's food source and the use of pesticides kill the skylark's main food source of insects and weeds which are particularly important to fledglings (RSPB, 2010).

The Aporia decision making model can incorporate any of the three socio-economic storylines (see **section 3.3**). The storylines have an effect on both the profitability of the land owner's fields and their social acceptability. The typologies (see **section 3.4**) given to the land owner is also incorporated into the decision making model. This determines which regimes of crops the land owners will be growing within their fields.

3.5.3 Lunan ABM Model Indicators

As well as the sub-models listed above, the socio-economic storylines presented in **section 3.3** are also implemented which enables the sub-models to interact with different market prices, subsidies, technological advancements and the social acceptability of land use. These socio-economic indicators vary depending on the storyline being used.

Model Output	**Indicators**
Economic output	• The crop type and yield of the landparcel (the type of crop is determined by the typology assigned to the land owner) • The current market (calculated by market curves found in the socio-economic storylines)
Environmental output	• The number of skylarks found within the landparcel • The amount of biomass produced by each landparcel • The amount of energy produced by the crop type
Social Output	• The social perception of the land use or crop type (this varies depending on the socio-economic storyline that is being addressed)

Table 3-1: The indicators used within the Lunan agent based model

The implementation of the Lunan agent based model is explained in greater detail in **chapter 4.2**.

3.6 R.S.V.T Indicators

The indicators for the Lunan agent based model were a work in progress during the development of the model. Hence a representative subset of indicators that would allow the testing and development of R.S.V.T, was selected by the researcher and colleagues at Edinburgh early in R.S.V.T's project development lifecycle. These can be seen in table 3-2 below.

R.S.V.T Output	**ABM Indicator**
Economic Output	The crop type and yield of the landparcel (the type of crop is determined by the typology assigned to the land owner)
Environmental Output	The number of skylarks found within the landparcel
Social Output	The social perception of the land use or crop type (this varies depending on the socio-economic storyline that is being addressed)

Table 3-2: The indicators used by R.S.V.T. These are a subset of the indicators found in the Lunan agent based model

These indicators values are communicated from the Lunan agent based model to R.S.V.T through the network communications protocol (see **chapter 4.3**). The indicators are visualised via the 3D V.E (see **chapter 4.4.4**).

3.7 R.S.V.T Stakeholders and Storylines

A stakeholder can be defined as any individual or group that has, actively or not, an interest in the performance and operation of an organisation or project. The stakeholders actively involved with the Lunan catchment include farmers and landowners, policy makers, agent based modellers and researchers. R.S.V.T does not target the same stakeholder groups. The 3D nature of R.S.V.T's front-end presents an opportunity to encourage previously disengaged stakeholders (those less directly or less consciously involved) to raise awareness of the effects land use change has on the Lunan landscape. R.S.V.T targets the general public (non-expert stakeholders).

R.S.V.T makes use of the three storylines described in **section 3.3**. These storylines form the basis of the testing strategy implemented in this project. This ensures R.S.V.T's visual output provides stakeholders enough information to come to conclusions about the social, economic and environmental state of the Lunan catchment. Implementing the three socio-economic storylines as a testing strategy is discussed in detail in **Chapter 6**.

3.8 Conclusions

This chapter presents background information relating to the Lunan Catchment and its contribution to the larger project – the EcoChange project. The indicators chosen by R.S.V.T were shown and their implementation is discussed in the next chapter. Also, the storylines available to the ABM were highlighted. These storylines form part of the testing strategy of R.S.V.T which can be found in **Chapter 6**.

Chapter 4 – Design, Methodology & Implementation

This chapter describes the fundamental concepts and components of the software design and engineering process used to create R.S.V.T. These concepts are defined as the design approach used to develop the system and include the software engineering methodology that was implemented as well as the design patterns that are followed. Following on from this introduction and explanation of the concepts is a comprehensive presentation of the components that make up the R.S.V.T system with a particular focus on the components that make up the 3D user interface and the data communication protocol which acts as a messaging system, delivering ABM-specific data to the user interface so it can be rendered appropriately.

4.1 Approach to Software Engineering & Design

A software engineering methodology is a collection of the methods, procedures and processes that are used to design and maintain a software engineering project. Many different software engineering methodologies exist although most can be categorised as either sequential or cyclical, often referred to as the Waterfall and Spiral methodologies (Burback, 1998). Each approach has its own strengths and weaknesses and it is left to the developer to determine which methodologies to adopt. Broadly, if there are simple and well defined requirements the waterfall approach is used whereas prototype software, problems with ill-defined requirements or software with expected revisions of objectives and goals tend to adopt the cyclical approach.

The Waterfall Approach

The waterfall approach follows a strict and sequential workflow pattern that is illustrated in Figure 4.1.

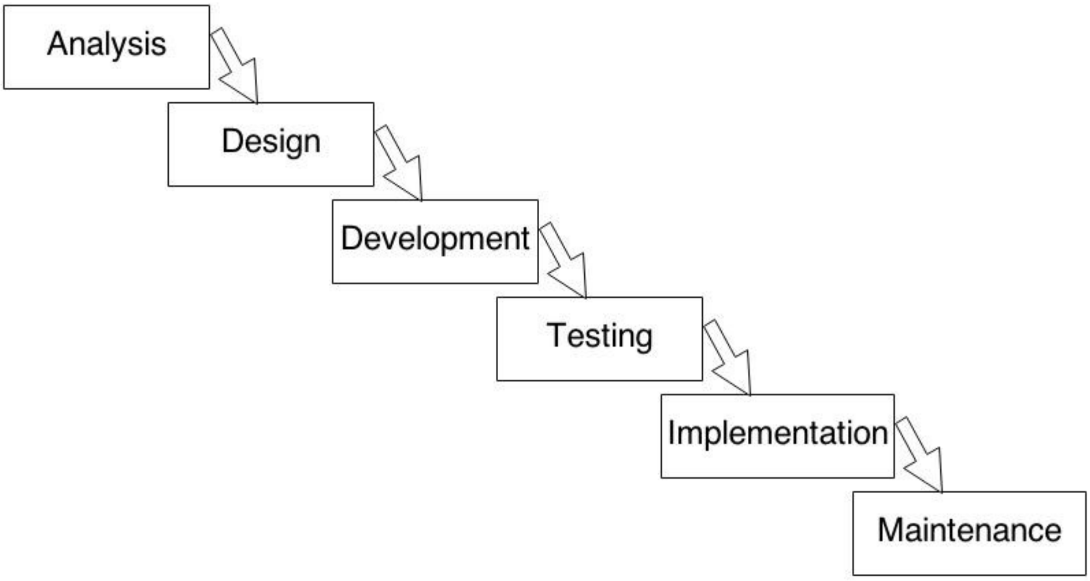


Figure 4.1: The waterfall software development life cycle. Each stage must be fully complete before moving forward. Moving backwards is not permitted.

The rigid nature of the waterfall ensures that all project requirements are identified at the beginning of the project, making it difficult to change or add new requirements at any point during the development process. For projects that need more flexibility due to client demands or working with new and untested technology the spiral methodology is often used.

The Spiral methodology

The Spiral methodology (sometimes referred to as the cyclical methodology) follows a more iterative workflow demonstrated in figure 4.2. The figure shows an initial requirements stage followed by an undetermined number of iterations towards fulfilling the initial requirements. At the end of each iteration a review is undertaken to assess progress and adjust targets as necessary. The cyclical methodology allows for working software to be produced at the end of each cycle but is flexible enough to handle requirement changes requested by either the client or required development changes if poor design decisions have been made in previous iterations (Balaji, 2012).

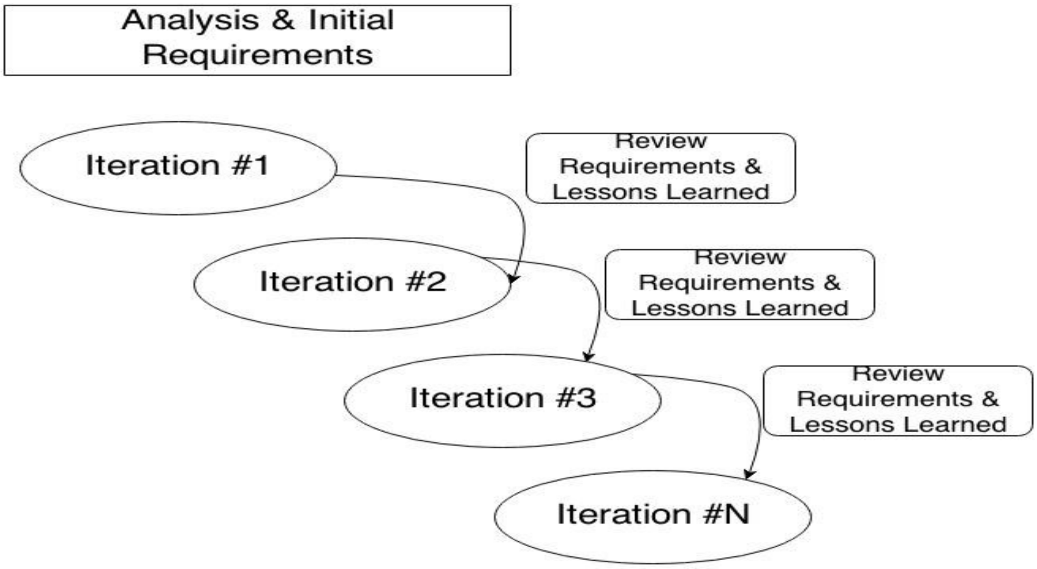

Analysis & Initial Requirements
Iteration #1
Review Requirements & Lessons Learned
Iteration #2
Review Requirements & Lessons Learned
Iteration #3
Review Requirements & Lessons Learned
Iteration #N

Figure 4.2 An example of an RAD life cycle (a derived cyclical methodology). After initial requirements are identified each iteration has a review stage. Progress can be made forwards and backwards.

4.1.1 Issues of Lone Development

There are many variations of spiral development cycles such as Rapid Application Development (RAD), Agile and Scrum. RAD is a software development lifecycle that take an iterative approach to development and was the first popular methodology to break from the strict waterfall approach. The RAD lifecycle is presented in the next section – **section 4.1.2**. Below is a brief overview of both Agile and Scrum which both use RAD-like methods as part of their software development lifecycle.

Agile development moves away from traditional internal documentation to a more personal and real-time communication between team members while working software is preferred to comprehensive system documentation. The focus is on individuals and interaction over processes and tools as well as encouraging the software development team to include user experts in the entire process, often making a place for them within the Agile team (Beck *et al.*, 2001). Agile also favours time-boxing over scope-boxing. Time-boxing sets a specific amount of time to spend on a feature rather than determine exactly how the feature should be implemented which is the case with scope-boxing. Time-boxing rather than scope-boxing is also known to cause problems in inexperienced teams where the time for tasks are often set too short or too long and result in project failures (Shiohama *et al.*, 2012). Finally, tasks are allocated to developers based on their skills, expertise and personal goals through discussion and debate rather than simply assigned by a line manager.

Scrum borrows heavily from Agile methods, particularly in regard to time-boxing. Every task in Scrum is called a “sprint” and each sprint is given a time to complete. Everything in Scrum, from meetings to code development, testing to deployment, is done using sprints. Scrum also encourages highly collaborative and self-organising teams. This has been documented to lead to internal power struggles within teams and a lack of leadership and direction (Chamberlain et al., 2006; Hosbond & Nielsen, 2008).

Although many spiral-influenced methodologies were created specifically to work well with small teams of developers there remains a gap in the literature and indeed a lack of defined methodologies for lone developers, likely because most, if not all, business-level software is created by larger teams. The fact remains that little literature exists to aid lone developers looking to follow a particular software engineering process. Lone development is often referred to as "*Cowboy Programming*" and is usually surrounded with the implication of poor development practice. This is an unfair assessment of the lone developer strategy which is commonly found at educational institutions and even large software companies such as Google (Whitten, 2004). It is worth noting that both RAD and Agile methodologies have, in the past, been labelled *"Cowboy developments"* (Evans, 2006) yet the number of IT solutions using Agile and RAD indicates it is now the "status quo" among developers and its implementation is expected to increase over the coming years (AppsOnTheMove, 2010).

Akpata & Riha (2004) present the argument that not all lone development should be regarded as having little respect to the application of development methodologies, stating that a lack of a formal software engineering methodology can be indicative of the size of the project or a project with an experimental nature rather than poor programming standards and practice. Since little guidance is offered to lone developers it is left to their experience of which methodologies will yield the best results while limiting the possibility of over-engineering a solution to the detriment of the project.

4.1.2 Rapid Application Development with Test Driven Development: An implementation of a cyclical methodology

RAD is a spiral software development methodology that favours rapid prototyping over planning. As is the case with all software engineering projects, planning is an important and necessary part of RAD. In RAD, initial requirements planning is carried out but during this stage a much more abstract view of what the system will do is conducted. This makes it easier to change requirements at any stage during the software development life cycle. At the end of each iteration a working prototype that addresses the needs of the requirements is delivered and deliberated on, at which point the requirements are revisited and amended to incorporate any additional requirements.

The iterative nature of rapid prototyping is well suited to the development of R.S.V.T. R.S.V.T coexists with an ABM that is in simultaneous development by the modelling team at the University of Edinburgh (Murray-Rust *et al.*, 2011; Guillem, 2012). As both systems are in concurrent development it results in ever-changing requirements as both R.S.V.T and the ABM near completion. Additionally, the experimental nature of creating a loosely coupled visualisation tool means it is impossible to state, categorically, what impact additional features would have on system resources until the features are implemented. Adopting a RAD approach allows quicker prototyping of new features which can quickly determine the feasibility of each new feature without having made the commitment of it being included in the final release.

Finally, a major advantage of RAD is the use of *Test Driven Development* (TDD). TDD focuses on writing test cases that ensure the code that will be written acts as it is intended to. Test cases are written first, before any code has been implemented and, as such, every test fails in the first instance. When a test has been written, and fails, the developer is required only to write relatively small amounts of code to get the test to pass. Once the test has passed, the developer writes the next test case and the process starts again. This approach allows the developer to focus on the task in hand rather than having to think of the system as a whole by reducing the scope of what the developer has to implement (Venners, 2007). By following TDD the timescale between making a design decision and gathering feedback related to the implementation and performance of the design decision is significantly reduced, which keeps the *Cost of Change* (economic and time costs of changing features) lower (Bohem, 1991). Finally, TDD creates a regressive test bed of functionality – this ensures that previously working code remains as such even after the implementation of new code. This limits breaking changes during the development stage. By continuously running tests a developer can quickly discover if new design implementations has unwanted side effects on other parts of the system as any breaking changes will cause a test to fail.

4.1.3 Design Patterns and the Use of Model-View-Presenter

Software systems have developed into some of the most complex constructions (Gamma, 1995). Software design patterns are solutions to some of the most common problems developers face and provide abstract, reusable solutions to a given problem. Design patterns allow developers to construct software through the use of eloquent coding standards that have been tried and tested, improving re-usability and readability.

R.S.V.T is a system comprised of three components. These components are:

1) The agent based model
2) The 3D user interface
3) The communication protocol which is responsible for driving the 3D user interface

With the system broken down into these three distinct components it is possible to identify a suitable design pattern, namely the Model-View-Presenter (M-V-P) as seen in Figure 4.3. The components of R.S.V.T can now be better described through the use of M-V-P.

- The agent based model (**Model**)
- The 3D user interface (**View**)
- The communication protocol which is responsible for driving the 3D user interface (**Presenter**)

The M-V-P design pattern originally started as a web-centric pattern and was created by Taligent to address the static nature of HTML, the language of the web (Potel, 1996). HTML is a static scripting language which means that once the browser has rendered the HTML to your screen it does not change. It lacks the dynamic interfaces and interaction that you find in high-level programming languages such as C# and Java. This makes writing interactive websites somewhat difficult. M-V-P was created to help solve this problem by providing cleaner separation between the View, the Model and the Presenter so that each component can be written and tested independently.

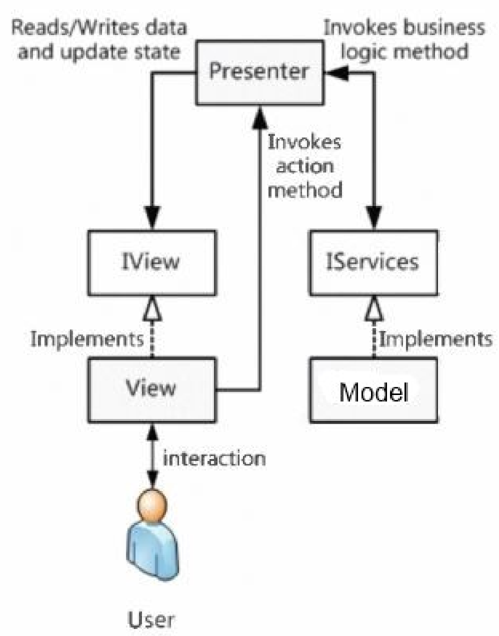


Figure 4.3: An example of the data flow and execution of an MVP enabled application.

In figure 4.3 it is shown that the user interacts with the View only. The View then passes any actionable responses to the Presenter which in turn passes it to the Model. The Model continues to run, processing business logic and sends this information to the Presenter which in turn updates the View. Once the View receives this information the user will see the effects of any change to the business logic appear on screen. Note that at no point does the View interact with the Model and vice versa. All information handling is done by the Presenter. IServices and IView are programmable Interfaces that forces data being passed or received to conform to pre-defined data types. IServices is a list of known services the model is capable of fulfilling. The IServices list includes market prices, access to land parcel information and access to skylark data all within the model. This ensures the Presenter only receives data it can process.

The steps required to start R.S.V.T are:

1) The user interacts with the view by clicking on the start button located within the user interface.

2) The View passes the *start event* to the Presenter. The Presenter handles the user input, and if necessary transfers the information to the model.

3) The Model receives the information from the Presenter and updates any logic states, in this case it starts running model.

4) The Model sends all IServices to the Presenter (see **section 4.3.1**). Once the Presenter receives this it sends any information relating to IView to the View which in turn updates what the user is seeing.

By using M-V-P the system has become three independent components. Although linked through the Presenter both the View and the Model can be designed and tested independently. This de-coupled nature allows the View and the Model to be written in the programming language best suited to the problem domain. All that is required is that the interfaces (IView and IServices) are conformed to.

4.1.4 Initial requirements Specification for R.S.V.T

After consultation with potential users the set of initial requirements were established and categorised into functional and non-functional requirements which can be seen in table 4-1 below.

Requirement	Type
Real-time loose coupling of model and 3D User Interface	Non-functional
Present ABM results as 3D visualisation methods	Functional
3D User Interface processes ABM in equal time to the 2D ABM interface	Non-functional
3D User Interface is interactive, representational and immersive	Functional
Can present the social, ecological & economic status of land parcels	Functional

Table 4-1: R.S.V.T's initial requirements specification

4.1.5 Technologies and Licences used by R.S.V.T's Model-View-Presenter

In **section 4.1.3** the Model-View-Presenter design pattern is presented. It gives an abstract overview of how M-V-P is implemented and the reasons why M-V-P was chosen. This section provides details on how the M-V-P pattern was assigned in the context of R.S.V.T and which technologies constrained by the initial requirements specification were used for the Model, View and Presenter.

4.1.5.1 Model Technologies

The agent based model used in R.S.V.T is part of a larger agent based framework written for the EcoChange project (see **section 4.2**). The agent based model is written in Java through the use of a tightly integrated modelling system called Repast Simphony.

4.1.5.2 View Technologies

The View is R.S.V.T's 3D user interface. The underlying programming language used is C#. Due to the graphical nature of the View and the required immersive and interactive characteristics required of the 3D virtual environment, the XNA 4.0 library is also used. XNA is a graphical library, released by Microsoft that mimics much of the behaviour that is exhibited in DirectX 9/10 without adding the complexities of memory and resource management. Originally XNA 3.1 was used however the release of XNA 4.0 provided additional features that were of benefit to the project. XNA 4.0 was a major update that had code breaking changes (Hargreaves, 2010). Additionally, the XNAInput (Christie, 2008) was used to enable the Xbox controller to control camera movement as well as updated mouse/keyboard functionality. The XNAInput library is released under the MIT licence.

4.1.5.3 Presenter Technologies

The Presenter is R.S.V.T's network communication protocol allowing bidirectional communication between R.S.V.T and the ABM. It is responsible for delivering messages (data) both to and from the View and the Model. After an evaluation of the possible technologies to use for the Presenter (see

Chapter 5) Protocol Buffers were selected. Protocol Buffers is described as a "language-neutral, platform-neutral, serialized data structure for use in communications protocols and data storage" (Google, 2008). Protocol Buffers is released under the New BSD License. This license applies only to the Protocol Buffers used to generate the Model-side messages as the original Protocol Buffers supports only C++, Java and Python. To generate messages from the View the protobuf-net library was used which was created by Marc Gravell. This .NET implementation of Protocol Buffers is released under the Apache license.

4.1.5.4 License Compatibility

The external libraries used (and their respective licenses) are all GPL3 compatible. This guarantees that the use of R.S.V.T in either a commercial or personal environment cannot be charged for by the creators of any libraries that are used as part of R.S.V.T. It should be noted that changes made to the XNAInput library require the MIT licence to be updated to show these changes. All licenses can be found in **Appendix A**.

4.2 Implementation of the Model – The Lunan ABM

The Lunan ABM is made up of three components.

1) A skylark individual based model (Guillem, 2012)
2) A farmer decision making model called Aporia (Murray-Rust *et al.*, 2011; Guillem *et al.*, 2011)
3) A sub model that determines how much energy is produced by food and biofuel crops.

A dataflow diagram that shows the interactions between each system can be found in figure 4.4 below.

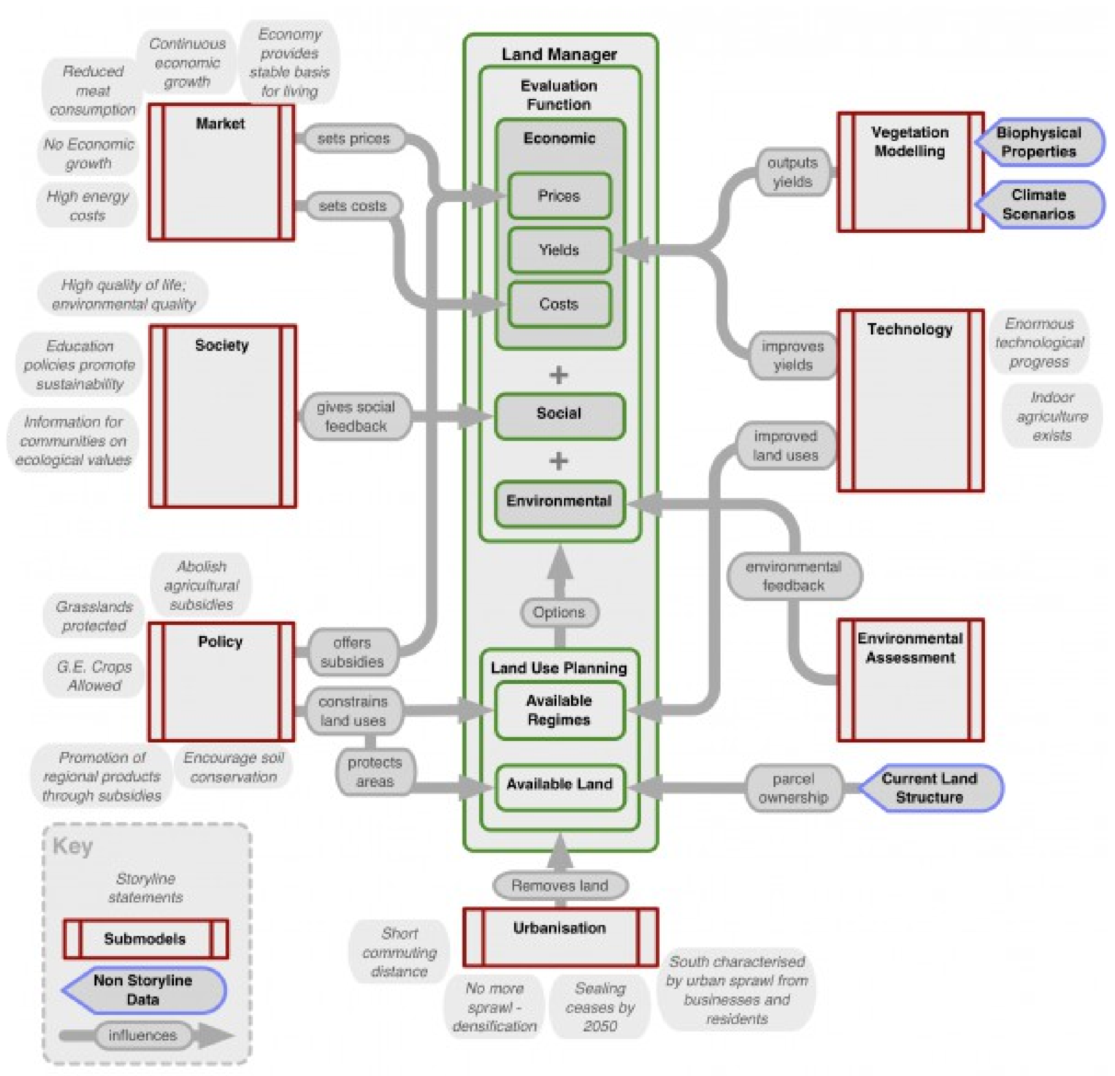


Figure 4.4: A dataflow diagram of the ABM (from Murray-Rust, 2013).

The decision making by land managers is informed through the evaluation function (green boxes/middle – figure 4.4). Each land manager has a range of options available to them based on the land use planning data such as available land and possible regimes. The regimes themselves are determined through which scenario the ABM is running (BAMBU, SEDG, GRAS) as discussed in **section 3.3**, and the type of behaviour each land owner exhibits as determined through their farmer typology settings as discussed in **section 3.4**. The model was written on the OSX platform using Repast Simphony – the ABM toolkit.

4.3 Implementation of the Presenter – Data Transfer via Protocol Buffers

The Presenter acts as a mediator between the Model and the View, receiving interactions from both and updating the each as necessary. In order to ensure that any changes are rendered correctly by the View a designated language or communication protocol is required. This is important because:

- The View should never receive any communication or command that it cannot understand. To do so could result in unexpected behaviour of the View.
- The Model should be a domain specific, independent and a loosely coupled component of the system. By ensuring the Model only sends communications that can be understood by the Presenter we are ensured the View will update as expected.

Referring back to figure 4.3 the IServices interface dictates, explicitly, what data can be sent to and received from the Presenter. Similarly, the IView dictates what data can be sent to and received from the Presenter and forms the basis of the communication protocol.

4.3.1 Protocol Buffers as an Intermediary Language

Protocol Buffers are already described as a “language-neutral, platform-neutral, serialized data structure for use in communications protocols and data storage”. It is used by Google for almost all of its internal RPC protocols and file formats (Google, 2008). Platform neutrality was the deciding factor when determining which communication protocol to use with Thrift (Apache, 2010) being disqualified due to lack of OSX support. As previously mentioned in **section 4.2** which outlines the ABM outputs, OSX was the primary development environment for the Lunan agent based model.

Protocol Buffers works through the creation, serialization and deconstruction of messages. These messages become the intermediary language between the Model and the View. Determining what messages should be understood by the Presenter was done through identifying the various *design elements and visualisation methods* that are included in the 3D User Interface – the View. These *design elements and visualisation methods* are explained in greater detail in **section 4.4.4.4** but an overview is provided below in the context of the Presenter’s messaging format.

4.3.2 The Presenter’s Message Format

Protocol Buffers is based around messages which are essentially data packets. Since the structure of these messages is known by both the Model and the View it acts as an intermediary language for passing data to and from the Model and View via the Presenter. Each message can be sent either over-the-wire (a local network, the Internet) – this is defined within R.S.V.T as **online-mode**, or stored as either text or in database tables – this is defined within R.S.V.T as **offline-mode** (see **section 4.3.4**). Regardless of the method of message delivery, the message structure stays the same. The messaging format is described below.

LandParcels

This message encapsulates all data relating the landparcels within the Model. It includes the following data:

ID	This is a unique identifier given to every land parcel
Owner ID	This is used to identify the owner of the field
Land Use	A string representation of the current land use of the land parcel
Vegetation Height	This is the current height of the crop

Skylarks

The skylark data encapsulates the following data:

Lifecycle Stage	Each skylark has four possible lifecycle stages. These are egg, nestling, fledgling and adult.
Age	This is the age, in days, of the skylark
Latitude	The current latitude of the skylark
Longitude	The current longitude of the skylark. Coupled with the latitude this gives an accurate geolocation of the skylark
Parcel ID	This acts like a foreign key to the landparcels ID. It determines the location of the skylarks nest.

ModelState

The model state provides generic information about the current agent based model run:

Current Tick	Each tick is a virtual day within the ABM. This indicates how many ticks have elapsed since the model started.
Date	This is the current date the ABM is simulating.
Run ID	This is a unique identifier generated for each new model run.

ModelControl

This message encapsulates LandParcels, Skylarks and ModelState messages. By encapsulating these three messages into one larger message the Model and Presenter is only required to handle one message per tick.

Ensuring that Protocol Buffers is able to serialize, transfer and de-serialize in a timeframe equal or superior to 2D user interface provided by RepastS is a key deliverable of the R.S.V.T project and a pilot test was created to ensure this was possible. The pilot test can be found in **Chapter 5**. However a brief description of its goals is appropriate here as it is the basis of the further development of the Presenter.

4.3.3 Overview of Pilot Testing

One of the issues surrounding the implementation of Repast Simphony's 2D User Interface is the poor performance of updating the 2D graphical interface. In the literature this was found to be caused by little support for GPU-accelerated rendering within Repast Simphony and the close coupling of the Model and the View. The addition of the Presenter is designed to loosely couple the model to R.S.V.T's View (the 3D User Interface) which makes heavy use of GPU-acceleration on the graphics pipeline. The pilot testing provides an early platform to test the performance of the system and was conducted 18 months into the project.

The pilot testing also formed the evaluation of network protocols carried out in **chapter 5** and as a result of the testing, additional requirements to the system were added (see **section 4.3.4**). Two important constraints from the results of the pilot testing are worth noting. These are:

The terrain generation algorithm used during the pilot testing was not optimal and required adaptation to better suit large-scale terrains. This was rectified by implementing splitting the terrain into quadtrees and implementing a LOD (Level-Of-Detail) algorithm (see **section 4.4.3**)

The agent based model is also a CPU resource-intensive application. An assumption was made that the CPU resources required for the ABM would only increase as the ABM continued its development path. Reducing the resources required to run the agent based model is the basis of developing R.S.V.T's **offline-mode**, which is discussed below in **section 4.3.5**.

4.3.4 Refined Requirement Specification

The additional requirements as a consequence of pilot testing are in highlighted in bold in table 4-2 below.

Requirement	Type
Real-time loose coupling of model and 3D User Interface	Non-functional
Present ABM results as 3D visualisation methods	Functional
3D User Interface processes ABM in equal time to the 2D ABM interface	Non-functional
3D User Interface is interactive, representational and immersive	Functional
Can communicate the social, ecological & economic status of land parcel strategies	Functional
LOD techniques to improve loading and rendering of large terrains	**Non-functional**
Users can run R.S.V.T with a live simulation or offline from saved file or database	**Non-functional**
Users can compare future scenarios	**Functional**

Table 4-2: Refined requirements specification for R.S.V.T.

4.3.5 R.S.V.T's Offline-Mode

During the pilot testing processor usage for the agent based model peaks at ~75% on the host system. As further development on R.S.V.T (particularly the 3D User Interface) i.e. visualisation of many 3D objects will be required, there will be an associated increase in the required CPU resources. A solution is required that enables a previously run ABM simulation to be pre-loaded via a file or database. This would unencumber the CPU from processing the ABM, leaving the computing resources free for visualisation. Additionally this would enable users to reload interesting scenarios and would be critical for scenario development using a set of possible future scenarios.

The nature of Protocol Buffers messaging format (see **section 4.3.2**) allows messages to be stored in their serialized format for later retrieval and de-serialization. This makes R.S.V.T's messaging format well suited to be stored in either a file or database.

4.3.5.1 File-Based Storage

In R.S.V.T's **online-mode** serialized messages that store data are sent to the Model via the Presenter to the 3D User Interface at the end of each tick. This transfer happens over some form of network (either a local network or the Internet). The development of R.S.V.T's **offline-mode** states that the serialized message is not sent over the network but instead is saved to a file in a custom file format that can be seen in table 4-3 below.

Tick	Message Type	Serialized Message
1	ModelState	Protocol Buffer's Binary object
1	LandParcel	Protocol Buffer's Binary object
1	LandParcel	Protocol Buffer's Binary object
1	…..	Protocol Buffer's Binary object
1	Skylarks	Protocol Buffer's Binary object
1	Skylarks	Protocol Buffer's Binary object
1	…..	Protocol Buffer's Binary object

Table 4-3: The layout of the custom file format for R.S.V.T's offline-mode

The table above gives an example of the structure of tick 1 – the start of the model. The first message type is ModelState, which holds data on the current tick, runID and the current date of the simulation. This must be the first message type to ensure backward compatibility with the 3D User Interface. There are no limitations on the order of subsequent message types, the structure in table 4-3 imitates the order in which the data is serialized within the Model. Both LandParcel and Skylarks message types are of variable size because not all landparcels or skylarks require updating at every tick. The smallest message size, which is created during the first model tick is 64 kilobytes in size. The largest land parcel message is often created towards the end of the model simulation with the largest recorded message being 250 kilobytes. There will never be more than 1149 LandParcel message types per tick as there are only 1149 unique landparcels within the Model. The number of skylarks that are being modelled within the simulation varies from simulation to simulation. The largest number of skylarks, recorded during a SEDG scenario (**chapter 3.3**), is over 3,500 skylark messages for one model tick.

A ten year simulation will produce 3652 message files. File I/O is often a bottleneck in software applications as read/write speeds from hard disks are much slower than processors. R.S.V.T files are loaded in blocks of 100 ticks at a time which reduces the number of read operations. As the 3D User Interface reaches tick 100 it creates a new background thread to load the next 100 ticks. The background thread is created after 80 ticks of the current block is reached to prevent the 3D User Interface from having to wait, it seamlessly carries from tick 100 to 101 without any noticeable performance degradation.

Scalability Issues Surrounding File-Based Storage

A major disadvantage of this type of file based storage is poor scalability. A typical 10 year simulation stored in R.S.V.T-format takes up ~500MB of hard disk space. The file size for each tick is directly related to each message size. It is known that only 1149 landparcels exist within the Model so the maximum message size for the LandParcel message type can be calculated as

$$1149 \times 106 = 121794\, bytes$$

106 bytes is the maximum message size for a landparcel message (see **section 5.2.3**) and the assumption is that every landparcel needs updating which results in a message size of 119 kilobytes (121,794 bytes).

Since the number of skylarks varies depending on the intricacies of each simulation it is impossible to predict the amount of space required for a simulation. This leads to a possible situation of having a file of tens or even hundreds of Megabytes in size for a single tick. This scenario would have severe performance implications when attempting to read the file from disk. To ensure future scalability of the R.S.V.T messaging system R.S.V.T makes use of MongoDB – a NoSQL database.

4.3.5.2 Database Storage with MongoDB

MongoDB (from the word *"humongous"*) is a document-oriented NoSQL database. NoSQL databases have proven popular with developers working with large datasets due to:

- The ease of integrating developer data structures through the use of documents and rather than rows and tables.
- Its speed in which data can be accessed compared to traditional Relational Database Management Systems (RDBMS) such as MySQL.

NoSQL databases provide the ability to automatically scale to ever increasing datasets through the use of sharding. A shard is a container that contains a subset of a collections data. Each shard is a separate instance of the MongoDB service. This serves to automatically load balance the data across different hard disk or machines should the database become too large.

R.S.V.T makes use of benefits listed above while the ability to shard is one of scalability and ensures that R.S.V.T will run as expected even if the agent based model grows beyond its current state.

Data Structure Integration

In RDBMS data is stored in rows and organised in tables. MongoDB and other NoSQL database systems use the terms *"document"* and "*collection"* to roughly the same effect. A document can be interpreted as a row and many documents make up a collection. Referring back to **table 4-3** each row in the table becomes a document and all the messages associated with each tick is the collection.

A key benefit of NoSQL is that it is well suited for storing and retrieving hierarchical data such as XML and complex objects (which are essentially user-defined software objects that have non-standard data types such as other user-defined software objects I.e classes) when compared to traditional RDBMS. Often complex objects do not map directly to one row in a traditional RDBMS table. An example of this is R.S.V.T's Skylark message (see **section 4.3.2**). The Skylark message has five defined datatypes including lifecycle, age, latitude, longitude and parcelID. The first four defined datatypes are strictly related to a skylark's state at any given point within the model whereas the parcelID is not strictly skylark related as it is related to a LandParcel as well which would be defined in a separate table within a RDBMS. If the data were being stored in a RDBMS and should the data within R.S.V.T's codebase change to require the skylark object to keep track of the skylark's mate and any nestlings/fledglings it has, then multiple changes would be required. R.S.V.T's codebase would require some change in order to represent the additional data expected from the object. Similarly, changes would be required to the RDBMS table to represent the changes made in R.S.V.T's codebase. This can cause problems when software requirements are likely to change (such as using a RAD software methodology) as each change requires two separate technology stacks to be changed.

MongoDB supports Data Structure Integration which ensures the relationships between different pieces of data works in much the same way an object oriented programmer would code the object. This means when a developer is trying to access, update or store data they can follow the same object relationships used within their codebase rather than an object relationship model that is enforced through primary and foreign key methods which are found in RDBMS and not at all related to object-oriented programming. This type of structure aids software developers due to its similarity to high level programming languages. This is shown in the example code snippets below that returns all landparcels with a land use type of *wheat* that is at least 2 feet high.

<u>MySQL</u>

*SELECT * FROM model1_tick1 WHERE landUseType=”wheat” AND cropHeight > “2”*

<u>MongoDB</u>

DB.model1_tick1.find({landUseType: “wheat”, cropHeight: gt: 2.00});

The MongoDB syntax is much closer to that of high-level languages whereas the MySQL statements uses operators and clauses such as **SELECT**, **WHERE** and **AND** which are not normally considered elements of high-level object-oriented programming.

Memory Mapping

MongoDB provides read/write speed increases when compared to traditional file storage and RDBMS. This is due to the automatic memory mapping of data within documents (MongoDB, 2012). Memory mapping is the process of placing the data contents into virtual memory which provides a fast method for accessing and manipulating data.

4.3.6 Summary of the Presenter – Protocol Buffers

This section highlights the important role the Presenter plays by encapsulating data from the Model and making it available to the View both in online and offline mode. The main Presenter technology (Protocol Buffers) is introduced and the four key messages, *model state*, *land Parcel data*, *skylark data* & *model control* that drive the 3D User Interface were described in

section 4.3.2. These form the basis of communicating the social, environmental and social indicators from the ABM to the visualisation.

The results of the pilot test prove the system is loosely coupled and is capable of visualising model output faster than the traditional 2D output (see **chapter 5**). Finally, the pilot testing also highlighted potential performance issues that were addressed through the creation of R.S.V.T's offline mode that can be found in **section 4.3.4**.

The chosen presenter technology with the flexibility of online and offline mode conforms to the requirements 1, 2, 4 & 7 which can be seen below.

Requirement	Type
1) Real-time loose coupling of model and 3D User Interface	**Non-functional**
2) Present ABM results as 3D visualisation methods	Functional
3) 3D User Interface processes ABM in equal time to the 2D ABM interface	**Non-functional**
4) 3D User Interface is interactive, representational and immersive	Functional
5) Can communicate the social, environmental & economic status of land	**Functional**

parcel strategies	
6) LOD techniques to improve loading and rendering of large terrains	Non-functional
7) Users can run R.S.V.T with a live simulation or offline from saved file or database	**Non-functional**
8) Users can compare future scenarios	Functional

The next section describes how the View, R.S.V.T's 3D User Interface, handles the Presenter messages through dynamically altering its visual state to visualise the output of the Model.

4.4 Implementation of the View – The 3D User Interface

The technological characteristics of Virtual Environments (VE) that contribute to positive learning outcomes and engagement are: first order experiences such as free navigation and first person view, autonomy and presence (Mikropoulos & Natsis, 2011). Dalgarno and Lee (2010) argue that it is the fidelity of representation and interactivity that are properties of the VE which lead to immersion and consequently a strong sense of presence. In the development of the 3D user interface the literature on virtual environments was drawn on.The 3D interface is developed using computer game technologies therefore embodied actions including view control, navigation and object manipulation can be achieved. Free navigation and first person point of view are features attributed to first order experiences which enhance learning and promote engagement and immersion. The 3D user interface permits interaction similar to that of computer games, allowing the user complete control over how the environment is viewed, existing systems will primarily lock the user into a single viewpoint, usually navigable top down view for GIS or static abstract camera view for CAD renders. Interactivity is a key element of the 3D interface where users can explore and develop their own narratives and scenarios.

Based on contemporary computer games and rendering techniques the 3D V.E renderer has been specifically designed to portray a realistic representation of the landscape therefore adhering to the requirement of fidelity of representation. The representation of the physical environment is derived from geospatially indexed datasets such as GIS files, satellite imagery and digital elevation models (DEMs). Boundaries pertaining to the landscape i.e. land parcels can be visualized using shapefiles.

Elements of the digital landscape that are the focus of investigation i.e. parameters in the computational models can be positioned in the landscape using representational rather than abstract methods (McCreadie et. al, 2012). This chimes with the use of natural semantics in 3D interfaces and virtual environments which avoid the use of difficult to learn and remember semantics.

The remaining requirements are linked to the 3D user interface and shown in bold:

Requirement	Type
Real-time loose coupling of model and 3D User Interface	Non-functional
Present ABM results as 3D visualisation methods	**Functional**
3D User Interface processes ABM in equal time to the 2D ABM interface	Non-functional
3D User Interface is interactive, representational and immersive	**Functional**
Can communicate the social, environmental & economic status of land parcel strategies	Functional
LOD techniques to improve loading and rendering of large terrains	**Non-functional**
Users can run R.S.V.T with a live simulation or offline from saved file or database	Non-functional
Users can compare future scenarios	**Functional**

A description as to how the 3D interface is designed and rendered is provided in this section, cross referencing the requirements. It is described in terms of: Terrain, Surrounding world, Landscape Features and Interactivity.

4.4.1 Terrain

The process of creating computer generated terrain requires the processing a *digital terrain model*, sometimes referred to as a *digital elevation* *model* and their respective acronyms (DTMs and DEMs). Subsequently a texture is applied and in the case of large terrains optimisation methods are employed.

4.4.1.1 Topography

Digital terrains come in two formats, *Triangulated Irregular Networks* (TINs) and *Heightmaps* (Brostuen & Cox, 2004; Larsen & Christensen, 2003). TINs are meshes created from points with variable distance between them. An example of this can be seen in figure 4.5. A *heightmap*, which can be seen in figure 4.6, is a matrix of equidistant points in the *x* and *z* directions, with each point representing a *latitude* and *longitude* and *y* value is generated from a *heightfield* which can be found in figure 4.7.

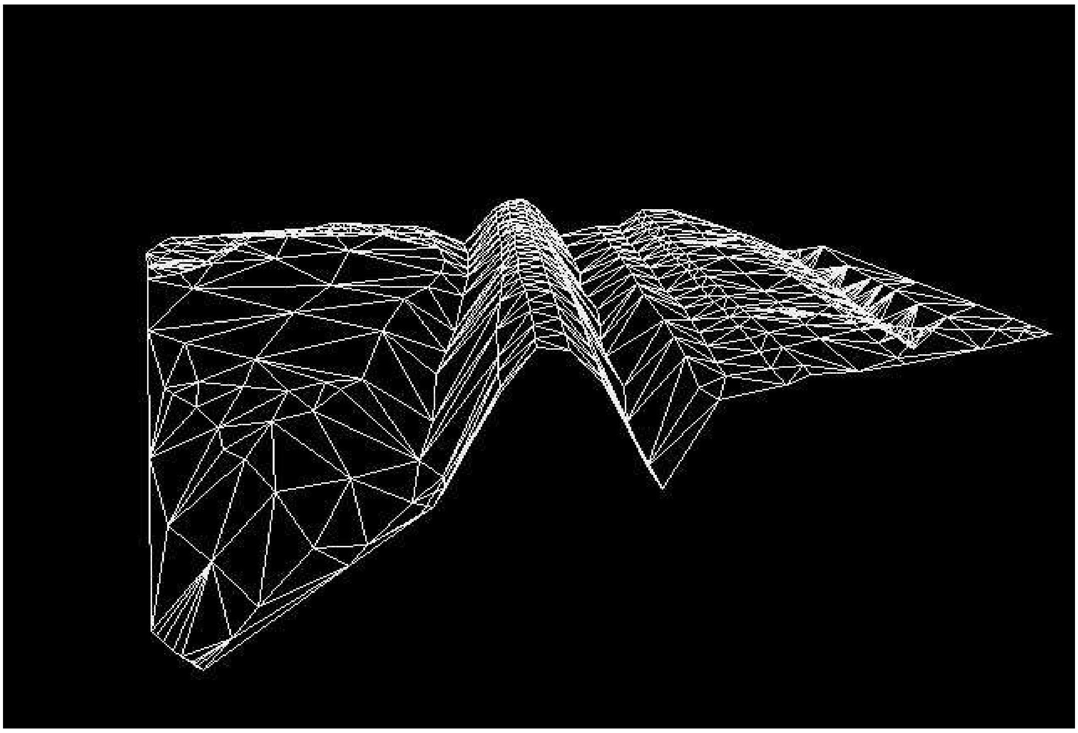


Figure 4.5: An example of a terrain defined using triangulated irregular networks (TINs).

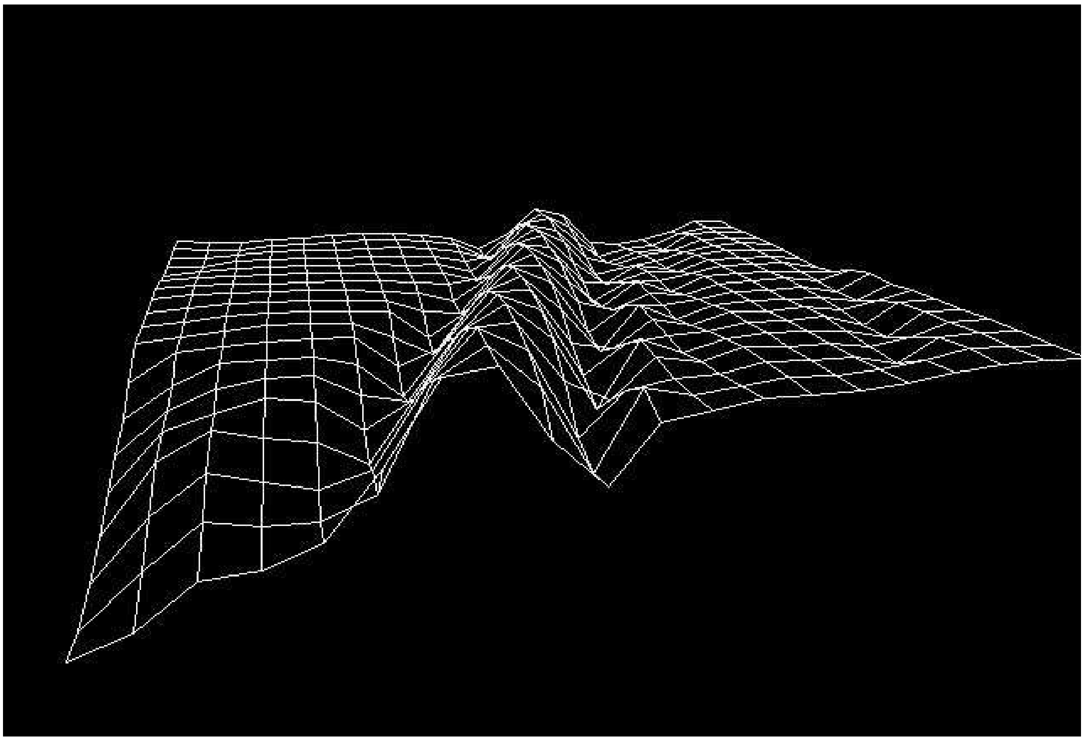


Figure 4.6: An example of terrain defined using a heightfield.

To recreate the Lunan terrain, an accurate source of elevation data for the Lunan region is required. This data can be acquired from the *Shuttle Radar Topography Mission* (SRTM) and is available to download from the *United States Geology Survey* (USGS, 2011). The accuracy of the elevation data is determined by a number of factors which are outlined in the section below.

4.4.1.2 Heightfield resolutions

The accuracy of the heightfield elevation data is determined by the resolution at which the data was acquired. The resolution of elevation data is defined in arcseconds, which represents the difference in angular measurement of the Earth's surface. Since 2002, most GIS applications, and indeed other applications that make use of elevation data, use data acquired from the SRTM. SRTM data was captured at various resolutions including 1 arcsecond which is the equivalent of 30 metres between each elevation value. The 1 arcsecond dataset was only recorded for territories within the United States while the rest of the world has an elevation dataset at a resolution of 3 arcseconds which is the equivalent of 90 metres between each elevation value. All freely available datasets for UK regions are in 3 arcseconds format. Organisations such as the Ordnance Survey and the British Geological Survey may have higher resolution data for certain areas but these higher resolutions incur additional financial expenses. The Lunan heightmap has a resolution of 3 arcseconds however after development had started on R.S.V.T the accuracy of freely available UK data had improved to 1 arcsecond for many parts of the United Kingdom (UK Government Open Data, 2012).

Understanding data resolution helps frame the spatial surface area that is being analysed. However without any height values, this spatial area is flat. Height values are present for each square within the elevation data's resolution grid. An example of the Lunan heightfields with a 3 arcsecond grid overlay can be seen in image 4.7.

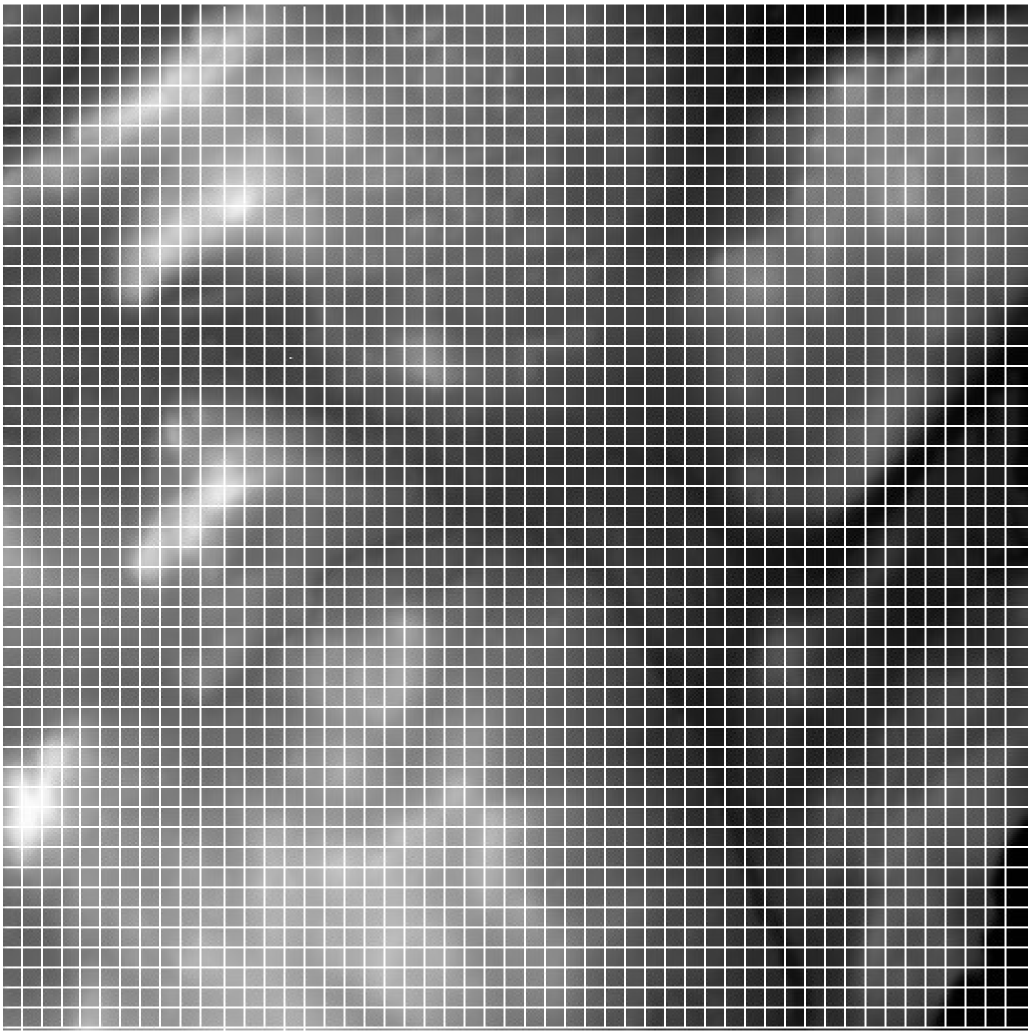


Figure 4.7: The Lunan heightfields with 3 arc-second resolution grid

Heightfields are raster images that store surface elevation values for each pixel of the image. These values are based on *floor* and *ceiling* values which represent the minimum and maximum heights of the terrain respectively. In the case of the Lunan region, a relatively flat region, the *floor* value is 0 (sea-level) which is black. The *ceiling* value was 113 (113m above sea-level) and this is represented as white. The 8bit RGB bitmap used to represent the elevation values of the Lunan terrain can store one of 256 values, where each value represents a height of one pixel.

A standard 8-bit RGB image such as the one in figure 4.7 is capable of showing 256 values and therefore is limited to 256 different heights. The minimum unit of displacement for each pixel height can be calculated using the formula below

$$\frac{ceiling - floor}{Number\ of\ \ possible\ height} = Displacemen$$

With substituted values:

$$\frac{113\,m - 0\,m}{256} = 0.44\,metres$$

Identifying the resolution of elevation data as well as the limitations of greyscale raster images is critical before development on image processing and terrain generation begins. The limit of 256 values for heights is a technological limitation of 8-bit images. If the surface area being modelled is very flat a smaller unit of displacement may be desirable. A smaller unit of displacement would allow very small changes in terrain height to be identified and rendered. Another example that may require additional height values would be if the range between the floor and ceiling values is large. If a heightmap encapsulating hills and surrounding flatter areas were to be rendered using an 8-bit heightmap the rendering would suffer from what is known as “stepped terrain” such as that in figure 4.8.

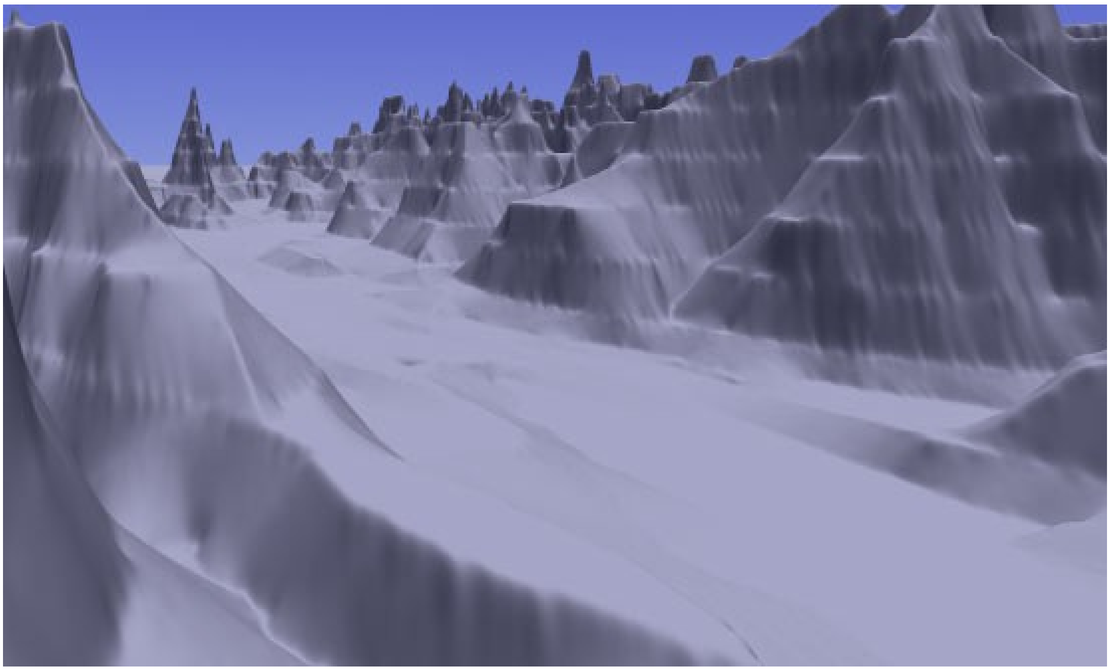


Figure 4.8: An example of "stepped terrain" which is the result of a high displacement value.

4.4.1.3 Lunan Brute force Terrain

The original method of terrain generation is referred to as *Bruteforce Terrain Generation*. This method transforms an entire heightmap into regularly spaced grid and renders the entire mesh without any optimization. This method lacks scalability because of its inability to sort which parts of the terrain to render. Terrain that is being obstructed from the camera view by covering hills or other models obscuring the view is still sent to the graphics card for rendering, wasting valuable resources. The number of pixels required to render terrain using the bruteforce method can be calculated as follows:

number of squares = heightmap width x heightmap length

number of triangles = number of squares x 3

number of vertices = number of triangles x 3

And the equation with the values of the Lunan terrain heightmap

number of squares = 1024 x 512

number of triangles = 524,288 x 2

number of vertices = 1,048,576 x 3

total vertices = 3,145,728

Rendering the terrain using this method requires slightly over 3 million vertices which prompted concerns that performance issues could arise when adding new objects to R.S.V.T in the future. The development cycle devoted to recreating the Lunan terrain was revisited to find a solution to this problem through the implementation of Level of Detail Algorithms.

4.4.1.4 LOD Terrain Implementation

A more *resource-aware* terrain implementation would contain Level-Of-Detail (LOD) optimisations to increase performance. LOD terrains can dynamically reduce the number of vertices used to draw portions of the terrain by determining how far away the portion of landscape is from the camera and reducing the complexity of the area mesh to produce fewer polygons for landscape perceived as far away from the camera. This allows the majority of

CPU and GPU resources to focus on rendering landscape areas that are close to the camera.

Vertex Reduction

Figure 4.9 shows two examples of a small heightmap. The left portion of the image shows the heightmap being rendered at its maximum resolution as used in the Bruteforce Terrain. In LOD, the maximum resolution is *Level 0* and in this example requires 32 triangles to create a continuous mesh. The right portion of the image shows the same heightmap being rendered at one lower resolution (*Level 1*) and requires only 8 triangles to create a continuous mesh.

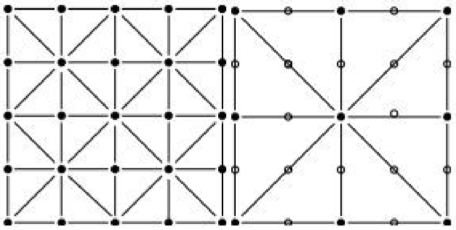


Figure 4.9: Left - The initial and highest resolution heightfield (Level 0). Right - The same heightfield at a lower resolution (Level 1).

As each level increases from 0 the number of triangles required to render a continuous mesh is reduced by 75%.

Quadtrees

In practical use, a 5x5 grid is not going to build a continuous mesh of terrain. To create a terrain of a useful size the terrain is split into 5x5 chunks using quadtrees which conforms to other's use of LOD terrain (Lindstrom, 1998; Cunningham, 2003; Inria, 2008). Quadtrees provide a good solution to dividing the terrain in subsections; they were identified as an efficient method of subdividing space through recursion and they can be more efficiently searched than rasters whilst requiring less space (Goodchild & Mark, 1987). An example of the Lunan terrain, split into quadtree chunks can be seen in figure 4.10.

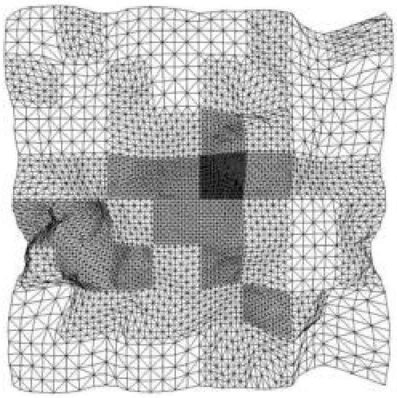


Figure 4.10: The heightmap updated with a LOD implementation. Higher resolution areas are darker than lower resolution areas.

Implementing LOD terrain poses a development obstacle when considering how to determine which level of detail to use for each quad (**section 4.4.3.2** below). Even after selecting the appropriate level of detail, terrain cracking – an unwanted artefact when two adjacent quads have different levels of detail – needs to be dealt with in a timely manner to ensure terrain doesn't *pop* – the name given to the noticeable change in terrain detail.

Selecting the appropriate level-of-detail

The viewer of the terrain will notice distant polygons are rendered smaller than polygons closer to them, this is known as perspective projection.

Avoiding cracks – Terrain stitching

Terrain cracking occurs when adjacent quadtree segments are rendered at different levels of detail as shown in figure 4.11. A technique known as *terrain stitching* is used to ensure a seamless terrain.

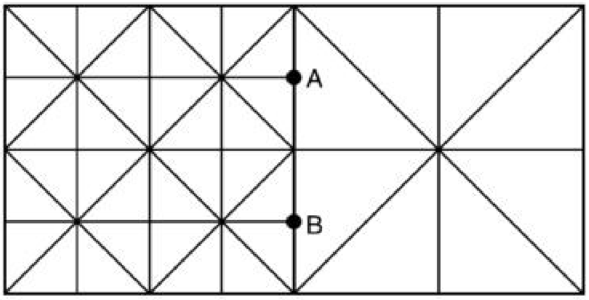


Figure 4.11: Two quadtree segments. The left quadtree has a higher resolution than the right quadtree with the problem vertices highlighted as A and B.

Figure 4.11 highlights the points that will not be processed correctly. Both triangles that contain points A and B would not be rendered as expected resulting in artefacts throughout the terrain. The most efficient way to deal with terrain cracking is to remove points A and B from the quadtree on the left as shown in figure 4.12.

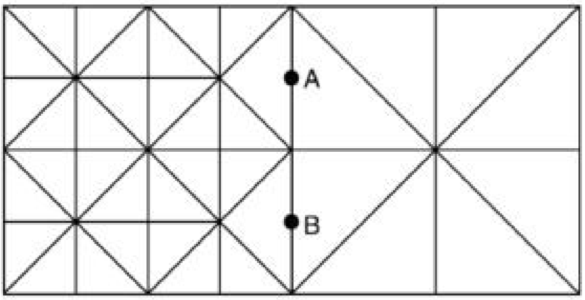


Figure 4.12: The left quadtree has had two vertices removed, the horizontal point from A and the horizontal point from B. This is known as the vertex reduction method.

This form of terrain stitching is referred to as reductionist stitching since this method removes vertices in order to achieve a seamless terrain. Another method (not implemented in R.S.V.T) is to add four additional vertices to each point (**A** and **B**) in the right quad. This is known as the additive stitching method.

4.4.1.4 Terrain Texturing

Texturing is the process of applying an image to a shape, polygon or mesh. In this case, the mesh is the terrain mesh shown in the previous sections. The terrain mesh is required to have some form of texturing to ensure visual quality and convey its land cover. Texturing was not required for the individual land parcels within the Lunan catchment, they are updated by the agent based model therefore their land cover data (and the visual representation through texturing) is to be dynamic in nature, driven by the Model informing the Presenter of the changes (see **chapter 4.3**). However the surrounding terrain not simulated by the model still requires some form of land use cover. Three techniques were designed, two of which are classified as *abstract* texturing and the final technique being classified as *aerial photography*.

Abstract Texturing

Accurate recreation of the land use cover for areas outside the boundaries of the agent based model would have been unachievable. The reasons for this are:

- A lack of data for the land parcels not included in the model and this lack of data is not limited to land use data. Other GIS-related data such as the boundaries of each of the fields were not readily available and would have been expensive to procure.
- The fields outside the parameters of the agent based model would have no effect on the input supplied to the agent based model and would not be affected by any of the events that happened during model simulation.

This led to the decision that surrounding terrain textures should be abstract in nature rather than trying to imitate reality. From this came two texturing techniques for the Lunan terrain, textured and multi-textured.

Basic Texturing

The single texturing method implemented on the terrain is a repeating grass texture that covered the entire terrain. This method is simple but visually repetitive. The biggest concern was the repeating pattern that draws the attention of the user in figure 4.13.

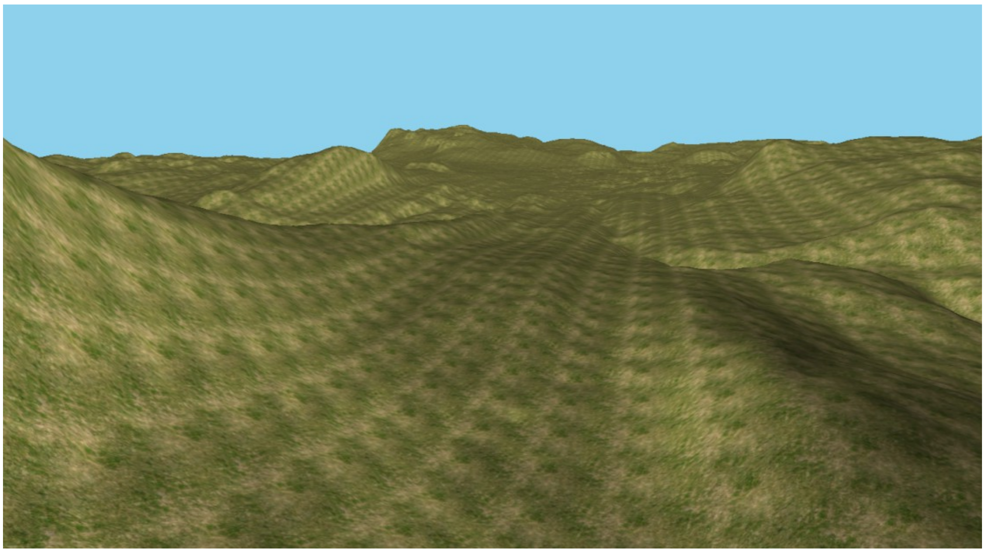


Figure 4.13: The grass texture can be seen repeating, with the effect increasing as the distance from camera does.

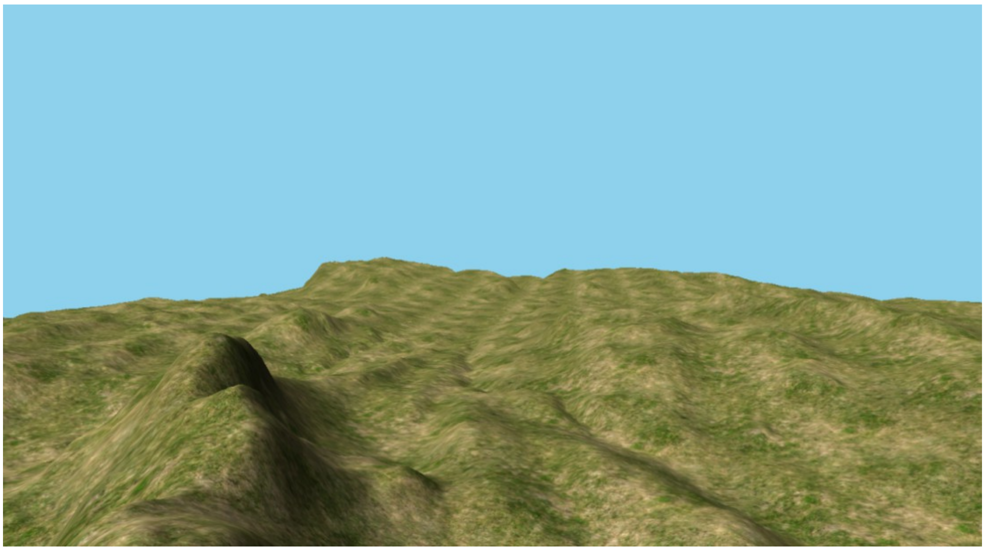


Figure 4.14: A similar scene to figure 4.13 but with larger texture coordinates. Now the furthest parts of the landscape do not visibly repeat but close up texturing looks pixelated.

The first solution increased the texture coordinates by a factor of 5 – this essentially decreases the number of times the texture is repeated over the landscape by 5 times that which is visible in figure 4.13. In figure 4.14 the distant landscape textures do not appear to be visibly repeating but the closer terrain textures looks less appealing – it now appears blocky and pixelated as it is stretched over the terrain. The solution to this was to implement the notion of depth in the pixel shader that renders the terrain.

The pixel shader is capable of altering the appearance of each pixel on the screen before or after the rendering has been done - this is known as pre-pixel processing and post-pixel processing respectively. With regards to the terrain texturing issue a pre-pixel processor was used to introduce the concept of depth.

Virtual cameras, whether used in conjunction with a virtual landscape or a commercial game, need to be given a target to look it, a location in the world, near and far planes and field of view. This results in a region of the world that is in view according to the virtual camera. Only objects within this in-view volume also known as the frustum are rendered. This is illustrated in figure 4.15.

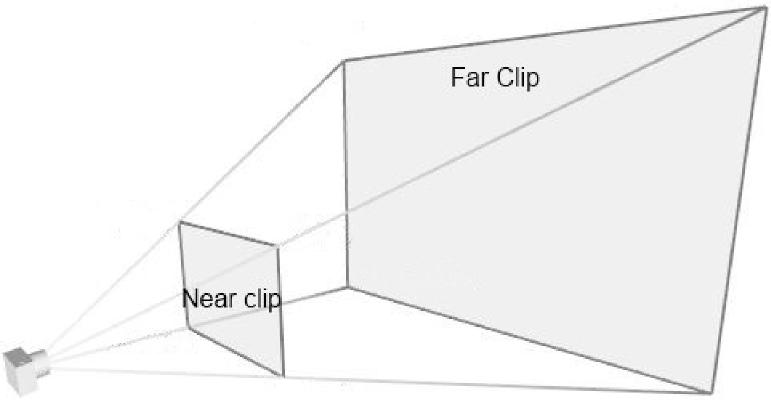


Figure 4.15: The near clip plane is where rendering can start from. The far clip plane tells the camera when to stop rendering objects and landscape. The space between the near plane and the far plane is known as the viewing frustum.

The pixel shader calculates the depth of each pixel based on where the camera is positioned and where the pixel is on the landscape in relation to the camera. All pixel depth values are ranged from 0.0 – 1.0 where 0 references the near clip plane and 1 references the far clip plane. The pixel shader has two developer supplied values named *blendDistance* and *blendWidth*. These values can be changed to alter the distance at which the blending starts (*blendDistance*) and how much the pixel should blend (*blendWidth*) with its neighbours.

The *blendDistance* determines how far away a pixel must be before being subject to blending. If the pixel exceeds this value then blending will occur. How much blending is dependent on the *blendWidth*. The purpose of this technique is to subtly make changes to the edges of the texture coordinates to ensure the repeating texture is not identified by repeating edges.

Multi-Texturing

Multi-texturing, sometimes known as texture splatting is a technique for texturing terrains with high resolution tiled textures (Bloom, 2000). This technique blends together multiple textures to give the landscape a more realistic look. In figure 4.16 below there are three separate texture images. The images on the left and centre are individual textures while the final image on the right shows the textures from left and middle combined.

Figure 4.16: A fully opaque sand texture. Middle - a partial grass texture with alpha channels. Right - The resulting texture after mixing both Left and Middle textures together.

This technique is applied to the terrain during the creation of the heightmap mesh. Each height value processed is assigned a value between 0 and 1 based on how close it is to the minimum or maximum heights of the Lunan catchment. A value of 0 applies the additional texture as a fully transparent while a value of 1 applies the additional texture as fully opaque. The closer the height of any given vertex to the maximum height of the terrain the more of the additional texture will be visible.

Aerial Photography

Aerial photography was the final texturing technique to be added to the terrain generation. This method provides a more accurate representation of the land use cover of the surrounding terrain in 2010 (which is the year the agent based model is initialized from). The aerial photography was acquired using the Google Maps API and used to texture the entire landscape (see figure 4.17). The Google Maps satellite imagery consists of tiles which results in larger than required datasets. To crop the image to the exact latitude and longitude of the map corners the image is processed in MicroDEM (Guth, ND).

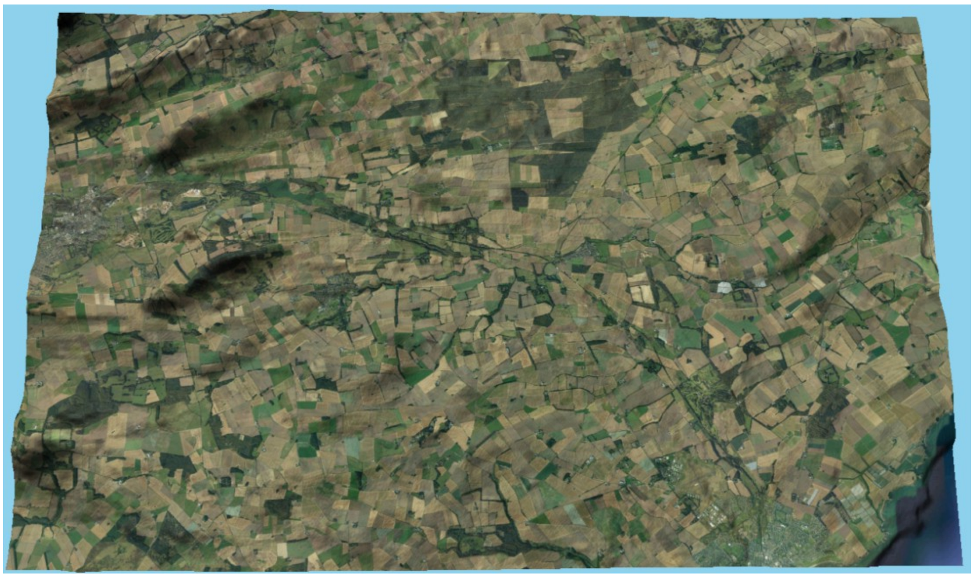


Figure 4.17: The Lunan terrain with accurate geolocation aerial photography.

4.4.2 The Surrounding World

Computer generated terrain is just one element required in building an accurately rendered outdoor scene. Appleton et al. (2002) argue that sky, clouds and shadows are also important considerations when evaluating the realisim of a scene. The free-roaming camera allows large portions of the sky to be viewed at any one time and sky is regarded as a component of realistic representation of outdoor terrains (Appleton et al., 2002). A technique used in computer games known as a skydome is used to render sky in R.S.V.T.

4.4.2.1 Skydome

A skydome is a 3D mesh that comprises to make a half-sphere and can be seen in figure 4.18. This 3D mesh is rendered first, before any terrain is drawn along with a sky texture. One benefit of the skydome is that it only needs one texture and often non-repeating textures can be used without any visual degradation of the skyline.

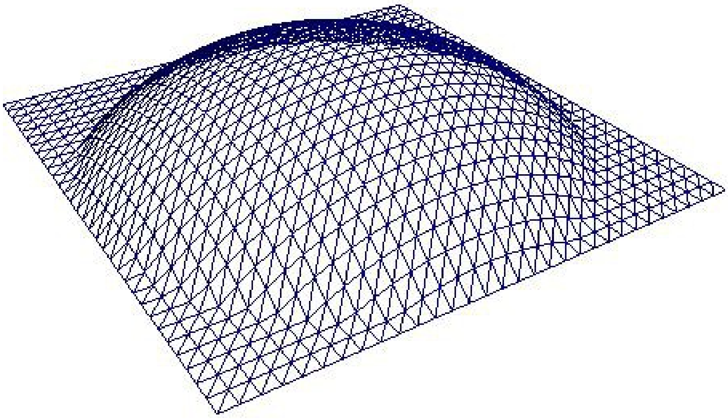


Figure 4.18: A skydome mesh.

The sky must be drawn infinitely far away i.e. located at the far plane and drawn before the terrain and its features. The sky must appear fixed in the scene even whilst the camera is moving through the terrain. This is achieved by translating the camera to the centre of the skydome at each draw frame. This creates the illusion that the sky is infinitely far away.

4.4.2.2 Land Parcels - Shape File Reader

In the previous section three different types of terrain texturing to indicate land use cover in Lunan catchment is discussed. This type of terrain cover is only suitable for the static land parcels in and around the Lunan catchment. The majority of the land parcels within the Lunan area are part of the Agent Based Model and these land parcels will require dynamic updating of its land use cover change. To achieve this, geospatial information of the landparcel i.e. its physical location according to a map projection is required. This information is stored in a shapefile that was previously created for the ABM by researchers at the University of Edinburgh.

A shapefile is a geospatial data format written by ESRI for their popular ArcGIS products (ESRI, 1998). Shapefiles can have more than one shape in them and each shape is a point, line or polygon. The Lunan shapefile contained 1149 shapes (one shape per landparcel) and all shapes were polygon as it is the only way to accurately represent landparcel boundaries. The shapefile holds the geospatial information required to render the landparcel onto the correct location of the terrain along with the attribute table (part of the shapefile format) which is a database holding the starting attributes of each land parcel such as landparcel size, starting land use cover and farmer rights and ownership. Gathering this data was a large part of the previous research and is made up of information gathered from questionnaires and census data that were answered by farmers taking part in the EcoChange Lunan ABM.

The 3D user interface requires that the shapefile be drawn on top of the Lunan terrain and the aerial photography (see **section 4.4.4.1**) overlaying the landscape as it would be simple to determine if the shapefile matches the satellite imagery. However the accuracy of the shapefile proves problematic. The entire shapefile consists of 116,222 points that determines the geospatial location of every landparcel. This would have overloaded the graphics pipeline and slowed the visualisation to an unacceptable framerate especially when considering that more elements other than terrain and land parcels are required to be visualised. Reducing the number of points within the shapefile can be done using the *Douglas-Peucker* algorithm (sometimes known as the *split and merge* algorithm) which can significantly reduce the number of points in a curve. This is done through the use of the online tool MapShaper (MapShaper.com, 2010) which accepts a shapefile as input along with the simplification percentage required. The Lunan shapefile manages approximately 70% simplification without visible loss of the shapefile's integrity and this transformation can be seen in Appendix B. The final shapefile used within R.S.V.T contains only 18,384 points.

ESRI provides a technical whitepaper for developers to help with the creation of custom tools such as shapefile readers and parsers (ESRI, 1998). The whitepaper outlines the ESRI file format of a shapefile and the whitepaper is the basis of R.S.V.T's custom shapefile importer. The creation of R.S.V.T's shapefile importer is the reason for switching from XNA 3.0 to XNA 3.1. XNA 3.1 allows developers to add assets, such as shapefiles, that previously had no way of handling the file format. This allows R.S.V.T to generate the landparcel boundaries as well as their 3D meshes at build time rather than runtime.

The performance increase gained through using the content pipeline is directly related to converting coordinate space into screen space before the visualisation is loaded. The shapefile points are stored as Universal Transverse Mercator coordinate projections. For example, the coordinate below represent the centre of the shapefile:

Latitude: 2.695815

Longitude: 56.633

All the coordinates stored within the shapefile need to be converted from latitude/longitude to x/y screen coordinates. This can be done using linear equations as each term is a constant (see figure 4.19).

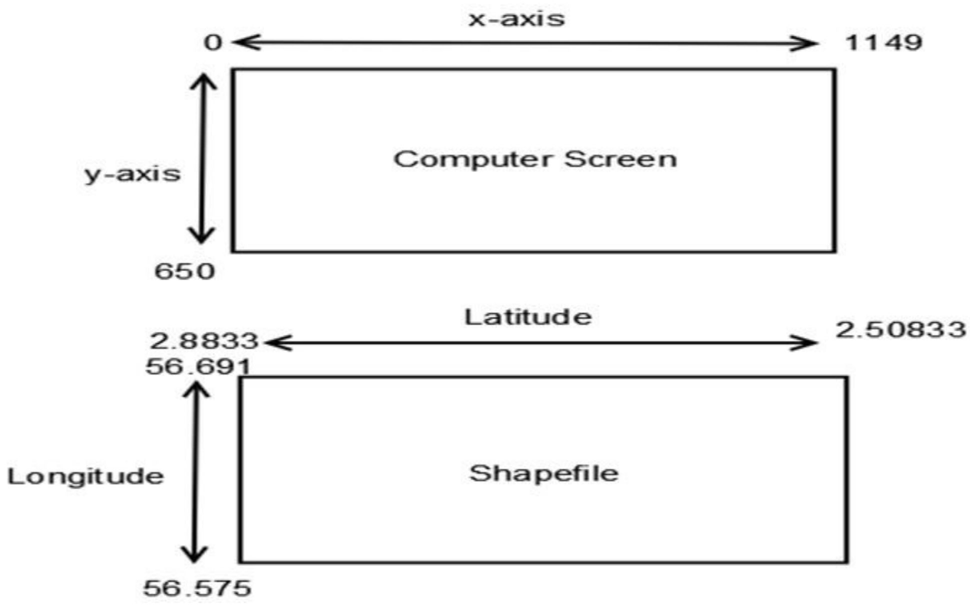


Figure 4.19: The basis of R.S.V.T's linear equation to transform Mercator coordinates to screen coordinates.

Linear Equation

$i = Ax + B$

$j = Ay + B$

where ***i*** is the horizontal pixel and x is the longitude. So, when ***x*** = 2.8833 then ***i*** must equal 0 and when ***x*** = 2.50833 then ***i*** must equal 1149. Where ***j*** is the vertical pixel and ***y*** is the longitude. So when ***y*** = 2.508 then j must equal 1149. ***A*** and ***B*** are shown to be the displacement of latitude and longitude beween each pixel on the heightmap.

Linear Equation:

$0 = A(2.8833) + B \quad 1149 = A(2.50833) + B$

Rearrange:

$A(2.8833) + B = 0 \quad A(2.50833) + B = 0$

Subtract A(2.8833) to get value for B:

$B = 0 - A(2.8833)$

Substitute B:

$A(2.50833) - A(2.8833) = 1149$

Results in:

$A(-0.3749) = 1149$

Divide each side:

$A = -3064.817$

Substitute A to get value B:

$B = 0 - (03.64.817\, x\, 2.8833) \quad B = -8836.786$

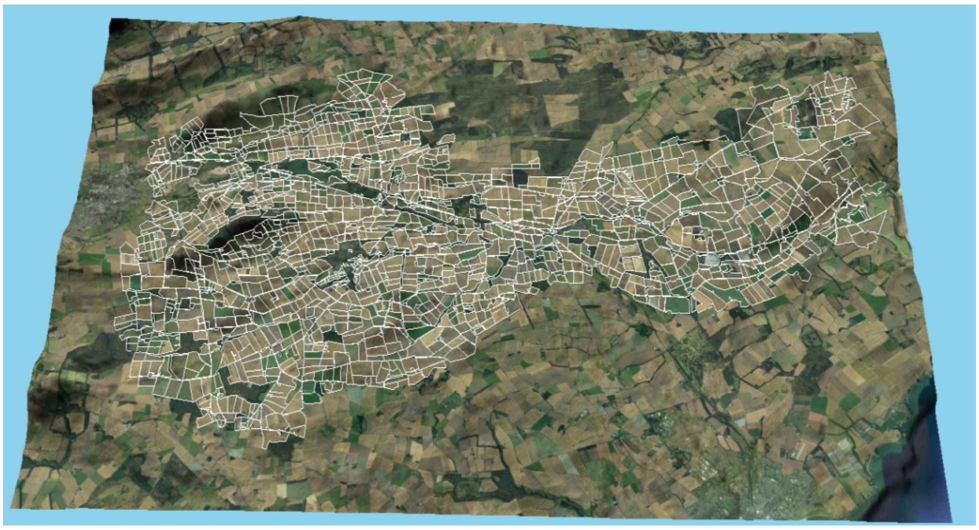


Figure 4.20: The state of the 3D User Interface with aerial photography and land parcel boundaries.

The rendered 3D V.E thus far can be seen in figure 4.20 above. The use of the aerial photography texture overlay and the landparcel boundaries illustrates the Mercator to screen coordinate transformation matches well. However the underlying aerial photography can be seen within the land parcels themselves. The landparcels with visible boundaries are all modelled within the agent based model and a mechanism for displaying the current land use cover of each landparcel, dictated by the model is the focus of next section.

4.4.2.3 Tesslelating the shapefiles

The shapefile content importer, discussed in the previous section, provides a mechanism that allows the 3D user interface to render the boundaries of each landparcel within the agent based model. The 3D user interface is required to dynamically change the data being displayed inside each polygon to accurately reflect the state of the agent based model. To achieve this, the polygon must be tessellated into a series of triangles allowing a 3D mesh to be constructed, rendered and, most importantly, textured. Through the use of tessellation it is possible to replace the underlying terrain texture (single textured grass, multi-textured or satellite imagery) with a custom texture that is determined by the model value and passed to the 3D user interface via the Presenter.

The 3D user interface relies on the *folding ears* method of triangulation (sometimes referred to as the *ear clipping* method. An ear of a polygon is defined as three consecutive vertices V_0, V_1 and V_2 for which no other vertices of the polygon are inside *V* (Eberly, 2002). Any polygon with four or more sides will have at least two non-overlapping ears. Each point of the polygon is stored in a computational data structure known as a linked list. Each vertex V^i and its corresponding triangle (V^i-$_1$, V^i, V^i+$_1$) are stored and every other point is tested to ensure that it is not inside the triangle. If no point is found to be within *V* then an ear is found. This is done recursively until there are no more ears to be found (see figure 4.21) and you are left with a triangulated polygon (see figure 4.22).

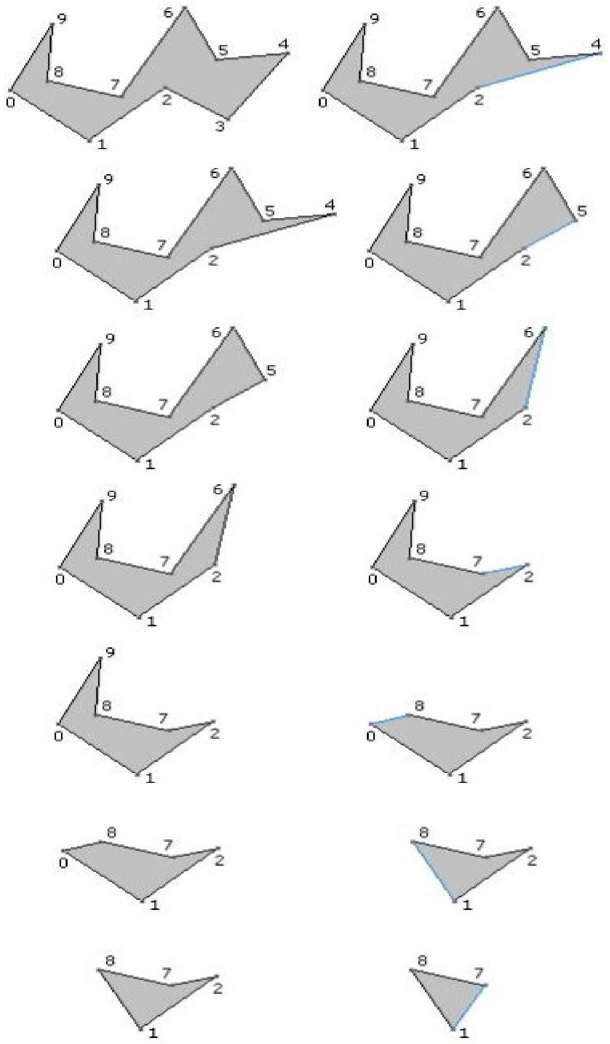


Figure 4.21: Top left shows the original polygon while top right shows the polygon with one ear folded (ear 2, 3 and 4). This is a recursive process and the images below illustrate the steps taken to reduce the polygon to one ear.

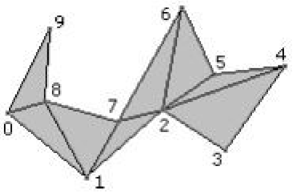


Figure 4.22: The triangulated polygon, made up of all its folded ears.

A folding ears implementation is written into R.S.V.T's shapefile importer allowing the triangulation process to take place at buildtime rather than runtime. After the processing of the shapefile is complete, the 3D user interface has mesh models for each land parcel that is referenced within the model. Importantly this allows dynamic texturing of the landparcels to represent the current state of land use cover, as dictated by the Model via the Presenter (see **section 4.4.3.4 colour mapping** and **texture mapping**).

4.4.3 User Interface Design Elements & Visualisation Techniques

The user interface design elements are created with the literature on semiotics in mind (see **chapter 2.5.2**). The design elements can be split into two groups of visual imagery:

- Visual Imagery for representing land use cover
- Visual Imagery for representing multivariate individual/regional land parcel sustainability

Visual Imagery for Representing Land Use Cover

Three separate techniques were developed to represent the dynamic land use cover of the Lunan region. These techniques are colour mapping, texture mapping and 3D crops. Both colour mapping (see **figure 4.23**) and texture mapping (see **figure 4.25**) employ an indexical mode of visual imagery. Both techniques display a complimentary colour/texture legend making it easy for user to quickly look up the land use type without having to commit to memory which colour/texture maps to which land use type.

The 3D models (see **figure 4.27** & **figure 4.28**) are examples of iconic visual imagery, creating a direct link between sign and meaning with no previous learning or knowledge required from the user before being able to interpret the sign.

Visual Imagery for Representing Multivariate Sustainability Data

The display of multivariate sustainability data is packaged into three visualisation techniques. The techniques pillared view and exploded view (see **chapter 4.4.3.2**) provide the user with sustainability information for an individual land parcel. The pillared view uses symbolic visual imagery to convey the data to the user through the use of 3D bar charts (see **figure 4.30**). Although symbolic signs require some form of previous learning or knowledge it was felt that bar charts were a common enough form of symbolically conveying data to merit its use.

The exploded view (see **figure 4.2.9**) implements layering and separation of data as a way to order data and emphasise the more important content whilst de-emphasising the less important content (Zimmermann, 1997). Finally, the regional land use sustainability is calculated daily and reported as a small multiples visualisation (see **figure 4.3.1**). This allows the user to quickly see the differences between each day (and across separate model simulations when used in split-screen mode).

4.4.3.1 Visualising Land Cover

The land use cover type forms part of each message sent by the Presenter based on the current state of the agent based model. The land use cover data can be broken down into:

1) The current land use type
2) The height of the current crop

The 3D user interface requires visualisation methods capable of portraying this information to the user. The 3D user interface makes use of three different visualisation techniques to achieve this. They are

1) Colour mapping
2) Texture mapping
3) 3D models

With the exception of 3D models, all 10 major land use types that are modelled in the agent based model are visualised. These land use types are *barley*, *field beans*, *carrots*, *grass*, *oats*, *peas*, *succession* (no current land use), *potatoes*, *turnips* and *wheat*. The remainder of this section will discuss each of the visualisation methods in turn.

Colour Mapping

The colour mapping visualisation technique that is implemented in the 3D user interface allocates a colour to each of the land use types. This satisfies the Presenter's message requirement that the current land use type is dealt with. The Presenter message also insists that the height of the current land use type is sent. To satisfy this demand the colour mapping uses an alpha transparency slider to determine how opaque the colour should be based on height as seen in figure 4.23. Figure 4.24 shows the colour mapping technique implemented on the Lunan terrain during a live model run.

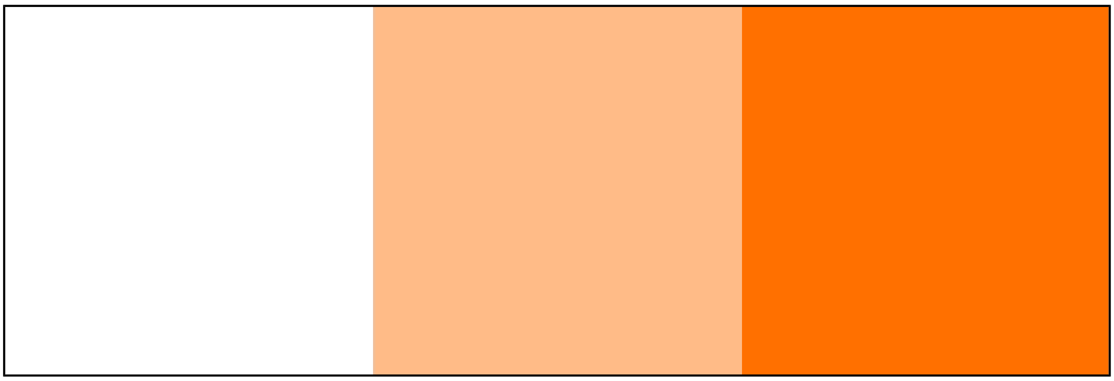


Figure 4.23: An example of the colour mapping technique where the colour depicts crop type (carrot in this example) and opacity mapped to crop height land use cover type at 0% height, at 50% its maximum height and at 100% its maximum height.

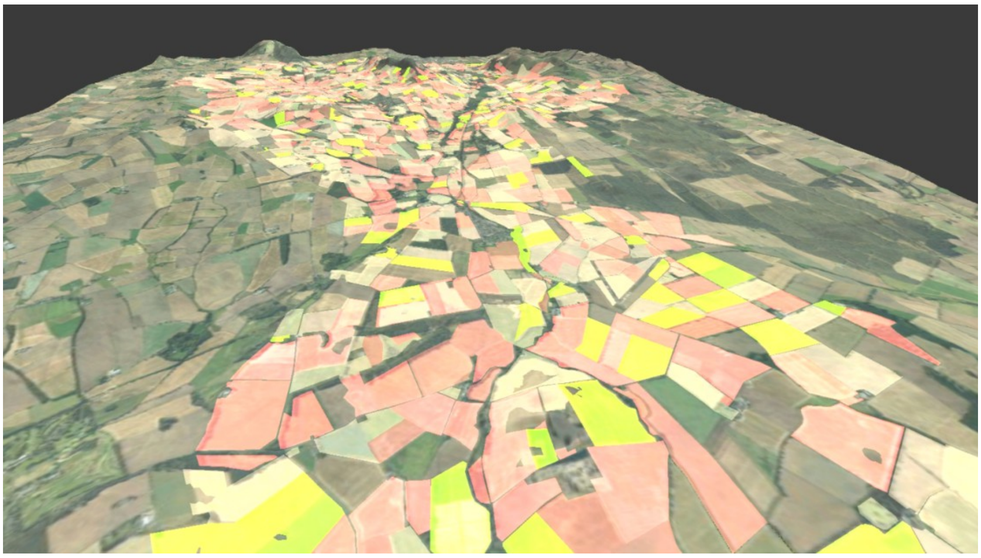


Figure 4.24: The Lunan catchment with colour mapping. All landparcels are representative of data passed to it by the Presenter (Yellow=Barely, Pink=Succession, Green=Setaside Grass)

Texture Mapping

The texturing mapping applies the same principles as the colour mapping but replaces colours with textured images. Again, an alpha transparency slider is used to determine how much of the texture if visible on the 3D mesh that makes up the landparcel as seen in figure 4.25 and figure 4.26.

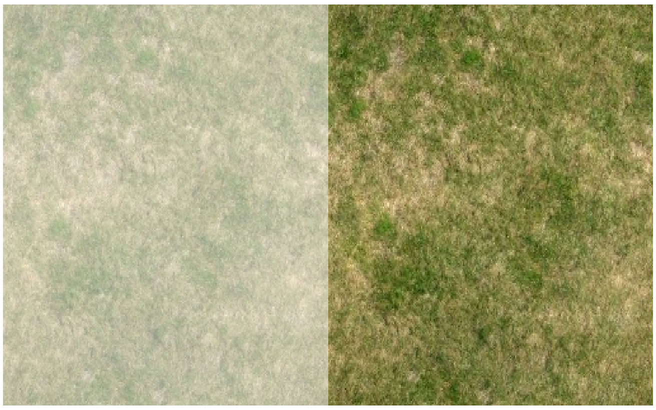


Figure 4.25: An example of the texture mapping technique where the texture depicts crop type (grass in this example) and opacity mapped to crop height land use cover type at 0% height, at 50% its maximum height and at 100% its maximum height.

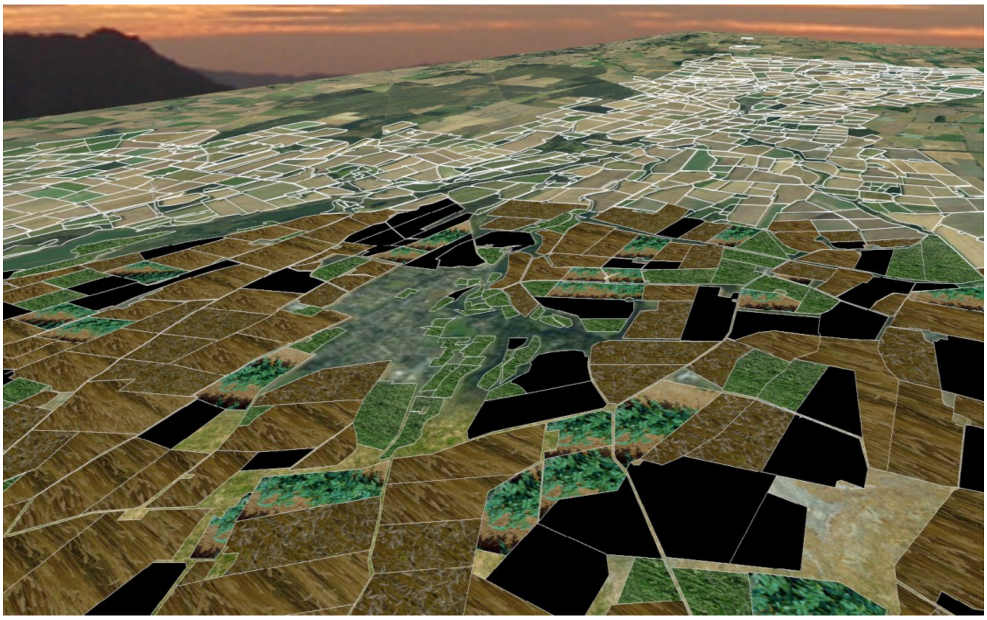


Figure 4.26: The Lunan catchment with texture mapping. All landparcels are representative of data passed to it by the Presenter.

3D Models

The third visualisation technique uses dynamic 3D models rather than simply a texture or colour map. This technique was introduced to ensure that the 3D User Interface has features similar to those found in Visual Nature Studio (see chapter **2.3.2**) and ensures that users of the system have a variety of visualisation options when viewing land use cover. To convey crop growth, R.S.V.T makes use a mesh/model swapping, a technique that swaps a 3D mesh/model for another 3D mesh/model and can be seen in figure 4.27 and figure 4.28. The crop model starts out as the mesh on the left and switches to the centre mesh at 35% of its maximum height. Finally, the mesh switches to the fully grown crop at 75% of its maximum height. Each crop mesh grows into the Y-Axis based on the Presenter's crop height message.

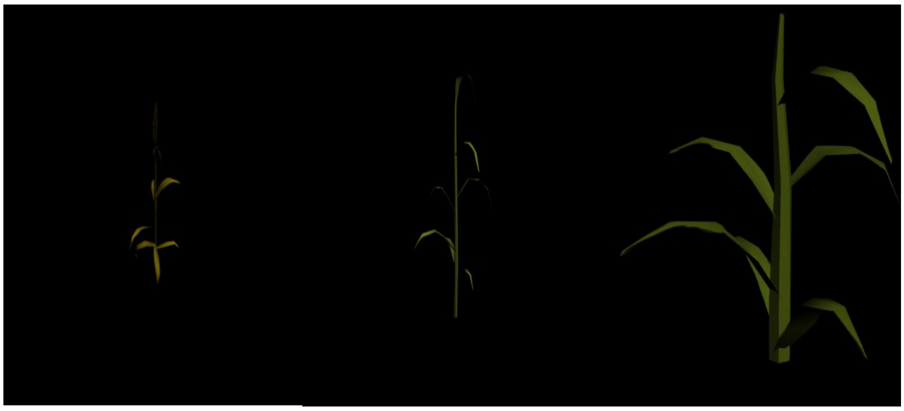
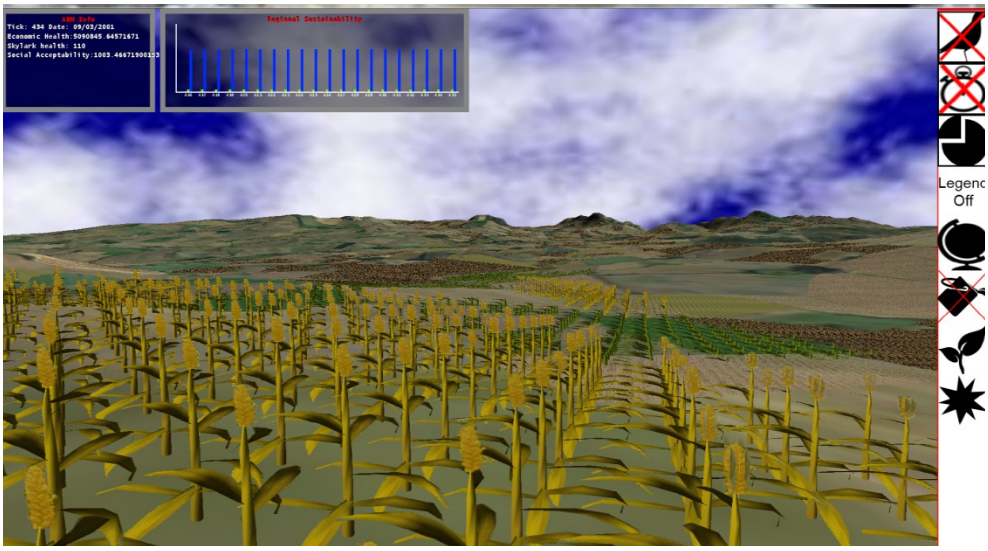


Figure 4.27: Left: The original mesh. Middle: The mesh swapped to after 35% of height is reached. Right: The mesh swapped to after 75% of height is reached.



Tick: 434 Date: 09/03/2001
Economic Health:5090845.64571671
Skylark health: 110
Social Acceptability:1003.46671900153

Figure 4.28: The 3D user interface displaying growing crops based on the Presenter's message. In this example, the crop type is barley and its height is at 50%.

The 3D crops use a technique known as hardware instancing to boost performance. This technique uses the power of the GPU to create many thousands of copies of a model without requiring a *Draw* call to be made to the GPU for each one. It is a popular method of mass visualisation of complex model such as crowd simulation (Ashraf, 2006) and real-time particle systems

(Drone, 2007). Covering the entire Lunan terrain with hardware instanced models is not something that is yet feasible and therefore the decision to limit the number of land use types visualised was taken.

Land use cover change is only a part of the Presenter message that drives the 3D user interface, the Model also calculates key model indicators at the end of each modelled day to better understand the current state of each landparcel in relation to its economic, environmental and social output. The following section discusses this in more detail.

4.4.3.2 Visualising Key Model Indicators

At the end of each day within the agent based model an indicator value between 0 and 1 is calculated for each landparcel in the context of economic, environmental and social indicators (see **section 3.5.3** and **section 3.6** for more information). This number is based on a number of elements:

- The current economic value of the landparcel based on crop type, size and yield. All economic values are based on a market curve, projecting market values from 2000 – 2050.
- The current environmental value of the landparcel based on crop type, bio-diversity within the landparcel (this is heavily influenced by the number of skylarks reported in the landparcel).
- The social value of the landparcel which is a single indicator based on the crop type.

The 3D user interface allows users to see these indicators as visual representations in two ways: **Exploded View** and **Pillared View**.

Exploded View

The exploded view is available to the user once they click on a particular landparcel they wish to query. Upon selecting the exploded indicator three additional meshes, identical to the shape of the queried landparcel appear vertically above the original, this is known as *exploding* and can be seen in figure 4.29. Each of the newly created *exploded* land parcels have a colour designation to identify which indicator it relates to. These colour indicators are:

Red: The economic output of the landparcel.

Green: The environmental output of the landparcel.

Blue: The social output of the landparcel.

The indicator value itself is represented by the opaqueness of each *exploded* landparcel's colour.

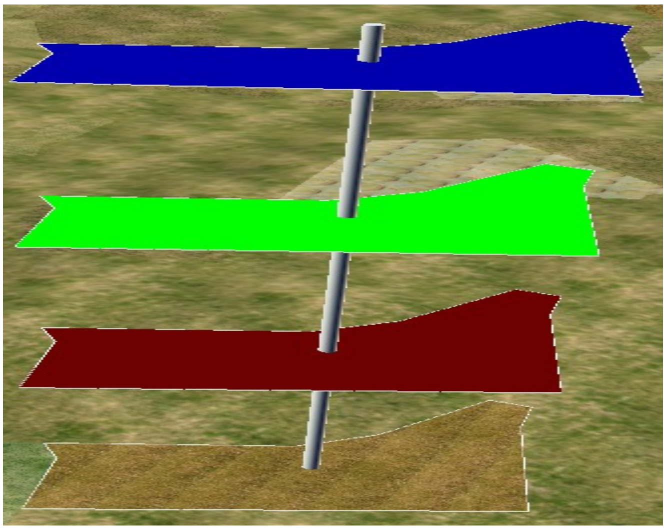


Figure 4.29: A landparcel making use of the exploded view.

Pillared View

As with the *exploded view* the *pillared view* option is available to the user once they have selected the landparcel they wish to query. Once the *pillared* option is selected three cylindrical pillars appear around the centre of the landparcel as seen in figure 4.30. Each pillar has a colour assigned to identify which indicator it relates to. These colours are the same as the *exploded view*.

Red: The economic output of the landparcel.

Green: The environmental output of the landparcel.

Blue: The social output of the landparcel.

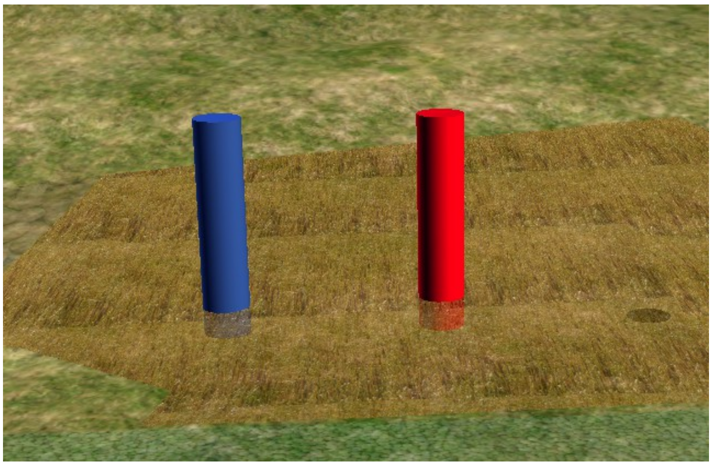
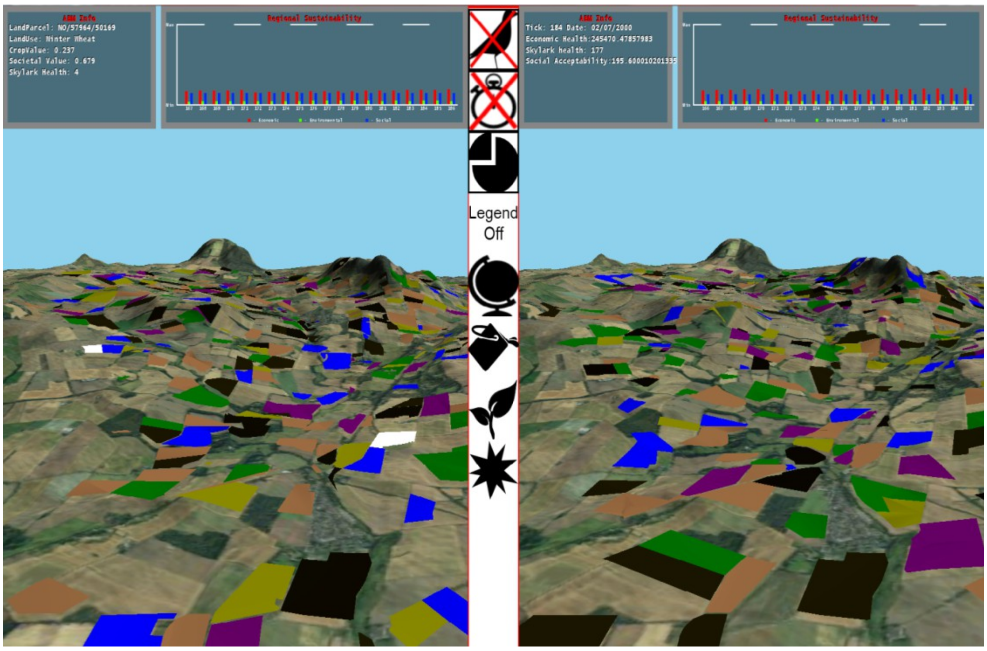


Figure 4.30: A landparcel making use of the pillared view.

SplitScreen Mode

R.S.V.T allows the user to view two different model simulations side by side. This satisfies requirement 9 of the refined requirements specification which states **Users can compare future scenarios**. R.S.V.T vertically splits the screen into two views of equal size. Each view is capable of rendering the ABM output of a loaded model and the model can be in either online-mode or offline-mode. Through the presentation of multiple scenarios it encourages stakeholders to compare the outcomes of different scenarios, providing a platform for discussion and debate. It is argued that the addition of this feature fulfils the requirement for positive learning outcomes which is a feature of V.E's.



ABM Info
LandParcel: NO/57964/50169
LandUse: Winter Wheat
CropValue: 0.237
Societal Value: 0.679
Skylark Health: 4

ABM Info
Tick: 184 Date: 02/07/2000
Economic Health:245470.47857983
Skylark health: 177
Social Acceptability:195.600010201335

Figure 4.31: Splitscreen view. Left view and right view showing different scenarios.

4.4.4 Summary of the View – The 3D User Interface

The 3D User Interface is largest component of the R.S.V.T system. In **section 4.4.1** the importance of understanding the technical specifications of heightmaps was discussed. This was followed by an introduction to simple terrain generation procedures known as bruteforce terrain in **section 4.4.2** and then optimized the terrain for large and seamlessly terrains using quadtrees and a Level-of-Detail algorithm. Finally the different visualisation methods and how they were constructed (**section 4.4.3**) with the implementation of terrain texturing, shapefile rendering, mesh building through triangulation, land use cover visualisation and the visualisation of key model indicators. The construction of the 3D User Interface conforms to the all but one of the remaining requirements specifications:

4.5 R.S.V.T – A System Summary

R.S.V.T, the system, satisfies the requirements specification laid out in the refined requirements section (see **section 4.3.4**).

Requirement 1: Real-time loose coupling of Model and Visualisation

The Model-View-Presenter promotes component independence and this was shown to be useful in building loosely coupled systems. The visualisation has been shown to implementation of a binary based messaging format such as Protocol Buffers couples the Model to the visualisation. The pilot testing (see **Chapter 5**) confirms the model is loosely coupled.

Requirement 2: Present ABM results as 3D visualisation techniques

Multiple methods of 3D techniques are created from ABM data including the terrain, land parcels, skylarks, land use types and abstract methods of data visualisation such as exploded views and 3D pillars (see **section 4.3.4**).

Requirement 3: 3D interface is just as fast as 2D interface

The pilot testing (see C**hapter 5**) confirms that the 3D visualisation tool is faster than the 2D tool.

Requirement 4: 3D interface is interactive, representational and immersive

The 3D interface has shown it has visualisation methods which are representational and immersive such as terrain with aerial photography texturing, dynamic 3D crops and birds. The user can move freely though the terrain, change visualisation methods and start/stop simulations. This ensures the visualisation is interactive.

Requirement 5: To communicate the social, environmental and economic impact of land management strategies

Social, economic and environmental data is shown to be passed from the model to the visualisation in **section 4.3** and this data is used as input for visualisation methods including land use cover rendering (see **section 4.4.3**).

Requirement 6: LOD techniques to improve loading and rendering of large terrains

The pilot testing highlighted the need for an optimised method of large terrain generation. **Section 4.4.1.4** shows the series of steps required to create large scale interactive terrains (quadtrees, terrain stitching).

Requirement 7: User can run R.S.V.T from a real-time model or a previously run model.

The development of the flat-file database (see **section 4.3.5.1**) and the implementation of the MongoDB database (see **section 4.3.5.2**) provide a storage facility for recording interesting simulation results that can be replayed within R.S.V.T as required.

Requirement 8: Users can compare future scenarios

The implementation of the splitscreen feature is found at the end of section 4.4.3. Its addition allows stakeholders to view two unique simulations under different scenario conditions. A combination of the storylines presented in **Chapter 3** can be loaded into R.S.V.T to better visualise the difference between plausible future scenarios.

Chapter 5 – Evaluation of Network Communications

This chapter discusses three Web Service technologies with particular emphasis on the format of data that each Web Service uses to communicate with the client-side application of R.S.V.T – the 3D VE. The chapter starts with an explanation of what Web Services are and why they are important. The three Web Services are identified, namely Simple Object Access Protocol (SOAP), Thrift and Protocol Buffers and their communications protocols. This chapter concludes with justification of the chosen communication protocol - Protocol Buffers. This is demonstrated by comparing the speed of data transfer of the three communication protocols together with a feasibility test that shows Protocol Buffers implemented in R.S.V.T meets two project requirements.

5.1 The Need for Web Services

A web service is defined by the W3C (World Wide Web Consortium) as *"a standard means of interoperating between different software applications, running on a variety of platforms and/or frameworks"* (W3C, 2004). Although the word *"web"* is specified in the term it applies to all forms of application interoperation over any network, not just the Internet. Web services provide a way to handle the expected efficiency of modern applications which depend on the cooperation of other online or network enabled services (Guinard & Trifa, 2010). This *"expected efficiency"* of modern applications is in a large part due to the push towards mobile technologies.

The last few years have seen Internet traffic from mobile and tablet devices increase. It is now estimated that 15% (7% smartphones + 8% tablets) of all Internet traffic is executed from mobile devices (Adobe, 2013; Forbes, 2013). Tablets have only been widely available for three to four years and they have already taken over smartphones in the percentage of global Internet traffic. Cisco (2013) published its traffic forecasting for 2012 – 2017 using data gathered during 2000 – 2012. The most notable points are listed below:

- Global mobile traffic grew 70% in 2012 compared to 2011.
- Smartphone Internet traffic has grown 81% in 2012. The average smartphone Internet traffic was 342MB per month in 2012 compared with 189MB per month in 2011.
- The number of mobile-connected tablets increased by 250% taking the total number of tablets to approximately 36 million. Tablet Internet

traffic was over double the usage of smartphone with approximately 820MB per month compared to smartphones 342MB per month.

The rapid increase in mobile Internet traffic has caused many software service providers to rethink their approach to Web Services (Berners-Lee, 2010). At a time when desktop computers were the primary target of Web Services, there was little demand for optimising the speed and size of the data being relayed to the client. Developers targeting mobile users as well as desktop users have to rethink their method of software delivery. One particular method of software delivery is closely related to Web Services and that is the delivery model of Software-as-a-Service.

5.1.1 Software-as-a-Service

Software-as-a-Service (*SaaS*) is a software delivery model with its purpose defined as a model to separate possession and ownership of software from its use (Turner et al., 2003). It is sometimes called a *subscription service*. The goal of *SaaS* is to keep ownership of software with developers; the developers then permit user access to the software on an on-demand basis via a client-side application.

One example of a software-as-a-service is Google Maps. A web developer can embed a Google map image into a website using a variety of different frameworks and can be run on any platform. In this example Google retains all ownership of its data and the processing takes place on Google's servers. The client is able to demand a map service that Google will provide. This allows clients to interact with data and services without incurring the costs of buying or leasing the data and processing the datasets.

5.1.2 R.S.V.T – A Web Service using SaaS

The terminology of the W3C's definition of Web Services is directly comparable to the components that make up R.S.V.T. For clarity; the definition is *"a standard means of interoperating between different software applications, running on a variety of platforms and/or frameworks".*

R.S.V.T's Software ApplicationsR.S.V.T is made up of two software applications, the agent based model (see **chapter 4.2**) and the 3D User Interface (see **chapter 4.4**).

R.S.V.T's Platforms

Both software applications above are written under different platforms. The agent based model is developed on OS/X to run in both OS/X and Windows environments while the 3D User Interface is developed on, and is run using, Windows.

R.S.V.T's Frameworks

Both software applications are being developed under different frameworks. The agent based model is developed under Repast Simphony - a Java based framework tailored to developing agent based models. The 3D User Interface is developed using XNA – a C# framework aimed at developing advanced 2D and 3D graphical applications.

R.S.V.T's SaaS

The agent based model (see **chapter 4.2**) adopts the *SaaS* delivery method while the 3D User Interface (see **chapter 4.4**) is equivalent to the client-side application.

By listing the components above it is shown that **R.S.V.T.** shares common functionality with a typical Web Service. What is not listed above is *how* the interoperability between applications is done. This is done through a Web Service's communication protocol.

5.2 Exploration of Communication Protocols for Web Services

This section examines three Web Services: SOAP, Thrift and Protocol Buffers. SOAP makes use of XML as a messaging format between server and client while both Thrift and Protocol Buffers use their own binary encoding to form messages that are passed from server to client.

5.2.1 Simple Object Access Protocol and XML Encoding

SOAP is a protocol for the exchange of information in a decentralised environment (W3C, 2004). It makes use of eXtensible Markup Language

(XML) to construct *envelopes* which are made up of a header and a message body as seen in figure 5.1.

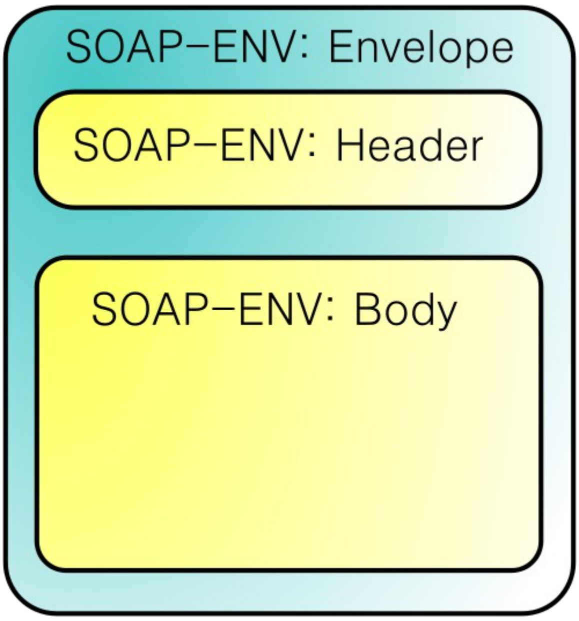


Figure 5.1: A visual representation of a SOAP message envelope. Its components consist of a header and a message body

All SOAP messages (both header and body) are encoded in XML. This makes the messages somewhat easier to understand as XML is a human readable language. The SOAP header component contains instructions on how to interpret the data contained within the body. The SOAP body provides a mechanism for exchanging mandatory data between applications. This is done through invoking Remote Procedure Calls. Remote Procedure Calls allow other applications to execute subroutines on a non-native codebase. An example landparcel SOAP/XML message that was used by R.S.V.T can be seen in figure 5.2.

Figure 5.2: Example code snippet of the SOAP/XML message for a land parcel

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <LandParcel xmlns="rsvt.landparcel">
      <landParcelID>NO/4344/3324</landParcelID>
      <ownerID>100222</ownerID>
      <landUse>5</landUse>
      <vegetationHeight>3.432</vegetationHeight>
    </LandParcel>
  </s:Body>
</s:Envelope>
```

in R.S.V.T

The size of this message is 298 bytes. Each character is encoded using UTF-8, which requires at least 1 but no more than 8 bytes. It is fully dependent on the character it is encoding. If the character has a corresponding ASCII value then only one byte is needed. In the case of the message above, each character is one byte in length.

5.2.2 Thrift with Binary Encoding

Thrift was designed as a fast messaging protocol by engineers at Facebook and later moved to Apache Incubator to allow open source developers to expand on its potential (Apache, 2010). Developers at Facebook wanted a faster way of transferring data between server/clients. This led to the creation of a binary encoding which is automatically generated using a platform independent language named Thrift. It is designed around messages consisting of 7 base types. These are:

- Bool – A Boolean value (true/false)
- Byte – An 8-bit signed integer
- i16 – A 16-bit signed integer
- i32 – A 32-bit signed integer

- i64 – A 64-bit signed integer
- Double – A 64-bit signed integer
- String – A text string encoded using UTF-8 (min. 1byte: max. 8bytes per character)

An example landparcel message in Thrift that was used by R.S.V.T can be seen in figure 5.3 below:

```
struct LandParcel
{
    1: String landParcelID //(12 characters/96 bytes)
    2: optional Byte ownerID //(1 byte)
    3: optional Byte landUse  //(1 byte)
    4: optional Double vegetationHeight //(8 bytes)
}
```

Figure 5.3: A land parcel message written in Thrift. Total message size ~ 100 bytes.

The actual size of this message can vary between a minimum of 96 bytes and a maximum of 106 bytes. This is due to the *optional* keyword present in values 2, 3 and 4. The optional keyword indicates to the Thrift compiler that data may or may not be sent. If the optional field has no value attached to it during transmission then it is not sent. This can significantly reduce the size of a message whilst offering flexibility when defining message specifications.

5.2.3 Protocol Buffers with Binary Encoding

Protocol Buffers is a binary messaging format written by Google (2008). Its intended use was to replace Google's internal XML messaging service with a smaller, faster and simpler mechanism for serialising structured data. Protocol Buffers follows a similar binary encoding to Thrift although with a greater number of base types:

- Double – A 64-bit floating point number
- Float – A 32-bit floating point number
- Int32 – A variable length encoding for 32-bit integers. Positive or negative numbers.
- Int64 – A variable length encoding for 64-bit integers. Positive or negative numbers.
- Uint32 – A variable length encoding for 32-bit integers. Suitable for positive numbers only.
- Uint64 – A variable length encoding for 64-bit integers. Suitable for positive numbers only.
- Sint32 – A variable length encoding for 32-bit integers. Suitable for negative numbers only.
- Sint64 – A variable length encoding for 64-bit integers. Suitable for negative numbers only.
- Bool – A Boolean value (true/false).
- String – A text string encodied using UTF-8 (min. 1byte: max. 8bytes per character)

The most notable difference in Protocol Buffers is that unsigned and signed integers can be dealt with separately. Thrift assumes everything is a signed integer. The other difference is the variable-length of all *int* data types. This allows small numbers being sent over an int32 to be shortened to one or two bytes rather than using up a full 4 bytes. This can be of great benefit if a large amount of small integers are required in a message. By using variable-lengths integers the size of each message can be greatly reduced.

As with the previous two examples, the protocol buffers message that encapsulates the data from an R.S.V.T landparcel can be found in figure 5.4 below.

```
package RSVTCommMessages;
message LandParcelMessage
{
	required string landParcelID = 1; //(12 character/96 bytes)
	optional int32 ownerID = 2; //(min 1 byte: max 4 bytes)
	optional int32 landUse = 3; //(min 1 byte: max 4 bytes)
	optional double vegetationHeight = 4 [default = 0]; //(8 bytes)

}
```

Figure 5.4: R.S.V.T's land parcel message using Protocol Buffers with a total message size of ~ 100 bytes

The message size of a Protocol Buffer message is, like Thrift, variable. This is again due to the “optional” keyword present for fields 2, 3 and 4. These fields are not sent as part of the message if no data is contained in them.

5.3 Evaluation of SOAP, Thrift and Protocol Buffers

The previous section introduced the communication protocols; SOAP, Thrift and Protocol Buffers. The differences in message size are highlighted but message size is one of three indicators of actual performance. The other 2 indicators are the amount of time it takes to serialize the message into an object for transmission and the amount of time it takes to deserialize the object once it has reached its destination. Microsoft's Query Performance Timer (Microsoft, 2012) is used to calculate the deserialization times and the *System.NanoTime();* function exposed by the HotSpot VM (Oracle.com, 2006) is used to calculate the serialization times for the three data transfer methods.

The data in its original format can be seen in table 5-1 below and as a graph in figure 5.5.

	SOAP	Thrift	Protocol Buffers
Creation (ns)	167.87	221.7	470.51
Serialization (ns)	25673.13	7332.88	7223.31
Deserialization(ns)	68998.2	7877.33	4331.6
Total (ins)	94839.2	15431.91	12025.42

Table 5-1: The results of timing each communication protocol at creating, serializing and deserializing the land parcel message. Times are in nanoseconds

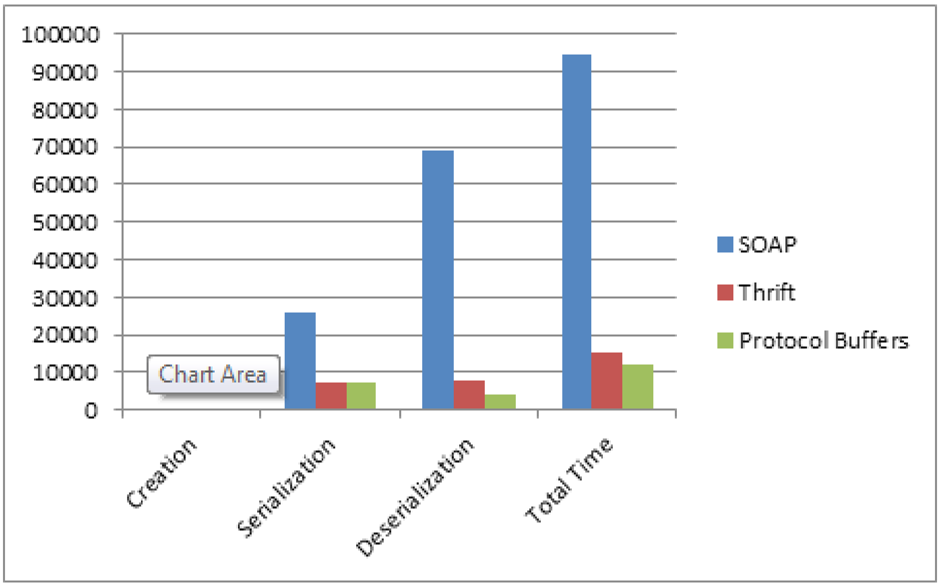


Figure 5.5: A bar chart to illustrate the time differences between the three communications protocols. Times are in nanoseconds.

Table 5-1 and figure 5.5 show that SOAP has the longest times for the serialization and deserialization of data. The one area in which SOAP outperforms both Thrift and Protocol Buffers is in object creation. This is because object creation is done natively, on the server-side (Java) or the client-side (C#). Either way, object creation is extremely fast in both Java and C# due to the object already being in-memory, allocating it to another area in memory is trivial for a high-level programming language.

Serialization is fastest with Protocol Buffers, taking only 7223.31 nanoseconds (0.072 milliseconds). This is closely followed by Thrift which takes 7332.88 nanoseconds (0.073 milliseconds). The slowest serialization is done by SOAP taking 25673.13 nanoseconds (0.025 milliseconds).

The results of the deserialization follow a similar pattern with Protocol Buffers recording the fastest time of 4331.6 nanoseconds (0.0043 milliseconds). Thrift was second ranked, taking 7877.33 nanoseconds (0.0078 milliseconds) to deserialize a landparcel message. Again, SOAP is the slowest of the three communication protocols as it takes 68998.2 nanoseconds (0.0689982 milliseconds) to deserialize its XML-based message.

The total time it takes for a message to be created, serialized then deserialized is calculated by summing the previous results together. By this definition Protocol Buffers are the fastest communications protocol, taking 12025.42 nanoseconds (0.012 millseconds). Thrift is only slightly slower, taking 15431.91 nanoseconds (0.015 milliseconds) to create, serialize and deserialize a message.

Although SOAP's total time of 0.095 milliseconds seems insignificant it is important to remember that this is **one** message that contains information relating to **one** landparcel. There are 1149 landparcels contained within the model that will require updating at every model tick. There will also be a variable number of skylarks modelled, these will require messages also. A final graph showing the times to create, serialize and deserialze 1149 landparcels can be found in figure 5.6 below (times in milliseconds).

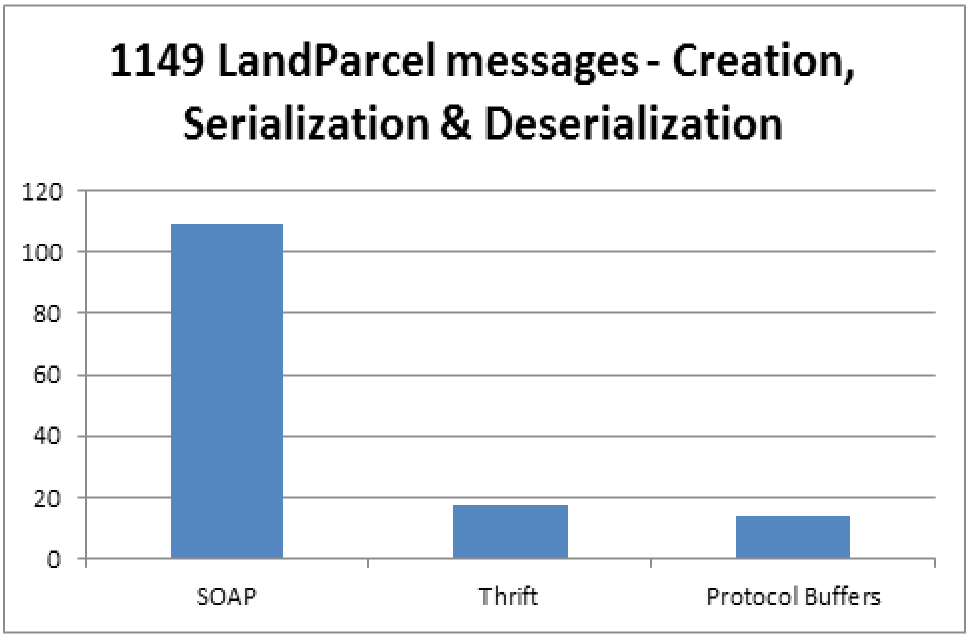


Figure 5.6: A comparison of the total time taken to create, serialize and deserialize messages for each landparcel within the agent based model (times in milliseconds)

SOAP takes 110 milliseconds compared to Thrift's 17.2 milliseconds and Protocol Buffer's 13.8 milliseconds when creating, serializing and deserializing the messages of 1149 landparcels. The longer timings (and poorer performance) of SOAP is directly related to the size of its messages. By using XML, SOAP is not optimized for large amounts of small messages. Thrift and Protocol Buffers both show potential for the fast communication of data between applications.

5.4 Evaluation of RepastS 2D ABM Output and R.S.V.T

The previous section highlighted the performance differences between SOAP, Thrift and Protocol Buffers. This section presents an experiment to evaluate if R.S.V.T meets two project requirements. These requirements are:

1) That R.S.V.T is loosely coupled. Data from the model is never directly accessed or changed by the 3D User Interface.
2) That R.S.V.T is capable of running simulations just as fast as Repast Simphony's 2D User Interface.

Based on the previous results R.S.V.T's choice of data transfer is Protocol Buffers. There are few differences between Protocol Buffer's and Thrift with both performing exceptionally well against traditional XML-based communications. However, over 99% of deserialization is done by the 3D User Interface and Protocol Buffer's fast deserialization is the justification to use Protocol Buffers as R.S.V.T's communication protocol.

5.4.1 Methodology

An experiment was carried out to test if R.S.V.T, using Protocol Buffers, is comparable in speed and performance with its 2D counterpart. Given the potential value three dimensional graphics can bring to end-users R.S.V.T is aiming for performance parity with the 2D User Interface. This would ensure R.S.V.T is a viable alternative to the 2D systems already available to agent based modellers and their stakeholders.

The experiment entailed running an identical ten year model simulation in R.S.V.T and its 2D counterpart Repast Simphony and timing the length of time it took to complete. This was done by setting the same starting random seed each time the simulation was run. A laptop with commodity hardware was used (Quad-core Intel Core i5 Processor @ 2.27GHz, 4GB RAM, NVIDIA GeForce GTX 260M) to ensure the heavy CPU demand of the agent based model was met as well as the significant graphical output being rendered via the graphics card.

For parity, both visualisations were run without any user input for 3652 ticks (each tick is representative of one model-simulated day). The 3D camera in the 3D User Interface was positioned in such a way that mimicked the top-down view of the Repast Simphony 2D User Interface as seen in figure 5.7 and figure 5.8.

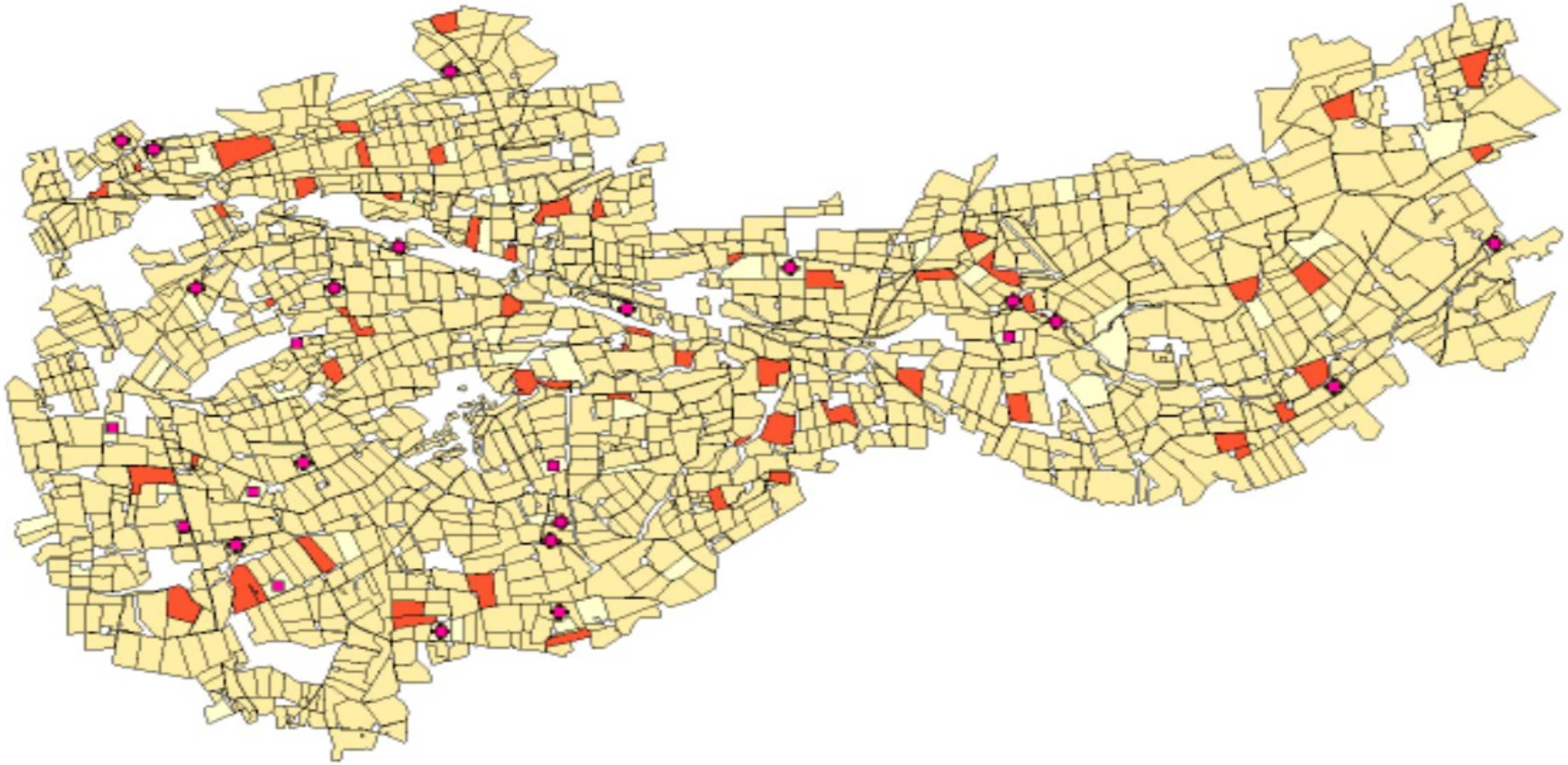


Figure 5.7: The 2D output of the Lunan ABM, visualised through the ABM toolkit RepastS

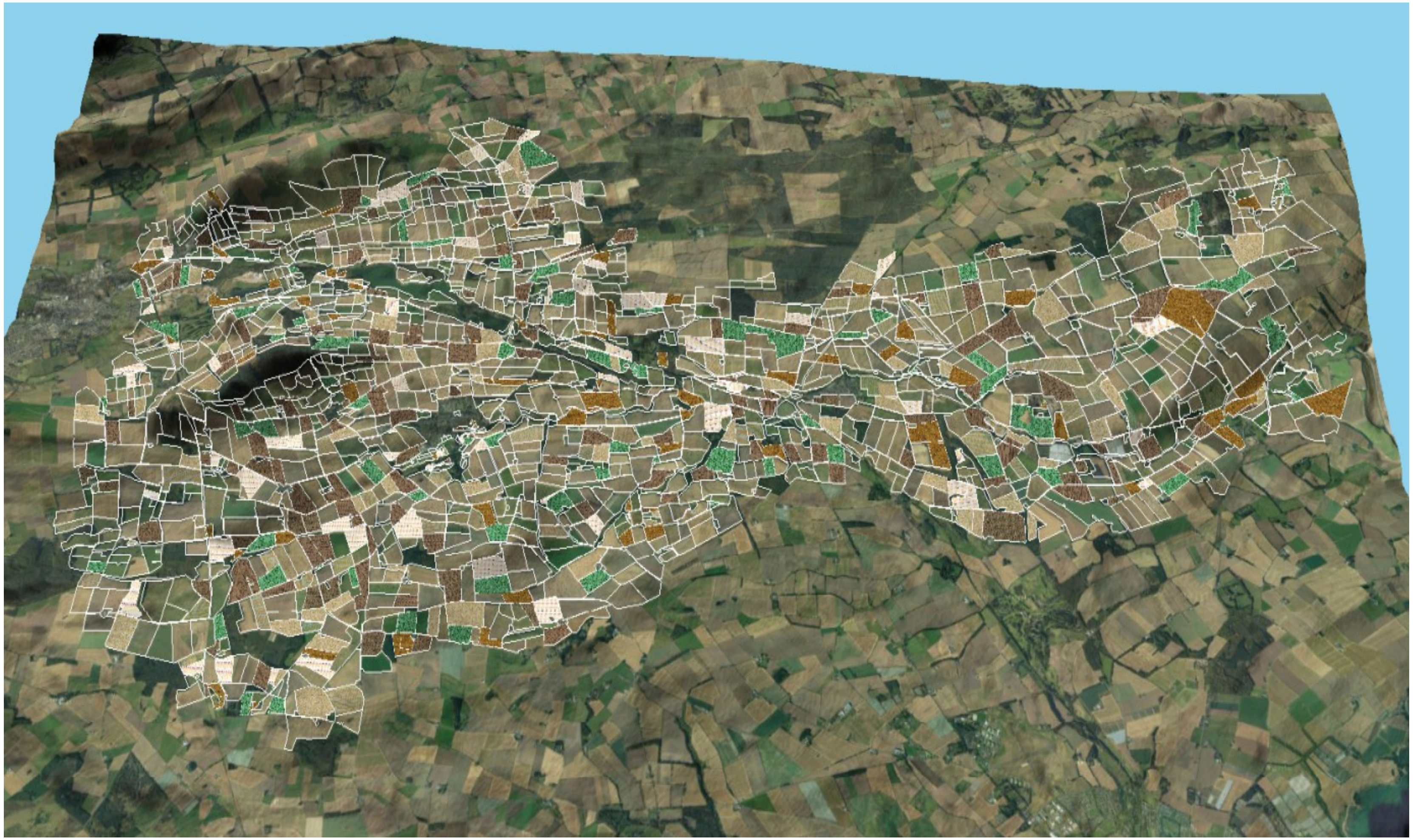


Figure 5.8: The 3D output of the Lunan ABM, visualised through R.S.V.T

Figure 5.8 shows the state of the 3D User Interface at the time of testing. The components that make up the 3D Interface are:

1) Bruteforce terrain
2) Shapefile overlay
3) Land parcel mesh building
4) Land parcel texturing
5) Terrain texturing

The timings recorded for both the 2D User Interface and the 3D User Interface can be seen in results section and table 5-2 below:

5.4.2 Results

Simulation Run	**RepastS 2D Completion time (in minutes)**	**R.S.V.T Completion Time (in minutes)**
1	45.23	9.28
2	35.42	9.23
3	27.78	9.28
4	41.32	9.26
5	34.02	9.28
6	36.19	9.27
7	31.84	9.26
8	34.25	9.27
9	40.22	9.28
10	35.91	9.28
Average	**36.22**	**9.27**

Table 5-2: The time (in minutes) each 10 year simulation run took under both RepastS and R.S.V.T

The RepastS simulation runs are on average 26.95 minutes longer than R.S.V.T which had an average simulation completion time. All R.S.V.T simulations completed in less than ten minutes with only one of the RepastS simulations run's finishing under 30 minutes. The longest simulation run from RepastS was 45.23 minutes compared to 9.28 minutes for R.S.V.T.

5.4.3 Discussion of Results

Previous exploration of the literature surrounding the graphical output of agent based models highlights the issue of being processor-bound (see **chapter 2.2.2**). Agent based modelling toolkits like RepastS are developed to accelerate the development of building agent based systems. However the graphical output for agent based toolkits is often found in the form of in-built 2D graphical libraries with little or no hardware acceleration. This forces the processer to run the simulation in addition to rendering all graphical output where even simple lines and shapes require significant processing power, especially when the underlying data is volatile and subject to constant change.

R.S.V.T performed significantly faster which can be directly attributed to its exploitation of the GPU. While it is true that drawing 3D shapes and models is more computationally expensive than a 2D shape, the GPU is designed to process this form of information in an extremely efficient way. R.S.V.T uses the GPU extensively through the use of hardware acceleration; this allows much of the complex processing to take place on the GPU, freeing up the CPU to process the agent based model.

Although the completion time for R.S.V.T was encouraging it had yet to be determined what impact further development and more complexity may have on the completion times. The feasibility test implemented only a subset of the features available in the final release of R.S.V.T. It was important to consider that additional processing would be required when implementing additional design elements (see **chapter 4.4.4**).

This prototype of R.S.V.T has an inefficient algorithm to render the terrain in the form of bruteforce drawing and only contained one visual technique of interest which was changing land parcel colours based on crop yields calculated within the agent based model. It was known that further design elements and visualisation techniques would be added to R.S.V.T. This led to the development of R.S.V.T's **offline-mode** which allows previously run simulations to be stored in a custom format that can be accessed via a network stream or file. R.S.V.T then uses this file or network stream as if it were connected to a real-time agent based model. This further reduced the CPU cycles required by the ABM as it is running a completed simulation, therefore not requiring the additional resources of processing the agent based model. Additionally, the offline-mode lends itself to scenario development due to the ability to recall previously run simulations.

5.5 Conclusions

This chapter introduced the reader to Web Services and SaaS and why they are needed (see **section 5.1**). The technologies that exist for communicating services and data transfer over a network in the form of SOAP, Thrift and Protocol Buffers (see **section 5.2**) were discussed. **Section 5.3** evaluates the performance of the three communication protocols using the landparcel message described in **chapter 4.3.2** which shows XML based messaging formats to be unsuitable for near real-time systems. The chapter ended with a performance evaluation of R.S.V.T (using Protocol Buffers) and Repast Simphony to prove that Protocol Buffers meets two previously stated requirements that 1) it enables loose coupling of the ABM and the 3D User Interface and 2) that the 3D scene renders as fast as that of its 2D counterpart.

The speed difference between the 3D User Interface with Protocol Buffers and Repast Simphony's native 2D interface was significant. The Protocol Buffer enabled User Interface was able to simulate and visualise the model in just over one quarter (25.6%) of the time it takes Repast Simphony's 2D interface to complete the same task. This is because Protocol Buffers negates the need for Repast Simphony's 2D User Interface to be rendered which removes the CPU bottleneck commonly found in ABM toolkits (see **section 2.3.1**).

Protocol Buffers has shown that it can serialize ABM data, transmit over a network, deserialize ABM data for rendering within a 3D environment and provide an increase in performance. The design and implementation of R.S.V.T's offline-mode protects the performance of the visualisation by ensuring that minimal CPU resources are required to run the ABM, reducing the risk of bottlenecks and freeing up processor resources to improve R.S.V.T's 3D environment through the addition of additional visualisation methods.

Chapter 6 – Testing Strategy

The purpose of this chapter is to review and summarize the existing literature on methods used to evaluate the effectiveness of visualisation tools. This chapter concludes with an explanation of the two testing methods implemented as part of this projects testing strategy.

6.1 Visualisation Evaluation Strategies

When testing the effectiveness of a visualisation it is imperative to understand the purpose of the visualisation. As highlighted in **Chapter 2.4.1** there are a high number of visualisation usability studies that offered participants only two choices; they either like the visualisation or they do not (North, 2006). This method of visualisation evaluation fails to consider a number of important factors. A participant may have preferred one visualisation overall but liked minor parts of another or a participant may have no preference whatsoever. In such cases, binary outcome alone is insufficient to provide a complete and thorough overview of the effectiveness of a visualisation tool.Plaisant (2004) carried out a review of the visualisation evaluation strategies used in 50 visualisation projects. Plaisant (2004) then categorised the different methods of visualisation evaluations. The author places visualisation evaluation into four distinct categories which are:

- Comparing design elements within a controlled experiment
- Usability evaluation of the visualisation tool

- Comparing two are more visualisation tools within a controlled experiment
- Case studies of using visualisation tools within a realistic environment

Further explanation of each of the four visualisation evaluation strategies can be found below.

6.1.1 Controlled Experiment Strategy

This form of visual evaluation is used to compare specific design elements within a visualisation. Irani and Ware (2003) use comparative design element testing, showing both 2D and 3D information diagrams [see figure 6.1] to participants and asking them which visualisation they prefer.

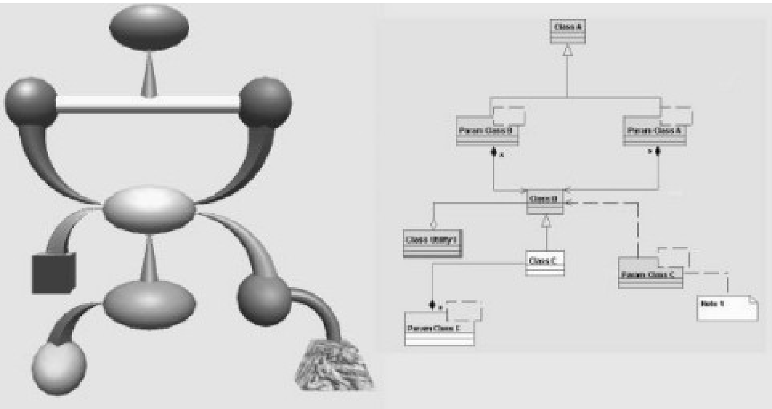


Figure 6.1: An example of comparative design elements. Left: 3D Representation of a Unified Modelling Lanuage (UML) diagram. Right: A traditional 2D UML diagram (from Irani & Ware, 2003).

6.1.2 Usability Evaluation Strategy

Usability evaluation is carried out when developers wish to improve upon a visualisation's effectiveness and functionality when in use. Sutcliffe *et* *al.* (2000) carried out usability evaluation to test the effectiveness of information retrieval by participants when using different visual thesaurus visualisation tools. Hoashi & Hamawaki (2009) compared 2D and 3D visualisation tools of *Music Retrieval Systems.* Typically this type of visualisation evaluation involves task based activities whereby participant's interaction with the tool are recorded such as

- How long did it take to carry out the task?
- Which of the available tools were used to complete the task?
- How accurate or successful was the task carried out?

6.1.3 Comparative Controlled Experiment Strategy

Comparing two or more visualisation tools is a common type of visualisation evaluation. Examples of comparative visualisation tools can be found in Sebrechts (1999) who evaluated three different visualisation tools that included text, 2D and 3D interfaces of search results. These search results consisted of returned documents from a keyword search used by National Institute of Science & Technology's (NIST) PRISE search engine. Each of the results were then visualised using a text-based system, a 2D user interface and a bespoke 3D user interface system. Plaisant *et al.* (2002) carried out a similar experiment with three different visualisation tools of conventional node link tree diagrams. This technique of visualisation evaluation allows data to be gathered relating to which tool(s) the participants prefer under different testing conditions.

6.1.4 Case Study Strategy

This approach allows the researcher/developer to evaluate how a visualisation tool works in a real-life environment by watching and recording user interactions whilst they are interacting with the visualisation tool. Such a study was carried out by Trafton *et al.* (2000) who studied the different visual tools used by professional weather forecasters in their professional environment. This approach is time-comsuming for researcher/developer and the data gathered is subjective to the view of both the participant and the researcher. Visualisation tools being evaluated in this manner should have already completed usability evaluations to ensure only the performance of the tool was being scrutinised by the participant in the real-life situations in which such a tool might be used.

6.1.5 R.S.V.T's Visualisation Evaluation Strategies

The current state of the literature on testing visualisation effectiveness introduced four categories of visualisation evaluation: ***controlled experiments comparing design elements***, ***usability evaluation of a tool***, ***controlled experiments comparing two or more tools*** and ***case studies***. The two controlled experiments (***comparing design elements*** and ***comparing two or more tools***) encourage the participant to identify a preference for a particular design element or identify their preference for a particular tool. Usability studies focus on the participant's interaction with a tool, usually checking the accuracy of responses to set questions and/or the speed it takes to carry out a particular task. Finally, case studies can be used to obtain a thorough evaluation of the performance of a visualisation tool when applied to its problem domain.

There is a requirement to test R.S.V.T for overall preference and design element suitability. This will only be possible by adopting multiple visualisation evaluation strategies [see table 6-1].

Visualisation Evaluation	Objectivet
Controlled experiments comparing two or more tools	To determine the overall preference when comparing R.S.V.T and RepastS
Controlled experiments comparing design elements	To determine suitability of design elements implemented within R.S.V.T
Usability Evaluation	To determine the usability of response of participants and identify general usability of the tool

Table 6-1: R.S.V.T's approach to visualisation evaluation and their corresponding objectives

A multiple-evaluation approach was also used by Sebrechts *et al.* (1999) who conducted usability evaluation in addition to controlled experiments when comparing two or more tools to determine both participant preference and to record their response accuracy whilst carrying out tasks. Hoashi & Hamawaki (2009), when comparing the efficiency of different visual tools for Music Retrieval Systems also recorded user preference of the different tools to determine if user preference had an impact on the accuracy of responses.

With R.S.V.T's visualisation evaluations decided upon, an overview of the methods these evaluation strategies uses is appropriate. Two methods were identified as being prominent throughout each of the visualisation evaluations chosen for R.S.V.T - these are **Visual Preference Surveys** and **Task Based Testing**.

6.2 Controlled Experiments and the Visual Preference Survey

There is evidence to suggest that people are able to elicit emotive responses to different landscapes through their own personal experiences of the landscape but also when shown images of landscapes they have no previous knowledge of (Ulrich 1985, Parsons 1991). Many planning and design consultations use comparative images (Bailey *et al.,* 2001; Ewing 2001; Grossardt 2004) and 3D renders (Bouwman 2006, Isaacs 2011) to gather information on public and stakeholder preference. The methodology of visual preference surveys was created by Nelessen (1994) which Nelessen describes as a method for *"ranking images of places, spaces and land use".* Visual preference surveys have been used extensively to explore personal preferences relating to the aesthetical value of rural and non-urban landscape management. Howley (2012) performed an extensive landscape preference study whereby participants were asked to rate 16 different farming landscapes. Hands (2002) carried out a visual preference survey to assess participant preference of ecological rehabilitation of decommissioned industrial lands while Misgav (2000) conducted a visual preference survey on selected native and planned forests in Israel.

6.2.1 Components of a Visual Preference Survey

A visual preference survey is made up of a visual scene which is defined as a collection of design elements which, when combined together, create a complex visual scene (Hull & Revell, 1989). In the cases of comparative visual preference surveys participants will be asked to rate two visual scenes side by side. Individual design elements are categorised as the parts of the visual scene that change between image sets. These are often the elements that researchers wish to document differences between participant preferences. For example, Bailey *et al.* (2001) had multiple design elements for the effect different highway management measures would look i.e. what the surrounding land use is and width of traffic lanes. An example question from Bailey is – "*does the participant prefer a grass verge or a metal railing as part of a planned highway restructuring?"* Ranking these elements is done using a likert scale with most scales varying between 1-5 and 1-10.

6.2.2 Participant Numbers for Visual Preference Surveys

Participant numbers vary with few authors prepared to give a definitive answer on minimum or maximum participant numbers – often stating that it depends on what is being tested. Kosara (2003) writes that small user groups are more than acceptable when the experiment contains numerous repeated measures with only minor changes such as changing design elements. However, there seems to be an increase in the number of participants when conducting visual preference of larger spatial areas such as Howley (2012) who collected 430 responses and Misgav (2000) who collected 150. Both Howley and Misgav carried out demographic analysis including gender, age, marital status and region the participants lived in. This sort of demographic study requires larger participant numbers to get a viable spread of data.

Smaller participant numbers tend to be found within projects using comparative visual preferences – that is rating against another image rather than rating each image individually. Participant numbers have ranged from 21 participants (Grossardt) to 39 participants (Hands). In one case participant numbers were limited due to the technological resources used to record input from participants via electronic voting pads with 30 participants taking part in Bailley's Highway improvement Visual Preference Survey. One exception to having smaller participant numbers when using comparative images was Miller *et al.* (2009) - they had 139 participants taking part in a visual preference survey via electronic voting - comparing computer generated images of forested areas in the present and 100 years into the future.

6.2.3 Advantages & Disadvantages of Visual Preference Surveys

When conducting visual assessment of a landscape the objectivity, reliability and validity of the quantitative methods are important considerations (Roth 2006). Traditionally methods for landscape data acquisition is through the use of on-site surveys or coloured slides/prints - often photographs, of the landscape in question. Roth cites the expenses incurred when using photograph-based visual preference surveys. This type of visual preference survey uses physical copies of photographs which are given to the participant group to rank in order of preference. This option is expensive and physical copies run the risk of being damaged beyond any useful function which incurs more expense. Photographs do have the advantage of being able to demonstrate both real-life and possible future scenarios through the use of computer generated images. However, this is not a valid option when conducting an on-site survey. On-site surveys require that

a The landscape in question must already exist. This option would exlude most public participation related visualisation tools as the landscape will have already been built.

b When comparing more than one landscape the participants are asked to go from place to place. There is the issue of the added expense of transporting the participrants from one site to another and the reliance on the participant's memory to recall the previous landscape correctly.

Yet visual preference surveys are still a commonly used method of visual landscape assessment. Ewing (2001) says that a visual preference survey *"helps citizens and community leaders envision design alternatives in ways that words, maps and other communications media cannot"*. When visual preference surveys are used in conjunction with a visioning project, a project that has possible and plausible future outcomes; that are not predetermined; the results gathered aid not only the stakeholders interested in the project but those involved in creating the visualisation tool. The disadvantages mentioned above can be planned for and avoided if a testing strategy is devised with these issues in mind. The advantages of a visual preference survey are clear:

1 The data acquisition of variously sized partipatory groups can be gathered quicker than face-to-face interviews and results in more quantitative data than questionnaires.

2 The visual preference survey works especially well when comparing two (or more) landscapes rather than a single landscape.

3 Provides important feedback on the quality and effectiveness of the visual techniques used.

It remains clear however that to fully test the effectiveness of a visualisation tool, a visual preference survey alone is not sufficient.

6.3 Usability Evaluation and the Implementation of Task Based Testing

Effectiveness and usefulness of a visualisation cannot be judge by a participant' stated visual preference alone. Task based testing, or Information-seeking testing as it is sometimes called, is a method of testing used within usability evaluation. The accuracy of the information assimilated by the participant from the visualisation is an important factor to consider, and test. Cockburn and McKenzie have evaluated various comparative visualisation tools, most of which compare 2D and 3D interfaces. Such examples can be found during their evaluation of cone tree visualisation tools (Cockburn & McKenzie, 2000) also, their evaluation of documents management systems (Cockburn & McKenzie, 2001). Participants were timed during tasks and the accuracy of their responses recorded. Their findings for cone tree visualisation tools indicated that participants preferred the 3D visualisation although the time to complete tasks increased and the accuracy of responses dropped when the dataset increased in size. Results from their document management systems evaluation showed a decrease in the time it took participants to complete tasks when using the 3D interface rather than the 2D interface. One final example of Cockburn and McKenzie's work (2002) showed a decrease in participant performance when using a 3D interface compared to its 2D counterpart.

There are a number of issues that need to be considered by the moderator when asking participants to retrieve data from any visualisation tool. These are:

- What constraints are placed on the participants with regards to interaction with the tool? Will they be using the tool themselves or will the tool be driven by the experiment moderator?
- How open and flexible is the task based session? Are participants permitted to ask questions of the moderators? Are participants permitted to confer amongst themselves to try and extract information?

6.3.1 Components of Task Based Testing

The components that make up task based testing vary with each visualisation evaluation. The purpose of the visualisation tool will define what tasks can be carried out and determinations have to be made when choosing which tasks to test. Common methods do arise throughout most task based testing literature with the most popular being timing participants while they carry out the chosen task (Byrd 1999, Plaisant et al. 2002). It should be noted that the length of time allocated to participants to carry out these tasks range from 15 – 60 minutes, there is broad agreement that anything shorter can result in a lack of data through participant's lack of understanding and anything longer has a negative effect on participant's abilities to carry out their tasks. Other task based testing examples found within the literature seem to be less consistent. Examples of participants having enforced deadlines to perform tasks ranging from 30 seconds during Plaisant's experiment, 2 minutes during Sebrechts' experiment and 5 minutes during Byrd's experiment. Finally, the amount of information given to the participants varies. Again, this variability is a direct result of what the visualisation tool does and what is actually being tested. Some visualisation evaluation experiments allowed the participants up to an hour of practice time while others (Tory, 2006) gave each participant 5 practice runs of a task and then recorded 20 task results after. This was repeated for each unique task. Others gave no practice time or introduction to the visualisation tool at all, instead relying on the user to figure out the workings of the tool on their own. In the cases of Plaisant (2002), Byrd (1999), Saraiya (2005) and Gilford (2013) this was a deliberate condition to ensure the tool was more accessible to non-expert computer users and non-expert stakeholders.

6.3.2 Participant Numbers for Task Based Testing

The number of participants used within task based testing tends to be limited. Byrd (1999) uses only 6 participants, Sebrechts (1999) uses 15 participants to evaluate the comparative information retrieval of textual, 2D and 3D interfaces. Similarly, Plaisant (2002) compares three computer tree systems with 18 participants. The low number of participants required for task based testing can be attributed to the fact that it does not take many participants, or participants with any sort of computer experience, to find software bugs and design flaws. If the participant has an expectation that is not met by the tool then the total amount of people that feel the same way is almost irrelevant, it still remains an unmet expectation. Task based testing is not just about technical usability i.e the number of bugs found or fixed. Information retrieval or the participant's ability to extract information from the visualisation tool is imperative. Only a small number of participants are needed to determine if data retrieval is working as expected or not. If a significant portion of the participant group is carrying out information retrieval incorrectly then the tool has not delivered an accurate method of information retrieval. Further participants may push the percentage higher but can be a convenient detraction from the failings of the visualisation tool.

6.3.3 Advantages & Disadvantages of Task Based Testing

The greatest advantage of task based testing is the watchable and recordable nature of the participant's interactions (Scholtz, 2004). This presents a situation whereby the researcher can gather critical usability information based on what the participant does and not relying on what the participant says. Additionally, in-depth analysis can be performed on many aspects of a visualisation tool when conducting task based testing. For example, assume a method of task based testing has been set up to ensure that users can easily navigate their way around the User Interface of a generic visualisation tool. If the researcher decides to record the time taken to complete a task, the number of mouse clicks or key-presses required to complete the task and the accuracy of the completed task a number of additional results may be found after data analysis that was not part of the original usability evaluation strategy. It could lead to the discovery of excessive mouse clicks on a different area of the User Interface that was not originally tested. The possibility of excess data collection can also be a disadvantage. Often the data gathered from usability studies is complex and can be time consuming to analyse. Keeping group sizes small and testing sessions short helps reduce the impact of the negative consequences of task based testing.

6.4 R.S.V.T's Testing Strategy

From the literature presented it was identified that R.S.V.T needs to adopt multiple visualisation evaluation strategies. This was presented in table 6-1. This figure has been further updated to include the methods discussed in **section 6.2** and **section 6.3** which were the Visual Preference Survey and Task Based Testing respectively and can be seen in table 6-2 below.

Visualisation Evaluation	Objective	Method
Controlled experiments comparing two or more tools	To determine the overall participant preference when comparing R.S.V.T and RepastS	Visual Preference Survey
Controlled experiments comparing design elements	To determine suitability of different design elements implemented within R.S.V.T	Visual Preference Survey/Task Based Testing
Usability Evaluation	To determine the information retreival of participants and identify general usability of the tool	Task Based Testing

Table 6-2: An updated table of 6-1 to show the methods used in R.S.V.T's testing strategy

A visual preference survey was chosen to gather participant preferences of both RepastS and R.S.V.T. The visual preference survey also gathered data on participant preference of the different design elements used within R.S.V.T. Task based testing was also applied to the different design elements to determine if any of the visualisation methods have a significant impact on the accuracy of response and determine and to determine generally usability of the tool.

6.5 Testing Objectives

The objectives listed in table 6-2 are re-defined as testing questions. This was crucial towards determining how both the visual preference survey and task based testing was implemented – specifically what images to include in the visual preference survey and what questions would be asked of the participants during the task based testing. These questions are:

1 Does the user find more aesthetically pleasing when shown the visual scenes of R.S.V.T or RepastS?

2 Which of R.S.V.T's visual methods and design elements were preferred by participants?

3 Can the participant extract information about model and land parcel parameters from R.S.V.T?

1 The research questions above, along with the feasibility test carried out on Protocol Buffer's technical feasibility of data transfer and storage formed the basis of the testing hypotheses below.

2 The visualisation is sufficiently effective at conveying Land Use Cover Change.

3 The chosen communication protocol will transfer data at an acceptable rate.

4 The methods of data capture and storage are flexible and scalable to support further development.

R.S.V.T’s objectives, along with the testing questions and hypotheses that were previously introduced can be viewed in table 6-3.

Objective	Testing Question	Testing Hypotheses
To determine the overall participant preference when comparing R.S.V.T and RepastS.	Does the user find more aesthetically pleasing when shown the visual scenes of R.S.V.T or RepastS?	Participants will find scenes being displayed through R.S.V.T more visually appealing than those of its RepastS counterpart.
To determine suitability of different design elements implemented within R.S.V.T.	Which of R.S.V.T’s visual methods and design elements were preferred by participants?	Participants will identify combinations of R.S.V.T’s visualisation methods visually appealing with a sufficiently adequate accuracy at data retrieval.
To determine the accuracy of response of participants and identify general usability of the tool.	Can the participant accurately extract information about model and land parcel parameters from R.S.V.T?	Participants will be able to identify minimum and maximum values of land parcel parameters.
		Participants will be able to identify a range of values related to model parameters.
		Participants will identify combinations of R.S.V.T’s visualisation methods visually appealing with a sufficiently adequate accuracy at data retrieval.

Table 6-3: R.S.V.T's testing objectives, questions and hypotheses

6.5.1 R.S.V.T's Testing Methodology

Email invitations were extended to over 80 non-expert stakeholders consisting of soil scientists, 3D artists, software engineers as well as students and the general public. A total of 31 participants agreed to be part of R.S.V.T's testing. It would have been logistically difficult to gather all participant feedback as one large group of 31 therefore the participants were split into groups with each group containing between 4 and 6 participants. Each group took part in two sessions that were based on the two visualisation evaluation methods that were chosen: the Visual Preference Survey and Task Based Testing.

Over a two week period a total of 31 participant responses were collected from 5 separate participant groups. Participants were not required to have any knowledge of either land use modelling or computer visualisation. Any participant who was viewed to have knowledge more akin to an expert stakeholder was not permitted to participate in the experiments. No participant interaction with the computer equipment or the visualisation was required by the participants therefore a participants computing experience was not relevant. Each participant group was given an overview of the project which clearly explained how each session was structured. A roundtable and large monitor was situated in the middle of room large enough for 8 people. The sessions were not timed and although the participants were told the sessions should take no longer than 30 minutes they were also made aware that no time constraints were put on them. Participants were also told, and reminded throughout the sessions, that discussion between themselves and the moderator was allowed and encouraged.

The remainder of this chapter will discuss each testing session in greater detail. Example visual preference surveys and corresponding images can be found at appendix C. The task based testing question sheet can be found at appendix D with the videos used for the task based testing located on the DVD in appendix D-1.

6.6 Testing Session 1: Visual Preference Survey

During this session participants were shown 19 slides containing side by side images of RepastS and R.S.V.T. Participants were asked to rate which image they preferred for each of the 19 slides. Participants were given a copy of the image rating scale seen below in table 6-4.

Ranking	Description
1	Much prefer RepastS image
2	Slightly prefer RepastS image
3	Neither prefer RepastS or R.S.V.T image
4	Slightly prefer R.S.V.T image
5	Much prefer R.S.V.T image

Table 6-4: The likert scale used in R.S.V.T's Visual Preference Survey

6.6.1 Image Selection

The process of selecting which images to use was determined by the design elements available to R.S.V.T. In most cases a design element that can be found within RepastS has a corresponding (although visually different) design element within R.S.V.T but this is not the case for every feature. Therefore any image used within the visual preference tool must only contain design element common to both R.S.V.T and RepastS. This formed the basis for selecting which images to use. Images were created for both RepastS and R.S.V.T visualising a variety of design elements as seen in table 6-5 below.

More detailed information on each of the design elements listed below can be found at **section 6.6.3** and **section 6.6.4**.

Repast Design Elements	**R.S.V.T Design Elements**
Land Use Change – Colour Map	Land Use Change – Colour Map Land Use Change – Texture Map Land Use Change – 3D Models (Hardware Instancing)
Parameters for Individual Land Parcels – Textual Information	Parameters for Individual Land Parcels – Textual Information Parameters for Individual Land Parcels – 3D Pillars Parameters for Individual Land Parcels – Exploded View
Skylark Population – Coloured Dots & Shapes	Skylark Population – 3D Models (Hardware Instancing)
Topographical representation – 2D Shapefile outline	Topographical Representation – 3D Terrain, textured (with grass), shapefile outline Topographical Representation – 3D Terrain, textured (aerial photography), shapefile outline

Table 6-5: A breakdown of the design elements common to both RepastS and R.S.V.T

6.6.2 Data Parity between RepastS and R.S.V.T

Images were only permitted if they contained design elements common to both visualisation tools. The communication protocol (Protocol Buffers) that links the agent based model to R.S.V.T was adapted to save model runtime data to file rather than send it over a network. This file was then loaded into R.S.V.T and RepastS to ensure that the visual output of each image shows the same underlying data at any given point during the model run.

6.6.3 RepastS Design Elements

This section provides an overview of the design elements used within RepastS. All RepastS design elements noted in table 6-5 are included here, along with relevant images to provide a visual demonstration of the design element that was tested.

6.6.3.1 Land Use Change – Colour Map

The default design element used within Repast S to visualise land use change is through use of a colour map as seen in figure 6.2. The surrounding polygon indicates the shape of the land parcel whilst the colour indicates the land use type. Lighter shades of the land use colour are used to show crop height during runtime.

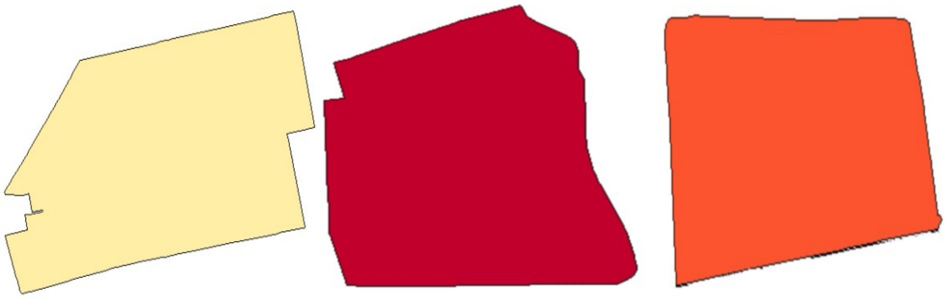
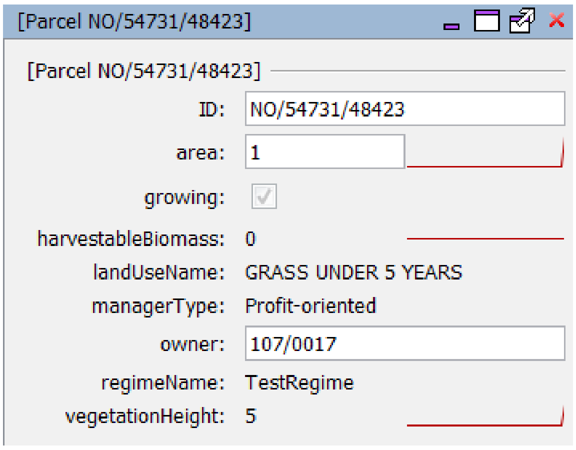


Figure 6.2: An example of different colour mapping examples for land parcels within RepastS

6.6.3.2 Individual Land Parcel Sustainability – Textual Information

Repast S displays information pertinent to the model parameters of individual land parcels through a textual information box as seen in figure 6.3. Details such as land parcel ID, land use type and crop height can be found via this design element.



[Parcel NO/54731/48423]

[Parcel NO/54731/48423]

ID: NO/54731/48423

area: 1

growing: ☑

harvestableBiomass: 0

landUseName: GRASS UNDER 5 YEARS

managerType: Profit-oriented

owner: 107/0017

regimeName: TestRegime

vegetationHeight: 5

Figure 6.3: The textual information as shown in RepastS. This displays key data about the current state of a selected land parcel

6.6.3.3 Skylark Population – Coloured Dots & Shapes

Repast S portrays skylarks as coloured dots or shapes, geo-located within their nesting land parcel as seen in figure 6.4. Each shape corresponds to a different skylark type. These are

- Egg. An unhatched skylark egg
- Female. Indicates a female skylark
- Fledgling. Indicates a fledgling skylark
- Male. Indicates a male skylark
- Nestling. Indicates a recently hatched skylark

These are the most interactive design elements from RepastS as they are spatially aware and change shape and colour over time as they mature.

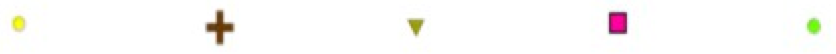


Figure 6.4: The design elements for visualising skylark states. From left to right: Egg, Female, Fledgling, Male, Nestling

6.6.3.4 2D Topographical Representation – Shapefile Outline

Repast S visualises the land parcels contained within the Lunan region through black outlining of the Lunan shapefile as seen in figure 6.5. Each individual land parcel can be identified by its black outline. This image is of the initial conditions of a simulation with beige colouring indicating an unknown crop type and a crop height of 0.

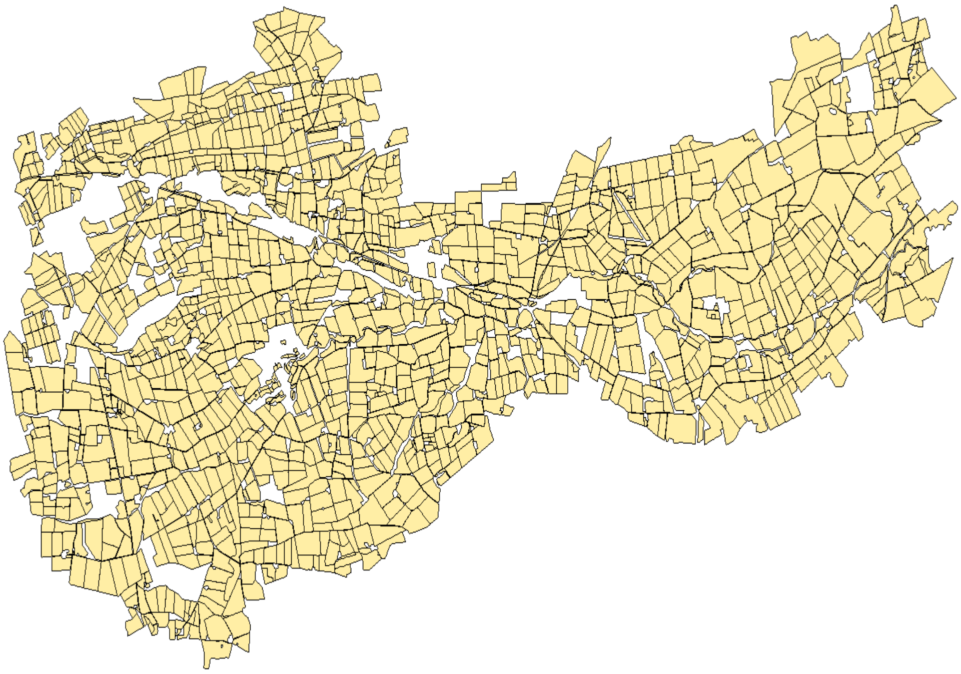


Figure 6.5: The shapefile outline representing the land parcels in Lunan. The black outline indicates the boundaries of each land parcel

6.6.4 R.S.V.T Design Elements

This section provides an overview of the design elements used within R.S.V.T. All R.S.V.T design elements noted in table 6-5 are included here, along with relevant images to provide a visual demonstration of the design element that was tested.

6.6.4.1 Land Use Change – Colour Map

R.S.V.T uses a colour map to represent land use with a colour gradient to indicate the crop height for each land parcel as seen in figure 6.6.

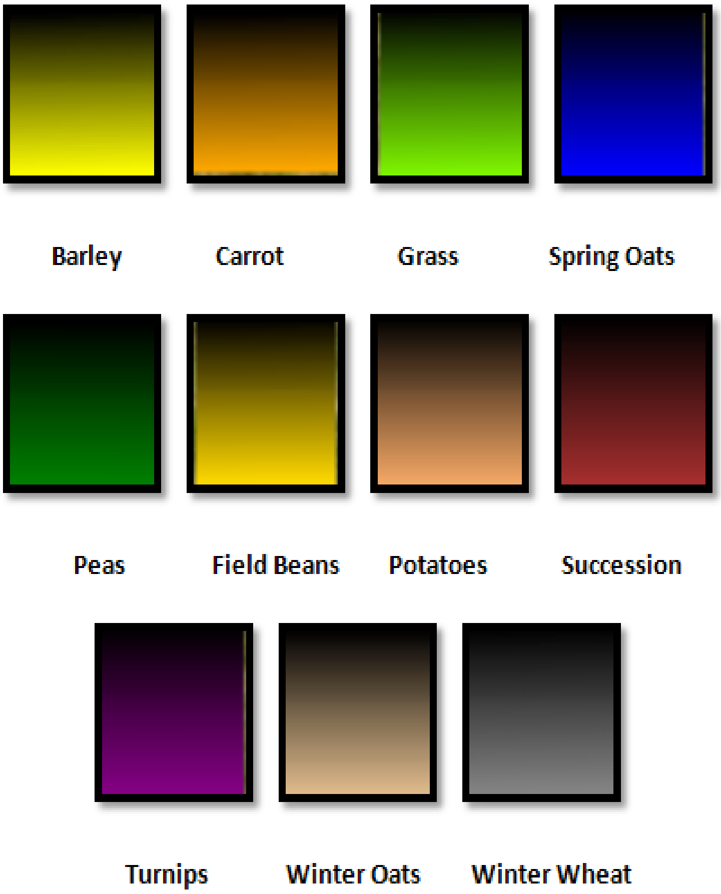


Figure 6.6: Each of the land use type gradient colour maps used in R.S.V.T

6.6.4.2 Land Use Change – Texture Map

R.S.V.T also uses a texture map to represent land use with a transparency slider to indicate the economic yield for each land parcel as seen in figure 6.7.

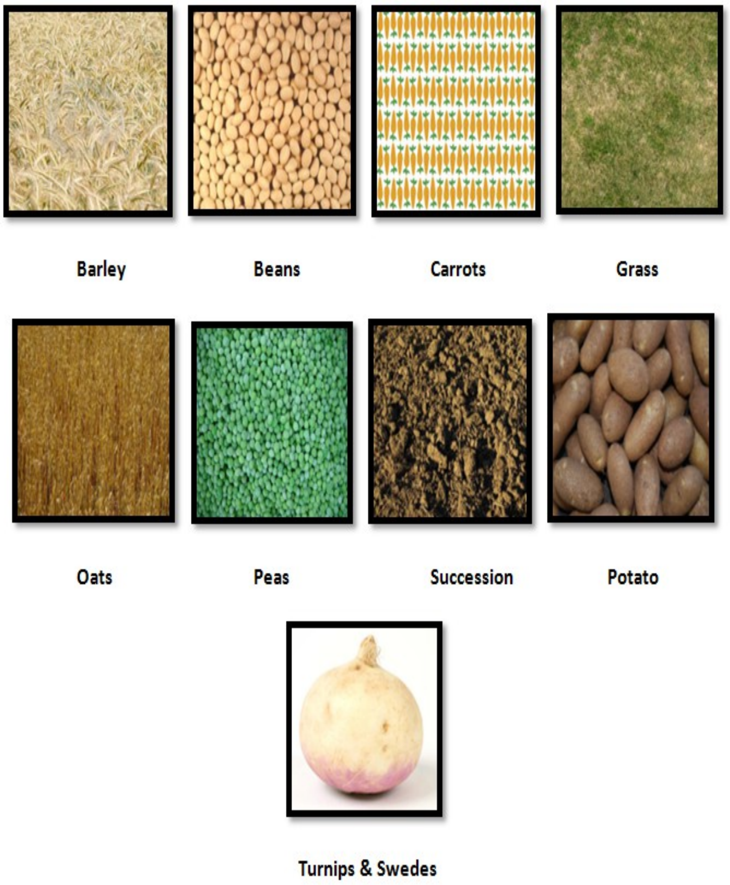


Figure 6.7: The texture maps for each of the land use types found in R.S.V.T

6.6.4.3 Land Use Change – 3D Models (Hardware Instanced)

The final method used within R.S.V.T to visualise land use cover and change is through detailed 3D models that grow and change shape according to its crop height as seen in figure 6.8.

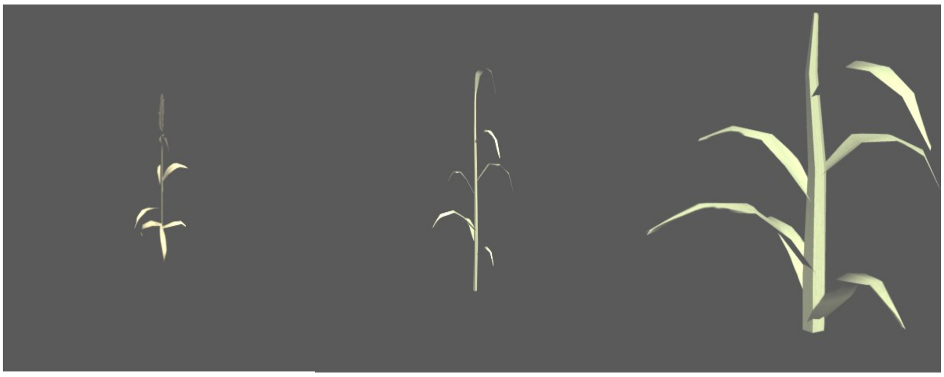
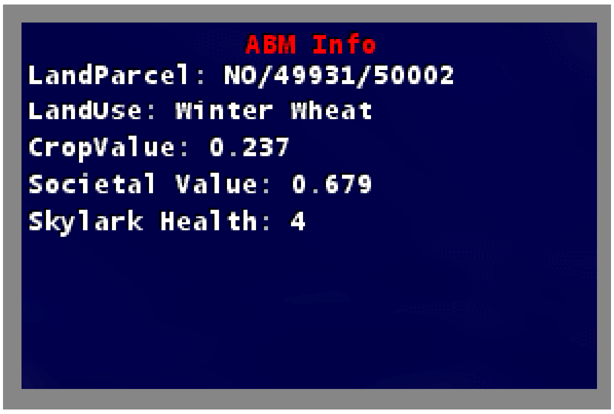


Figure 6.8: 3D barley model found in R.S.V.T at different stages of growth

6.6.4.4 Individual Land Parcel Model Parameters – Textual Interface

R.S.V.T displays textual information relating to the model parameters of individual land parcels as seen in figure 6.9.



```
ABM Info
LandParcel: NO/49931/50002
LandUse: Winter Wheat
CropValue: 0.237
Societal Value: 0.679
Skylark Health: 4
```

Figure 6.9: Textual information showing key land parcel data

6.6.4.5 Individual Land Parcel Model Parameters – 3D Pillars

R.S.V.T uses dynamically changing 3D pillars to show the model parameters of individual land parcels with each colour indicating what model parameter it represents (economic, environmental, social) as seen in figure 6.10.

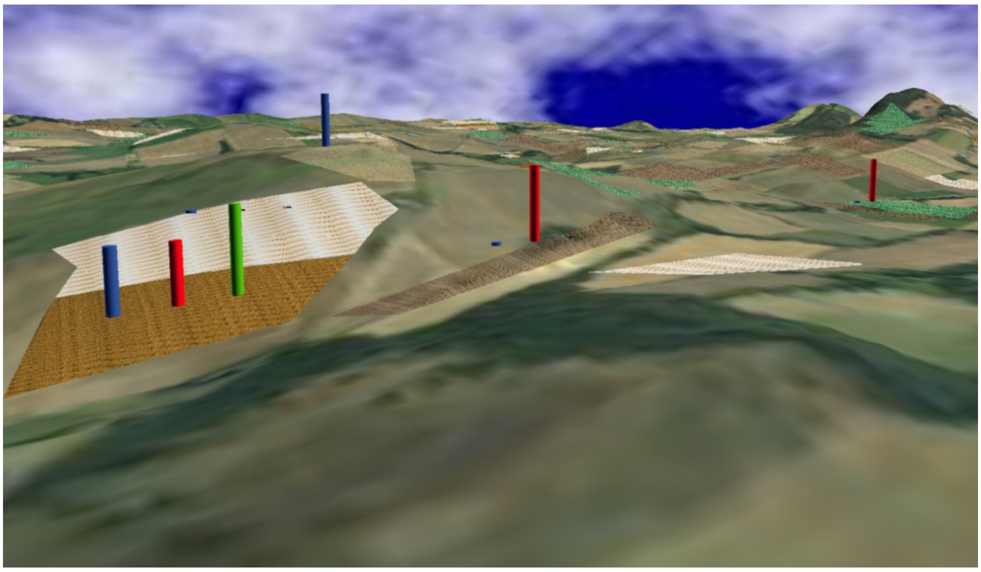


Figure 6.10: An example of land parcel parameters (economic, environmental and social) being displayed using the 3D pillars technique

6.6.4.6 Individual Land Parcel Model Parameters – Exploded View

R.S.V.T's final method for visualising individual land parcel model parameters is through its exploded view, creating 3 identical land parcels that explode from the centre of the original, with colour indicating what model parameter (economic, environmental or societal) it represents. The various levels of opaqueness indicate how high or low the model parameters are and can be seen in figure 6.11.

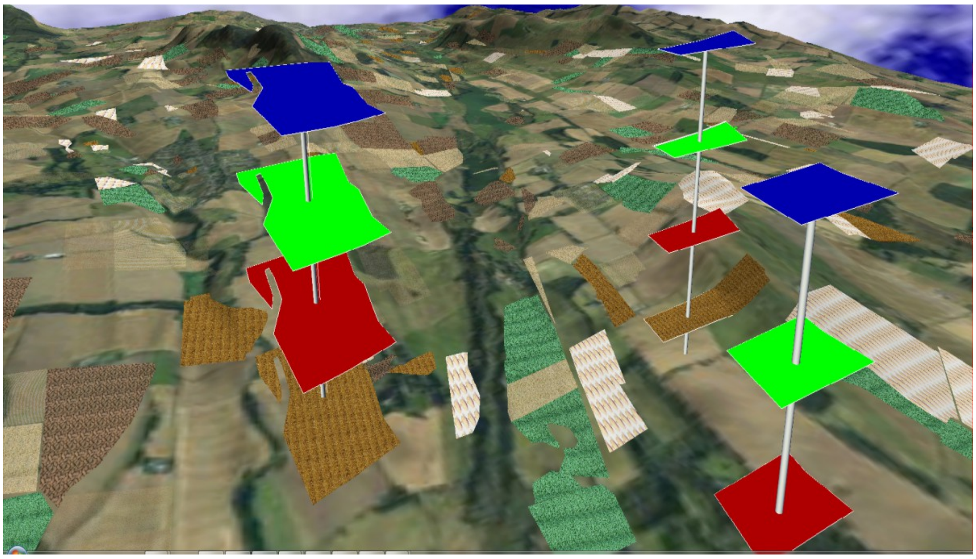


Figure 6.11: An example of land parcel parameters (economic, environmental and social) being displayed using the exploded view

6.6.4.6 Skylark Population – 3D Models (Hardware Instanced)

R.S.V.T uses a detailed 3D model to represent the regional skylark population as well as placing them geo-accurately within the 3D scene. The model can be seen in figure 6.12.

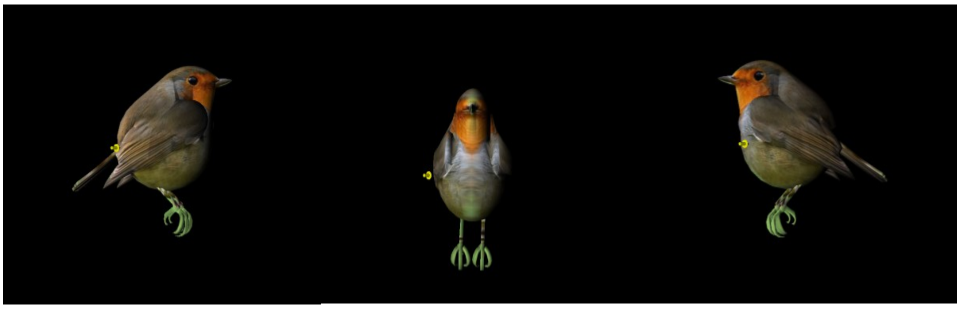


Figure 6.12: Left, front & right views of the bird model used to represent the distribution of the skylark population within R.S.V.T

6.6.4.8 3D Topographical Representation – Terrain Textured (Grass), Shapefile Outline

R.S.V.T creates an accurate 3D terrain based on heightmap data. This terrain is then textured with a repeating grass texture. Finally, the shapefile outline showing the boundaries of each land parcel are rendered over the terrain based on the land parcels geolocation. An example can be seen in figure 6.13 below.

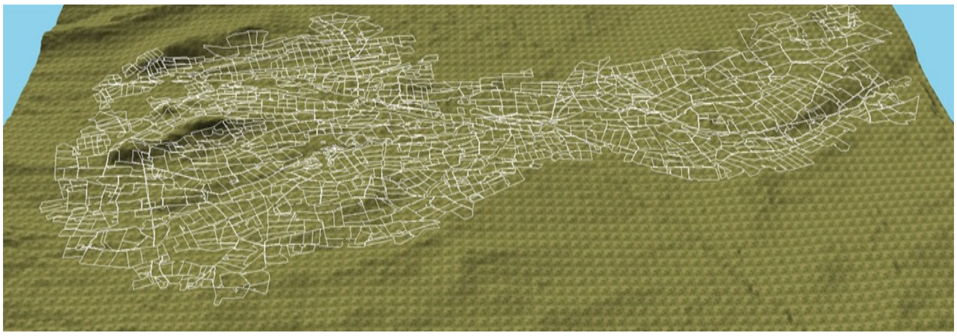


Figure 6.13: Grass textured terrain with a shapefile overlay from R.S.V.T

6.6.4.9 3D Topographical Representation – Terrain Textured (Aerial Photography), Shapefile Outline

The terrain mesh can be textured with aerial photography of the region acquired from Google Earth. The shapefile outline showing the boundaries of each land parcel are rendered over the terrain based on the land parcels geo-location and can be seen in figure 6.14 below.

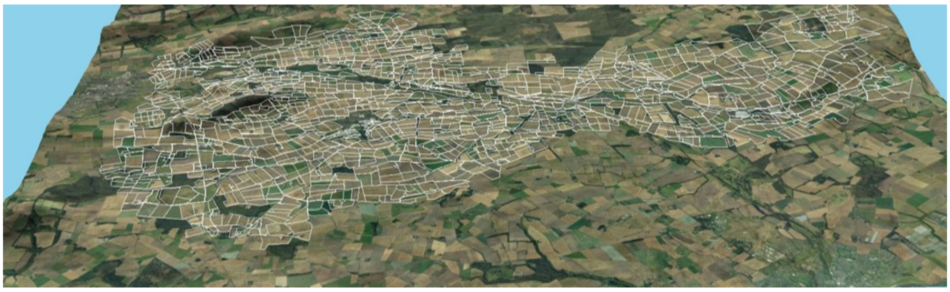


Figure 6.14 Aerial textured terrain with a shapefile overlay from R.S.V.T

6.6.5 Summary of Visual Preference Survey

Due to R.S.V.T having multiple visualisation methods for the same design elements, different combinations of design elements were tested. A table showing each image's design elements can be found below in table 6-6.

	R.S.V.T Component						
Image Number	Grass	Aerial Photography	Colour Map	Texture Map	3D Models	Exploded View	3D Pillars
1	X						
2		X					
3	X			X			
4		X	X				
5	X		X				
6		X		X			
7	X			X		X	
8	X		X			X	
9		X		X		X	
10		X	X			X	
11	X			X			X
12	X		X				X
13		X		X			X
14		X	X				X
15	X				X		
16	X			X	X		
17	X		X		X		
18		X		X	X		
19		X	X		X		

Table 6-6: The design elements that make up each image of R.S.V.T's Visual Preference Survey

The table above shows that images were made up of one or more of the following:

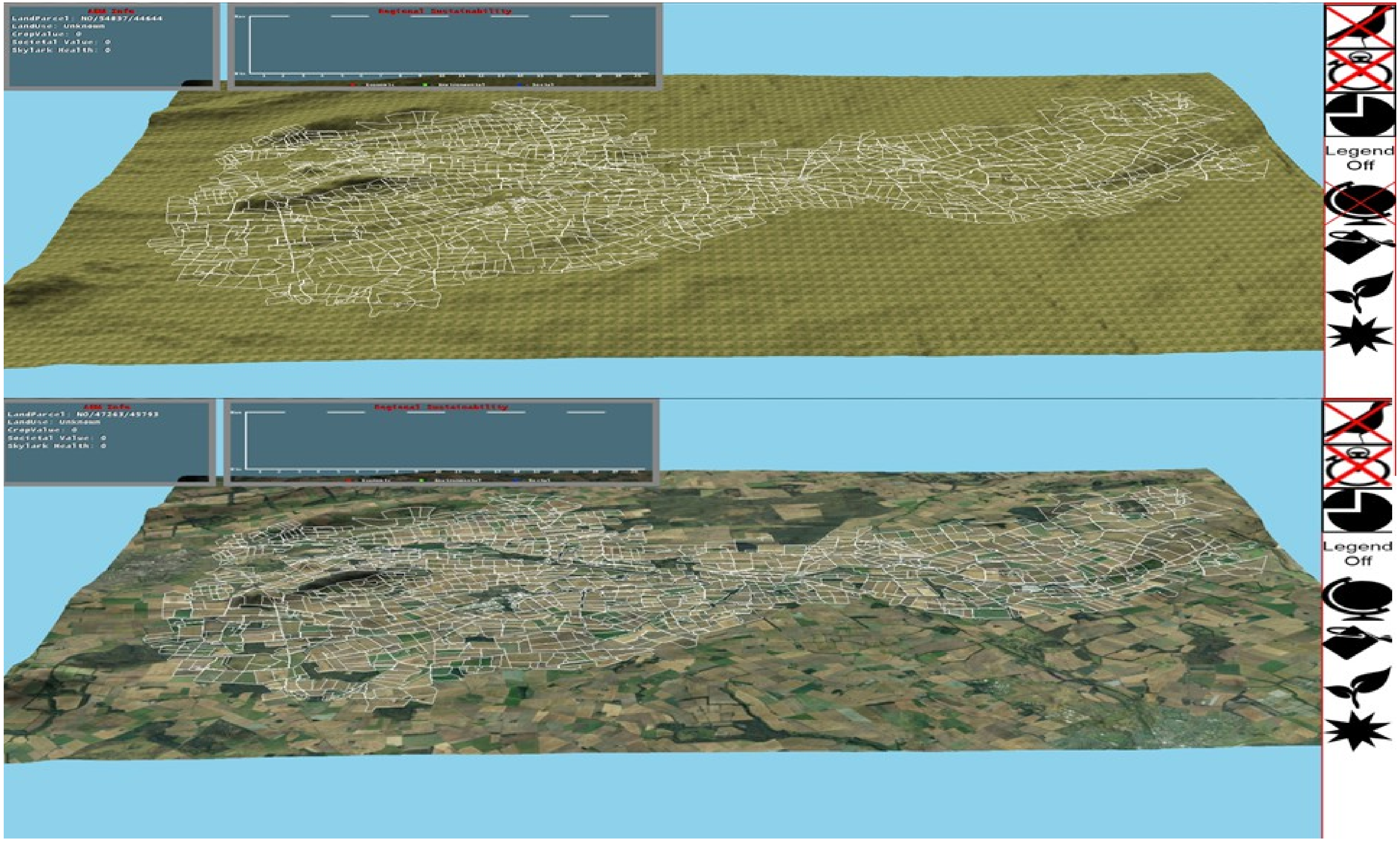
- A grass overlay or aerial photography [e.g figure 4.15] for the Lunan catchment

- A texture map or a colour map [e.g figure 4.16] for land cover

- Exploded view or pillar view [e.g figure 4.17]

- 3D models representing land use and skylarks [e.g figure 4.18]

- A grass overlay or aerial photography [e.g figure 4.15] for the Lunan catchment
- A texture map or a colour map [see figure 4.16]
- Exploded view or 3D pillars [see figure 4.17]
- 3D models representing land use and skylarks [see figure 4.18]



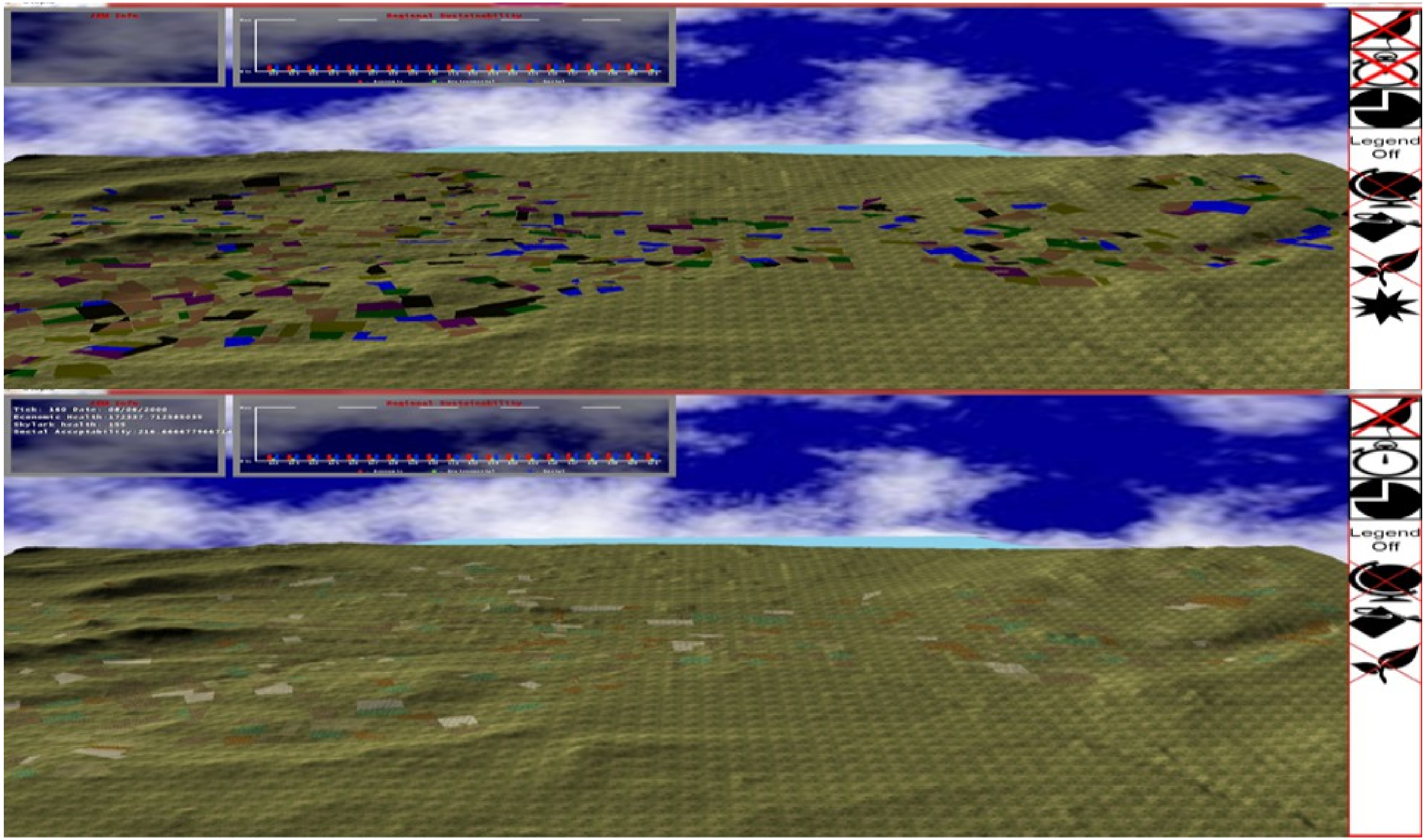
ABM Info
LandParcel: NO/54837/44644
LandUse: Unknown
CropValue: 0
Societal Value: 0
Skylark Health: 0

Regional Sustainability

ABM Info
LandParcel: NO/47263/45793
LandUse: Unknown
CropValue: 0
Societal Value: 0
Skylark Health: 0

Regional Sustainability

Figure 6.15: Grass overlay (Top) and Aerial overlay (Bottom) example



Tick: 160 Date: 08/08/2000
Economic Health:172337.712585039
Skylark health :155
Social Acceptability:216.666677966724

Figure 6.16: Colour map (top) and texture map (bottom) example

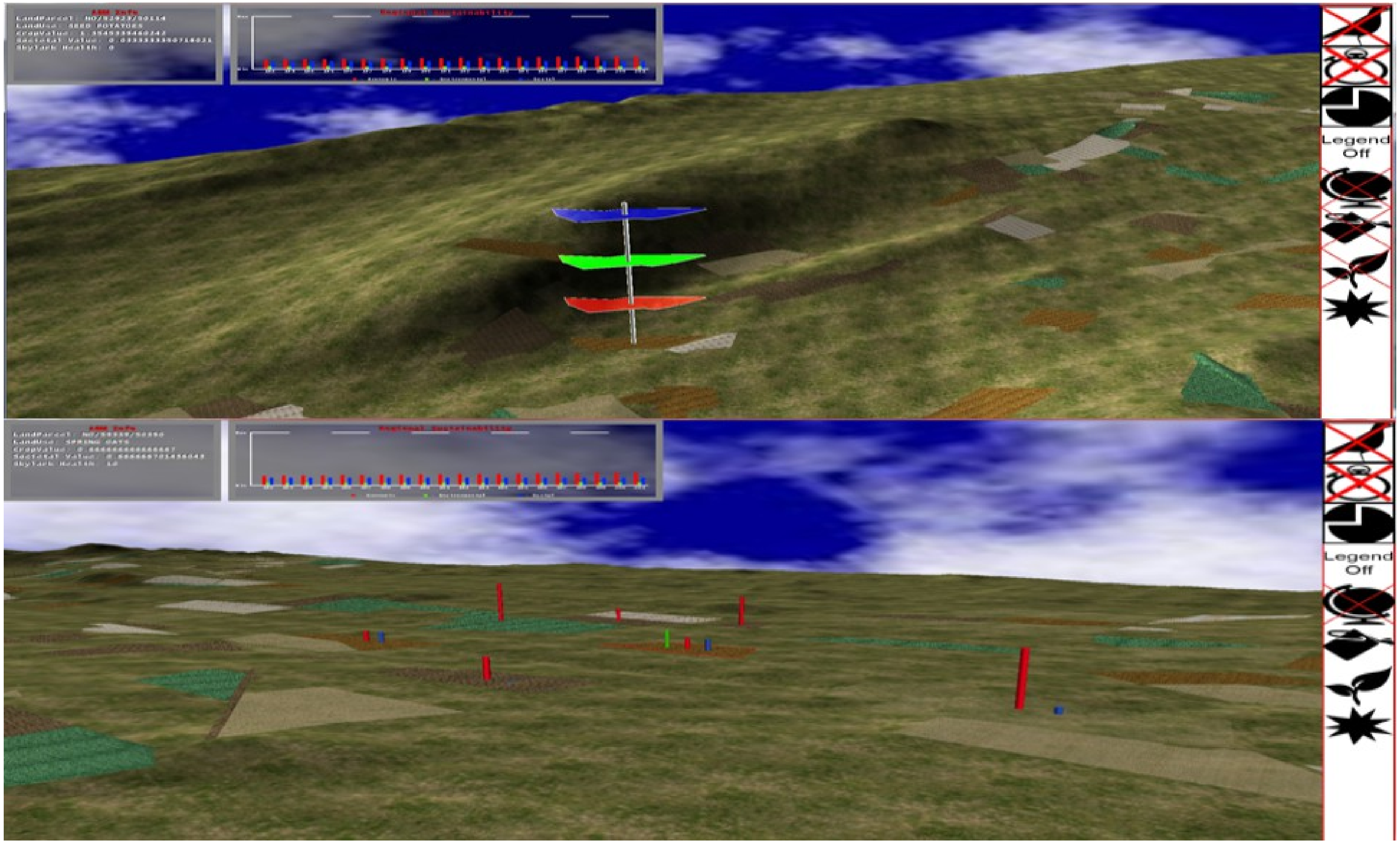
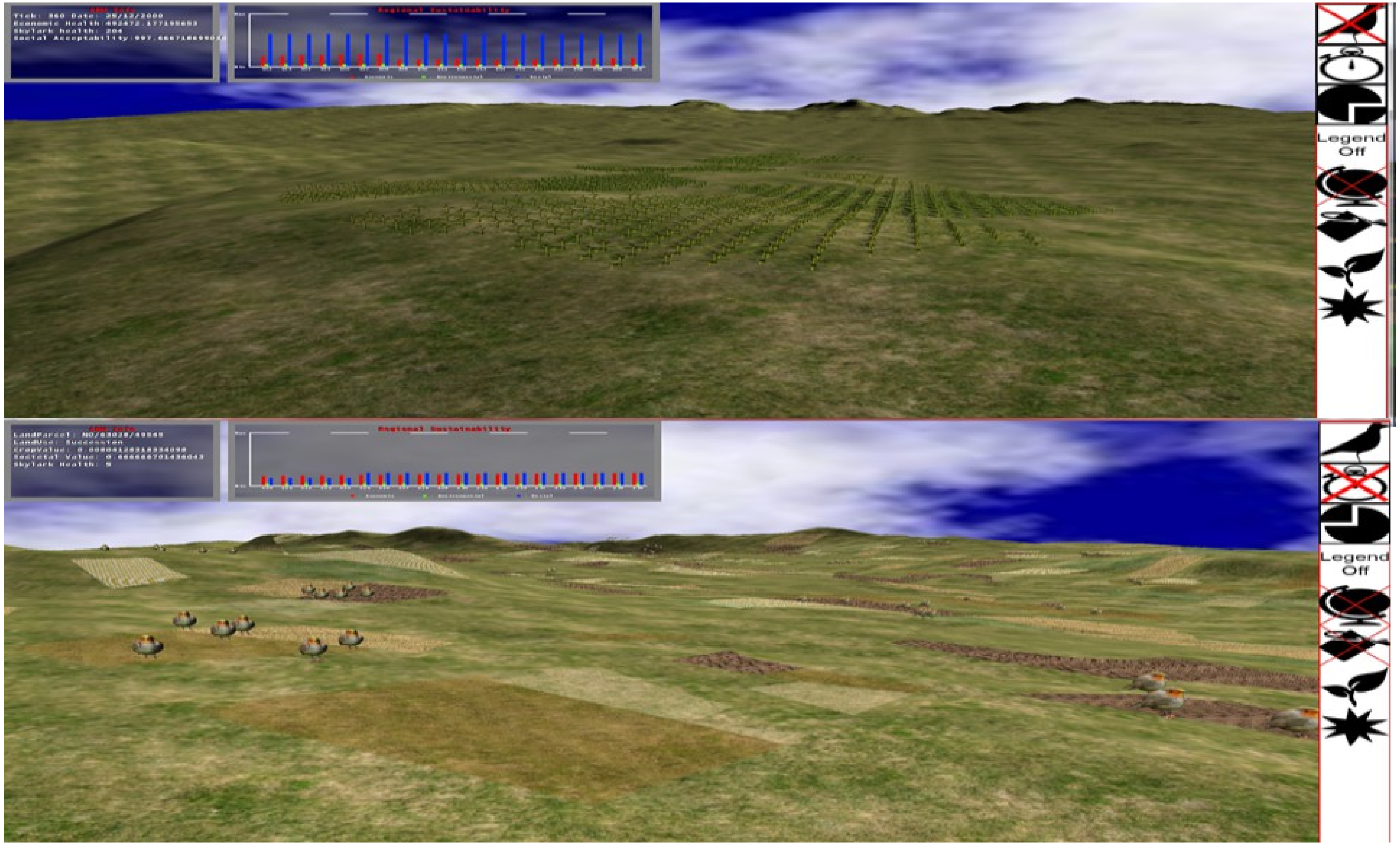


Figure 6.17: Exploded (top) and 3D pillars (bottom) model parameters example



Tick: 360 Date: 25/12/2000
Economic Health:492472.177195653
Skylark health: 204
Social Acceptability:997.666718696[illegible]

LandParcel: NO/83028/49949
LandUse: Succession
CropValue: 0.0000412831[illegible]
Societal Value: 0.666666701438043
Skylark Health: 5

Figure 6.18: 3D models found within R.S.V.T with 3D crop models (top) and 3D skylark models (bottom)

6.7 Testing Session 2: Task Based Testing

This session required participants to view short videos, approximately 30 seconds in length, showing a variety of common tasks being carried out within R.S.V.T. Allowing participants full control of R.S.V.T was considered but deemed infeasible for two reasons. Firstly, the purpose of R.S.V.T's task based testing was to identify which visualisation methods work best together. As the 3D User Interface can be used to drive the ABM spatially and temporally it would have been possible for the participant to judge a visualisation method during a time or place that was unsuitable. For example, a participant compares the exploded view and pillared view and determines that the pillared view was much more effective as the exploded view does not show anything they consider useful. However the participant is unaware that they were attempting to view the visualisation method of a recently harvested land parcel – which would be expected to have little or no data attached to it. The use of pre-recorded videos ensured each participant was being asked to evaluate the same visualisation methods under the same conditions._In all cases the session was carried out immediately after the visual preference survey.

The participants were asked to complete four tasks, each task having one or more questions to be answered. They were reminded that discussion between themselves and the moderator was permitted and told that there was no time limit to answering.

6.7.1 Setting the Tasks & Questions

Tasks within R.S.V.T were chosen by identifying the most common tasks that RepastS users would carry out. Discussions as to which tasks should be included were held with Dave Murray-Rust and Eleonore Guillem, both of whom have worked extensively on building the agent based model and its 2D interface. The most common tasks identified along with information the participants would find useful are below:

1. **Selecting a land parcel:** A user will want to select one land parcel and expect to find relevant information relating to the land parcels parameters.
2. **Identify land use change:** A user should be able to identify when a land parcel's land use changes.
3. **Locate general model parameters:** A user should be able find general model parameters such as regional sustainability, model date and current model state.
4. **Interpret land parcel parameters:** A user should be able to identify values of land parcel parameters.

Task	Question
Selecting a land parcel	Please indicate which Land Parcel was selected as well as any additional information about the land parcel such as crop type, crop value and skylark health within the land parcel?
Locate general model parameters	
Identify land use change	Please indicate which land use is most abundant within the video?
Interpret land parcel parameters	Please indicate which land parcel parameter is the highest and which land parcel parameter is the lowest?

Table 6-7: The questions asked of the participants

6.7.2 Creating Task Based Videos

Four videos were created for each of the four tasks. This was done to determine if particular combinations of visualisation methods had an effect on the responses. Table 6-8 shows the breakdown of the visualisation methods used in each video.

	Visualisation Component						
Video #	Grass	Aerial Overlay	Colour Map	Texture Map	3D Models	Exploded View	Pillar View
1-1	X		X				
1-2		X	X				
1-3	X			X			
1-4		X		X			
2-1	X		X				
2-2		X	X				
2-3	X			X			
2-4		X		X			
3-1	X		X		X	X	
3-2		X	X		X	X	
3-3	X			X	X	X	
3-4		X		X	X	X	
4-1	X		X		X		X
4-2		X	X		X		X
4-3	X			X	X		X
4-4		X		X	X		X

Table 6-8: A breakdown of the visualisation methods found in each task based video

In the table above the video number column should be read as **task number – video number**. For example **Video 1 – 3** is selecting a land parcel (task **1**) and is the third (**3**) combined visualisation method used (grass overlay and texturing mapping in the case of Video 1-3).

6.8 Summary of Testing Strategy

The objective of this chapter was to present a more extensive review of the existing literature on evaluating the effectiveness of visualisation tools than that found in the literature review in **Chapter 2.4**. This was presented through identifying four common categories that visualisation evaluation falls into. These categories – **usability evaluation**, **case studies**, **controlled experiments comparing two or more tools** and **comparing design elements** – provided insight into the different methods used to evaluate visualisation tools. Visual Preference Surveys have been identified as a way to rank a participant's preference for a given image. This same technique is also shown to work when asking participants to rank comparative images. It has also been shown that usability studies are commonly used to determine if a tool is suited to specific tasks of information retrieval. Finally, the visual preference survey and task based testing used as R.S.V.T's testing strategy was presented.

Chapter 7 – Results & Analysis

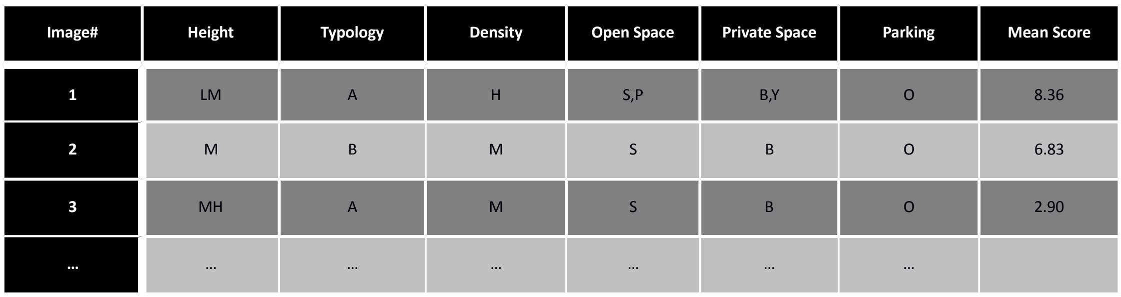
The purpose of this chapter is to present the results gathered from the testing strategy detailed in the previous chapter. The chapter begins with justification of the chosen methods to analyse the results from the testing strategy to illustrate that the data collected was used in ways that are consistent with the current literature. Next, the results gathered through the implementation of the testing strategy are presented before concluding the chapter with a summary of the findings.

7.1 Methods of Results Analysis

Two methods of visualisation evaluation were chosen as part of R.S.V.T's testing strategy to test user preference of landscape (Visual Preference Survey – see **Chapter 6.6**) and the usability of the R.S.V.T software (Task Based Testing – see **Chapter 6.7**). It was necessary to identify the most appropriate ways to analyse the testing results through a review of methods that are commonly used by other researchers. Two methods of analysis were found to be commonly used, *image properties matrix* – which have been used to display the results of visual preference survey (Bailey 2001, Grossardt 2004), and *correctness of response* - which has been used to display the results of usability and task based testing (Koua 2006, Liu 2012, Seipel 2012).

7.1.1 Analysing Visual Preference Survey Results

The visual preference survey implemented in the previous chapter was significantly influenced by the experiments conducted by Bailey (2001) and Grossardt (2004). Their aim was to identify user preference for a particular image as well as the individual design elements that make up the image. Bailey – who experimented with user preference of design elements in the field of highway management – and Grossardt – who was testing the user preferences of conceptual light rail design – both used means scoring of design elements when analysing their results. The means scoring is presented as an *image properties matrix* and an example of Grossardt's can be seen in table 7-1 below.



Image#	Height	Typology	Density	Open Space	Private Space	Parking	Mean Score
1	LM	A	H	S,P	B,Y	O	8.36
2	M	B	M	S	B	O	6.83
3	MH	A	M	S	B	O	2.90
…	…	…	…	…	…	…	

Table 7-1: The image properties matrix for visual preference results analysis (Grossardt, 2004)

The table above shows an image properties matrix. Each row corresponds to an image that was shown to the participants during testing. Each column is representative of a design element that can be found within the image and the character encoded cells indicate which visualisation method was used to visualise the design element. The character encoding for the height design element can be interpreted as the following: **L**ow-**M**edium (**LM**), **M**edium (**M**) and **M**edium-**H**igh (**MH**). This is also known as the design vocabulary (Grossardt, 2004; Blandford, 2008). The final column in the table shows the mean preference score, with Grossardt opting for a 10 point Likert scale – values closer to 10 are greatly preferred while values closer to 0 are disliked. The image properties matrix is useful in identifying visualisation methods of design elements that have a positive or negative impact on the overall means score by comparing the results of images that are created using the same design elements but different visualisation methods.

7.1.2 Analysing Task Based Results

A *correctness of response* table is used to display the results of task-based testing. The correctness of response has been used in task-based visualisation evaluations of landscape visualisation software by Cockburn & McKeznie (2000; 2001; 2002), Koua (2007) and Van Lammeren (2010) as a measure of performance when evaluating visualisation. To carry out correctness of response analysis each task is ranked by the percentage of correct answers associated with the participant's responses for that task. This is consistent with other task based testing methods employed by Steinitz (1990), Ewing (2001) and Chang (2009). A response that is correct or accurate is given a value of 1 while wrong or inaccurate responses are given a value of 0. Mean success response rates are then calculated for each task.

7.2 Results – Visual Preference Survey

A design vocabulary for R.S.V.T's visualisation methods can be found in table 7-2. Each visualisation method relates to a corresponding design element which can be found in table 7-3.

Visualisation Method	Character encoding
Grass Overlay	G
Aerial Overlay	A
Colour Mapping	C
Texture Mapping	T
3D models	3D
Textual Interface	TI
3D Pillars	P
Exploded View	E

Table 7-2: The character encoding for R.S.V.T's visualisation methods

Design Element	Visualisation Method		
Landscape Cover	G	A	
Land Use Change	C	T	3D
Land Parcel Parameter Info	T	P	E
Skylark Agent	3D		

Table 7-3: The design vocabulary for R.S.V.T

An image properties matrix was created from the results of the 31 participant responses when asked to rate the 19 images created as part of the visual preference survey carried out during the testing phase of R.S.V.T. These images contain a scene from RepastS and the corresponding scene visualised under R.S.V.T. The images, as well as the Likert scale values (from 1 – 5), can be found in Appendix C for reference. The results can be found in table 7-4 below.

Image #	Topographical Cover	Land Use Change	Land Parcel Parameter	Skylark Agent	Mean Score	Standard Deviation
1	G	-	-	-	3.91	1.17
2	A	-	-	-	3.94	1.24
3	G	T	-	-	2.1	1.38
4	A	C	-	-	3.03	1.4
5	G	C	-	-	2.31	1.03
6	A	T	-	-	3	1.11
7	G	T	E	-	2.69	1.38
8	G	C	E	-	3.34	1.01
9	A	T	E	-	3.59	1.13
10	A	C	E	-	3.78	1.09
11	G	T	P	-	3.19	1.18
12	G	C	P	-	3.5	1.1
13	A	T	P	-	3.16	1.17
14	A	C	P	-	3.72	1.02
15	G	3D	-	-	3.3	1.43
16	G	3D	-	3D	3.67	1.31
17	G	3D	-	3D	3.97	1.06
18	A	3D	-	3D	3.72	1.1
19	A	3D	-	3D	3.72	1.2

Table 7-4: The image properties matrix for the Visual Preference Survey

The figure above illustrates which visualisation method was used for each of the design elements present in each of the visual preference survey images. Images started with one design element, topographical cover and additional design elements were added, making the visual scene more sophisticated. The topographical cover design element is the only ever-present design element and the two visualisation methods of grass overlay and aerial photography are compared first. Both scored highly, with a mean preference of 3.91 and 3.94 respectively. These images had the second and third highest overall mean preference of all the images.

Images number 3 – 6 showed the available combinations of the topographical cover and land use change design elements. This particular grouping of design elements had the lowest overall preference scores with an average score of 2.1 and 2.31 for a combination of grass overlay/texture mapping and grass overlay/colour mapping. This suggests that the grass overlay is particularly off putting to participants although when aerial photography was combined with both the colour mapping and texture mapping the participant responses were fairly neutral with preference scores of 3.03 and 3 respectively.

The next visualisation methods added to the images are those of the land parcel parameters design element (**image numbers 7 – 14**). All of these images were generally preferred however the combination of the grass overlay, texture mapping and the exploded view was not popular, scoring only an average preference of 2.69. This particular combination was described as cluttered and out of place by some of the participants, referring to the different visual styles that each design element combining to poor effect. This can be expected given that previously mentioned combination of the grass overlay and texture mapping scored the lowest preference score with only 2.1. It is worth noting that the two highest mean preference scores for the combination of these three design elements were for the combinations of aerial photography, colour mapping and exploded view which scored an average preference of 3.78 and aerial photography, colour mapping and pillars which scored an average of 3.72.

The final design elements added were those that used 3D models as a visualisation technique. **Images 15 – 19** all scored highly with the combination of aerial photography, 3D land use models and 3D skylarks being the most preferred of the group and of the visual preference survey with a preference score of 3.97.

The mean preference for each image can be shown as a scatter plot in figure 7.1 below, which illustrates the spread of the mean responses.

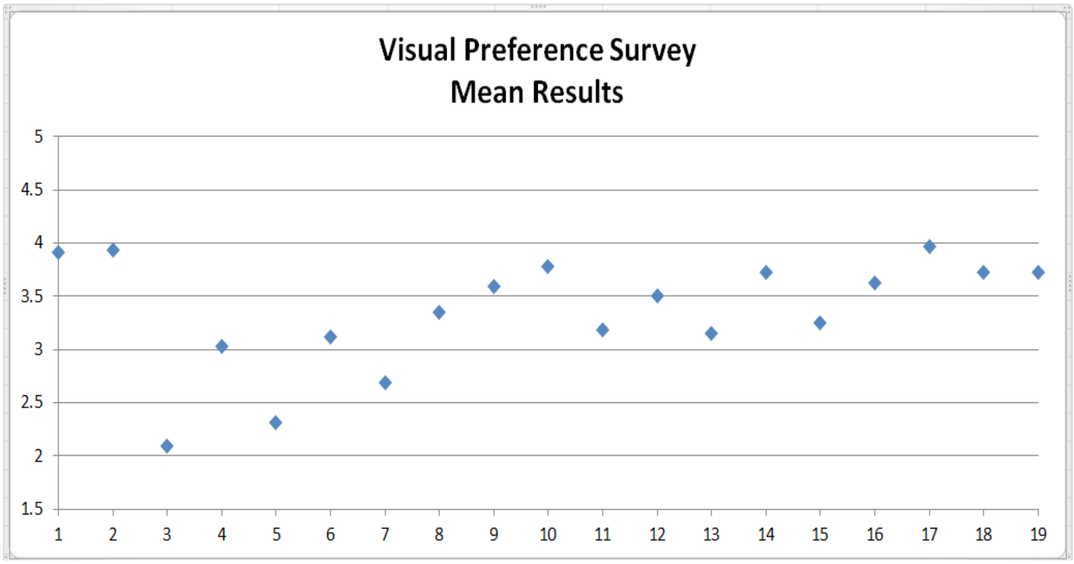


Figure 7.1 Scatter plot of the mean preference scores for each image of the Visual Preference Survey

The figure above confirms that the majority of images were rated higher than 3 (where 3 is no preference for either RepastS or R.S.V.T). 15 of the 19 images had a mean preference of greater than 3, indicating a visual preference towards R.S.V.T while only 3 images (3, 5, 7) showed a visual preference towards RepastS. This suggests that in general the participants preferred the visual output of R.S.V.T compared to the visual output of RepastS when visualising comparative design elements.

7.3 Results – Task Based Testing

This section presents the results of the task based testing through the use of correctness of response percentages that were calculated for each task. A brief reminder of the tasks and questions posed for each task can be found in table 7-5 below.

Task	**Question**
Selecting a land parcel	Please indicate which Land Parcel was selected as well as any additional information about the land parcel such as crop type, crop value and skylark health within the land parcel?
Locate general model information	
Identify land use change	Please indicate which land use is most abundant within the video?
Interpret land parcel parameters	Please indicate which land parcel parameter is the highest and which land parcel parameter is the lowest?

Table 7-5: Task list and questions asked during the task based testing

The table shows three questions that were asked to cover four different tasks. Each question was asked under different visual conditions by changing the visualisation methods of R.S.V.T's design elements. The variable design elements were topographical cover (switching between grass overlay and aerial photography), land use change (switching between colour mapping and texture mapping) and land parcel parameters (switching between exploded view and 3D pillars). The design elements were changed to determine what impact, if any, different visualisation had on accuracy of response. The videos shown to the participants can be found in Appendix D on the attached DVD.

7.3.1 Question 1 – Selecting a Land Parcel & Locate General Model Information

Participants viewed a video showing the selection of a land parcel. They were asked to identify the land parcel by ID and any other information they could find through R.S.V.T's User Interface. The results can be seen in table 7-6.

Visualisation Methods	% of Correct Responses
(A)erial + (C)olour	69.5%
(A)erial + (T)exture	68.8%
(G)rass + (T)exture	68.8%
(G)rass + (C)olour	60.9%

Table 7-6: Ranked % of correct responses for task-based testing question 1

The overall correctness of response was between 60% and 70% bracket for all visualisation methods. When combing the aerial photography with the colour map the % of correct responses was at its highest with 69.5% of participants answering correctly.

7.3.2 Question 2 – Indicate the Most Abundant Land Use

Participants were shown videos of a full model year and asked to identify which was the most abundant land use type during this time. This was to ensure participants could identify land use changed during a model run and correctly identify the visual methods used to portray the land use cover type. The results of this task under each of the visual conditions set can be seen in table 7-7.

Visualisation Methods	% of Correct Responses
(A)erial + (C)olour	84.38%
(A)erial + (T)exture	65.63%
(G)rass + (C)olour	53.13%
(G)rass + (T)exture	40.63%

Table 7-7: Ranked % of correct responses for questions on abundant land use

Again it can be seen that the combination of aerial photography and colour mapping being ranked first with 84.38% of participants answering correctly when these visualisation methods were used. This is almost 20% higher than the second placed visualisation methods of aerial photography and texture mapping with 65.63% of correct answers. The lowest response score came from the combination of the grass overlay and texture mapping with only 40.63% of participants answering correctly. This is the first clear indication that different visualisation methods have significant impact on the accuracy of data retrieval by the participants.

7.3.3 Question 3 – Identifying High & Low Land Parcel Parameters

Participants were shown videos of both the exploded view and 3D pillars. These techniques were designed to aid the viewer interpret land parcel palrameters. The participants were asked to identify the highest and lowest values of a land parcel's parameters which were representative of economic, environmental and social data contained within the agent based model. The results of the exploded view are presented in table 7-8 and the results of the 3D pillars presented in table 7-9.

Visualisation Methods	% of Correct Responses
(G)rass + (T)exture	68.75%
(A)erial + (T)exture	68.75%
(G)rass + (C)olour	54.69%
(A)erial + (C)olour	43.75%

Table 7-8: Ranked % of correct responses for questions on high and low land parcel parameters using the exploded view visualisation method

Visualisation Methods	% of Correct Responses
(G)rass + (T)exture	87.5%
(A)erial + (T)exture	87.5%
(G)rass + (C)olour	85.94%
(A)erial + (C)olour	84.38%

Table 7-9: Ranked % of correct responses for questions on high and low land parcel parameters using the 3D pillars visualisation method

Comparison of table 7-8 and table 7-9 immediately shows a large variance in correct responses. The top ranked visualisation method of 3D pillars (table 7-9**)** showed a correct response rate of 87.5%. The highest ranked visualisation method for the exploded view (table 7-8) is 18.75% lower, with a correct response rate of 68.75%. The spread of correct responses across the visualisation methods are also much closer together for the 3D pillars. The lowest visualisation method of aerial photography and colour mapping generating a correct response rate of 84.38%, only 3.12% difference from the highest ranked visualisation methods. This suggests that 3D pillars as a visualisation method has the ability to convey data accurately to the user.

This was not the case for the exploded view however. The top ranked visualisation of grass overlay/texture mapping and aerial photography/texture mapping was acceptable at 68.75% however this was 25% higher than the worst ranked visualisation methods of aerial photography and colour mapping which had a correct response rate of 43.75%. This indicates that the exploded view visualisation method only has use under specific visual constraints and is not well suited for data extraction.

7.3.4 Overall Response Rates

All response rates were grouped by task and compared to the response rate of other tasks and the results made available through a bar chart which can be seen in figure 7.2.

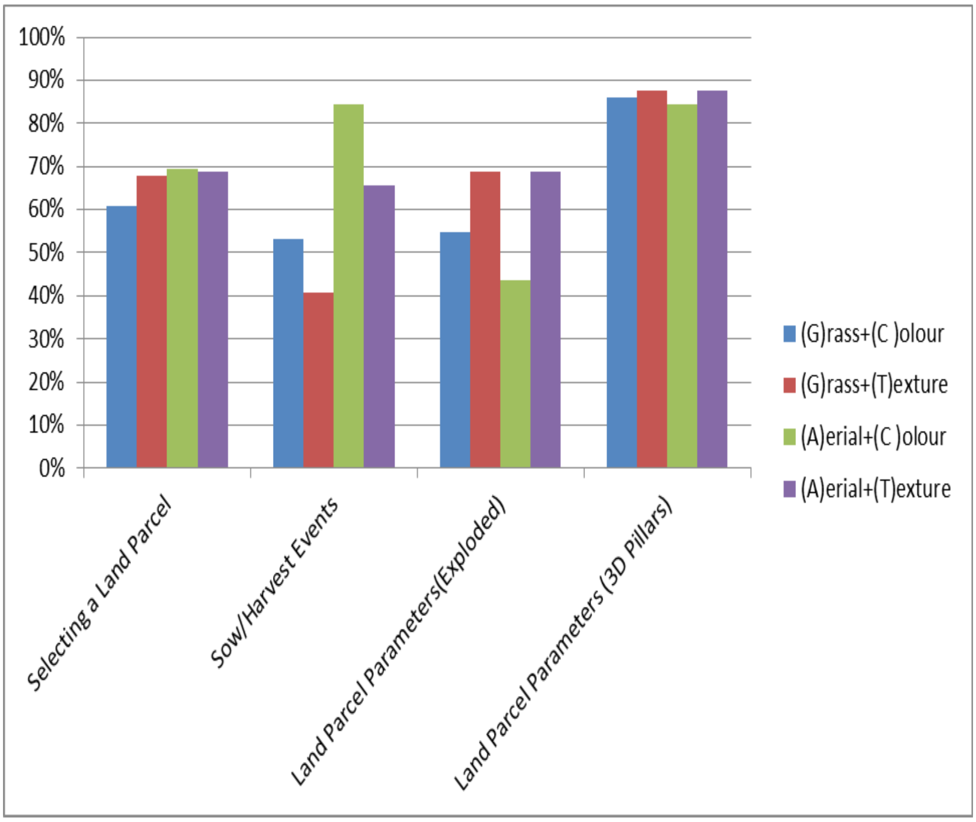


Figure 7.2: Combined correctness or response scores for all task based testing

From **figure 7.2** it can be determined that the most successful task identifying land parcel parameters using the 3D pillars. As mentioned earlier, the difference in correctness of response between the exploded view and the 3D pillars seems to be attributable to the implementation of the exploded visualisation method itself, proving to be useful only under certain visual conditions while the 3D pillars proved to be effective at portraying data under a wider range of visual circumstances.

Although some of the lowest correctness of response rates were found in **Question 2 – Indicate Abundant Land Use** there was found to be one outstanding combination of visualisation methods when combining aerial photography and colour mapping. This could be attributed to the nature of the task. The task involved watching dynamic textures or colours which would be considered a non-trivial task.

7.4 Analysis of Results

Chapter 6 outlined the testing objectives as well as particular questions that needed to be answered (**see chapter 6.5**). A reminder of these questions:

1) Does the user find more aesthetically pleasing when shown the visual scenes of R.S.V.T or RepastS?
2) Which of R.S.V.T's visual methods and design elements were preferred by participants?
3) Can the participant extract information about model and land parcel parameters from R.S.V.T?

The mean preference scores were presented to satisfy ***testing question 1***. The mean preference scores show that only 3 out of 19 2D RepastS images shown were preferred to their corresponding 3D counterpart. Of the 16 remaining images, all except image number 6 showed some degree of positive emotional response.

During the analysis of visual preference results it was shown that certain combinations of design elements were preferred to others. This can be seen by the low visual preference scores given to image containing the grass overlay and texture mapping visualisation methods while images with the aerial photography and colour mapping scored considerably higher. The 3D models were also shown to be liked by participants, scoring highly under the variable visual conditions, showing it to have more impact than that of the 2D visualisation methods of texture mapping and colour mapping.

Finally, through the use of correctness of response tables and graphs it was shown that users of the tool can use it to extract information. While some individual visualisation methods scored poorly, the lowest correctness for any given question was 68.75% with the highest correctness being 87.5%. The low scoring of the exploded view was identified during the post-session discussions. The exploded view used transparency to indicate the health of land parcel parameters. This proved problematic for some participants when the exploded land parcel overlapped pre-existing land parcels already on the terrain. The alternative strategy using 3D pillars to display land parcel parameters proved much more successful with correctness of response rates well into the 80% range.

Summary Recommendations

1. Information retrieval relies upon multiple visualisation methods of design elements

The results of the task-based testing show that the landscape visualisation combination of aerial photography overlay with coloured mapped land use cover scored highest for information retrieval in 50% of the tasks and lowest in the remaining task. This suggests that no one visualisation method is appropriate for mixing with other design elements.

2. Visualisation methods should be chosen with care

The exploded view and pillared gives an example of two different visualisation methods of the same design element. Both methods were designed to offer the user information on the current sustainability of a particular land parcel. However the results of the task based testing clearly show that one visualisation is superior for carrying out the task in hand. To ensure that only the most appropriate visualisation methods are developed it is recommended that the stakeholders be continually involved in the development of new visualisation methods.

3. 3D techniques improve the quality of a scene

The results of the visual preference survey show that generally the 3D user interface is preferred over the 2D interface. The mean scores are at their highest when more advanced graphical rendering is being displayed. This is demonstrated by the high scores for images 16, 17, 18 & 19 of the visual preference survey in which the 3D models of skylarks and crop models were being rendered.

Chapter 8: Conclusions & Future Work

8.1 Introduction

This section concludes the research by assessing the progress made towards the original aims and objectives and whether the hypothesis of the research project can be accepted or rejected. The original aims and objectives are:

Aims and Objectives

- Evaluate gaps in the visualisation of 2D and 3D ABM output
- Determine suitable computing and computer games technology techniques that can be applied to 3D ABM
- Create a custom 3D visualisation system that is capable of communicating LUCC ABM output to non-expert stakeholders
- Test the effectiveness of the tool by carrying out visualisation evaluation methods to ensure stakeholder understanding and engagement

8.1.1 Evaluate Gaps in the Visualisation of 2D and 3D ABM Output

In **Chapter 2** a review of the existing tools for 2D and 3D ABM output was provided. It was shown that traditional 2D output of ABMs lack the visual quality that modern hardware can now provide via computer games technology. The existing tools that provide 3D output fall into two categories they are either loosely coupled or tightly coupled to the ABM. The current implementations of both types of 3D ABMs have been shown to have limitations.

Closely coupled systems require the model and visualisation to be thought of “as one”. This was shown to be problematic as current ABM toolkits do not have “out of the box” support for 3D visual output and developing a 3D virtual environment for a model is a non-trivial task. The alternative for closely coupled systems is to move the ABM away from the traditional ABM toolkit and into a programming environment more suited to graphical programming such as C++ with DirectX. This is also a time consuming task as the ABM has to be ported to another language without the benefits that ABM toolkits offer modellers.

Loosely coupled systems negate the need for porting an ABM. However they were shown to have either poor updating performance (the model cannot transfer data fast enough to the visualisation), which hinders the goal of realtime interaction and in some cases no interactivity between the user and the visualisation. Poor updating performance has been shown to be caused through the use of existing Virtual Environments such as Second Life. To have full control over the speed and priority of incoming data packets a bespoke solution was required.

8.1.2 Determine suitable computing and computer games technology techniques that can be applied to 3D ABM

Chapter 5 highlighted that R.S.V.T can be thought of as a Web Service and, as such, made use of methods and technologies commonly found in other Web Services. The purpose of a Web Service is to decouple the information requested by the user from the acquisition, processing, storage and transmission of data.

Chapter 4.4.3 details the design elements and visualisation methods that were developed for use within R.S.V.T. The choice of design elements were based on the ABM indicators and the subset indicators used by R.S.V.T which was presented in **Chapter 3.6**. The design elements and their corresponding visualisation methods were:

- Terrain Topographical Cover
    - Single Textured
    - Multi Textured
    - Aerial Photography
- Land Use Representation
    - Colour Mapping
    - Texture Mapping
    - 3D Crops
- Social, economic and environmental indicators
    - Exploded View
    - Pillared View

8.1.3 Create a custom 3D visualisation system that is capable of communicating LUCC ABM output to non-expert stakeholders

Chapter 4 and **Chapter 5** describe the three main components of R.S.V.T. These were:

- The Model (Lunan ABM)
- The View (3D User Interface)
- The Presenter (Data exchange via Protocol Buffers)

The combination of these components has been shown to result in a loosely coupled 3D ABM as proven in the pilot testing carried out in **Chapter 5**. The development process of the following visual components was discussed: recreating accurate terrain including sky; overlaying shapefiles; 3D mesh building from triangulation and the implementation of the visual methods for the design elements mentioned in previous section (**section 8.1.2)**. The development of offline-mode makes it possible to run previously simulated scenarios.

The 3D User Interface was created as a bespoke visualisation engine which allowed the implementation of visualisation techniques that cannot be found in current 3D ABMs. The loose coupling of the 3D User Interface and the model was important because development of the ABM was not finished until 2012. This allowed Eléonore Guillem and Dave Murray-Rust to make changes and additions to the Lunan ABM without impacting the code being developed for R.S.V.T.

8.1.4 Test the effectiveness of the tool by carrying out visualisation evaluation methods to ensure stakeholder understanding and engagement

The evaluation of the effectiveness of R.S.V.T and the subsequent results can be found, in full, in **Chapter 6** and **Chapter 7** respectively. The testing strategy set out the following hypotheses:

1) The visualisation is sufficiently effective at conveying Land Use Cover Change.
2) The chosen communication protocol will transfer data at an acceptable rate.
3) The methods of data capture and storage are flexible and scalable to support further development

Visualisation of Land Use Cover Change

The visual preference survey carried out in **Chapter 7** shows participants found the 3D output of R.S.V.T to be more visually appealing than the 2D output offered in RepastS with 15 out of 19 images rated as “better than 2D counterpart” and that colour mapping of land use was the preferred visualisation method of the land use design element. The aerial photography design element was the preferred method of visualising the terrain cover although the combination of aerial photography, texture mapping land use and exploded land parcel indicators was unpopular. After discussions with each group it became clear that the textures and colours used for each of the 3 visualisation methods (aerial photography, texture mapping, exploded view) were overwhelming and the participants were unsure which texture/colour belonged to which visualisation method. All visualisation techniques received high scores when mixed with the 3D crop models and/or 3D skylarks.

The accuracy of response varies between design elements which can be seen in **table 7.8** and **table 7.9**. There was at least an 84% accuracy of response when participants extract data using the 3D pillar view compared to the poorer performing exploded view which had an accuracy of response of, at best 68% and 43% at worst.

Overall, participants preferred the 3D user interface over the 2D interface and that the accuracy of response is influenced by the visualisation methods used to render design elements. Therefore it is suggested that further research into suitable visualisation methods for 3D ABM output is required.

Speed of Data Transfer

The advantages of a loosely coupled model are presented in **chapter 2.3** which shows the need for a communications protocol to bridge the data gap between the separate codebases of R.S.V.T and the ABM. Protocol Buffers was selected as the communication protocol of choice. A messaging model was designed (see **chapter 4.3**) and tested against other communication protocols (see **chapter 5.4**). Of the three communication protocols tested, Protocol Buffers showed the fastest times at both serialization and deserialization (see **figure 5.5**) and by implementing Protocol Buffers, R.S.V.T is able to render real-time land use cover changes driven by a loosely coupled ABM.

Optimised Data Storage

Protocol Buffers was shown to be a workable solution to ensuring high-speed data transfer between the ABM and 3D User Interface however the impact on performance due to further development, including the addition of new visualisation methods and design elements, could not be quantified without first knowing what additional visualisation methods would be added.

The offline-mode, presented in **chapter 4.3.5**, shows a novel way of storing data in a NoSQL database. The model's codebase was modified so that a model run, even a simulated run using the 2D RepastS interface, would save its Presenter message to the database – using the same sequencing and format that R.S.V.T would expect from a live model run. This ensures R.S.V.T could load a previously run simulation from the database with a reduced CPU load as this approach negates the need to have the model running parallel with the visualisation. This ensures that further development can be done on both the ABM and the 3D User Interface whilst limiting resource bottlenecks.

8.2 Conclusion of the validity of the original hypotheses

The original project hypotheses is:

"Can 3D visualisation be used to portray ABM output to non-expert stakeholders that accurately conveys LUCC in a real-time immersive environment?"

The literature review in **Chapter 2** shows that there is currently no framework that is able to output ABM data to a 3D interactive environment in real-time. **Chapters 4** and **5** show that computer games technology can be used to build a 3D virtual environment that is capable of receiving data from an independent and loosely coupled ABM that dynamically updates the 3D virtual environment as the states of the ABM change. R.S.V.T's real-time interaction, when driven by the model, was significant. The level of performance reached when evaluating the communication protocols (see **chapter 5**) was much better than the level of performance seen using Repast Simphony. This was achieved by loosely coupling both ABM and the 3D User Interface through Protocol Buffers which provides a stable platform for the further development of both the ABM and the 3D User Interface as shown by the development of R.S.V.T's offline mode which made use of Protocol Buffers and MongoDB.

Chapter 6 and **Chapter 7** show that in many ways the tool is effective however it is acknowledged that more work needs to be done to improve the effectiveness of some of the visualisation methods employed in R.S.V.T. The exploded view has shown it is not well equipped to portray information to the end user. Additionally, when mixing various design elements together there was confusion over which texture/colour was representative of which design element. This could be a matter of the scene simply being too crowded but the colour scales and textures could still be improved. During the testing sessions carried out as part of the testing strategy it was clear that the tool was capable of provoking discussion and debate among the participants and many commented on the overall visual quality of 3D user interface.

8.3 Future Work

The project was designed to be generic, ensuring that adding support for another LUCC ABM would be possible. The terrain engine and shapefile processor is capable of handling new regions assuming the correct data is supplied (heightmap and associated shapefile). The larger and more complicated issue surrounding making the tool generic is ensuring that modellers, devoting their time to the ABM, can quickly and easily add 3D features to their ABM knowing the tool will interpret them correctly.

For this to be a reality a larger study of the plausible visualisation methods would need to be carried out, identifying agreed upon visualisation methods for typical land use model behaviour. This was identified as a partly failed objective within the project whereby some users had difficulty in extracting information when certain visualisation methods were rendered together. Future work should look at the methods modellers would like to see incorporated into land use change ABMs. Allowing some customisation for developers and users such as changing colour scales for the colour mapping technique or adding their own land use textures would also be required.

Another area in which the tool could improve would be in its storage and communication of previously simulated models. It would be useful for the tool to have the option of automatically uploading completed models run (either run in Repast Simphony, R.S.V.T or without a user interface) to a centralised online repository. This would allow researchers to batch run their models and then have the option of viewing a list of their simulations. This would ensure easy loading and re-running of interesting results as well as the option of sharing these 3D-capable ABMs with colleagues. Other than the visualisation of land use, it would be feasible to develop a 3D User Interface for many of the domain-specific problems ABM helps to better understand. The tool could be adapted to visualise the proposed emergency evacuation of a building, rural flooding. The 3D User Interface already has components for rendering user-defined models, detecting collisions and model animation.

The components that make up the technology stack used for this research project are capable of being extracted from the codebase and put to use on systems completely unrelated to agent based modelling. In fact, there is no requirement for a Model in the traditional sense it has been used throughout this project – the Model can be a static and already defined dataset such as voting results for General Elections. The generic nature of the User Interface allows for the possibility of putting in a different heightmap so rather than having the heightmap of Lunan there would be a heightmap of the United Kingdom. In place of the land parcel shapefile there would be a shapefile showing the current (and, if require, previous) electoral boundaries.

This electoral visualisation would benefit from some of the visualisation techniques already present and available within R.S.V.T such as colour mapping for colouring individual boundaries to represent which political party holds which seat and the pillared view which could show the % of votes each party delegate received from each area. Although it is common for voting data to be made publicly available this has not been the case for many large government collected datasets. Many of these datasets have become so large or complex that traditional database management tools are no longer appropriate for querying the data contained within them. These datasets are known as “Open Data” and R.S.V.T components can help visualise these datasets.

The purpose of “Open Data” is to present finalised datasets to researchers, data scientists and the general public free of charge. By making the data freely available to anyone the hope is that those who download the dataset will create tools for querying, evaluating and visualising the data within the datasets. This provides the company or government that released the open data with an ever increasing number of people working with the dataset – many of whom validate parts of the dataset or report inaccuracies should they arise. Some examples of dataset currently available from the UK government’s data website include annual public spending, salary details, crime statistics, 100+ years of meteorological data, bathing water quality and the number of abandoned shopping trolleys in Bristol’s rivers. From this list the data was made available through Shapefiles, Comma-Separated Values (CSV), Excel spreadsheets and html files. With the exception of html files, R.S.V.T is capable of handling the data formats listed above and would be capable of visualising information from those datasets with limited changes to the codebase.

Finally, there is one other application domain that the technology stack of R.S.V.T could be integrated with and that is the domain of visualising “Big Data”. Big Data, not to be confused with Open Data, is the description given to data so large and complex that traditional methods of data capture, analysis, storage and querying are no longer adequate. Big Data refers to the collective process of multiple machines or servers taking on this responsibility rather than just one server being responsible for the database. When considering datasets with massive storage requirements (5 Terabytes+) it forces the developer to consider the practicalities of accessing, editing and adding records. Traditional RDBMS have proven to be inadequate for this type of data with developers choosing NoSQL database varieties when dealing with massive datasets due to its horizontal scaling and sharding capabilities that were discussed in **Chapter 4.3.5.2**. The ability of R.S.V.T to interact with MongoDB suggests it could be used to integrate with a database containing much larger amounts of data. For example, Google Flu Trends was shown to reliably show areas of high influenza infection rates within the US and Europe by collecting and analysing search data. By connecting to their NoSQL database R.S.V.T components can interact with the data and produce visualisation based on the real-time results of search trends.

References

Abildtrup, J, E Audsley, M Feketefarkas, C Giupponi, M Gylling, P Rosato, and M Rounsevell. (2006). "Socio-economic scenario development for the assessment of climate change impacts on agricultural land use: a pairwise comparison approach." *Environmental Science & Policy* 9, no. 2 (April 2006): 101-115. http://linkinghub.elsevier.com/retrieve/pii/S1462901105001553.

Adobe. (2013). Tablets trump smartphones in global website traffic. Available at http://blogs.adobe.com/digitalmarketing/digital-index/tablets-trump-smartphones-in-global-website-traffic/

Akpata, E., & Riha, K. (2004). Can Extreme Programming be used by a Lone Programmer ? 167–175.

Apache. (2010). Apache Thrift Download. http://thrift.apache.org/about/

Appleton, K., Lovett, A., Sünnenberg, G., & Dockerty, T. (2002). Rural landscape visualisation from GIS databases: a comparison of approaches, options and problems. *Computers, Environment and Urban Systems*, *26*(2), 141-162.

Appleton, K., & Lovett, A. (2003). GIS-based visualisation of rural landscapes: defining "sufficient" realism for environmental decision-making. *Landscape and Urban Planning*, *65*(3), 117–131. doi:10.1016/S0169-2046(02)00245-1

AppsOnTheMove. (2010). "AppsOnTheMove | iPhone iPod and iPad Productivity. Available at http://appsonthemove.com/

Archirapatkave, V., Sumilo, H., See, S. C. W., & Achalakul, T. (2011). GPGPU Acceleration Algorithm for Medical Image Reconstruction. *2011 IEEE Ninth International Symposium on Parallel and Distributed Processing with Applications*, 41–46.

Ashraf, G., & Zhou, J. (2006). Hardware accelerated skin deformation for animated crowds. *Advances in Multimedia Modeling*, 226–237. Retrieved from http://link.springer.com/chapter/10.1007/978-3-540-69429-8_23

Atkin, A. *Peirce's Theory of Signs*. (2010). The Stanford Encyclopedia of Philosophy. Edward N. Zalta (ed.). Accessed December 29, 2012.

Audsley, E, K Pearn, C Simota, G Cojocaru, E Koutsidou, M Rounsevell, M Trnka, and V Alexandrov. (2006). "What can scenario modelling tell us about future European scale agricultural land use, and what not?" *Environmental Science & Policy* 9, no. 2 (April 2006): 148-162

August, P. et al., (2001). Agent-Based Models of Land-Use and Land-Cover Change. *Proceedings of an*, (6). Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.154.7647&amp;rep=rep1&amp;type=pdf [Accessed August 2, 2011].

Bailey, K., Brumm, J., & Grossardt, T. (2002). Integrating visualization into structured public involvement: case study of highway improvement in central Kentucky. *Transportation Research Record: Journal of the Transportation Research Board*, *1817*(1), 50-57.

Balaji, S. (2012). WATEERFALLVs V-MODEL Vs AGILE : A COMPARATIVE STUDY ON SDLC, *2*(1), 26–30.

Bandini, Stefania, Manzoni, Sara and Vizzari, Giuseppe (2009). 'Agent Based Modelling and Simulation: An Informatics Perspective'. Journal of Artificial Societies and Social Simulation 12(4)4 <http://jasss.soc.surrey.ac.uk/12/4/4.html

Batty, M. (2003). Agent-based pedestrian modelling. *Advanced spatial analysis: The CASA book of GIS*, 81-106.

Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., & Thomas, D. (2001). Manifesto for agile software development.

Belaid, M.A., (2003). Urban-Rural Land Use Change Detection and Analysis Using GIS & RS Technologies Urban-Rural Land Use Change Detection and Analysis Using GIS & RS Technologies.

Berkhout, F, J Hertin, & Jordan. (2002). "Socio-economic futures in climate change impact assessment: using scenarios as 'learning machines'." *Global Environmental Change* 12, no. 2 (July2002):83-95.

Berners-Lee, Tim., (2010) "Long Live the Web." *Scientific American* 303.6 (2010): 80-85

Bishop, I, C Stock, and K Williams.(2009). "Using virtual environments and agent models in multi-criteria decision-making." *Land Use Policy* 26, no. 1 (January 2009): 87-94. http://linkinghub.elsevier.com/retrieve/pii/S0264837708000173. Centre, U C L, for Advanced, and Spatial Analysis. "PAPERS." *Urban Systems* 44, no. 0 (n.d.): 0-34.

Blandford, B., Grossardt, T., Ripy, J., & Bailey, K. (2008). Integrated Transportation and Land Use Scenario Modeling by Visual Evaluation of Examples: Case Study of Jeffersonville, Indiana. *Transportation Research Record: Journal of the Transportation Research Board*, *2076*(1), 192-199.

Boehm, B. W. (1991). Software risk management: principles and practices.*Software, IEEE*, *8*(1), 32-41.

Bohunovsky, L., Jäger, J. & Omann, I., (2010). Participatory scenario development for integrated sustainability assessment. *Regional Environmental Change*, 11(2), pp.271–284. Available at: http://www.springerlink.com/index/10.1007/s10113-010-0143-3 [Accessed February 9, 2012].

Browne, S., J. Vickery & D. Chamberlain (2000): Densities and population estimates of breeding Skylarks Alauda arvensis in Britain in 1997, Bird Study, 47:1, 52-65

Borjeson, L. et al., (2006). Scenario types and techniques: Towards a user's guide. *Futures*, 38(7), pp.723–739. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0016328705002132 [Accessed July 16, 2011].

Borsboom-van Beurden, J. A. M., Van Lammeren, R. J. A., Hoogerwerf, T., & Bouwman, A. A. (2006). Linking land use modelling and 3D visualisation. In *Innovations in design & decision support systems in architecture and urban planning* (pp. 85-101). Springer Netherlands.

Bouwman, A. A. (2006). Linking land use modelling and 3D visualisation. Retrieved from http://www.lumos.info/resources/File/lu_models_and_3d.pdf

Burback, R. (1998). Software engineering methodology: the Watersluice. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.62.3186&rep=rep1&type=pdf

Box, D., Ehnebuske, D. & Kakivaya, G., (2000). Simple object access protocol (SOAP) 1.1. , (May) Technical Specification paper. Available at:

http://www.immagic.com/eLibrary/ARCHIVES/SUPRSDED/W3C/W000520N.pdf [Accessed April 26, 2012].

Byrd, D. (1999). A scrollbar-based visualization for document navigation. *Proceedings of the fourth ACM conference on Digital libraries - DL '99*, 122–129.

Carter, T.R., La Rovere, E.L., Jones, R.N., Leemans, R., Mearns, Nakic´enovic ´L.O., Nakicenovic. (2001). Developing and applying scenarios. In: McCarthy,J.J., Canziani, O.F., Leary, N.A., Dokken, D.J., White, K.S.(Eds.), Climate Change 2001: Impacts, Adaptation and Vulnerability. Cambridge University Press, Cambridge, pp. 145–190.

Castiglione, F., (2006). Agent Based Modelling. [Online]. Available at: http://www.scholarpedia.org/article/Agent_based_modeling [Accessed November 2012].

Chamberlain, S., Sharp, H., & Maiden, N. (2006). Towards a framework for integrating agile development and user-centred design. In *Extreme programming and agile processes in software engineering* (pp. 143-153). Springer Berlin Heidelberg.

Chang, R. & Ziemkiewicz, C., (2009). Defining insight for visual analytics. Available at: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4797511 [Accessed May 28, 2013]

Christie, S. (2008). XNAInput Code Library. Available at http://xnainput.codeplex.com/

Cisco. (2013). Cisco Visual Networking Index – Global Mobile Traffic. http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.html

Cockburn, A., & McKenzie, B. (2000). An evaluation of cone trees. In *Sheffield, England: People and Computers XIV: British Computer Society Conference on Human Computer Interaction* (Vol. 2000, pp. 425–436). Retrieved from http://ir.canterbury.ac.nz/handle/10092/522

Cockburn, A., & McKenzie, B. (2001). What do Web users do? An empirical analysis of Web use. *International Journal of human-computer studies*, *54*(6), 903-922.

Cockburn, A. & McKenzie, B., (2002). Evaluating the effectiveness of spatial memory in 2D and 3D physical and virtual environments. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves*. ACM, pp. 203–210. Available at: http://dl.acm.org/citation.cfm?id=503413 [Accessed March 5, 2012].

Conway, J. (1970). The game of life. *Scientific American*,

Cowen, M. B. (2001). Information availability in 2D and 3D displays. *IEEE Computer Graphics and Applications*, *272*, 01.

Coors, V. (2003). 3D-GIS in networking environments. *Computers, Environment and Urban Systems*, *27*(4), 345-357.

Crawford, J.W. et al., (2005). Towards an evolutionary ecology of life in soil. *Trends in ecology & evolution*, 20(2), p.81-7. Available at: http://www.ncbi.nlm.nih.gov/pubmed/16701347.

Crooks, A., Castle, C., & Batty, M. (2008). Key challenges in agent-based modelling for geo-spatial simulation. *Computers, Environment and Urban Systems*, *32*(6), 417-430.

Crooks, Andrew, Hudson-Smith, Andrew and Dearden, Joel (2009). 'Agent Street: An Environment for Exploring Agent-Based Models in Second Life'.Journal of Artificial Societies and Social Simulation 12(4)10 <http://jasss.soc.surrey.ac.uk/12/4/10.html>.

Crooks, A. T., Hudson-Smith, A., & Patel, A. (2010). Building 3D Agent-Based Models for Urban Systems.

Cunningham, R. (2003). Developing 3D Graphics and Terrain Engines (coded in C ++ utilizing DirectX 8).

Dalgarno, B., & Lee, M. J. (2010). What are the learning affordances of 3-D virtual environments?. *British Journal of Educational Technology*, *41*(1), 10-32.

Dreborg, KH. (2004) "Scenarios and Structural Uncertainty", Department of Infrastructure, Royal Institute of Technology, Stockholm, Sweden.

Dockerty, T., Lovett, A., Sünnenberg, G., Appleton, K., & Parry, M. (2005). Visualising the potential impacts of climate change on rural landscapes. *Computers, Environment and Urban Systems*, *29*(3), 297–320. doi:10.1016/j.compenvurbsys.2004.05.004

Donal, P.F., Gree, R.E. & Heath, M.F., (2001). Agricultural intensification and the collapse of Europe's farmland bird populations. *Proceedings. Biological sciences / The Royal Society*, 268(1462), p.25-9. Available at: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1087596&tool=pmcentrez

Drone, S. (2007). Real-time particle systems on the GPU in dynamic environments. In *ACM SIGGRAPH 2007 courses* (pp. 80-96). ACM.

EcoChange. Project Objectives – EcoChange. (2011). Available at http://www.ecochange-project.eu/ecochange-project/project-objectives [Accessed 01/09/2011]

EcoChange. EcoChange Briefing Sheet. (2009). Available at http://www.ecochange-project.eu/PublicFiles/briefing-sheets/A1_LCC_LUC_final.pdf [Accessed 01/09/2011)

ESRI. (1985). ARC/INFO Users Manual Version 3. Environmental Systems Research Institute, Redlands, California.

ESRI. (1998). ESRI Shapefile Technical Description. An ESRI White Paper. [Online] Available at <http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>

ESRI. (2009). ESRI International User Conference, Available at http://events.esri.com/uc/QandA/index.cfm?fuseaction=answer&conferenceId=2a8e2713-1422-2418-7f20bb7c186b5b83&questionId=2441

Evans, T. P., & Kelley, H. (2004). Multi-scale analysis of a household level agent-based model of landcover change. *Journal of Environmental Management*, *72*(1), 57-72.

Ewing, R. (2001). Using a Visual Preference Survey in Transit Design. *Public Works Management & Policy*, *5*(4), 270–280

Facebook., (2007) http://blog.facebook.com/blog.php?post=2261927130

Falconer R, E, J. L. Bown, N. A. White, and J. W. Crawford. (2005). Biomass recycling and the origin of phenotype in fungal mycelia.

Falconer R, E, J. L. Bown, N. A. White, and J. W. Crawford. (2008). Modelling interactions in fungi.

Forbes.com, (2013). Tablet Web Traffic Greater Than Mobile Phones. Accessed 01/04/2013 Avaliable at http://www.forbes.com/sites/chuckjones/2013/03/11/tablet-web-traffic-greater-than-smartphones/

Freitas, C.M.D.S. et al., (2002). On evaluating information visualization techniques. *Proceedings of the Working Conference on Advanced Visual Interfaces - AVI '02*, p.373. Available at: http://portal.acm.org/citation.cfm?doid=1556262.1556326.

Fuller, R. (2000). Relationships between recent changes in lowland British agriculture and farmland populations: an overview. In Aebischer, N.J., Evans, A.D., Grice, P.V. & Vickery, J.A (eds) Ecology and Conservation of Lowland Farmland Birds: 5 – 16

Fuller, R., Gregory, R., Gibbons, D., Marchant, J., Wilson, J., Baillie, S. and Carter, N. (1995), Population Declines and Range Contractions among Lowland Farmland Birds in Britain. Conservation Biology, 9: 1425–1441.

Gamma, H. (1995). OO Design Patterns.

Gibson, R. B., & Hassan, S. (2005). Sustainability assessment: criteria and processes.

Gilford, J., Falconer, R. E., Wade, R., & Scott-Brown, K. C. (2013). 3D visualisation and artistic imagery to enhance interest in 'hidden environments'–new approaches to soil science. *European Journal of Engineering Education*, (ahead-of-print), 1-16.

Gobron, S., Çöltekin, A., Bonafos, H., & Thalmann, D. (2011). GPGPU computation and visualization of three-dimensional cellular automata. *The Visual Computer*, *27*(1), 67-81.

Goguen, J., & Harrell, D. F. (2005). 7 Information visualisation and semiotic morphisms. *Studies in Multidisciplinarity*. Retrieved from http://www.sciencedirect.com/science/article/pii/S1571083104800352

Goodchild, M., & Mark, D. M. (1987). The fractal nature of geographic phenomena. *Annals of the Association of American Geographers*, *77*(2), 265–278. Retrieved from http://www.tandfonline.com/doi/abs/10.1111/j

Google. (2008). Protocol Buffers Download. https://developers.google.com/protocol-buffers/

Government Office for Science, (2010), Foresight Land Use Futures Project. [Online] Available at <http://www.bis.gov.uk/foresight/our-work/projects/current-projects/land-use-futures> [Accessed 05 December 2011].

Grossardt, T., & Bailey, K. (2004). Community Visualization of a Light Rail Transit Oriented Development. *Program*, (April). Retrieved from http://worldcat.org/digitalarchive/content/CDM266401.cdmhost.com/CBT/p266401cdi/1208907611839/PB2008101504.pdf

GSC Research Society, (2001). Science.ca Profile: Roger Tomlinson. [Online] Available at <http://www.science.ca/scientists/scientistprofile.php?pID=420> [Accessed February 8th, 2012]

Guillem, E.E. & Barnes, a., (2013). Farmer perceptions of bird conservation and farming management at a catchment level. *Land Use Policy*, 31, pp.565–575. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0264837712001627 [Accessed March 13, 2013].

Guillem, E.E., (2012). "Human Behaviour and Ecosystem Services in Sustainable Farming Landscapes – An Agent-Based Model of Socio-Ecological Systems".

Guillem, E.E. et al., 2012. Refining perception-based farmer typologies with the analysis of past census data. *Journal of environmental management*, 110, pp.226–35. Available at: http://www.ncbi.nlm.nih.gov/pubmed/22805711 [Accessed March 13, 2013].

Guinard, D. & Trifa, V., (2010). Interacting with the soa-based internet of things: Discovery, query, selection, and on-demand provisioning of web services. *Services Computing, …*, 3(3), pp.223–235. Available at: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5416674 [Accessed April 25, 2013].

Han, T. D., Member, S., Abdelrahman, T. S., & Member, S. (2011). hi CUDA : High-Level GPGPU Programming, *22*(1), 78–90.

Hands, D. E., & Brown, R. D. (2002). Enhancing visual preference of ecological rehabilitation sites. *Landscape and Urban Planning*, *58*(October 2001). Retrieved from http://www.sciencedirect.com/science/article/pii/S0169204601002407

Hargreaves, S. (2010). Breaking changes in XNA Game Studio. Available at http://blogs.msdn.com/b/shawnhar/archive/2010/03/16/breaking-changes-in-xna-game-studio-4-0.aspx

Heckbert, S., Baynes, T., & Reeson, A. (2010). Agent-based modeling in ecological economics. Annals of the New York Academy of Sciences, 1185(1), 39-53.

Hoashi, K., & Hamawaki, S. (2009). Usability Evaluation of Visualization Interfaces for Content-Based Music Retrieval Systems. … *Information Retrieval …*, (Ismir), 207–212. Retrieved from http://ismir2009.ismir.net/proceedings/PS2-3.pdf

Holland, John H.; (2006). "Studying Complex Adaptive Systems." Journal of Systems Science and Complexity 19 (1)

Holman, I.P. et al., (2005). A Regional, Multi-Sectoral And Integrated Assessment Of The Impacts Of Climate And Socio-Economic Change In The Uk. Climatic Change, 71(1-2), pp.9–41. Available at: http://www.springerlink.com/index/10.1007/s10584-005-5927-y [Accessed July 5, 2011].

Hosbond, J. H., & Nielsen, P. A. (2008). Misfit or Misuse ? Lessons from Implementation of Scrum in Radical Product Innovation, 21–31.

Howley, P., Donoghue, C. O., & Hynes, S. (2012). Exploring public preferences for traditional farming landscapes. *Landscape and Urban Planning*, *104*(1), 66–74.

Hudson-Smith, A., (2009). Digital Urban: Crowds and delegates in 3D Max – Agent Based Modelling and Emergent Behaviour. [Online] Available at <http://www.digitalurban.org/2009/06/crowds-and-delegates-in-3d-max-agent.html>

Huk, T. (2006). Who benefits from learning with 3D models? The case of spatial ability. *Journal of Computer Assisted Learning*, *22*(6), 392-404.

Hull IV, B., & Revell, G. R. (1989). Issues in sampling landscapes for visual quality assessments. *Landscape and Urban Planning*, *17*(4), 323-330.

Inria, M. S., & Science, C. (2008). *Real-Time Realistic Rendering of Nature Scenes with Dynamic Lighting*.

Irani, P., & Ware, C. (2003). Diagramming information structures using 3D perceptual primitives. *ACM Transactions on Computer-Human Interaction (TOCHI)*, *10*(1), 1-19.

Irwin, E. G., & Geoghegan, J. (2001). Theory, data, methods: developing spatially explicit economic models of land use change. Agriculture, Ecosystems & Environment, 85(1), 7-24.

Isaacs, J. P., Falconer, R. E., Gilmour, D. J., & Blackwood, D. J. (2011). Enhancing urban sustainability using 3D visualisation. Proceedings of the ICE-Urban Design and Planning. 164 (3).

James, P., Fernando, T. & Hamilton, A., (2004). Enhancing the Decision-making Process in Urban Spatial Planning Using Advanced ICT Issues Paper This paper along with the accompanying technical report form the final report of this research Contents Chapter Abstract Acknowledgements Executive Summary Int. Sustainable Development,

Kazmierczak, E. (2001). A semiotic perspective on aesthetic preferences, visual literacy, and information design. *Information design journal*, 176–187. Retrieved from http://www.ingentaconnect.com/content/jbp/idj/2001/00000010/00000002/art00011

Keehner M, Montello D.R., Hegarty M. & Cohen C. (2004) Effects of interactivity and spatial ability on the comprehension of spatial relations in a 3D computer visualization. In *Proceedings of the 26th Annual Conference of the Cognitive Science Society* (edsK.Forbus, D.Gentner & T.Regier). Erlbaum, Mahwah, NJ, 1576pp.

Kolb, A., & Cuntz, N. (2005). Dynamic particle coupling for GPU-based fluid simulation. *Proc. Symposium on Simulation …*, (Section 2). Retrieved from http://www.cg.informatik.uni-siegen.de/data/Publications/2005/asim_fluid_simulation_paper.pdf

Kosara, R., Christopher, G., Carolina, N., David, H., & Ware, C. (2003). User Studies : Why , How , and When .

Koua, E. L., Maceachren, a., & Kraak, M. -J. (2006). Evaluating the usability of visualization methods in an exploratory geovisualization environment. *International Journal of Geographical Information Science*, *20*(4), 425–448.

Kraak, M. J. (2004). The role of the map in a Web-GIS environment. Journal of Geographical Systems, 6(2), 83-93.

LaGro, J. A. (2011). *Site analysis: A contextual approach to sustainable land planning and site design*. John Wiley & Sons.

Lambin, E. F., Rounsevell, M. D. A., & Geist, H. J. (2000). Are agricultural land-use models able to predict changes in land-use intensity?. Agriculture, Ecosystems & Environment, 82(1), 321-331.

Lambin, E.F. et al., (2001). The causes of land-use and land-cover change : moving beyond the myths. Global Environmental Change, 11, p.261-269.

Lambin, E.F., Geist, H.J. & Lepers, E., (2003). Dynamics of Land Use and L and Cover Change in Tropical Regions. Annual Review of Environment and Resources, 28(1), p.205-241. Available at: http://www.annualreviews.org/doi/abs/10.1146/annurev.energy.28.050302.105459

Lanir, I. (2012) “Charles Sanders Peirce: Symbolic, Iconic, and Indexical Signs

Latu, S., (2009). Sustainable Development: The Role of GIS and Visualisation. Information Systems p38.

Lindstrom, P., Koller, D., Ribarsky, W., Hodges, L. F., Faust, N., & Turner, G. A. (1996). Real-time, continuous level of detail rendering of height fields. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques* (pp. 109-118). ACM.

Liu, Z., Cai, S., Swan, J. E., Moorhead, R. J., Martin, J. P., & Jankun-Kelly, T. J. (2012). A 2D flow visualization user study using explicit flow synthesis and implicit task design. *IEEE transactions on visualization and computer graphics*, *18*(5), 783–96.

Macal, C. M., & North, M. J. (2010). Tutorial on agent-based modelling and simulation. *Journal of Simulation*, *4*(3), 151-162.

Malczewski, J. (2004). GIS-based land-use suitability analysis: a critical overview. *Progress in Planning*, *62*(1), 3–65. doi:10.1016/j.progress.2003.09.002

MapShaper. (2010). Mapshaper – Shaping maps. Available at http://www.mapshaper.org/

Marine Conservation Society. (2011). MCS Good Beach Guide. [Online] <Available at http://www.goodbeachguide.co.uk/search-results?beach_name=Lunan&wq_grade=0&postcode=&region=Scotland+%28north%29&distance=10&lifeguards=0&facility_level=0&dog_restrictions=0&page_size=5&op=Search&form_build_id=form-53f3966dab7c1c861cb326cf80c84c25&form_id=stbeach_search_long_form>

Massive. (2011). Massive Software – Simulation Life. [Online] Available at <http://www.massivesoftware.com/index.html> [Accessed 30 November 2011].

McDonnel, B., & Elmqvist, N. (2009). Towards utilizing GPUs in information visualization: a model and implementation of image-space operations. *IEEE transactions on visualization and computer graphics*, *15*(6), 1105–12.

McCreadie, C ., Isaacs, J. P., Stojanovic, V., Falconer, R. E., Gilmour, D. J., & Blackwood, D. (2012). The SAVE Concept: Sustainability Assessment and Enhancement Through Novel Visualisation

Meert, H., van Huylenbroeck, G., Vernimmen, T., Bourgeois, M., and van Hecke, E., 2005. Farm household survival strategies and diversification on marginal farms. Journal of Rural Studies 21, 81-97.

Merrick, K.E., and M.L. Maher. (2007). “Motivated reinforcement learning for adaptive characters in open-ended simulation games.” In Proceedings of the international conference on Advances in computer entertainment technology, 127–134. ACM, 2007. http://dl.acm.org/citation.cfm?id=1255073.

Microsoft (2010). Acquiring high-resolution time-stamps. [Online] Available at <http://support.microsoft.com/kb/172338>

Mikropoulos, T. a., & Natsis, A. (2011). Educational virtual environments: A ten-year review of empirical research (1999–2009). *Computers & Education*, *56*(3), 769–780

Miller, C. C. (2006). A beast in the field: The Google Maps mashup as GIS/2.Cartographica: The International Journal for Geographic Information and Geovisualization, 41(3), 187-199.

Miller, D., Vogt, N., Nijnik, M., Brondizio, E., Fiorini, S. (2009). Integrating Analytical and Participatory Techniques for Planning the Sustainable Use of Land Resources and Landscapes. In “Planning Support Systems Best Practice and New Methods”. GeoJournal Library Vol.95

Misgav, A. (2000). Visual preference of the public for vegetation groups in Israel. *Landscape and Urban Planning*, *48*(3-4), 143–159

Mitchell, M., (2009). Complexity A guided tour. Oxford. Oxford University Press.

Möller, T., Haines, E., & Hoffman, N. (2008). Real-time rendering. AK Peters Ltd.

Muller, J., Lorenz, M. & Geller, F., (2010). Assessment of communication protocols in the EPC network-replacing textual SOAP and XML with binary google protocol buffers encoding. Available at: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5646586 [Accessed April 29, 2013].

Murray-Rust, D., Dendoncker, N., Dawson, T., Acosta-Michlik, L., Karali, E., Guillem, E., and Rounsevell, M., (2011). Conceptualising the analysis of socio- ecological systems through ecosystem services and agent based modelling. Journal of Land Use Science 6, 83-99.

Murray-Rust, D., (2013). Aporia | Mo-Speph. Aporia Diagram. Available at http://www.mo-seph.com/academic/aporia

Nakicenovic, N.; Alcamo, J.; Davis, G.; de Vries, B.; Fenhann, J.; Gaffin, S.; Gregory, K.; Grübler, A.; Yong Jung, T.; Kram, T.; Lebre La Rovere, E.; Michaelis, L.; Mori, S.; Morita, T.; Pepper, W.; Pitcher, H.; Price, L.; Riahi, K.; Roehrl, A.; Rogner, H-H.; Sankovski, A.; Schlesinger, M.; Shukla, P.; Smith, S.; Swart, R.; van Rooijen, S.; Victor, N., and Zhou, D., (2000). Special report on emissions scenarios. Intergovernmental panel on climate change. Cambridge University Press, Cambridge, UK.

Nelessen, A. C. (1994). Visions for a new American dream. Chicago: American Planning Association.

Nilsson, Fredrik, and Vince Darley. (2006). “On complex adaptive systems and agent-based modelling for improving decision-making in manufacturing and logistics settings: Experiences from a packaging company.” International Journal of Operations & Production Management 26, no. 12 (2006): 1351-1373. http://www.emeraldinsight.com/10.1108/01443570610710588.

North, M., Macal, C. and Campbell, P. (2005), “Oh behave! Agent-based behavioral representations in problem solving environments”, Future Generation Computer Systems, Vol. 21 No. 1, pp. 1192-8.

North, Chris. (2006) “Toward measuring visualization insight.” IEEE computer graphics and applications 26, no. 3 (2006): 6-9. http://www.ncbi.nlm.nih.gov/pubmed/16711210.

Oracle. 2006 “Inside the hotspot VM Clocks”. [Online] Available at <https://blogs.oracle.com/dholmes/entry/inside_the_hotspot_vm_clocks>

Owens., J. M. Houston, D. Luebke, S. Green, J. Stone and J. Phillips, (2008). GPU computing, Proc. IEEE 96(5) (2008), 879– 899

Parker, D. (2009), “Introduction to agent-based and cellular automaton models of land use change”. [Online]. Available at http://virgo.unive.it/changes/docs/ws_venice/03a1_parker.pdf.

Paar, P. (2006). Landscape visualizations: Applications and requirements of 3D visualization software for environmental planning. *Computers, environment and urban systems*, *30*, 815–839.

Parry, M., & Carter, T. (1998). *Climate impact and adaptation assessment: a guide to the IPCC approach*. Earthscan Publications Ltd.

Parry, M. et al., (2004). Effects of climate change on global food production under SRES emissions and socio-economic scenarios. Global Environmental Change, 14(1), p.53-67. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0959378003000827

Parsons, R. (1991). The potential influences of environmental perception on human health. *Journal of Environmental Psychology*, *11*(1), 1–23.

Pavon, J, M Arroyo, S Hassan, and C Sansores. (2008). “Agent-based modelling and simulation for the analysis of social patterns.” Pattern Recognition Letters 29, no. 8. Available at <http://linkinghub.elsevier.com/retrieve/pii/S0167865507002152>

Peirce, C. *Collected Writings* (8 Vols.). (1931-58). Ed. Charles Hartshorne, Paul Weiss & Arthur W Burks. Cambridge, MA: Harvard University Press.

Pettit, C., Cartwright, W., & Berry, M. (2006). Geographical visualization: A participatory planning support tool for imagining landscape futures. Applied GIS, 2(3), 22-1.

Pielke, Roger A Sr., Land Use and Climate Change. (2005) Science 9 vol.310 no5754 pp`625-1626.

Plaisant, C. (2004). The challenge of information visualization evaluation. In *Proceedings of the working conference on Advanced visual interfaces* (pp. 109-116). ACM.

Polhill, J. G., Gotts, N. M., & Law, A. N. R. (2001). Imitative versus nonimitative strategies in a land-use simulation. Cybernetics & Systems, 32(1-2), 285-307.

Potel, M. (1996). MVP: Model-View-Presenter the taligent programming model for c++ and java. *Taligent Inc*.

Prensky, M. (2001). True believers: Digital game-based learning in the military. Digital game-based learning.

Rajan, K. S., & Shibasaki, R. (2000). A GIS based integrated land use/cover change model to study human-land interactions. International Archives of Photogrammetry and Remote Sensing, 33(B7/3; PART 7), 1212-1219.

Reynolds, C. W. (1987) Flocks, Herds, and Schools: A Distributed Behavioral Model, in Computer Graphics, 21(4) (SIGGRAPH '87 Conference Proceedings) pages 25-34.

Reynolds, C., 2010. Individual Based Models. [Online] Available at <http://www.red3d.com/cwr/ibm.html> [Accessed 29 Novemeber 2011].

Richmond, P., & Romano, D. (2008). Agent based gpu, a real-time 3d simulation and interactive visualisation framework for massive agent based modelling on the gpu. In Proceedings International Workshop on Supervisualisation.

Richmond, P., Coakley, S., & Romano, D. M. (2009). A high performance agent based modelling framework on graphics card hardware with CUDA. In Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 2 (pp. 1125-1126). International Foundation for Autonomous Agents and Multiagent Systems.

Road Traffic Act. 1988

Robertson, G.G., Mackinlay, J.D. & Card, S.K., (1991). Cone trees: animated 3D visualizations of hierarchical information. In Proc. of the SIGCHI Conference on Human factors in computing systems: Reaching through technology. p. 194. Available at: http://ir.canterbury.ac.nz/handle/10092/522 [Accessed January 31, 2012].

Roth, M. (2006). Validating the use of Internet survey techniques in visual landscape assessment—An empirical study from Germany. *Landscape and Urban Planning*, *78*(3), 179–192.

Rounsevell, M. D. A., Annetts, J. E., Audsley, E., Mayr, T., & Reginster, I. (2003). Modelling the spatial distribution of agricultural land use at the regional scale. *Agriculture, Ecosystems & Environment*, *95*(2), 465-479.

Rounsevell, M, F Ewert, I Reginster, R Leemans, and T Carter. (2005). “Future scenarios of European agricultural land use. Projecting changes in cropland and grassland.” Agriculture, Ecosystems & Environment 107, no. 2-3 (May 20, 2005): 117-135. http://linkinghub.elsevier.com/retrieve/pii/S0167880904003615.

RSPB. (2011). The RSPB:Skylark. [Online] <Available at http://www.rspb.org.uk/wildlife/birdguide/name/s/skylark/index.aspx> Accessed 12th January 2012.

RSPB. (2010). The Skylark Advisory Sheet (Scotland). [Online] <Available at http://www.rspb.org.uk/Images/Skylark_tcm9-133209.pdf> Accessed 12th January 2012.

Saraiya, P., North, C. & Duca, K., (2005). An insight-based methodology for evaluating bioinformatics visualizations. IEEE transactions on visualization and computer graphics, 11(4), pp.443–56. Available at: http://www.ncbi.nlm.nih.gov/pubmed/16138554.

Sasaki, Y and Paul Box. (2003). "Agent-Based Verification of von Thunen's Location Theory". Journal of Artificial Societies and Social Simulation (vol 6, no 2). (2003)

Scardamalia, M., & Bereiter, C. (2002). Knowledge Building. In Encyclopedia of Education (2nd ed.).

Scholtz, J. (2004). Usability evaluation. *National Institute of Standards and Technology*.

Scottish Government. (2011). Profitable and Sustainable Agriculture – Plants. [Online]. <Available at http://www.scotland.gov.uk/Resource/Doc/175356/0120937.pdf>

Scottish Government. (2012). Getting the best from our land. A land use strategy for Scotland. [Online] <Availabe at http://scotland.gov.uk/Resource/0039/00395990.pdf>

Sebrechts, M. M., Cugini, J. V., Laskowski, S. J., Vasilakis, J., & Miller, M. S. (1999). Visualization of search results: a comparative evaluation of text, 2D, and 3D interfaces. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 3–10). ACM. Retrieved from http://dl.acm.org/citation.cfm?id=312634

Seipel, S., & Carvalho, L. (2012). Solving Combined Geospatial Tasks Using 2D and 3D Bar Charts. *2012 16th International Conference on Information Visualisation*, 157–163.

SEPA. (2009). SEPA – Monitoring and classification. [Online] <Available at http://www.sepa.org.uk/water/monitoring_and_classification.aspx> Accessed 12th January 2012

Shanks, G. G., & Darke, P. (1998). Understanding Data Quality and Data Warehousing: A Semiotic Approach. In *IQ* (pp. 292-309).

Shiohama, R., Washizaki, H., Kuboaki, S., Sakamoto, K., & Fukazawa, Y. (2012). Estimate of the Appropriate Iteration Length in Agile Development by Conducting Simulation. In *Agile Conference (AGILE), 2012* (pp. 41-50). IEEE.

Southern, A. (2010) Designing Multi-objective whole landscape. *Climate Policy*.

Steinitz, C. (1990). Toward a sustainable landscape with high visual preference and high ecological integrity: the loop road in Acadia National Park, U.S.A. *Landscape and Urban Planning*, *19*(3), 213–250.

St John, M., Cowen, M. B., Smallman, H. S., & Oonk, H. M. (2001). The use of 2D and 3D displays for shape-understanding versus relative-position tasks. *Human factors*, *43*(1), 79–98. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/11482314

Stone, R. J. (2005). Serious gaming—virtual reality's saviour. In Proceedings of Virtual Systems and MultiMedia annual conference, VSMM (pp. 773-786).

Stonedahl, S. H., Harvey, J. W., Wörman, A., Salehin, M., & Packman, A. I. (2010). A multiscale model for integrating hyporheic exchange from ripples to meanders. *Water resources research*, *46*(12).

Sutcliffe, A. G., Ennis, M., & Hu, J. (2000). Evaluating the effectiveness of visual user interfaces for information retrieval. *International Journal of Human-Computer Studies*, *53*(5), 741–763. doi:10.1006/ijhc.2000.0416

Taylor, P., Parish, D.M.B. & Sotherton, N.W., (2010). Game crops and threatened farmland songbirds in Scotland : a step towards halting population declines. Bird Study, (January 2012), p.37-41.

Terdiman, D. (2011). New technology revs up Pixars 'Cars2'. Online. Available at http://news.cnet.com/8301-13772_3-20068109-52/new-technology-revs-up-pixars-cars-2/ [Accessed 3rd September 2011]

Thalmann, N., Foni, A. E., Papagiannakis, G., & Cadi-Yazli, N. (2007). Real Time Animation and Illumination in Ancient Roman Sites. *IJVR*,*6*(1), 11-24.

Ting, S., Tan, T., West, G., Squelch, A., & Foster, J. (2011, November). Quantitative assessment of 2D versus 3D visualisation modalities. In *Visual Communications and Image Processing (VCIP), 2011 IEEE* (pp. 1-4). IEEE.

Tory, M., & Kirkpatrick, A. (2006). Visualization task performance with 2D, 3D, and combination displays. Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1541995

Trafton, J., Kirschenbaum, S. S., Tsui, T. L., Miyamoto, R. T., Ballas, J. A., & Raymond, P. D. (2000). Turning pictures into numbers: extracting and generating information from complex visualizations. *International Journal of Human-Computer Studies*, *53*(5), 827-850.

Tufte, Edward R. "Envisioning information." *Optometry & Vision Science* 68.4 (1991): 322-324.

Turner, M. and Budgen, D. and Brereton, P. (2003) 'Turning software into a service.', Computer., 36 (10). pp. 38-44.

Ulrich, R. S. (1986). Human responses to vegetation and landscapes. *Landscape and Urban Planning*, *13*, 29–44. doi:10.1016/0169-2046(86)90005-8

USGS. (2011). United States Geology Survey SRTM Data. Available at http://srtm.usgs.gov/index.php

Van Lammeren, R. et al., (2010). Affective appraisal of 3D land use visualization. Computers, Environment and Urban Systems, 34(6), pp.465–475. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0198971510000657 [Accessed November 16, 2012].

Veldkamp, A., & Verburg, P. H. (2004). Modelling land use change and environmental impact. *Journal of Environmental Management*, *72*(1), 1-3.

Venners, B. (2007). Test-Driven Development: A Conversation with Martin Fowler, Part V. … *Development Rhythms: Harmonizing Agile* …. Retrieved from http://onlinelibrary.wiley.com/doi/10.1002/9780470192672.ch9/summary

Vinten, A.J.A., Stutter, M.I., Sample, J., Dunn, S., Birkel, C., Potts, J., MacDonald, J., Napier, F., Jeffrey,W., Christian, C., (2010). How effective is the implementation of controls on diffuse pollution under the water framework directive in Scotland? Answers and questions from the Lunan diffuse pollution monitored catchment project. In: Proceedings of the 14th International

Conference, IWA Diffuse Pollution Specialist Group: Diffuse Pollution and Eutrophication, Quebec, Canada, 12e17 September 2010, pp. 330E343.

Vinton, A., (2011). Assessing diffuse pollution and land management impacts on water quality in the Lunan Water, using event-based monitoring [Online] <Available at http://www.knowledgescotland.org/briefings.php?id=212> [Accessed 18 Aug 2011]

Vizzari, G., Pizzi, G., & da Silva, F. S. C. (2008). A framework for execution and 3D visualization of situated cellular agent based crowd simulations. *Proceedings of the 2008 ACM symposium on Applied computing - SAC '08*, 18. doi:10.1145/1363686.1363692

Von Neumann, J. (1966). Theory of self-reproducing automata.

Wang, J., Rubin, N., Wu, H., & Yalamanchili, S. (2013). Accelerating Simulation of Agent-Based Models on Heterogeneous Architectures.

Watson, B. & Friedman, A., (2000). Using naming time to evaluate quality predictors for model simplification. Proceedings of the SIGCHI, 2(1), p.113-120. Available at: http://dl.acm.org/citation.cfm?id=332040.332414

Watson, B., Friedman, A. & McGaffey, A., (2001). Measuring and predicting visual fidelity. In Proceedings of the 28th annual conference on Computer graphics and interactive techniques. ACM, pp. 213–220. Available at: http://dl.acm.org/citation.cfm?id=383259.383283 [Accessed January 26, 2012].

Wergles, N., & Muhar, A. (2009). The role of computer visualization in the communication of urban design—A comparison of viewer responses to visualizations versus on-site visits. Landscape and urban planning, 91(4), 171-182.

Whitten, A. (2004). Making security usable, (May). Retrieved from http://reports-archive.adm.cs.cmu.edu/anon/anon/usr/ftp/usr0/ftp/2004/CMU-CS-04-135.pdf

Whitmarsh, L. (2009). Participation of Experts and Non-Experts in a Sustainability Assessment of Mobility, *250*(June), 232–250

Wiegand, P. (2003). School students' understanding of choropleth maps: Evidence from collaborative mapmaking using GIS. Journal of Geography,102(6), 234-242.

Wilensky, U. (2003), NetLogo Traffic Grid Model, Available at http://ccl.northwestern.edu/netlogo/models/TrafficGrid [Accessed 30 November , 2011]

Wilkie, D., (2009)Agent Based Intersection Traffic Simulation. [online].Available at http://www.cs.unc.edu/~wilkie/PBMReport.pdf [Accessed 21st November 2011].

Wijk, J. Van, (2005). The value of visualization. Visualization, 2005. VIS 05. IEEE, pp.79–86. Available at: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1532781 [Accessed May 28, 2013].

Wiss, U., Carr, D. & Jonsson, H., (1998). Evaluating three-dimensional information visualization designs: A case study of three designs. In Information Visualization, 1998. Proceedings. 1998 IEEE Conference on. IEEE, pp. 137–144. Available at: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=694211 [Accessed January 23, 2012].

Wissen, U., Schroth, O., Lange, E., & Schmid, W. A. (2008). Approaches to integrating indicators into 3D landscape visualisations and their benefits for participative planning situations. *Journal of Environmental Management*, *89*(3), 184-196.

Wolfram, S. (1998). Cellular automata as models of complexity. Nonlinear Physics for Beginners: Fractals, Chaos, Solitons, Pattern Formation, Cellular Automata, Complex Systems, 311, 197.

World Wide Web Consortium (W3C). (2004). Available at http://www.w3.org/TR/ws-arch/#id2260892 [Accessed 12 November 2012]

Yourdon, E., & Constantine, L. L. (1979). Structured design: fundamentals of a discipline of computer program and systems design. Prentice-Hall, Inc.

Zimmermann, B. B. (1997, October). Applying Tufte's principles of information design to creating effective Web sites. In *Proceedings of the 15th annual international conference on Computer documentation* (pp. 309-317). ACM.

Appendices

Appendix A: Software Licences

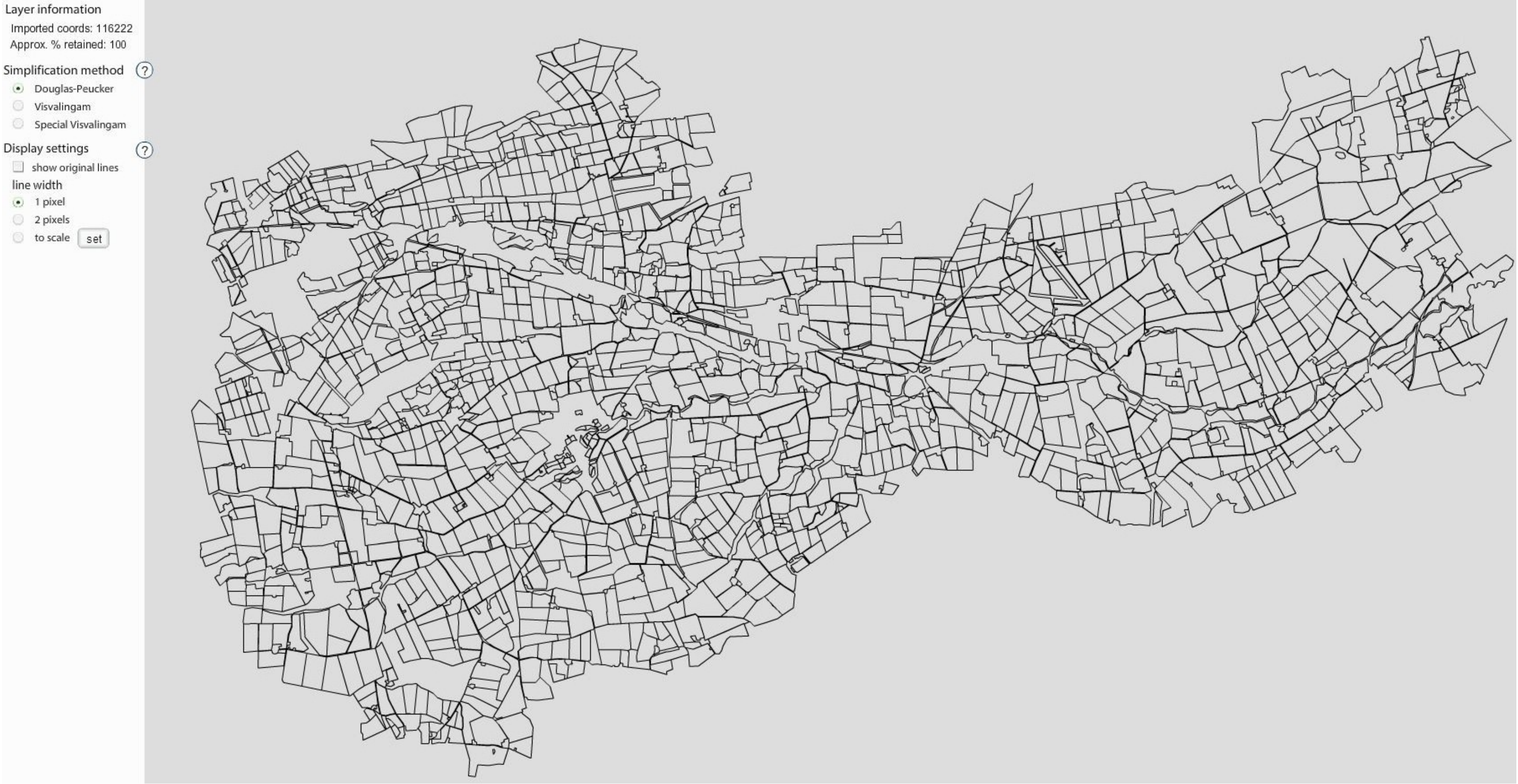
GPLv3 Licence

MIT Licence

New BSD Licence (Ammendments made to XNAInput)

Apache Licence

Appendix B: Shapefile polygon reduction



Layer information

Imported coords: 116222

Approx. % retained: 100

Simplification method

● Douglas-Peucker

○ Visvalingam

○ Special Visvalingam

Display settings

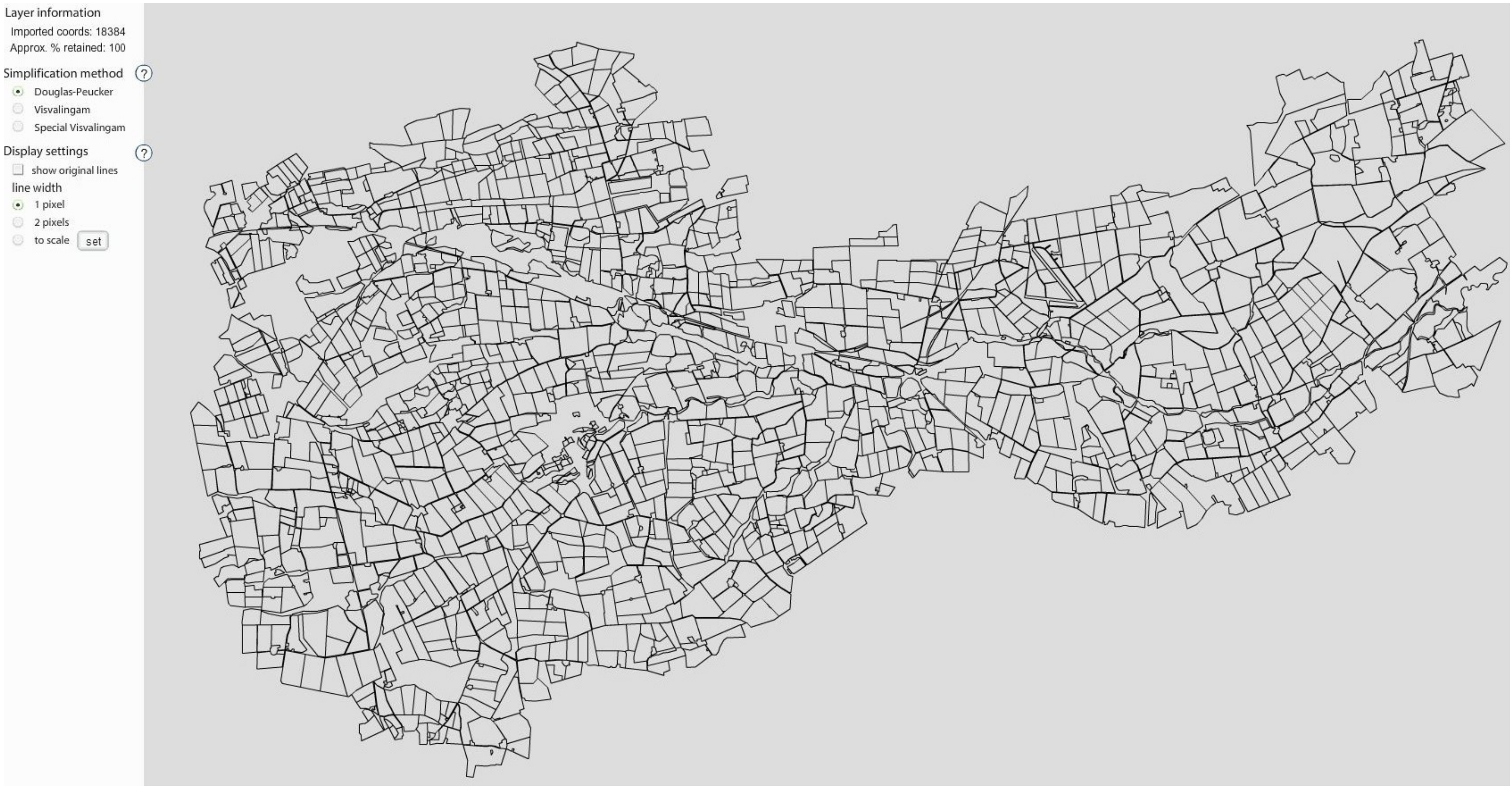
□ show original lines

line width

● 1 pixel

○ 2 pixels

○ to scale



Layer information

Imported coords: 18384

Approx. % retained: 100

Simplification method

- ● Douglas-Peucker
- ○ Visvalingam
- ○ Special Visvalingam

Display settings

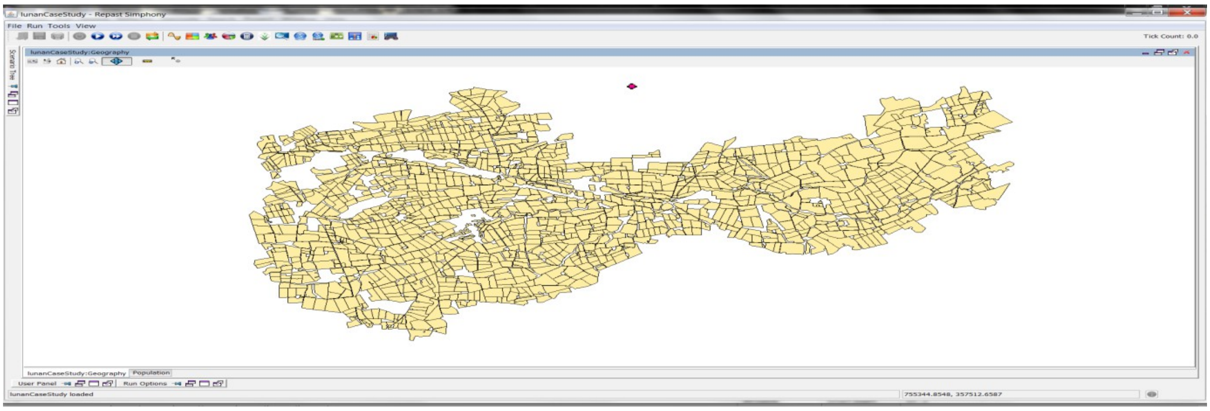
- □ show original lines

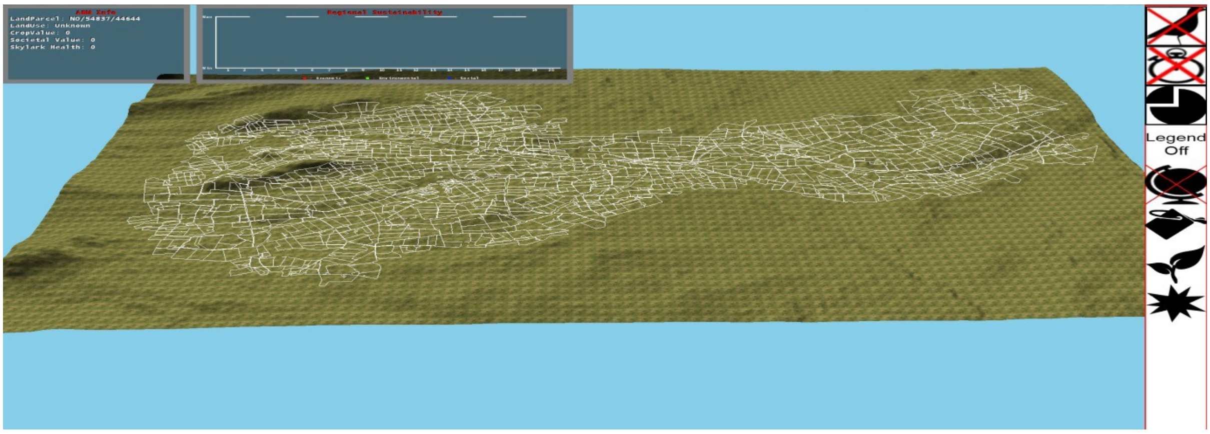
line width

- ● 1 pixel
- ○ 2 pixels
- ○ to scale

Appendix C: Visual Preference Survey Image Sets

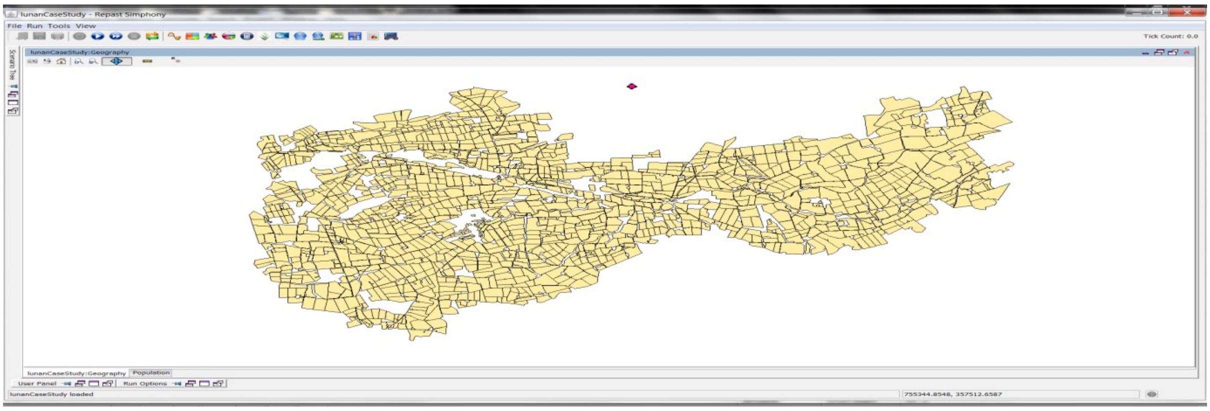
VPS 1





LandParcel: NO/54837/44644
LandUse: Unknown
CropValue: 0
Societal Value: 0
Skylark Health: 0

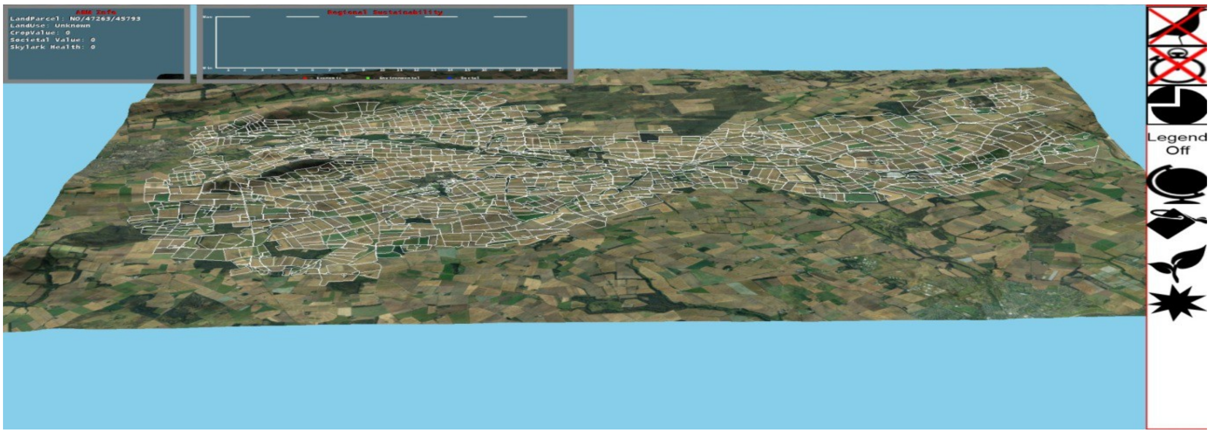
VPS 2



Tick Count: 0.0

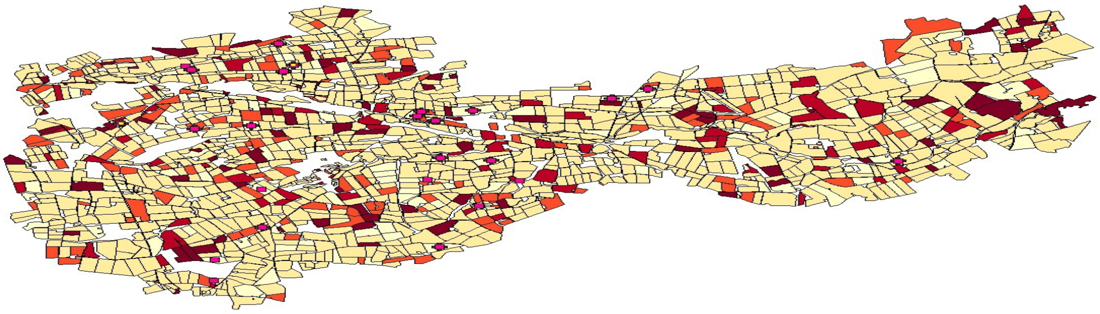
lunanCaseStudy loaded

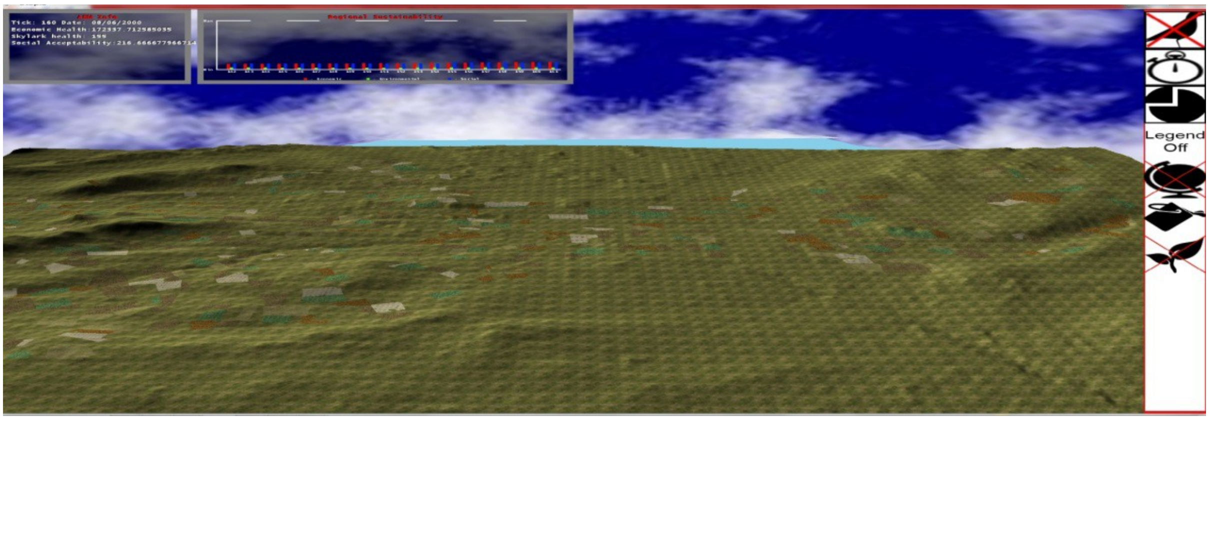
755344.8548, 357512.6587



LandParcel: NO/47263/45793
LandUse: Unknown
CropValue: 0
Societal Value: 0
Skylark Health: 0

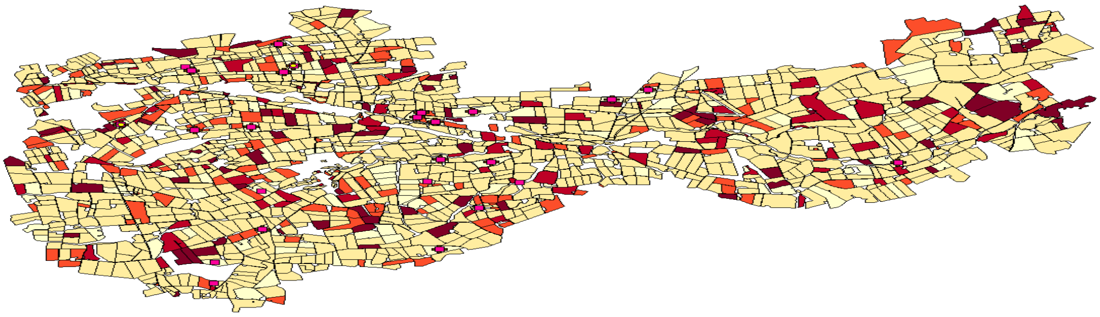
VPS 3

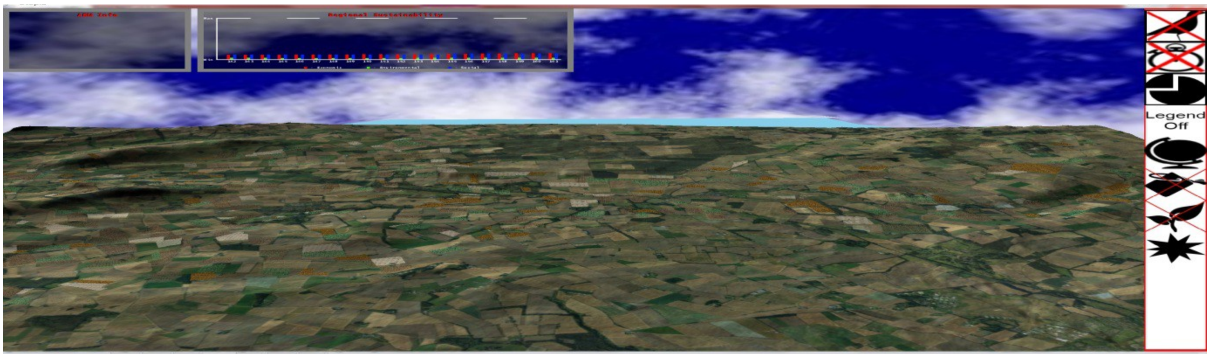




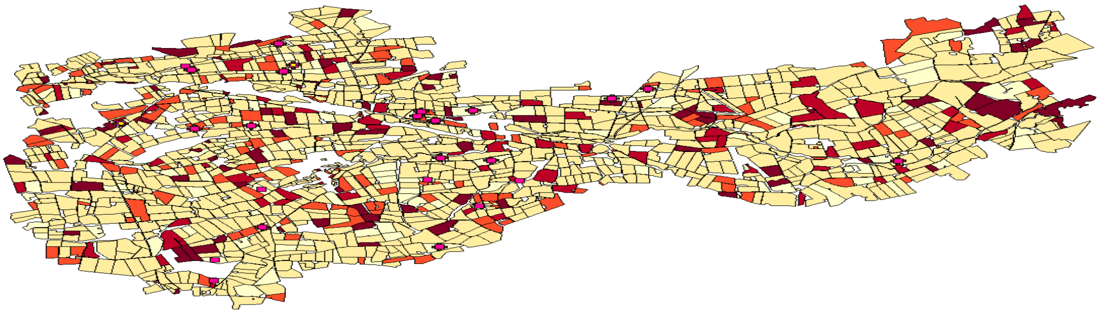
Tick: 160 Date: 08/06/2000
Economic Health:172337.712585035
Skylark health: 155
Social Acceptability:216.666677966714

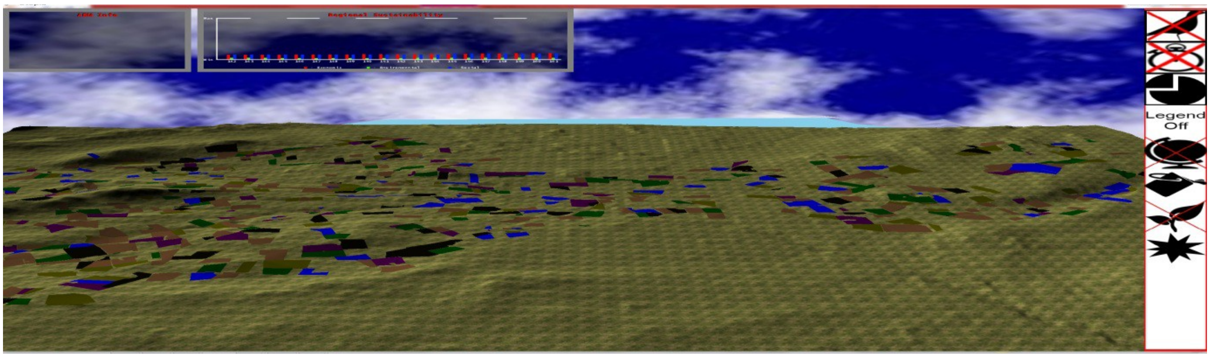
VPS 4



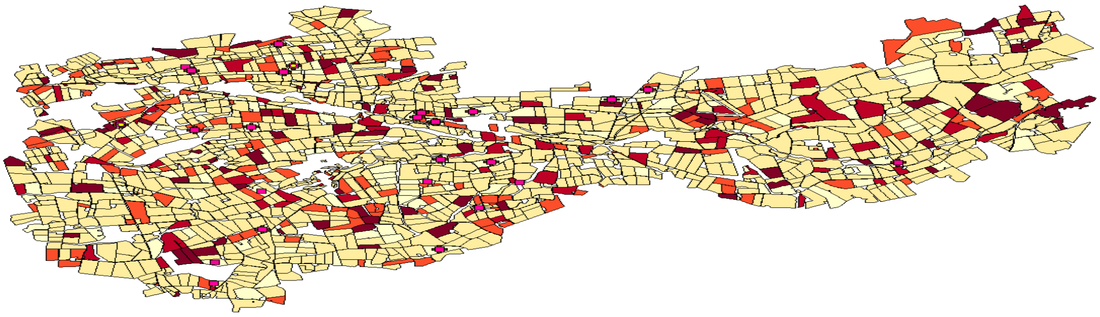

ABM Info
Regional Sustainability
Max
Min
Economic
Environmental
Social
Legend
Off

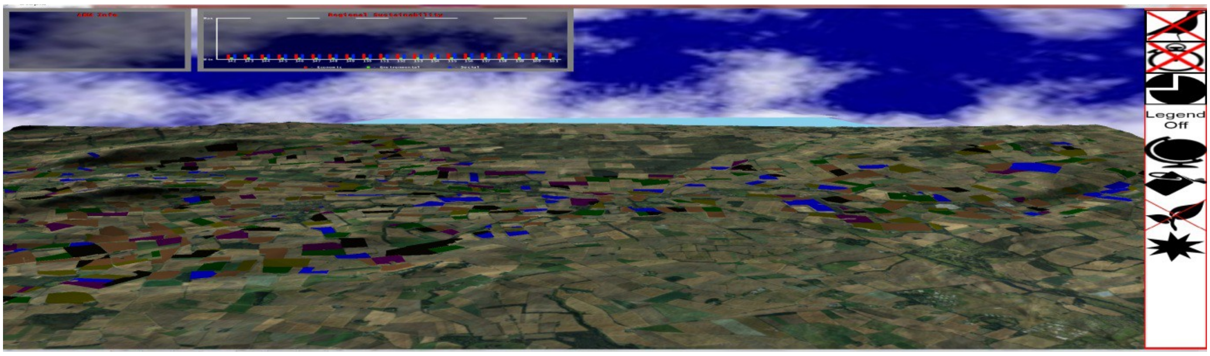
VPS 5



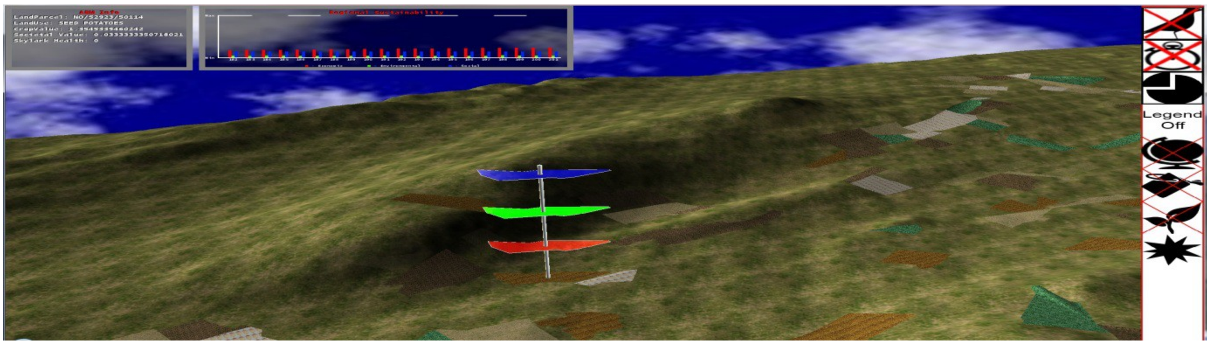

ABM Info
Regional Sustainability
Max
Min
Economic
Environmental
Social
Legend
Off

VPS 6



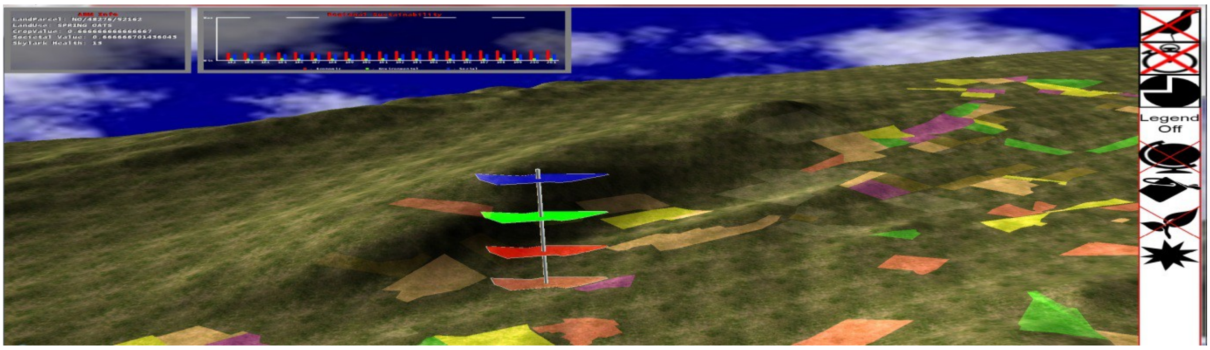

ABM Info
Regional Sustainability
Max
Min
Economic
Environmental
Social
Legend Off

VPS 7



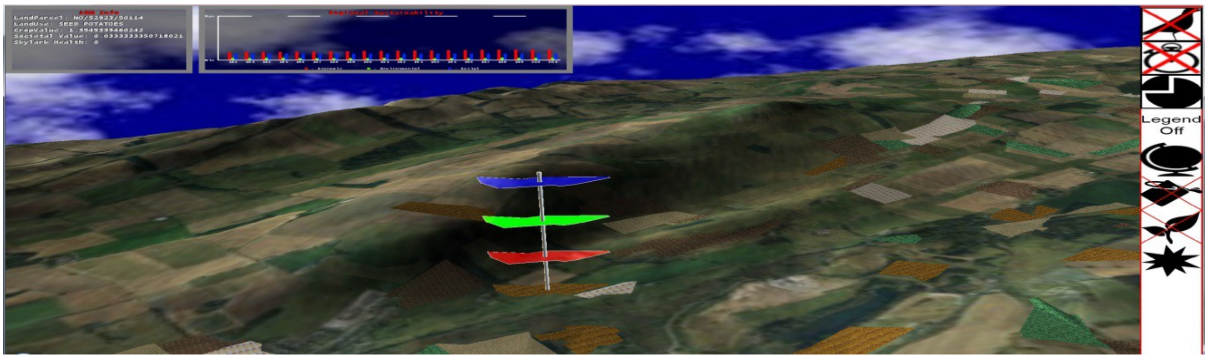
ABM Info
LandParcel: NO/52923/50114
LandUse: SEED POTATOES
CropValue: 1.35499994460242
Societal Value: 0.0333333350718021
Skylark Health: 0

VPS 8



ABM Info
LandParcel: NO/48276/92162
LandUse: SPRING OATS
CropValue: 0.6666666666666667
Societal Value: 0.666666701436043
Skylark Health: 13

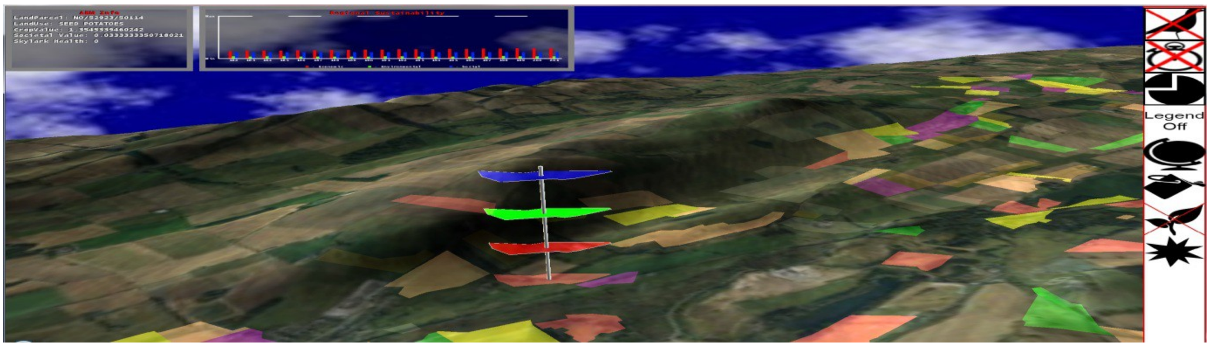
VPS 9



ABM Info

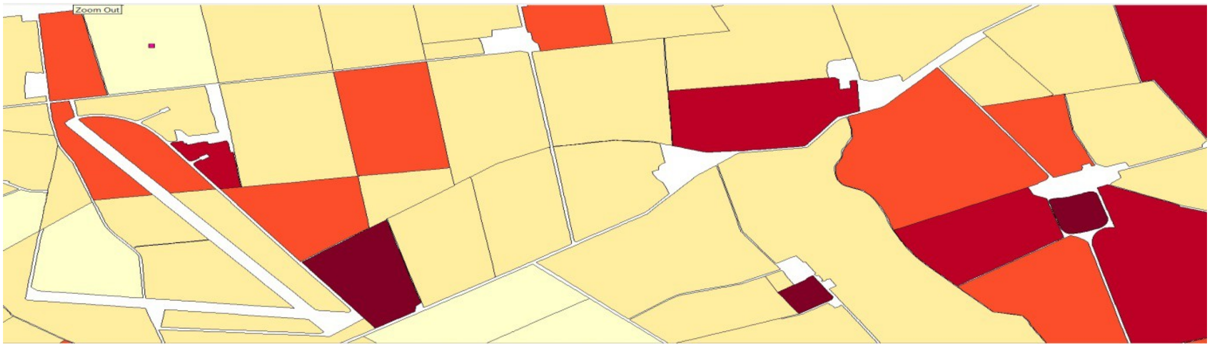
LandParcel: NO/52923/50114
LandUse: SEED POTATOES
CropValue: 1.3545339460242
Societal Value: 0.0333333350718021
Skylark Health: 0

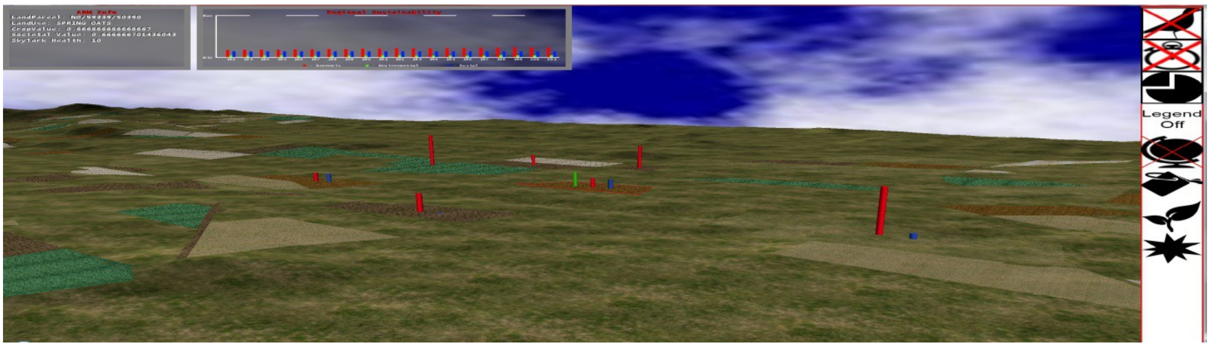
VPS 10



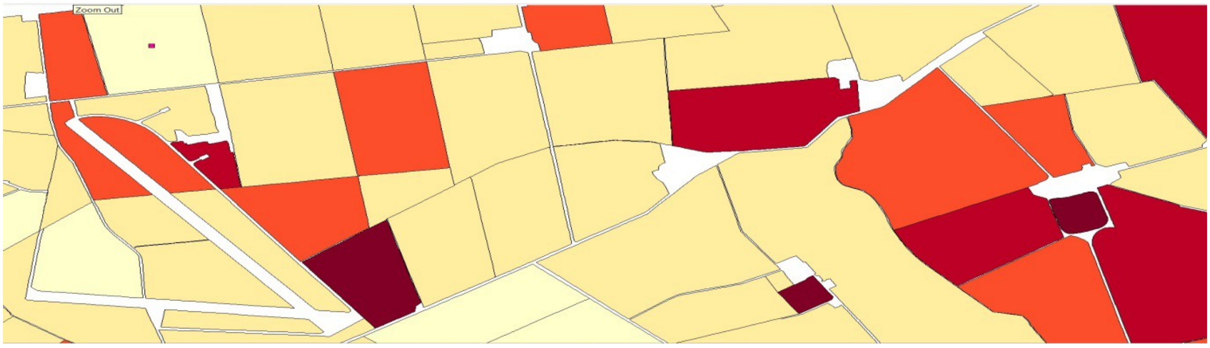
ABM Info
LandParcel: NO/52923/50114
LandUse: SEED POTATOES
CropValue: 1.3545339460242
Societal Value: 0.0333333350718021
Skylark Health: 0

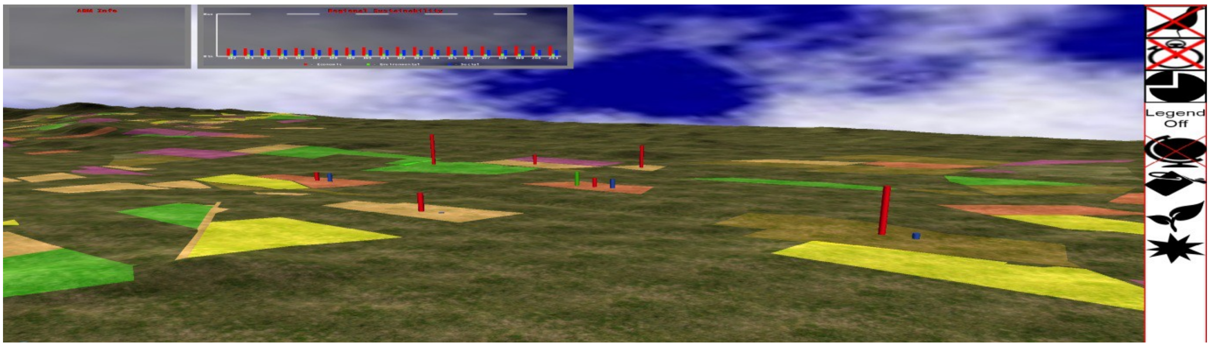
VPS 11



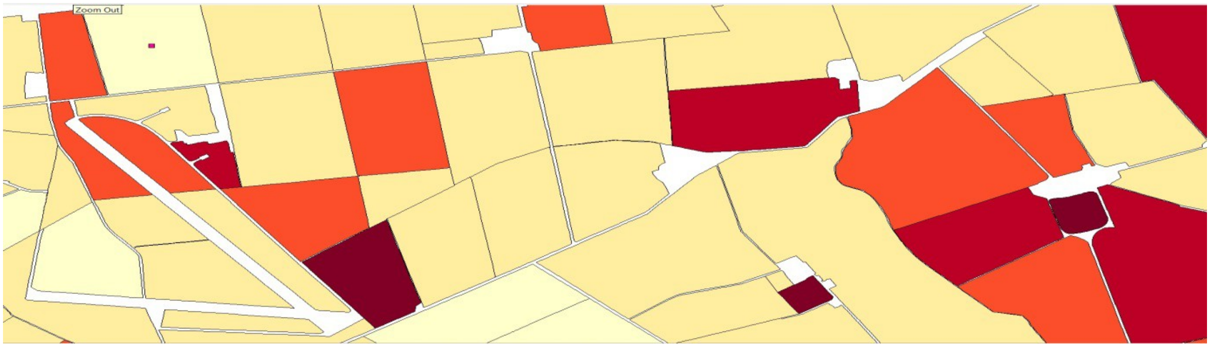

ABM Info
LandParcel: NO/59839/50390
LandUse: SPRING OATS
CropValue: 0.6666666666666667
Societal Value: 0.666666701436043
Skylark Health: 10

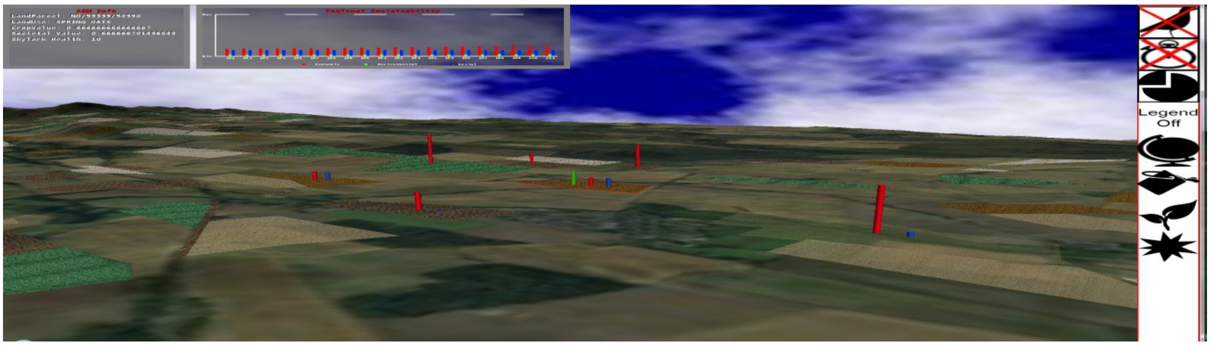
VPS 12


Zoom Out

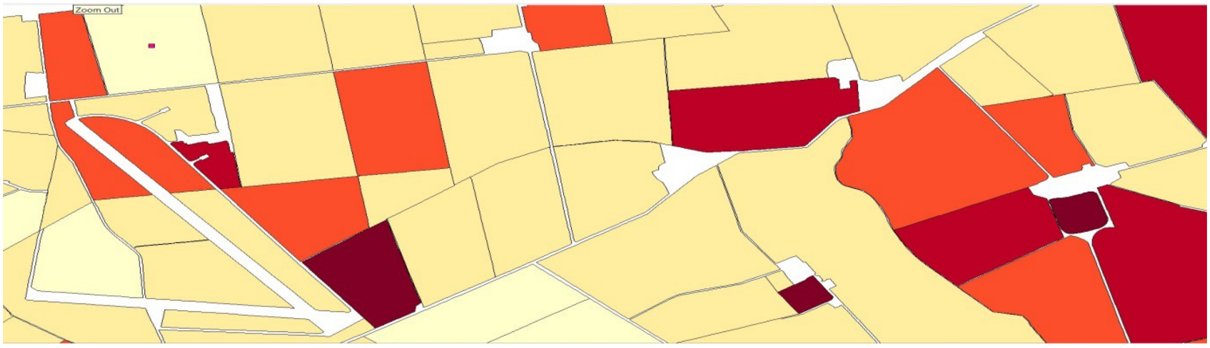

ABM Info
Regional Sustainability
Max
Min
Economic
Environmental
Social
Legend
Off

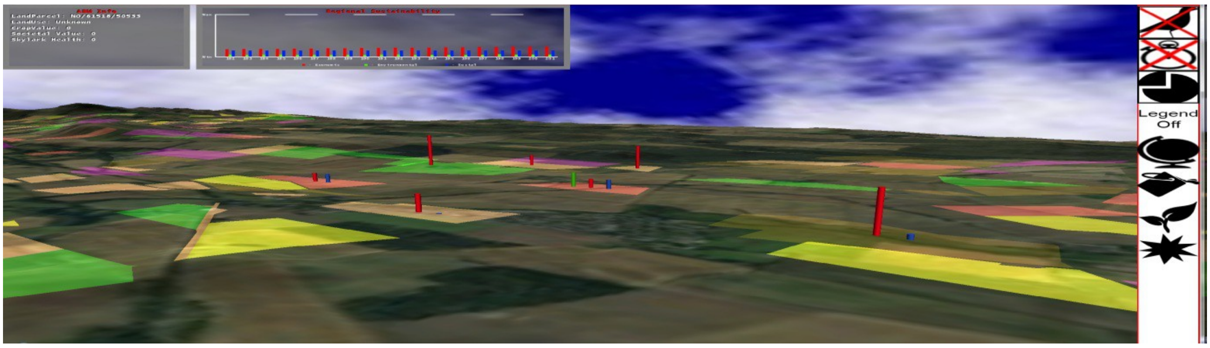
VPS 13




ABM Info
LandParcel: NO/59339/50390
LandUse: SPRING OATS
CropValue: 0.6666666666666667
Societal Value: 0.666666701446044
Skylark Health: 10

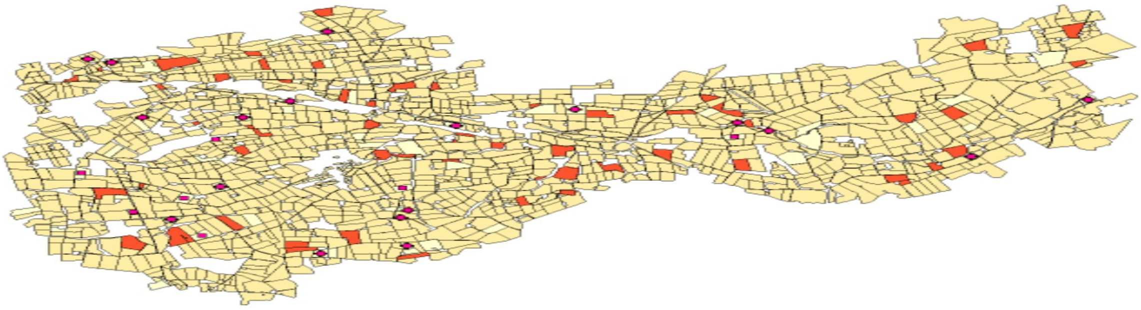
VPS 14

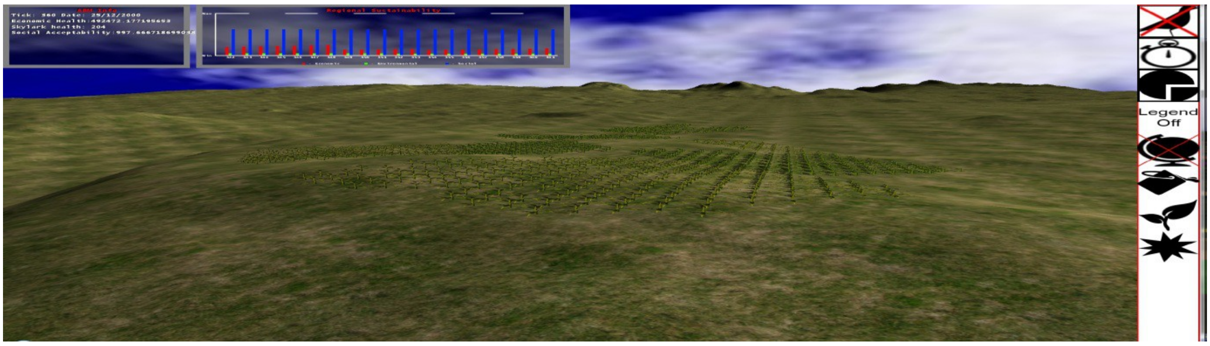




ABM Info
LandParcel: NO/61518/50533
LandUse: Unknown
CropValue: 0
Societal Value: 0
Skylark Health: 0

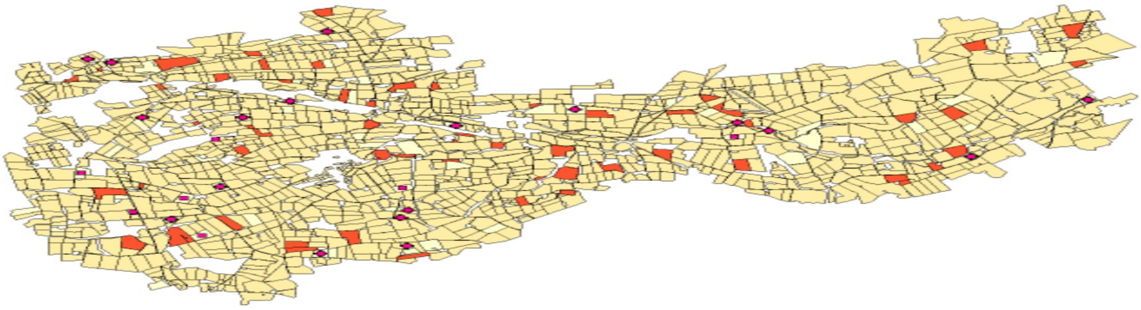
VPS 15

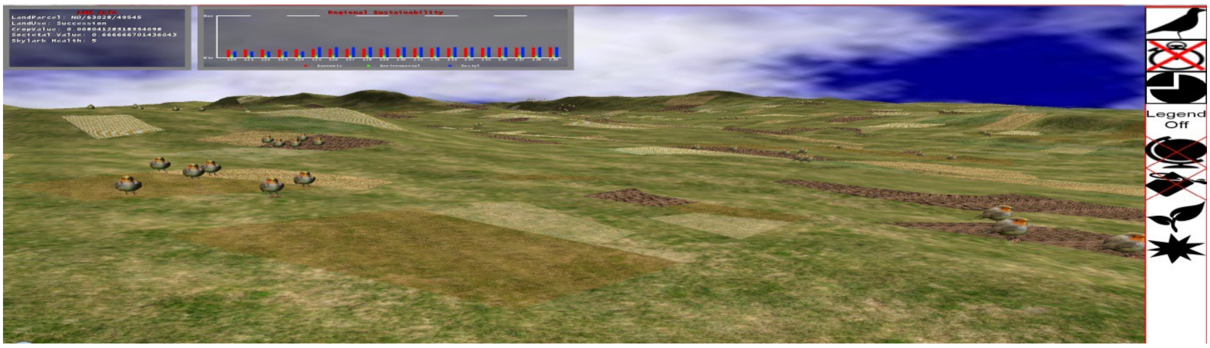




ABM Info
Tick: 360 Date: 25/12/2000
Economic Health:492472.177195653
Skylark health: 204
Social Acceptability:997.666718699038

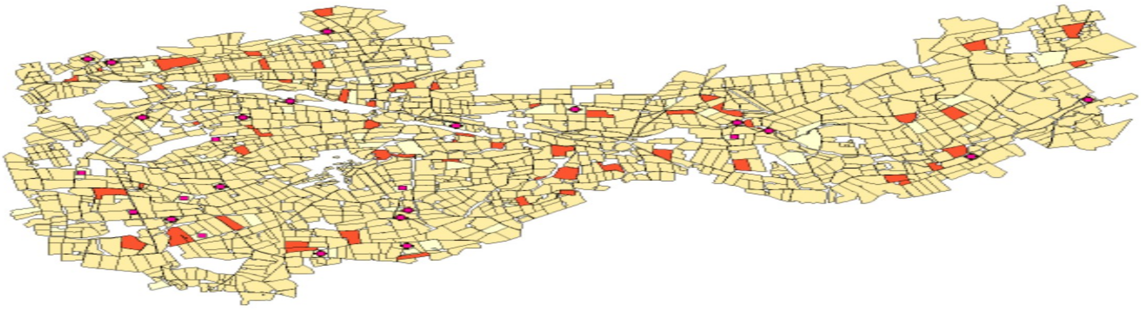
VPS 16

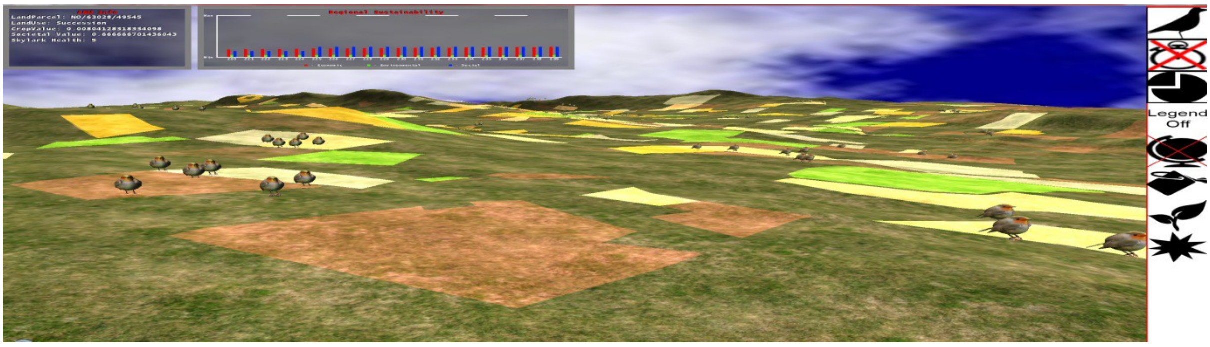




ABM Info
LandParcel: NO/63020/49545
LandUse: Succession
CropValue: 0.00804128318334098
Societal Value: 0.666666701436043
Skylark Health: 5

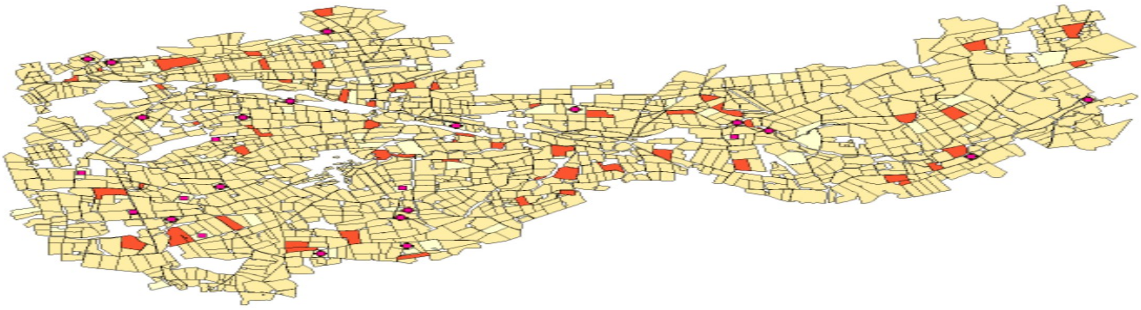
VPS 17

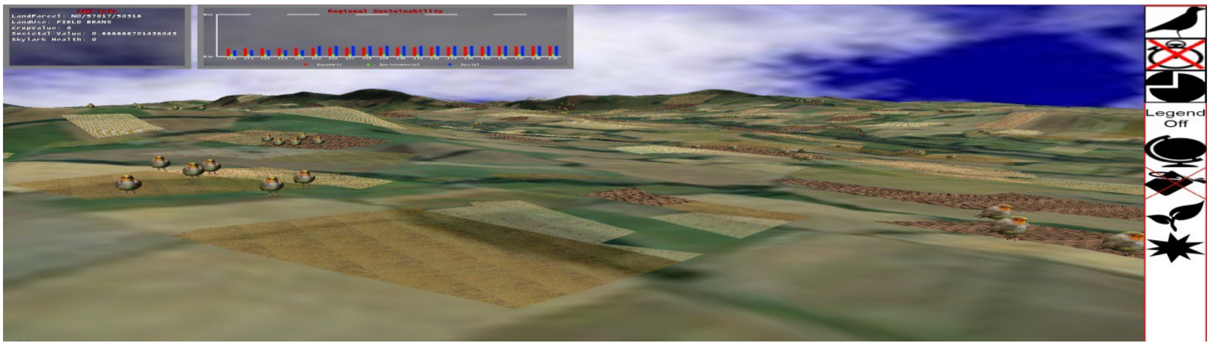




ABM Info
LandParcel: NO/63028/49545
LandUse: Succession
CropValue: 0.00804128518554098
Societal Value: 0.666666701436043
Skylark Health: 5

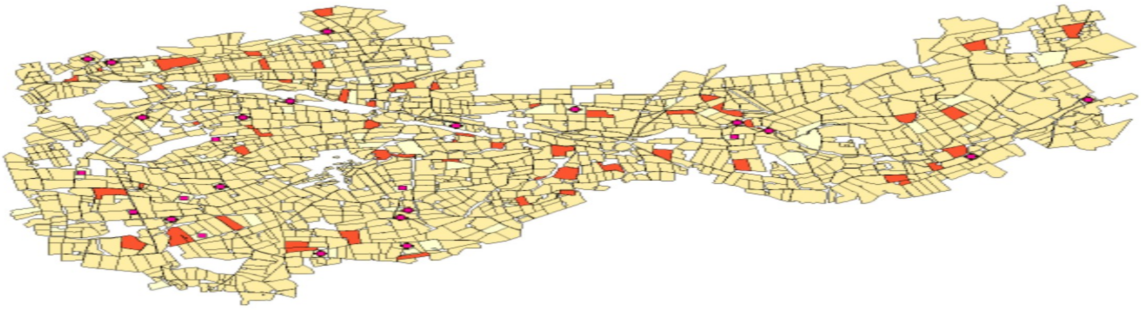
VPS 18

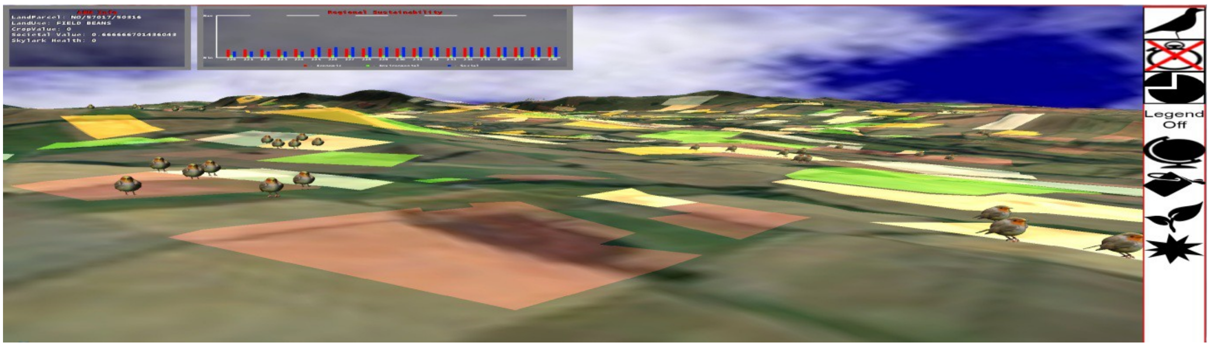




ABM Info
LandParcel: NO/57017/50316
LandUse: FIELD BEANS
CropValue: 0
Societal Value: 0.666666701436043
Skylark Health: 0

VPS 19





ABM Info

LandParcel: NO/57017/50316
LandUse: FIELD BEANS
CropValue: 0
Societal Value: 0.666666701436043
Skylark Health: 0

Appendix D – Task Based Testing Question Sheet

You will be shown a video of a particular task being carried out within both R.S.V.T (the 3D front-end) or RepastS (the 2D front-end). The variety of different tasks that you will see will be discussed before you see the images and you can expect to find things like

- Identifying land parcels
- Identifying harvests
- Calculating regional sustainability
- Calculating individual land parcel sustainability
- Calculating skylark populations

You will then be asked to write down what task you believe you viewed and how effective the video was in communicating the task.

Name……………………………………………………………………………………

Video Set 1: Land Parcel Selection

Video 1

I found the video to be ………………………………………………………..

(1)	**(2)**	**(3)**	**(4)**	**(5)**
Not helpful at all	**Slightly unhelpful**	**Neither helpful or unhelpful**	**Slightly helpful**	**Very helpful**

More information: Please indicate which Land Parcel was selected as well as any additional information about the land parcel such as crop type, crop value and skylark health within the land parcel.

…………………………………………………………………………………………

……………………………………………………………………

Video 2

I found the video to be ………………………………………………………..

(1)	**(2)**	**(3)**	**(4)**	**(5)**
Not helpful at all	**Slightly unhelpful**	**Neither helpful or unhelpful**	**Slightly helpful**	**Very helpful**

More information: Please indicate which Land Parcel was selected as well as any additional information about the land parcel such as crop type, crop value and skylark health within the land parcel.

…………………………………………………………………………………………

……………………………………………………………………

Video 3

I found the video to be …………………………………………………………..

(1)	(2)	(3)	(4)	(5)
Not helpful at all	Slightly unhelpful	Neither helpful or unhelpful	Slightly helpful	Very helpful

More information: Please indicate which Land Parcel was selected as well as any additional information about the land parcel such as crop type, crop value and skylark health within the land parcel.

……………………………………………………………………………………………

…………………………………………………………………

Video 4

I found the video to be …………………………………………………………..

(1)	(2)	(3)	(4)	(5)
Not helpful at all	Slightly unhelpful	Neither helpful or unhelpful	Slightly helpful	Very helpful

More information: Please indicate which Land Parcel was selected as well as any additional information about the land parcel such as crop type, crop value and skylark health within the land parcel.

……………………………………………………………………………………………

…………………………………………………………………

Video Set 2: Sowing and Harvesting

Video 1

I found the video to be ……………………………………………………..

(1)	**(2)**	**(3)**	**(4)**	**(5)**
Not helpful at all	**Slightly unhelpful**	**Neither helpful or unhelpful**	**Slightly helpful**	**Very helpful**

More information: Please indicate which land use is most abundant within the video.

……………………………………………………………………………………………

……………………………………………………………

Video 2

I found the video to be ……………………………………………………..

(1)	**(2)**	**(3)**	**(4)**	**(5)**
Not helpful at all	**Slightly unhelpful**	**Neither helpful or unhelpful**	**Slightly helpful**	**Very helpful**

More information: Please indicate which land use is most abundant within the video.

……………………………………………………………………………………………

……………………………………………………………

Video 3

I found the video to be ……………………………………………………..

(1)	**(2)**	**(3)**	**(4)**	**(5)**
Not helpful at all	**Slightly unhelpful**	**Neither helpful or unhelpful**	**Slightly helpful**	**Very helpful**

More information: Please indicate which land use is most abundant within the video.

………………………………………………………………………………………

…………………………………………………………………

Video 4

I found the video to be ……………………………………………………..

(1)	**(2)**	**(3)**	**(4)**	**(5)**
Not helpful at all	**Slightly unhelpful**	**Neither helpful or unhelpful**	**Slightly helpful**	**Very helpful**

More information: Please indicate which land use is most abundant within the video.

………………………………………………………………………………………

…………………………………………………………………

Video Set 3: Individual Sustainability (Exploded)

Video 1

I found the video to be …………………………………………………………..

(1)	(2)	(3)	(4)	(5)
Not helpful at all	**Slightly unhelpful**	**Neither helpful or unhelpful**	**Slightly helpful**	**Very helpful**

More information: Please indicate which sustainability indicator is the highest and which sustainability indicator is the lowest.

………………………………………………………………………………………………

…………………………………………………………………

Video 2

I found the video to be …………………………………………………………..

(1)	(2)	(3)	(4)	(5)
Not helpful at all	**Slightly unhelpful**	**Neither helpful or unhelpful**	**Slightly helpful**	**Very helpful**

More information: Please indicate which sustainability indicator is the highest and which sustainability indicator is the lowest.

………………………………………………………………………………………………

…………………………………………………………………

Video 3

I found the video to be …………………………………………………..

(1)	**(2)**	**(3)**	**(4)**	**(5)**
Not helpful at all	**Slightly unhelpful**	**Neither helpful or unhelpful**	**Slightly helpful**	**Very helpful**

More information: Please indicate which sustainability indicator is the highest and which sustainability indicator is the lowest.

……………………………………………………………………………………

…………………………………………………………

……………………………………………………………………………………

…………………………………………………………

Video 4

I found the video to be …………………………………………………..

(1)	**(2)**	**(3)**	**(4)**	**(5)**
Not helpful at all	**Slightly unhelpful**	**Neither helpful or unhelpful**	**Slightly helpful**	**Very helpful**

More information: Please indicate which sustainability indicator is the highest and which sustainability indicator is the lowest.

……………………………………………………………………………………

…………………………………………………………

Video Set 4: Individual Sustainability (Pillars)

Video 1

I found the video to be ……………………………………………………..

(1)	(2)	(3)	(4)	(5)
Not helpful at all	**Slightly unhelpful**	**Neither helpful or unhelpful**	**Slightly helpful**	**Very helpful**

More information: Please indicate which sustainability indicator is the highest and which sustainability indicator is the lowest.

……………………………………………………………………………………………

……………………………………………………………

Video 2

I found the video to be ……………………………………………………..

(1)	(2)	(3)	(4)	(5)
Not helpful at all	**Slightly unhelpful**	**Neither helpful or unhelpful**	**Slightly helpful**	**Very helpful**

More information: Please indicate which sustainability indicator is the highest and which sustainability indicator is the lowest.

……………………………………………………………………………………………

……………………………………………………………

Video 3

I found the video to be ………………………………………………………..

(1)	(2)	(3)	(4)	**(5)**
Not helpful at all	**Slightly unhelpful**	**Neither helpful or unhelpful**	**Slightly helpful**	**Very helpful**

More information: Please indicate which sustainability indicator is the highest and which sustainability indicator is the lowest.

…………………………………………………………………………………………

…………………………………………………………….

Video 4

I found the video to be ………………………………………………………..

(1)	(2)	(3)	(4)	**(5)**
Not helpful at all	**Slightly unhelpful**	**Neither helpful or unhelpful**	**Slightly helpful**	**Very helpful**

More information: Please indicate which sustainability indicator is the highest and which sustainability indicator is the lowest.

…………………………………………………………………………………………

…………………………………………………………….