

Chapter 13

Visualising Large-Scale Behaviours: Presenting 4D Data in Archaeology

Philip Murgatroyd

Abstract Agent-based modelling (ABM) is an excellent technique for investigating certain types of complex behaviours. Archaeologists have started to explore the possibilities of ABM as a means of simulating the processes that shape past human societies, processes that can be hard to reconstruct by other means. However, the complex nature of these models and the four dimensional (4D) nature of the data they produce can be problematic to present via traditional archaeological means of publication such as journals and monographs. This article reviews the problems faced when presenting 4D data via 2D media and presents some ways in which the Medieval Warfare on the Grid project attempts to solve those problems.

Keywords Byzantine · Military logistics · 3D modelling · Visualisation · Agent-based modelling · Archaeology

13.1 Introduction

Agent-based modelling (ABM) and archaeology seem like the perfect complement to each other. ABM uses spatial data to create models of complex systems that change over time based on the interactions of discrete entities simulated within, the “agents” of the title. Archaeology also looks at changes over space and time, wrought by the interactions of discrete entities, usually humans, with each other and their environment. ABM as a computer modelling technique has grown in popularity during the past 30 years as increased processing power and code manageability in object-oriented programming languages have rendered it viable

P. Murgatroyd (✉)

IBM Visual and Spatial Technologies Centre, The University of Birmingham,
Edgbaston B15 2TT, UK
e-mail: P.S.Murgatroyd@bham.ac.uk

across a whole range of disciplines but even before the widespread adoption of ABM within the wider scientific community, some archaeologists had noticed the overlap (Doran 1970). Whereas archaeologists were largely restricted to inferring processes from patterns, simulation was an excellent tool to enable the inference of patterns from processes (Kohler and Varien 2012, p. 9). Yet if the form of an ABM mirrors that of some of the problems that archaeology is tasked with investigating, the format of the data produced can be a very poor fit with the publication methods commonly employed in archaeological projects.

Despite recent responses to the technology of the World Wide Web, archaeology publications remain primarily paper-based. Journals and books still form the bulk of outputs from archaeological projects. These are static, sequential, 2D. The increase in use of portable document format (PDF) files does nothing to change this, being more associated with improvements in the efficiency of distribution. In contrast, an ABM is commonly not just a means to a conclusion, the process of arriving there itself can be useful (Ch'ng et al. [forthcoming](#)). ABMs commonly require multiple runs with varying parameters to quantify the interactions modelled within the system. This result is not only in a set of 4D data, 3D spatial data modified over time, but multiple sets of 4D data to be compared in a variety of ways, often with a bewildering number of combinations. It is dynamic, fungible and often very, very large in its raw state. Merely explaining the workings of an ABM can be a verbose and unwieldy process, showing the results satisfactorily sometimes seems in on impossible.

Another complicating factor in presenting an ABM involves the intended audience, a group that may include people with very different interests and levels of expertise, but will also include the creator of the ABM themselves. A researcher using an ABM will need some way of viewing the running of the model during operation in order to verify its behaviour. An extensively cited early ABM, the Schelling model (Schelling 1971), was recommended by its creator to be resolved on a checkerboard with actual counters rather than on the computers of the time precisely because their visualisation capabilities could not satisfactorily show the process of the model at work (Hegselmann 2012).

Once the simulations have been run, the results need to be shown to others. ABM researchers are accustomed to producing and interpreting graphs, 2D still images and animations of basic shapes in sparse landscapes. General archaeologists may not necessarily have that background and may need more visual cues to understand the processes at work within the model. Interested members of the general public may have even less experience of interpreting the outputs of models and may need more context and possibly different forms of publication altogether. With archaeological ABMs it is not just the results that are of interest, the passage of the agents through the process can give us an insight into past lives. The path of an individual agent through the simulation can act as the basis for visualising, via words, images or animations, an individual person's experience of the system

being modelled. These “simulated biographies” are an attractive tool when dealing with a non-specialist audience who are often interested in what the lives of individuals were like. Just as a model needs to be created with only appropriate processes simulated, so the output should focus on the message that is being communicated and be disseminated via a useful medium.

13.2 Visualisation of ABM in Archaeology

Publication within the simulation community operates under some of the same restrictions as it does in archaeology. Traditional paper publishing still forms a large part of academic publishing within computer science and so even here complex 4D behaviours are presented as 2D stills and/or aggregated statistics and graphs (e.g. Cioffi-Revilla 2010). These outputs are well suited to presenting behaviours that can be reduced to formulaic expressions. A defining characteristic of complex behaviours however is that they can't be reduced to a single, simple formula (Holland 1998, p. 14) and as such their presentation presents a challenge for traditional publishing. Visualisation within the ABM community tends towards simplicity and clarity, often consisting of 2D images or, where possible, animations. An audience of researchers experienced in simulation will be comfortable with the interpretation of these outputs. Few archaeologists have experience of the process and results of ABMs and how the practise of simulation can be translated into archaeological data and therefore the approach to presenting results may need to be revised. Archaeology also frequently has to present its results to a wider audience, including interested members of the general public with no specific experience of the outputs of either archaeological research or ABMs.

Axelrod (2007, p. 96), in an article designed to suggest how simulation can be moved forward from the early stages he sees it in, presents three reasons why disseminating the results of simulations can be problematic.

- Simulation results are highly dependent on the detail of the model so, if the model is not described in detail, the results may be open to misinterpretation or misunderstanding.
- Although results can usually be summarised via statistical methods, the results often also include a narrative description of the run of a model. This commonly takes up a lot of space.
- Simulation results are often presented to an interdisciplinary audience, necessitating detailed explanation of parts of the model that some of the intended audience will find unnecessary.

The problems highlighted by Axelrod all apply to ABM within an archaeological framework and as yet no method of dissemination has presented a satisfactory solution to them. Thankfully the rise in popularity of ABMs has coincided with the rise of the World Wide Web and many ABMs have been made available to download, in some cases also including animations showing their results.

Nevertheless, there is a lack of a standard method of describing the workings of ABMs with the same degree that mathematical notation can be used to describe top-down systems. This is largely due to the complexity of multi-agent simulations and the relative youth of the field.

Within archaeology, the history of ABM visualisation contains comparatively few examples to examine. After an initial spurt of enthusiasm in the 1970s, when computing power was rare, arcane and expensive enough to not be of much use, it wasn't until the 1990s that archaeological ABMs became more widespread (Costopoulos et al. 2010). Early efforts tended to present their results in statistical format (Wobst 1974), comparing the data from each run of the simulation via aggregated indicators of performance. More recently, models have failed to take advantage of increases in computing power and visualisation technology.

One of the largest and most complex archaeological ABMs, a simulation of Bronze Age Mesopotamian settlement system dynamics (Wilkinson et al. 2007), relies on presenting the summarised results of many runs of the simulation as a way of addressing specific research questions regarding the system's resilience to various stresses. The source code of their ABM is unavailable and highlights a problem with ABMs modular nature, their ENKIMDU framework relies on data from systems from other organisations which may not be openly available. Access to source code from non-open sources may become a potentially damaging factor limiting the spread of successful models in the future.

The Village Ecodynamics Project (VEP) (Kohler and Varien 2012) is probably the largest, most complex archaeological ABM implemented so far. It consists of over 17,000 lines of code and draws in evidence from pollen cores, tree-ring dating, hydrology and a series of pre-existing modelling work dating as far back as the 1980s. The project examines the processes of population fluctuation in the Pueblo societies of south-western Colorado. It has produced a prolific range of publications across a variety of traditional journals (e.g. Kohler et al. 2007, 2008; Varien et al. 2007) along with a monograph dealing with the first phase of the project (Kohler and Varien 2012). The traditional nature of these publications, mainly produced in paper format albeit sometimes with an electronic copy, restricts the presentation of data. Illustrations are also limited and are usually in the form of graphs. There seems to be no easy way of extracting map-based outputs from the model, much less animations that require change over time. The VEP has published its models in their most primitive form, making the source code available for other researchers to simulate the same experiments, and run different experiments with the same model but different parameters or to expand the model in whichever way they see fit. This allows additional visualisation methods to be added by others and, although examples of published models being reused to verify results are rare (Janssen 2009 being a notable exception), publishing the model itself also acts as documentation of its workings.

MASON (Luke et al. 2005) is an attempt to provide an ultra-lightweight multi-agent simulation toolkit. Produced at George Mason University, it was written by computer scientists with considerable input from social scientists and provides a Java infrastructure and set of tools that can be used to build ABMs. It has the

advantage of separating the model from the visualisation so that different forms of visualisation, both 2D and 3D, can be attached to a simulation while it is running and can be unplugged and changed at will.

13.3 The Medieval Warfare on the Grid Project

The Medieval Warfare on the Grid project (MWGrid) was funded by a joint AHRC-EPSRC-JISC e-science grant and attempts to use ABM to investigate the complex relationships involved with medieval military logistics (Craenen et al. 2010). Due to its scale and intended audience, the project needs to visualise the running and results of its ABM in both traditional and non-traditional ways. Using the march of the Byzantine army across Anatolia to the Battle of Manzikert in AD 1071 as its case study, it serves as a framework to quantify the supply implications of moving over different routes and with different forms of organisation. This case study was chosen as it was an excellent example of a historical question needing new sources of data. The Byzantine army marched over 1,000 km across Anatolia to face the Seljuk Turks at the battle of Manzikert and yet key data is not recorded in the historical sources available to us. The size of the army is unknown and although some aspects of the route it travelled are recorded, the logistical organisation needed to ensure it reached its destination is ignored in first-hand accounts. Anecdotal stories of events on the route take precedence over quantitative data regarding army speed, supplies required or number of participants and the hypotheses generated by historians has suffered from a lack of testability (Haldon 2006).

The MWGrid ABM allows the simulation of a day's march of a force of any given size, consisting of both cavalry and infantry using several different types of organisation. This allows quantitative analysis of how the size, composition and organisation of the force affects its overall movement speed and expended energy. This has a consequent effect on the supplies required from the settlements through which the army passes. The ABM represents members of the Byzantine army at a 1:1 resolution with each environment cell measuring approximately 5 m square and each tick of the simulation representing approximately 3.7 s of real time. Thus each day's march consists of up to tens of thousands of agents moving over thousands of environment cells over the course of around 10,000 ticks. The ABM records the location of each agent during each tick of the simulation as well as providing a summary file detailing each agent's distance travelled and calories expended during the day. This represents a large amount of data that has to be visualised, initially to the programmer in order for software errors to be detected and resolved, and subsequently for the researcher in order to draw conclusions from the model. Finally, the project is disseminated to a wider audience of interested parties.

13.3.1 Issues in Visualisation

Depending on the numbers of agents and length of the simulation, the ABM can take anywhere between a couple of minutes to a number of days to run. This presents problems to the production of working code. Some method of visualising the results of the model is required to ensure the ABM is running as intended, yet real-time methods of visualisation take up resources that could be used in the processing of the ABM itself. Also the scale of the environment and the large numbers of agents mean it can be problematic to display the behaviour of the model within a single display. Either the whole army can be shown in which case it is difficult to determine the behaviour of individual agents or only some of the agents can be on screen at any given time.

Any visualisation method also needs to be able to satisfy the need to present the data to the project's intended audience. This includes simulation researchers along with military historians, Byzantine historians and members of the public. The visualisation needs to be flexible enough to produce quick, low quality output that nevertheless enables all the workings of the ABM to be accessed for debugging purposes in addition to high quality output for presentation of the wider community of interested parties. It needs to be able to present the movement of the army as a whole in order to highlight macro-scale patterns whilst also being able to focus on individuals and to track an agent's path through the simulation. It needs to produce still images for traditional publications as well as animations that can present the movement of the army as a whole. As the effects of organisation and size on overall speed and energy consumption are also being investigated, statistics and graphs are required to communicate aggregated information on both individual runs of the simulation and multiple runs with different parameters.

13.3.2 The Architecture of the Project

Although it would be possible to design a bespoke visualisation method which will allow the user to move about the environment and focus on areas of interest while the ABM was running, no attempt at real-time visualisation was made due to the development time involved and the likely impact on the performance of the ABM. A modular system was designed (Fig. 13.1) in which the ABM outputs trace files that can be visualised outside the ABM by various means. This modular nature ensured that once run, a simulation did not need to be rerun for a different method of visualisation to be used. It also meant that any visualisation could be produced in parallel by copying the trace files onto a different machine while the ABM gets on with running the next simulation. The lack of real-time visualisation was partially mitigated by having continuous and customisable levels of text output on screen during the running of the ABM allowing the user to know at which point the

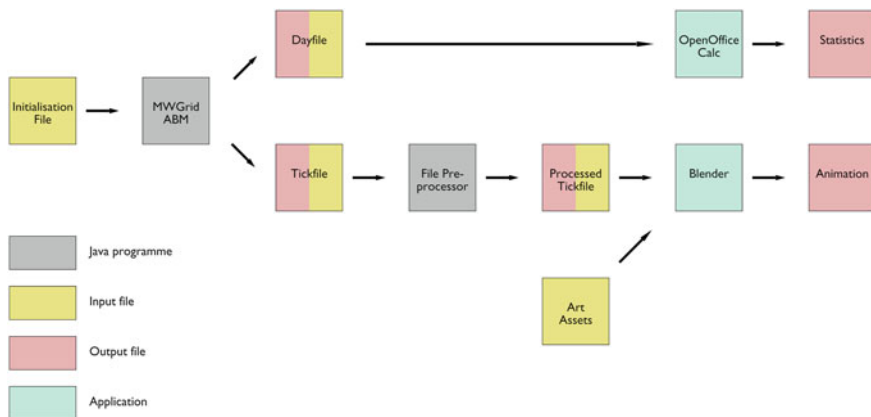


Fig. 13.1 The architecture of the MWGrid software

Table 13.1 Description of values contained on each line of the MWGrid tickfile

Value	Description
Tick number	The tick number of the simulation to which the data refers
Agent number	The agent’s unique id number
Agent type	A numerical representation of the agent’s type (1 = Column Leader, 2 = Cavalry Officer etc.)
X location	The agent’s position along the environment’s X axis
Y Location	The agent’s position along the environment’s Y axis

model was of its run at any given time. This proved sufficient to detect most problems.

The MWGrid ABM itself is a Java programme written specifically for the task. It takes in the parameters of its operation from the initialisation file, a plain text file containing a list of variables and their values. The ABM creates two output files during operation: a tick file (Table 13.1) containing the location of every agent at every tick of the simulation and a day file (Table 13.2) which contains one entry per agent summarising the activities that agent has engaged in, including distance travelled and calories expended. The day file is imported into an OpenOffice Calc spreadsheet which automatically summarises the information on one screen (Fig. 13.6). The tick file is used for visualisation but can be several gigabytes in its raw state if tens of thousands of agents are involved. For this reason a separate Java programme was written, the file pre-processor, which removes entries for ticks in which an agent does not move. These ticks are only used for debugging purposes and have no effect on the visualisation of agent movement. The processed tick file is then read by a Python programming script into Blender, an open source 3D modelling package. This creates each agent as a simple geometric shape

Table 13.2 Description of values contained on each line of the MWGrid dayfile

Value	Description
Agent number	The agent's unique id number
Agent type	A numerical representation of the agent's type (1 = Column Leader, 2 = Cavalry Officer etc.)
Arrival tick	The tick number on which the agent arrives at the destination camp
Total distance moved	Total distance moved by the agent in metres
Start tick	The tick on which the agent started its march from camp
Calories expended	The total number of calories expended during movement

or as a more complex animated figure depending on a series of art assets contained in separate Blender files. The result can either be rendered as 2D still images or animations depending on the requirements of the modeller.

13.3.3 Data Processing with Blender

The processed tick file is imported into Blender via a bespoke Python script. The Python programming language was chosen as it is the default scripting language for Blender and is heavily integrated with the application. The script creates an object for each agent based on its type and moves this object on a tick-by-tick basis depending on the location specified in the tick file. The object it creates can either be a simple geometric shape or a rigged, animated character depending on the requirements of the visualisation.

An initial version of the ABM focussed on individual agents performing complex behaviours involving item manipulation. It used as a case study, the setting up of the army's camp at the end of a day's march and included such behaviours as setting up each squad's tent and setting fires to cook food. As this involved behaviours that require multiple steps performed by multiple agents, a visualisation system was implemented to show the agents' behaviours in more detail, such as the tasks the agents are performing and the tasks they intended to perform in the future. The Python script used for reading the data into Blender inserts fully animated, humanlike models into the virtual scene to represent each agent. These humanlike models contained the mesh data for the agent itself and data for each animated action they could perform, along with speech and thought bubbles for tasks each agent was performing or intended to perform in the future, respectively. The Python script also imported 3D models from other Blender files for each type of item such as firewood, food and tents in both packed and erected forms.

The output worked well from a debugging point of view, as it was easy to identify each agent's behaviour at any given time. The modular nature, with the models being held in separate files, meant that it was easy to create two different types of 3D models, such as basic, low-polygon stick figures (Fig. 13.2) and more



Fig. 13.2 Low-polygon 3D models of a squad setting up camp

realistic 3D models (Fig. 13.3). This modular nature also allows professional 3D models to be integrated within the visualisation system if required. Although the system worked well for small numbers of agents, the details involved with this visualisation resulted in computational times that were unacceptable for large-scale simulations.

Users have the flexibility of manual positioning of the virtual camera within the display viewport, so that they can decide which agent behaviours to show in the final output. As the Blender file created by the Python script can be saved and easily “re-edited” by moving the camera into different locations around the scene, the same simulation datasets can be viewed from different angles for the users intended purpose.

As the focus of modelling shifted from setting up camp to simulating large-scale movement of thousands of agents, a much faster way of visualising ABM output was required. The Python script was pared down to represent each agent as a 2D geometric shape. Blender’s camera can be set to orthographic projection, this removes perspective distortions that may occlude the results. The end result was a 2D representation of the agents’ movement familiar to ABM researchers (Fig. 13.4). This meant that tens of thousands of agents could be modelled in a reasonable timescale. At this point of the development the focus had shifted to the movement of the agents over the day’s march and so the extra mechanisms used to



Fig. 13.3 Higher detail 3D models visualising the same tick of the simulation

indicate items or the intention of the agents were not necessary. The result shows movement well but has no way of representing the underlying terrain.

This type of output that is represented by simple geometric shapes, represented orthogonally on a plain background is commonly seen in academic publications related to ABMs. It, however, provides no indication of the way route finding works within the model. Within the MWGrid ABM, each column of the army, by way of its behaviour, selects a route based on the terrain and create the shortest possible route to their destination while avoiding unnecessary climbing of hills. A better way of visualising the route planning algorithm is to represent the agents' movement in a 3D landscape (Fig. 13.5). The existing Python script used earlier for producing the 2D representation of the agents' movement was modified to create 3D geometric shapes for each agent. The landscape was created in a separate script which simulated a 3D landscape of blocks representing the actual terrain data that is used in the ABM. The orthogonal camera view used in the 2D representation does not adequately show the 3D nature of the terrain so a perspective-based camera view is used. This allows freedom to create multiple visualisations from the same Blender file but requires the user to decide which behaviours in the model are most important and from which angle they can best be visualised.

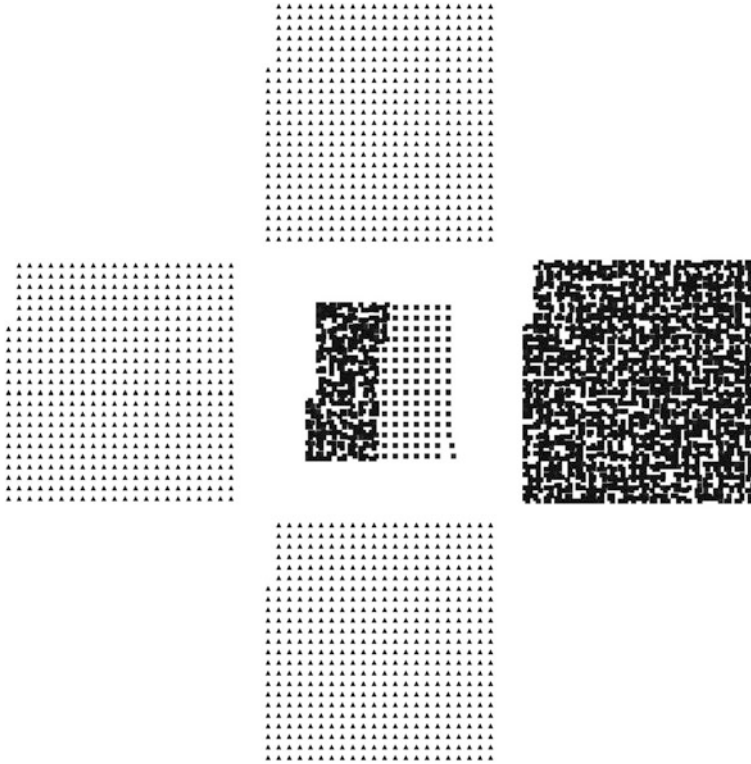


Fig. 13.4 A 2D representation of MWGrid agents

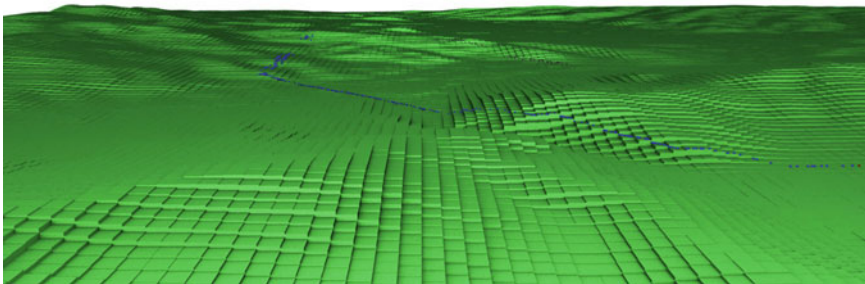


Fig. 13.5 A 3D representation of MWGrid agents moving across the landscape. The *lined dots* represent agents

13.3.4 Data Processing with OpenOffice

The model involves tens of thousands of agents all moving via subtly different routes at different times and expending different amounts of energy. Presenting this to others poses a problem as different aspects of a model's results are of interest to different people. One of the key points of using ABM is that the idea of what an army does, how far it moves and how much energy it expends, is an abstraction covering a lot of complexity in the behaviour of individual agents. The raw data is too complex to investigate and therefore, some form of data aggregation has to be conducted. A statistical summary of the simulation is needed at this point. The summary is derived from the day file, a text file output from the ABM showing aggregated data for each agent from the whole run of the simulation. This includes the total distance moved, ticks spent resting and energy expended in calories. An OpenOffice Calc spreadsheet was created which took this data and aggregated it into a series of statistics in table format (Fig. 13.6) illustrating the average, highest and lowest values for a range of values. It also shows the number of agents and the time used to complete the march. The unit of time the last agent arrives in camp provides an indication of the total amount of time taken by the army.

Fig. 13.6 Aggregated statistical output of a day's march

Average arrival time	6446.21
Average distance covered	27009.46
Average arrival time (Officers only)	6156.22
Last arrival tick	8197
Squads	1260
Agents	7061
Column Leaders	1
Cavalry Officers	1060
Infantry Officers	200
Cavalry Soldiers	4000
Infantry Soldiers	1800
Average Calories Expended	1480.98
Average travel time	4876.69
Column Leader cals	1037.36
Cavalry Officer average cals	1252.1
Cavalry Officer high cals	1467.06
Cavalry Officer low cals	1056.64
Cavalry Soldier average cals	1288.15
Cavalry Soldier high cals	1447.48
Cavalry Soldier low cals	1087.22
Infantry Officer average cals	1983.8
Infantry Officer high cals	2041
Infantry Officer low cals	1948
Infantry Soldier average cals	1988.64
Infantry Soldier high cals	2110
Infantry Soldier low cals	1956

13.3.5 Formats of Visualisation

The MWGrid ABM and its associated programmes are able to generate structured and highly visual datasets via Blender and OpenOffice Calc. This allows the presentation of the data as either statistical tables and graphs, or still images and animations. Different formats have different uses. Still images can give a snapshot of the model at work and can be a good way of examining spatial distribution of agents and resources, animation is often the most effective way to visualise behaviour over time, tables can show large amounts of data at the same time and graphs can track individual values over time. By using Blender as a rendering engine, still images can be made of any tick of any simulation. If the end result is to be reproduced in a printed journal then, given the scale of the Manzikert model and the number of agents used, any image that reproduces the complete army on the march will be at the wrong scale to identify individual features but can show patterns in the movement of the army as a whole. However, there is no reason for the output to be restricted to paper publications. Publication via the Internet allows much higher resolution images in which macro-scale patterns can be seen but each individual agent's movement can be focused on by zooming into an appropriate scale. Once the data is imported into Blender and the appropriate frame of the animation chosen, a high resolution still image can be generated.

In order to see the movement of agents within the model during its operation, animations must be created. Unlike stills, animations cannot easily be created so that the user chooses which parts to focus on, there needs to be an element of directorial control at the point of creating the animation. The virtual camera through which the animation is seen needs to be placed in appropriate locations and to have appropriate scale to visualise the required activity. This may result in interesting behaviour elsewhere in the model being missed due to the camera being elsewhere. Animations however are ideal for illustrating the actual micro-scale behaviours which make up the macro-scale results that are more often shown via statistics and graphs. Graphs can more accurately show data but they are not good ways of illustrating what an army on the march actually looked like.

The most complete way of visualising the behaviours of complex systems is to provide access to the systems themselves and allow users to perform their own simulations. If the system is published with full featured visualisation it allows users to specifically create the information they wish to see. They may be able to see the simulation running in real time or they may produce data from which visualisations can subsequently be made. This is an ideal scenario for simulations which require minimal processing time and small numbers of complete runs of the simulation but will be impractical for models requiring parallel processing, many thousands of iterations or days or even weeks of processing time. In these cases aggregated data is usually much more practical. The modular nature of the MWGrid system enables, for instance, the tick files from certain simulations to be made available so that they may be visualised in different ways by different people.

Others may even create their own Blender models which, as long as they adhere to certain naming conventions, can easily replace the models used in the visualisation seen here.

The production of statistics, stills and animations allows the project to present its data in traditional paper-based publications as well as via the Internet. In addition to using the outputs of the model to explain the results of the project, the art assets and initial Python script were used to produce an explanatory video in the design stages of the project. The Road to Manzikert (Murgatroyd 2009) was part of a university-funded pilot scheme to provide the equipment and technical expertise to allow Ph.D. students to present their research in a short film. The result has been a useful introduction to the project and although it describes the project in an earlier form rather than as it subsequently developed, it was a useful exercise in presenting a specialised project to a lay audience.

13.4 Conclusion

As can be seen from the MWGrid project, ABMs can produce complex data involving large numbers of agents performing a variety of actions over long periods of time. The raw data requires some form of visualisation to make sense but the appropriate form of this visualisation depends on the target audience, the method of distribution and the technical skills available. Due to the fact that it is not only the results but also the process used to produce them that is important, multiple methods and media are required to completely document an ABM. As ABMs are still a relatively new investigative technique within archaeology, existing work is presented in a variety of ways, sometimes as graphs, sometimes with 2D or 3D graphics and sometimes the whole model is available to use and alter.

In order to communicate both the process and results of an ABM, non-traditional outputs can widen the usefulness of research as well as the scope of the audience although traditional publishing is valuable in order to use the audiences that journals and books have access to. Some electronic supplemental data can provide access to more detailed still images, animations and the actual model itself, the most complete method of documentation and presentation. With the easy availability of Internet publication of text and images as well as large audiences for animation and video via services such as YouTube and Vimeo, traditional publishing methods need not, and arguably should not be the only method of dissemination.

There is as yet no fully featured way to produce and visualise ABMs without some technical knowledge and so the techniques used to create archaeological ABMs will depend on the skills available within a project. Issues with visualisation will exist regardless of whether technical solutions are provided by existing toolkits or with bespoke software. Publication via static 2D methods will prefer static 2D data. Data generated from ABMs are dynamic, and in 4D. It shows the

processes involved in the agents operations. Any proposed technical solution will have to be able to cope with the variety of scales, topics and levels of detail possible within archaeological ABMs.

At present, the MWGrid project is in the final stages of data collection and analysis with a view to publication via a variety of media. Traditional publication remains an important part of a research but additional publication via methods more suited to 4D data will be key if a proper representation of the processes involved with both producing the ABM and moving the Byzantine army is to be achieved. Finally, the ABM and any accompanying visualisation methods will be freely available for researchers to use, reuse, improve and otherwise do what they wish with.

Simulation is a growing area of research and archaeological simulation will grow with it. The increasing use of simulation in archaeology means that there are an increasing number of archaeologists with the skills and experience necessary to create, reuse and alter ABMs. As the number of archaeological ABMs increase with demands, the range and size of their audiences will grow in proportion, new ways of representing data through visualisation will need to be developed.

References

- Axelrod, R. (2007). Simulation in the social sciences. In J.-P. Rennard (Ed.), *Handbook of research on nature inspired computing for economics and management* (pp. 90–100). Hershey: IGI Global.
- Ch'ng, E., Gaffney, V. L., Chapman, H. (forthcoming) From product to process: New directions in digital heritage. In S. Wu, H. Ding (Eds.), *digital heritage and culture : strategic and implementation*. Singapore: World Scientific.
- Cioffi-Revilla, C., Rogers, J. D., & Latek, M. (2010). The MASON HouseholdsWorld model of pastoral nomad societies. In K. Takadama, C. Cioffi-Revilla, & G. Deffuant (Eds.), *Simulating interacting agents and social phenomena: The second world congress* (Vol. 7, pp. 193–204). Tokyo: Springer-Verlag.
- Costopoulos, A., Lake, M., & Gupta, N. (2010). Introduction. In A. Costopoulos & M. Lake (Eds.), *Simulating change: Archaeology into the twenty-first century* (pp. 1–8). Salt Lake City: University of Utah Press.
- Craenen, B., Theodoropoulos, G., Suryanarayanan, V., Gaffney, V., Murgatroyd, & P., Haldon, J. (2010) Medieval military logistics: A case for distributed agent-based simulation. *Proceedings of the 3rd international ICST Conference on simulation tools and techniques*. doi:10.4108/ICST.SIMUTOOLS2010.8737.
- Doran, J. (1970). Systems theory. *Computer Simulations and Archaeology. World Archaeology*, 1(3), 289–298.
- Haldon, J. (2006). *General issues in the study of medieval logistics*. Leiden: Brill.
- Hegselmann, R. (2012). Thomas C. Schelling and the computer: Some notes on Schelling's essay "On Letting a Computer Help with the Work". *Journal of Artificial Societies and Social Simulation*, 15(4), 9.
- Holland, J. H. (1998). *Emergence: From chaos to order*. Boston: Addison-Wesley Longman.
- Janssen, M. A. (2009). Understanding artificial anasazi. *Journal of Artificial Societies and Social Simulation*, 12(4), A244–A260.

- Kohler, T. A., Johnson, C. D., Varien, M., Ortman, S., Reynolds, R., Kobti, Z., et al. (2007). Settlement Ecodynamics in the prehispanic central Mesa Verde region. In T. A. Kohler & S. E. van der Leeuw (Eds.), *The model-based archaeology of socionatural Systems* (pp. 61–104). Santa Fe: School for Advanced Research.
- Kohler, T. A., & Varien, M. D. (2012). *Emergence and collapse of early villages: Models of central Mesa Verde archaeology*. Berkeley: University of California Press.
- Kohler, T. A., Varien, M. D., Wright, A., & Kuckelman, K. A. (2008). Mesa Verde migrations new archaeological research and computer simulation suggest why ancestral Puebloans deserted the Northern Southwest United States. *American Scientist*, 96, 146–153.
- Luke, S., Cioffi-Revilla, C., Panait, L., Sullivan, K., & Balan, G. (2005). MASON: A multiagent simulation environment. *Simulation*, 81(7), 517–527.
- Murgatroyd, P. S. (2009) The road to Manzikert. Retrieved 8 Mar 2013 from <http://www.youtube.com/watch?v=xnZK1qlX6UI>.
- Schelling, T. C. (1971). Dynamic models of segregation. *The Journal of Mathematical Sociology*, 1(2), 143–186.
- Varien, M. D., Ortman, S. G., Kohler, T. A., Glowacki, D. M., & Johnson, C. D. (2007). Historical ecology in the Mesa Verde region: Results from the village Ecodynamics project. *American Antiquity*, 72(2), 273–299.
- Wilkinson, T. J., Christiansen, J. H., Ur, J., Widell, M., & Altaweel, M. (2007). Urbanization within a dynamic environment: Modeling bronze age communities in upper Mesopotamia. *American Anthropologist*, 109(1), 52–68.
- Wobst, H. M. (1974). Boundary conditions for paleolithic social systems: A simulation approach. *American Antiquity*, 39(2), 147–178.