Medieval Military Logistics: A Case for Distributed Agent-based Simulation

Bart Craenen
Georgios Theodoropoulos
Vinoth Suryanarayanan
School of Computer Science
The University of Birmingham
United Kingdom
{ gkt, b.g.w.craenen, vys}
@cs.bham.ac.uk

Vincent Gaffney
Philip Murgatroyd
Institute of Archaeology and Antiquity
University of Birmingham
United Kingdom
{v.l.gaffney, psm703}@bham.ac.uk

John Haldon
History Department
Princeton University
U.S.A
jhaldon@Princeton.edu

ABSTRACT

Historical studies are frequently perceived to be characterised as clear narratives defined by a series of fixed events or actions. In reality, even where critical historic events may be identified, historic documentation frequently lacks corroborative detail that supports verifiable interpretation. Consequently, for many periods and areas of research, interpretation may rarely rise above the level of unproven assertion and is rarely tested against a range of evidence. Simulation provides an opportunity to break cycles of academic claim and counter-claim. This paper discusses the development and utilisation of large scale distributed Agent-based simulations designed to investigate the medieval military logistics in order to generate new evidence to supplement existing historical analysis. The work aims at modelling logistical arrangements relating to the battle of Manzikert (AD 1071), a key event in Byzantine history. The paper discusses the distributed simulation infrastructure and provides an overview of the agent models developed for this exercise.

Categories and Subject Descriptors

I.6 [**Simulation And Modeling**]: Types of Simulation - *Discrete event, Distributed, Gaming.* Simulation Support Systems - *Environments*

I.2.11 [**Distributed Artificial Intelligence**] *Multiagent systems*

General Terms

Algorithms, Design, Experimentation

Keywords

Agent-based modelling, distributed simulation, historical studies, medieval, military logistics

1. INTRODUCTION

The analysis of humanities data sets offers considerable challenges to computational science [9]. Large, complex and often characterised as partial or fuzzy, their interpretation is frequently presented in the form of assertion and the prospect of formal analysis is often dismissed as mechanistic and inappropriate to the complexities of human action or behaviour. Historical studies are a good example of the difficulties associated with such research. For many, historical interpretation is associated with clear narratives defined by a series of fixed events or actions. In reality, even where critical historic events may be identified, contemporary documentation frequently lacks corroborative detail that supports verifiable interpretation. Consequently, for many periods and areas of research, interpretation may rarely rise above the level of unproven assumptions, rarely or never tested. Constant, subjective, argument over the same texts hampers a deeper understanding of the finite evidence that is available to historians.

In such an academic context there is an imperative to provide alternative, novel paths toward interpretation. Computer simulation provides an opportunity to break cycles of academic claim and counter-claim. Having made such a statement it would not be true to suggest that there is no prior quantitative, or computational, base to historical studies. Many areas of research have voluminous data sets, although their study, generally, provides abstract numeric outputs and provides a limited insight into detailed or individual action, and this limitation has been the object of heated debated within historical disciplines for several decades [15]. Where significant computational linkage exists the technology of choice for many historical disciplines has been GIS [1][3]. The applications of GIS have been variously significant or insightful but, most often, can be characterised as static models, frequently dependent on a limited, economic database, even where they may have aspirations towards the explanation of larger behavioural patterns[10].

The application of agent-based modelling, as a means to explore the effect of individual action, has recently emerged as an area of interest for the historical disciplines. Application of such technologies has, however, been extremely limited. The analysis of resource exploitation by Mesolithic hunter gatherer groups, for instance, was an early example of such work [18], whilst the process of evolution of complex societies in the fertile crescent has also been the object of sustained study [27]. Examples such as these which build on more traditional historical and archaeological work often have the benefit of being able to use certain well established known points as validation for their models. It is notable, however, that most recent studies have generally involved the analysis of small-scale groups at individual or household level rather than larger societies [17].

DISIO 2010 March 15, Torremolinos, Malaga, Spain.

The work presented in this paper seeks to study behavioural action at a larger scale, involving tens of thousands of agents within the context of modelling logistical arrangements relating to the battle of Manzikert (AD 1071), a key event in Byzantine history that resulted in the collapse of Byzantine power in central Anatolia [13]. Distributed simulation is the only viable approach to deal with a problem of such scale and complexity.

This work is part of the "Medieval Warfare on the GRID" (MWGrid[1]) project, one of the seven strategic pilot projects of the e-Science Programme of the Arts and Humanities Research Council in the UK, which builds on a series of parallel research developments in Birmingham and Princeton relating to the analysis of medieval military logistics and the development of novel computational infrastructure designed to investigate complex systems. The project aims to capture and analyse the behaviour of a major Byzantine army marching through a digital representation of the Asia Minor terrain to study the effects that the environment, command structures and decision making have on the wellbeing of such a large military force. The representation of the Asia Minor terrain is developed using a unique set of data including multi-spectral satellite mosaics, vegetation maps, geology maps, elevation models and historical maps incorporating detailed topography, route and demographic data.

The paper provides a overview of the MWGrid framework and discusses the agent models developed for such an exercise and the distributed simulation infrastructure utilised. The rest of the paper is structured as follows: Section 2 provides an overview of the domain problem and the requirements from the simulation model. Section 3 presents a high level view of the overall simulation framework, while sections 4 and 5 discuss respectively the Agent based models and the distributed simulation infrastructure. The paper concludes with a summary and an outline of future work.

2. MEDIEVAL MILITARY LOGISTICS: THE MANZIKERT CAMPAIGN

A key problem in historical studies is to analyse and understand the implications of the need for medieval states to collect and distribute resources to maintain armies. It is apparent from the historical record that these requirements affected all aspects of political organisation and, at critical times, when armies failed, the results could prove disastrous to society as a whole. Despite this, it is also clear that military studies seldom progress past the study of existing texts to bear out the pragmatic consequences of military behaviour, even though military activity in terms of resource allocation and consumption was decisive in shaping pre-modern societies.

Study of the events associated with the Byzantine army's march across Anatolia to the Battle of Manzikert in AD1071 is particularly attractive in this context. This was a major logistical challenge that involved the largest Byzantine army for over 50 years travelling more than 700 miles across what is now part of the modern state of Turkey, from near Constantinople (modern Istanbul) to Manzikert (modern Malazgirt) just north of Lake Van (Figure 1).

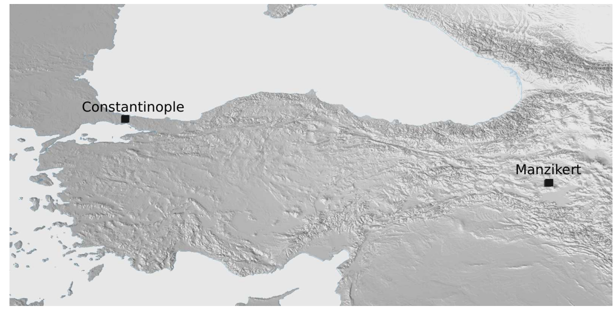

Figure 1. Anatolia

The army set out from near Constantinople in March of 1071 and arrived at the border fortress of Manzikert in August. The Byzantines' subsequent defeat by the Seljuk Turks and their Sultan Alp Arslan on August 26th was considered so catastrophic that the Byzantine chroniclers dubbed it “the dreadful day” [6], and it was the last time the Byzantine Empire exerted even minimal control over the whole of Anatolia.

Considering how important the battle was to the Byzantine Empire, the Seljuks Turks and the modern republic of Turkey, the historical sources leave several significant gaps. Historical sources are vague, contradictory or absent for such factors as how many people were in the Byzantine army or which route it took. Byzantine writers do not give numbers for the size of the Byzantine army, their foreign contemporaries give figures between 200,000 and 1 million but these are largely motivated by a desire to emphasize the scale of the Byzantine defeat. Modern historians tend to put the figure at between 40-70,000 men but there is a definite need to introduce new types of evidence into a debate where all currently available data sources have been analysed.

In addition to the number of people in the Byzantine army, the route to Manzikert is also uncertain. There are reports that the army passed through Ankyra, Sebasteia and Theodosiopolis (modern Ankara, Sivas and Erzerum, respectively) but the rest of the route is largely unknown. The details of the journey to Manzikert fill far less of the contemporary accounts than that of the battle itself. Despite this, understanding the context of how the Byzantine state supported the army and its progress underpins not only our understanding of the likely composition of the force and its capacity for movement, it also supports in some manner how we understand the momentous outcome of the campaign. The less controversial aspects of the historical record are also useful in providing verifiable details against which to compare the model's outputs.

Formal modelling of military activity is of course, quite common when we consider battlefield scenarios. The are frequently associated with, for instance, variations on combat or game theory [8]. However, sophisticated analysis of logistical behaviour is essentially unknown in the literature for medieval military history and where discussion does take place these often rely upon comparison with later armies or data for completely unrelated campaigns [12].

The MWGrid project aims to sue an agent-based model of the Byzantine army's march to Manzikert to investigate the transport of tens of thousands of people, horses, mules along with tons of

[1] http://www.cs.bham.ac.uk/research/projects/mwgrid

equipment over 700 miles through the Anatolian summer in order to provide an insight into this pivotal historic event and. By modelling different scenarios based on historical records and modern interpretations of how the Byzantine infrastructure supported an army on campaign, we aim to draw valuable conclusions regarding how the transport, taxation, agricultural production and military organisation systems interconnect.

The project centres around agents representing all the members of the army. The commander through to the lowliest servant occupy part of an military structure with one clear goal; to arrive at a destination in a fit state to win a battle. They act as part of a hierarchical organisation but have a certain amount of autonomous decision making capability and travel through an environment that contains a variety of resources required to complete their journey. Multiple executions of the agent-based model are required with different numbers of people and animals, different levels of food availability and different types of organisation and route planning. The simulation will record both the state and progress of the army as well as the effects on the communities impacted by the progress of the army.

An army of between 40-70,000 people with attendant horses and pack animals requires over 100,000 agents in order to be modelled on a 1:1 basis. Clearly the processing power and the memory requirements needed for this simulation far exceed the capabilities of any sequential von Neumann machine. Distributed simulation and the harnessing of distributed computing resources emerge as the only viable approach to deal with a problem of such scale.

3. THE MWGrid FRAMEWORK

A high-level view of the MWGrid framework is provided in Figure 2. It can be seen as consisting by two major parts: the simulation system and the analysis environment.

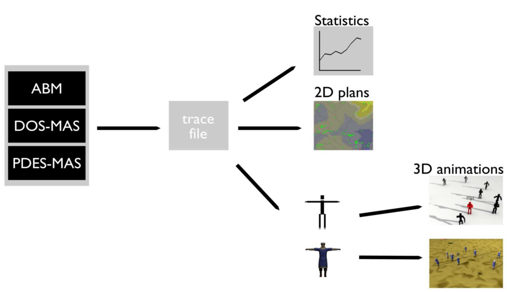

Figure 2. The MWGrid Framework

The simulation system executes the model and produces detailed trace files that are fed into the analysis system for off-line post-processing. This is achieved via a range of packages depending on the output required. Statistics can be produced detailing movement rates, food consumption, agent health status, amount of time spent on the move and the state of the environment after the army has moved on. These can involve individual agents or the aggregation of statistics of the whole army or certain subgroups. It is also possible, via a Python script, to export the trace data into a 3d modelling package so that 3d visualisations can automatically be created. This allows the creation of different sets of art assets that can be used to produce different types of animation. The ability to display the model's results in a more realistic and instinctively understandable 3D representation is an important tool in communicating the results of the model to its intended audience, some of whom are unused to processing the typical 2D output of traditional agent-based modelling. Realistic outputs have their dangers though, as they can convey an artificial sense of authority due to their persuasive nature. For this reason, the ability to produce more abstract results is useful.

The simulation system consists of three layers:

1. The Agent-based Model (ABM)
2. The distributed simulation kernel (PDES-MAS)
3. The Middleware (DOS-MAS), which provides the 'glue' between the other two.

These three layers interact with each other in clearly described ways, using specific interfaces. A description of these layers is provided in the following sections.

4. THE MODEL

The model consists of two main elements: the environment representing the terrain, infrastructure and resources of Anatolia and the agents, representing the human and animal members of the Byzantine army's campaign on a 1:1 basis.

4.1 The Environment

11th century Anatolia represents a large and rich environment for an agent-based modelling, the distance from Constantinople to Manzikert is over 700 miles as the crow flies and contains a variety of different terrain types. The data used for modelling the environment is split into a series of slices, each dealing with a different aspect of the environment, listed in Table 1.

Table 1. Environment slices and their sources

Environment Slice	Data type	Source
Terrain	Height	Shuttle Radar Topography Mission[23]
Water sources	Presence and amount of water	Large rivers from modern sources. Smaller streams procedurally generated based on climatic and environmental conditions
Transport infrastructure	Presence and size of roads	Tabula Imperii Byzantini[14]
Settlements	Presence and size of settlements	Tabula Imperii Byzantini with finer detail procedurally generated based on more detailed studies
Animal fodder	Type and amount of suitable vegetation	University of Birmingham's Vegetation Modelling and Cultural Landscapes in SW Turkey project
Weather	Rainfall and temperature	Procedurally generated, based on above project

The data comes from a variety of sources in a variety of formats and resolutions. Some data such as water sources, terrain and weather has to be based on modern data. Where we are uncertain about a particular area of the environment, different scenarios can be created to investigate how competing hypotheses affect the system as a whole. There is still much uncertainty regarding pre-modern Mediterranean food production and the surpluses that would be available to an army on the march[16]. Altering the food surplus available along the route in line with existing hypotheses will enable us to assess their ability to support an army on campaign.

Some environmental resources such as transport infrastructure and terrain will remain static while others such as water, forage and firewood can be altered during the run of the simulation. When agents take a resource such as firewood, the value in the environment is reduced and an object is created that can be carried and manipulated by agents. Water, for instance, is not a static resource and taking water from a stream will result in less water being available downstream.

4.2 The Agents

The granularity and complexity of the agents is dictated by the project's aims. The practical problems involved with moving large numbers of people across broken terrain and through narrow ravines were known to medieval military leaders. Average movement rates have been calculated for armies based on historical itineraries but the relationship between army size and composition and the speed of march is one aspect that the model is ideally suited to address.

Table 2. Agent attributes

Attribute	Description
Agent ID	Unique agent identification number.
Rank	Numerical representation of rank to resolve issues of superiority.
Unit Number	Unique identifier for the unit the agent belongs to.
Health	Number from 1-100 indicating how well (100) or ill (1) the agent is.
Vision Range	A numerical value in meters giving the maximum unobstructed distance an agent can see.
Movement Speed	The maximum speed in meters per second that an agent can move.
Energy	The amount of food energy in kilojoules that the agent has received through food consumption.
Role	An identifier of the agent's role in the army's organisation e.g., Infantry, Bureaucrat, Emperor etc.

With this is mind we have decided to model the people and pack animals of the army at a ratio of one agent per human or animal. This will result in a more convincing crowd movement model and simplify the modelling task as there will be no problems with deciding how to aggregate several individuals into one agent.

4.2.1 Architecture

An agent consists of a plan queue, a messaging inbox and a perception base along with a series of private and public variables modelling personal characteristics (Figure 3). The plan queue contains a list of the tasks the agent has to perform. The inbox contains a list of messages from other agents, including orders from superiors and messages from comrades. The perception base contains the information that the individual agent has regarding the world, including information gathered by the agent's own senses and information introduced by communication with other agents.

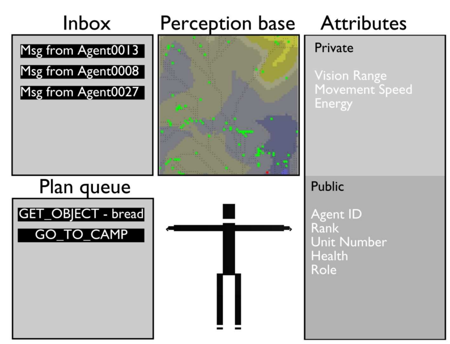

Figure 3. The MWGrid Agent

Each agent has a series of variables, depending on its type (table 2). These are dictated by the need to model the organisation of the army and its movement and the effects of the march on each individual agent.
An agent's variables can be public (those that are apparent to other agents such as health and rank) and private (movement speed, vision range). It can perform range queries to access the public variables of every agent within a certain distance. Orders are given when another agent passes a message to an agent with an order as the content. Providing the message comes from a valid source (a superior or a trusted comrade) the content of the message is added to the agent's plan queue.

4.2.2 Movement

Route planning exists in a number of forms. The main route of the army is decided by the head of the army (usually the Emperor), as is any decision to move the army or not during a day. Rest days will be needed, not only for the human members of the army but also for the animals, horses need at least one rest day per week[4]. Any decisions on the main route of march will be taken based on transport infrastructure and levels of supplies relative to where supplies can be found in the environment. These decisions will be based on intelligence received from other army members.

Route planning by individual army members is limited, one of the attractive qualities of using the march of an army as a case study is that autonomy is strictly regulated. Each agent is not free to create their own route across Anatolia, they must follow the route the head of the army selects. Route planning will be necessary for foragers, parties sent from the main body of the army to gather supplies from settlements not along the route of march.

This limited use of route planning enables us to adopt a tiered movement model where commanders of units which have autonomous goals can plan routes over a probabilistic roadmap (PRM) node network and their subordinates can follow using modified flocking behaviours (Figure 4).

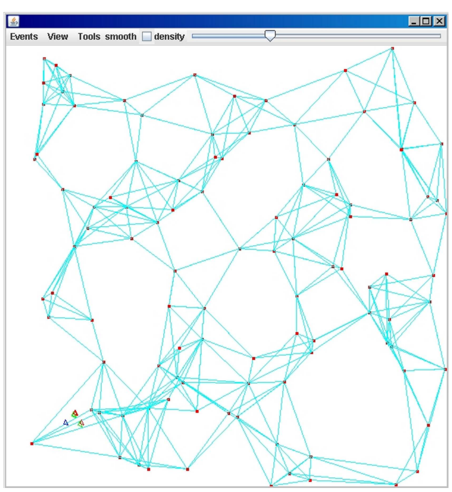

Figure 4. A PRM node network with agents

Probabilistic roadmaps involve the creation of a series of randomly placed nodes that are then joined by routes along which the agent may travel. A cost is associated with each route derived from any undulations in terrain or other factors which may help movement (roads, clear terrain) or hinder it (broken ground, streams, woods). This allows route planning agents to use A* planning to find the route that has the least 'cost' involved between their current location and their destination.

4.2.3 Planning

Each agent has a plan queue where an agent's designated tasks are stored in the order in which they need to be performed. The current plan to be executed consists of a series of actions. Plans further down in the queue consist of one symbolic action that is expanded into a series of appropriate actions when it is executed (Figure 5).

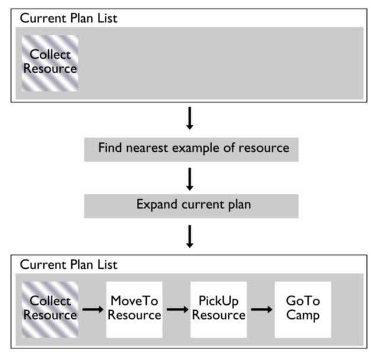

Figure 5. The expansion of a plan into a series of actions

If these subsequent actions need revising then the action queue can be cleared back to the original first action and then expanded again. This is useful if an agent has a plan to pick up a resource, creates a plan to move to the nearest example of the resource then finds that someone else has taken the resource in the meantime. In this case all subsequent actions can be cleared and the original action can be reprocessed, leading to the next nearest resource being found.

This planning process works well with the limited number of actions required in this agent-based modelling. Due to its highly logistical nature, the majority of the tasks involve moving somewhere, picking an object up or dropping it. With the addition of a 'priority' attribute and a 'plan to execute in event of failure' in each plan, each agent should have enough information to be able to prioritise its own tasks and fail in an intelligent way.

As an example, a group unit of 9 soldiers were created with the task to set up a camp for the night. This involved setting a fire, making some bread and setting up their tent. The unit leader gave a series of orders to his subordinates who had to gather the resources needed for these tasks (randomly placed around an otherwise empty landscape in this example) and carry out any preparations needed. This was accomplished with six symbolic actions and three other actions that could be used when the symbolic actions were expanded (Table 3).

The scenarios we wish to model as part of the Byzantine army's march across Anatolia can be modelled with relatively basic object handling plans (Collect Resource, Pickup Object, Drop Object), basic message passing instructions (Give Order, Request Information, Pass Message) and some plans referring to specific situations needed during the army's march (Set Fire, Setup Tent, Dig Ditch, Patrol). The limited number of plans required and the restricted set of circumstances in which they will be use means this approach in which the process of performing tasks is largely hard-coded doesn't increase the time involved with programming the model unreasonably.

Table 3. Sample actions used in setting up camp

Symbolic Actions	Other Actions
Give Order Collect Resource Set Fire Make Bread Serve Bread Setup Tent	Pickup Object Drop Object Go To Camp

4.2.4 Messaging

Every agent in the army has an appreciation of where they are in the army's organisational structure. They know who their superiors are and who is under their command. They also know the rank of all other agents. In this way an agent can prioritise messages received from other agents.

Each message consists of the Agent ID of the sender, the Agent ID of the intended recipient and the content which can be a piece of information about the environment, some information about another agent or an order. It is placed in the recipient's message inbox to await processing. On being processed either the information is added to the agent's perception base or the order, if from a valid source, is added to the agent's plan queue.

5. THE DISTRIBUTED SIMULATION KERNEL

As described in the previous section, the agents in MWGrid are based on a simple sense–think–act cycle that each agent executes repeatedly. Information obtained from the environment via sensing is used, together with the agent's state (e.g. its beliefs and goals) to choose one or more actions which are then executed, updating the environment. The cycle then repeats. The environment therefore constitutes a key medium for the agents' interactions. The environment may be represented as a simple 'passive' data structure which records the public attributes of objects and agents in the simulation, (e.g. position), which is updated directly by the agents, or it may be managed by one or more 'environment agents' which are responsible for computing the consequences of the action(s) of the agents and updating the environment accordingly.

The simulation of situated agents, as those utilised for military logistics modelling, presents particular challenges for standard parallel discrete event simulation (PDES) models and techniques as described in [7][5].

In a conventional decentralized event-driven distributed simulation, the simulation model is divided into a network of Logical Process (LPs). Each LP maintains its own portion of the simulation state and LPs interact with each other in a small number of well-defined ways. The topology of the simulation is determined by the topology of the simulated system and is largely static.

In contrast, in an multi-agent system an agent's interaction with other agents and its environment is hard to predict in advance indeed, discovering how the agents interact with each other and their environment is often a primary goal of the simulation. For example, what a mobile agent can sense is a function of the actions it performed in the past which is in turn a function of what it sensed in the past. This makes it hard to determine an appropriate topology for a MAS simulation a priori, and simulations of MAS typically have a large shared state which is only loosely associated with any particular process [21].

Another important problem is synchronisation. In conventional distributed simulations we often know the lower bound on the timestamp of an event generated by an LP in response to an input event. In contrast, a defining characteristic of agents is their autonomy [28]. In a parallel discrete event simulation of a multi-agent system, agents may spontaneously generate an event at any point without there being a preceding input event. As a result, simulations of MAS typically have zero lookahead [26].

To address these issues, we have developed the PDES-MAS framework for the simulation of multi-agent systems [21]-[24].

PDES-MAS adopts a standard parallel discrete event approach using optimistic synchronization strategy as this theoretically gives the greatest speedup and avoids the problem of lookahead.

Within PDES-MAS each agent is modelled as a single Agent Logical Process (ALP). An ALP has both private state and shared state. The private state is maintained within the ALP. In PDES-MAS, the sensing and acting phases of the agents' sense–think–act cycle are realized in terms of time-stamped operations on the shared state. ALPs interact with the shared state by reading and writing shared state variables (SSVs): sensing gives rise to read events, and acting gives rise to write events. SSVs are similar to space-time memory [11] and other work on shared state variables in distributed simulation (e.g., [22]), and have similar advantages in offering a more natural problem representation and improved performance when state variables must be accessed by distinct logical processes.

In PDES-MAS, the shared state is maintained by a tree-structured set of logical processes referred to as Communication Logical Processes (CLP), which cluster agent models and shared state according to the agents'. As the access patterns on the shared state change, so does the configuration of the tree and the distribution of state (i.e., its allocation to CLPs) to reflect the logical topology of the model.

Redistribution of shared state can be achieved in a number of ways, such as by creating/deleting CLPs, by migrating ALPs through the tree, or by migrating state between CLPs. For the purposes the MWGrid project, we have chosen to use a fixed tree of CLPs and migrate SSVs through the tree to achieve redistribution. SSVs are migrate dynamically closer to the ALPs that access them most frequently, reducing the total access cost and thus contributing to the scalability of the framework.

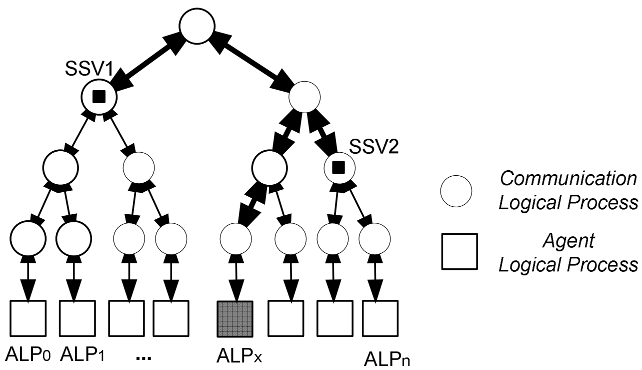

Figure 6: The PDES-MAS Framework

The framework does not make use of proxies and only one instance of an SSV is present in the tree at any particular moment. ALPs link to the leaf CLP nodes in the tree. An ALP issues requests to access shared state variables through its parent CLP. If the required SSV is not held locally, the parent CLP passes the request up through the tree. The return data and other control messages are also conveyed to the ALP via its parent CLP.

5.1 The Middleware

The Model and PDES-MAS are "glued" by means of a middleware layer which provides the bindings between the two. The middleware implements a Distributed Object Model and provides the Model layer with basic templates and abstract objects to use, as well as an interface to access shared state variables. It also provides bindings that translate these access requests into calls to the underlying PDES-MAS system.

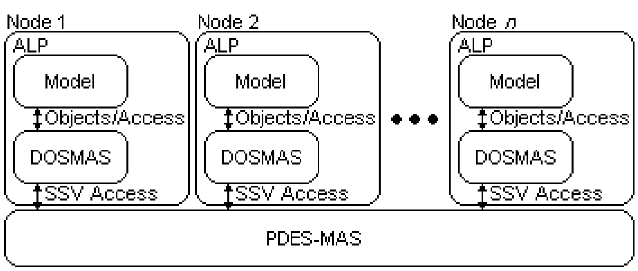

Figure 7. Simulation Architecture

Another aspect of the functionality of the Middleware layer is to provide initialisation, start-up, and eventual output collection of/from the simulation.

6. SUMMARY AND FUTURE WORK

This paper has described an infrastructure for the distributed simulations of large scale agent-based models to study medieval military logistics. An important event that shaped the world history, the Manzikert campaign, is used as a case study. Details in the historical record provide important, well established details that can be used to validate the model's outputs.

This model deals with the organisation and movement of resources and the effects of their presence, or absence, on the progress of the army. Other aspects may also affect the state of both the army and it's soldiers. Disease can spread quickly among large bodies of people living in often unsanitary conditions. Its spread depends on a variety of factors such as general health, food and water quality, interactions between agents and provision of healthcare. Time limitations and the absence of any record that it formed a significant part of the Manzikert campaign mean it is not included in any detail in this model, it would be a useful component of any future work.

From the point of view of historians, by modelling the Byzantine army in such an unprecedented detail we will be able to add new types of evidence into a debate that has up until now focussed largely on historical sources. Key issues such as the ability of the army to cross broken and restricted terrain, the amount of food required to support the army and the way in which transport infrastructure and settlement patterns affect route planning can all be modelled in detail for the first time. The results will not give us definite answers to these questions, rather the scenarios we model will help us start to define parameters within which we can reassess historical sources.

From the point of view of Computer Science, the development of models of such scale and complexity present the simulation community with opportunities but also with important challenges: what agent architectures are appropriate to capture the essential characteristics of the problems and support the required scenarios?; what is an appropriate model for distribution for models with such behavioural characteristics? As the development and integration of the infrastructure described in this paper is coming to completion, it will enable us to provide some answers to the above questions and evaluate the suitability and performance of our approach.

7. ACKNOWLEDGMENTS

This project is funded by a joint AHRC/EPSRC/JISC e-Science research grant.

8. REFERENCES

[1] Chapman, H 2006. Landscape Archaeology and GIS. Tempus , Stroud.

[2] Chen, D. Ewald, R. Theodoropoulos, G. Minson, R. Oguara, T. Lees, M. Logan, B. and Uhrmacher, A. "Data Access in Distributed Simulation of Complex Systems", Journal of Systems and Software, Elsevier, Volume 81, Issue 12, December 2008, Pages 2345-2360.

[3] Connolly, J. and Lake, M. 2006 Geographic Information Systems and Archaeoilogy. Cambridge University Press.

[4] Engels, D.W. 1978. Alexander the Great and the Logistics of the Macedonian Army. University of California Press.

[5] Ferscha, A. 1996. Parallel and distributed simulation of discrete event systems. In Parallel and Distributed Computing Handbook, A. Y. Zomaya (ed), pp. 1003–1041.

[6] Friendly, A. 1981 The Dreadful Day. Hutchinson & Co. Ltd. London.

[7] Fujimoto, R. 1990. Parallel discrete event simulation. Communications of the ACM 33(10), 31–53.

[8] Gaffney, H. 2005. Superiority of Numbers: Methodologies for modelling the behaviour of armies. In General issues in the study of medieval logistics: sources, problems, methodologies. J.F. Haldon, (Ed). Brill, Leiden

[9] Gaffney, V. 2008. In the Kingdom of the Blind: Visualisation and E-Science in Archaeology, the Arts and Humanities M. Greeengrass and L. Hughes (Eds) The Virtual Representation of the Past. 125-135 Ashgate

[10] Gaffney, V. and White, R. 2007. Wroxeter, the Cornovii and the Urban Process. Journal of Roman Archaeology. Supplementary Series. 68. (Portsmouth (Rhode Island)

[11] Ghosh, K. and R. M. Fujimoto. 1991. Parallel discrete event simulation using space-time memory. In Proceedings of the International Conference on Parallel Processing, volume III, Algorithms & Applications, pp. 201–208.

[12] Haldon, J. 2005. Why model logistical systems? In General issues in the study of medieval logistics: sources, problems, methodologies. J.F. Haldon, (Ed). Brill, Leiden

[13] Haldon, J.F. (Ed) 2005 General issues in the study of medieval logistics: sources, problems, methodologies. Brill, Leiden.

[14] Hellenkemper, H. & Hild, F. 2004. Lykien und Pamphylien. Verlag der Österreichischen Akademie der Wissenschaften, Wien

[15] Jenkins, K. 1991 Rethinking History. Routledge.

[16] Koder, J. 2005 Land Use and Settlement: Theoretical Approaches. In General issues in the study of medieval logistics: sources, problems, methodologies. J.F. Haldon, (Ed). Brill, Leiden

[17] Lake, M. 2007 Agent-based modeling: a question of scale. AHRC ICT Methods Network Workshop on Space and Time: Methods of Geospatial Computing for Mapping the Past http://www.nesc.ac.uk/talks/772/Lake_e_sci_07_v1.pdf

[18] Lake, M. W. 2000. MAGICAL computer simulation of Mesolithic foraging. In Dynamics in Human and Primate Societies: Agent-Based Modelling of Social and Spatial Processes, Kohler, T. A. and Gumerman, G. J., (Eds), 107-143. Oxford University Press, New York.

[19] Lees, M. Logan, B. and Theodoropoulos, G. "Using Access Patterns to Analyze the Performance of Optimistic Synchronization Algorithms in Simulations of MAS", Transactions of the Society for Computer Simulation International, Transactions of the Society for Computer Simulation International, Volume 84, Number 10/11, 481-492. ISSN**:** 0037-549

[20] Lees, M. Logan, B. and Theodoropoulos, G. "Analysing Probabilistically Constrained Optimism", Concurrency and Computation: Practice and Experience Journal, special issue on "Distributed Simulation, Virtual Environments and Real Time Applications", Vol 21, 11, 1467-1482, ISSN: 1532-0626 DOI: 10.1002/cpe.1397.

[21] Logan, B. and G. Theodoropoulos. 2001. The distributed simulation of multi-agent systems. Proceedings of the IEEE 89(2), 174–186.

[22] Mehl, H. and S. Hammes. 1993. Shared variables in distributed simulation. In Proceedings of the Seventh Workshop on Parallel and Distributed Simulation (PADS'93), pp. 68–75.

[23] NASA Jet Propulsion Laboratory, California Institute of Technology. http://www2.jpl.nasa.gov/srtm/

[24] Oguara, T. Chen, D. Theodoropoulos, G. Lees, M. and Logan, B. "An Adaptive Load Management Mechanism for Distributed Simulation of Multi-agent Systems", The 9-th IEEE International Symposium on Distributed Simulation and Real Time Applications (DSRT 2005), October 10-12, 2005 Montreal, Qc. Canada. ISBN: 0-7695-2462-1, Pages: 179-186

[25] Suryanarayanan, V. Minson, R. Theodoropoulos, G. "Synchronised Range Queries in Distributed Simulations of Multi-agent Systems", 13th IEEE International Symposium on Distributed Simulation and Real Time Applications (DS-RT 2009) October 2009, Singapore.

[26] Uhrmacher, A. M. and K. Gugler. 2000. Distributed, parallel simulation of multiple, deliberative agents. In Proceedings of the Fourteenth Workshop on Parallel and Distributed Simulation (PADS'2000), pp. 101–110, IEEE.

[27] Wilkinson, T. J. , Gibson, M., Christiansen, J. H., Widell, M., Schloen, D., Kouchoukos, N., Woods, C., Sanders, J., Simunich, K., Altaweel, M., Ur, J.A., Hritz, C., Lauinger, J., Paulette, T. & Tenney, J. 2007. Modeling Settlement Systems in a Dynamic Environment: Case Studies from Mesopotamia. In Model Based Archaeology. Kohler, T. & van der Leeuw, S. E. (Eds). School for Advanced Research Press, Santa Fe.

[28] Wooldridge, M. and N. R. Jennings. 1995. Intelligent agents: Theory and practice. Knowledge Engineering Review 10(2), 115–15