BaSi: Multi-Agent Based Simulation for Medieval Battles

Ambra Molesini    Enrico Denti    Andrea Omicini

Alma Mater Studiorum – Università di Bologna
{ambra.molesini, enrico.denti, andrea.omicini}@unibo.it

WOA 2011 – Dagli Oggetti agli Agenti
Progettazione ed analisi di sistemi complessi mediante modellazione e simulazione basate su agenti
Rende, CS, Italy
5th July 2011
1 Foreword

2 Background
   • Scenario
   • Tools

3 Development of BaSi

4 Conclusion & Future Work

5 Bibliography
Outline

1. Foreword

2. Background
   - Scenario
   - Tools

3. Development of BaSi

4. Conclusion & Future Work

5. Bibliography
Scenario

- Simulation aims at modelling and reproducing some natural, ethological, social or conceptual phenomena.
- Traditional simulation techniques often model the system dynamics based on systems of equations.
- This can hardly be made with complex systems, like for instance ecosystems and human communities, where many autonomous individuals interact continuously.
- Multi-Agent Based Simulation (MABS henceforth) promote a different approach to simulation, where interactive entities live, behave and interact in an agent society that represents the system to be modelled.
- In this context, emerging behaviours can be observed as the result of individual interactions and choices.
- The consequent relations emerging between individual behaviour and structural system properties help formulating/validating theories about ethological, sociological, psychological systems.
In this work, we experimented with MABS by taking *Medieval Battles* as our reference case study.

Why is it interesting?
- It is a non-numeric domain involving both quantitative and qualitative aspects.
- The domain features multiple roles and requires a clear mapping of their relationships.
- The scenario inherently emphasizes individual warrior/agent autonomy, while calling for a clear definition of social strategies and tactics.
- There is a widespread focus on interaction issues.
- Environment has a prominent role.
- The scenario promotes emergent behaviours.
- It is a good testbed from the methodological viewpoint.
- It is a good testbed from the infrastructural viewpoint.
Outline

1. Foreword

2. Background
   - Scenario
   - Tools

3. Development of BaSi

4. Conclusion & Future Work

5. Bibliography
Outline

1. Foreword

2. Background
   - Scenario
   - Tools

3. Development of BaSi

4. Conclusion & Future Work

5. Bibliography
Medieval Battles: The Roles

- During the Roman empire, the army was structured according to a rigid, hierarchical organisation, rooted on infantry.
- In the subsequent centuries, however, structure and discipline became gradually less relevant, in favor of individual qualities of soldiers.
- Battles occurred between groups of soldiers, with little or no coordination among them, often without a clear command chain.
- The coming of cavalry, from the V century, changed the scenario, confining infantry to a complementary role.
- Knights quickly become sorts of “human tanks”, virtually impossible to face in an open battlefield: the typical formation consisted of a single knight line, with squires at their back.
- Specialised troops – pikemen, mostly from the XII century – constituted a real danger for knights: wisely used, they could lead to the total destruction of the cavalry.
- In such cases, bowmen troops were used instead of cavalry, saving knights for a later time.
Medieval Battles: Emergent Behaviours

- Tactics were quite simple: troops layout were mostly standard, only seldom adapted to the conformation of the ground or other factors, so victory or defeat typically depended on the size of the army and, to some extent, individual qualities.
- Due to the lack of discipline, structure and order, even simple tactics could make the difference in medieval battles and often be enough to compensate even quite a large numeric disadvantage.
- On the other side, however, the lack of (global) coordination in the command chain, coupled with personal visibility goals of individual warriors, could easily vanify the tactic abilities of a commander.
- So, the final result of a battle was more an emergent behaviour coming out from a collection of individual choices and performance, rather than the expected result of a clearly-planned strategy.
Outline

1. Foreword

2. Background
   - Scenario
   - Tools

3. Development of BaSi

4. Conclusion & Future Work

5. Bibliography
SODA: Societies in Open and Distributed Agent spaces

SODA . . .

- is an agent-oriented methodology for the analysis and design of agent-based systems
- focuses on inter-agent issues, like the engineering of societies and environment for MAS [Omicini, 2001]
- adopts agents and artifacts – after the A&A meta-model – as the main building blocks for MAS development [Molesini et al., 2006a]
- introduces a simple layering principle in order to cope with the complexity of system description [Molesini et al., 2006b]
- adopts a tabular representation
SODA: An Overview

Requirements Analysis
- Requirements Tables
- Domain Tables
- Relations Tables

Analysis
- Responsibilities Tables
- Dependencies Tables
- Topologies Tables

References Tables

Transitions Tables

Design
- Architectural Design
  - Entities Tables
  - Interaction Tables
  - Constraints Tables
  - Topological Tables

Detailed Design
- Agent/Society Design Tables
- Environment Design Tables
- Interaction Design Tables
- Topological Design Tables

Mapping Tables

Molesini, Denti, Omicini (UniBo)
TuCSoN: Tuple Centres Spread Over the Network

TuCSoN . . .

... is an infrastructure for the coordination of distributed autonomous agents via *tuple centres* [Omicini and Zambonelli, 1999]

... agents access tuple centres associatively, by writing, reading, and consuming tuples via the TuCSoN coordination primitives

... tuple centre is a coordination abstraction perceived by agents as a standard tuple space [Gelernter and Carriero, 1992], whose behaviour can be defined to embed the laws of coordination via the ReSpecT language [Omicini, 2007].

... adopts *agents* and *artifacts* as the main building blocks for MAS development
Outline

1. Foreword
2. Background
   - Scenario
   - Tools
3. Development of BaSi
4. Conclusion & Future Work
5. Bibliography
Requirements

- Our goal is to realise a simulation system that enables users to:
  - define two armies
  - modify the default properties of the different kinds of soldiers
  - start the battle simulation
  - observe the battle progress and result

- Soldiers have to manifest autonomous behaviour, and the simulation has to evolve with no user intervention.

- The simulation ends when one of the two army is defeated—i.e., all of its soldiers are dead.

- Defining an army means to specify at least:
  - the number of soldiers, per type—i.e. the number of knights, bowmen and pikemen, respectively
  - the army commander
  - the army organisation—informal, two-levels, three-levels
The problem is decomposed in two main sub-problems:
- managing the simulation (army creation, battle management, etc.)
- managing the user-system interaction (capturing the user commands and graphically rendering the battle)

These sub-problems can be assigned to two sub-systems:
- the simulation sub-system
- the user-interface sub-system
to be designed independently from each other, while taking into proper account the information flow between them.

The user-interface sub-system (despite its potential graphical complexity) is simply concerned with getting initial parameters from the user and providing graphical results, and therefore does not require autonomous/intelligent entities; its design can then follow a standard object-oriented process, and will not be discussed in detail.
The simulation sub-system, instead, is well suited for an agent-oriented approach, as the agent abstractions seem particularly adequate to capture the key aspects of medieval battles:

- agents' autonomous and intelligent behaviour can easily model the soldiers: in turn, this makes it easy to model the army informal organisation, where each agent is able to decide its disposition in the battlefield and its actions during the fight.
- agent societies capture the social aspects of the army, such as its social goals (e.g. enemy destruction) and the social rules that govern its organisation: in particular, agent societies can map also the cases where the army organisation changes during the battle.
- the multiagent system (MAS henceforth) environment can capture both the topological aspect of the battlefield and the presence of the abandoned objects, which covers the last key aspect of the simulation sub-system.
Preliminary Requirements Analysis III

- As a preliminary step of the design process of the simulation sub-system, a soldier model has to be defined.

- Henceforth, we assume the following soldiers’ characterisation:
  - **Type**: soldier type (knight, bowman or pikeman)
  - **Army**: belonging army
  - **Speed**: maximum soldier speed in the battlefield
  - **Vigour**: maximum tolerable damage (before dying)
  - **Attack strength**: maximum damage inflicted to an enemy
  - **Defence strength**: maximum decrease of the damage caused by an enemy
  - **View scope**: maximum distance at which the soldier can see
  - **Attack scope**: maximum distance at which the soldier can attack
  - **Killing**: number of enemies killed when attacking
  - **Loading**: maximum load that the soldier can carry
  - **Equipment**: list of the soldier’s objects (weapons, food, money, etc.)
### SODA Requirements Analysis

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Army Definition</td>
<td>definition of the armies in terms of soldiers and their parameters</td>
</tr>
<tr>
<td>Simulation</td>
<td>management of the simulation</td>
</tr>
<tr>
<td>Monitoring</td>
<td>monitoring the battle progress and identification of the army winner</td>
</tr>
</tbody>
</table>
## SODA Analysis

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>State Area</td>
<td>providing the state of a specific battlefield area</td>
</tr>
<tr>
<td>Update Soldier State</td>
<td>updating soldier state</td>
</tr>
<tr>
<td>Battlefield State</td>
<td>providing battlefield state</td>
</tr>
<tr>
<td>Update Battlefield State</td>
<td>updating battlefield state</td>
</tr>
<tr>
<td>Rendering</td>
<td>showing the battlefield state changes</td>
</tr>
<tr>
<td>Command</td>
<td>obtaining user command</td>
</tr>
<tr>
<td>Ext Update Soldierd</td>
<td>obtaining user modifications to soldier parameters</td>
</tr>
</tbody>
</table>
### SODA Architectural Design

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attack Rule</td>
<td>The attack action is possible only if the enemy is in the scope of the attacker</td>
</tr>
<tr>
<td>Win Rule</td>
<td>An Army wins the battle only if the number the enemy army soldiers reaches zero</td>
</tr>
<tr>
<td>PickUp Rule</td>
<td>A soldier can pick up an abandoned object if this last is near to the soldier</td>
</tr>
<tr>
<td>Army Head Rule</td>
<td>The army head's command is the more priority command</td>
</tr>
<tr>
<td>Start Rule</td>
<td>The simulation starts only if the state of Interface Resource is start</td>
</tr>
<tr>
<td>Monitoring Battlefield Rule</td>
<td>The state of the BattleField Resource can change only if the simulation is started</td>
</tr>
</tbody>
</table>
SODA Detailed Design

<table>
<thead>
<tr>
<th>Artifact</th>
<th>Usage Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Army Artifact</td>
<td>receive_headCommand, send_headCommand, add_soldier, remove_soldier, set_strategy, get_strategy, get_position, set_position...</td>
</tr>
<tr>
<td>(social)</td>
<td></td>
</tr>
<tr>
<td>Simulation Artifact (social)</td>
<td>start_simulation, stop_simulation, pause_simulation...</td>
</tr>
<tr>
<td>Properties Artifact</td>
<td>set_property, get_property, add_soldierType, add_property, get_soldierTypes, get_armyTypes...</td>
</tr>
</tbody>
</table>

Molesini, Denti, Omicini (UniBo)
User-interface Sub-system: Overview
Overall BaSi Structure

- Simulation Facade
- User-interface Sub-System
- User-interface Artifact
- Simulation Artifact
- Army Artifact
- Simulation Society
- Army Society
The Prototype I
The Prototype II

- knight
- bowman
- pikeman
- commanders
- energy
The Prototype III

BaSi: an agent based Battle Simulation

Start | Stop | Pause

Aggiungi guerrieri a volontà:

- Esercito N°1: Vichinghi
- Esercito N°2: Sassoni
- CapoEsercito
- Cavaliere
- Arciere
- Picchiero
- CapoCavalieri
- CapoArcieri
- CapoPicchieri

OK | Cancel
The Prototype IV

<table>
<thead>
<tr>
<th>Tipo guerriero</th>
<th>speed</th>
<th>attack</th>
<th>defence</th>
<th>viewRange</th>
<th>attackRange</th>
<th>load</th>
<th>energy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cavaliere</td>
<td>5.0</td>
<td>50.0</td>
<td>12.0</td>
<td>120.0</td>
<td>30.0</td>
<td>32.0</td>
<td>820.0</td>
</tr>
<tr>
<td>Anciere</td>
<td>3.0</td>
<td>25.0</td>
<td>8.0</td>
<td>120.0</td>
<td>100.0</td>
<td>14.0</td>
<td>650.0</td>
</tr>
<tr>
<td>Picchiere</td>
<td>2.0</td>
<td>30.0</td>
<td>8.0</td>
<td>120.0</td>
<td>40.0</td>
<td>24.0</td>
<td>800.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tipo guerriero</th>
<th>av</th>
<th>ae</th>
<th>Be</th>
<th>tm</th>
<th>c</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cavaliere</td>
<td>2.5</td>
<td>2.5</td>
<td>12.5</td>
<td>12.5</td>
<td>8.0</td>
<td>11.0</td>
</tr>
<tr>
<td>Anciere</td>
<td>1.25</td>
<td>1.25</td>
<td>6.25</td>
<td>6.25</td>
<td>3.0</td>
<td>3.0</td>
</tr>
<tr>
<td>Picchiere</td>
<td>1.0</td>
<td>1.0</td>
<td>7.5</td>
<td>7.5</td>
<td>4.0</td>
<td>5.0</td>
</tr>
</tbody>
</table>
Outline

1 Foreword

2 Background
   • Scenario
   • Tools

3 Development of BaSi

4 Conclusion & Future Work

5 Bibliography
The main objective of this work was to investigate the suitability of MABS in the development of an articulated scenario such as medieval battles.

So, we deliberately left apart aspects that are normally relevant for a simulation system, such as a rigorous and efficient simulation engine, the realisation of both a complex graphical interface and a detailed animation of the graphical elements.

With respect to the simulation context discussed in this work, further work will be devoted to improve the prototype, improving the simulation engine and adopting a better graphical rendering engine.

More complex and intelligent behaviours for soldiers could also be added, as well as user functionalities for defining specific military strategies for each army.
The experiment turned out to be an interesting testbed for the SODA methodology, highlighting some benefits as well as some limitations that we plan to address in the near future:

- the tabular representation is clearly more suitable for an automatic tool than for a human designer, due to the large amount of tables to be filled in at each stage
- the definition – or the adoption – of a language for specifying SODA rules and interactions in a more precise and formal way, overcoming the implicit limitations of the natural language which is currently adopted in the tabular representation
- evaluating whether to enrich SODA with methods for the internal design of agents and artifacts
Outline

1. Foreword

2. Background
   • Scenario
   • Tools

3. Development of BaSi

4. Conclusion & Future Work

5. Bibliography


BaSi:
Multi-Agent Based Simulation for Medieval Battles

Ambra Molesini      Enrico Denti      Andrea Omicini

Alma Mater Studiorum – Università di Bologna
{ambra.molesini, enrico.denti, andrea.omicini}@unibo.it

WOA 2011 – Dagli Oggetti agli Agenti
Progettazione ed analisi di sistemi complessi mediante modellazione e simulazione basate su agenti
Rende, CS, Italy
5th July 2011