Multi-Agent Systems Simulation
An Introduction
Agent-Oriented Computing Course

Sara Montagna
sara.montagna@unibo.it

Ingegneria Due
Alma Mater Studiorum—Università di Bologna a Cesena

Scuola di Dottorato in Ingegneria e Scienze dell’Informazione
Outline

1 Simulation
   - Meaning, Motivation & Application

2 Type of Simulation
   - Continuous vs. Discrete Simulation
   - Deterministic vs. Stochastic
   - Micro, Macro and Multi-level Simulation

3 A Methodology
   - Domain, Design, Computational Model

4 Traditional Model and Simulation
   - Differential Equations: ODE, PDE, Master Equations
   - Computational Models

5 Agent-based Model and Multi-agent based Simulation
   - Why we need ABM?
   - What are ABM and MABS?
   - When to use ABM?
   - How to use ABM and MABS?
   - How to implement ABM?

6 A Case Study
   - The Morphogenesis of Biological Systems
   - Domain, Design & Computational Model
Simulation

1. Meaning, Motivation & Application
2. Type of Simulation
   - Continuous vs. Discrete Simulation
   - Deterministic vs. Stochastic
   - Micro, Macro and Multi-level Simulation
3. A Methodology
   - Domain, Design, Computational Model
4. Traditional Model and Simulation
   - Differential Equations: ODE, PDE, Master Equations
   - Computational Models
5. Agent-based Model and Multi-agent based Simulation
   - Why we need ABM?
   - What are ABM and MABS?
   - When to use ABM?
   - How to use ABM and MABS?
   - How to implement ABM?
6. A Case Study
   - The Morphogenesis of Biological Systems
   - Domain, Design & Computational Model
Scientific Method

Traditional science workflow

- Traditional scientific method
  - identification of the phenomena of interest
  - direct observation of the phenomena
  - formulation of theories / working hypothesis
  - reasoning on theories and phenomena through an empirical observation
  - quantitative analysis: measuring of phenomena in laboratory under controlled conditions
  - validation / invalidation of theories

[Parisi, 2001]
Definition of Simulation

- A new way for describing scientific theories

[Parisi, 2001]
- Simulation is the process with which we can study the dynamic evolution of a model system, usually through computational tools

[Banks, 1999]
- Simulation is the imitation of the operation of a real-world process or system over time
Simulation Requires a Model

M. Minsky – Models, Minds, Machines

A model (M) for a system (S), and an experiment (E) is anything to which E can be applied in order to answer questions about S.

- A model is a representation / abstraction of an actual system
- A model is a formalisation of aspects of a real process that aims to precisely and usefully describe that real process
- A model involves aggregation, simplification and omission
- The model implements theories which have to be verified during the simulation

Typical questions in model construction

- How complex should be the model?
- Which assumptions should be done?
Simulation Requires a Model

A model (M) for a system (S), and an experiment (E) is anything to which E can be applied in order to answer questions about S.

- A model is a representation / abstraction of an actual system
- A model is a formalisation of aspects of a real process that aims to precisely and usefully describe that real process
- A model involves aggregation, simplification and omission
- The model implements theories which have to be verified during the simulation

Typical questions in model construction

- How complex should be the model?
- Which assumptions should be done?
From Model to Simulation...

Computer simulation

- The models are designed to be run as processes within a computer
- Simulation creates a **virtual laboratory**
  - virtual phenomena observed under controlled conditions
  - possibility to easily modify the components of an experiment (variables, parameters, simulations’ part)
- Subsequent simulations imitate the operations of the modelled process
  - generation of an artificial evolution of the system
... and Back

- The observation of the evolution carries out deductions on the actual dynamics of the real system represented
- Simulation results make it possible to evaluate theories constructing the model

Model validation [Klugl and Norling, 2006]

- If the predicted and observed behaviour do not match, and the experimental data is considered reliable, the model must be revised
Why Do We Need Simulations?

- [Parisi, 2001], [Klugl and Norling, 2006]
- The simulated system cannot actually be observed
  - for either ethical or practical reasons
- The time scale of the real system is too small or too large for observation
- The original system is not existing anymore or not yet
- The system is complex
  - simple pattern of repeated individual action can lead to extremely complex overall behaviour
  - impossible to predict a-priori the evolution of the system
What Simulations Are Used For?

- Making prediction to be tested by experiments
- Exploring questions that are not amenable to experimental inquiry
- Obtaining a better understanding of some features of the system
  - verifying hypothesis and theories underlying the model that try to explain the systems behaviour
- Asking “what if” questions about real system
  - analysing the effects of manipulating experimental conditions without having to perform complex experiments
Applications of Simulation

- Interdisciplinary domain
- Complex Dynamical Systems
  - Brain
  - Social Systems
  - Ecosystems
  - Economic Systems
  - Coordinating Systems (swarm, flocking)
  - ...

→ systems too complex to be understood from observations and experiments alone

A multi-disciplinary research field
Maths, Physics, Informatics, Biology, Economy, Philosophy, ...
Features of Complex Systems in a Nutshell

- Nonlinear dynamics
- Presence of positive and negative feed-backs
- Ability of evolution and adaptation
- Robustness
- Self-organisation
- Hierarchical organisation
- Emergent phenomena which result from the interactions of individual entities → the whole is more than the sum of its parts
1. **Simulation**
   - Meaning, Motivation & Application

2. **Type of Simulation**
   - Continuous vs. Discrete Simulation
   - Deterministic vs. Stochastic
   - Micro, Macro and Multi-level Simulation

3. **A Methodology**
   - Domain, Design, Computational Model

4. **Traditional Model and Simulation**
   - Differential Equations: ODE, PDE, Master Equations
   - Computational Models

5. **Agent-based Model and Multi-agent based Simulation**
   - Why we need ABM?
   - What are ABM and MABS?
   - When to use ABM?
   - How to use ABM and MABS?
   - How to implement ABM?

6. **A Case Study**
   - The Morphogenesis of Biological Systems
   - Domain, Design & Computational Model
Continuous Simulation

- [Uhrmacher et al., 2005]
- The state variables change continuously with respect to time
- Series of infinite intervals and states

Main example of continuous simulation

- Time-changes described by a set of differential equations

Drawbacks

- The underlying assumption is that the system behaves continuously with an infinite number of close state transitions in each time interval
Continuous Simulation

- [Uhrmacher et al., 2005]
- The state variables change continuously with respect to time
- Series of infinite intervals and states

Main example of continuous simulation
- Time-changes described by a set of differential equations

Drawbacks
- The underlying assumption is that the system behaves continuously with an infinite number of close state transitions in each time interval
Discrete Simulation

- [Uhrmacher et al., 2005]
- Time evolves through discrete time steps
- The number of states is finite
- Synchronous or Asynchronous simulation update
  - synchronous — the state of all the components of the system is updated at the same time
  - asynchronous — the state of the system components is updated asynchronously following predefined rules which depends on the components themselves

Main example of discrete simulation

- Discrete time stepped approaches: time advances in equidistant steps *time-driven simulation*. Clock advances by one tick in every step and all the events scheduled at that time are simulated
- Discrete event approaches: *discrete event simulation*
Discrete Events Simulation – Event Driven Simulation

Algorithm of a discrete event simulation

- **Clock**: this variable holds the time up to which the physical system has been simulated.

- **Event list**: this is normally a data structure that maintains a set of messages, with their associated time of transmissions, that are scheduled for the future.

- At each step the message with the smallest associated future time is removed from the event list the event list and the corresponding message is simulated.

- The list of the events is updated:
  - adding new messages for future events
  - cancelling previously scheduled messages

- The clock is advanced to the time of the event just simulated.
Type of Simulation

- Deterministic vs. Stochastic
- Continuous vs. Discrete Simulation
- Micro, Macro and Multi-level Simulation

A Methodology

- Domain, Design, Computational Model

Traditional Model and Simulation

- Differential Equations: ODE, PDE, Master Equations
- Computational Models

Agent-based Model and Multi-agent based Simulation

- Why we need ABM?
- What are ABM and MABS?
- When to use ABM?
- How to use ABM and MABS?
- How to implement ABM?

A Case Study

- The Morphogenesis of Biological Systems
- Domain, Design & Computational Model
Deterministic vs. Stochastic

**Deterministic**
- The simulation evolves following deterministic laws

**Stochastic**
- The variables are probability distribution, or the laws to update the variables are stochastic laws
- Stochastic processes represent one means to express the uncertainty of our knowledge
- It is possible to compute just a probability distribution of the future histories, rather then a single outcome
1 Simulation
   - Meaning, Motivation & Application

2 Type of Simulation
   - Continuous vs. Discrete Simulation
   - Deterministic vs. Stochastic
   - Micro, Macro and Multi-level Simulation

3 A Methodology
   - Domain, Design, Computational Model

4 Traditional Model and Simulation
   - Differential Equations: ODE, PDE, Master Equations
   - Computational Models

5 Agent-based Model and Multi-agent based Simulation
   - Why we need ABM?
   - What are ABM and MABS?
   - When to use ABM?
   - How to use ABM and MABS?
   - How to implement ABM?

6 A Case Study
   - The Morphogenesis of Biological Systems
   - Domain, Design & Computational Model
Granularity of Simulation Elements: Macro-simulation

- [Uhrmacher et al., 2005]
- The macro model describes the system as one entity
- The model attempts to simulate changes in the averaged characteristics of the whole population
- Variables and their interdependencies, which can be expressed as rules, equations, constraints... are attributed to this entity
- Modelling, simulating and observation happens on one level: the global level
- The characteristic of a population are averaged together
Granularity of Simulation Elements: Micro-simulation

- The micro model describes the system as a set of entities
  - smaller entities with distinct state and behaviour
  - the system is thought as comprising a numbers of entities
  - the system entities interact with each other
- The micro level models the behaviour of the individuals
- The macro level
  - exists only as it aggregates results of the activities at micro level
  - is used for reflecting emergent phenomena
Granularity of Simulation Elements: Multi-level Simulation

- It is an intermediate form
- The multi-level model describes a system at least at two different levels
- Interactions are taking place within and between the different levels
- The system is described at different time scales

Advantages of Multi-level simulation

- It facilitates taking spatial and temporal structured processes into consideration
- It model the complex systems hierarchical organisation
- It allows the description of upward and downward causation
Down-ward and Up-Word Causation

The whole is to some degree constrained by the parts (upward causation), but at the same time the parts are to some degree constrained by the whole (downward causation).\(^a\)

How To Choose Between Different Approaches

Which kind of simulation?

- Modelling and simulating approaches are chosen on demand and thus address the diverse needs of modelling and simulation of the systems.
- Multi-level simulation is considered the most suitable approach for studying complex systems.
Simulation

1. Meaning, Motivation & Application

2. Type of Simulation
   - Continuous vs. Discrete Simulation
   - Deterministic vs. Stochastic
   - Micro, Macro and Multi-level Simulation

3. A Methodology
   - Domain, Design, Computational Model

4. Traditional Model and Simulation
   - Differential Equations: ODE, PDE, Master Equations
   - Computational Models

5. Agent-based Model and Multi-agent based Simulation
   - Why we need ABM?
   - What are ABM and MABS?
   - When to use ABM?
   - How to use ABM and MABS?
   - How to implement ABM?

6. A Case Study
   - The Morphogenesis of Biological Systems
   - Domain, Design & Computational Model
## Simulation Workflow

### Main steps in a simulation study

- [Klugl and Norling, 2006]
- Starting with a real system analysis
  - understanding its characteristics
- Building a model from the real system
  - retaining aspects relevant to simulation
  - discarding aspects irrelevant to simulation
- Constructing a simulation of the model that can be executed on a computer
- Analysing simulation outputs
  - model validation and verification
How to Build a Model: Methodology

Model design

- **Concept model phase** – *Domain model*
  - Analysis of the real system characteristic

- **Specification phase** – *Design model*
  - translation of the information from the needs’ into a formal model
  - aim: build a model independent of any tool and any software platform

- **Implementation phase** – *Computational model*
  - translation of the model resulting from the design on a particular software platform
Simulation Execution and Validation: Methodology

Experimentation phase – Simulation design

- Specifying the simulation goals
- Identifying of the information needed to the simulation
- Identifying useful experiments.
- Planning a list of experiments
- Performing the experiments

Validation and Verification

- Analyse simulation results
  - Comparing the behaviours of computer-executable models with experimental observation
  → Is the model suitable or not?
- Eventually improving the model
Simulation Execution and Validation: Methodology

Experimentation phase – Simulation design

- Specifying the simulation goals
- Identifying of the information needed to the simulation
- Identifying useful experiments.
- Planning a list of experiments
- Performing the experiments

Validation and Verification

- Analyse simulation results
  - Comparing the behaviours of computer-executable models with experimental observation
  → Is the model suitable or not?
- Eventually improving the model
Simulation
- Meaning, Motivation & Application

Type of Simulation
- Continuous vs. Discrete Simulation
- Deterministic vs. Stochastic
- Micro, Macro and Multi-level Simulation

A Methodology
- Domain, Design, Computational Model

Traditional Model and Simulation
- Differential Equations: ODE, PDE, Master Equations
- Computational Models

Agent-based Model and Multi-agent based Simulation
- Why we need ABM?
- What are ABM and MABS?
- When to use ABM?
- How to use ABM and MABS?
- How to implement ABM?

A Case Study
- The Morphogenesis of Biological Systems
- Domain, Design & Computational Model
Differential Equations

- System described by a set of state variables
- Different types of differential equations:
  - ODE: how do they vary in time
  - PDE: how do they vary in time and space
  - SDE: which is the probability that the variable has a certain value
- Time-dependent variables are assigned to different measuring or not-measurable quantities of the system
- The continuous state changes are modelled by a sum of rates describing the increase and decrease of quantities amounts.

Features

- Continuous model
- Deterministic or Stochastic Model
- Macro model
An example of ODE

- The state variable is referenced as $X_i$ which is a macroscopic collective variable.
- The collection of values of all these state variables $\{X_1, X_2, \ldots, X_n\}$ denote a complete set of variables to define the *instantaneous state* of the system $\mathbf{X}$.
- The time evolution of $X_i(t)$ will take the form, through a mathematical expression (ODE):

\[
\frac{dX_i}{dt} = F_i(X_1, X_2, \ldots, X_n; \gamma_1, \gamma_2, \ldots, \gamma_m)
\]

where:

- $F_i$ may be a complex function of the state variables: the structure of the function $F_i$ will depend in a very specific way on the system considered.
- $\gamma_1, \gamma_2, \ldots, \gamma_m$, are the parameters of the problem (*control parameters*).
Simulation of Differential Equations Models

Analytical solution of differential equations

- Exact solution of a class of differential equations
- It is possible under very special circumstances
  - i.e. when the function $F_i$ is linear
- Example of analytic solution:
  - the solution of a set of ODEs in terms of exponential functions, $\exp(\lambda_i t)$, and harmonic functions, $\sin(\omega_i t + \phi_i)$
Simulation of Differential Equations Models

Numerical solution of differential equations

- Also called *numerical integration*
- The exact solution of the equations is approximated by calculating approximate values \( \{X_1, X_2, ..., X_n\} \) for \( \mathbf{X} \)
- Time step is reduced to arbitrary small discrete intervals: values at consecutive time-points \( t_0, t_1, ..., t_m \)
- It uses different numerical algorithms:
  - Euler’s method for ODEs
  - Taylor series method for ODEs
  - Runge-Kutta method
  - Runge-Kutta-Fehlberg method
  - Adams-Bashforth-Moulton method
  - Finite Difference method for PDEs
  - ...
Simulation of Differential Equations Models

Qualitative solution of differential equations

- It answers qualitative questions such as:
  - what will the system do for $t \to \infty$?
  - under which condition the system is stable?

- Definition of system attractors
  - equilibrium points
  - limit cycles
  - strange attractors

- Bifurcation analysis
  - how the system’s dynamic (solution) changes under the change of its parameters
Modelling a Complex System

To remind you...

- Important features of a complex systems
  - systems that draw their dynamics from flexible local interactions
  - systems where individuality and/or locality is important
  - systems with a strong hierarchical organisation
  - emergent Phenomena and Self-Organising systems
  - down-ward and up-ward systems dynamics

- Remind them if you wish to model a complex system

- They are important for analysing and choosing modelling approaches and tools
Analysis of Differential Equations I

Advantages of ODE and PDE

- They are a really well understood and established framework
- They are relatively simple
- They have a strong formal aspect

- Where do differential equations fail?
Are they able to capture complex systems features?

Tod-down approaches – Macro model

- The model is built upon the imposition of global laws
- The model loses the representation of the actors of the system
- Focusing only on the population, the model loses the representation of the individual and of its locality
- The model doesn’t allow the study of global dynamics as emergent phenomena from local interaction
- The model ignores the local processes performed by low-level components
- A particular entity is no longer accessible
Simulation
- Meaning, Motivation & Application

Type of Simulation
- Continuous vs. Discrete Simulation
- Deterministic vs. Stochastic
- Micro, Macro and Multi-level Simulation

A Methodology
- Domain, Design, Computational Model

Traditional Model and Simulation
- Differential Equations: ODE, PDE, Master Equations

Computational Models

Agent-based Model and Multi-agent based Simulation
- Why we need ABM?
- What are ABM and MABS?
- When to use ABM?
- How to use ABM and MABS?
- How to implement ABM?

A Case Study
- The Morphogenesis of Biological Systems
  - Domain, Design & Computational Model
Computational Models

- Graphs and Networks
  - boolean networks
- Petri-Nets
- Stochastic-\(\pi\)-calculus
- Cellular Automata
- Agent-based Model
Agent-based Model and Multi-agent based Simulation

1 Simulation
   - Meaning, Motivation & Application

2 Type of Simulation
   - Continuous vs. Discrete Simulation
   - Deterministic vs. Stochastic
   - Micro, Macro and Multi-level Simulation

3 A Methodology
   - Domain, Design, Computational Model

4 Traditional Model and Simulation
   - Differential Equations: ODE, PDE, Master Equations
   - Computational Models

5 Agent-based Model and Multi-agent based Simulation
   - Why we need ABM?
   - What are ABM and MABS?
   - When to use ABM?
   - How to use ABM and MABS?
   - How to implement ABM?

6 A Case Study
   - The Morphogenesis of Biological Systems
   - Domain, Design & Computational Model
Why we need agent-based model?

Need to...\(^a\)

\(^a\)[Sun and Cheng, 2000, Bonabeau, 2002, Macal and North, 2006]

1. model the individual properties that cannot be fully taken into account in the state variables of the model
2. understand how individual properties determine the system’s level properties
3. capture the hierarchical organisation of complex systems
4. explore the role of the environment
Agent-based Model and Multi-agent based Simulation

What are ABM and MABS?

Simulation
- Meaning, Motivation & Application

Type of Simulation
- Continuous vs. Discrete Simulation
- Deterministic vs. Stochastic
- Micro, Macro and Multi-level Simulation

A Methodology
- Domain, Design, Computational Model

Traditional Model and Simulation
- Differential Equations: ODE, PDE, Master Equations
- Computational Models

Agent-based Model and Multi-agent based Simulation
- Why we need ABM?
- What are ABM and MABS?
- When to use ABM?
- How to use ABM and MABS?
- How to implement ABM?

A Case Study
- The Morphogenesis of Biological Systems
- Domain, Design & Computational Model
What is Agent-based Model

Agent-based model is a specific individual-based computational model for studying macro emergent phenomena through the definition of the system micro level which is modelled as a collection of interacting entities.

- MAS provides designers and developers with...
  - Agents
    ...a way of structuring a model around autonomous, heterogeneous, communicative, possibly adaptive, intelligent, mobile and...entities
  - Society
    ...a way of representing a group of entities whose behaviour emerges from the interaction among elements
  - Environment
    ...a way of modelling an environment characterised by a topology and complex internal dynamics
- MAS gives methods to...
  - model individual structures and behaviours of different entities
  - model local interactions among entities and entities-environment
  - model the environment structures and dynamics
What is Multi-agent Based Simulation

Execute an ABM

- Running an ABM
- Study its evolution
  - observing individual and environment evolution
  - observing global system properties as emergent properties from the system’s constituent units interactions (from the bottom-up)
  - making in-silico experiment
Agent-based Model and Multi-agent based Simulation

When to use ABM?

1. Simulation
   - Meaning, Motivation & Application

2. Type of Simulation
   - Continuous vs. Discrete Simulation
   - Deterministic vs. Stochastic
   - Micro, Macro and Multi-level Simulation

3. A Methodology
   - Domain, Design, Computational Model

4. Traditional Model and Simulation
   - Differential Equations: ODE, PDE, Master Equations
   - Computational Models

5. Agent-based Model and Multi-agent based Simulation
   - Why we need ABM?
   - What are ABM and MABS?
   - When to use ABM?
   - How to use ABM and MABS?
   - How to implement ABM?

6. A Case Study
   - The Morphogenesis of Biological Systems
   - Domain, Design & Computational Model
When to Use Agent-based Model

- When there are decisions and behaviours that can be defined discretely (with boundaries)
- When the individual behaviour
  - is non-linear
  - can be characterised by thresholds or if-then rules
  - exhibits memory and path-dependence or even adaptation and learning capabilities
- When interactions with flexible individual participants have to be represented
- When in-homogeneous space is relevant
- When the topology of the interactions is heterogeneous
- When the system consists in mutable interacting participants
  - agents can be erased
  - new agents can enter in the scenario
- When averages will not work
Advantages and Problems of Agent-based Model

- **Advantages**
  1. It captures emergent phenomena
  2. It provides a natural description of a system
     → it makes the model seem closer to reality
  3. It is flexible

- **Problems**
  1. There is a lack of
     - an unified formal framework for unambiguously representing ABM elements and rules
     - a widely accepted methodology for developing MABS
  2. It increases the amounts of parameters
  3. Software development remains a significant barrier to the use of ABM
     - inconsistence and incongruence between agents of the conceptual model and agents of the computational model
1 Simulation
   - Meaning, Motivation & Application

2 Type of Simulation
   - Continuous vs. Discrete Simulation
   - Deterministic vs. Stochastic
   - Micro, Macro and Multi-level Simulation

3 A Methodology
   - Domain, Design, Computational Model

4 Traditional Model and Simulation
   - Differential Equations: ODE, PDE, Master Equations
   - Computational Models

5 Agent-based Model and Multi-agent based Simulation
   - Why we need ABM?
   - What are ABM and MABS?
   - When to use ABM?
   - How to use ABM and MABS?
   - How to implement ABM?

6 A Case Study
   - The Morphogenesis of Biological Systems
   - Domain, Design & Computational Model
The Methodology in a Figure
Defining the agents of an ABM

Identifying agents, accurately specifying their behaviours, and appropriately representing agent interactions are the keys for developing useful ABM.

An Agent in ABM requires mechanisms for

- Receiving the input e.g., through sensors
- Storing history, e.g., through a state
- Devising next action, e.g., through decision rules which define the
  - reactive behaviour: how an agent reacts to external stimuli
  - proactive behaviour: how an agent behaves in order to reach its goals/tasks
- Carrying out the action e.g., through effectors

Agents might also have capability of learning
  → processing the ability of adapting to changing environment
Agent-based Model and Multi-agent based Simulation

1. Simulation
   - Meaning, Motivation & Application

2. Type of Simulation
   - Continuous vs. Discrete Simulation
   - Deterministic vs. Stochastic
   - Micro, Macro and Multi-level Simulation

3. A Methodology
   - Domain, Design, Computational Model

4. Traditional Model and Simulation
   - Differential Equations: ODE, PDE, Master Equations
   - Computational Models

5. Agent-based Model and Multi-agent based Simulation
   - Why we need ABM?
   - What are ABM and MABS?
   - When to use ABM?
   - How to use ABM and MABS?
   - How to implement ABM?

6. A Case Study
   - The Morphogenesis of Biological Systems
   - Domain, Design & Computational Model
Multi-agent based simulation Platforms

Agent-based toolkits providing frameworks and libraries that simplify the procedures of establishing models and performing simulations

Standard issues

- Model structure, i.e. agent behaviour, agent communication mechanisms, environment and topology
- Discrete event simulation
  - **Scheduling**: to control which specific actions are executed and when (in simulated time)
  - **Marsenne Twister**: random number generation
- Distributed simulation
- Facilities for storing and displaying the simulation state

Most of the agent-based simulation platforms are based on the object-oriented paradigm
Swarm

Swarm\textsuperscript{a}

\begin{itemize}
\item Objectives
  \begin{itemize}
  \item to ensure a widespread use across scientific domains
  \item to implement a model
  \item to provide a virtual laboratory for observing and conducting experiments
  \end{itemize}
\item Swarm is implemented in Objective-C
\end{itemize}

\textsuperscript{a}http://www.swarm.org/

Java Swarm

\begin{itemize}
\item Set of Java classes that allow the use of Swarm’s Objective-C libraries from Java
\end{itemize}
Repast

Objectives
- to implement Swarm in Java
- to support the specific domain of social science (it includes specific tools to that domain)
- to make it easier for inexperienced users to build models

free and open source
MASON

Objectives

- models with many agents executed over many iterations
- to maximize execution speed
- to assure complete reproducibility across hardware
- to detach or attach graphical interfaces
- to be not domain specific

Basic capabilities for graphing and random number distributions

http://cs.gmu.edu/ eclab/projects/mason/
Limitations of these Platforms

- Difficulty of use
- Insufficient tools for building models
- Lack of tools for documenting and communicating software
- Incoherence between the design model and the computational model
  - computational agents ≠ conceptual agents
  - no first class abstraction for modelling the environment
NetLogo

http://ccl.northwestern.edu/netlogo/

The Logo family of platforms has followed a different evolution

- Objectives
  - to be ease of use
- Educational tool
- NetLogo is recommended for models
  - with short-term, local, interactions of agents
  - base on grid environment
  - not extremely complex
- Useful for prototyping models (quickly) and exploring design decisions
- Provided by an own programming language
  - high level structures and primitives
  - all code in the same file

Still miss the coherence between design and computational model
More details on Repast Simphony

- An optional point-and-click model development environment that generates Java classes
- A point-and-click model execution environment that includes
  - built-in results logging
  - graphing tools
  - automated connections to a variety of external tools including R, VisAD, Weka, popular spreadsheets, MATLAB, and iReport
- A flexible definition of space for modelling and visualise 2D and 3D environments
- A fully concurrent multithreaded discrete event scheduler
- Libraries for genetic algorithms, neural networks, regression, random number generation, and specialized mathematics
- Automated Monte Carlo simulation framework
Creating a Repast Model with the Visual Agent Editor

- for more details see the on-line tutorial

1 http://repast.sourceforge.net/download.html
Creating a Repast Model with Java Objects

1. **ContextBuilder** defines the main components of the system
   - the environment (with the number and dimension of the grids) where the agents can move
   - the type and initial number of agents that populate the environment

2. **Agent Classes** such as **SimpleAgent**
   - the core method is `step` which precedes methods to be scheduled
   - it has several options: start time, the updated interval
   - it is override by the subclasses
A Case Study: The Morphogenesis of Biological Systems

1. Simulation
   - Meaning, Motivation & Application

2. Type of Simulation
   - Continuous vs. Discrete Simulation
   - Deterministic vs. Stochastic
   - Micro, Macro and Multi-level Simulation

3. A Methodology
   - Domain, Design, Computational Model

4. Traditional Model and Simulation
   - Differential Equations: ODE, PDE, Master Equations
   - Computational Models

5. Agent-based Model and Multi-agent based Simulation
   - Why we need ABM?
   - What are ABM and MABS?
   - When to use ABM?
   - How to use ABM and MABS?
   - How to implement ABM?

6. A Case Study
   - The Morphogenesis of Biological Systems
   - Domain, Design & Computational Model
On the Morphogenesis of Living Systems

*Developmental Biology researches the mechanisms of development, differentiation, and growth in animals and plants at the molecular, cellular, and genetic levels.*

<table>
<thead>
<tr>
<th>Animal developmental steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Fertilisation of one egg</td>
</tr>
<tr>
<td>2. Mitotic division</td>
</tr>
<tr>
<td>3. Cellular differentiation</td>
</tr>
<tr>
<td>- diverse gene expression</td>
</tr>
<tr>
<td>4. <strong>Morphogenesis</strong></td>
</tr>
<tr>
<td>- control of the organised spatial distribution of the cell diversity</td>
</tr>
</tbody>
</table>
Each region of the developing organism expresses a given set of genes

- Developmental Biology recognise as important actors in the emergence of embryonic patterning – self-organised structures
  - transcriptional control mechanisms
  - signalling pathways
  - cell-to-cell direct interaction
  - short and long range signals (*morphogenes*)

→ interplay between cells internal activity and cell-to-cell interactions
Simulation

1. Meaning, Motivation & Application
2. Type of Simulation
   - Continuous vs. Discrete Simulation
   - Deterministic vs. Stochastic
   - Micro, Macro and Multi-level Simulation
3. A Methodology
   - Domain, Design, Computational Model
4. Traditional Model and Simulation
   - Differential Equations: ODE, PDE, Master Equations
   - Computational Models
5. Agent-based Model and Multi-agent based Simulation
   - Why we need ABM?
   - What are ABM and MABS?
   - When to use ABM?
   - How to use ABM and MABS?
   - How to implement ABM?
6. A Case Study
   - The Morphogenesis of Biological Systems
   - Domain, Design & Computational Model

Montagna (Università di Bologna)
**Goal of the Model**

- Reproducing the cell diversity regionalisation in **three layers**

<table>
<thead>
<tr>
<th>Layer</th>
<th>Type</th>
<th>mRNA Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>Endoderm</td>
<td>Type A</td>
<td>mRNA1</td>
</tr>
<tr>
<td>Mesoderm</td>
<td>Type B</td>
<td>mRNA2</td>
</tr>
<tr>
<td>Ectoderm</td>
<td>Type C</td>
<td>mRNA3</td>
</tr>
</tbody>
</table>
Domain Model

- Cells are the individual entities of the model
- The extra-cellular matrix is the environment in the model
  - the environment is a 2D grid
  - each location contains food and cues
- The entity’s state models
  - the mRNA concentration (mRNA1, mRNA2, mRNA3)
  - the cell dimension
- Entity behaviour
  - Eat the food in its environment location
  - Divide when its dimension is up to P
  - Differentiate into type A, B or C
  - Move randomly if the cell is not differentiated
  - Migrate if the cell is differentiated
Design Model: Eating, Dividing & Differentiating
Design Model: Moving

Stato Iniziale

Stato Attesa

Movimento Casuale

Stato Cellula Specializzata

Foglietto raggiunto

Movimento Deterministico

Stato Finale
Implementation Model

1. AgentStyle2D
2. AgentStyle3D
3. DeutoplasmaStyle2D
4. DeutoplasmaStyle3D → instructions for the graphics visualisation of the agents
5. DeutoplasmaCel → represents a location of the grid
6. Cell → is the agent
7. SimulationSuiteContextCreator → creates the context
Simulation
Introduction to simulation. 

Agent-based modeling: Methods and techniques for simulating human systems. 
PNAS, 99(suppl. 3):7280–7287.
Bibliography II

An automatic quantification and registration strategy to create a gene expression atlas of zebrafish embryogenesis.
In 31st Annual International Conference of the IEEE EMBS, pages 1469–1472, Minneapolis, Minnesota, USA.

Agent-based simulation: Social science simulation and beyond.
Bibliography III


Bibliography IV


Multi-Agent Systems Simulation
An Introduction
Agent-Oriented Computing Course

Sara Montagna
sara.montagna@unibo.it

Ingegneria Due
Alma Mater Studiorum—Università di Bologna a Cesena

Scuola di Dottorato in Ingegneria e Scienze dell’Informazione