Simulation & Multi-Agent Systems
An Introduction

Multiagent Systems LS
Sistemi Multiagente LS

Andrea Omicini & Sara Montagna
{andrea.omicini, sara.montagna}@unibo.it

Ingegneria Due
Alma Mater Studiorum—Università di Bologna a Cesena

Academic Year 2007/2008

1. Simulation
   - Meaning
   - Motivation
   - Application
2. Type of Simulation
   - Continue vs. Discrete Simulation
   - Deterministic vs. Stochastic
   - Micro, Macro and Multi-level Simulation
3. A Methodology
   - Domain, Design, Computational Model
4. Traditional Model and Simulation
   - Graphs and Networks
   - Differential Equations: ODE, PDE, Master Equations
   - Critical Analysis
5. Computational Model
   - Agent Based Model and Multi-agent based Simulation
   - Multi-agent based simulation Platforms

Outline

Scientific Method

Traditional science workflow

- [Parisi, 2001]
- 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

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

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?

From Model to Simulation. . .

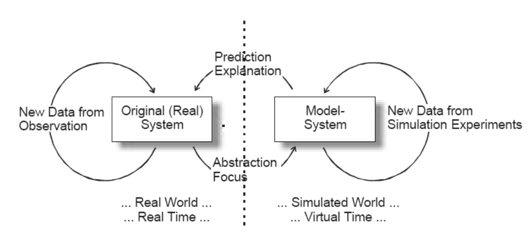
Computer simulation

- The models are designed to be run as processes within a computer
- The computational model simulates the processes as they are thought to exist in the real system
- Subsequent simulations imitate the operations of the modelled process
  - generation of an artificial evolution of the system
- 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

. . . and Back

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



Simulation Creates a Virtual Laboratory

- A virtual laboratory makes it possible to perform experiments
  - virtual phenomena observed under controlled conditions
  - possibility to easily modify the components of an experiment (variables, parameters, simulations' part)
  - tools to make predictions on theories
  - tools to make inferences on simulation results

Outline

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 (look into the future) to be tested by experiments
- Exploring questions that are not amenable to experimental inquiry
- Obtaining a better understanding of some features of the system
- Describing and analysing the behaviour of a system, asking "what if" questions about real system
  - rapidly analysing the effects of manipulating experimental conditions without having to perform complex experiments
- Potentially assisting in discovery and formalisation
- Verifying hypothesis and theories underlying the model that try to explain the systems behaviour

A Science for Simulation

Popper (1972)

Popper proposes as a logical necessity that scientific theories can only be refuted

- no amount of supporting experimental evidence constitutes proof of a theory
- yet a single repeatable piece of counter-evidence can require that the theory is developed or replaced

Scientific hypothesis should consist only of refutable statements

Simulation can be used as a tool to validate, or better, potentially refute formal models

A Science for Simulation

Popper (1972)

Popper proposes as a logical necessity that scientific theories can only be refuted

- no amount of supporting experimental evidence constitutes proof of a theory
- yet a single repeatable piece of counter-evidence can require that the theory is developed or replaced

Scientific hypothesis should consist only of refutable statements

Simulation can be used as a tool to validate, or better, potentially refute formal models

Outline

Applications of Simulation

Main applicative domains

- Interdisciplinary domain
- Complex Dynamical Systems
  - systems too complex to be understood from observations and experiments alone
- Predicting changes
- Observing systems evolution

A Brief Introduction to Complex Systems

Systems as

- Brain
- Social Systems
- Ecosystems
- Economic Systems
- Coordinating Systems (swarm, flocking)
- ...

... are recognised as complex systems

A Brief Introduction to Complex Systems

A multi-disciplinary research field

- Maths
- Physics
- Informatics
- Biology
- Economy
- Philosophy
- ...

Features of Complex Systems in a Nutshell

In general, complex systems are observed to feature

- Presence of different elements that interact
- Nonlinear dynamics
- Presence of positive and negative feed-backs
- Ability of evolution and adaptation
- Robustness
- Self-organisation

Complex Systems ask for Holistic Approach: Simulation

- Reductionism
  - belief that the behavior of a whole or system is completely determined by the behavior of its parts
  - if the laws governing the behavior of the parts are known, one should be able to deduce the laws governing the behavior of the whole.
- Holism – Systems theory
  - anti-reductionist stance: the whole is more than the sum of the parts
  - the whole has "emergent properties" which cannot be reduced to properties of the parts

Outline

Continue Simulation

- [Uhrmacher et al., 2005]
- The variables of the system change continuously during time
- Series of infinite intervals and states

Main example of continue simulation

- Time-changes described by a set of differential equations

Critical Analysis of Continuos Simulation

Benefits

- Perfectly suited for the reproduction of measured time-dependent trajectories
- Easily fitting of the parameters

Drawbacks

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

Critical Analysis of Continuos Simulation

Benefits

- Perfectly suited for the reproduction of measured time-dependent trajectories
- Easily fitting of the parameters

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
  - canceling previously scheduled messages
- the clock is advanced to the time of the event just simulated.

Critical Analysis of Discrete Models

Benefits

- No continuity of behaviour needs to be assumed

Drawbacks

- Sequential Simulation
  - in each cycle of simulation only one item is removed from the event list, its effects simulated and the event list, possibly, updated.
  - the algorithm cannot be readily adapted for concurrent execution on a number of processors, since the list cannot be effectively partitioned for such execution.

Critical Analysis of Discrete Models

Benefits

- No continuity of behaviour needs to be assumed

Drawbacks

- Sequential Simulation
  - in each cycle of simulation only one item is removed from the event list, its effects simulated and the event list, possibly, updated.
  - the algorithm cannot be readily adapted for concurrent execution on a number of processors, since the list cannot be effectively partitioned for such execution.

Outline

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

Outline

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

Main example of macro-simulation

- Differential equations

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 huge numbers of entities
- 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

Main example of micro-simulation

- Cellular automata

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

Main example of multi-level simulation

- Multi-agent systems

Advantages of Multi-level simulation

- It facilitates taking spatial and temporal structured processes into consideration
- It allows the description of upward and downward causation

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

Main example of multi-level simulation

- Multi-agent systems

Advantages of Multi-level simulation

- It facilitates taking spatial and temporal structured processes into consideration
- 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]

[a] F. Heylighen. http://pespmc1.vub.ac.be/DOWNCAUS.html

How To Choose Between Different Approaches

Which kind of simulation?

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

Outline

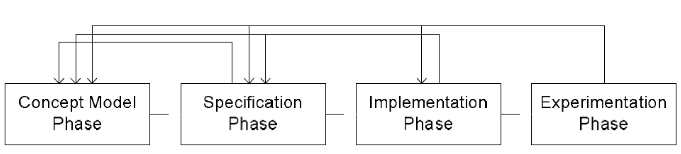
Simulation Workflow

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

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

How to Perform a Simulation: Methodology

Experimentation phase – Simulation design

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

Validation and Verification

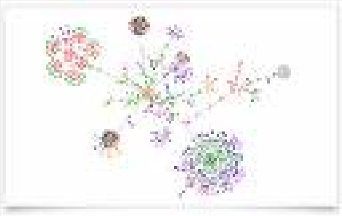
- Analyse simulation results
  - Detailed behaviours of computer-executable models are first compared with experimental observation
  - Comparing the predictions with the observed experimental data gives an indication of the adequacy of the model

Outline

Graphs

Graphs as models of system network's structure

- Static representation of pairwise relations between objects of the system
  - nodes or vertices: entities of the system
  - edges that connect pairs of vertices: interaction between entities



Features

- Static model
- Micro-model

Graph Theory

Topology of a graph

- Analysis of structural properties of a network

Topological features of an N-nodes network

- The **degree** of a node i is the number of its connections (or neighbors), k_i
- The **average degree** of a network is

$$\langle k \langle = \frac{1}{N} \sum_i k_i$$

- The **degree distribution** function $P(k)$ which measures the proportion of nodes in the network having degree k

Graph Theory

Topology of a graph

- Analysis of structural properties of a network

Topological features of an N-nodes network

- The **degree** of a node i is the number of its connections (or neighbors), k_i
- The **average degree** of a network is

$$\langle k \langle = \frac{1}{N} \sum_i k_i$$

- The **degree distribution** function $P(k)$ which measures the proportion of nodes in the network having degree k

Graph Types

Random graph

- The vertices typically have $\langle k \rangle$ edges and the vertices having significantly more or less edges than $\langle k \rangle$ are extremely rare

Scale-free graph

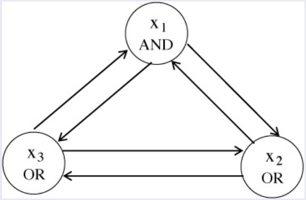
- These types of graphs are inhomogeneous, in that most of the vertices have few edges, whereas some vertices, called (*hubs*), have many edges

Hierarchical graphs

- These types of graphs describes modular networks, i.e. they are formed by the repetition of nodes' cluster.

Boolean Networks

- Introduced by Stuart Kaufmann as a model of gene regulation networks
- Directed graph with N nodes
- Nodes ↔ Boolean Function
- Node state: value of the boolean function (binary state)

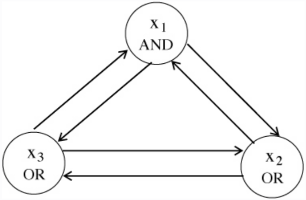


Features

- Discrete synchronous model
- Deterministic model
- Micro-model

Boolean Networks

- Introduced by Stuart Kaufmann as a model of gene regulation networks
- Directed graph with N nodes
- Nodes $\leftrightarrow$ Boolean Function
- Node state: value of the boolean function (binary state)



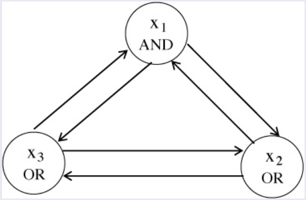
Features

- Discrete synchronous model
- Deterministic model
- Micro-model

Boolean Network Dynamic

An example

t			$t+1$		
x_1	x_2	x_3	x_1	x_2	x_3
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	1	1
1	0	0	0	1	1
1	0	1	0	1	1
1	1	0	0	1	1
1	1	1	1	1	1



Outline

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

- Continuos 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 $\{X_1, X_2, ..., X_n\}$ 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):

$$\frac{dX_i}{dt} = F_i(X_1, X_2, ..., X_n; \gamma_1, \gamma_2, ..., \gamma_m)$$

- 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
  - $\gamma_1, \gamma_2, ..., \gamma_m$, 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 answer qualitative questions such as:
  - what will the system do for $t \longrightarrow \infty$
  - 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

Outline

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-organizing 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?

Analysis of Differential Equations II

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

Analysis of Networks

Are they able to capture complex systems features?

Bottom up approaches – Micro model

- The model is built upon the identification of systems entities and of the interactions between them
- The model does not allow the representation of autonomous behaviour of the components
- The behaviour of the entire system dynamically emerges from the interactions between its parts

Outline

What is ABM

MAS provide designers and developers with...

- Agents
  ...a way of structuring a model around autonomous and communicative etities
- Society
  ...a way of representing a group of entities whose behaviour emerges from the interaction among elements
- Environment
  ...a way of modelling the environment

MAS give methods to. . .

- Model individual structures and behaviours of different entities
- Model local interactions between entities
- Model the environment structures and dynamics

An Agent in ABM

Properties of agents in ABM

- Autonomous
- Heterogeneous
- Articulated internal structure
- Possibly adaptive, intelligent, mobile, . . .

An Agent in ABM

Defining the agents of an ABM

- Sensors & effectors
- Internal autonomous behaviour
  - reactive behaviour: it defines how an agent reacts to external stimuli
  - proactive behaviour: it defines how an agent behaves in order to reach its goals/tasks
- State

Environment in ABM

Defining the environment of an ABM

- Topology definition
- Complex internal dynamics

The agents can interact with the environment

What is ABM

Execute an ABM

- Running an ABM
- Study its evolution
  - observing individual and environment evolution
  - observing global system properties as emergent properties from agent-environment and inter-agent interaction
  - making in-silico experiment

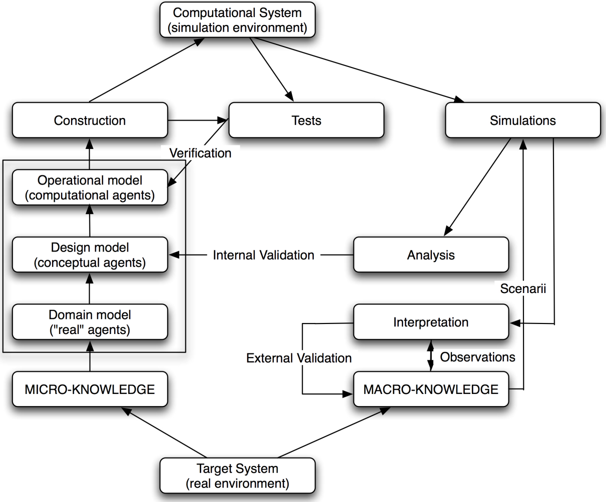
Advantages of ABM

- When flexible conditional or even adaptive individual behaviour has to be formulated
- When interactions with flexible individual participants have to be represented
- When inhomogeneous space is relevant
- When the simulation consists in mutable interacting participants
    - agents can be erased
    - new agents can enter in the scenario

Problems of ABM

- There exists neither an unified formal framework for ABM nor a widely accepted methodology for developing MABS
- Poor of formal definition of the modelling elements and rules
- Lack of conceptual language
- Increased amounts of parameters
- Software development remains a significant barrier to the use of ABM
  - there is a serious inconsistence and incongruence between agents of the conceptual model and agents of the computational model

ABM and MABS Methodology in a Figure



Outline

Simulation Platform Issues

Standard issues [Railsback et al., 2006]

- Model structure
- Discrete event simulation
  - *Scheduling*: to control which specific actions are executed and when (in simulated time)
  - *Marsenne Twister*: random number generation
- Distributed simulation

Swarm[1]

Swarm

- Objectives
  - to ensure a widespread use across scientific domains
  - to implement a model
  - to provide a virtual laboratory for observing and conducting experiments
- Swarm is implemented in Objective-C

[1] http://www.swarm.org/

Repast[2]

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

[2] http://repast.sourceforge.net/

MASON

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

NetLogo

- 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

What do Existing Simulation Frameworks Miss?

- Incoherence between the design model and the computational model
  - computational agents $\neq$ conceptual agents
  - No first class abstraction for modelling the environment

ABM e MABS for Biological Systems

Modelling and simulating hematopoietic stem cells

- Cell is modelled as an agent
  - agent's sensors are the cell's membrane proteins
  - agent's state is defined as the gene expression level
  - agent's reactive behaviour models the signalling transduction pathways
  - agent's proactive behaviour models the gene regulation networks
- Cell's micro-environement is modelled as agents environment
- Cell's interaction
  - direct interaction is modelled as agent-agent communication
  - indirect interaction is modelled through the liberation in the environment of molecules
- [Montagna et al., 2007]

Bibliography I

Banks, J. (1999).
Introduction to simulation.
In Farrington, P., Nembhard, H. B., Sturrock, D. T., and Evans, G. W., editors, *Proceedings of the 1999 Winter Simulation Conference*, pages 7–13.

Klugl, F. and Norling, E. (2006).
Agent-based simulation: Social science simulation and beyond.
Technical report, The Eighth European Agent Systems Summer School (EASSS 2006), Annecy.

Bibliography II

Montagna, S., Omicini, A., Ricci, A., and d'Inverno, M. (2007).
Modelling hematopoietic stem cell behaviour: An approach based on multi-agent systems.
In Allgöwer, F. and Reuss, M., editors, *2nd Conference "Foundations of Systems Biology in Engineering" (FOSBE 2007)*, pages 243–248, Stuttgart, Germany. Fraunhofer IBR Verlag.

Parisi, D. (2001).
Simulazioni - La realtà rifatta al computer.
Società editrice il Mulino.

Railsback, S. F., Lytinen, S. L., and Jackson, S. K. (2006).
Agent-based simulation platforms: Review and development recommendations.
Simulation, 82(9):609–623.

Bibliography III

Uhrmacher, A. M., Degenring, D., and Zeigler, B. (2005).
Discrete Event Multi-level Models for Systems Biology.
In Priami, C., editor, *Principles of Organization in Organisms: Proceedings of the Workshop on Principles of Organization in Organisms*, volume 3380 of *Lecture Notes in Computer Science*, pages 66–89. Springer.

Simulation & Multi-Agent Systems
An Introduction

Multiagent Systems LS
Sistemi Multiagente LS

Andrea Omicini & Sara Montagna
{andrea.omicini, sara.montagna}@unibo.it

Ingegneria Due
Alma Mater Studiorum—Università di Bologna a Cesena

Academic Year 2007/2008