Agent-Based Modelling & Simulation

Rosaldo Rossetti
rossetti@fe.up.pt
Http://www.fe.up.pt/~rossetti
Associate Professor at FEUP – Faculty of Engineering of the University of Porto
Researcher at LIACC – Artificial Intelligence and Computer Science Lab.

Simulation Approaches

- **Types of discrete models**
  - Event-oriented
  - Process-oriented
  - Activity-oriented
  - Object-oriented
  - Agent-based

Simulation Approaches

- **As for Agent-Based Modelling and Simulation (ABMS)**
  - Agent-Based Modelling
    - Agents as a metaphor for system modelling
    - Simulation methodology for MAS simulation
  - Agent-Directed Simulation
    - Agents steer and manage the whole simulation process
    - ML and Meta-modelling for intelligent calibration and scenario management
  - Agent-Oriented Simulation SW
    - SW architectures for Simulation IDEs based on MAS

Motivation for ABMS

- **Traditional Simulation Drawbacks:**
  - Systems are getting more complex
  - Complex systems are difficult to model as a whole (aggregate)
  - Higher level tools available
  - Human behaviour is often neglected or over simplified in the simulation process

- **Distributed Applications Challenges (DDD problems):**
  - Need for coordination of heterogeneous entities
  - Entities with local processing/decision capabilities
  - Human vs Artificial entities

- **Agent Based Modelling and Simulation:**
  - Entities represented by Agents with Autonomous Behaviour

Motivation for ABMS

- **Agent-Based Modelling**
  - Computational method
  - Experiments
  - Models
    - Scale models
    - Ideal-type models
    - Analogical (analogy) models
    - Mathematical or equation-based models
  - Agents
  - The environment

Artificial Intelligence

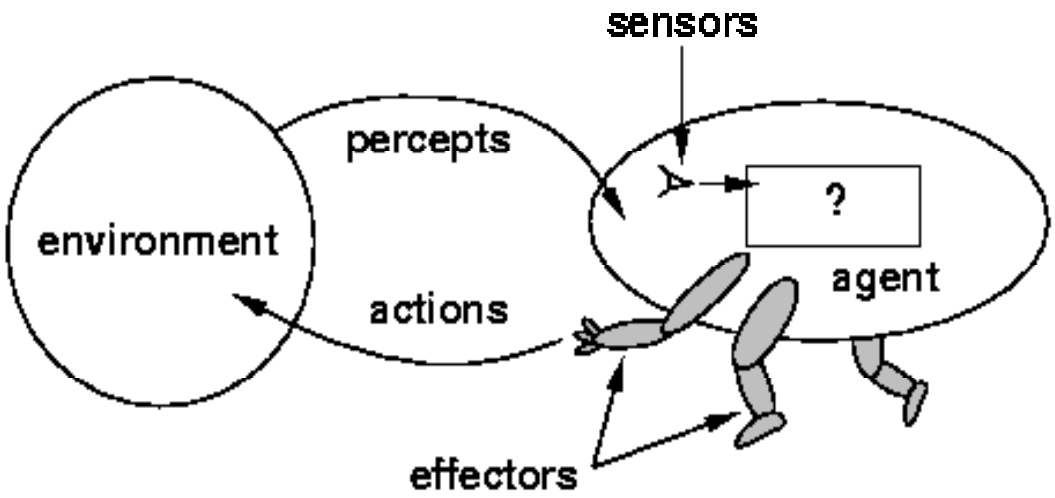
- **Intelligence**
  - "Capacity to **solve new problems** through the use of **knowledge**" (do things correctly!)

- **Artificial Intelligence**
  - "Science concerned with building **intelligent machines**, that is, machines that perform tasks that when performed by humans require intelligence"
    - The Turing Test
    - Dijkstra 's Submarine:

"The question of whether a computer can think is no more interesting than the question of whether a submarine can swim."

Autonomous Agents

Computational System, situated in a given **environment**, that has the ability to **perceive** that environment using **sensors** and **act**, in an **autonomous way**, in that environment using its **actuators** to fulfill a given **function**."



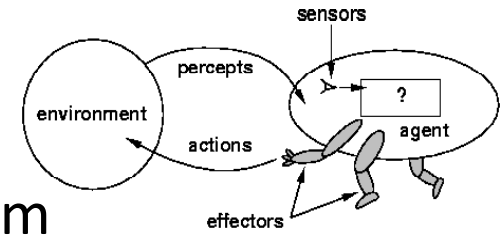
Agent Requirements

- Requisites:
  - Perceive its environment (sensors)
  - Decide actions to execute ("think")
  - Execute actions in/upon the environment using actuators
  - Communicate?
  - Perform a complex function?
- Agents vs Objects:
  - Agents decide what to do
  - Object methods are called externally
  - Agents react to sensors and control actuators

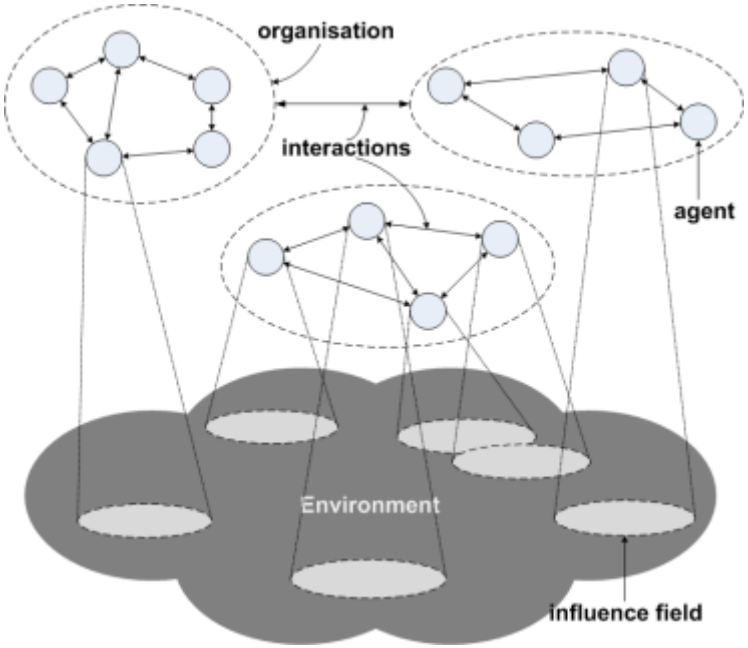
"Objects do it for free; agents do it for money"

Multi-Agent System (MAS)

- Composed by **multiple agents** that:
  - Exhibit **autonomous behavior**
  - **Interact** with the other agents in the system



- **MAS Motivation:**
  - Problem Dimensions
  - Legacy Systems
  - Natural Solution (distributed problems)
  - Distributed knowledge or information
  - Human-machine interface
  - Project Clarity and simplicity
  - Efficiency
  - Robustness and Scalability
  - Problem division
  - Information privacy



Multi-Agent System (MAS)

- **To build individual autonomous intelligent agents is important!**

- **However:**
  - Agents don't leave alone…
  - Necessary to work in group…
  - Multi-Agent Applications…
  - Coordination is necessary: "to work in harmony in a group to achieve a given goal"

Reasons for Coordination

- Dependencies in agent actions
- Need to respect global constraints
- No agent, individually has enough resources, information or capacity to execute the task or solve the complete problem
  - Tragedy of the commons
  - Nash equilibrium
  - (Evolutionary) Game theory
  - Social dilemmas

- Efficiency:
  - Information exchange or tasks division
- Prevent anarchy and chaos:
  - Partial vision, lack of authority, conflicts, agent's interactions

Cooperative vs. Competitive MAS

- **Cooperative MAS:**
  - Usually projected by a single entity
  - Global utility and global performance

- **Competitive MAS ("self-interested agents"):**
  - Each agent has a distinct designer
  - Agents have their own motivation and agenda
  - Agents are interested in their own utility
  - Usual in negotiation, electronic commerce, internet

MAS & Intelligent Robotics

- **Robotics**
  - Science and technology for projecting, building, programming and using Robots
  - Study of **Robotic Agents:** natural physical embodiment for an agent
  - **MAS → Multi-Robot Systems**
  - Increased Complexity:
    - **Environments**: Dynamic, Inaccessible, Continuous e Non-deterministic!
    - Perception: Vision, **Sensor Fusion**
    - Action: Robot Control
    - Robot Architecture (Physical / Control)
    - Navigation in unknown environments
    - **Interaction** with other robots/humans
    - **Multi-Robot Systems: robots with social abilities**

The "Name" - ABMS

ABMS is known by many names:

- ABM: "Agent-based modeling" or "anti-ballistic missile?"
- ABS: "Agent-based simulation" or "anti-lock braking system?"
- IBM: "Individual-based modeling" or "International Business Machines Corporation?"

ABM, ABS, and IBM are all widely-used acronyms, but "ABMS" will be used throughout this course

ABMS is not the same as "mobile agents"

Need for Agent-based Modeling

We live in an increasingly complex world!

Systems are More Complex:

- Systems are becoming more complex: more variables and interactions
- Decentralization of Decision-Making
- Increasing Physical and Economic Interdependencies

New Tools, Toolkits, Modeling Approaches:

- New tools exist to analyze complex systems
- Economic markets and the diversity among economic agents
- Social systems, social networks
- Robotic systems - interaction

Avalaible Data:

- Micro-data available in databases (finer levels of granularity) enables micro-simulations!

Computational Power :

- Computational power advancing – micro-simulations!

ABMS as a New Field

Agents

- Discrete entity with its their own perceptions, actions, goals and behaviors
- Autonomous, capable of adapting and modifying its behaviors

Assumptions

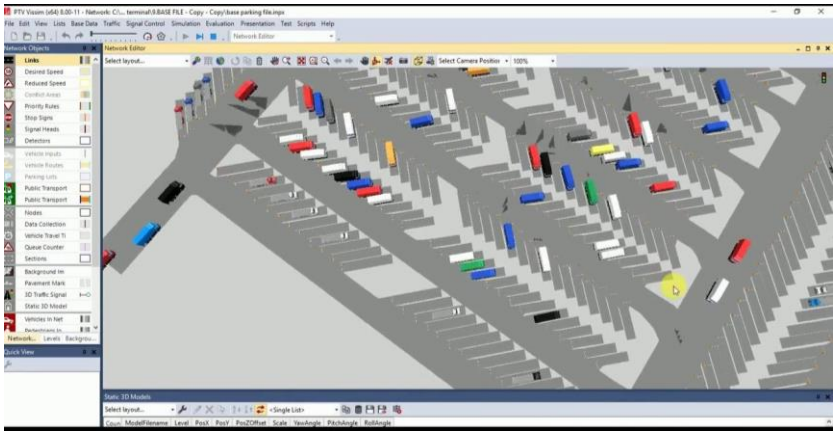
- Some key aspect of behaviors can be described
- Mechanisms by which agents interact can be described
- Complex social processes and a system can be built “bottom-up”

Features of Agent-Based Models

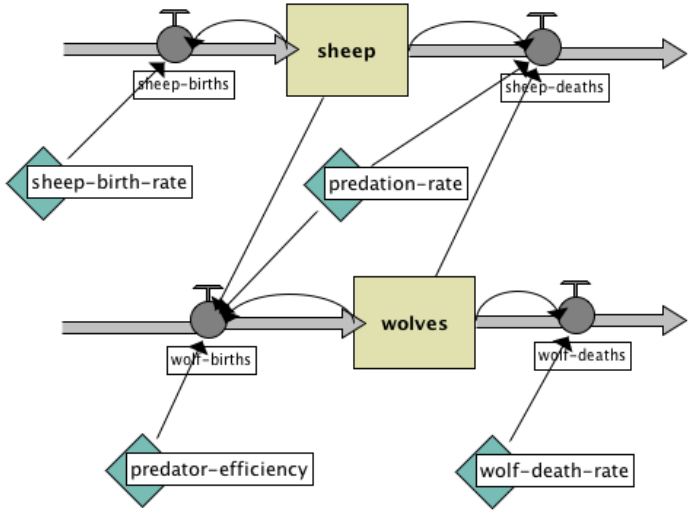
- **Ontological correspondence**
- **Heterogeneous agent**
- **Representation of the environment**
- **Agent interactions**
  - Simply passing data to one another
  - Based on more sophisticated language representations (e.g. NLP)
- **Bounded rationality**
- **Learning**
  - Individual learning
  - Evolutionary learning
  - Social learning

Other Modelling Approaches

- **Microsimulation**

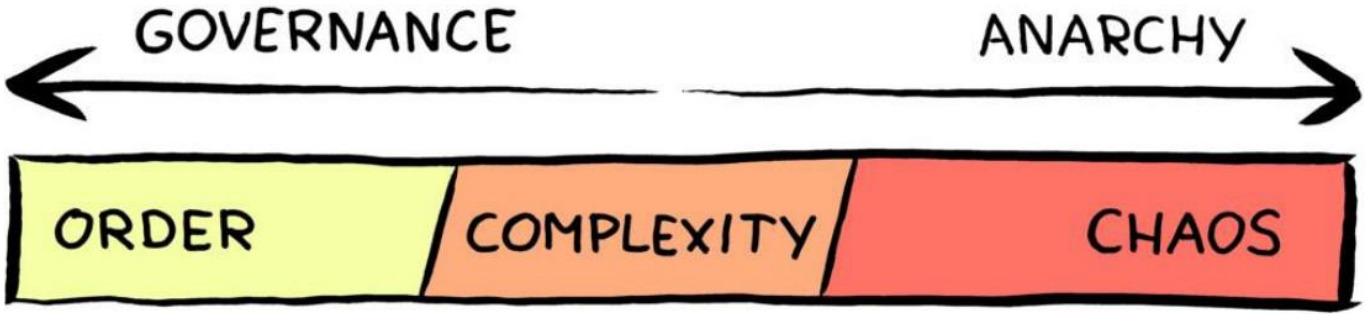


- **System Dynamics**



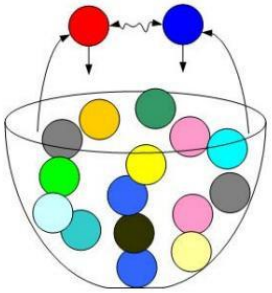
Agent-Based Model

- **An agent-based model consists of:**
  - A set of agents (part of the user-defined model)
  - A set of agent relationships (part of the user-defined model)
  - The environment in which agents cohabit
  - A framework for simulating agent behaviors and interactions (provided by an ABMS toolkit or other implementation)
- **Unlike other modeling approaches, agent-based modeling begins and ends with the agent's perspective**

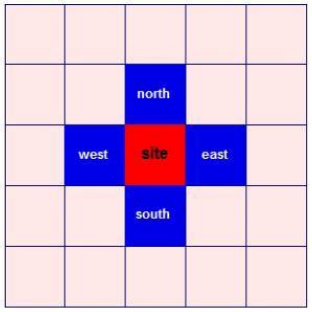


Agent-Based Model

- **Types of environments**

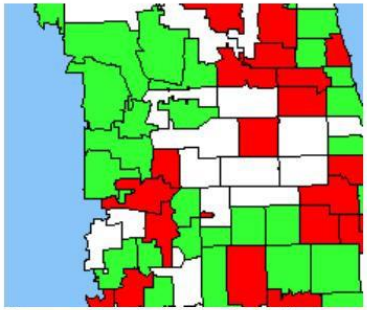


(a) "Soup" Model (Aspatial)

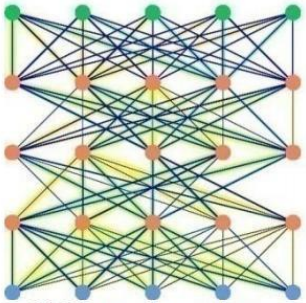


(b) Cellular Automata (von Neumann)

(c) Euclidean Space (2-D)



(d) Geographic Information System (GIS)



(e) Network topology

Agent-Based Model

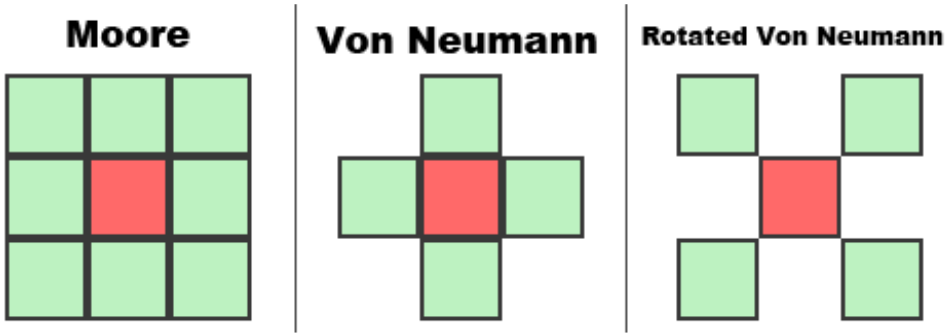
- **Types of interactions**

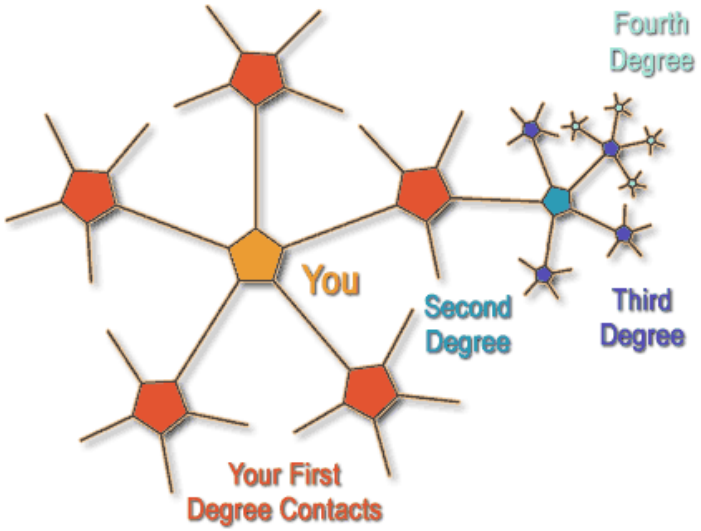
  Depending on the environment agents interact differently.

  – Typically, agents interact with neighbours.

  – In a network, first-degree links are neighbours

  – Interactions can be implicit or explicit



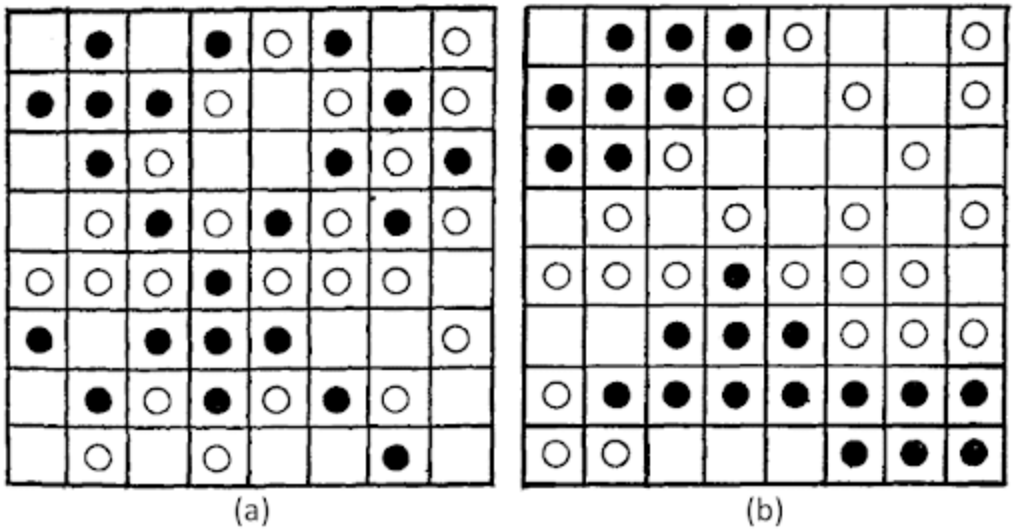


Agent-Based Model

- **Finding patterns…**

Agent-Based Model

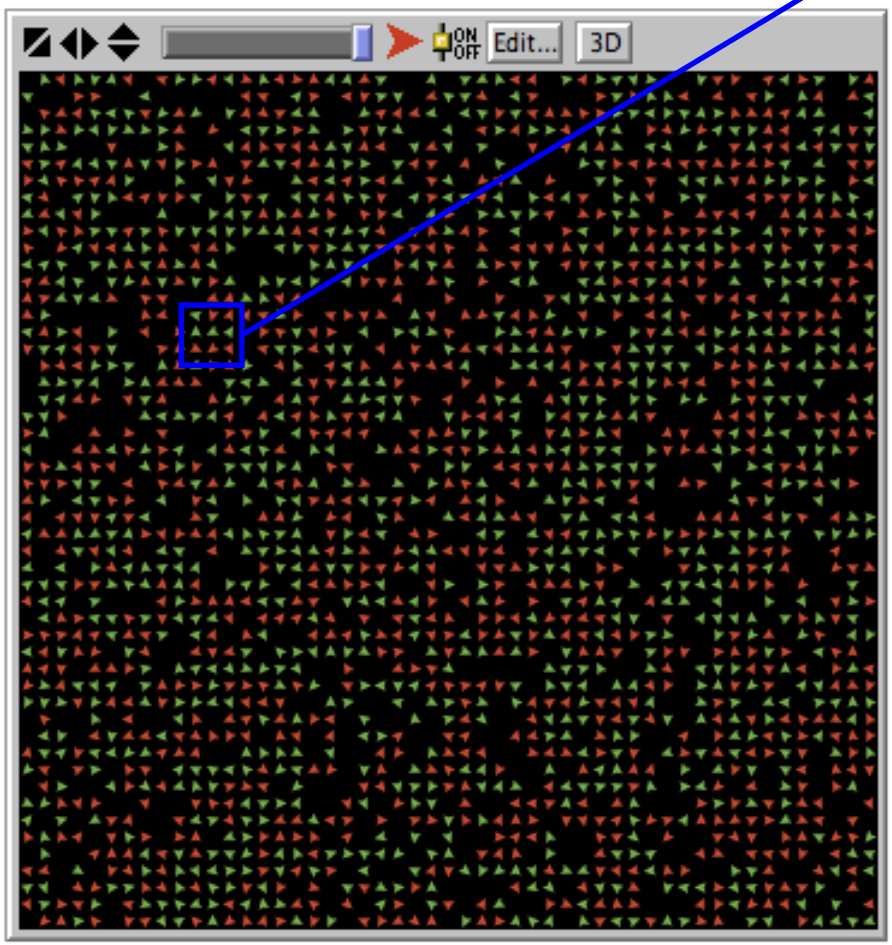
- **Finding patterns…**

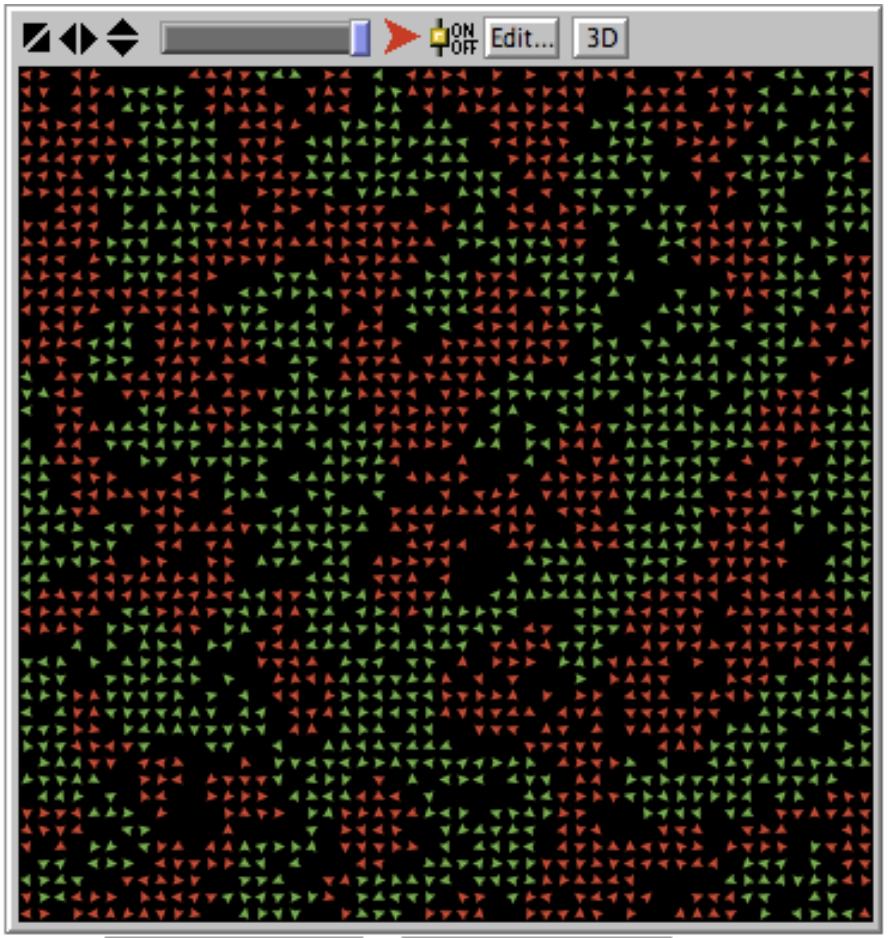


- The Schelling **model** of **segregation** (Schelling 1971, 1978) is one of the earliest agent-based **models** of social science.

Agent-Based Model

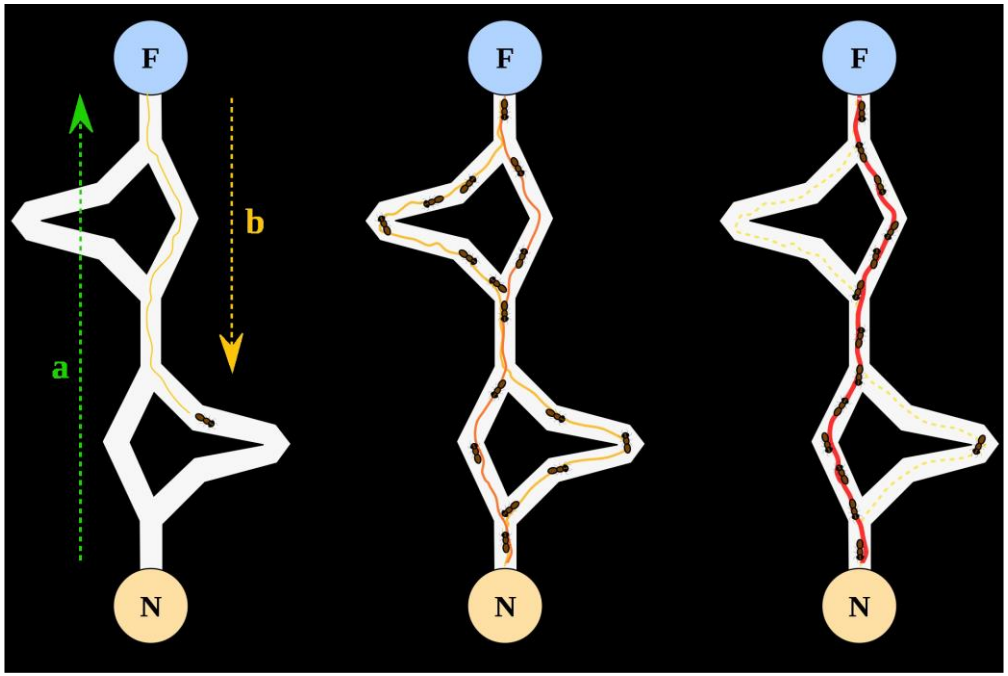
- **Finding patterns…**





Agent-Based Model

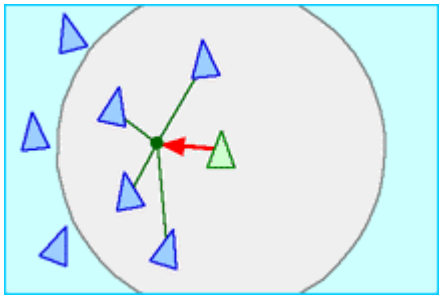
- **Finding patterns…**



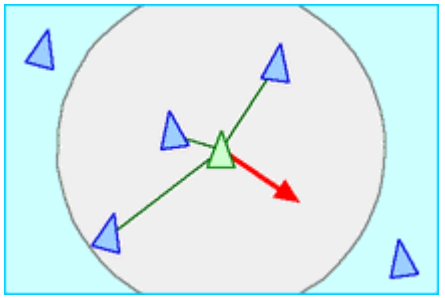
http://agentbase.org/model.html?b24f11b263d0de2610f1#

Example: Modeling Simple Flocking Behavior with Agent Rules

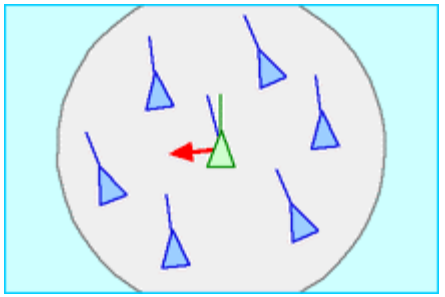
- ***Cohesion:***
  Steer to move toward the average position of local flockmates



- ***Alignment:***
  Steer towards the average heading of local flockmates

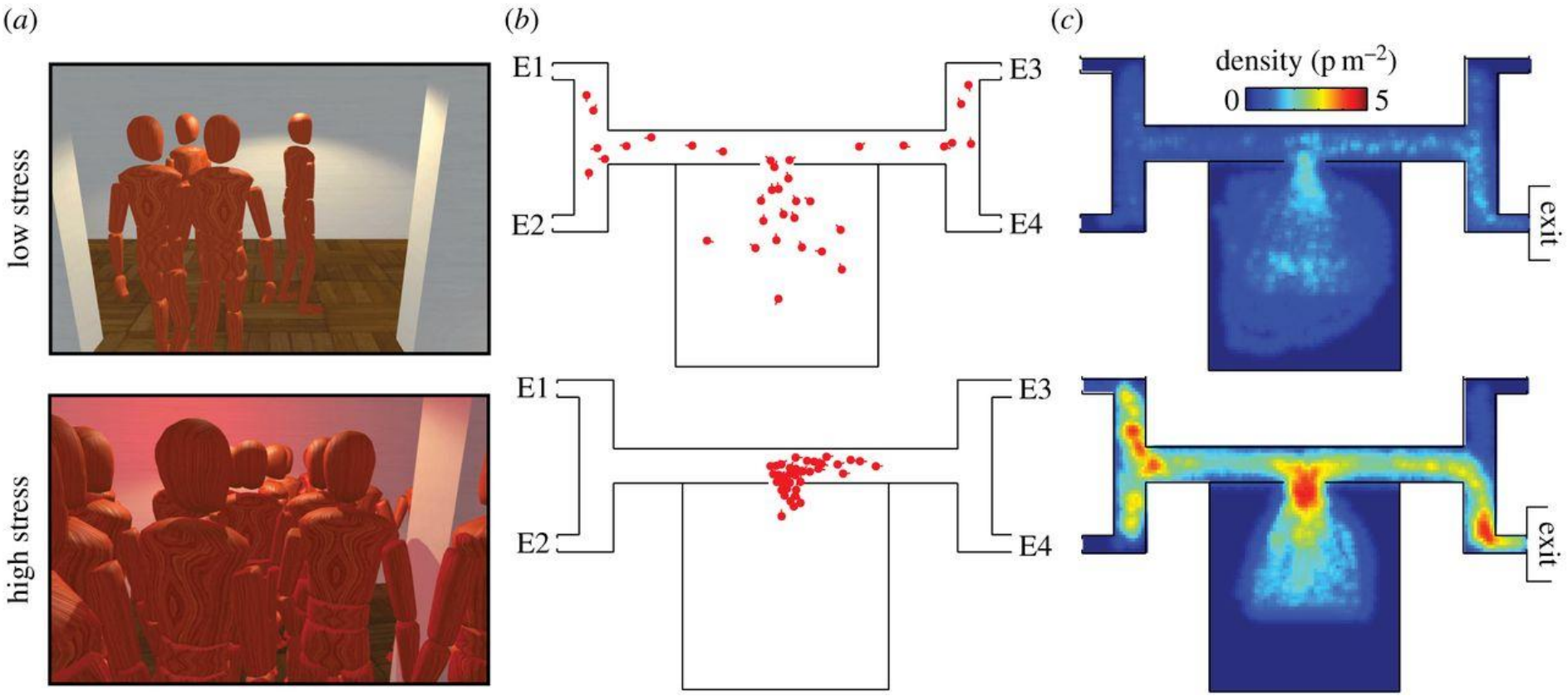


- ***Separation:***
  Steer to avoid crowding local flockmates



"Boids"by Craig Reynolds, http://www.red3d.com/cwr/boids/

Example: Modeling Simple Flocking Behavior with Agent Rules



Other examples

- **Flows**
- **Markets**
- **Organisations**
- **Diffusion**
  - Urban models
  - Opinion dynamics
  - Consumer behaviour
  - Supply chain management
  - Electricity market
  - Participative and companion modelling

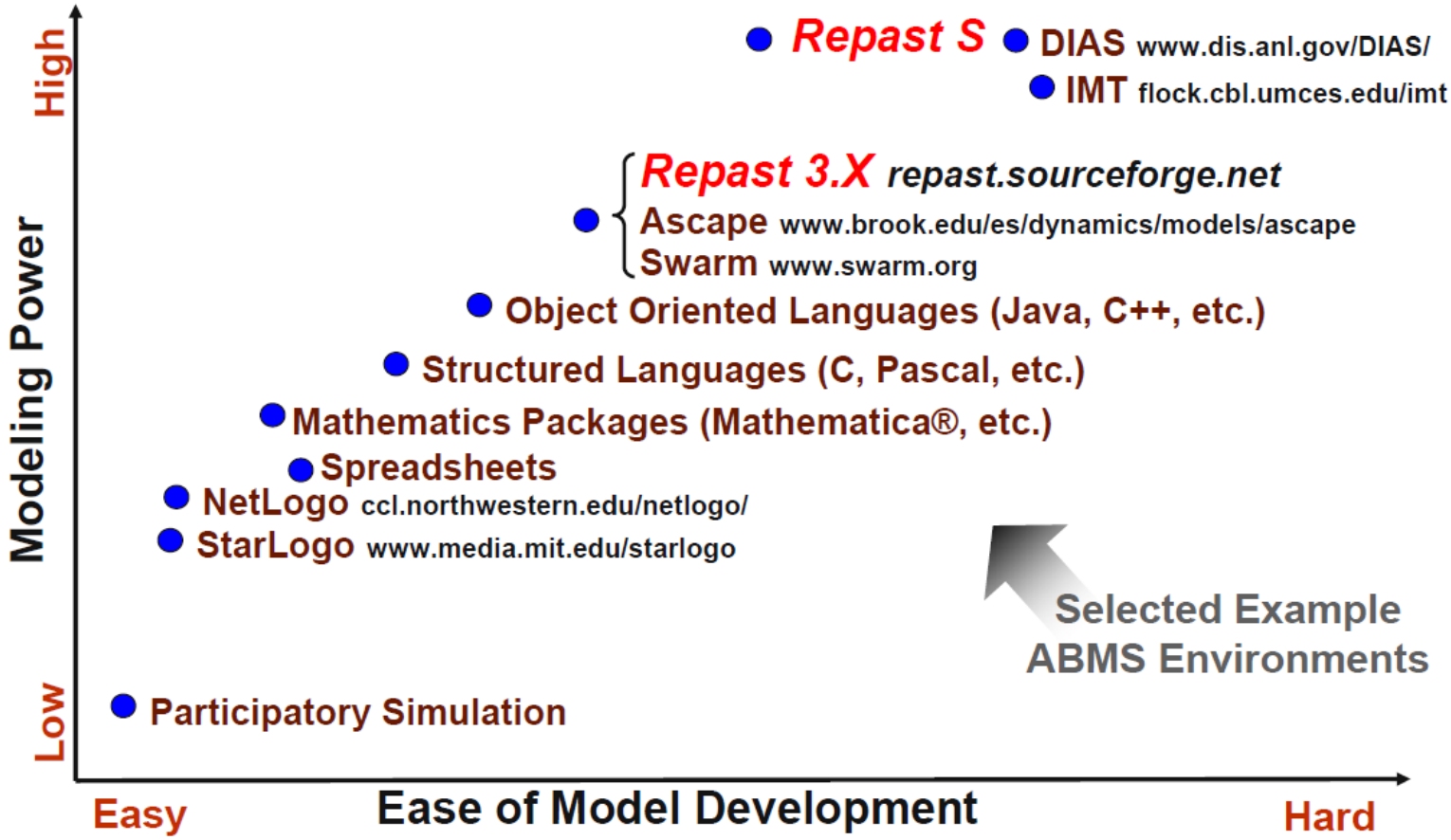
ABMS Applications

- **Business and Organizations**
  - Manufacturing Operations
  - Supply chains
  - Consumer markets
  - Insurance industry
- **Economics**
  - Artificial financial markets
  - Trade networks
- **Infrastructure**
  - Electric power markets
  - Transportation
  - Hydrogen infrastructure
- **Crowds**
  - Pedestrian movement
  - Evacuation modeling
- **Society and Culture**
  - Ancient civilizations
  - Civil disobedience
  - Social determinants of terrorism
  - Organizational networks
- **Military**
  - Command & control
  - Force-on-force
- **Biology**
  - Population dynamics
  - Ecological networks
  - Animal group behavior
  - Cell behavior and subcellular processes

ABMS Platforms

- **Agent-based Modeling and Simulation Toolkits**
  - Repast (Java) –similar to Swarm (Objective C, Java)
  - NetLogo, StarLogo (Logo, Lisp)
  - MASON
  - AnyLogic (commercial)
- **General Tools**
  - Spreadsheets, with macro programming
  - Computational Mathematics Systems: MATLAB and *Mathematica*
- **General Programming Languages (Object-oriented)**
  - *Java, C++, Pascal*
- ***Agent-based model development process often makes use of several tools***

ABMS Platforms



When to use ABM?

- When agents are the natural representation metaphor:
  - When there are decisions and behaviors that can be well-defined discretely
  - When it is important that agents adapt and change their behaviors
  - When it is important that agents have a dynamic relationship with other agents, and agent relationships form, change and decay
  - Agents learn and engage in dynamic strategic behaviours and decision-making
  - Organizational dynamics (adaptation and learning are important at the organization level)
  - Spatial component inherent to their behaviours and interactions
  - When the past is no predictor of the future because the processes of growth and change are dynamic

Why to use ABM?

- **Agent-based models** represent individuals, their behaviours and their interactions
- **Equation-based models** represent aggregates and their dynamics
- Agents have decision-making abilities and an **understanding of their environment**
- **Micro to Macro:** Agent behaviours emerging toward System Behaviours

Questions?

Agent-Based Modelling and Simulation