

LECTURE 1: INTRODUCTION TO AGENTS AND MAS

Introduction to Multi-Agent Systems (MESIIA, MIA)

Universitat Rovira i Virgili

Outline of the lecture

1. Main trends in Computer Science
 2. Agents and multi-agent systems
 3. Objections to MAS
 4. Agents vs. related technologies
 5. Viewpoints on agent technology
-

Important aspects in Computer Science

- Five trends are marking the history of computing:
 - *ubiquity*
 - *interconnection*
 - *intelligence*
 - *delegation*
 - *human-orientation in programming methodologies*

Ubiquity



- The continuous reduction in size and cost of computing capability has made it possible to introduce processing power into places and devices that would have once been uneconomic / unimaginable (e.g. smartwatches)
- As processing capability spreads, sophistication and intelligence become ubiquitous

Ambient Intelligence

- Aimed at seamless delivery of services and applications, based on ubiquitous computing, ubiquitous communication and intelligent user interfaces.
 - The vision
 - An environment of potentially **thousands of embedded and mobile devices** (or software components) interacting to support user-centered goals and activity.
 - **Autonomy, distribution, adaptation** and **responsiveness** are key characteristics of these independent and distributed components, and in this sense they share similar characteristics to intelligent **agents**.
-

Transportation
Intelligent Station



Commerce
Intelligent Exhibition



Leisure
Intelligent Playground



Education
Intelligent Classroom



Work
Intelligent Office



Home
Intelligent Living-room

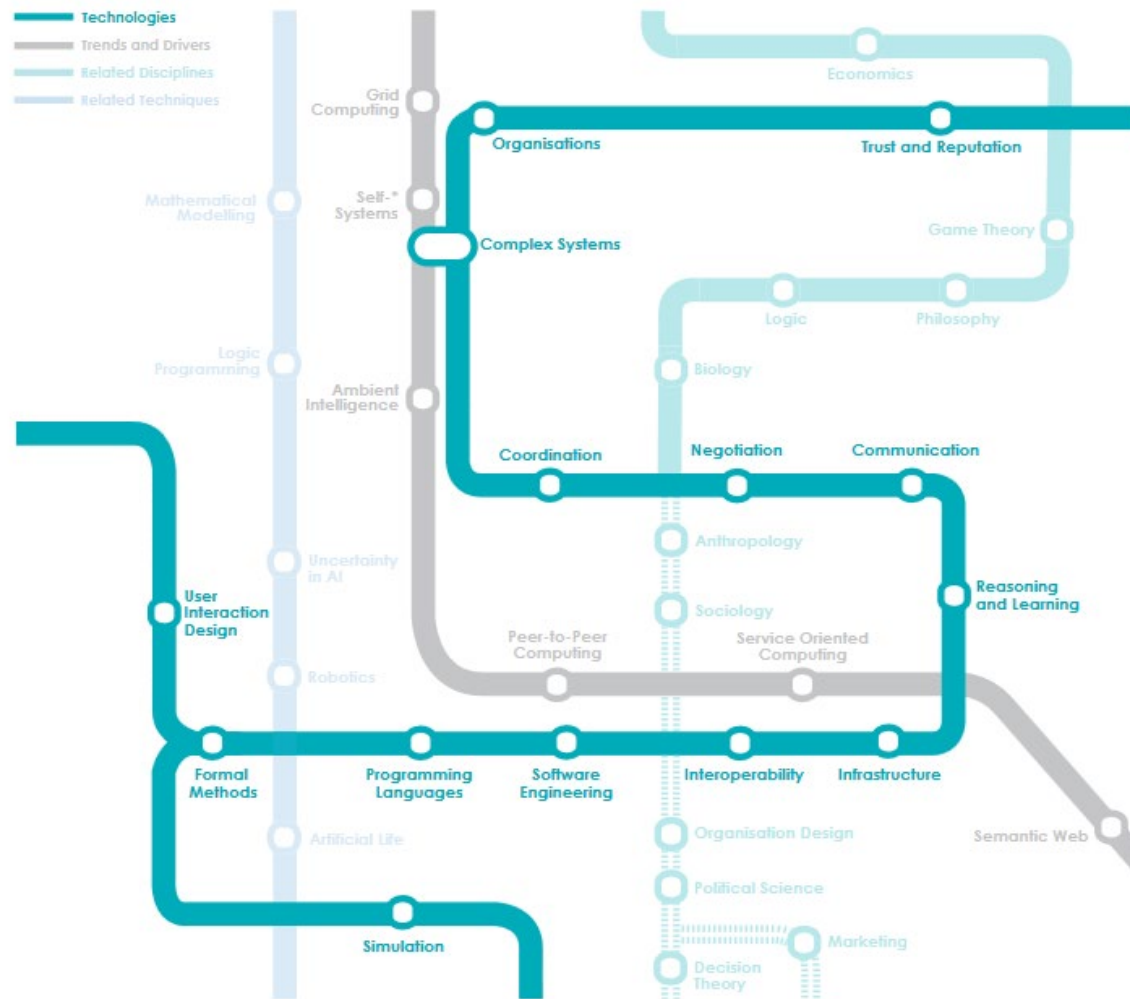


Interconnection

- Computer systems today no longer stand alone, but are networked into large distributed systems
 - Obvious example: Internet
 - Since distributed and concurrent systems have become the norm, some researchers are putting forward theoretical models that portray *computing as primarily a process of interaction*
-

Agent Technology: Computing as Interaction

A Roadmap for Agent Based Computing



Compiled, written and edited by

Michael Luck, Peter McBurney, Onn Shehory, Steve Willmott and the AgentLink Community

Connectivity -Technological context

- Semantic Web

- The Semantic Web is based on the idea that the data on the Web can be defined and linked in such a way that it can be used by machines for the automatic processing and integration of data across different applications (Berners-Lee et al., 2001).

- Peer-to-Peer computing

- Grid computing

Grid Computing

What is all about Grid ?

- 1) Sharing, Selecting, aggregation of resources with a policy universally agreed.
- 2) Distributive supercomputing for a better future



Copyright: adas1@cio

Intelligence

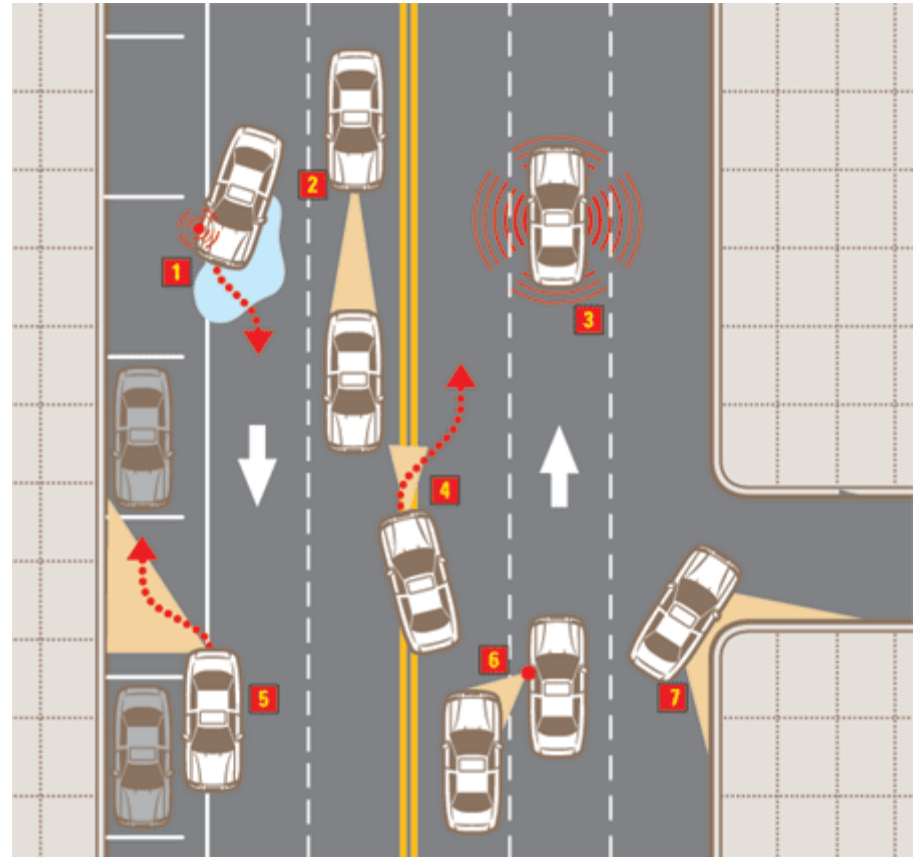
- The **complexity** of tasks that we are capable of automating and delegating to computers has grown steadily
 - That requires an increasing amount of domain knowledge and inference capabilities, in short, more **intelligent** problem solving techniques
-

Delegation

- Computers are doing more for us – without our intervention
 - We are *giving control* to computers, even in safety critical tasks
 - One example: fly-by-wire aircraft, where the machine's judgment may be trusted more than an experienced pilot
 - Next on the agenda: self-driving cars, intelligent braking systems, cruise control that maintains distance from car in front...
-

Driving automation

1. Road Condition Reporting
2. Adaptive Cruise Control
3. Omnidirectional Collision System
4. Lane-Departure Prevention
5. Auto Parallel Park
6. Blind-Spot Sensors
7. Corner Speed



Human-Oriented Programming

- The movement away from machine-oriented views of programming toward concepts and metaphors that more closely reflect the way we ourselves understand the world
 - Programmers (and users!) relate to the machine differently
 - Programmers conceptualize and implement software in terms of higher-level – more human-oriented – abstractions
-

Agent-Oriented Software Engineering

- Programming has progressed through
 - machine code
 - assembly language
 - machine-independent programming languages
 - sub-routines
 - procedures & functions
 - abstract data types
 - objectsto *agents*

Where does it bring us?

- *Delegation* and *Intelligence* imply the need to build computer systems that can act effectively on our behalf
- This implies:
 - The ability of computer systems to act *independently*
 - The ability of computer systems to act in a way that *represents our best interests* while interacting with other humans or systems

Interconnection and Distribution

- *Interconnection* and *Distribution* have become core motifs in Computer Science
 - But Interconnection and Distribution, coupled with the need for systems to represent our best interests, implies systems that can *cooperate* and *reach agreements* (or even *compete*) with other systems that have different interests (much as we do with other people)
-

So Computer Science expands...

- All of these trends have led to the emergence of *multiagent systems*



2 Agents and Multiagent Systems

- An agent is a computer system that is located in a dynamic environment and is capable of *independent* action on behalf of its user or owner (figuring out what needs to be done to satisfy design objectives, rather than constantly being told)
- *Intelligent* action, probably using Artificial Intelligence tools and techniques
- An agent should satisfy a certain number of properties [lecture 3]

Multiagent Systems

- A multiagent system is one that consists of a number of agents, which *interact* with one-another
 - In the most simple case, all agents are programmed by the same team and they collaborate to complete a task
 - In the most general case, agents will be acting on behalf of users with different goals and motivations
 - To successfully interact, they will require the ability to *cooperate*, *coordinate*, and *negotiate* with each other, much as people do
-

Related areas

- ❑ Several sub-disciplines of information technology are related to software agent technology:
 - computer networks,
 - software engineering,
 - Artificial Intelligence,
 - human-computer interaction,
 - distributed and concurrent systems,
 - ...
-

Micro and macro aspects

- Agent technologies can be grouped into three categories
 - Agent-level [micro level]
 - technologies and techniques concerned only with **individual agents** — for example, procedures for agent reasoning and learning [Agent architectures – lecture 2]
 - Interaction-level
 - technologies and techniques that concern the **communication** between agents
 - communication languages, interaction protocols and resource allocation mechanisms [Agent communication-lecture 4. Coordination and cooperation mechanisms-lectures 5 to 8]
 - Organization-level [macro level]
 - technologies and techniques related to **agent societies** as a whole
 - structure, trust, norms, obligations, etc. [Agent societies-lecture 9]
-

Multiagent Systems

- In Multiagent Systems, the main questions we address are:
 - What kinds of languages can agents use to **communicate**?
 - How can **cooperation** emerge in societies of self-interested agents?
 - How can self-interested agents recognize conflict, and how can they (nevertheless) reach **agreement**?
 - How can autonomous agents **coordinate** their activities so as to cooperatively achieve goals?

Multiagent Systems

- While these questions are all addressed in part by other disciplines (notably Economics and Social Sciences), what makes the multiagent systems field unique is that it emphasizes that the agents in question are *computational, information processing* entities.
-

Example 1 - Spacecraft Control

- When a space probe makes a flight to the outer planets, a ground crew is usually required to continually track its progress, and decide how to deal with unexpected eventualities. This is costly and, if decisions are required *quickly*, it is simply not practicable. NASA is investigating the possibility of making probes more *autonomous* — giving them richer decision making capabilities and responsibilities.
 - *NASA's DS1 started with it in 1999 and continues improving this approach*
-

[Topics](#)[Missions](#)[Galleries](#)[NASA TV](#)[Follow NASA](#)[Downloads](#)[About](#)[NASA Audiences](#)

Latest

Related

[SpaceCraft Oxygen Recovery \(SCOR\)](#)

5 months ago

[Explore: Transformative Missions and Discoveries](#)

7 months ago

[Cooperative Autonomous Distributed Robotic Explorers \(CADRE\)](#)

7 months ago

[Distributed Spacecraft Autonomy \(DSA\)](#)

10 months ago

[Deployable Composite Booms \(DCB\)](#)

a year ago

[High Performance Spaceflight Computing \(HPSC\)](#)

a year ago

[Autonomous Deployable Flat Folding](#)

Feb 5, 2021

Cooperative Autonomous Distributed Robotic Explorers (CADRE)

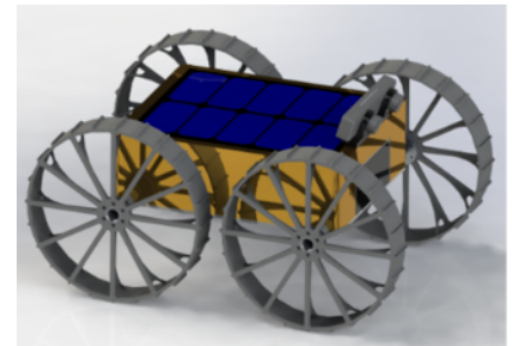


NASA's Cooperative Autonomous Distributed Robotic Explorers (CADRE) project is developing a network of shoe-box-sized mobile robots that could enable future autonomous robotic exploration of the Moon, Mars, and beyond. The CADRE robots are the latest version of NASA's [A-PUFFER](#) technology. Each robot contains an onboard computer with a wireless radio for communication and a stereo camera – which has multiple lenses and image sensors – for sensing the environment in front of it and capturing 3D imagery. NASA's Jet Propulsion Laboratory in Southern California is developing the robotic scouts, designing them to explore as a group and collect data in hard-to-reach places such as craters and caves on the Moon.

"CADRE robots could complement NASA's larger planetary robots and rovers," said Sonny Mitchell, a program element manager for NASA's Game Changing Development Program, which funds the project. "Multiple small, autonomous robots could cover more ground, potentially helping us map unexplored regions on the Moon."

How CADRE Robots Work

During launch and the trip through space, the robots would be stored in a compact position in a larger lander. Once the lander arrives at its destination, the CADRE scouts would eject onto the surface. The explorers are designed to be agile and smart enough to explore extreme terrain on the Moon independently, without being controlled by astronauts or engineers on Earth.



Computer-aided design of the Cooperative Autonomous Distributed Robotic Exploration (CADRE) robot concept.
Credits: NASA/JPL-Caltech



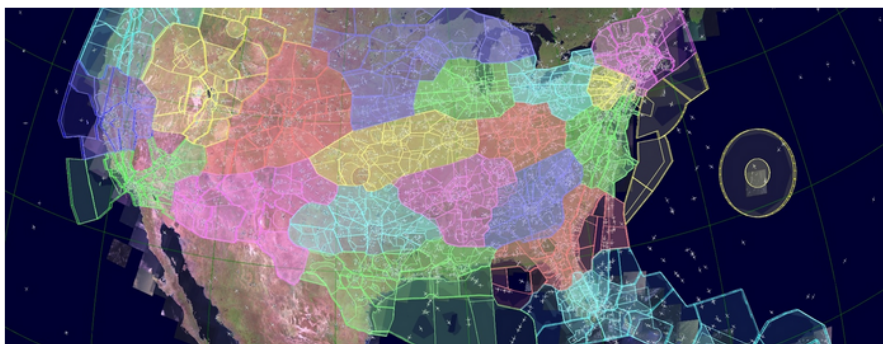
Example 2 - Air Traffic Control

“A key air-traffic control system suddenly fails, leaving flights in the vicinity of the airport with no air-traffic control support. Fortunately, autonomous air-traffic control systems in nearby airports recognize the failure of their peer, and cooperate to track and deal with all affected flights.”

- Systems **taking the initiative** when necessary
- Agents **cooperating** to solve problems beyond the capabilities of any individual agent

AgentFly ATM Simulation Suite

The AgentFly ATM simulation suite is a complex tool for modeling and simulation of air traffic and air traffic management. The ATM AgentFly is fast-time gate-to-gate simulation used in NextGen and SESAR programs. The simulation consists of IFR, VFR and unmanned air traffic and main actors - air traffic controllers, pilots, and airline operation centers. The AgentFly platform is agent-based simulation framework designed to be used as Fast-Time Simulation (FTS) as well as Real-Time Simulation (RTS).



The FTS mode is suitable for what-if studies and analysis, validation of new concepts or interconnection with other FTS systems. The system allows running quickly various options, settings and parameters to evaluate changes. The RTS mode is suitable for connection with other real-time systems and can include humans-in-the-loop, e.g., pseudo-pilots or air traffic controllers.

Agent-based simulation allows precise control of simulation time, large-scale scenarios with various actors (thousands of air traffic controllers and tens of thousands of aircraft) and controlled uncertainty and randomization. The

Example 3 – e-personal assistants

- E-personal agents can assist people in different kinds of tasks
- With access to Internet, they can plan, arrange, buy, negotiate – carry out arrangements of all sorts that would normally be done by their human user
- For example, preparing a holiday trip

Multiagent Systems is Interdisciplinary

- The field of Multiagent Systems is influenced and inspired by many other fields:
 - Economics
 - Philosophy
 - Game Theory
 - Logic
 - Ecology
 - Social Sciences
- This can be both a strength (infusing well-founded methodologies into the field) and a weakness (there are many different views as to what the field is about)
- This has analogies with Artificial Intelligence itself

3-Objections to MAS (I)

■ Isn't it all just Distributed/Concurrent Systems?

There is much to learn from this community, but:

- ❑ Agents are assumed to be autonomous, capable of making independent decision – so they need mechanisms to synchronize and coordinate their activities **at run time**
 - ❑ Agents are (can be) self-interested, so their interactions are “economic” encounters (negotiation)
-

Objections to MAS (II)

■ Isn't it all just AI?

- We don't need to solve all the problems of Artificial Intelligence (i.e., all the components of intelligence) in order to build really useful agents
- Classical AI ignored *social* aspects of agency. These are important parts of intelligent activity in real-world settings

Objections to MAS (III)

■ Isn't it all just Economics/Game Theory?

These fields also have a lot to teach us in multiagent systems, but:

- Insofar as game theory provides *descriptive* concepts, it doesn't always tell us *how* to compute solutions; we're concerned with computational, resource-bounded agents
- Some assumptions in economics/game theory (such as a *rational agent*) may not be valid or useful in building artificial agents

Objections to MAS (IV)

- Isn't it all just Social Science?

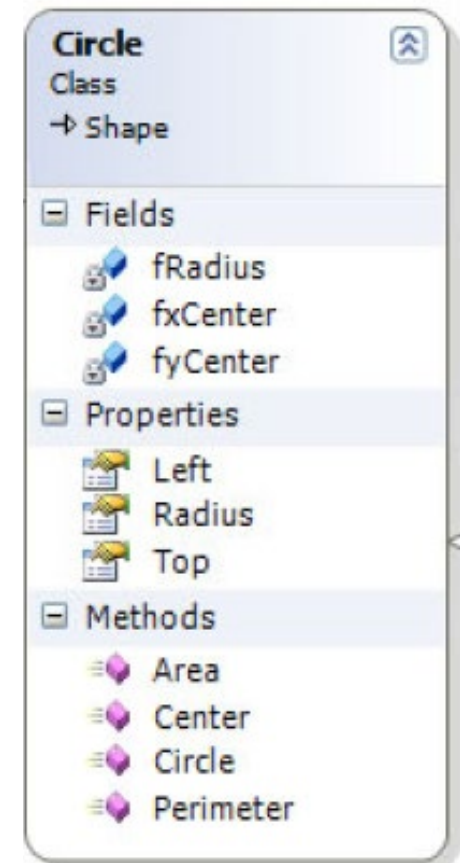
- We can draw insights from the study of human societies, but there is no particular reason to believe that artificial societies will be constructed in the same way
 - Again, we have inspiration and cross-fertilization, but hardly subsumption
-

4-Agents versus related technologies

- If agents are autonomous entities that display an intelligent behaviour, what makes them so different from other well known techniques, like object-oriented programming or intelligent (knowledge-based) systems?
-

Agents and Objects

- Are agents just **objects** by another name?
- Object:
 - encapsulates some state
 - communicates via message passing
 - has methods, corresponding to operations that may be performed on the state



Agents vs Objects (I)

- *Agents are **autonomous***
 - Agents embody a stronger notion of autonomy than objects
 - They decide for themselves whether or not to perform an action on request from another agent
 - When a method is invoked on an object, it is always executed

Agents vs Objects (II)

- *Agents are smart, intelligent*
 - Capable of flexible (reactive, proactive, social) behaviour
 - The standard object model has nothing to say about such types of behaviour
- *Agents are active*
 - A multi-agent system is inherently multi-threaded, in that each agent is assumed to have at least one thread of active control

Agents vs Objects (III)

Objects

- Encapsulate state and control over it via **methods**
- **Passive** – have no control over a method execution
- **Non autonomous**
- **Reactive to events**

Agents

- Encapsulate state, control over it via ***actions and goals***
- ***Active*** – can decide when to act and how
- ***Autonomous***
- ***Proactive***

In summary ...

- *Objects do it for free*

An object cannot refuse a method invocation

- *Agents do it because they want to*

The service requester is authorised, the agent has enough resources available, the action is convenient for the agent, ...

- *Agents do it for money*

The agent can get an economic profit

Agents and Expert Systems

- Aren't agents just **expert systems** by another name?
- Expert systems contain typically disembodied 'expertise' about some (abstract) domain of discourse
- Example: MYCIN knows about blood diseases in humans
 - It has a wealth of knowledge about blood diseases, in the form of rules
 - A doctor can obtain expert advice about blood diseases by giving MYCIN facts, answering questions, and posing queries

Agents and Expert Systems

- Main differences:

- agents *situated in an environment*:

MYCIN is not aware of the world — the only information that it obtains is by asking questions to the user

- agents *act*:

MYCIN does not operate on patients

- Expert systems typically work alone, whereas agents *cooperate* to achieve a common goal.

- Sometimes an expert system is *agentified* and included in a MAS

5-Some Views of the Field

- *Agents as a paradigm for software engineering:*

Software engineers have derived a progressively better understanding of the characteristics of complexity in software. It is now widely recognized that *interaction* is probably the most important single characteristic of complex software

Agents as design metaphor

- **Agent-oriented software engineering**
 - Agents allow software designers and developers to structure an application using autonomous, communicative components.
 - In this sense, software agents offer a new and often more appropriate route to the development of complex computational systems, especially in open and dynamic environments.
 - Modularity, reusability, collaborative behaviour



Agents as source of technologies

- Software agent technologies span a range of specific **techniques and algorithms**
 - balancing **reaction** and **deliberation** in individual agent architectures **[lecture 2]**
 - **learning** from and about other agents in the environment
 - eliciting and acting upon **user preferences**
 - finding ways to **negotiate** and **cooperate** with other agents
 - developing appropriate means of forming and managing **coalitions** (and other organizations)

Social simulation

- *Agents as a tool for understanding human societies:*

Multiagent systems provide a new tool for **simulating societies**, which may help shed some light on various kinds of social processes.

Agent-based simulation

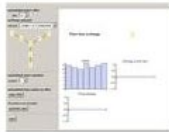
- Multi-agent systems provide strong models for representing complex and dynamic real-world environments.
 - modelling of the impact of climate change on biological populations
 - modelling the impact of public policy options on social or economic behaviour
 - modelling intelligent buildings
 - modelling traffic systems
 - modelling biological populations
-

Multi-agent Modeling

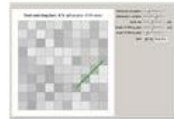
DEMONSTRATIONS

Demonstrations 1 - 18 of 18

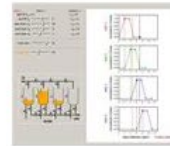
 [Subscribe to RSS feed](#)



Exchange Networks



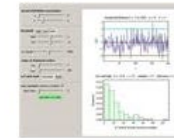
Fishing with Long Line or Gill Net?



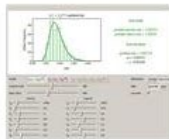
Estimating Processing Times in a Row of Storage Tanks



Ant Colony Optimization (ACO)



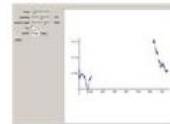
Estimating the Time between Mishaps from Quality Control Data



Assessing Total Risk from Interacting Factors



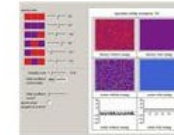
Boids: Simulated Flocking Behavior



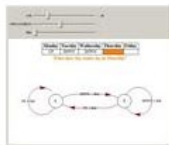
Mean Distance of Agents in Diffusion-Limited Aggregation



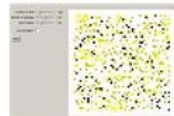
Sugarscape: Agent-Based Modeling



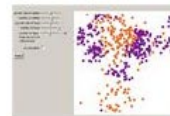
Emulating Land Use Evolution with a Cellular Automaton



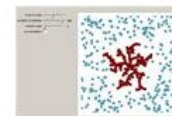
Trader Dynamics in Minimal Models of Financial Complexity



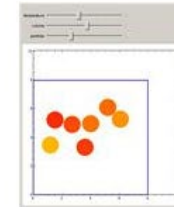
Garbage Collection by Ants



Predator-Prey Ecosystem: A Real-Time Agent-Based Simulation



Diffusion-Limited Aggregation: A Real-Time Agent-Based Simulation



Molecular Dynamics of Lennard-Jones Particles Using the Velocity Verlet Algorithm

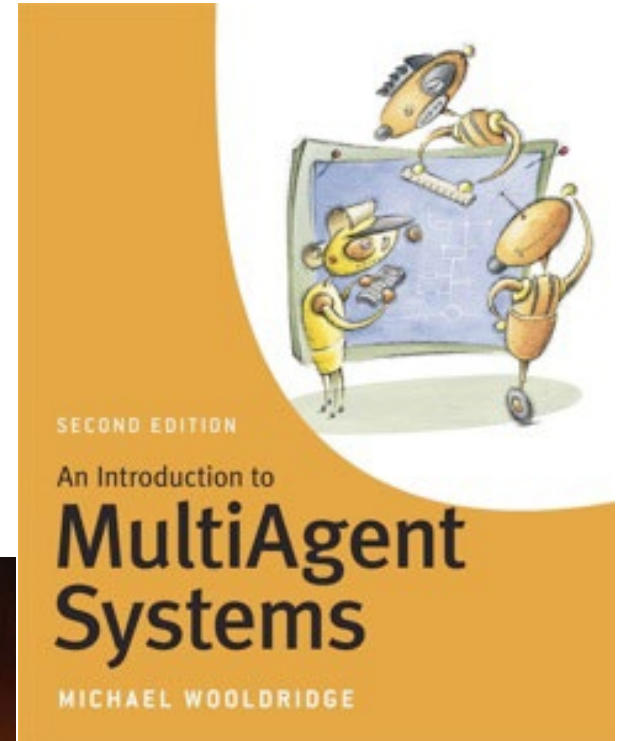
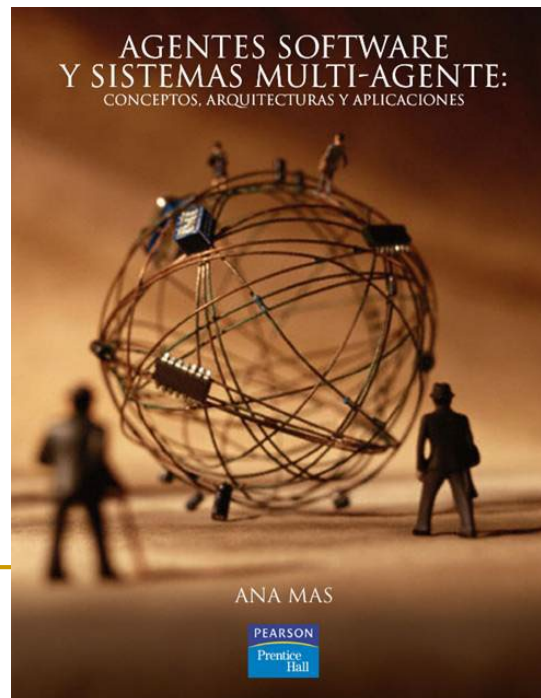
Research on MAS

- Organizations: [IFAAMAS](#)
- Conferences: [AAMAS](#), [PAAMS](#)
- Summer schools: [EASSS](#), [ISSIA](#)
- Journals: [Autonomous Agents and Multi-Agent Systems \(Springer\)](#)

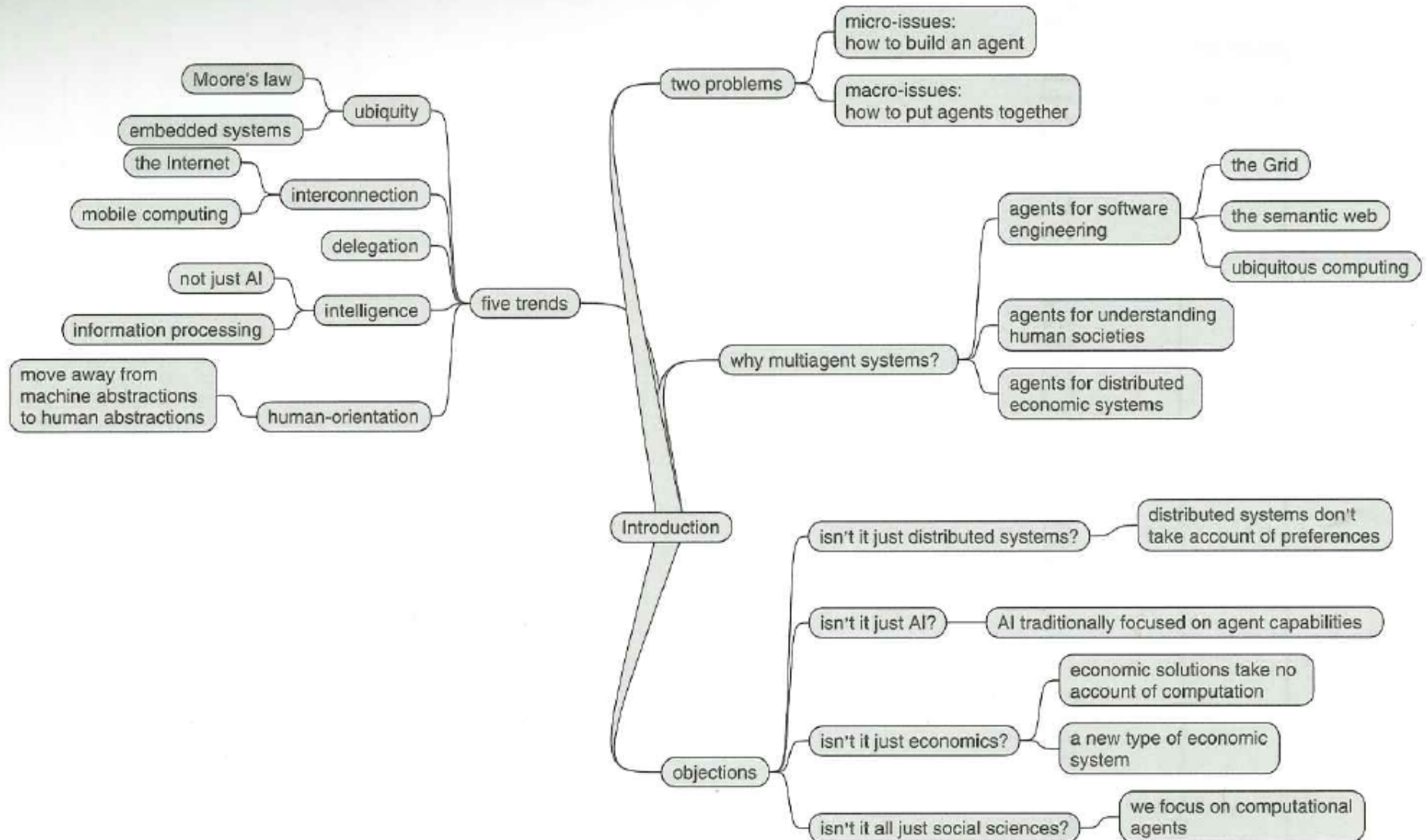


Readings for this week

- M.Wooldridge: *An introduction to MultiAgent Systems* – *chapter 1, beginning c.2*
- A.Mas: *Agentes software y sistemas multi-agente: conceptos, arquitecturas y aplicaciones* – *initial part of chapter 1*



Mindmap, chapter 1 (Wooldridge)



Videos (M.Wooldridge)

- Computer Science trends leading to Agent Technology.
 - Basic definition of agent and multiagent system.
 - Common objections to Agent Technology.
 - Objects vs. Agents.
-