
Intelligent Multiagent Systems

Francesco Amigoni

Course introduction (1)

- Web page:
<http://home.deib.polimi.it/amigoni/IntelligentMultiagentSystems.html>
- PhD course, 5 credits
- 7 classes, 4 hours each
 1. Monday, March 18, 2019, 9:00-13:00, sala seminari (ground floor, building 20)
 2. Monday, March 25, 2019, 9:00-13:00, sala seminari
 3. Wednesday, March 27, 2019, 9:00-13:00, sala seminari
 4. Friday, March 29, 2019, 9:00-13:00, sala seminari
 5. Monday, April 1, 2019, 9:00-13:00, sala seminari
 6. Monday, April 8, 2019, 9:00-13:00, sala seminari
 7. Wednesday, April 10, 2019, 9:00-13:00, sala seminari

Course introduction (2)

- What you are required to do (if you want a grade):
 - attend all classes (hard threshold: 70% of the classes)
 - read and present a paper (in classes 2-6)
 - develop and present (in class 7) a small software project
 - actively contribute to the classes
- Grades are from “rimandato” to “30 e lode”
 - Automatic conversion to A, B, ... for PhD students
- No other way to pass the exam or increase your grade: no additional projects, no oral exam, ...

Course introduction (3)

■ Paper presentations

- One student per paper
- 30 minutes (**strictly enforced!**)

■ Presentations outline

- Each presentation is intended to illustrate a problem or an algorithm in the context of the general framework of the course
- 2/3 days before each class, I will circulate the slides with the outline of the class and with the details of the problem/algorithm each presentation should address
- Presentations should be coordinated with the outline
- Presentation structure (suggested):
 - Main ideas of the proposed algorithms and techniques (few details!)
 - Simple examples of execution
 - How the proposed algorithms and techniques are shown to work
 - Pros and cons, critiques

Course introduction (4)

■ Paper assignment

- Papers are a representative sample of current and past research in multiagent systems
- Papers covering algorithms are for Computer Science and Engineering students
- Papers covering general approaches and problem formulations are for other students
- You will receive an email with detailed instructions very soon
 - Shared Google doc
 - First-come first-served
 - Indication of papers for students not from Computer Science and Engineering

Course introduction (5)

■ Software projects

- ❑ One student per project
- ❑ Two students only when one of the two has no programming experience
- ❑ Goal: addressing an application using an agent-oriented approach, not necessarily using agent technologies
- ❑ Small!
- ❑ Some ideas
 - JADE (<http://jade.tilab.com/>, latest version is 4.5)
 - Multi-Agent Programming Contest (<https://multiagentcontest.org/2018/>)
 - DCOP algorithms in Python (<https://pydcop.readthedocs.io/en/latest/implementation/algorithms.html>)
 - Agent-based modelling (<https://mesa.readthedocs.io/en/master/index.html>)
 - Other ideas at <http://home.deib.polimi.it/amigoni/IntelligentMultiagentSystems.html>

Course introduction (6)

- Project assignment
 - Up to you!
 - ...but please discuss with me the intended topic of the project, before implementing it
 - Project presentation
 - **Wednesday, April 10**, ~3 slides (and a short demo), 7-8 minutes per project
-

A few words about me

- Francesco Amigoni
 - francesco.amigoni@polimi.it
 - 02 2399-3475
 - Room 105, first floor, DEIB building 20
 - Associate professor at DEIB since 2007
 - Laurea magistrale (M.Sc.) courses:
 - Artificial Intelligence
 - Autonomous Agents and Multiagent Systems
 - Research on agents from mid-1990s
 - In the past: architectures for multiagent systems, modelling complex systems, ...
 - Currently: autonomous decision making in physical environments (agents are robots, appliances, satellites, ...)
-

Summary

- Examples and motivations
 - Definitions: agents and multiagent systems
 - Two key problems
 - “Views” of multiagent systems
 - Relations with other disciplines
 - Topics of this course
-

Some systems

1997 - Deep Blue

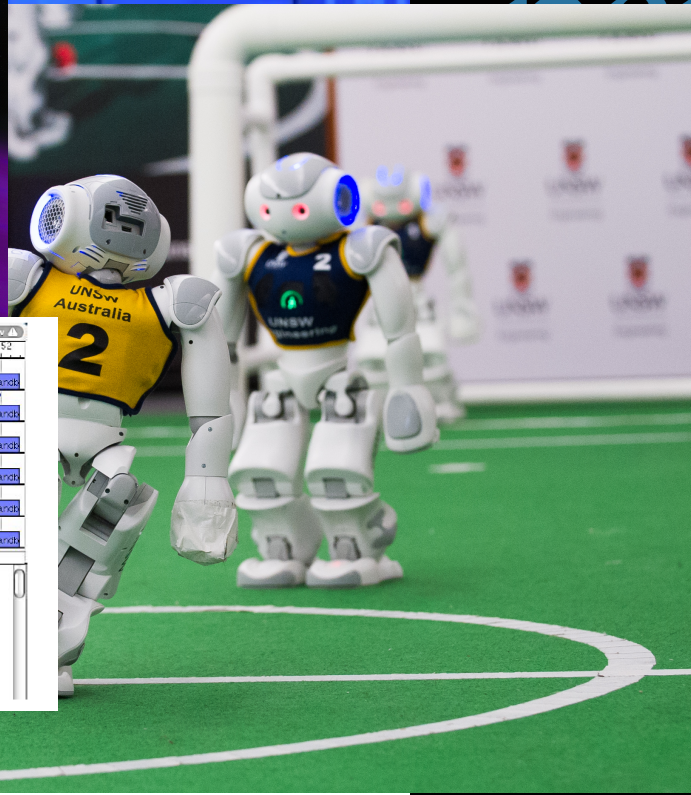
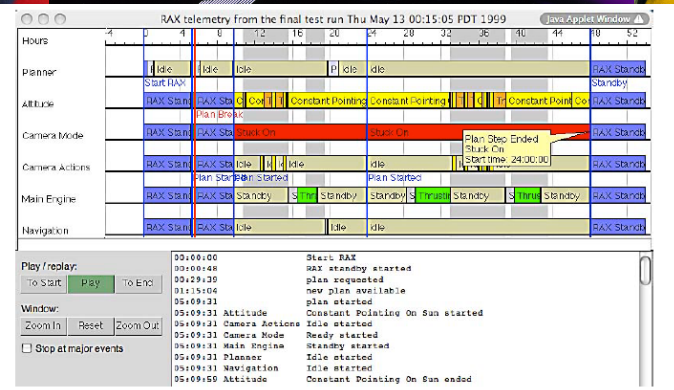
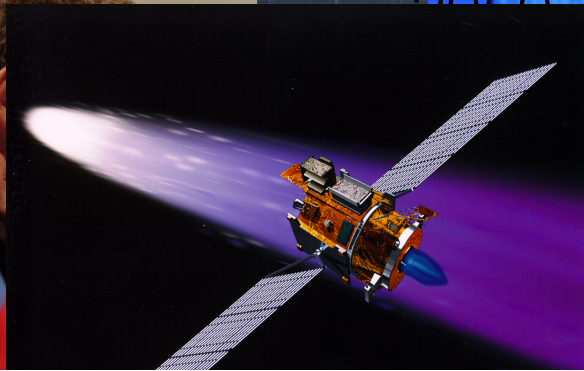
1998 - Deep Space 1

2009 - Google autonomous car

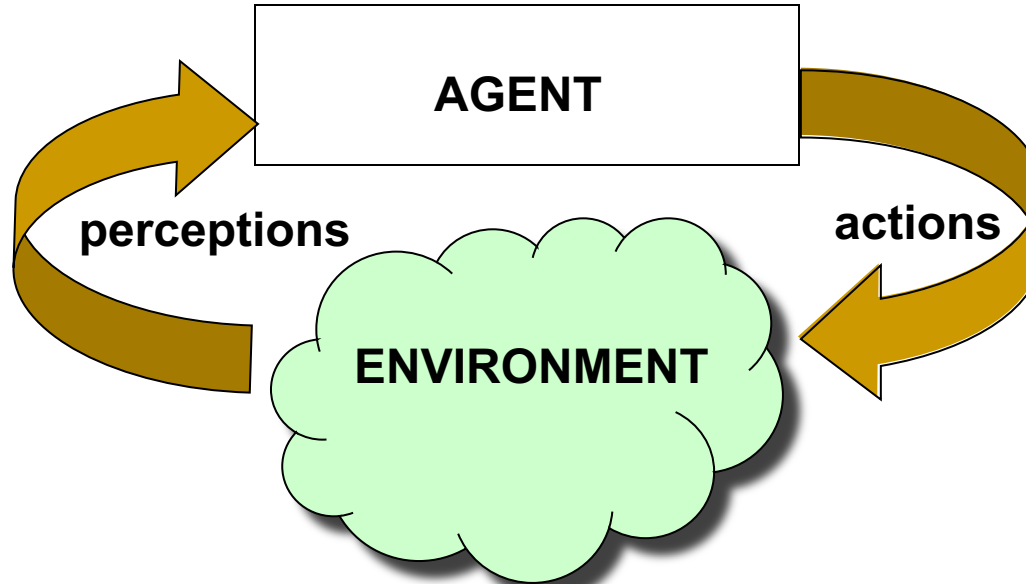
2011 - Watson

2016 - AlphaGo

2050 - RoboCup (?)

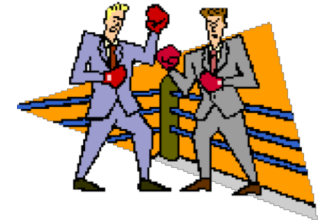


Agent



- *Agent*: robot, software program
- *Environment*: real or virtual (software environment)
 - Partially observable, non deterministic, dynamic, with other agents, ...

Defining an agent



- There is not “the” definition of agent
- Stuart Russell and Peter Norvig
 - “An agent is anything that can be viewed as *perceiving* its environment through sensors and *acting* upon that environment through effectors.”
- Pattie Maes
 - “Autonomous agents are computational systems that inhabit some complex dynamic environment, sense and act autonomously in this environment, and by doing so *realize a set of goals* or tasks for which they are designed.”

What is an agent?

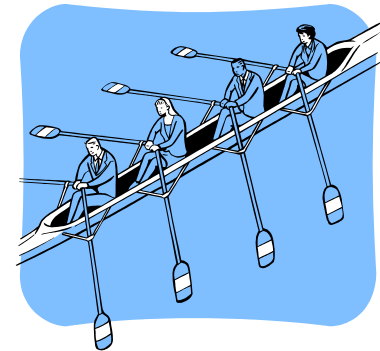
- An agent is characterized by listing some of its *properties*, according to Michael Wooldridge and Nicholas Jennings
 - Autonomy
 - Reactivity
 - Pro-activeness
 - Sociality
 - The examples show some of these properties
 - Other properties could be added: benevolence, rationality, ...
-

Multiagent systems

- A *multiagent system* is a system in which a number of agents interact with each other
- From the above examples, to successfully interact agents must cooperate, negotiate, coordinate, ...
- In general, different agents are pursuing different goals; only sometimes a global goal for the system can be identified

Two key problems

- How to design individual agents able to act autonomously in order to reach a goal (*micro problem, agent problem*)
- How to design systems in which more agents interact in a “useful” way (*macro problem, society problem*)



this course!

Multiagent systems as a design paradigm



- Modeling and design of distributed complex systems
- Agent architectures
 - Reactive, deliberative, hybrid
- Interaction mechanisms
 - Planning, coordination, matchmaking, auctions, negotiation strategies, ...
- Agent-oriented software engineering and agent-oriented programming
 - New abstractions: autonomous agent, cognitive agent, agent society, ...

Multiagent systems as a programming technology



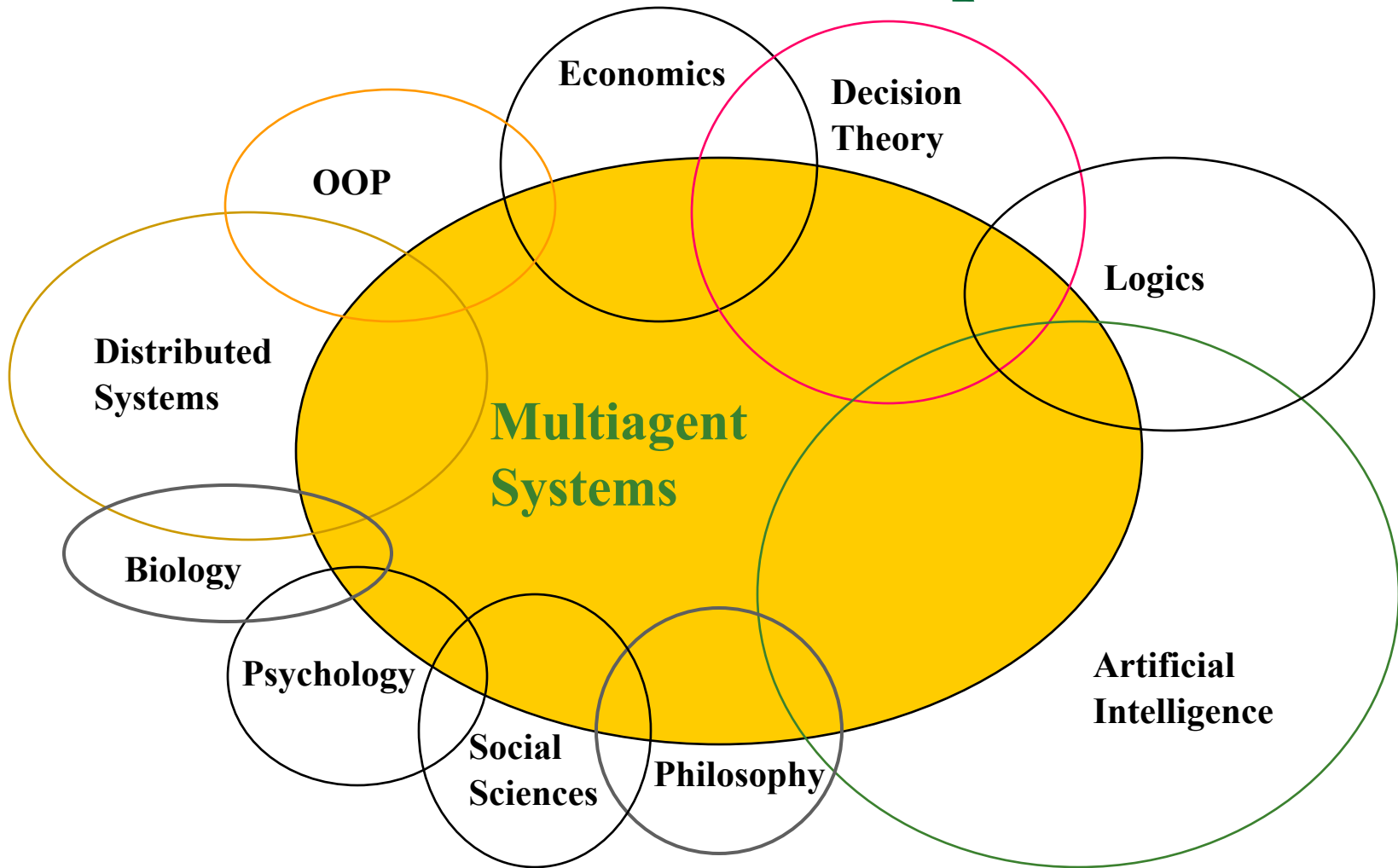
- Agent communication languages
 - FIPA ACL, KQML
- Ontologies
- Agent development frameworks
 - JADE, ZUES, ...

My personal view on agents:

More paradigm than technology

- Agents are a very useful abstraction to model a number of entities
 - Web services
 - Nodes in a grid
 - Robots
 - People
 - ...
 - Multiagent systems provide “general” techniques, which are rather independent from the possible implementation
 - Negotiation
 - Task allocation
 - Coordination
 - ...
-

Relations with other disciplines



Some problems...

- Agents vs. objects
 - An object's method is *invoked*, an agent is *requested* to execute an action
- Agents vs. distributed systems
 - Agents can be *self-interested*, their interactions being of the type studied in economics
- Agents vs. artificial intelligence
 - An agent includes different aspects of “intelligence”: planning, learning, ...
 - Artificial intelligence usually does not consider social interactions
- Agents vs. game theory
 - Game theory is usually only *descriptive*

Publication venues for agents

- Specific and transversal venues
- Journals
 - Journal of Autonomous Agents and Multi-Agent Systems (JAAMAS)
 - Artificial Intelligence Journal (AIJ), Journal of AI Research (JAIR), Journal of Machine Learning Research, IEEE Transactions on Robotics, Autonomous Robots, Robotics and Autonomous Systems, ...
- Conferences
 - Autonomous Agents and Multi-Agent Systems (AAMAS)
 - AAI, IJCAI, ...
 - ICRA, IROS, RSS, DARS, MRS, IAS, ...

Topics of this course

- In this course we will focus on some (not all!) aspects of *cooperative interaction* in multiagent systems
 - Concepts
 - Problems
 - Algorithms
 - Programming (project)
- Class schedule:
 1. Course introduction, agents and multiagent systems, agent architectures, motivating examples
 2. Basics of multiagent interactions
 3. Multiagent planning
 4. Distributed constraint handling
 5. Task allocation
 6. Multiagent path finding, other issues (consensus, swarms, ...)
 7. Project presentations and applications

What we leave out

- Game theory
 - Logics
 - Argumentation
 - Trust and reputation
 - Software engineering aspects (except the project): specification, validation, ...
 - ...
-