Intelligent Multiagent Systems

Francesco Amigoni
Course introduction (1)

- 5 credits
- 5 classes, 4 hours each
  1. Monday 27/4/2015, 9:00-13:00, sala seminari (ground floor, building 20)
  2. Tuesday 12/5/2015, 9:00-13:00, sala seminari (ground floor, building 20) → Monday 18/5/2015, 9:00-13:00
  3. Thursday 14/5/2015, 9:00-13:00, sala seminari (ground floor, building 20)
  4. Wednesday 20/5/2015, 14:00-18:00, sala seminari Alessandra Alario (fourth floor, building 21)
  5. Friday 22/5/2015, 9:00-13:00, sala seminari Alessandra Alario (fourth floor, building 21)
What you are required to do (if you want a grade):
- attend all classes (except one)
- read and present a paper (in classes 2, 3, and 4)
- develop and present a small software project (in class 5)
- actively contribute to the discussion (in all the classes!)
Course introduction (3)

- Paper presentations
  - One or two students per paper
  - 20 minutes (strictly enforced!)

- Presentation structure
  - Problem addressed by the paper
  - Motivation: application, gap in the literature, …
  - Main ideas of the proposed algorithms and techniques (no details!)
  - Simple examples of execution
  - How the proposed algorithms and techniques are shown to work
  - Pros and cons, critiques

- Paper assignment
  - Papers are a representative sample of current and past research in multiagent systems
  - You will receive an email with instructions very soon (Wednesday)
Course introduction (4)

- **Software projects**
  - One or two students per project
  - JAVA and JADE ([http://jade.tilab.com/](http://jade.tilab.com/), latest version is 4.3.3)
  - Small!
    - Electronic commerce (auctions, bargaining, ...)
    - Simulated mobile robots (information exchange, task allocation, ...)
    - Crosswords
    - Recommendation systems
    - ... (see other ideas at [http://home.dei.polimi.it/amigoni/IntelligentMultiagentSystems.html](http://home.dei.polimi.it/amigoni/IntelligentMultiagentSystems.html))

- **Project presentation**
  - **Last class**, ~3 slides per project (and a short demo), 6-7 minutes

- **Project assignment**
  - Up to you!
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, cooperative negotiation to model 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
First example: controlling spacecrafts

- Ground mission centers cannot always react promptly to unexpected events onboard of spacecrafts
- Spacecrafts should be *autonomous* in making decisions
- NASA’s Deep Space 1
Second example: searching for information

- Searching the Internet is boring, time consuming, and not always easy
- As users, we would like to have a software program that, on our behalf, searches the Internet, collects data from different sources, filters them, and presents the results
Third example: online reservations

- During a rigid winter, you would like to take a break in a warm place
- You would like to tell your PDA (Personal Digital Assistant) your requirements and wait while it Negotiates with some web sites to book hotels and flights to assemble a complete vacation package
Fourth example: robot soccer team

- A team of robotic soccer players play a game against another robotic soccer team.
- Robots should *cooperate* with teammates, but should *compete* with adversaries to win the game.
- Robocup
  (from [http://www.robocup.org](http://www.robocup.org))
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 four 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 four 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)
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
  - …
Aspects of agents (by keywords)

- **Self-awareness**
  - “Self-awareness is concerned with the availability, collection and representation of knowledge about something, by that something”

- **Cyber-physical systems**
  - “These are integrations of computational, network, and physical systems, whose operations are monitored, coordinated, controlled and integrated by a computing and communication core”
Relations with other disciplines

- Economics
- Decision Theory
- Logics
- Artificial Intelligence
- Multiagent Systems
- OOP
- Distributed Systems
- Biology
- Psychology
- Social Sciences
- Philosophy
- Biology
- Philosophy
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), …
- Conferences
  - Autonomous Agents and Multi-Agent Systems (AAMAS), Intelligent Agent Technology (IAT)
  - AAAI, IJCAI, …
  - ICRA, IROS, DARS, IAS, …
Topics of this course

- In this course we will focus on some (not all!) aspects of interaction in multiagent systems
  - Concepts
  - Algorithms
  - Programming (a little bit)

Class organization:
1. Course introduction, agents and multiagent systems, agent architectures
2. Agent communication and agent interaction: cooperation, coordination, negotiation
3. Agent interaction: cooperative search, distributed constraint optimization, multiagent planning, voting
4. Agent interaction: task allocation, social laws, formation control, swarms: emerging behaviors
5. Project presentations, development frameworks, applications
What we leave out

- Game theory
- Logics
- Learning
- Software engineering aspects (except the project): specification, validation,