Escola Tècnica Superior d'Enginyeria
Universitat Rovira i Virgili

[DEIM]

Lecture 1: Introduction to Agents and Multi-Agent Systems

Multi-Agent Systems

Universitat Rovira i Virgili

MESIIA – Master's Degree in Computer Security Engineering and Artificial Intelligence
MAI – Master's Degree in Artificial Intelligence

Outline

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

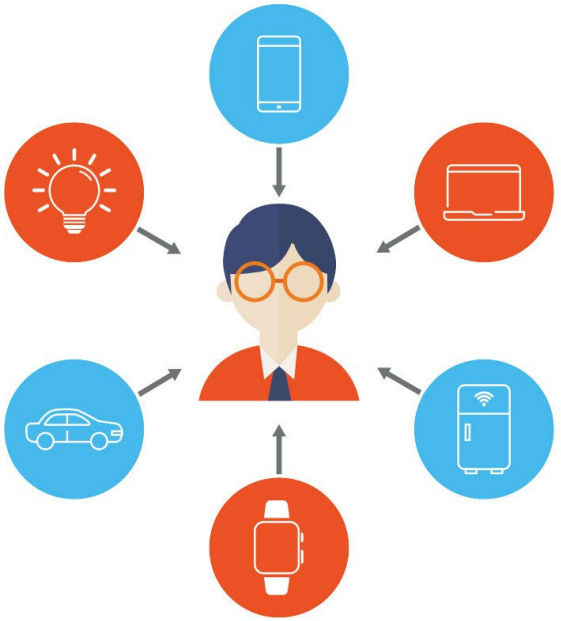
1. Main trends in Computer Science

Five trends are marking the history of computing:

- *Ubiquity*
- *Interconnection*
- *Intelligence*
- *Delegation*
- *Human-orientation in programming methodologies*

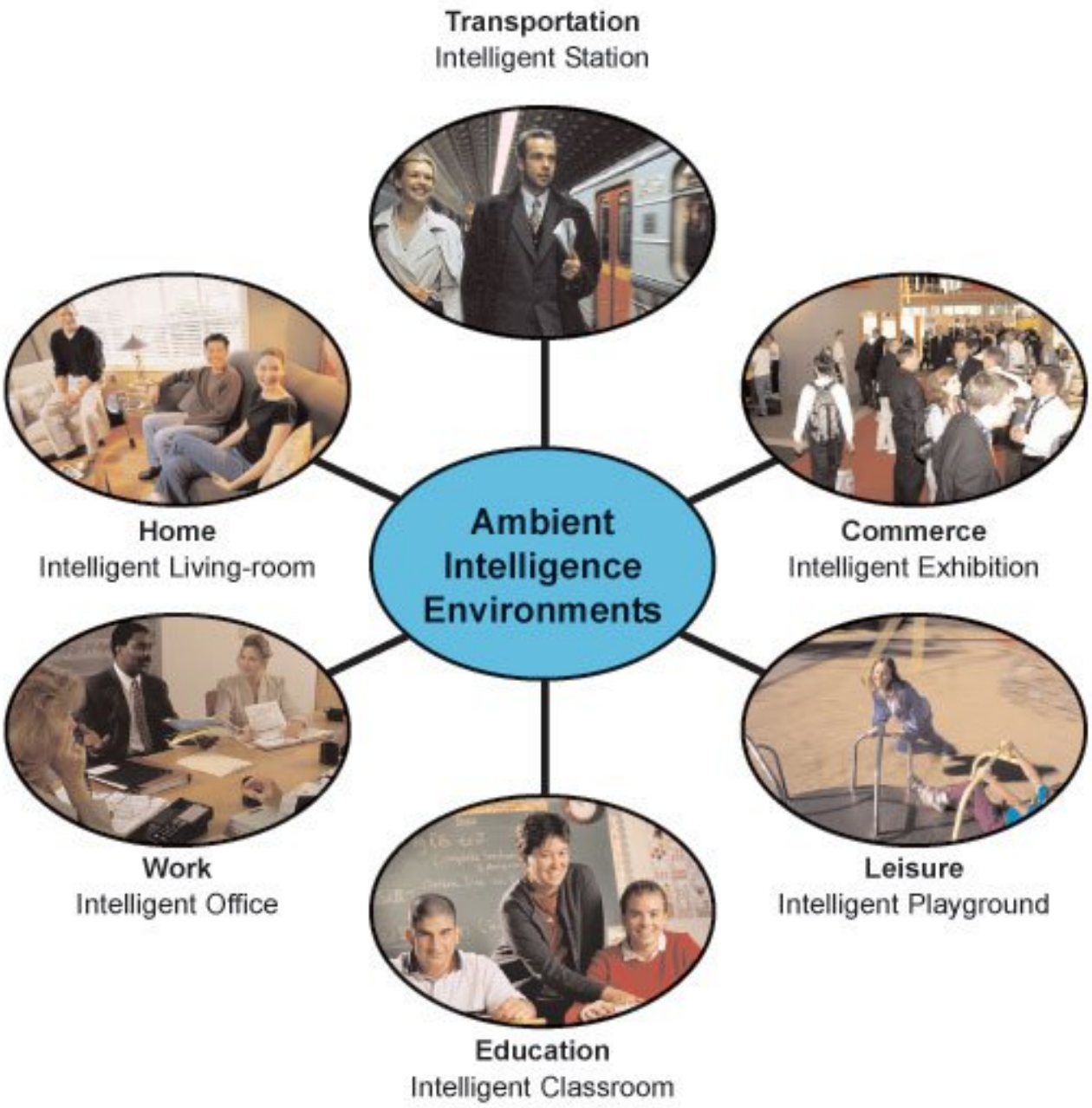
Ubiquity

- 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, IoT-Devices, ...)
- 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

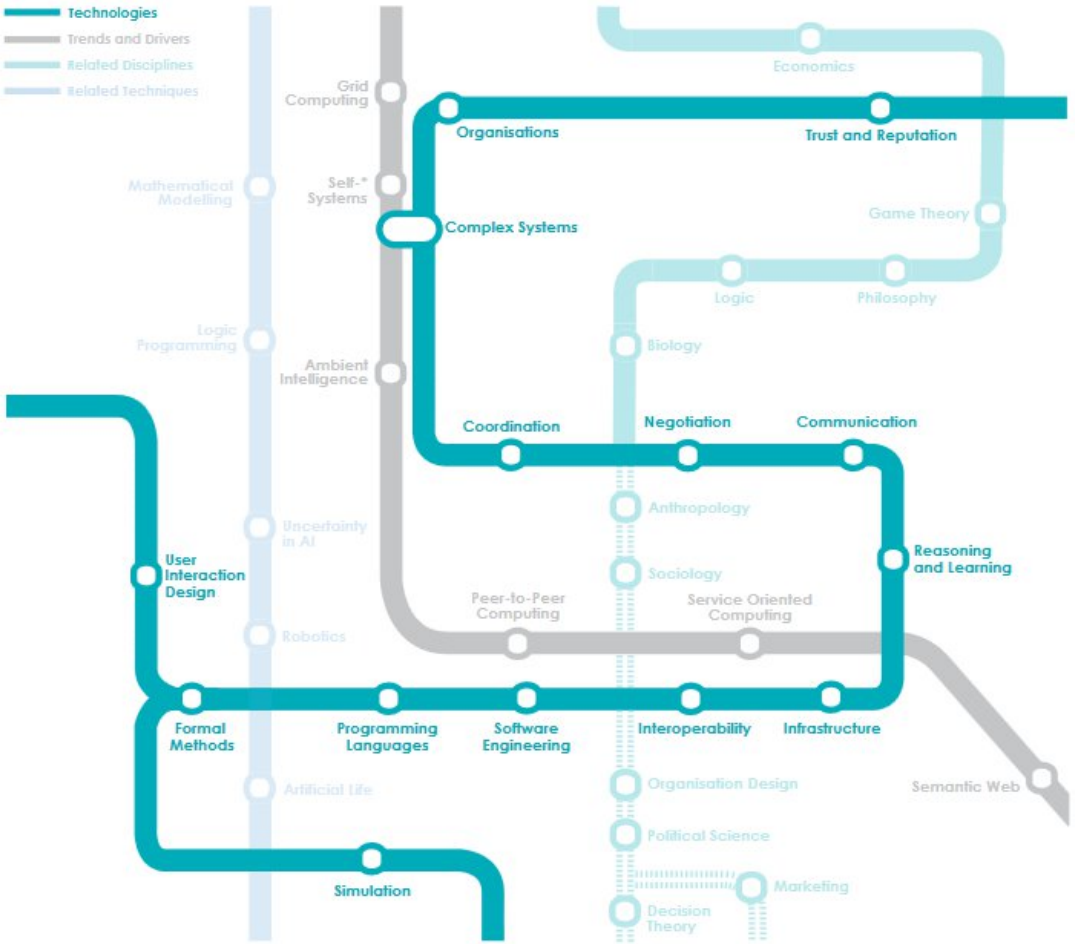

Transportation
Intelligent Station
Home
Intelligent Living-room
Ambient Intelligence Environments
Commerce
Intelligent Exhibition
Work
Intelligent Office
Leisure
Intelligent Playground
Education
Intelligent Classroom

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

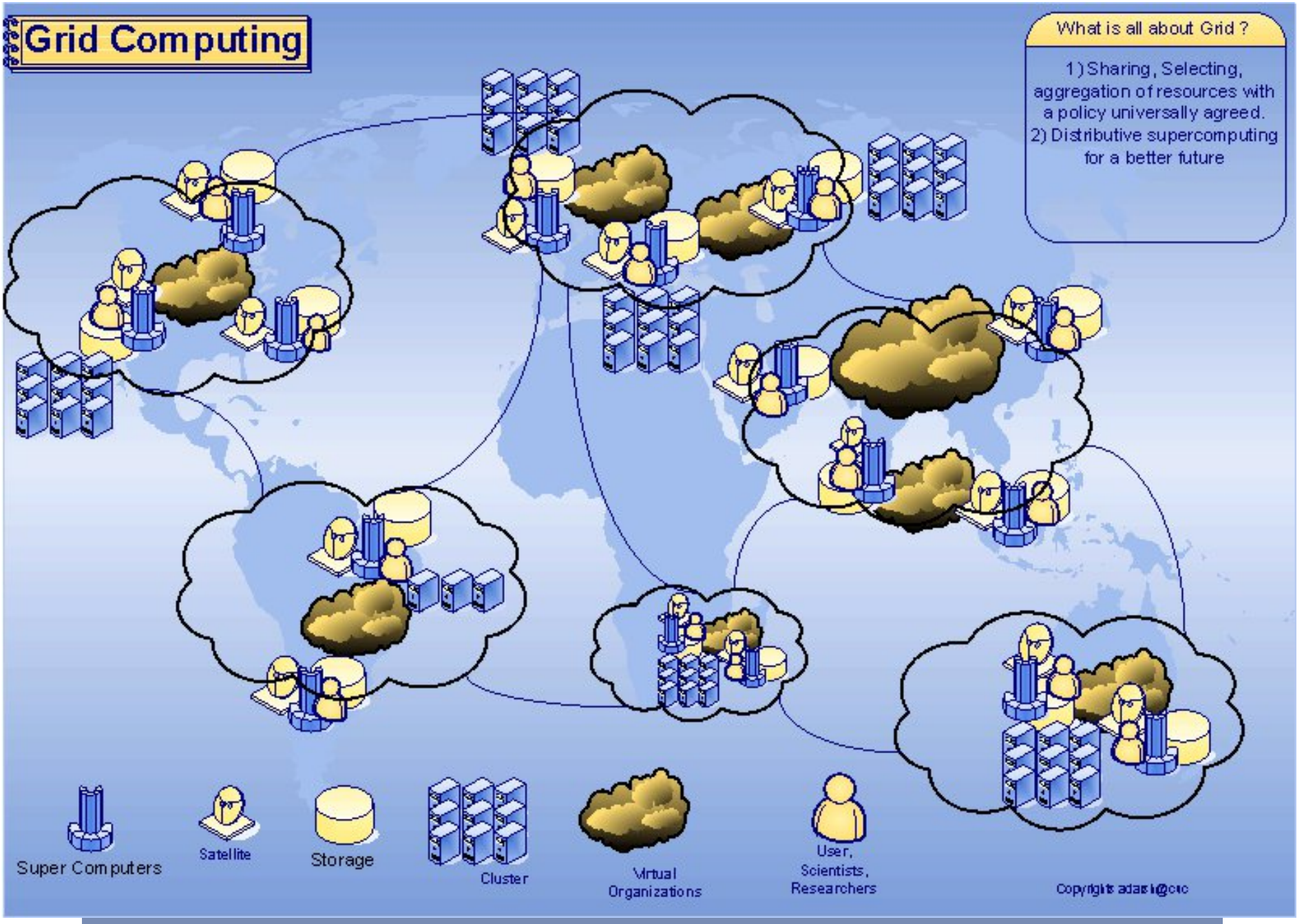


Compiled, written and edited by

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
Super Computers
Satellite
Storage
Cluster
Virtual Organizations
User, Scientists, Researchers

Intelligence

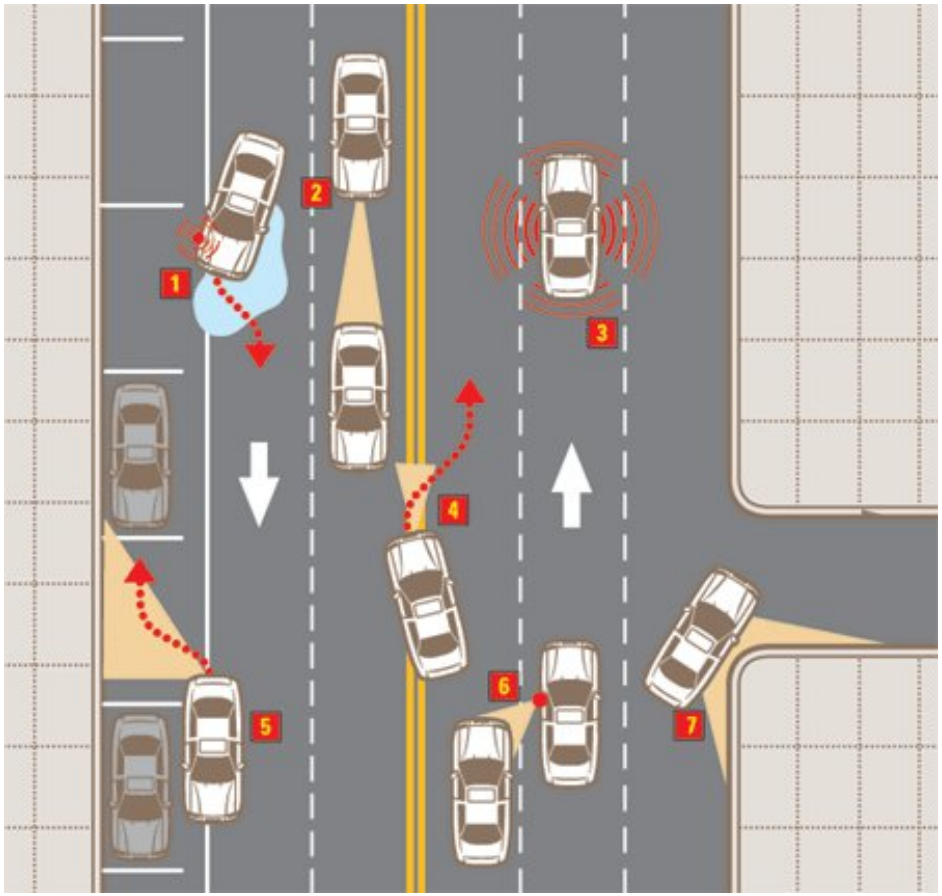
- The complexity of tasks that we are capable of automating and delegating to computers has grown steadily
- That requires an increasing amount of domain knowledge and inference capabilities, in short, more intelligent problem solving techniques

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 judgement may be trusted more than an experienced pilot
- Already being worked on: self-driving cars, intelligent braking systems, cruise control that maintains distance from car in front…

Example: Autonomous vehicles

- 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
  - Objects

    to *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 *Multi-Agent Systems*

2. Agents and Multi-Agent 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]

Multi-Agent Systems

- A Multi-Agent 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 10]

Multi-Agent Systems

- In Multi-Agent 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?

Multi-Agent Systems

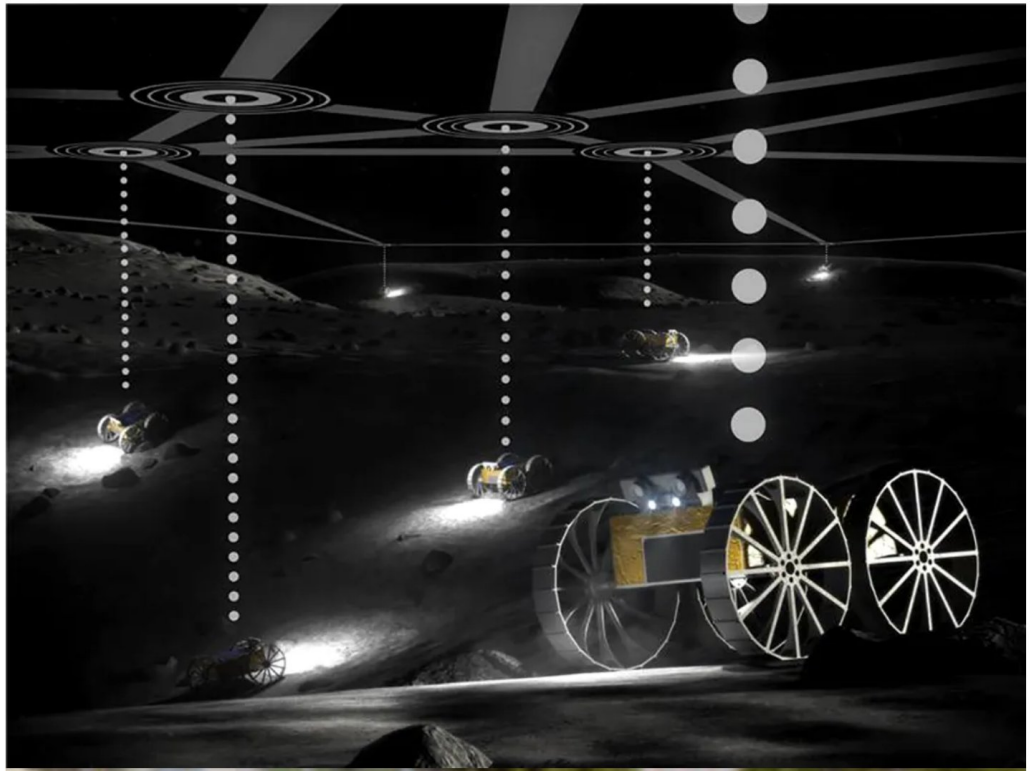
- While these questions are all addressed in part by other disciplines (notably Economics and Social Sciences), what makes the Multi-Agent systems field unique is that it emphasizes that the agents in question are *computational*, *information processing* entities

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

Example 1 – Spacecraft Control

Cooperative Autonomous Distributed Robotic Explorers (CADRE)



https://www.nasa.gov/centers-and-facilities/langley/cooperative-autonomous-distributed-robotic-explorers-cadre/

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

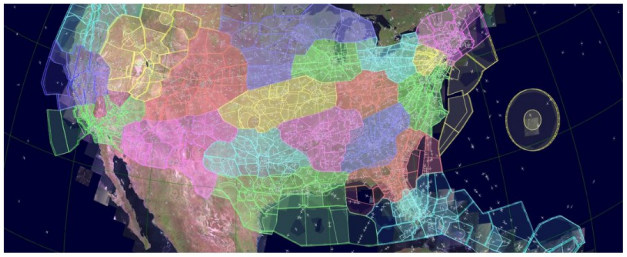
AgentFly Technologies

Home Technology Services About us Contact

AgentFly ATM Simulation | Model of Actors and Environment | What AgentFly ATM Offers

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 architecture of the system is highly modular, widely configurable and flexible, and it allows easy creation of scenarios.

https://www.agentfly.com/atm-modeling-and-simulation

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

Multi-Agent Systems is Interdisciplinary

- The field of Multi-Agent 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

- 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)

3. Objections to MAS

- 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

3. Objections to MAS

- Isn't it all just Economics/Game Theory?
- These fields also have a lot to teach us in Multi-Agent 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

3. Objections to MAS

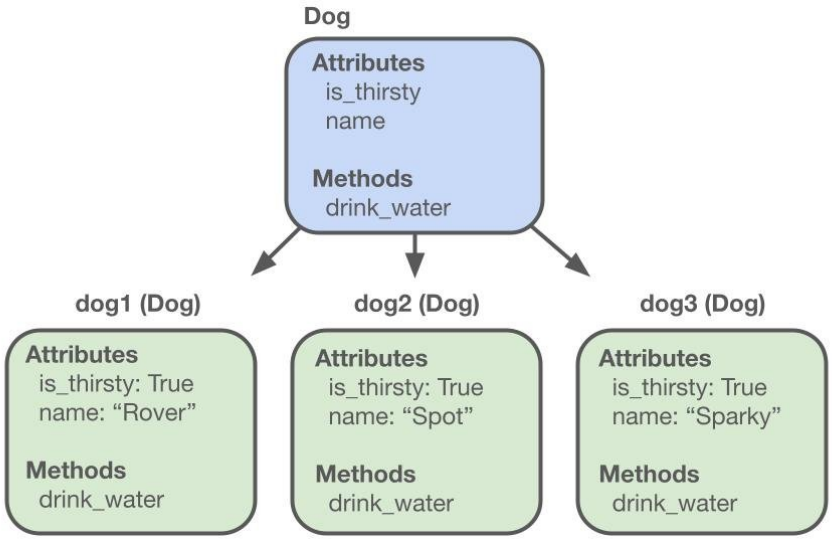
- Isn't it all just Social Science?
  - We can draw insights from the study of human societies, but there is no particular reason to believe that artificial societies will be constructed in the same way
  - Again, we have inspiration and cross-fertilization, but hardly subsumption

4. Agents vs 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 vs 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 its state



Agents vs Objects

- 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

- 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

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 vs 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 vs 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. Viewpoints on agent technology

- 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 a 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 a 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:

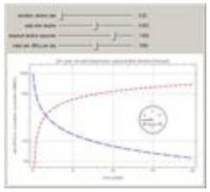
  Multiagent systems provide a new tool for simulating societies, which may help shed some light on various kinds of social processes

MULTI-AGENT MODELING

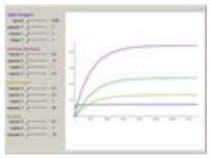
DEMONSTRATIONS

Demonstrations 1 - 20 of 20

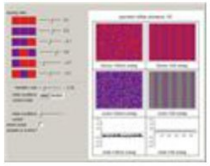
Subscribe to RSS feed



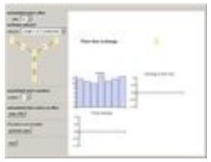
Oil Well Decline Modeling Using Hyperbolic-Exponential Forecasting



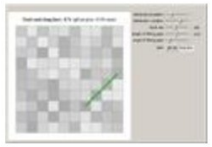
Task Partitioning in Insect Societies: A Model Based on Electric Circuits



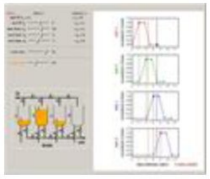
Emulating Land Use Evolution with a Cellular Automaton



Exchange Networks

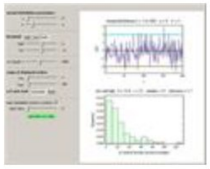


Fishing with Long Line or Gill Net?

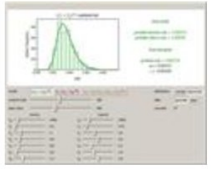


Estimating Processing Times in a Row of Storage Tanks

Ant Colony Optimization (ACO)

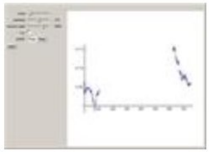


Estimating the Time between Mishaps from Quality Control Data



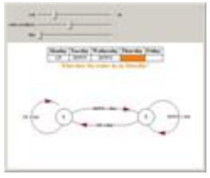
Assessing Total Risk from Interacting Factors

Boids: Simulated Flocking Behavior

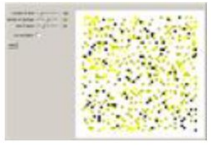


Mean Distance of Agents in Diffusion-Limited Aggregation

Sugarscape: Agent-Based Modeling



Trader Dynamics in Minimal Models of Financial Complexity



Garbage Collection by Ants

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

https://demonstrations.wolfram.com/topic.html?topic=Multi-agent+Modeling&limit=20

Research on MAS

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

Proposed readings

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