1: Introduction to Multi-Agent System

AI6125: Multi-Agent System

Assoc Prof Zhang Jie

Based on "An Introduction to MultiAgent Systems" by Michael Wooldridge, John Wiley & Sons, 2002/2009

Overview

- Motivation for agents – natural emergence
- Definitions
- The future of agents
- Views of the field (in comparison to others)
- Applications of individual agents and multi-agent systems

Trends

- Five ongoing trends have marked the history of computing:
  - *ubiquity*;
  - *interconnection*;
  - *intelligence*;
  - *delegation*; and
  - *human-orientation*

Ubiquity (Pervasiveness)

- The continual reduction in cost of computing capability has made it possible to introduce processing power into places and devices that would have once been uneconomic
- As processing capability spreads, sophistication (and intelligence of a sort) becomes ubiquitous
- What could benefit from having a processor embedded in it…?

Ubiquity (Pervasiveness)

Ubiquity (Pervasiveness)

Interconnection

- Computer systems today no longer stand alone, but are networked into large distributed systems
- The internet is an obvious example, but networking is spreading its ever-growing tentacles…
- 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

Interconnection

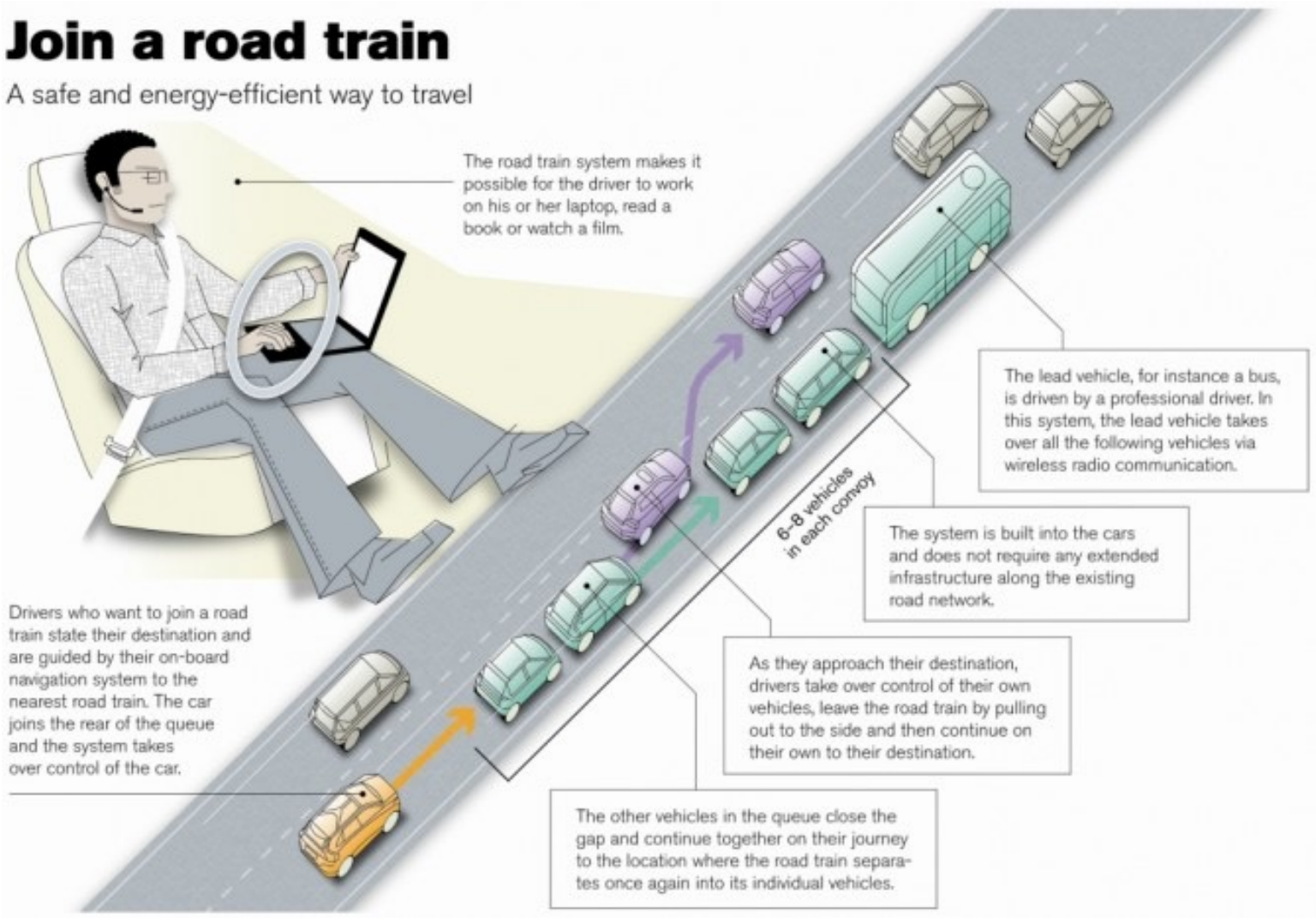
Intelligence

- By Intelligence we mean:
  - The complexity of tasks that we are capable of automating and delegating to computers has grown steadily
- If you don’t feel comfortable with this definition of “intelligence”, it’ s probably because you are a human

Delegation

- Computers are doing more for us – without our intervention (this ties in with Intelligence)
- 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: fly-by-wire cars (done), intelligent braking systems, cruise control that maintains distance from car in front…

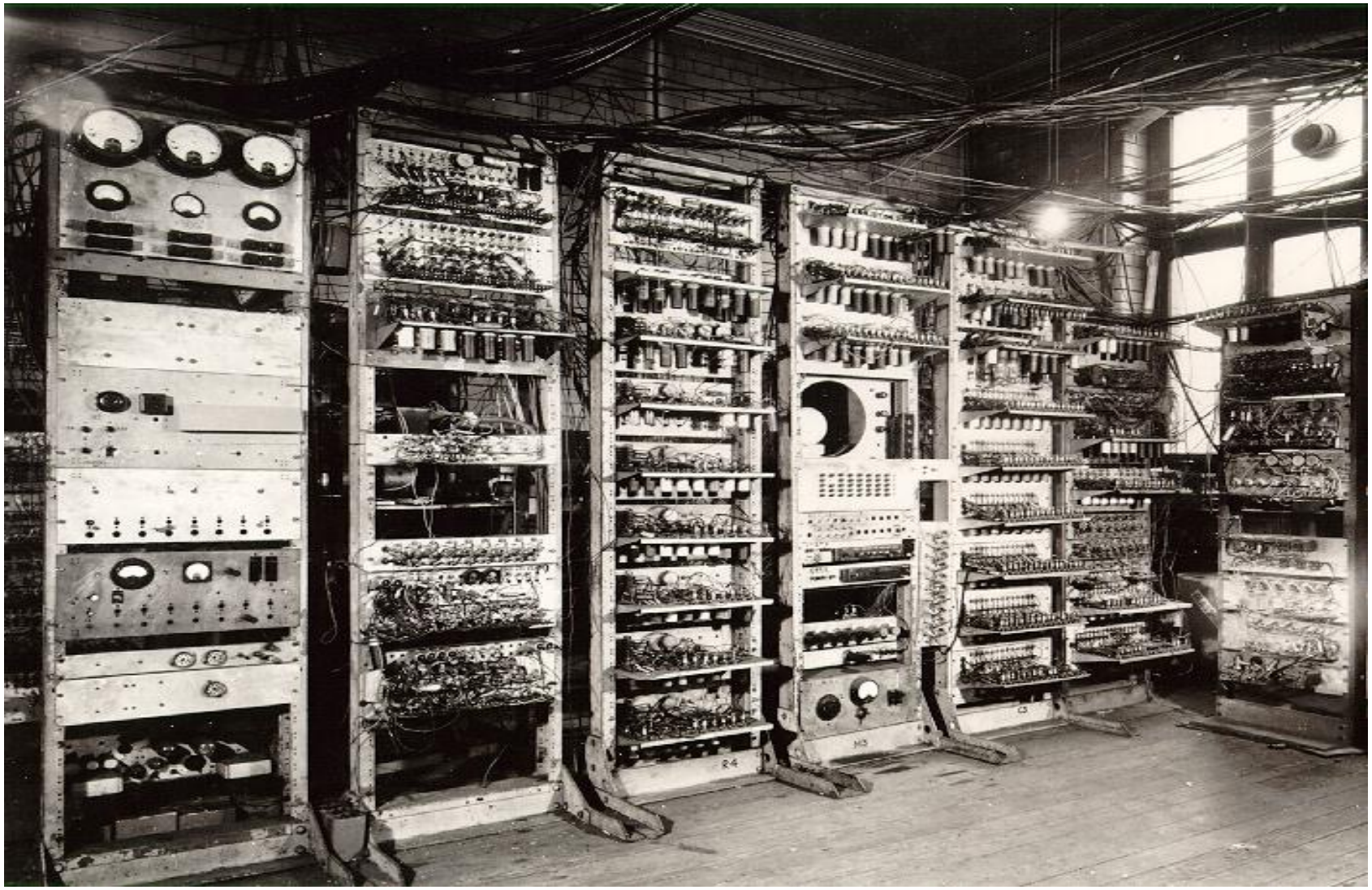
Delegation


Join a road train
A safe and energy-efficient way to travel
The road train system makes it possible for the driver to work on his or her laptop, read a book or watch a film.
Drivers who want to join a road train state their destination and are guided by their on-board navigation system to the nearest road train. The car joins the rear of the queue and the system takes over control of the car.
6–8 vehicles in each convoy
The lead vehicle, for instance a bus, is driven by a professional driver. In this system, the lead vehicle takes over all the following vehicles via wireless radio communication.
The system is built into the cars and does not require any extended infrastructure along the existing road network.
As they approach their destination, drivers take over control of their own vehicles, leave the road train by pulling out to the side and then continue on their own to their destination.
The other vehicles in the queue close the gap and continue together on their journey to the location where the road train separates once again into its individual vehicles.

Human Orientation

- 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

Human Orientation



Programming Progression…

- Programming has progressed through:
  - machine code;
  - assembly language;
  - machine-independent programming languages;
  - sub-routines;
  - procedures & functions;
  - abstract data types;
  - objects;

to *agents*.

Global Computing

- What techniques might be needed to deal with systems composed of 10^{10} processors?
- Don't be deterred by its seeming to be "science fiction"
- Hundreds of millions of people connected by email once seemed to be "science fiction"…
- Let's assume that current software development models can't handle this…

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…

- These issues were not studied in Computer Science until recently
- All of these trends have led to the emergence of a new field in Computer Science: *multiagent systems*

Agents, a Definition(s)

1. An agent is a computer system that 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).
2. An *agent* is a computer system that is *situated* in some *environment*, and that is capable of *autonomous action* in the environment in order to meet its delegated objectives.

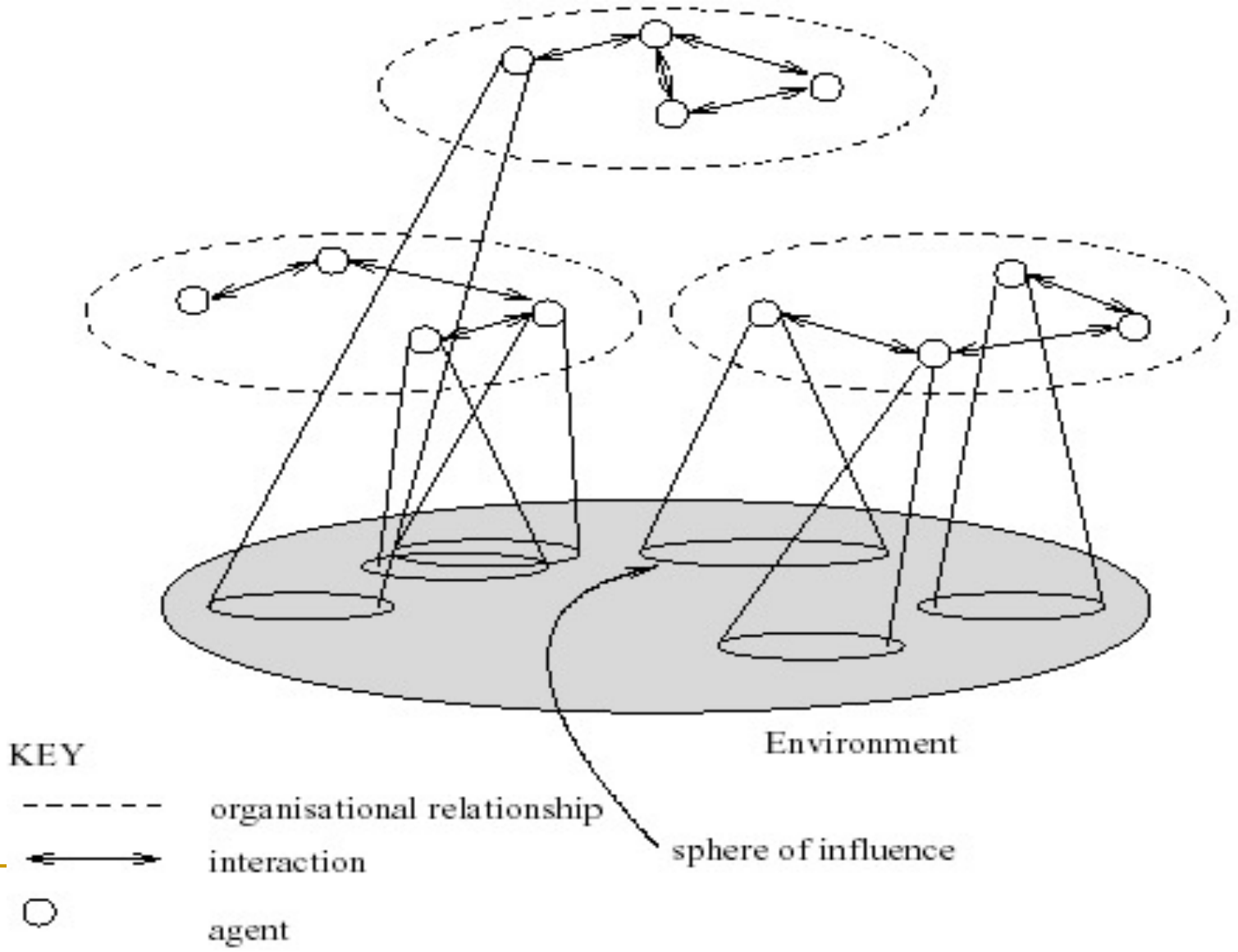
Agents, Definition(s) … a caveat.

- One embarrassing issue for the agent community (for many years) is the lack of a standardized definition.
- *Agent* means many things to many people.
- Recommended Reading:
  - ‘Is it an Agent, or Just a Program?’ (Franklin and Graesser 1997).

Multi-agent Systems, a Definition

- A multi-agent system is one that consists of a number of agents, which *interact* with one-another
- 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

What are *Multiagent* Systems?



Multiagent Systems

Thus a multiagent system contains a number of agents…

- …which interact through communication…
- …are able to act in an environment…
- …have different "spheres of influence" (which may coincide)…
- …will be linked by other (organizational) relationships

Agent Design, Society Design

- The course covers two key problems:
  - How do we build agents capable of independent, autonomous action, so that they can successfully carry out tasks we delegate to them?
  - How do we build agents that are capable of interacting (cooperating, coordinating, negotiating) with other agents in order to successfully carry out those delegated tasks, especially when the other agents cannot be assumed to share the same interests/goals?
- The first problem is *agent design*, the second is *society design* (micro/macro)

Multiagent 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.

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 a strength
  - Infusing well-founded methodologies into the field

The Views/Visions

- This can also be a weakness
  - Many different views as to what the field is about
  - Different researchers have different visions
  - The amalgamation of these visions (and research directions, and methodologies, and interests, and…) define the field
  - But the researchers clearly have enough in common to consider each other's work relevant to their own
- This has analogies with artificial intelligence itself

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
- Over the last two decades, a major Computer Science research topic has been the development of tools and techniques to model, understand, and implement systems in which interaction is the norm

Some Views of the Field

- *Agents as a tool for understanding human societies*:
  Multiagent systems provide a novel new tool for simulating societies, which may help shed some light on various kinds of social processes.
- This has analogies with the interest in "theories of the mind" explored by some artificial intelligence researchers

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

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

Objections to MAS

- 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

- 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

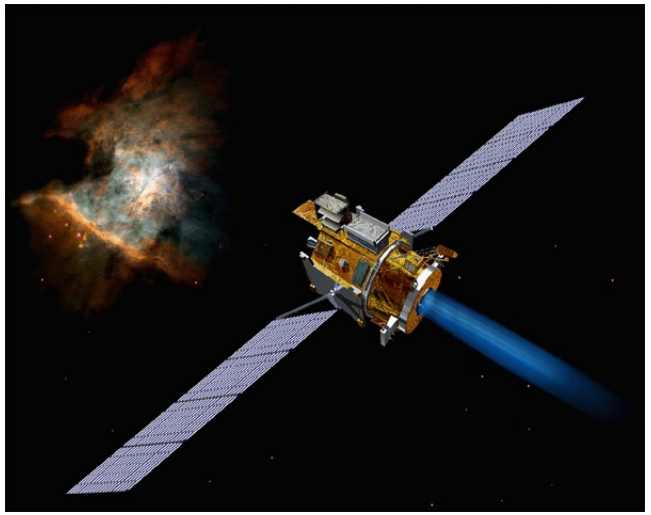
Applications – Individual Agents

- Individual/single agents are indicated for domains where *autonomous action* is required.
  - Agents (and their physical instantiation in robots) have a role to play in high-risk situations, unsuitable or impossible for humans
    - Spacecraft control
  - Personal software assistants: Play the role of proactive assistants to users working on some applications
    - Interface agents; Internet agents; E-commerce agents
- The degree of autonomy will differ depending on the situation (human intervention may be an alternative, but not always)

Spacecraft Control

- When a space probe makes its long flight from Earth 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 practical. For these reasons, organizations like NASA are seriously investigating the possibility of making probes more autonomous — giving them richer decision making capabilities and responsibilities.
- *This is not fiction:* NASA's *DS1 has done it!*

Example: Deep Space 1



- http://nmp.jpl.nasa.gov/ds1/
- "Deep Space 1 launched from Cape Canaveral on October 24, 1998. During a highly successful primary mission, it tested 12 advanced, high-risk technologies in space. In an extremely successful extended mission, it encountered comet Borrelly and returned the best images and other science data ever from a comet. During its fully successful hyperextended mission, it conducted further technology tests. The spacecraft was retired on December 18, 2001." – NASA Web site

Interface Agents

- The idea is to move away from the direct manipulation paradigm that has dominated for so long.
- Agents sit ‘over’ applications, watching, learning, and eventually doing things without being told — taking the initiative.
- Pioneering work at MIT Media Lab (Pattie Maes):
  - news reader;
  - web browsers;
  - mail readers.

Nicholas Negroponte's Vision

'The 'agent' answers the phone, recognises the callers, disturbs you when appropriate, and may even tell a white lie on your behalf. The same agent is well trained in timing, versed in finding opportune moments, and respectful of idiosyncracies. ' (p150)

'If you have somebody who knows you well and shares much of your information, that person can act on your behalf very effectively. If your secretary falls ill, it would make no difference if the temping agency could send you Albert Einstein. This issue is not about IQ. It is shared knowledge and the practice of using it in your best interests.' (p151)

'Like an army commander sending a scout ahead . . . you will dispatch agents to collect information on your behalf. Agents will dispatch agents. The process multiplies. But [this process] started at the interface where you delegated your desires.' (p158)

(From Being Digital, 1995.)

Example: Email Reading Assistants

- MAXIMS (Pattie Maes, 1994) 'learns to prioritize, delete, forward, sort, and archive mail messages on behalf of a user . . . '.
- Works by 'looking over the shoulder' of a user, and learning about how they deal with email.
- Each time a new event occurs (e.g., email arrives), MAXIMS records the situation -> action pairs generated
- Situation characterised by features of event:
  - sender of email; recipients; subject line; etc.
- When new situation occurs, MAXIMS matches it against previously recorded rules.

Email Reading Assistants

- Predicts user action, and generates a confidence level
- Confidence level compared against two threshholds: “tell me” and “do it”:
  - confidence < “tell me”: agent gets feedback
  - “tell me” < confidence < “do it”: agent makes suggestion
  - confidence > “do it”: agent acts
- Rules can be “hard coded”; even get help from other users.
- MAXIMS has a simple ‘personality’, (a face icon), communicating its ‘mental state’ to the user.

Internet Agents

- It is not easy to find the right information (even with the help of search engines).
- Systematic searches by human users are difficult:
  - human factors: we get bored by slow response times, find it difficult to read the WWW rigorously, get tired, miss things easily, misunderstand, get sidetracked;
  - organizational factors: structure on the net is superficial — no standards for home pages, not (yet) semantic markup to tell you what a page contains;
- The sheer amount of information presented to us leads to 'information overload'.

Internet Agents

- What we want is a kind of ‘secretary’: someone who understood the things we were interested in, (and the things we are not interested in), who can act as ‘proxy’, hiding information that we are not interested in, and bringing to our attention information that is of interest.
- We cannot afford human agents to do these kinds of tasks (and in any case, humans get suffer from the drawbacks we mentioned above). So we write an agent to do these tasks.
- The agent would typically be given a query that would require synthesizing pieces of information from various different Internet information sources. Failure would occur when a particular resource was unavailable, (perhaps due to network failure), or where results could not be obtained.

Example: Tour guides

- The idea here is to have agents that help to answer the question ‘where do I go next’ when browsing the WWW.

- Such agents can learn about the user’s preferences in the same way that MAXIMS does, and rather than just providing a single, uniform type of hyperlink actually indicate the likely interest of a link.

Agents for E-Commerce

- Another important rationale for internet agents is the potential for electronic commerce.
- Most commerce is currently done manually. But there is no reason to suppose that certain forms of commerce could not be safely delegated to agents.
- Examples:
  - find the cheapest copy of MS Office from online stores;
  - flight from Manchester to Dusseldorf with veggie meal, window seat, plus hotel, taxis, entertainment, restaurants.

Agents for E-Commerce

- First & Second Generation E-Commerce Systems
  - First generation: comparison shopping agents.
  - Examples:
    - 1995: Bargain Finder from Andersen;
    - 1997: Jango from NETBOT.
    - 2003: Froogle from GOOGLE
- Second-generation: negotiation, brokering, ...

Example: Jango

- Jango (Doorenbos et al, Agents 97) is a good example of e-commerce agent.
- Long-term goals:
  - Help user decide what to buy.
  - Finding specs and reviews of products.
  - Make recommendations.
  - Comparison shopping for best buy.
  - Monitoring “what’s new” lists.
  - Watching for special offers & discounts.

Jango

- Isn't comparison shopping impossible? WWW pages all different!
- Jango exploits several regularities in merchant WWW sites:
  - navigation regularity: sites designed so that products easy to find
  - corporate regularity: sites designed so that pages have same look'n'feel;
  - vertical separation: merchants use whitespace to separate products.
- Two key components of Jango:
  - learning vendor descriptions;
  - comparison shopping;

What if the agents become better?

- Personal assistant agents need not simply search
- They can plan, arrange, buy, negotiate – carry out arrangements of all sorts that would normally be done by their human user
- As more can be done electronically, software agents theoretically have more access to systems that affect the real-world
- But new research problems arise just as quickly…

Research Issues

- How do you state your preferences to your agent?
- How can your agent compare different deals from different vendors? What if there are many different parameters?
- What algorithms can your agent use to negotiate with other agents (to make sure you get a good deal)?
- These issues aren't frivolous – automated procurement could be used massively by (for example) government agencies
- The Trading Agents Competition…

Applications – Multi-Agent Systems

- Multiagent systems are indicated for domains where:
  - Control, data, expertise are distributed;
  - Centralized control is impossible or impractical;
  - Processing nodes have competing/conflicting viewpoints or objectives.
- Some applications of multi agents:
  - Air Traffic Control System
  - Agents for workflow and business process management

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

Agents for Workflow and Business Process Management

- Workflow and business process control systems
  - Automate the processes of a business
  - Ensure that different business tasks are expedited by the appropriate people at the right time
- ADEPT (Jennings et al., 1996)
  - A business organization is modelled as a society of negotiating, service providing agents
  - Providing customers with a quote for installing a network to deliver a particular type of telecommunications service.
    - The process involves different British Telecom departments: the Customer Service Division, the Legal Division, and so on.

Agents for Workflow and Business Process Management

- ADEPT (Jennings et al., 1996)
  - If the service requested by the customer is bespoke
    - Customer service division further analyses the requirements
    - Legal division checks the legality
    - If legal, then the Design division designs the service
  - Each department becomes an agent, each individual within a department becomes an agent
  - Each agent needs to achieve its individual objectives, and interacts with each other, in the form of negotiations about which services the agents would provide to one another, under different terms and conditions

What Will be Covered

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

Summary

- Understand 5 reasons for the emergence of the field.
- Why these have led to Multi-Agent Systems
- Definitions of Agents, multi-agent systems: difficulty in definition
- Know the related research fields
- Understand the differences and similarities between agents and other fields
- Applications of individual agent and multi-agent system
- Some research issues in the agents field