Overview

- Introduce the idea of an *agent* as a computer system capable of *flexible autonomous action*.
- Briefly discuss the issues one needs to address in order to build agent-based systems.
- Point to the main application areas for agent technology.
- Tell you where to find out more about agents and agent technology.
Reactivity

- If a program’s environment is guaranteed to be fixed, the program need never worry about its own success or failure — program just executes blindly. Example of fixed environment: compiler.
- The real world is not like that: things change, information is incomplete. Many (most?) interesting environments are dynamic.
- Software is hard to build for dynamic domains: program must take into account possibility of failure — ask itself whether it is worth executing!
- A reactive system is one that maintains an ongoing interaction with its environment, and responds to changes that occur in it (in time for the response to be useful).

What is an Agent?

- The main point about agents is they are autonomous: capable of acting independently, exhibiting control over their internal state.
- Thus: an agent is a computer system capable of autonomous action in some environment.
- Trivial (non-interesting) agents:
  - thermostat;
  - UNIX daemon (e.g., biff).
- An intelligent agent is a computer system capable of flexible autonomous action in some environment.
  By flexible, we mean:
  - reactive;
  - pro-active;
  - social.
Proactiveness

- Reacting to an environment is easy (e.g., stimulus $\rightarrow$ response rules).
- But we generally want agents to do things for us.
- Hence goal directed behaviour.
- Pro-activeness = generating and attempting to achieve goals; not driven solely by events; taking the initiative.
- Recognising opportunities.

Social Ability

- The real world is a multi-agent environment: we cannot go around attempting to achieve goals without taking others into account.
- Some goals can only be achieved with the cooperation of others.
- Similarly for many computer environments: witness the INTERNET.
- Social ability in agents is the ability to interact with other agents (and possibly humans) via some kind of agent-communication language, and perhaps cooperate with others.
**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 this state.

- **Main differences:**
  - standard object model has single thread of control (but concurrent object systems...)
  - no notion of flexible (goal directed, reactive) behaviour.

---

**Other Properties**

- Other properties, sometimes discussed in the context of agency:
  - **mobility:** the ability of an agent to move around an electronic network;
  - **veracity:** an agent will not knowingly communicate false information;
  - **benevolence:** agents do not have conflicting goals, and that every agent will therefore always try to do what is asked of it;
  - **rationality:** agent will act in order to achieve its goals, and will not act in such a way as to prevent its goals being achieved — at least insofar as its beliefs permit;
  - **learning/adaption:** agents improve performance over time.
**Agents and Expert Systems**

- Aren’t agents just expert systems by another name?
- Expert systems typically disembodied ‘expertise’ about some (abstract) domain of discourse (e.g., blood diseases).
- Main differences:
  - agents situated in an environment;
  - agents act.
- Some real-time (typically process control) expert systems are agents.

---

**Intelligent Agents and AI**

- Aren’t agents just the AI project?
  - Isn’t building an agent what AI is all about?
- AI aims to build systems that can (ultimately) understand natural language, recognise and understand scenes, use common sense, think creatively, etc — all of which are very hard.
- When building an agent, we simply want a system that can choose the right action to perform, typically in a very limited domain.
- Agents are thus no more the AI project than expert systems.
The philosopher Daniel Dennett coined the term *intentional system* to describe entities ‘whose behaviour can be predicted by the method of attributing belief, desires and rational acumen’.

Dennett identifies different ‘grades’ of intentional system:

‘A first-order intentional system has beliefs and desires (etc.) but no beliefs and desires about beliefs and desires. ... A second-order intentional system is more sophisticated; it has beliefs and desires (and no doubt other intentional states) about beliefs and desires (and other intentional states) — both those of others and its own’.

Is it legitimate or useful to attribute beliefs, desires, and so on, to computer systems?

Agents = Intentional Systems

Where do theorists start from?

The notion of an agent as an intentional system...

When explaining human activity, it is often useful to make statements such as the following:

Janine took her umbrella because she *believed* it was going to rain.

Michael worked hard because he *wanted* to possess a PhD.

These statements make use of a folk psychology, by which human behaviour is predicted and explained through the attribution of attitudes, such as believing and wanting (as in the above examples), hoping, fearing, and so on.

The attitudes employed in such folk psychological descriptions are called the intentional notions.
What objects can be described by the intentional stance?

As it turns out, more or less anything can... consider a light switch:

'It is perfectly coherent to treat a light switch as a (very cooperative) agent with the capability of transmitting current at will, who invariably transmits current when it believes that we want it transmitted and not otherwise; flicking the switch is simply our way of communicating our desires'. (Yoav Shoham)

But most adults would find such a description absurd!
Why is this?

McCarthy argued that there are occasions when the intentional stance is appropriate:

'To ascribe beliefs, free will, intentions, consciousness, abilities, or wants to a machine is legitimate when such an ascription expresses the same information about the machine that it expresses about a person. It is useful when the ascription helps us understand the structure of the machine, its past or future behaviour, or how to repair or improve it. It is perhaps never logically required even for humans, but expressing reasonably briefly what is actually known about the state of the machine in a particular situation may require mental qualities or qualities isomorphic to them. Theories of belief, knowledge and wanting can be constructed for machines in a simpler setting than for humans, and later applied to humans. Ascription of mental qualities is most straightforward for machines of known structure such as thermostats and computer operating systems, but is most useful when applied to entities whose structure is incompletely known.'
• The intentional notions are thus abstraction tools, which provide us with a convenient and familiar way of describing, explaining, and predicting the behaviour of complex systems.

• Remember: most important developments in computing are based on new abstractions:
  – procedural abstraction;
  – abstract data types;
  – objects.

Agents, and agents as intentional systems, represent a further, and increasingly powerful abstraction.

• So agent theorists start from the (strong) view of agents as intentional systems: one whose simplest consistent description requires the intentional stance.

‘... it does not buy us anything, since we essentially understand the mechanism sufficiently to have a simpler, mechanistic description of its behaviour’. (Yoav Shoham)

• Put crudely, the more we know about a system, the less we need to rely on animistic, intentional explanations of its behaviour.

• But with very complex systems, a mechanistic, explanation of its behaviour may not be practicable.

• As computer systems become ever more complex, we need more powerful abstractions and metaphors to explain their operation — low level explanations become impractical.

The intentional stance is such an abstraction.
Characterising Agents

- It provides us with a familiar, non-technical way of understanding & explaining agents.

- This intentional stance is an abstraction tool — a convenient way of talking about complex systems, which allows us to predict and explain their behaviour without having to understand how the mechanism actually works.

- Now, much of computer science is concerned with looking for abstraction mechanisms (witness procedural abstraction, ADTs, objects, …)

  So why not use the intentional stance as an abstraction tool in computing — to explain, understand, and, crucially, program computer systems?

- This is an important argument in favour of agents.

- 3 other points in favour of this idea:
Rational Action Lecture 1: Introduction

Post-Declarative Systems

- This view of agents leads to a kind of post-declarative programming:
  - in procedural programming, we say exactly *what* a system should do;
  - in declarative programming, we state something that we want to achieve, give the system general info about the relationships between objects, and let a built-in control mechanism (e.g., goal-directed theorem proving) figure out what to do;
  - with agents, we give a very abstract specification of the system, and let the control mechanism figure out what to do, knowing that it will act in accordance with some built-in theory of agency (e.g., the well-known Cohen-Levesque model of intention).

Nested Representations

- It gives us the potential to specify systems that *include representations of other systems*.
  It is widely accepted that such nested representations are essential for agents that must cooperate with other agents.
How to find out more

- Agents on the WWW:
  http://www.cs.umbc.edu/agents/
  http://www.AgentLink.org/
- Conferences:
  – International Conference on Multi-Agent Systems (ICMAS)
  – International Conference on Autonomous Agents (Agents)
  – Practical Application of Agents and Multi-Agent Systems (PAAM)
- Journal:
  Autonomous Agents and Multi-Agent Systems (Kluwer Academic Publishers)
- Agent standardisation:
  http://www.fipa.org/

An Aside

- We find that researchers from a more mainstream computing discipline have adopted a similar set of ideas…
- In distributed systems theory, logics of knowledge are used in the development of knowledge based protocols.
- The rationale is that when constructing protocols, one often encounters reasoning such as the following:
  
  \[
  \text{IF} \quad \text{process } i \text{ knows process } j \text{ has received message } m_1 \\
  \text{THEN} \quad \text{process } i \text{ should send process } j \text{ the message } m_2.
  \]
- In DS theory, knowledge is grounded — given a precise interpretation in terms of the states of a process; return to this later… we’ll examine this point in detail later.