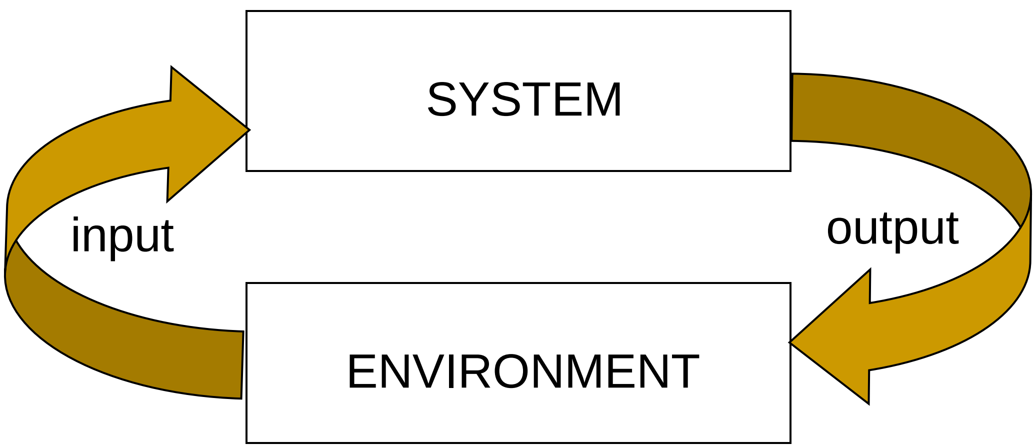
Intelligent Agents

1. Intelligent Agents
2. Agents and Objects
3. Agents and Expert Systems
4. Agents as Intentional Systems
5. Abstract Archtectures for Intelligent Agents
6. How to Tell an Agent What to Do
7. Synthesizing Agents

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*** *in order to meet its* **design objectives**



What is an Agent?

- 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*

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)

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 behavior*
- Pro-activeness = generating and attempting to achieve goals; <u>not driven solely by events</u>; taking the initiative
- **<u>Reco</u>**gnizing **<u>opportunities</u>**

Balancing Reactive and Goal-Oriented Behavior

- We want our agents to be reactive, responding to changing conditions in an appropriate (timely) fashion
- We want our agents to systematically work towards long-term goals
- These two considerations can be at **odds with one another**
- Designing an agent that can balance the two remains **an open** research problem

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

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 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

Agents and Objects

- Main **<u>differences</u>**:
  - *agents are autonomous:*
    agents embody <u>stronger notion of autonomy **than**</u> objects, and in particular, they **<u>decide</u>** for themselves whether or not to perform an action on request from another agent
  - *agents are smart:*
    capable of **<u>flexible</u>** (reactive, pro-active, social) behavior, and <u>the **standard object model** has nothing to say about such types of behavior</u>
  - *agents are active:*
    a multi-agent system is inherently multi-threaded, in that each agent is assumed to have **<u>at least</u>** one thread of active control

Objects do it for free…

- *agents do it because they* ***<u>want to</u>***
- *agents do it* ***<u>for</u>*** *…….*

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)
- 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 and Expert Systems

- Main **<u>differences</u>**:
  - agents ***<u>situated in an environment</u>**:*
    MYCIN is not aware of the world — only information obtained is by asking the user questions
  - agents ***<u>act</u>**:*
    MYCIN does not operate on patients
- **<u>Some</u>** *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?
- **<u>AI</u>** aims to build systems that can (ultimately) understand natural language, recognize and understand scenes, use common sense, think creatively, etc. — **<u>all of which</u>** are very hard
- So, <u>don’t we need to solve all of AI</u> to build an agent…?

Intelligent Agents and AI

- When building an agent, we simply want a system that can choose the right action to perform, typically in a limited domain

- We *do not* have to solve *all* the problems of AI to build a useful agent:

  a little intelligence goes a long way!

Environments – *Accessible* vs. *inaccessible*

- **An accessible** environment is one in which the agent can obtain complete, accurate, up-to-date information about the environment's state
- Most moderately complex environments (including, for example, the everyday physical world and the Internet) are **inaccessible**
- The more accessible an environment is, the simpler it is to build agents to operate in it

Environments –
Deterministic vs. *non-deterministic*

- A **deterministic** environment is one in which any action has a single guaranteed effect — there is no uncertainty about the state that will result from performing an action
- The physical world can to all intents and purposes be regarded as **non-deterministic**
- Non-deterministic environments present greater problems for the agent designer

Environments - *Episodic* vs. *non-episodic*

- In an **episodic** environment, the performance of an agent is dependent on a number of discrete episodes, <u>with no link between </u>the performance of an agent in different scenarios
- Episodic environments are simpler from the agent developer's perspective because the agent can decide what action to perform based only on the <u>**current**</u> episode — it need not reason about the interactions between this and future episodes

Environments - *Static* vs. *dynamic*

- A **static** environment is one that can be assumed to remain unchanged except by the performance of actions by the agent
- A **dynamic** environment is one that has other processes operating on it, and which hence changes in ways beyond the agent’s control
- Other processes can interfere with the agent’s actions (as in concurrent systems theory)
- The physical world is a highly dynamic environment

Environments – *Discrete* vs. *continuous*

- An environment is **discrete** if there are a fixed, finite number of actions and percepts in it
- Russell and Norvig give a chess game as an example of a discrete environment, and taxi driving as an example of a continuous one
- **Continuous** environments have a certain level of mismatch with computer systems
- Discrete environments could *in principle* be handled by a kind of "lookup table"

Agents as Intentional Systems

- 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 behavior 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

Agents as Intentional Systems

- The intentional notions are thus *abstraction tools*, which provide us with a convenient and familiar way of describing, explaining, and predicting the behavior 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