Agents & Agent-Based Programming

CIS 3.5, Lecture 4.1

Outline

I. References
II. "Agents" defined.
III. Agent-Based Programming Paradigm
IV. Motivations
V. Putting it all together
VI. Computer & Software Agents
VII. Software Agents - Disambiguation
VIII. Agent Components
IX. Types of Agents
X. Multi-agent system
XI. Properties of multi-agent systems
XII. Introduction to NetLogo

References

- Russell, Stuart J.; Norvig, Peter (2003), Artificial Intelligence: A Modern Approach (2nd ed.), Upper Saddle River, New Jersey: Prentice Hall, ISBN 0-13-790395-2, http://aima.cs.berkeley.edu/ , chpt. 2
- Stan Franklin and Art Graesser (1996); Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents; Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages, Springer-Verlag, 1996
- An Introduction to Multiagent Systems, by Michael Wooldridge, John Wiley & Sons, Ltd (2002).

"Agents" Defined

- What an "agent" is depends on the context of the domain.
  - Real estate agent.
  - Secret agent.
  - Chemical agent.
- The common component in almost all cases is that an "agent" is something that "acts" autonomously (independently) usually on behalf of some other entity.
- Agent is derived from the Latin *agere* (to do).

Agent "Paradigm"

- The term "agent-based programming" describes a paradigm (an abstraction, idea, or a concept) similar to OOP (which uses terms such as methods, functions, and objects).
- The concept of an agent provides a convenient and powerful way to describe an entity that is capable of acting with a certain degree of autonomy in order to accomplish tasks on behalf of its user.
- Four key notions distinguish agents from arbitrary programs:
  - persistence
  - autonomy
  - reaction to the environment
  - goal-orientation

Motivations

- Five trends in the history of computing have led to the development of agent-based and multi-agent systems:
  1. ubiquity
  2. interconnectivity
  3. intelligence
  4. delegation
  5. human-orientation

Ubiquity

- Computing devices are everywhere
  - Computational power (i.e., hardware) has become small and relatively inexpensive.
- Computing devices do all sorts of things
  - Many devices are hidden inside other devices (embedded computing).
- How do we design systems that might run anywhere, on anything?
- Can we think about the behaviors we want rather then the program we need to write.

Interconnectivity

- The rise of the internet and network-based systems has led to the idea that most computational devices do not run alone—they connect to a wide range of other types of computational devices in order to perform tasks collaboratively (also called distributed and concurrent or parallel systems)
- How do we design systems that can be executed on multiple processors/computers?
- Can we conceive of a system as a series of interacting devices rather then as a single system?

Intelligence

- As computers become more sophisticated, so must the software that operates them.
- What does "intelligence" mean in the context of computing?
- What do you believe makes a computer/ computational device intelligent?
- We can think about systems that can react to "unpredictable" events in its operating environment as intelligent.

Delegation

- Computers do things for us, often without our intervention (traffic lights, power grids/plants).
- Giving control to a computer is called "delegation"; are you comfortable with that? (3-mile Island)
- What kinds of tasks are you comfortable delegating to a computer?
  - tasks that involve safety (flying, driving)
  - spending money (bidding on eBay)
  - Combat operations (robot warriors)

Human Orientation

- Most computer systems are designed to interact with humans.
- Even if a human delegates tasks to a computer, there is still interaction before and after the task is completed.
- How do we design systems that can interface effectively with a wide range of human users?
- What kinds of human features do we need to construct/model in a system in order to increase its usefulness?
  - language?
  - emotion?
  - reasoning?
  - decision making?

Putting it all together…

- We come up with a desire for computational devices that can interact with each other to complete tasks. By that, we mean they might:
  - Compete -> perhaps for scarce resources.
  - Cooperate -> perhaps to accomplish tasks together that one cannot do alone.
  - Negotiate -> perhaps to manage the sharing or trading of resources.

"Computer Agent"

- “An agent is a computer system that is capable of independent (autonomous) action on behalf of its user or owner (figuring out what needs to be done to satisfy design objectives, rather than constantly being told what to do in detail).” (Wooldridge)
- An Agent "is an autonomous entity which observes and acts upon an environment and directs its activity towards achieving goals." Russell & Norvig 2003,

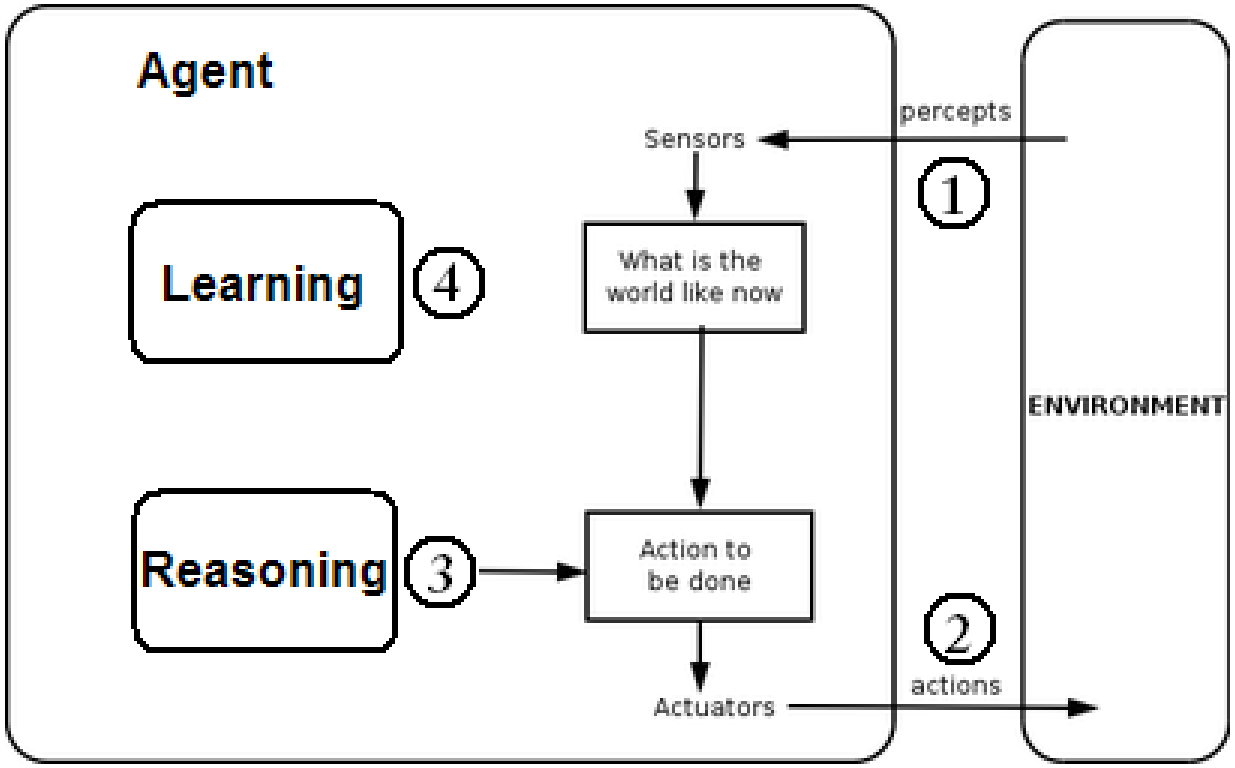
Computer Agent - Disambiguation

- Unlike arbitrary programs, agents: react to their environment, are autonomous, exhibit goal-orientation and persistence.
- Unlike objects which are defined in terms of methods and attributes, an agent is defined in terms of its behavior.
- NOTE: A "software agent" usually implies a program, a "computer agent" can be a hardware/software combination, and an "intelligent agent", can be anything (including people/groups of people).

A Software Agent is

- "anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors". (Stuart Russell & Peter Norvig, 2003)
- "a system that is situated in an environment, and which is capable of perceiving its environment and acting in it to satisfy some objectives." (Michael Wooldridge, 2002)

Components of a Software Agent



1. **Sensors**
2. **Actuators/Effectors**
3. **Reasoning** (maybe be very simple)
4. **Learning** (optional, defines an "intelligent agent")

Types of Agents

- Simple reflex agents -> If condition then action.

- Model-based reflex agents -> A model-based reflex agent keeps track of the current state of the world using an internal model. It then chooses an action in the same way as the reflex agent.

Types of Intelligent Agents

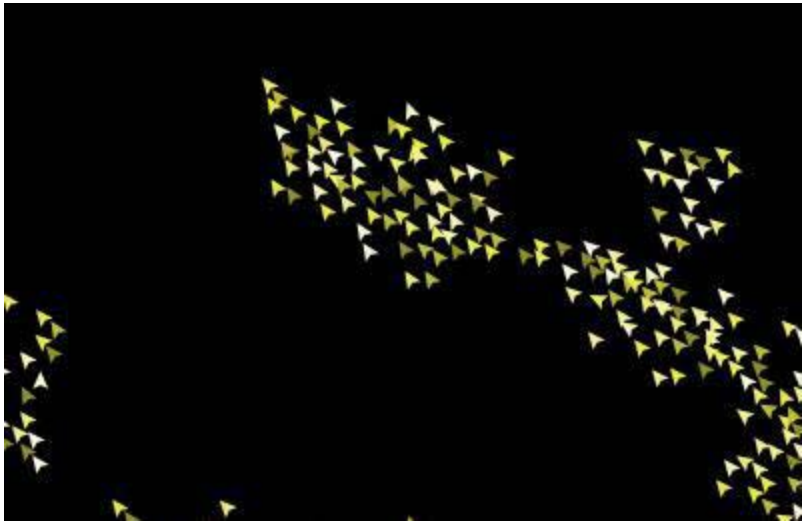
- Goal-based agents
  - Goal-based agents are model-based agents which store information regarding situations that are desirable. This allows the agent a way to choose among multiple possibilities, selecting the one which reaches a goal state.
- Learning agents
  - Learning has an advantage that it allows the agents to initially operate in unknown environments and to become more competent than its initial knowledge alone might allow.

Multi-Agent System

- “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 interact successfully, they will require the ability to cooperate, coordinate and negotiate with each other, much as people do.” (Wooldridge)

Multi-Agent System

- (MAS) is... an environment in which many (well, two or more) agents exist and interact.



Properties of multi-agent systems

- Individual agents are self-interested
  - they have their own goals
  - there may be team rewards for a group of agents achieving a goal together
- Cooperation is not governed,
  - It is emergent (if it happens at all)
- versus “distributed systems”, where
  - goals are only group-based
  - cooperation is engineered to be inherent in the system

Agent Based Programming… Why?

Are there advantages to using Agent-Based programming? Yes!

1. Dealing with complexity: Some problems are so complex that they are basically impossible to model as a single program. Examples: Weather, Internet, Travel Systems
2. Emergent Behavior: Some types of behavior are very difficult to program in directly but can result as a side effect of agents pursuing very simple goals.

Key Research Areas

1. Agent design (micro level):
   - how do we design individual agents to operate effectively?
2. Society Design (macro level):
   - how do we design groups of agents, or societies, to operate effectively?
3. Hive Intelligence (abstract level):
   - Can simple agents, working together, effectively demonstrate high level reasoning abilities.

NetLogo (1)

- **NetLogo** is a programmable modeling environment for simulating natural and social phenomena..
- **NetLogo** is particularly well suited for modeling complex systems developing over time. Modelers can give instructions to hundreds or thousands of "agents" all operating independently. This makes it possible to explore the connection between the micro-level behavior of individuals and the macro-level patterns that emerge from the interaction of many individuals.

NetLogo (2)

- **NetLogo** lets students open existing simulations exploring their behavior under various conditions. It is also an authoring environment (text-based IDE) which enables developers to create their own models. NetLogo is simple enough that students and teachers can easily run simulations or even build their own. And, it is advanced enough to serve as a powerful tool for researchers in many fields.
- **NetLogo** has extensive documentation and tutorials. It also comes with a Models Library, which is a large collection of pre-written simulations that can be used and modified.