cis3.5 fall2009 lecture IV.1

topics:

- today we will discuss agent-based programming
- some of the material from today's lecture comes from **An Introduction to Multiagent Systems**, by Michael Wooldridge, John Wiley & Sons, Ltd (2002).

five trends

- five trends in the history of computing have led to the development of agent-based and multiagent systems:
  - ubiquity
  - interconnectivity
  - intelligence
  - delegation
  - human-orientation

ubiquity

- computational power (i.e., hardware) has become small and relatively inexpensive
- as a result, there are computing devices *everywhere*
- that means that there are devices that do all sorts of things
- and are hidden in all sorts of devices (*embedded computing*)
- how do we design systems that might run anywhere, on anything?

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?

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
- when we give control to a computer, we call that "delegation"
- are you comfortable doing that?
- what kinds of tasks are you comfortable delegating to a computer? tasks that involve safety, like flying a plane or parking a car? what about spending money, like bidding for you on ebay?

human orientation

- the types of systems described above interact with humans in order to be useful
- 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?

interconnection and distribution

- putting these together, we come up with computational devices that need to *cooperate* together to complete tasks
- sometimes this means that they *compete*, perhaps for scarce resources
- sometimes this means that they *cooperate*, perhaps to accomplish tasks together that one cannot do alone
- sometimes this means that they *negotiate*

agent

- definition from Wooldridge:
  "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)."

multiagent system

- definition from Wooldridge:
  "A *multiagent 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."

example: internet agents

- use of agents to conduct tasks for you on the internet
- such as searching for information
- finding deals (buying a car, planning a vacation)
- how much would you *trust* the agent to do things for you? would you give the agent the authorization to access your bank account (and spend money)?

two key problems/research areas

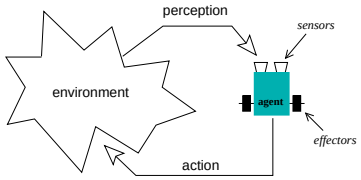
- agent design (micro level):
  how do we design individual agents to operate effectively?
- society design (macro level):
  how do we design groups of agents, or societies, to operate effectively?

software agents

an *agent* is...

- anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors (e.g., motors)

  (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)

different kinds of agents

- human "agent":

environment:	physical world	
sensors:	eyes, ears, ...	← input
effectors:	hands, legs, ...	← output

- software agent:

environment:	e.g., UNIX operating system	
sensors:	`ls`, `ps`, ...	← input
effectors:	`rm`, `chmod`, ...	← output

- internet agent:

environment:	the Internet	
sensors:	`http` requests	← input
effectors:	`http` commands	← output

- embodied (robotic) agent:

environment:	physical world	
sensors:	light meters, bumpers, thermometers, ...	← input
effectors:	motors attached to wheels, legs, grippers, ...	← output

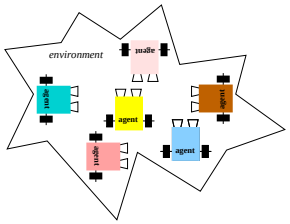
agent decision-making: what to do?

- need to know *what* to do in any given *state*
  - what = an *action* that the agent can take
  - state = a configuration of the agent and its environment
    for example: the position of all the pieces on a chess board, or the robots and the ball on a robot soccer field, or the position of a robot's gripper, or all the bids in an electronic market
- *autonomy*
  - a crucial concern for agents
  - run-time decisions are made by the agent *alone*—i.e., no human remote control
  - means behavior is based on *own* experience
  - implies *learning*, *adaptation*

multiagent system

- a *multiagent system (MAS)* is...

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



properties of multiagent systems

- individual agents are *self-interested*
  i.e., they have their own goals, even though there may be team rewards for a group of agents achieving a goal together
- cooperation is not governed
  it is *emergent*
  (and is not necessarily a feature of every multiagent system)
- versus "distributed systems", where
  - goals are only group-based
  - cooperation is engineered to be inherent in the system