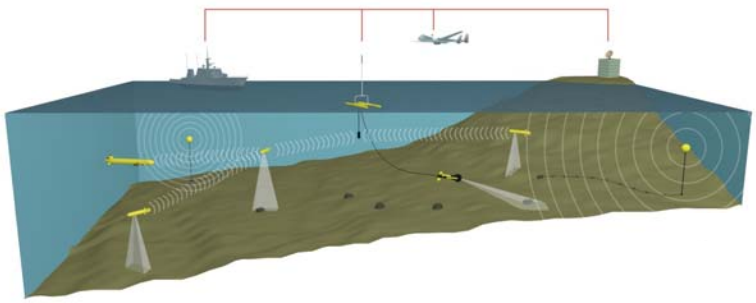
Autonomous Robotics 6905



Lecture 5: Introduction to Autonomous Robots and Agents

Dalhousie University

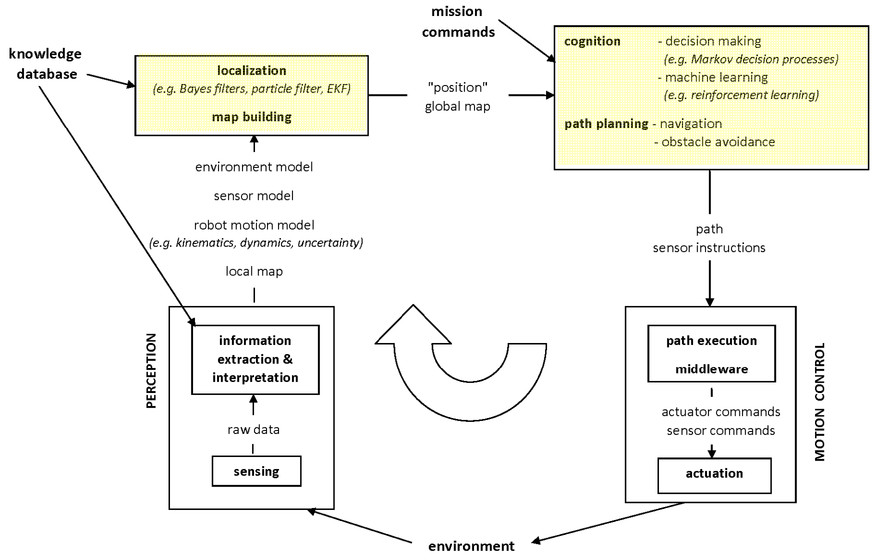
November 4, 2011

Lecture Outline

- Introduction
- Concluding Remarks

- based on diagrams and lecture notes adapted from:
  - Probabilistic Robotics (Thrun, et. al.)
  - Artificial Intelligent: A Modern Approach (Stuart Russell & Peter Norvig)

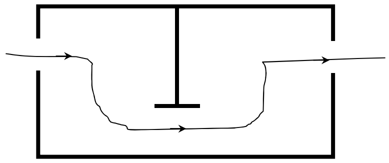
Control Scheme for Autonomous Mobile Robot



Control Scheme for Autonomous Mobile Robot – the plan

- Introduction
- Agent Definitions & Terms
- Agent Classifications
- Autonomous Robots & Agents
- Concluding Remarks

- have covered some areas in the learning aspects of autonomous robotics
- today's focus is on agents and their role with autonomous robots – largest impact is with the localization and cognitive capabilities in the previous plot



Motivation

- answer the questions:
- What are the different types of agents?

  What is the role of agents in autonomous robotics?

  What is the state-of-the-art in this area?

Agent Definitions

- an agent is an entity that interacts with its environment in two ways:
  – perception through sensors
  – actions through effectors or actuators
- an agent perceives its environment through sensors
  – complete set of inputs at a given time is a percept
  – the current percept, or a sequence of percepts, can influence the actions of an agent
- an agent can change the environment through actuators
  – operation using an actuator is an action
  – actions can be grouped into action sequences

Common Notions in Agent Definitions

hardware or software system that is:

- situated: embedded in some environment (may be physical world, a software environment, a community of agents) which can:
  - *sense* (through sensors, message receipt, or event detection to get partial info on environment state) and
  - *act upon* (via effectors, messages or event generation with possible non-deterministic outcomes)
- reactive: responds in a timely fashion to messages, sensed data or detected events - so actively monitors state of environment
- autonomous: operates without direct intervention of humans or other agents, with independent control over its actions and internal state

Common Notions in Agent Definition

- social: can interact with other agents and possibly humans using messages or actions that change the shared environment
- pro-active: has one or more goals which it tries to achieve by communicating with other agents or acting on its environment
- mentalistic model: has an internal architecture that can be understood in terms of mentalistic notions such as beliefs, desires, intentions and obligations

Common Notions in Agent Definition

human beings as agents

- situated
- sensors: eyes (vision), ears (hearing), skin (touch), tongue (taste), nose (olfaction), neuromuscular system (proprioception)
- percepts: at the lowest level – electrical signals;
  after preprocessing – objects in the visual field (location, textures, colors, …), auditory streams (pitch, loudness, direction), …
- actuators: limbs, digits, eyes, tongue, …
- actions: lift a finger, turn left, walk, run, carry an object
- (often): intelligent and autonomous

Another Agent Definition

- another definition from AgentLink Agent Roadmap:

  An agent is a computer system that is capable of flexible autonomous action in dynamic, unpredictable, typically multi-agent domains.

Agent Type Definitions

- an *agent* is anything that is capable of acting upon information it perceives
- an *intelligent* agent is an agent capable of making decisions about how it acts based on experience (sequence of percepts)
- an *autonomous intelligent* agent is an intelligent agent that is free to choose between different actions
- an *adaptive* agent improves its goal achieving competence with time (sometimes through ability to learn)
- a program that acts in a purely computational environment is a *software* agent

Autonomous Agent

- an agent *is autonomous* if it operates completely autonomously, i.e, *it* decides how to relate its sensory data to motor commands in such a way that it fulfills its goals
- an autonomous agent is "*a system situated within and a part of an environment that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future*“ [2]

Adaptive Agent

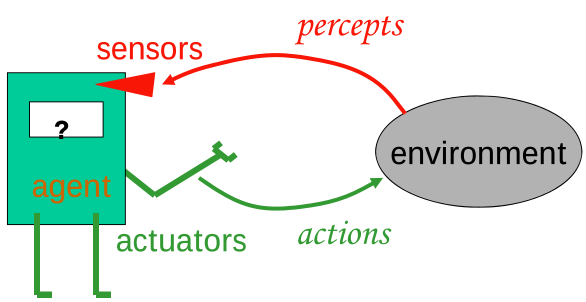
- an autonomous agent approach is appropriate for classes of problems that require a system to autonomously achieve several goals in a dynamic, complex, and unpredictable environment like the underwater one

Intelligent Agent

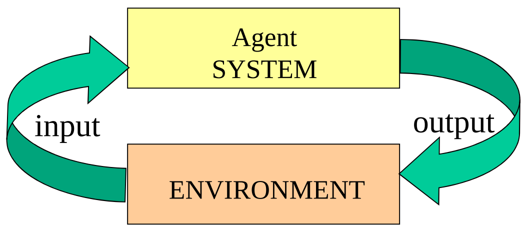
- anything that can be viewed as **perceiving its environment through sensors and acting upon that environment through its effectors to**
  - maximize progress towards its **goals**
    - PAGE (Percepts, Actions, Goals, Environment)
    - task-specific & specialized: well-defined goals and environment
    - a tool for analyzing systems, not an absolute characterization that divides the world into agents and non-agents
    - covers a broad spectrum of machines, from thermostats (which do not learn) to worms (which can learn) to humans (great learning capacity)

Intelligent Agent

- an intelligent agent perceives its environment via sensors and acts rationally upon that environment with its actuators



Intelligent Agent



Realizations of Agents

- "*A coupling of perception, reasoning, and acting comprises an agent. An agent acts in an environment. An agent's environment may well include other agents. An agent together with its environment is called a world.*" [1]
- for e.g. a coupling of a computational engine with physical sensors and actuators, called a robot, where the environment is a physical setting
- coupling of an expert system with a human who provides perceptual information and implements the task

Examples of Agents

1. human being
   - eyes, ears, skin, taste buds, etc. as sensors
   - arms, hands, fingers, legs, etc. for actuators
2. robot
   - camera, bump, infrared, ultrasound, etc. for sensors
   - wheels, lights, manipulators, etc. for actuators
3. software
   - functions as sensors – information provided as input to functions as encoded bit strings or symbols
   - functions as actuators – results deliver the output

Distinctions between Agents and Software

- agents are autonomous, that is, they act on behalf of the user
- agents contain some level of intelligence, from fixed rules to learning engines that allow them to adapt to changes in the environment
- agents don't only act reactively, but could act proactively
- agents have social ability – communicate with the user, system, and other agents as required
- agents may cooperate with other agents to carry out more complex tasks than they can do alone
- agents may migrate from one system to another to access remote resources

Why are agents useful?

- specialized tasks

  agents (and their physical instantiation in robots) have a role to play in high-risk situations, unsuitable or impossible for humans (dull, dirty, dangerous again)

- in applications where the data, control or resources are distributed:

  the system can be conceptualized as a collection of co-operating components

Why are agents useful?

- tool for understanding human societies:
  multi-agent systems provide a novel tool for simulating societies, which may help shed some light on various kinds of social processes
- tools for formalizing and experimenting with theories of cognition [3, 4]
- agents as a paradigm for software engineering:
  - increasing complex software
  - widely acknowledged that independence of components and their interaction are very important characteristics of complex software

Areas Agents are Applied

- Introduction
- Agent Definitions & Terms
- Agent Classifications
- Autonomous Robots & Agents
- Concluding Remarks

- computer games
  http://www.ai-junkie.com/books/toc_pgaibe.html
- work flow and business process management
  http://www.eil.utoronto.ca/iscm-descr.html
- simulation
  - social, economic, behavioural
    http://jasss.soc.surrey.ac.uk/5/1/7.html
  - complex systems
    http://www.jot.fm/issues/issue_2002_07/column3

Rational Agents

- a rational agent does "the right thing"
  - carries out the action that leads to the best outcome under the circumstances given percept sequence to date
- what is "the right thing"?
- how do you measure "best outcome"?
- performance measures are criteria for measuring how successful an agent is:
  - task dependent
  - can be multi-objective – i.e. considers many factors like how much energy is used, how well it did
  - time may be important – e.g. how long for a task
  - subjective measure: velocity, power, accuracy, usage

Rational Agents

- performance measures example: the Roomba
  - main measure:
    - the number of tiles cleaned during a given time interval
  - other possible effects on measure:
    - energy, noise, loss of useful objects, damaged furniture, not fall down the stairs

Rational Agent

- is *not* omniscient
  - it does not know the actual outcome of its actions
  - may not know certain aspects of the environment
  - does the best with what it knows
- while not omniscient (knows all), a rational agent will do the next best thing
- it selects the action that is expected to maximize its performance
  - based on performance measure
  - depends on the percept sequence (perceptual history), background knowledge, and feasible actions

Rational Agent

- agent autonomy
  - agent programs determine actions based on both precepts and on-board knowledge
  - if the agent uses only on-board knowledge, then it *lacks* autonomy
  - the agent is autonomous to the extent that its behavior is determined by its own experience
  - a truly autonomous intelligent agent should be able to operate successfully in a wide variety of environments, given sufficient time to learn and adapt

Characterizing Agents

- another way to characterize agent is to use the PAGE system

  Percepts: info acquired through the agent's sensory systems

  Actions: operations performed by the agent on the environment through its actuators

  Goals: desired outcome of the task with a measurable performance

  Environment: surroundings beyond the control of the agent

Characterizing Agents

- example 1: windshield wiper

  Percepts: raining, dirty

  Actions: wipe back and forth (off, slow, medium, fast)

  Goals: keep windshield clean, maintain visibility

  Environment: inner city, freeway, highway, weather . . .

Characterizing Agents

- example 2: characterizing an early Roomba

Percepts: position, orientation, floor dirtiness

Actions: move, pick-up dirt

Goals: cleanliness of the floor, time needed to suck, energy consumed

Environment: grid of tiles, dirt on tiles, possibly obstacles, dirt density

Characterizing Agents

- example 3: characterizing a late model Roomba

Percepts: range, orientation, floor dirtiness

Actions: move forward, turn right/left, vacuum on/off

Goals: maximize area of floor vacuumed per unit time

Environment: 2D workspace consisting of a flat grid of tiles in a square room, 5 static obstacles located in known position (e.g. map of furniture place), entire floor has dirt

Characterizing Agents

- example 4: characterizing an IVER2 AUV

Percepts: orientation, speed over ground, speed through water, depth, altitude, current profile, sonar to sonar to produce image of seabed, etc.

Actions: move in 3D space, dive/rise, circle a waypoint,

Goals: perform an efficient survey of an area (for e.g.), identify and classify targets in the area, etc.

Environment: 3D, underwater, uncertain bathymetry, varying temperature, conductivity, and density (and hence effect on buoyancy and sound speed)

Characterizing Agents

- artificial agent examples based on PAGE characterization

agent	Percepts	Action	Goal	Environment
financial forecaster	stock market data	pick stocks to buy/sell	maximize investment	stock market
medical diagnostic	symptom, test results	further tests, treatments	patient care	patient, hospital
deep blue	current board state	determine next move	win	chess board, opponent

Agent Structure

- agent = program + architecture
- agent program: implementation of function, $f: P^* \rightarrow A$, the

  agent's perception-action mapping

  function Skeleton-Agent (percept) returns action

  memory ← UpdateMemory (memory, percept)

  action ← ChooseBestAction (memory)

  memory ← UpdateMemory(memory, action)

  return action
- architecture: a computing device that can execute the agent program (e.g. computer, special camera, device, etc.)
  - makes percepts from sensors available to the program, runs the program, and feeds the program's action choices to the effectors as they are generated

Formal Description

- Introduction
- Agent Definitions & Terms
- Agent Classifications
- Autonomous Robots & Agents
- Concluding Remarks

DALHOUSIE UNIVERSITY *Inspiring Minds* ACADIA UNIVERSITY

- $A := \{a_1, a_2, \ldots, a_n\}$ is the set of actions

  $P := \{p_1, p_2, \ldots, p_m\}$ is the set of percepts

  $S := \{s_1, s_2, \ldots, s_l\}$ is the set of states of the environment
- what does an agent observe, in a certain state, s ?

  see $: S \rightarrow P$
- how does the environment develop (the state, s) when an action, a, is executed?

  env $: S \times A$
- agent is described as:

  action $: P \rightarrow A$

Intelligent Agent Types

- four basic types of agent programs to implement mapping of percepts to actions
  - in order of increasing generality:

1. simple reflex agents
2. model-based reflex agents
3. goal-based agents
4. utility-based agents

Intelligent Agent Types Overview

- *simple reflex agents* incorporate a lookup table to determine the agent's response to its environment
- *model based agents* are also use a lookup table but are aware of the current state of their environment and can hence differentiate between states that appear similar
- *goal based agents* have a state that they strive to be in; allows the agent to have some measure of its "happiness" with respect to achieving a goal
- *utility based agents* use a utility function to allow it to evaluate between varying degrees of "happiness"; states that are preferred over other states have a higher utility value
- *learning agents* explore their environment.

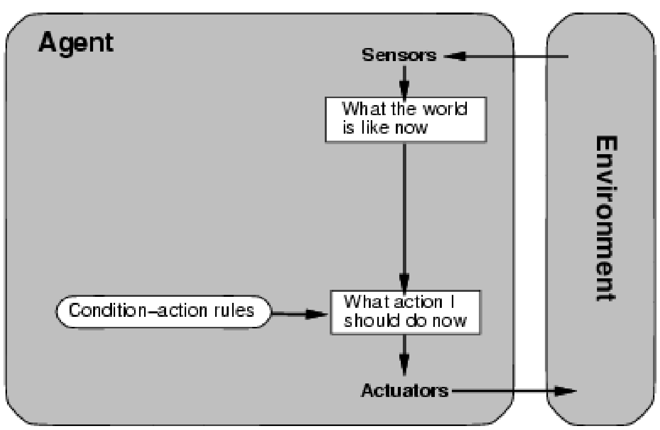
Simple Reflex Agent

- explicit table for an agent that plays chess has 35^{100} entries
- however, can summarize portions of table by noting certain commonly occurring input/output associations

- ***function*** *SIMPLE-REFLEX-AGENT(percept)* ***returns*** *action*
  - ***static****: rules, a set of condition-action rules*
  - *state ← INTERPRET-INPUT(percept)*
  - *rule ← RULE-MATCH(state, rules)*
  - *action ← RULE-ACTION[rule]*

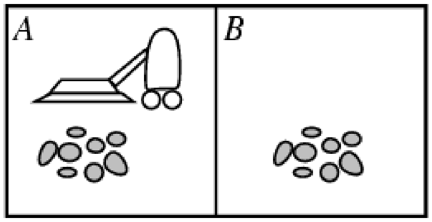
  return *action*

Simple Reflex Agent



Simple Reflex Agent

- Roomba automated vacuum example

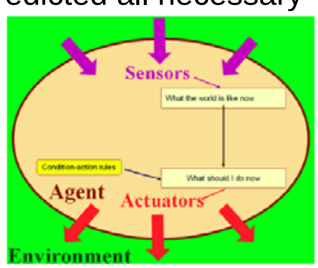


Percept	Action
at A, A dirty	vacuum
at A, A clean	move left
at B, B dirty	vacuum
at B, B clean	move right

Simple Reflex Agent

- acts only on basis of the *current* percept
- agent function is based on the
  condition-action rule: condition ⇒ action

- limited functionality, works well only when
  - the environment is fully observable and
  - the condition-action rules have predicted all necessary actions

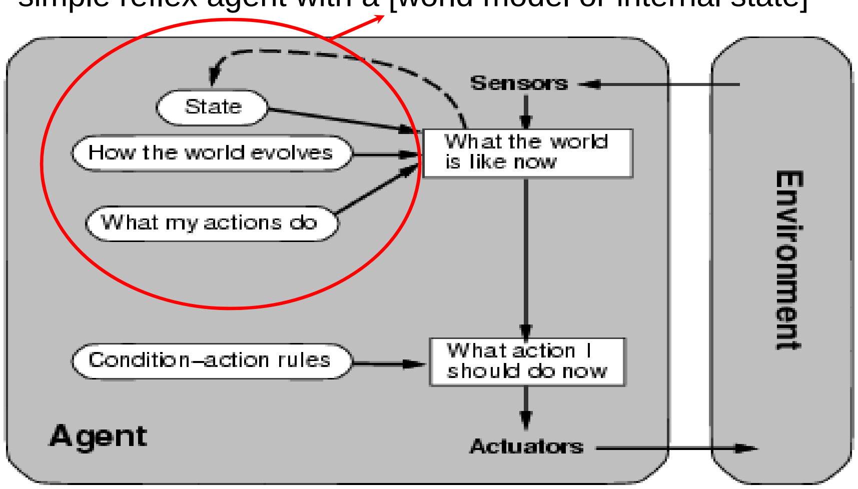


Model-Based Reflex Agents

- sensors do *not* provide complete information about the world; same precept may have different causes
- model-based reflex agent has information on how world behaves through a world model or internal state
  1. this works out information about parts of the world which are not observable
     - e.g. if agent has seen an object in a place and has since not seen any agent moving towards that object then the object is still at that place
  2. knows effects of their own actions on the world
     - e.g. if the agent has moved northwards for 5 minutes then it is 5 minutes north of where it was

Model-Based Reflex Agents

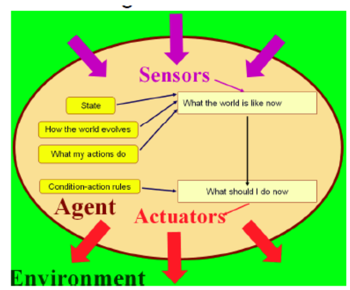
- simple reflex agent with a [world model or internal state]



Model-Based Reflex Agents

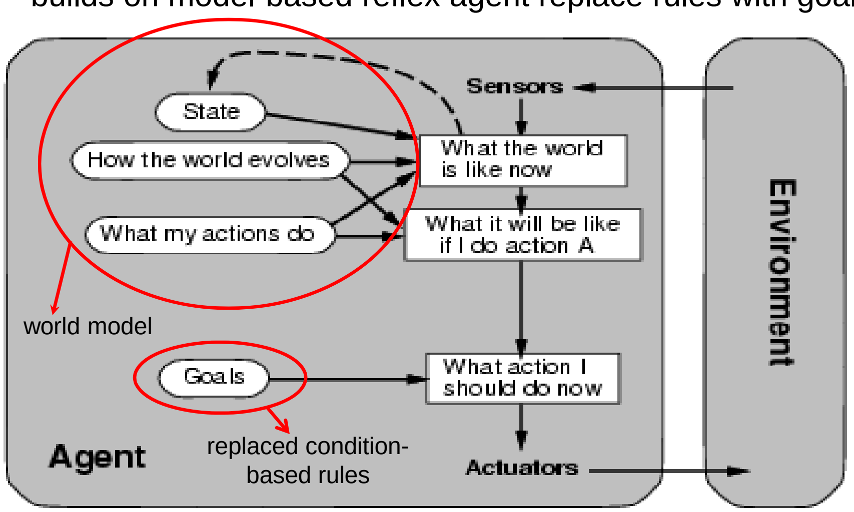
given a percept

- ***function*** *REFLEX-AGENT-WITH-STATE(percept)* ***returns*** *action*
  - ***static***: *state, a description of the current world state*
  - *rules, a set of condition-action rules*
  - *state ← UPDATE-STATE(state, percept)*
  - *rule ← RULE-MATCH(state, rules)*
  - *action ← RULE-ACTION[rule]*
  - *state ← UPDATE-STATE(state, action)*
- ***return*** *action*



Goal-Based Agents

- builds on model-based reflex agent replace rules with goals



Goal-Based Agents

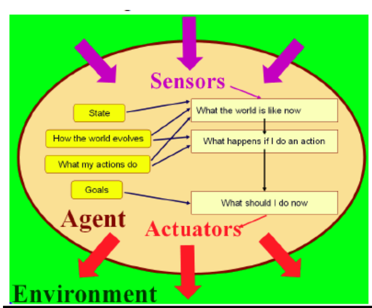
- current state of the world is not always enough to decide what to do
  - e.g. at a junction a car can go left, right or straight - it needs knowledge of its destination to make the decision which of these to choose

- goal-based agent = world model + goals
  - goals are situations that are desirable
  - goals allow the agent a way to choose among multiple possibilities, selecting the one which realizes a goal state

Goal-Based Agents

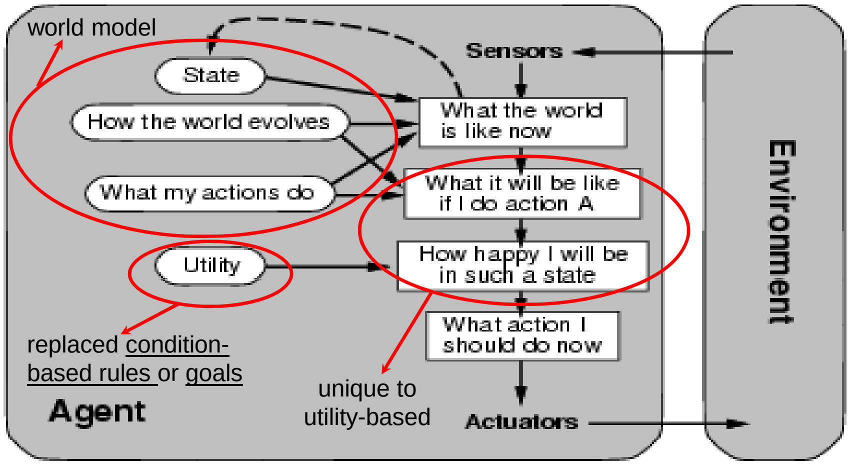
differences from reflexive agents:

- goals are explicit
- the future is taken into account
- reasoning about the future is necessary through search and planning which finds action sequences to achieve goals



Utility-Based Agents

- builds on goal-based agents

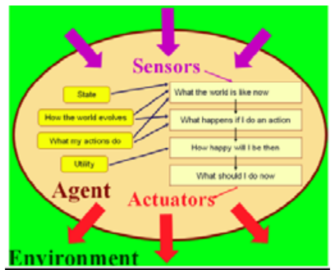


Utility-Based Agents

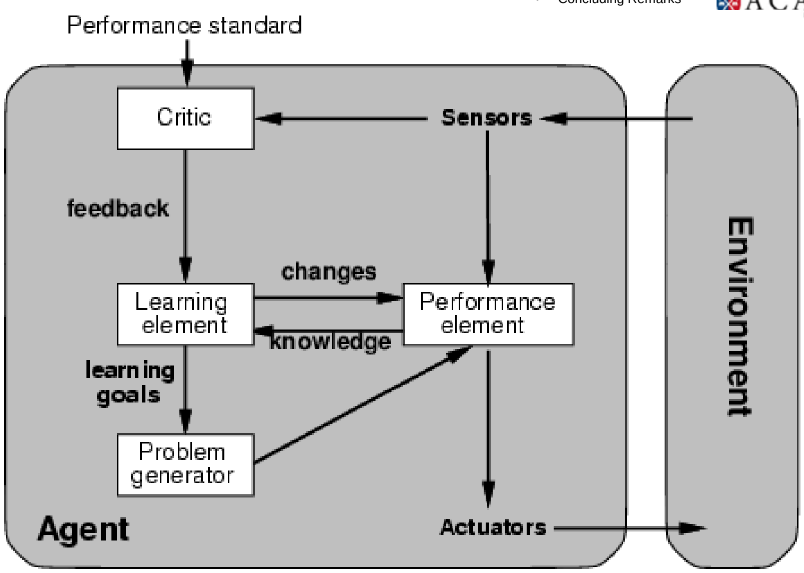
- what if there are multiple alternative ways of achieving the same goal?
- goals provide coarse distinction between “happy” and “unhappy” states
- utility-based agents have finer degrees of comparison between states
- utility-based agent = world model + goals + utility functions

Utility-Based Agents

- utility functions map states to a measure of the utility of the states, often real numbers, they are used to:
  - select between conflicting goals
  - select between alternative ways of achieving a goal
  - deal with cases of multiple goals, none of which can be achieved with certainty – weighing up likelihood of success against importance of goal



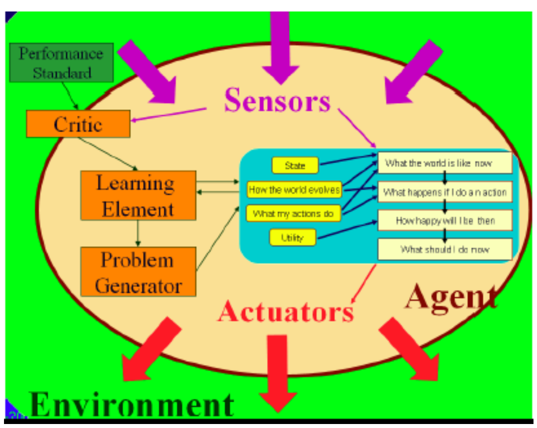
Learning Agents



Learning Agents

- performance element
  - selects actions based on percepts, internal state, background knowledge
  - can be one of the previously described agents
- learning element
  - identifies improvements
- critic
  - provides feedback about agent performance
  - can be external; sometimes part of the environment
- problem generator
  - suggests actions
  - required for novel solutions (creativity)

Learning Agents



Environment

- environments determine to a large degree the interaction between the ‘outside world’ and the agent
  – outside world is not necessarily the ‘real-world’ as we perceive it (nor does it have to be)

- in some cases, environments are implemented within a computer
  – may or may not have any correspondence to the real-world

Environment Types

- fully vs partially observable
  – sensors can measure everything about the environment (what we model through a Markov Decision Process)
- deterministic vs stochastic
  – changes in the environment are predictable
- static vs dynamic
  – no changes while the agent is thinking
- discrete vs continuous
  – limited number of distinct percepts and actions
- single vs multiple agents
  – interaction and collaboration among agents
  – competitive, cooperative, collaborative

Environment

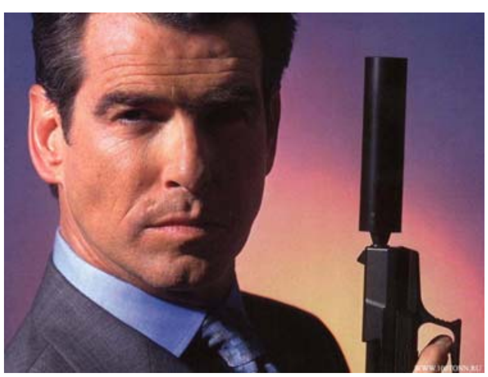
- environment simulators for experiments with agents
  – gives a percept to an agent
  – receives an action
  – updates the environment
- frequently provides mechanisms for measuring the performance of agents
- environment type largely determines the agent design
- real world is (of course) partially observable, stochastic, sequential, dynamic, continuous, multi-agent!

Cool Agent

- Introduction
- Agent Definitions & Terms
- Agent Classifications
- Autonomous Robots & Agents
- Concluding Remarks

DALHOUSIE UNIVERSITY *Inspiring Minds*
ACADIA UNIVERSITY

- human example



007

Cool Agent

- machine example - Mae's IVER2 AUV

Autonomous Robots and Agents

- ways to make controllers (brain = computer = AI)
  1. most robots use rule-based controllers
  2. neural networks
  3. stimulus-response mechanism
     - subsumption architecture (Brooks at MIT)
     - no memory and logical decision
     - hard-wired response to stimulation
  4. intelligent agents
- earlier, we had mentioned that robots are a physical instantiation of an agent (which is, now itself, controlled by an intelligent agent)!
- both true – keep track through context

Multiple Agents and Autonomous Robots

- in agent-based approaches to autonomous robotics, applications are *distributed* among *several* agents
- advantages:
  - increase robustness (fault tolerance) for entire system
  - larger range of task domains
  - greater efficiency (potentially)
  - distributed sensing and action
  - inherent parallelism
  - distributed decision-making and control
- challenges:
  - action selection – what next action to choose
  - conflict resolution or arbitration

7 – Autonomous Robots and Agents

Multi-Agent Systems

- Introduction
- Agent Definitions & Terms
- Agent Classifications
- Autonomous Robots & Agents
- Concluding Remarks

DALHOUSIE UNIVERSITY *Inspiring Minds*
ACADIA UNIVERSITY

- a **multi-agent system** (MAS) is composed of multiple interacting intelligent agents
- multi-agent systems can be used to solve problems which are difficult or impossible for an *individual* agent to solve
  - examples include online trading, disaster response, and modelling social structures, such systems:
    - cannot be handled by centralized systems
    - reflect insight gained in AI, psychology, and sociology that intelligence and interaction are coupled
- can implement complex multi-agent systems with networks of computers

MAS Characteristics

1. each agent has incomplete information or capabilities for solving the problem and, thus, has a limited viewpoint
2. there is no system global control
3. data are decentralized
4. computation is asynchronous

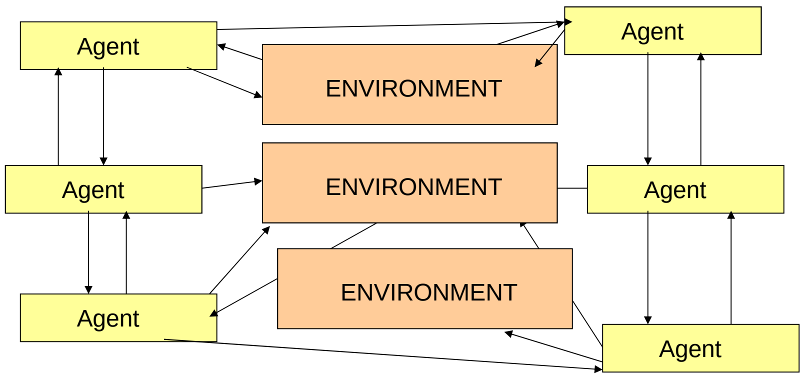
Multi-Agent Systems

- agent interaction requires:
  - communication languages
  - protocols
  - policies

  for MAS to successfully interact they need ability to:
- *coordinate* activities to cooperatively achieve goals
- *cooperate* when their goals are compatible
- *collaborate* when each has insufficient abilities to reach a goal on their own
- *compete* when their goals are incompatible
- *negotiation* – process of reaching agreement; governed by mechanism (rules of encounter – e.g. auctioning)

Multi-Agent Systems

- multiple agents interacting



Multi-Agent Robotics

- early motivation was to design life like robotic systems for:
  - formation
  - foraging
  - soccer (Robosoccer)
- focus on robot-robot and robot-environment interactions which could lead to robust, goal-oriented, and perhaps emergent group behaviors

Multi-Agent Robotics

- particularly challenging tasks include:
  - multi-robot path planning
  - traffic control
  - formation generation, keeping, and control
  - target tracking
  - exploration
  - localization
  - transport
  - collision avoidance
  - lifting an object collaboratively

Multi-Agent Robotics

• Introduction
• Agent Definitions & Terms
• Agent Classifications
• Autonomous Robots & Agents
• Concluding Remarks

DALHOUSIE UNIVERSITY Inspiring Minds
ACADIA UNIVERSITY

- agents inhabiting the physical world are typically robots
- in developing a multi-robot system, one of the primary concerns is how to enable individual robots to automatically program task-handling behaviors adaptive to the dynamic changes in their task environments

Multi-Agent Robotics Example

- Robosoccer is an early multi-agent system application

  http://www.youtube.com/watch?v=f5iyocQcC7I

Multi-Agent Robotics Example

- multi-agent system used to collaboratively lift an object
  - play video 'Coordinated athlete lift' outside of Blackboard Collaborate environment

Multi-Agent Robotics Example

- multi-agent navigation for Mars Exploration Robot
  - play video ‘Autonomous rover navigation’ outside of Blackboard Collaborate environment (3 min)

Concluding Remarks

- agents are a way of conceptualizing distributed systems whether it is across computer systems, robots, or within a robot
- robots are the physical instantiation of agents
- at the same time such robots can use intelligent agents as controllers
- common agent types include: reflexive, model-based, goal-based and utility-based
- agents differ from software in that they can interact with other agents (cooperate, collaborate, negotiate, be in conflict, etc.), are autonomous, can adapt, etc.

References

[1] Poole & Mackworth, Artificial Intelligence: Foundations of Computational Agents, 2010.

[2] Franklin, S., Graesser, A.C., "Is it an agent, or just a program? A taxonomy for autonomous agents," *Intelligent agents iii*:21–35. Berlin: Springer Verlag, 997.

[3] Sloman A., Architectural requirements for human-like agents both natural and artificial (What sort of machines can love?). In K. Dautenhahn (ed.) *Human cognition and social agent technology*. Advances in consciousness research, 163-195, 2000.

[4] Sloman A. & Logan B., Building cognitively rich agents using the Sim_Agent toolkit. Communications of the ACM, 42(3), 71-77, 1999.