CSC 480: Artificial Intelligence

Franz J. Kurfess

Professor
Computer Science Department
California Polytechnic State University
San Luis Obispo, CA, U.S.A.

fkurfess@calpoly.edu

CAL POLY

ignite your passion

Chapter Overview
Intelligent Agents

- **Motivation**
- **Objectives**
- **Introduction**
- **Agents and Environments**
  - "Situated" agents
  - Types of environments
- **Rationality**
  - Expected outcome
  - Performance
- **Agent Structure**
  - Percepts and actions
  - Agent architecture
- **Agent Types**
  - Simple reflex agent
  - Model-based reflex agent
  - Goal-based agent
  - Utility-based agent
  - Learning agent
- **Important Concepts and Terms**
- **Chapter Summary**

Logistics 1

- **Enrollment**
  - ?
- **PolyLearn 480-F01 for both sections**
  - may appear under “Other Courses” for 480-F03
  - groups will be set up for project teams
- **Project Teams & Topics**
  - presentation by Ebru Turgut-Dao (SCE/MUAS)
  - team project description, team members
  - project timeline

Logistics 2

- **Lab and Homework Assignments**
  - Lab 1 due tonight (23:59)
  - Lab 2 available: chat bot agents
    - due next Tue
- **Quizzes on Moodle**
  - Quiz 0 - Background Survey still available
  - Quiz 1 - AI Overview, Introduction
    - open all day today
    - refers to slides 1-Intro.[key|ppt|pdf]
- **"AI Nugget" Presentations**
  - signup sheet being finalized, will be accessible soon

Motivation

- agents are used to provide a consistent viewpoint on various topics in the field AI
- agents require essential skills to perform tasks that require intelligence
- intelligent agents use methods and techniques from the field of AI

Objectives

- **introduce the essential concepts of intelligent agents**
  - focus on rational agents
- **define some basic requirements for the behavior and structure of agents**
- **establish mechanisms for agents to interact with their environment**
  - related to “Interaction Spaces” project discussion (lab)

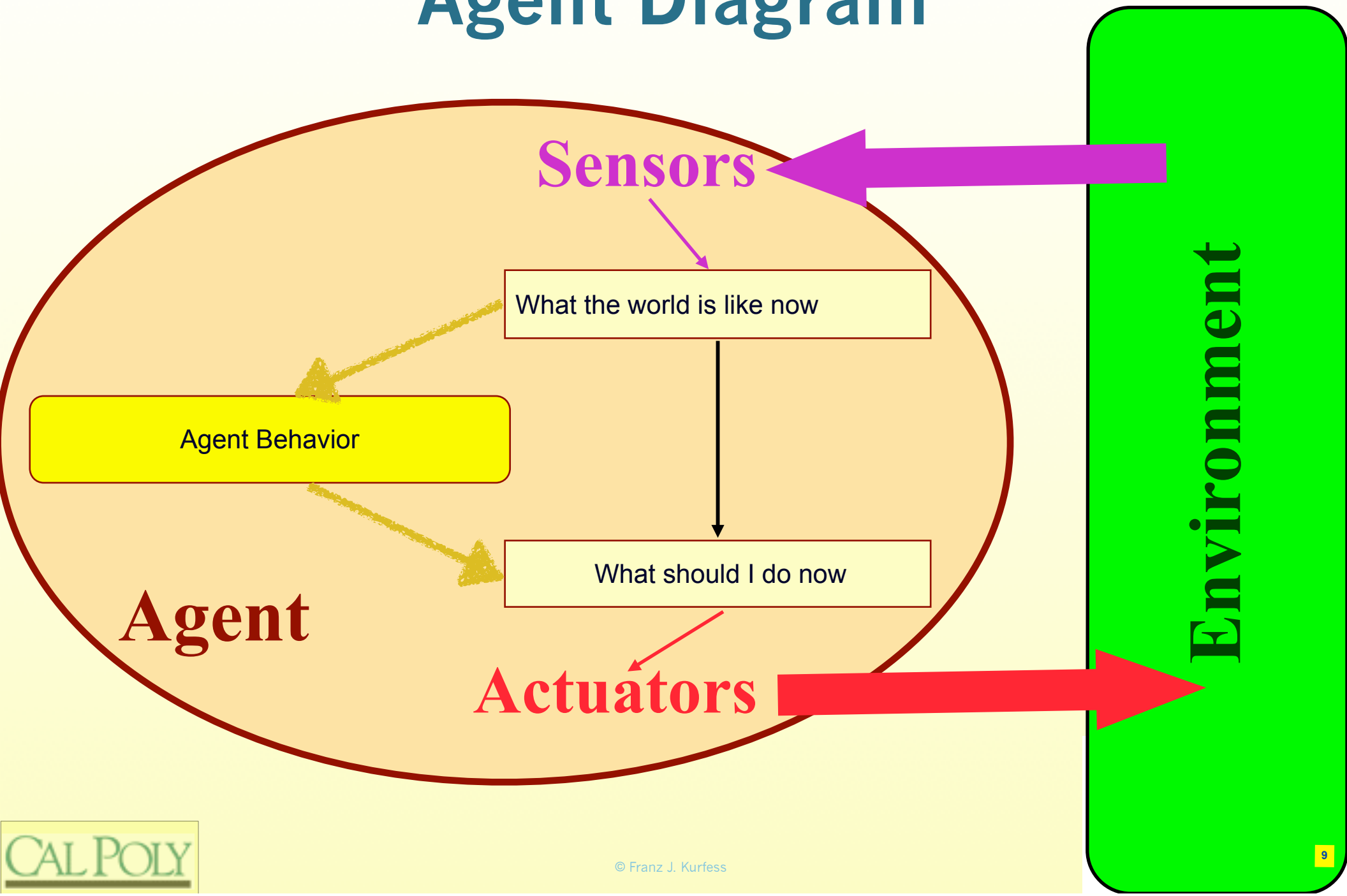
Agent Basics

What is an agent?
Environment
Interaction with the Environment
Agent Performance

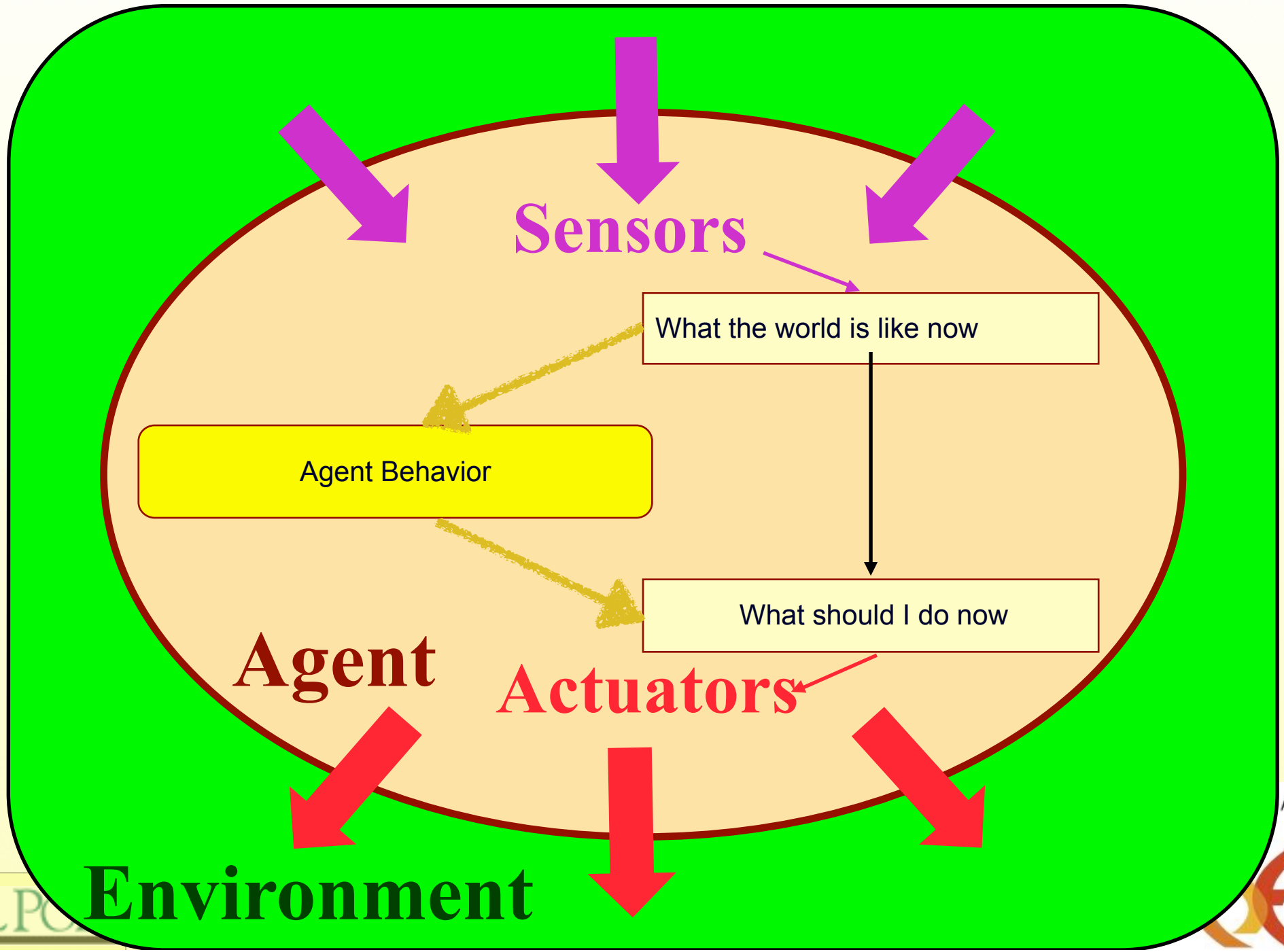
What is an Agent?

- **in general, an entity that interacts with its environment**
  - perception through sensors
  - actions through effectors or actuators

Agent Diagram



Agent Diagram 2



Examples of Agents

- human agent
  - eyes, ears, skin, taste buds, etc. for sensors
  - hands, fingers, legs, mouth, etc. for actuators
    - powered by muscles
- robot
  - camera, infrared, bumper, etc. for sensors
  - grippers, wheels, lights, speakers, etc. for actuators
    - often powered by motors
- software agent
  - input parameters as sensors
    - information provided as input to functions in the form of encoded bit strings or symbols
  - output parameters as actuators
    - results deliver the output

Agents and Environments

- **an agent perceives its environment through sensors**
  - the complete set of inputs at a given time is called a percept
  - the current percept, or a sequence of percepts may influence the actions of an agent
- **it can change the environment through actuators**
  - an operation involving an actuator is called an action
  - actions can be grouped into action sequences

Agents and Their Actions

- **a *rational agent* does "the right thing"**
  - the action that leads to the best outcome under the given circumstances
- **an *agent function* maps percept sequences to actions**
  - abstract mathematical description
- **an *agent program* is a concrete implementation of the respective function**
  - it runs on a specific agent architecture ("platform")
- **problems:**
  - what is "the right thing"?
  - how do you measure the "best outcome"?

Performance of Agents

- **criteria for measuring performance aspects**
  - goals vs. outcome
  - expenses of the agent
    - resource consumption
    - time
  - often subjective, but should be objective
  - task dependent
  - time may be important

Performance Evaluation Examples

- **vacuum agent**
  - number of tiles cleaned during a certain period
    - based on the agent's report, or validated by an objective authority
    - doesn't consider expenses of the agent, side effects
      - energy, noise, loss of useful objects, damaged furniture, scratched floor
    - might lead to unwanted activities
      - agent re-cleans clean tiles, covers only part of the room, drops dirt on tiles to have more tiles to clean, etc.

Rational Agent

- **selects the action that is expected to maximize its performance**
  - based on a defined performance measure
    - ideally objective and measurable
    - should allow comparisons
      - between different instances of attempts at the same task
      - between agents for the same task
  - depends on various factors
    - agent-specific
      - percept sequence, background knowledge, feasible actions, ...
    - external
      - environment, other agents, random events, ...

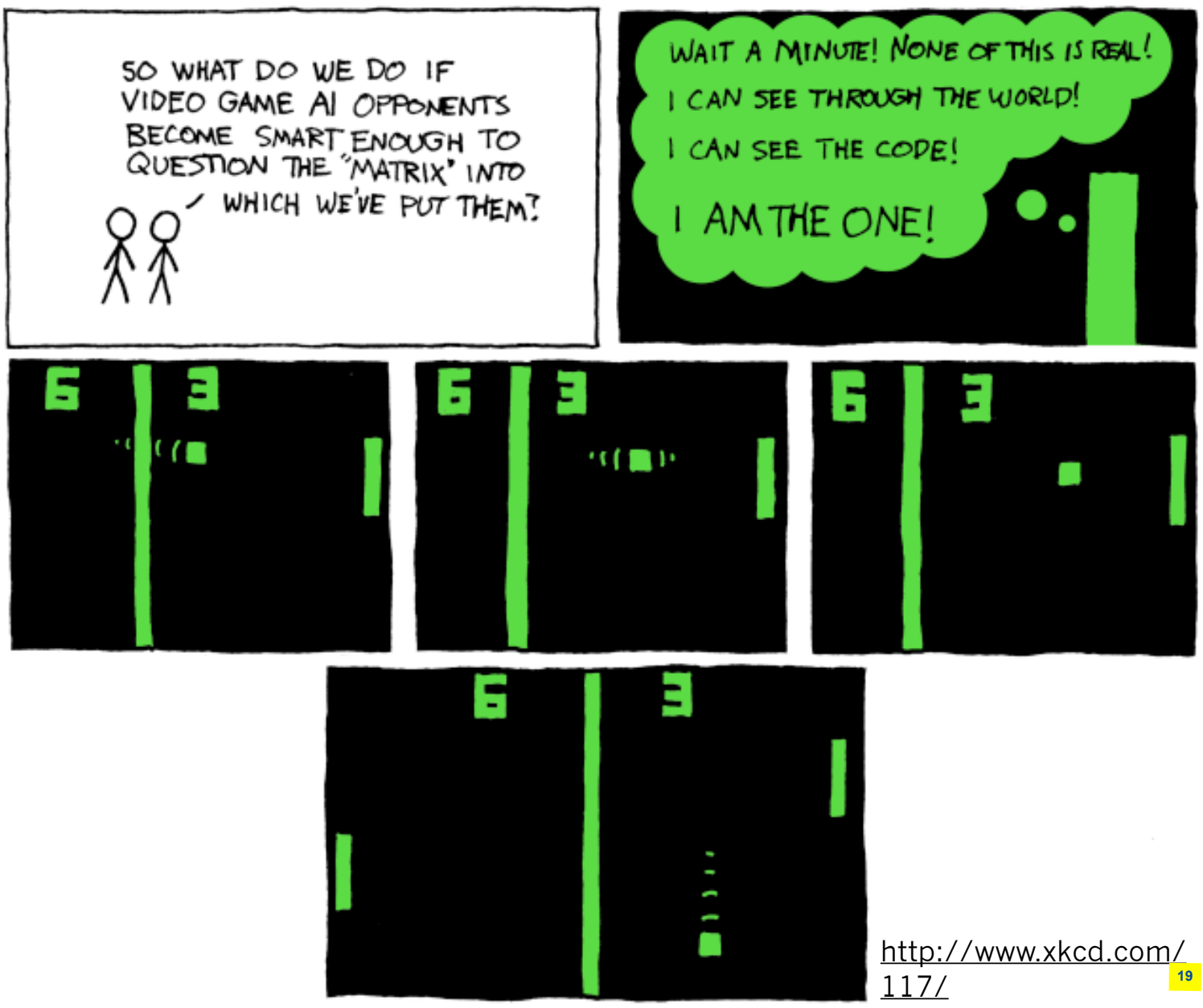
Rational Agent Considerations

- **performance measure for the successful completion of a task**
  - objective
  - ordering function
- **perceptual history (percept sequence)**
  - complete
    - all past percepts in full detail
  - partial
    - older percepts omitted
    - details omitted
- **background knowledge**
  - especially about the environment
    - dimensions, structure, basic “laws”
  - task, user, other agents
- **feasible actions**
  - capabilities of the agent

Omniscience

- **a rational agent is not omniscient**
  - it doesn't know the actual outcome of its actions
  - it may not know certain aspects of its environment
- **rationality takes into account the limitations of the agent**
  - percept sequence, background knowledge, feasible actions
  - it deals with the expected outcome of actions

- **Lab 10 Submissio**
  - by Justin Helmer
  - A note on omnis



http://www.xkcd.com/117/

Environments

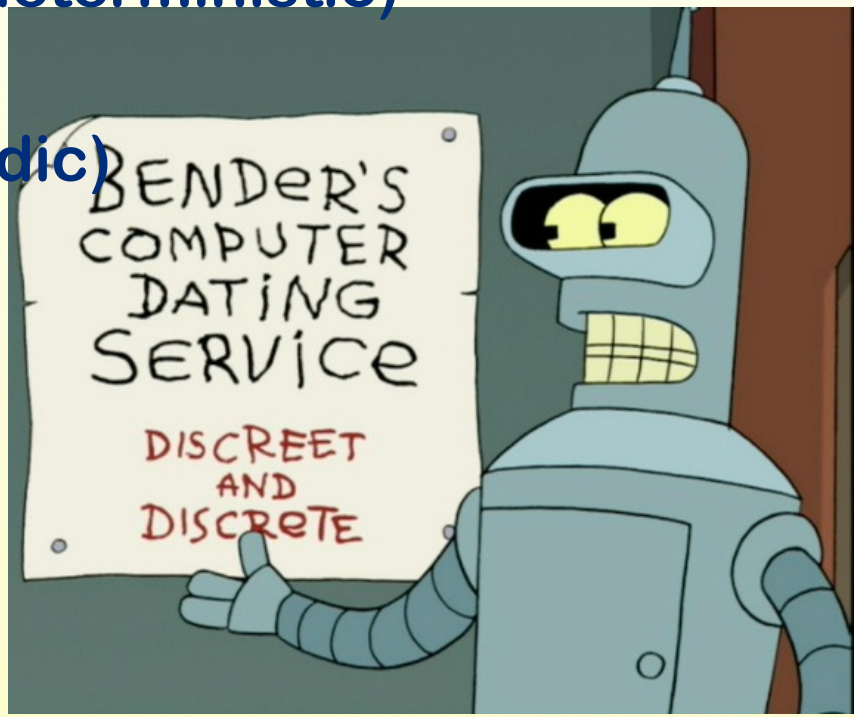
environment properties
environment programs

Environments

- **determine to a large degree the interaction between the "outside world" and the agent**
  - the "outside world" is not necessarily the "real world" as we perceive it
    - it may be a real or virtual environment the agent lives in
- **in many cases, environments are implemented within computers**
  - they may or may not have a close correspondence to the "real world"

Environment Properties

- **fully observable vs. partially observable**
  - sensors capture all relevant information from the environment
- **deterministic vs. stochastic (non-deterministic)**
  - changes in the environment are predictable
- **episodic vs. sequential (non-episodic)**
  - independent perceiving-acting episodes
- **static vs. dynamic**
  - no changes while the agent is "thinking"
- **discrete vs. continuous**
  - limited number of distinct percepts/actions
- **single vs. multiple agents**
  - interaction and collaboration among agents
  - competitive, cooperative



Environment Programs

- **environment simulators for experiments with agents**
  - gives a percept to an agent
  - receives an action
  - updates the environment
- **often divided into environment classes for related tasks or types of agents**
- **the environment frequently provides mechanisms for measuring the performance of agents**

Structure of Agents

Percepts and Actions
PEAS Descriptions
Software Agents

From Percepts to Actions

- **mapping from percept sequences to actions**
  - if an agent only reacts to its percepts, a table can describe this mapping
  - instead of a table, a simple function may also be used
    - can be conveniently used to describe simple agents that solve well-defined problems in a well-defined environment
      - e.g. calculation of mathematical functions
  - serious limitations
    - see discussion of "reflex agents"

Agent or Program

- **our criteria so far seem to apply equally well to software agents and to regular programs**
- **autonomy**
  - agents solve tasks largely independently
  - programs depend on users or other programs for “guidance”
  - autonomous systems base their actions on their own experience and knowledge
  - requires initial knowledge together with the ability to learn
  - provides flexibility for more complex tasks

Structure of Intelligent Agents

- **Agent = Architecture + Program**
- **architecture**
  - operating platform of the agent
    - computer system, specific hardware, possibly OS functions
- **program**
  - function that implements the mapping from percepts to actions

emphasis in this course is on the *program* aspect, not on the *architecture*

Software Agents

- **also referred to as “soft bots”**
- **live in artificial environments where computers and networks provide the infrastructure**
- **may be very complex with strong requirements on the agent**
  - World Wide Web, real-time constraints,
- **natural and artificial environments may be merged**
  - user interaction
  - sensors and actuators in the real world
    - camera, temperature, arms, wheels, etc.
  - augmented reality

PEAS Description of Task Environments

- **Performance Measures**
  - used to evaluate how well an agent solves the task at hand
- **Environment**
  - surroundings beyond the control of the agent
- **Actuators**
  - determine the actions the agent can perform
- **Sensors**
  - provide information about the current state of the environment

Exercise: VacBot PEAS Description

- use the PEAS template to determine important aspects for a VacBot agent

PEAS Description Template

used for high-level characterization of agents

- **Performance Measures**

  How well does the agent solve the task at hand?
  How is this measured?

- **Environment**

  Important aspects of the surroundings beyond the control of the agent:

- **Actuators**

  Determine the actions the agent can perform.

- **Sensors**

  Provide information about the current state of the environment.

Agent Programs

“Skeleton” Agent Program
Table Agent Program

Agent Programs

- **the emphasis in this course is on programs that specify the agent's behavior through mappings from percepts to actions**
  - less on environment and goals
- **agents receive one percept at a time**
  - they may or may not keep track of the percept sequence
- **performance evaluation is often done by an outside authority, not the agent**
  - more objective, less complicated
  - can be integrated with the environment program

Skeleton Agent Program

- basic framework for an agent program

```
function SKELETON-AGENT(percept) returns action
  static: memory

  memory    := UPDATE-MEMORY(memory, percept)
  action    := CHOOSE-BEST-ACTION(memory)
  memory    := UPDATE-MEMORY(memory, action)

return action
```

Look it up!

- **simple way to specify a mapping from percepts to actions**
  - tables may become very large
  - almost all work done by the designer
  - no autonomy, all actions are predetermined
    - with well-designed and sufficiently complex tables, the agent may appear autonomous to an observer, however
  - learning might take a very long time
    - so long that it is impractical
    - there are better learning methods

Table Agent Program

- agent program based on table lookup

```
function TABLE-DRIVEN-AGENT(percept) returns action

  static: percepts // initially empty sequence*
          table    // indexed by percept sequences
                   // initially fully specified

  append percept to the end of percepts

  action     := LOOKUP(percepts, table)

return action
```

* Note:the storage of percepts requires writeable memory

Agent Types

simple reflex agents
model-based agents
goal-based agents
utility-based agents
learning agents

Agent Program Types

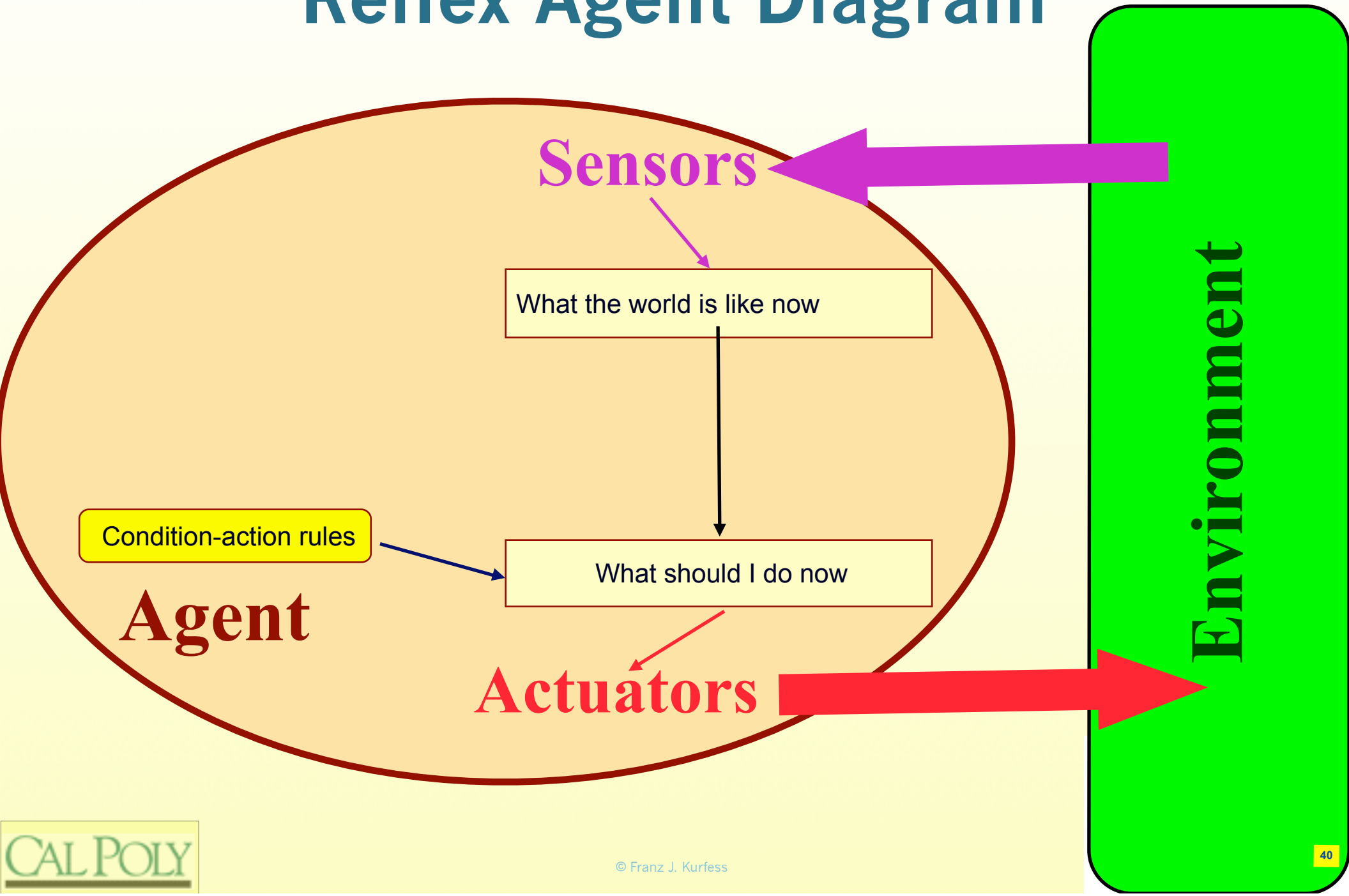
- different ways of achieving the mapping from percepts to actions
- different levels of complexity

- **simple reflex agents**
  - mapping percepts to actions
- **model-based agents**
  - keeping track of the world
- **goal-based agents**
  - working towards a goal
- **utility-based agents**
  - distinction between multiple goals, priorities
- **learning agents**
  - performance improvement over time

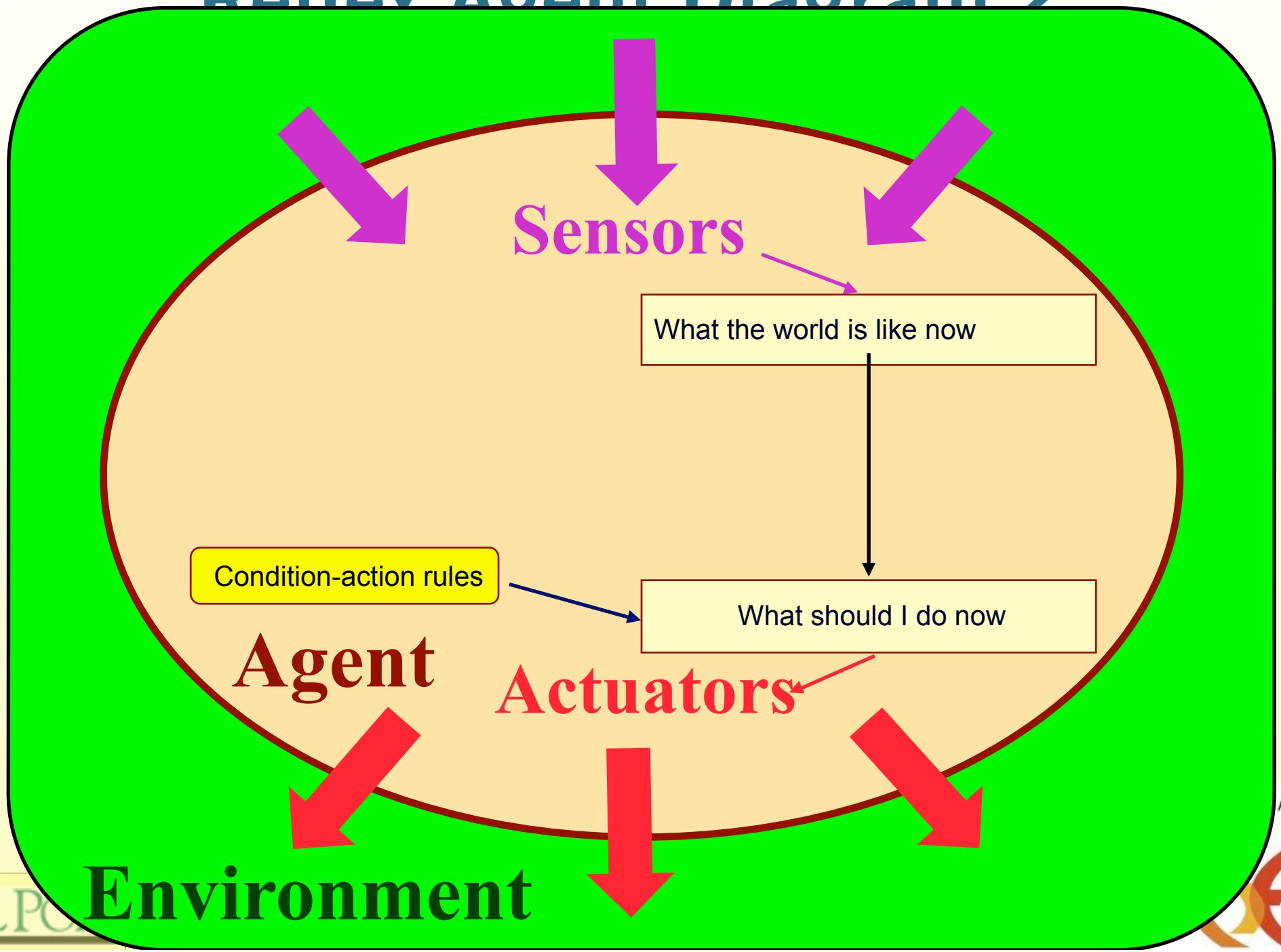
Simple Reflex Agent

- **instead of specifying individual mappings in an explicit table, common input-output associations are recorded**
  - requires processing of percepts to achieve some abstraction
  - frequent method of specification is through condition-action rules
    - **if** *percept* **then** *action*
  - similar to innate reflexes or learned responses in humans
  - efficient implementation, but limited power
    - environment must be fully observable
    - easily runs into infinite loops

Reflex Agent Diagram



Reflex Agent Diagram 2



Reflex Agent Program

- application of simple rules to situations

```
function SIMPLE-REFLEX-AGENT(percept) returns action
  static: rules   //set of condition-action rules

  condition := INTERPRET-INPUT(percept)
  rule              := RULE-MATCH(condition, rules)
  action            := RULE-ACTION(rule)

return action
```

Exercise: VacBot Reflex Agent

- specify a core set of condition-action rules for a VacBot agent

Politician Reflex Agent

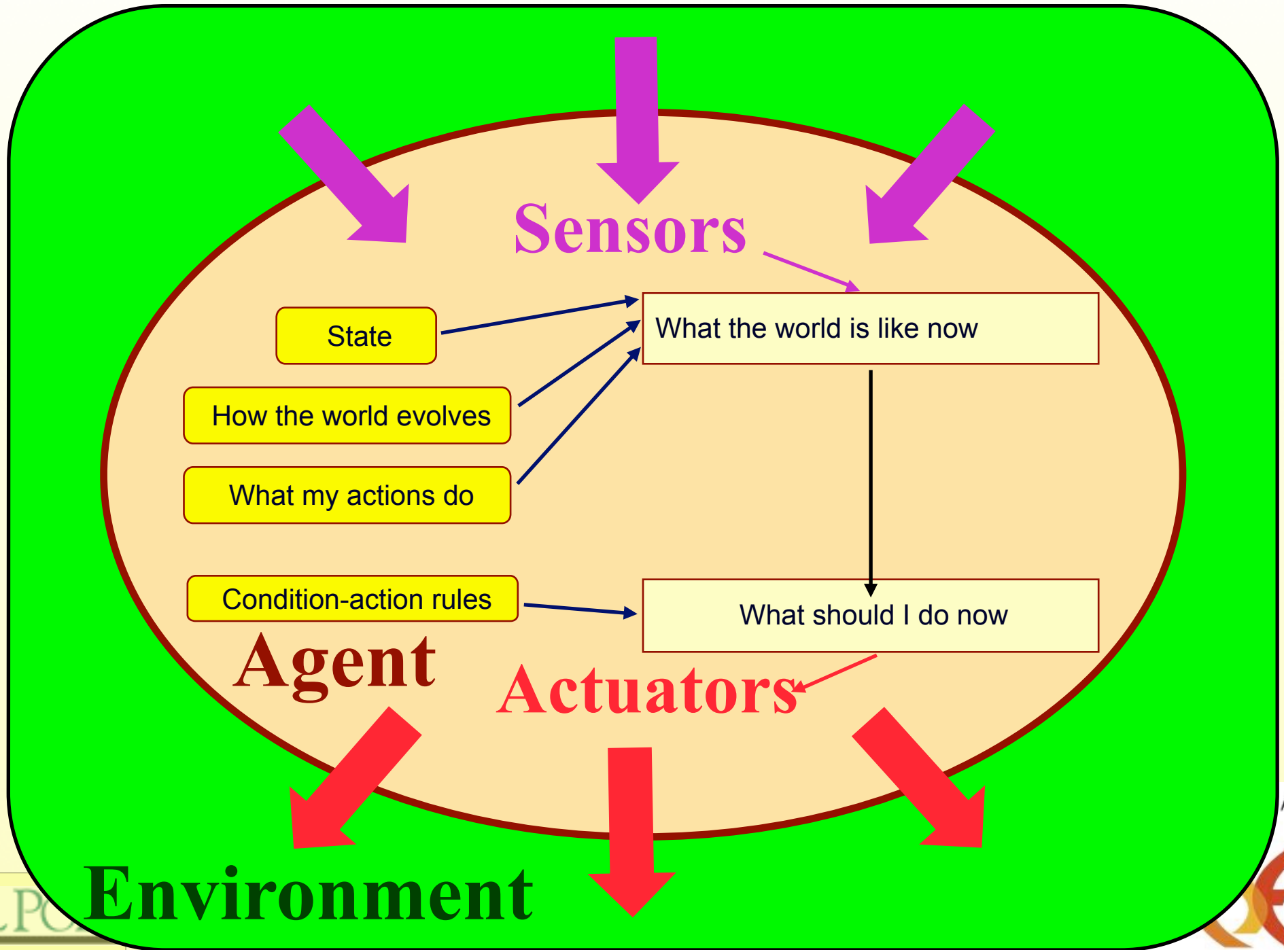
- **Lab 10 Submission: AI and Humor -> The**
  - by Donald Mitchell · Tuesday, November 27,
  - one among several AI cartoons

http://www.cartoonstock.com/newscartoons/cartoonists/lla/lowres/llan408l.jpg

Model-Based Reflex Agent

- **an internal state maintains important information from previous percepts**
  - sensors only provide a partial picture of the environment
  - helps with some partially observable environments
- **the internal states reflects the agent's knowledge about the world**
  - this knowledge is called a *model*
  - may contain information about changes in the world
    - caused by actions of the action
    - independent of the agent's behavior

Model-Based Reflex Agent



Model-Based Reflex Agent Program

- application of simple rules to situations

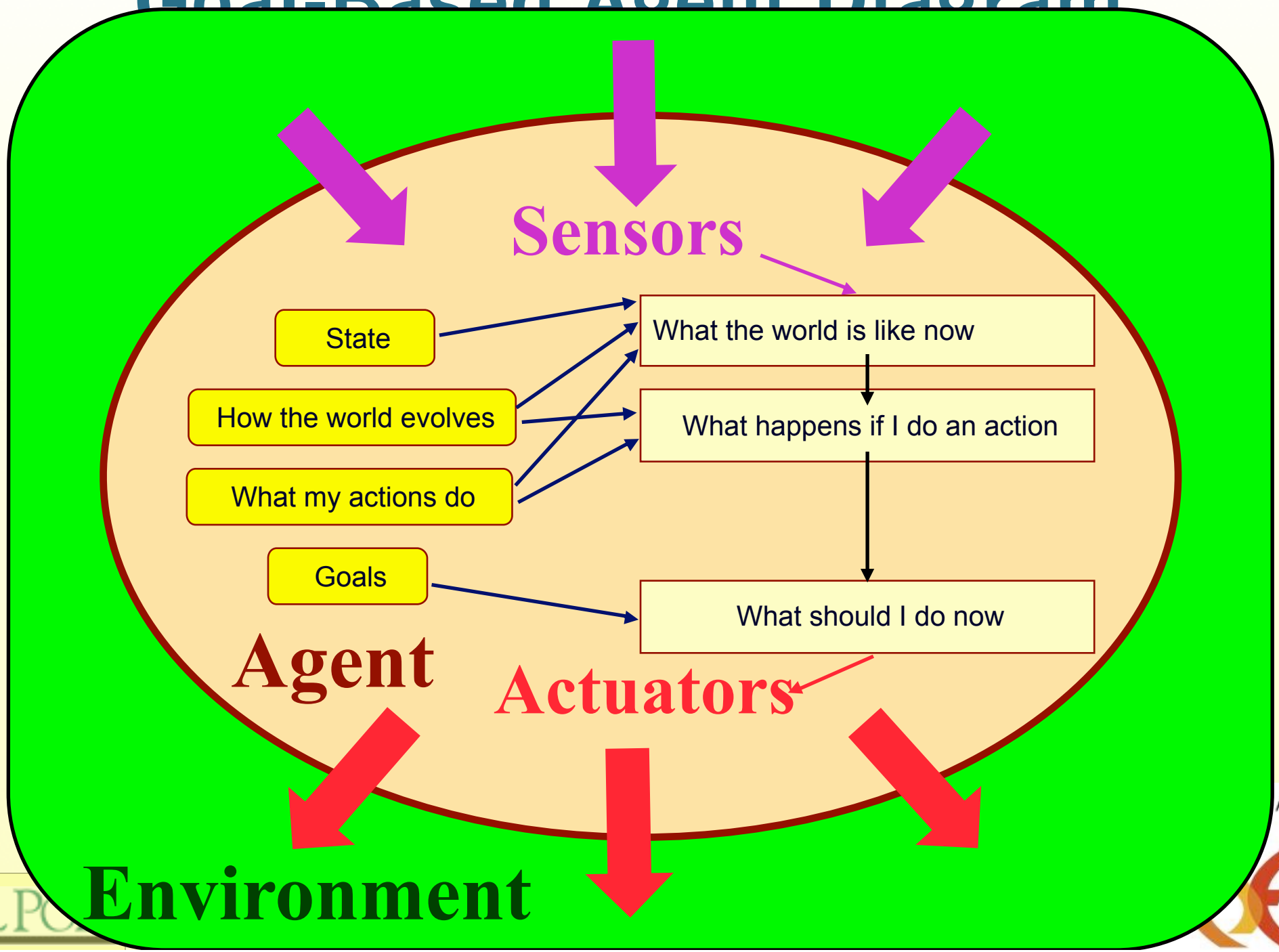
```
function REFLEX-AGENT-WITH-STATE(percept) returns action
  static: rules      //set of condition-action rules
          state      //description of the current world state
          action     //most recent action, initially none

  state       := UPDATE-STATE(state, action, percept)
  rule := RULE-MATCH(state, rules)
  action      := RULE-ACTION[rule]
  return action
```

Goal-Based Agent

- **the agent tries to reach a desirable state, the *goal***
  - may be provided from the outside (user, designer, environment), or inherent to the agent itself
- **results of possible actions are considered with respect to the goal**
  - easy when the results can be related to the goal after each action
  - in general, it can be difficult to attribute goal satisfaction results to individual actions
  - may require consideration of the future
    - what-if scenarios
    - search, reasoning or planning
- **very flexible, but not very efficient**

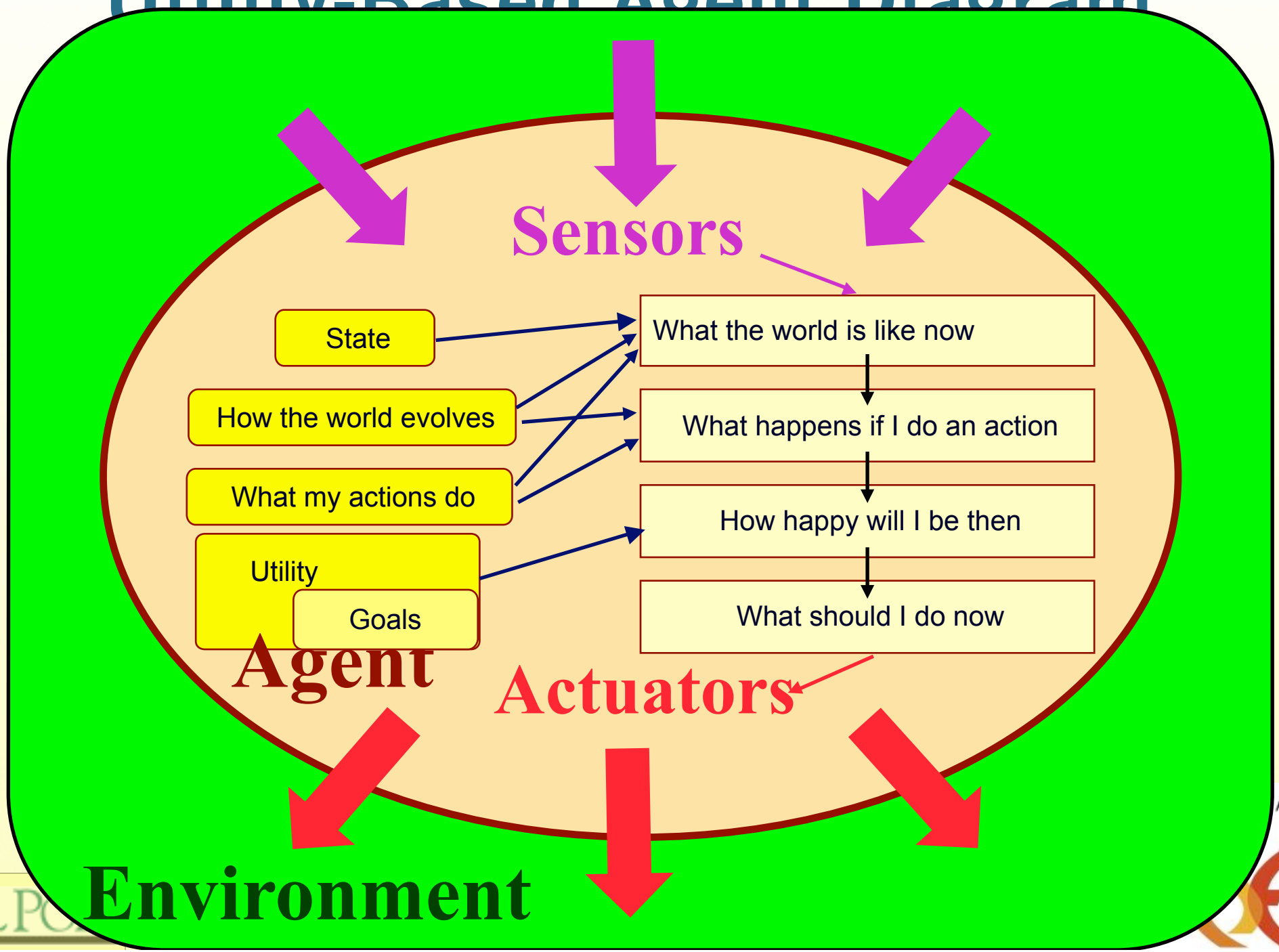
Goal-Based Agent Diagram



Utility-Based Agent

- **more sophisticated distinction between different world states**
  - a utility function maps states onto a real number
    - may be interpreted as “degree of happiness”
  - permits rational actions for more complex tasks
    - resolution of conflicts between goals (tradeoff)
    - multiple goals (likelihood of success, importance)
    - a utility function is necessary for rational behavior, but sometimes it is not made explicit

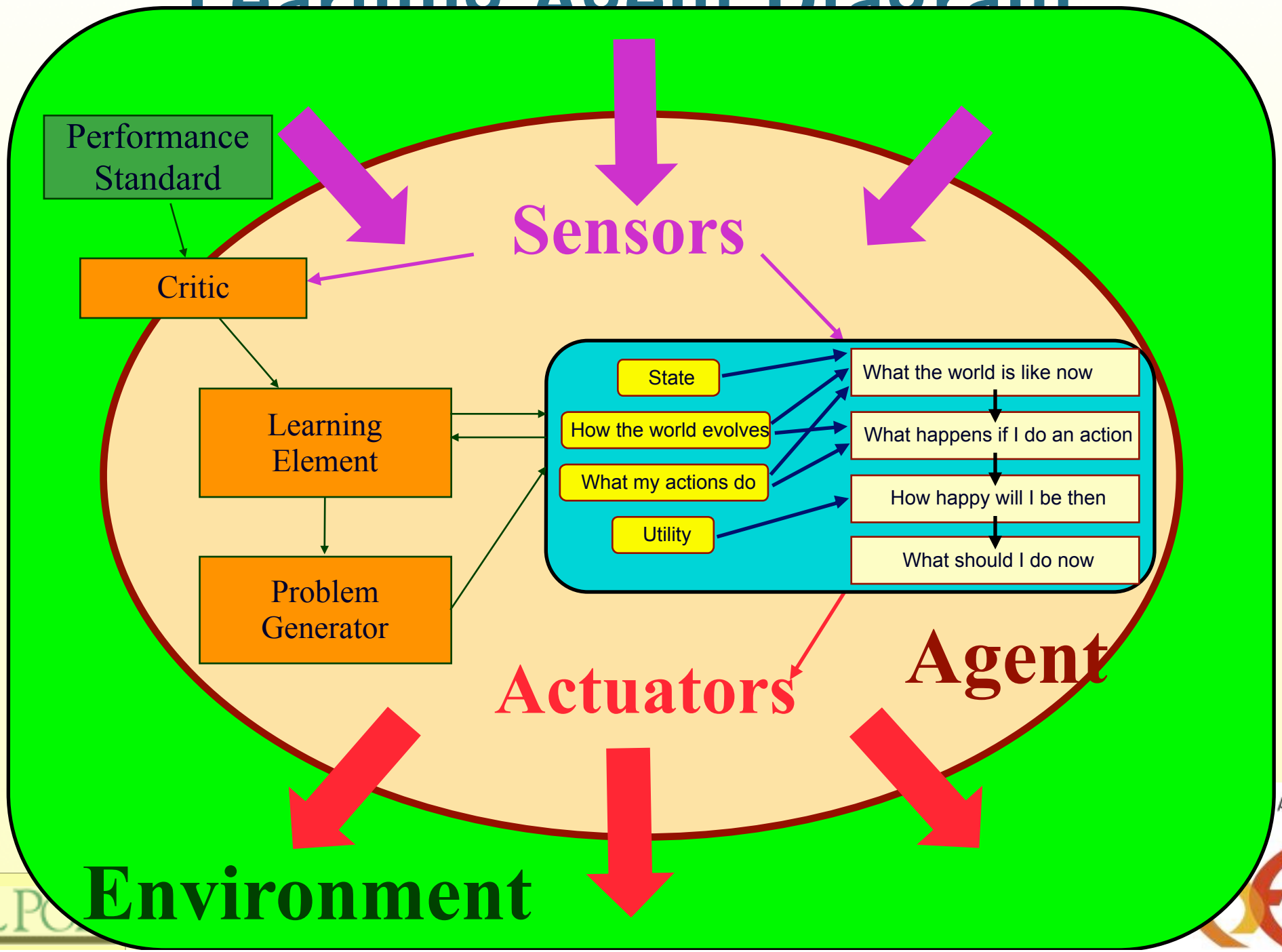
Utility-Based Agent Diagram



Learning Agent

- **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 the performance of the agent
  - can be external; sometimes part of the environment
- **problem generator**
  - suggests actions
  - required for novel solutions (creativity

Learning Agent Diagram



Important Concepts and Terms

- action
- actuator
- agent
- agent program
- architecture
- autonomous agent
- continuous environment
- deterministic environment
- discrete environment
- episodic environment
- goal
- intelligent agent
- knowledge representation
- mapping
- multi-agent environment
- observable environment
- omniscient agent
- PEAS description
- percept
- percept sequence
- performance measure
- rational agent
- reflex agent
- robot
- sensor
- sequential environment
- software agent
- state
- static environment
- stochastic environment
- utility

Chapter Summary

- **agents perceive and act in an environment**
- **ideal agents maximize their performance measure**
  - autonomous agents act independently
- **basic agent types**
  - simple reflex
  - model-based (reflex with state)
  - goal-based
  - utility-based
  - learning
- **some environments may make life harder for agents**
  - inaccessible, non-deterministic, non-episodic, dynamic, continuous