CSC 480: Artificial Intelligence

Dr. Franz J. Kurfess
Computer Science Department
Cal Poly

© 2000-2012 Franz Kurfess

Course Overview

- Introduction
- Intelligent Agents
- Search
  - problem solving through search
  - informed search
- Games
  - games as search problems
- Knowledge and Reasoning
  - reasoning agents
  - propositional logic
  - predicate logic
  - knowledge-based systems
- Learning
  - learning from observation
  - neural networks
- Conclusions

© 2000-2012 Franz Kurfess

Chapter Overview
Intelligent Agents

- Motivation
- Objectives
- Introduction
- Agents and Environments
- Rationality
- Agent Structure
- Agent Types
  - Simple reflex agent
  - Model-based reflex agent
  - Goal-based agent
  - Utility-based agent
  - Learning agent
- Important Concepts and Terms
- Chapter Summary

© 2000-2012 Franz Kurfess

Logistics

- Project Team Wikis, pages
  - project description
  - team members
- PolyLearn
  - Section 03 merged into Section 01
  - groups set up for project teams
- Lab and Homework Assignments
  - Lab 1 due tonight (23:59)
  - Lab 2 available: simple agents in the BotEnvironment
- Quizzes
  - Quiz 0 - Background Survey still available
  - Quiz 1 - Available Tue, Sep. 24, all day (0:00 - 23:59)

© 2000-2012 Franz Kurfess

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

© 2000-2012 Franz Kurfess

Objectives

- introduce the essential concepts of intelligent agents
- define some basic requirements for the behavior and structure of agents
- establish mechanisms for agents to interact with their environment

© 2000-2012 Franz Kurfess

What is an Agent?

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

© 2000-2012 Franz Kurfess

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
  - functions as sensors
    - information provided as input to functions in the form of encoded bit strings or symbols
  - functions as actuators
    - results deliver the output

© 2000-2012 Franz Kurfess

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

© 2000-2012 Franz Kurfess

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”

© 2000-2012 Franz Kurfess

Performance of Agents

- criteria for measuring the outcome and the expenses of the agent
  - often subjective, but should be objective
  - task dependent
  - time may be important

© 2000-2012 Franz Kurfess

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.

© 2000-2012 Franz Kurfess

Rational Agent

- selects the action that is expected to maximize its performance
  - based on a performance measure
  - depends on the percept sequence, background knowledge, and feasible actions

© 2000-2012 Franz Kurfess

Rational Agent Considerations

- performance measure for the successful completion of a task
- complete perceptual history (percept sequence)
- background knowledge
  - especially about the environment
    - dimensions, structure, basic “laws”
  - task, user, other agents
- feasible actions
  - capabilities of the agent

© 2000-2012 Franz Kurfess

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

© 2000-2012 Franz Kurfess

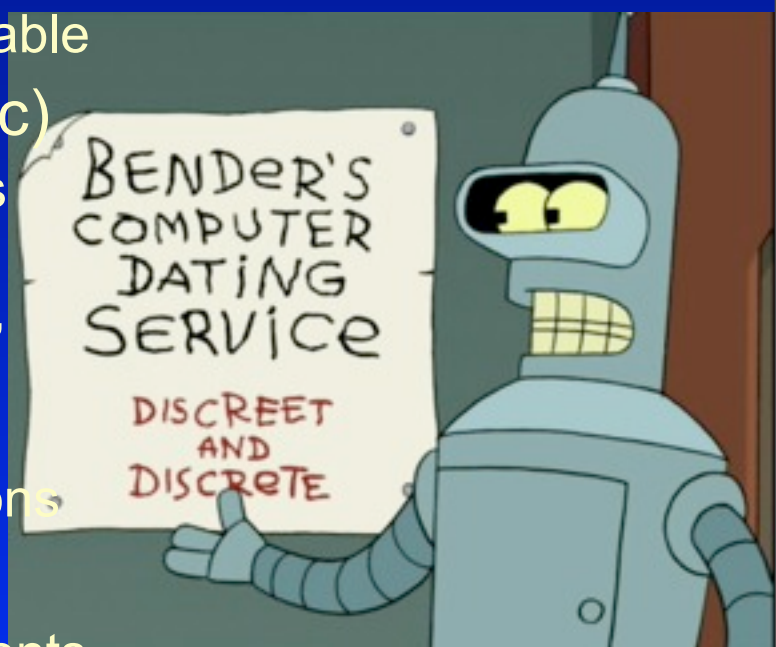
Environments

- 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”

© 2000-2012 Franz Kurfess

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



© 2000-2012 Franz Kurfess

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

© 2000-2012 Franz Kurfess

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”

© 2000-2012 Franz Kurfess

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

© 2000-2012 Franz Kurfess

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*

© 2000-2012 Franz Kurfess

Software Agents

- also referred to as “softbots”
- 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.

© 2000-2012 Franz Kurfess

PEAS Description of Task Environments

used for high-level characterization of agents

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

© 2000-2012 Franz Kurfess

Exercise: VacBot Peas Description

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

© 2000-2012 Franz Kurfess

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 theurroundings beyond the control of the agent:

Actuators Determine the actions the agent can perform.

Sensors Provide information about the current state of the environment.

© 2000-2012 Franz Kurfess

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

© 2000-2012 Franz Kurfess

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
```

© 2000-2012 Franz Kurfess

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

© 2000-2012 Franz Kurfess

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

© 2000-2012 Franz Kurfess

Agent Program Types

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

- simple reflex agents
- model-based agents
  - keep track of the world
- goal-based agents
  - work towards a goal
- utility-based agents
- learning agents

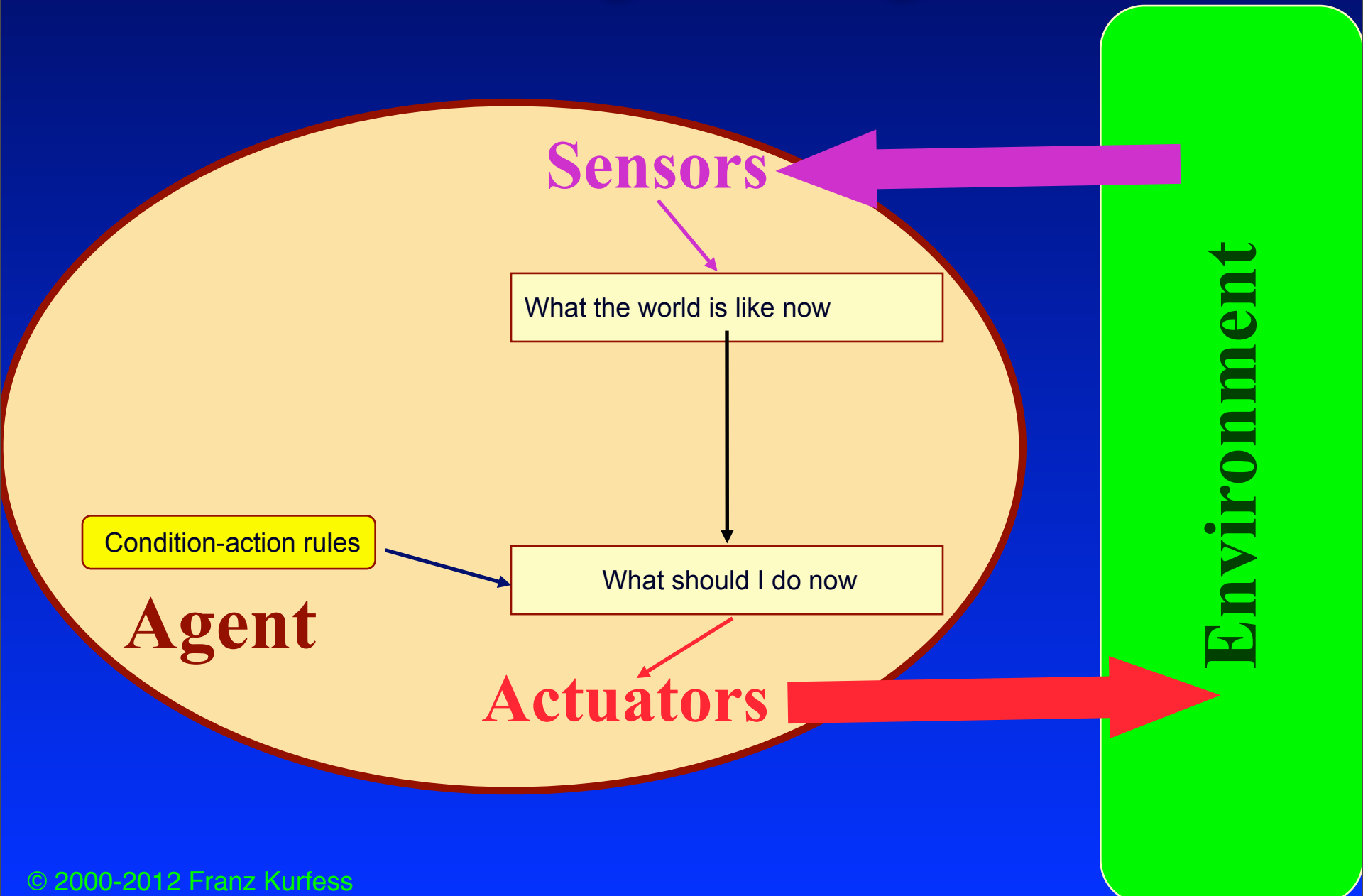
© 2000-2012 Franz Kurfess

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

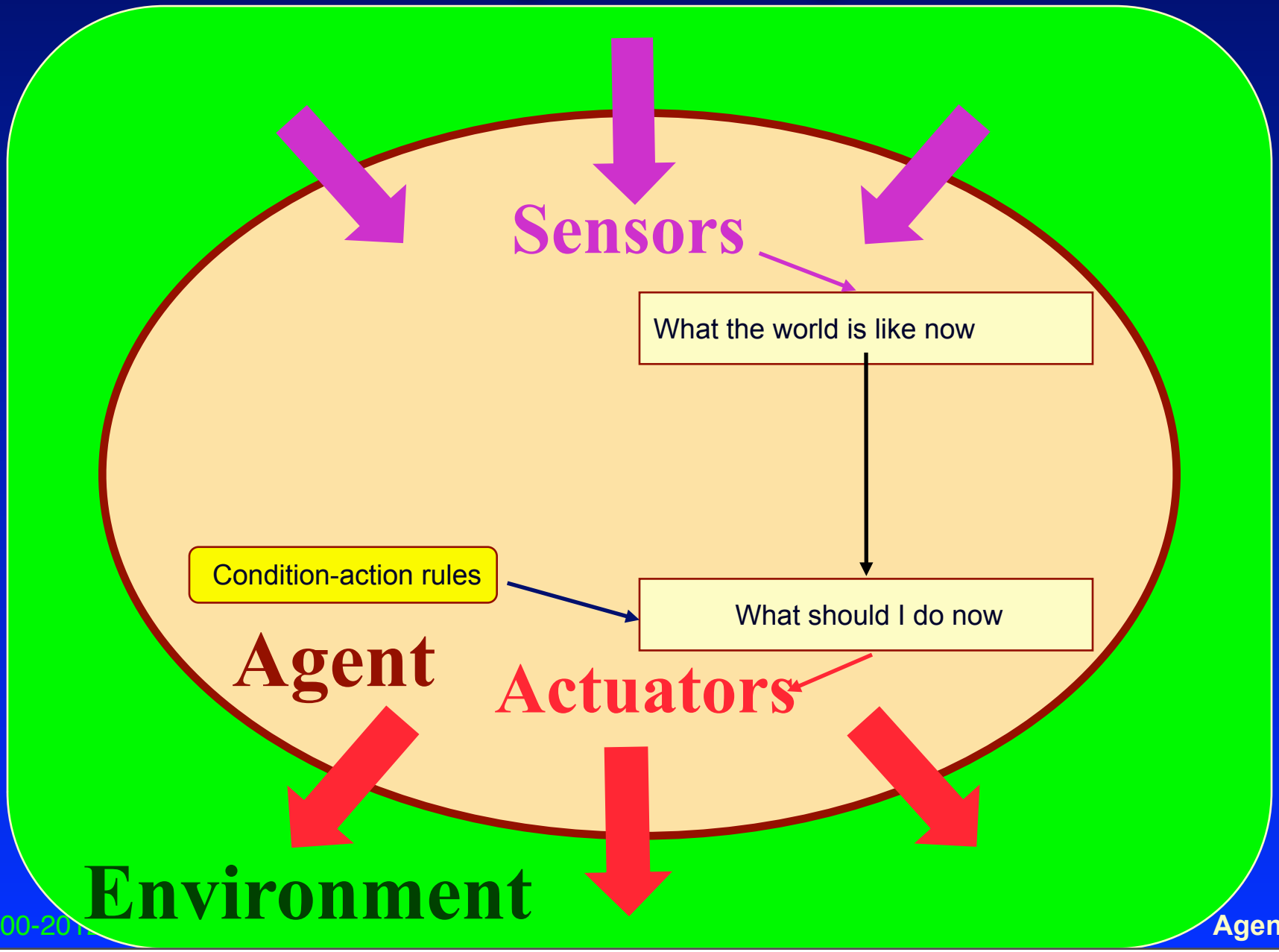
© 2000-2012 Franz Kurfess

Reflex Agent Diagram



© 2000-2012 Franz Kurfess

Reflex Agent Diagram 2



© 2000-201

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
```

© 2000-2012 Franz Kurfess

Exercise: VacBot Reflex Agent

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

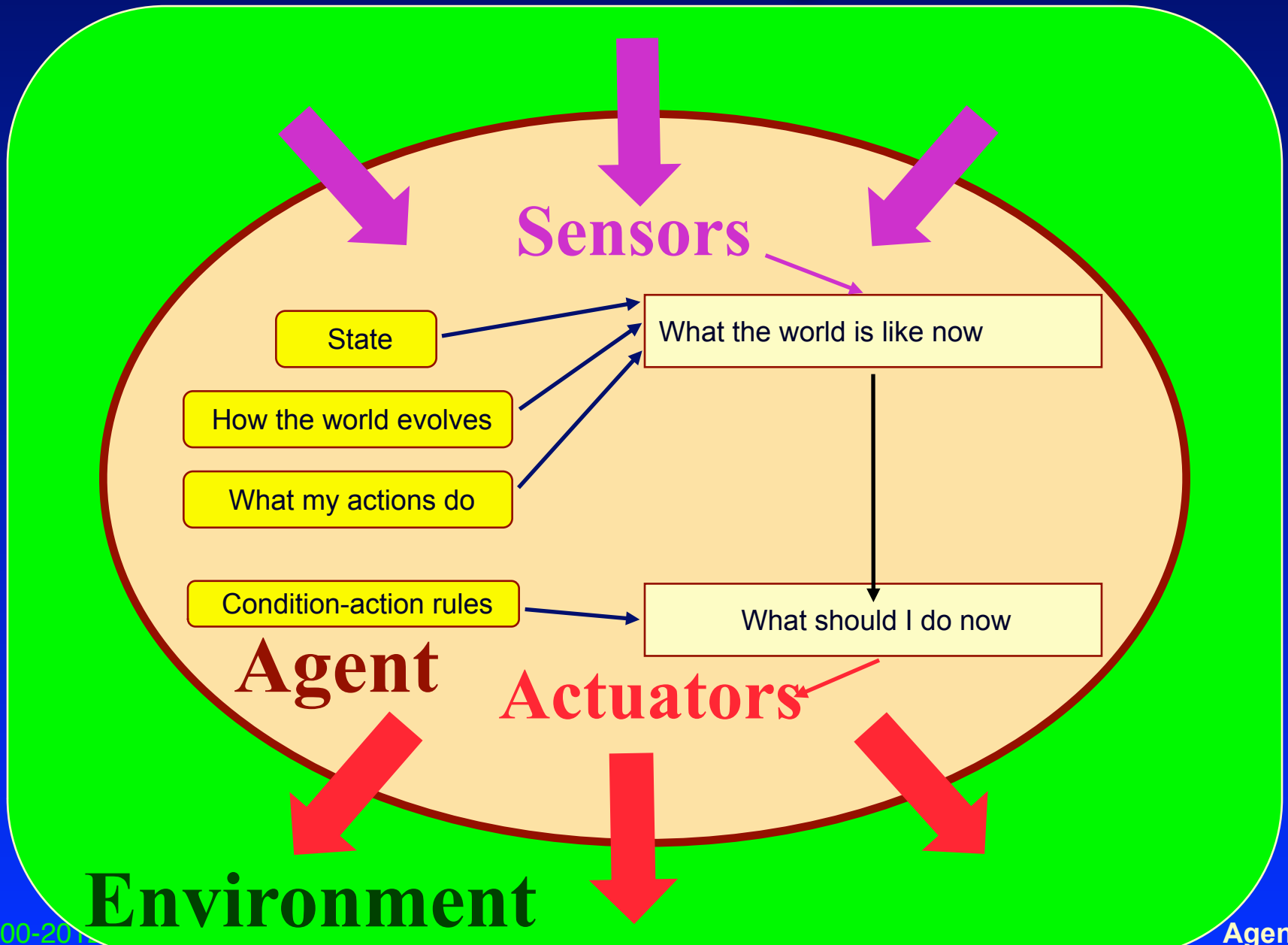
© 2000-2012 Franz Kurfess

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

© 2000-2012 Franz Kurfess

Model-Based Reflex Agent Diagram



© 2000-201

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
```

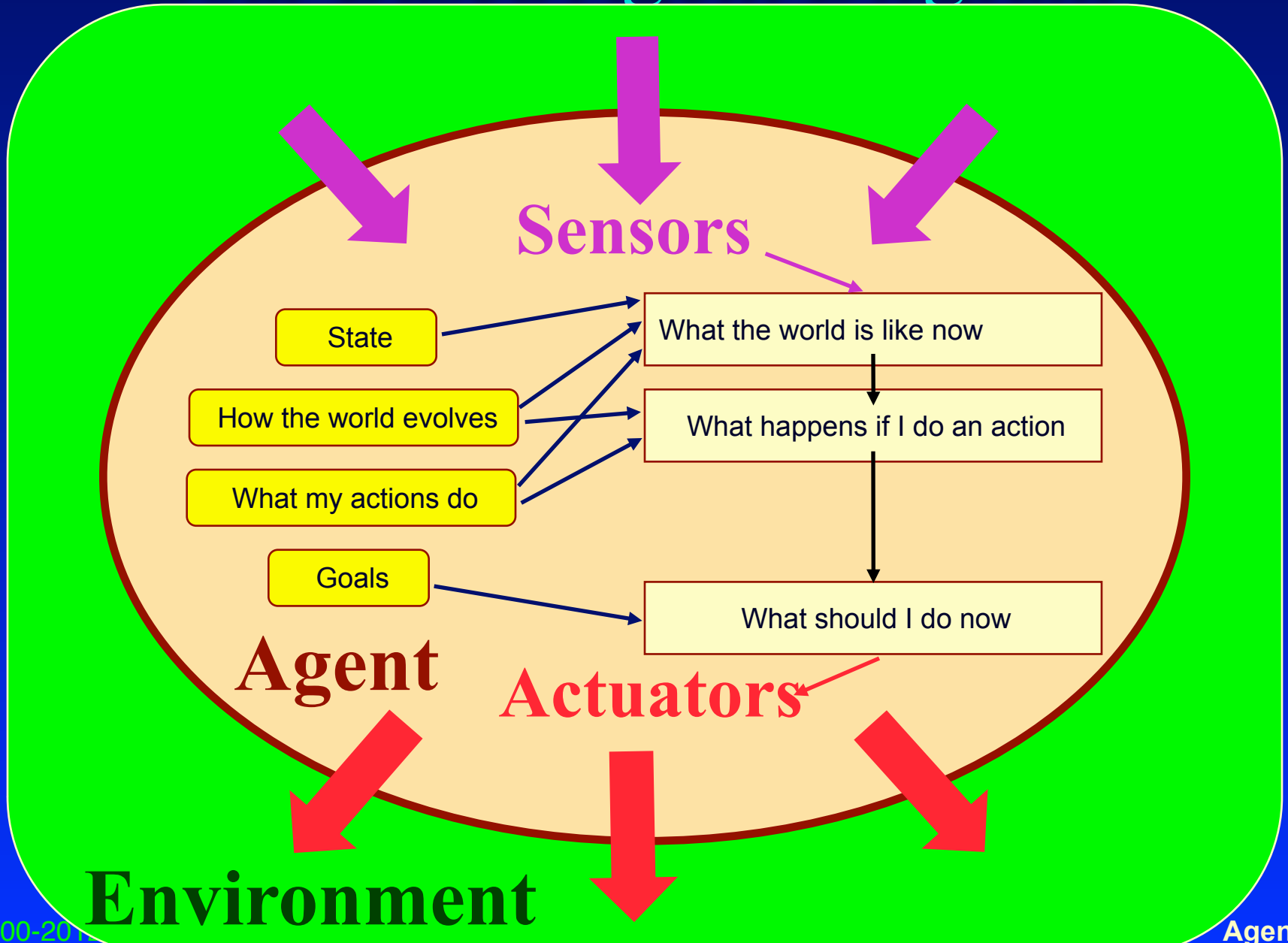
© 2000-2012 Franz Kurfess

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

© 2000-2012 Franz Kurfess

Goal-Based Agent Diagram



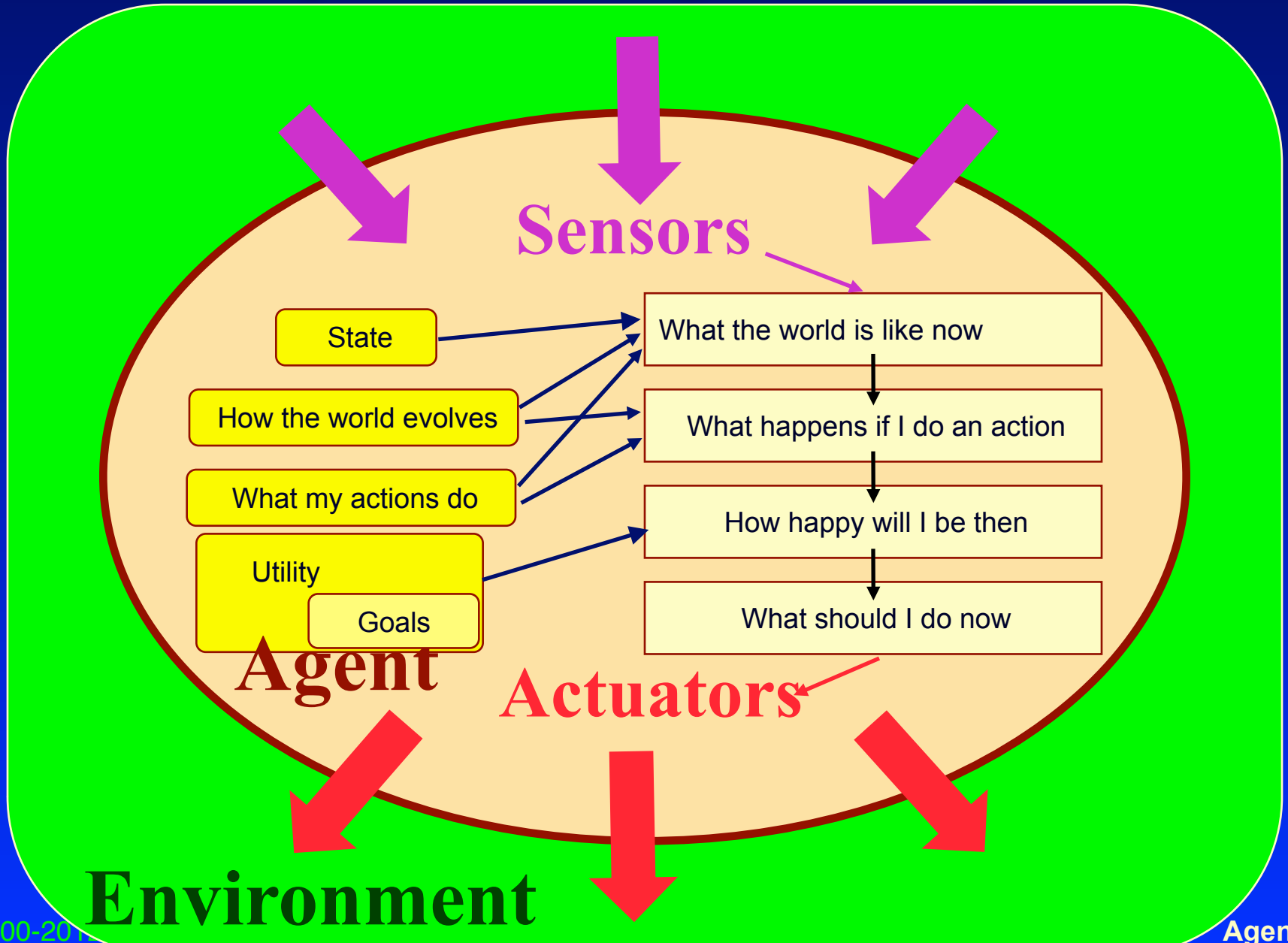
© 2000-201

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

© 2000-2012 Franz Kurfess

Utility-Based Agent Diagram



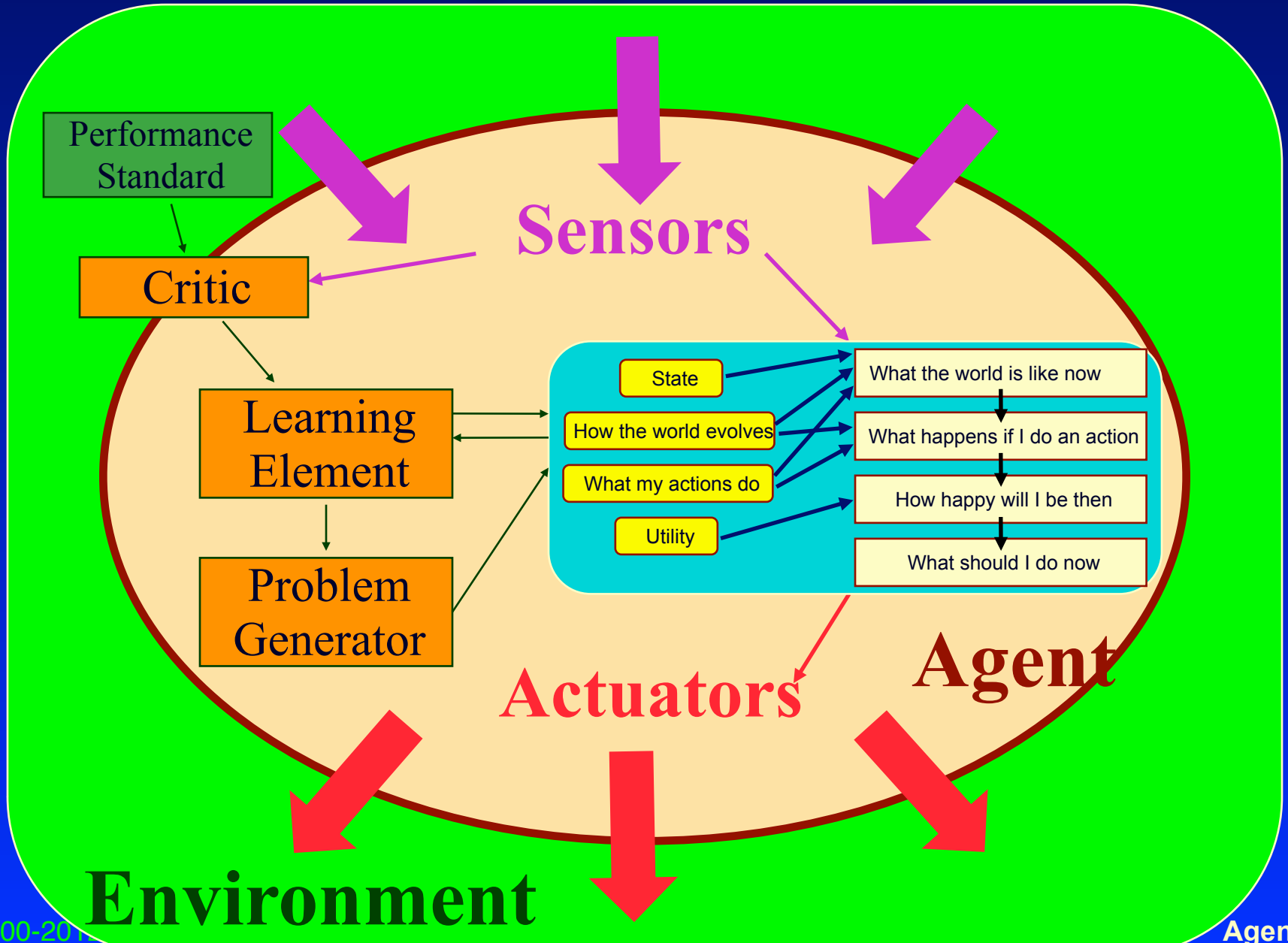
© 2000-201

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

© 2000-2012 Franz Kurfess

Learning Agent Diagram



© 2000-201

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
- sticastuc environment
- utility

© 2000-2012 Franz Kurfess

Chapter Summary

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

© 2000-2012 Franz Kurfess

© 2000-2012 Franz Kurfess