Artificial Intelligence
Unit 1-2 Intelligent Agents

BE CSE VII semester

Engels. R

History of AI

- Why AI is different from Control Theory, Operations Research, and Decision theory?
  - Do all have similar goals?
- Duplicating human faculties such as
  - creativity,
  - self-improvement,
  - language use
- Methodology
  - Branch of Computer Science (use of computer theory is mandatory)
  - attempts to build machines that will function autonomously in complex, changing environments
- Self study – 1.3.6 – 1.3.10

State of the art

- Robotic vehicles –Ex: automatic driving cars
- Speech recognition – UA flight booking agent
- Autonomous planning and scheduling
  - NASA Remote Agent program - first on-board autonomous planning program
- Game playing – IBM Deep Blue vs Kasporov, Gary
  - Kasparov said that he felt a “***new kind of intelligence***” across the board from him
  - Other human players used this game to study GK’s weaknesses
- Spam fighting – Close to 90% accuracy
- Logistics planning– Persian Gulf crisis (100K Units)
- Robotics- Roomba automatic vacuum cleaners
  - PackBot – Military use – Bomb Disposal, Hazmat, search and reconnaissance
- Machine Translation– Good results – Designers w/o language knowledge

What is AI for you?

- This course – **Rational Action**
- Are you concerned with thinking or behaviour?
- Do you want to model humans or work from an ideal standard?
- What would you think as a great tool with AI capabilities?
- Are these intelligent?
  - Supermarket bar code scanners.
  - Web search engines.
  - Voice-activated telephone menus.
  - Routing algorithms responding dynamically to state of n/w
- Why would evolution tend to result in systems that act rationally?
  - What goals are such systems designed to achieve?
- Self study 1.5 – Summary

Agents

- Perceives **environment** through **sensors**
- Acts upon that environment through **actuators**
- Example: A human Agent
  - Sensors: eyes, ears, and other organs
  - Actuators: hands, legs, vocal tract, etc.
- Example: A robotic agent
  - Sensors: Cameras, infrared range finders
  - Actuators: Various motors
- Example: A software agent
  - Sensory input: receives keystrokes, file contents, and network packets
  - Sensors: keyboard, network cable, etc
  - Sensory output: writing files, and sending network packets.
  - Actuators: screen, disc accessing hardware/software, network cable, etc.

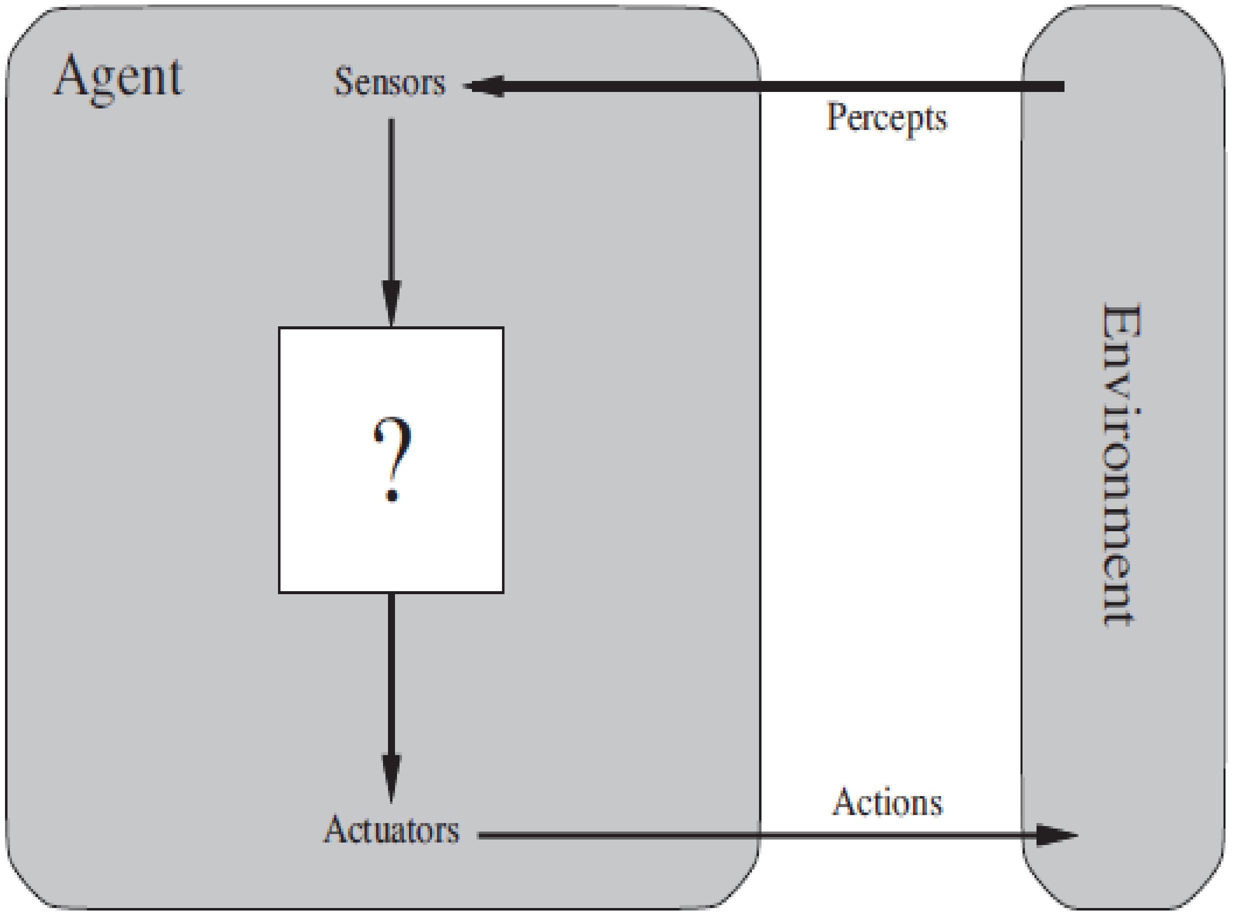
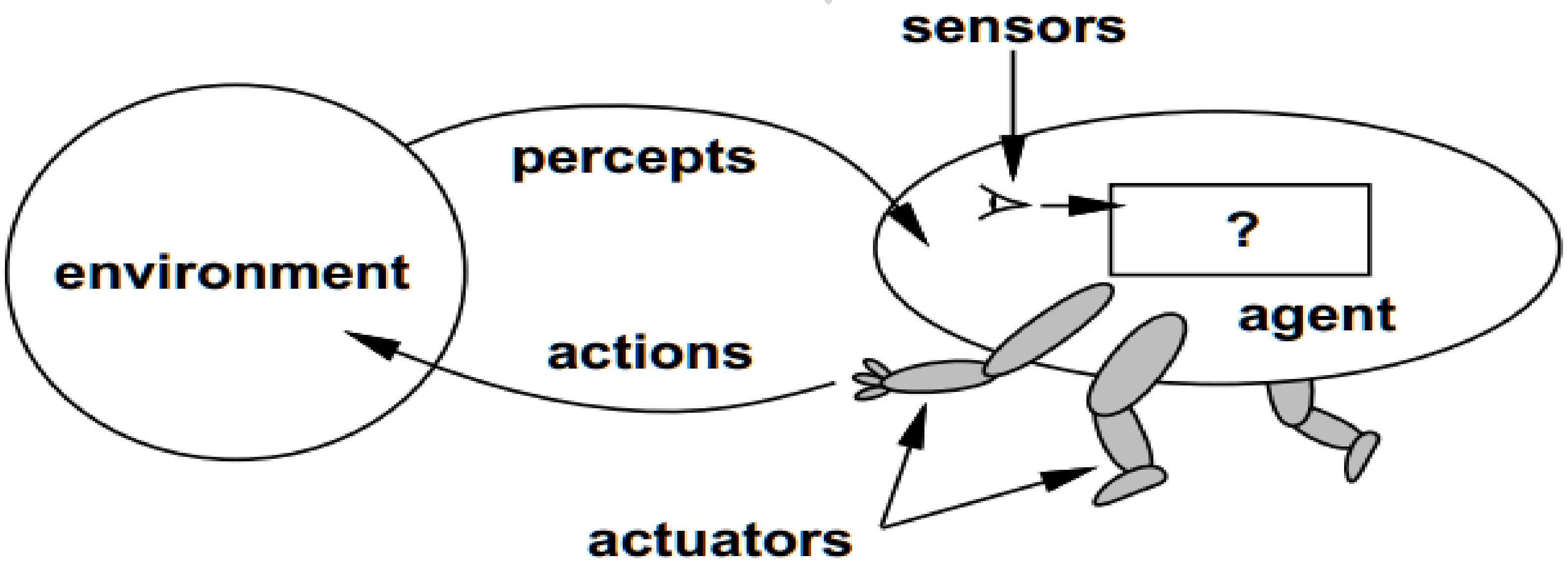


Figure 2.1 Agents interact with environments through sensors and actuators.

Percept and Agent Function

- Percept :
  - agent's perceptual inputs at any given instant.
- Percept sequence:
  - Complete history of everything the agent has ever perceived.
- Agent's choice of action at any given instant
  - can depend on the entire percept sequence observed to date
  - Not on anything it hasn't perceived.
- Agent Behavior / Agent Function
  - Described by the agent function that maps any given percept sequence to an action.
  - Implemented by an agent program

Agents and Environments



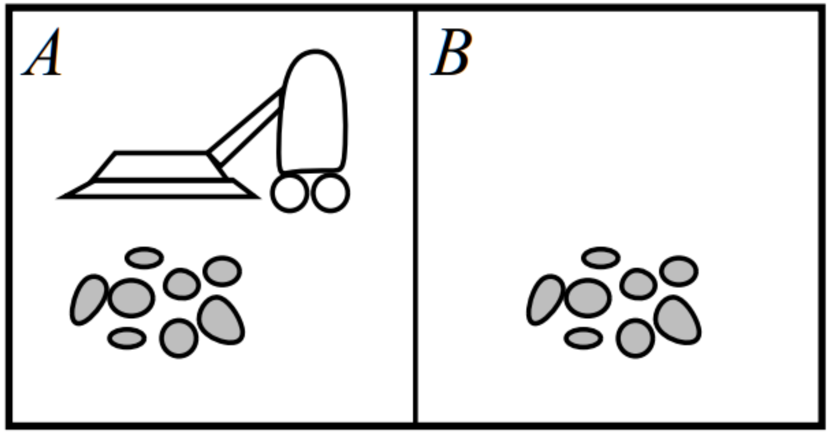
Agents include humans, robots, softbots, thermostats, etc.

The agent function maps from percept histories to actions:

$$f : \mathcal{P}^* \rightarrow \mathcal{A}$$

The agent program runs on the physical architecture to produce f

Vacuum Cleaner World



Percepts: location and contents, e.g., $[A, Dirty]$

Actions: $Left$, $Right$, $Suck$, $NoOp$

Challenge: How to fill the right hand columns of the table?

Percept sequence	Action
$[A, Clean]$	$Right$
$[A, Dirty]$	$Suck$
$[B, Clean]$	$Left$
$[B, Dirty]$	$Suck$
$[A, Clean], [A, Clean]$	$Right$
$[A, Clean], [A, Dirty]$	$Suck$
$\vdots$	$\vdots$
$[A, Clean], [A, Clean], [A, Clean]$	$Right$
$[A, Clean], [A, Clean], [A, Dirty]$	$Suck$
$\vdots$	$\vdots$

Rational Agent

- A **rational agent** is one that does the right thing
- **General Rule**: Better to design performance measures
- according to what one actually wants in the environment and
- **not** according to how one thinks the agent should behave

Rational Agent Definition: For each possible percept sequence, a **rational agent** should select an **action that is expected to maximize its performance measure**, given the **evidence** provided by the percept sequence and the **built-in knowledge** of the agent

Performance measure

- Rational NOT **Omniscience**
- Rationality – deals with **expected** (not actual!) maximization of performance
  - ***So not always successful***
- **Information gathering**
  - required to make future decisions better
- **Explorations** (with no/little prior information)
  - cannot be avoided in new environments
- **Autonomy**: Can the agent break away from the initial designed/specified percept-action (prior knowledge)
  - Initial minimal knowledge
  - Gradual increase of autonomy (by gathering knowledge)

Rationality depends on
1. performance measure (success criterion)
2. Prior knowledge
3. Performable actions
4. Percept sequence

PPPP MKAS

PEAS

- Task environment
- **PEAS**
  - **P**erformance
  - **E**nvironment
  - **A**ctuators
  - **S**ensors
- Example: Automated taxi driver (define PEAS)

Home work: What would be PEAS for an internet shopping agent?

Agent Type	Performance Measure	Environment	Actuators	Sensors
Taxi driver	Safe, fast, legal, comfortable trip, maximize profits	Roads, other traffic, pedestrians, customers	Steering, accelerator, brake, signal, horn, display	Cameras, sonar, speedometer, GPS, odometer, accelerometer, engine sensors, keyboard

PEAS - Examples

Agent Type	PM	Environment	Actuators	Sensors
Medical diagnosis system				
Satellite image analysis system				
Part-picking robot				
Refinery Controller				
Interactive English teacher				

PEAS - Examples

Agent Type	PM	Environment	Actuators	Sensors
Medical diagnosis system	Healthy patient, reduced costs	Patient, Hospital, Staff	Display of questions, tests, diagnosis, treatments, referrals	Entry of symptoms, findings, patient answers
Satellite image analysis system	Correct image categorization	Downlink from Satellite	Display of image categorization	Colour pixel arrays
Part-picking robot	Percentage of parts in correct bins	Conveyer belt with parts; bins	Jointed arm and hand	Camera, joint angle sensors
Refinery Controller	Purity, yield, safety	Refinery, operators	Valves, pumps, heaters, displays	Temperature, pressure, chemical sensors
Interactive English teacher	Student's score on tests	Set of students, testing agency	Display of exercises, suggestions, corrections	Keyboard entry

Properties of Task Environments

- Observable
  - Fully observable / partially observable?
- Agents
  - Single agent / Multi agent?
- Next state
  - Deterministic / Stochastic?
- Memory
  - Episodic / Sequential?

Properties of Task Environments

- Environment Change
  - Static / Dynamic?
- State and time
  - Discrete / Continuous?
- Prior Knowledge
  - Completely known / Unknown?

Examples of task environments

Task Environment	Observable	Agents	Deterministic	Episodic	Static	Discrete
Crossword puzzle	Fully	Single	Deterministic	Sequential	Static	Discrete
Chess with a clock	Fully	Multi	Deterministic	Sequential	Semi	Discrete
Poker	Partially	Multi	Stochastic	Sequential	Static	Discrete
Backgammon	Fully	Multi	Stochastic	Sequential	Static	Discrete
Taxi driving	Partially	Multi	Stochastic	Sequential	Dynamic	Continuous
Medical diagnosis	Partially	Single	Stochastic	Sequential	Dynamic	Continuous
Image analysis	Fully	Single	Deterministic	Episodic	Semi	Continuous
Part-picking robot	Partially	Single	Stochastic	Episodic	Dynamic	Continuous
Refinery controller	Partially	Single	Stochastic	Sequential	Dynamic	Continuous
Interactive English tutor	Partially	Multi	Stochastic	Sequential	Dynamic	Discrete

- **Home work:**
  - AIMA book Chapter 1 : 1.10 – 1.14
  - AIMA book Chapter 2: 2.3, 2.4, 2.5, 2.6, 2.7

Structure of Agents

- Agents include
  - Physical Architecture (actuators, sensors)
  - Agent program (implementing agent function)
  - Agent programs have only current percepts
    - If they need sequence, they have to remember
- Types
  - Table driven agent (impossible to implement, but will provide good performance)
  - Simplex
  - Model-based
  - Goal-based
  - Utility-based
- Each kind combines particular components in particular ways to generate actions

Table Driven Agent

- Construct a table
  - Containing the appropriate action for every possible percept sequence
  - Simple look up for action
- Daunting size of these tables
  - For chess we need 10^{150} percept sequence mapping
  - Number of atoms in universe < 10^{80}
- Issues
  - no physical agent in this universe will have the space to store the table,
  - the designer would not have time to create the table,
  - no agent could ever learn all the right table entries from its experience, and
  - no guidance about how to fill in the table entries
    - Even if the environment is simple enough to yield a feasible table size

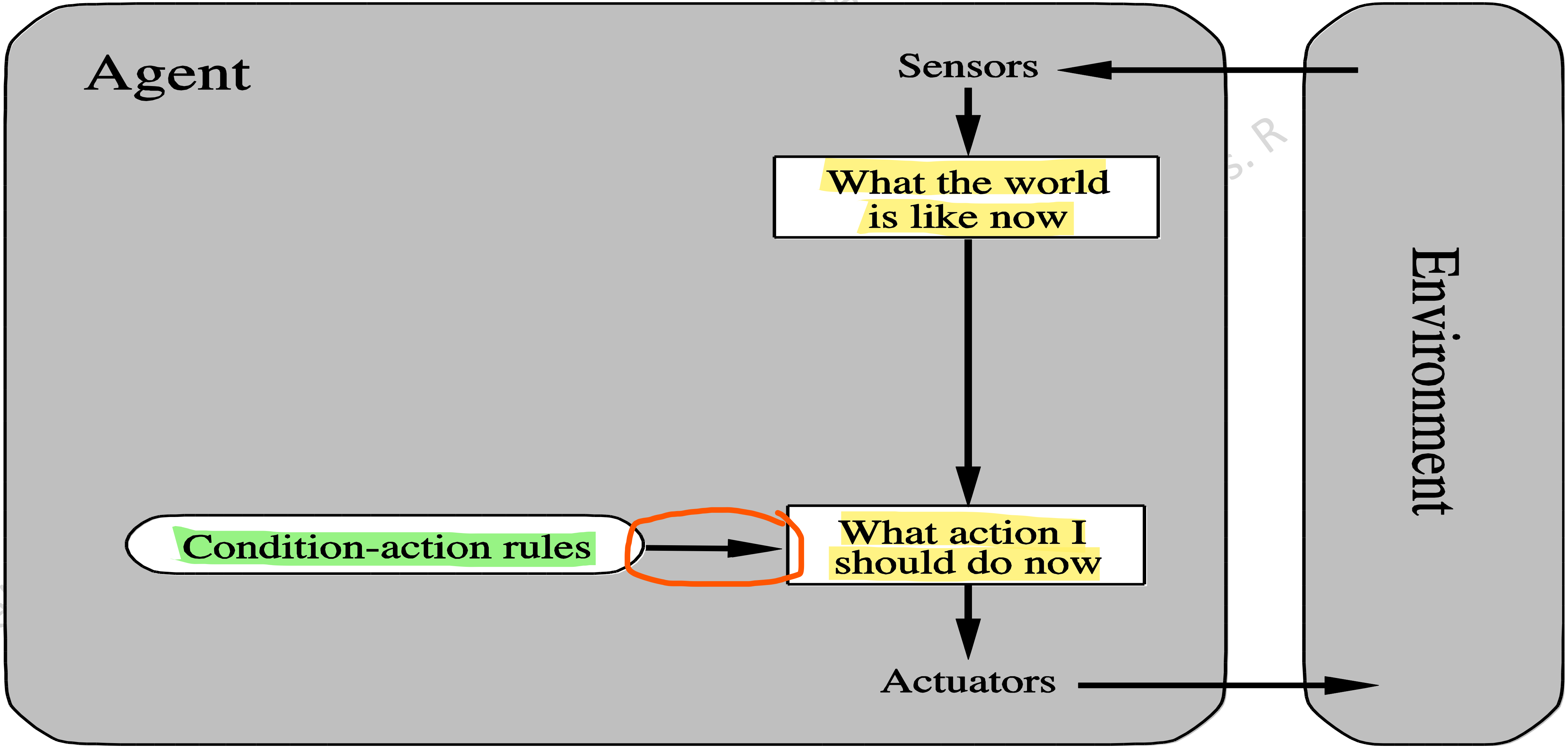
Simplex Agent

- Simplest type
- Considers only current percept, Does not have percept history
- Used where condition-action rule is present
  - If car in front is braking, apply brake
- Limited intelligence (Only based on current percept)
  - randomized simple reflex agent might outperform a deterministic simple reflex agent

function **SIMPLE-REFLEX-AGENT**(percept) returns an action
persistent, rates, a set of condition—action rules

state ← INTERPRET-INPUT(percept)
rule ← RULE-MATCH(state, rule)
action ← rule.ACTION
return action

Simplex Reflex Agent



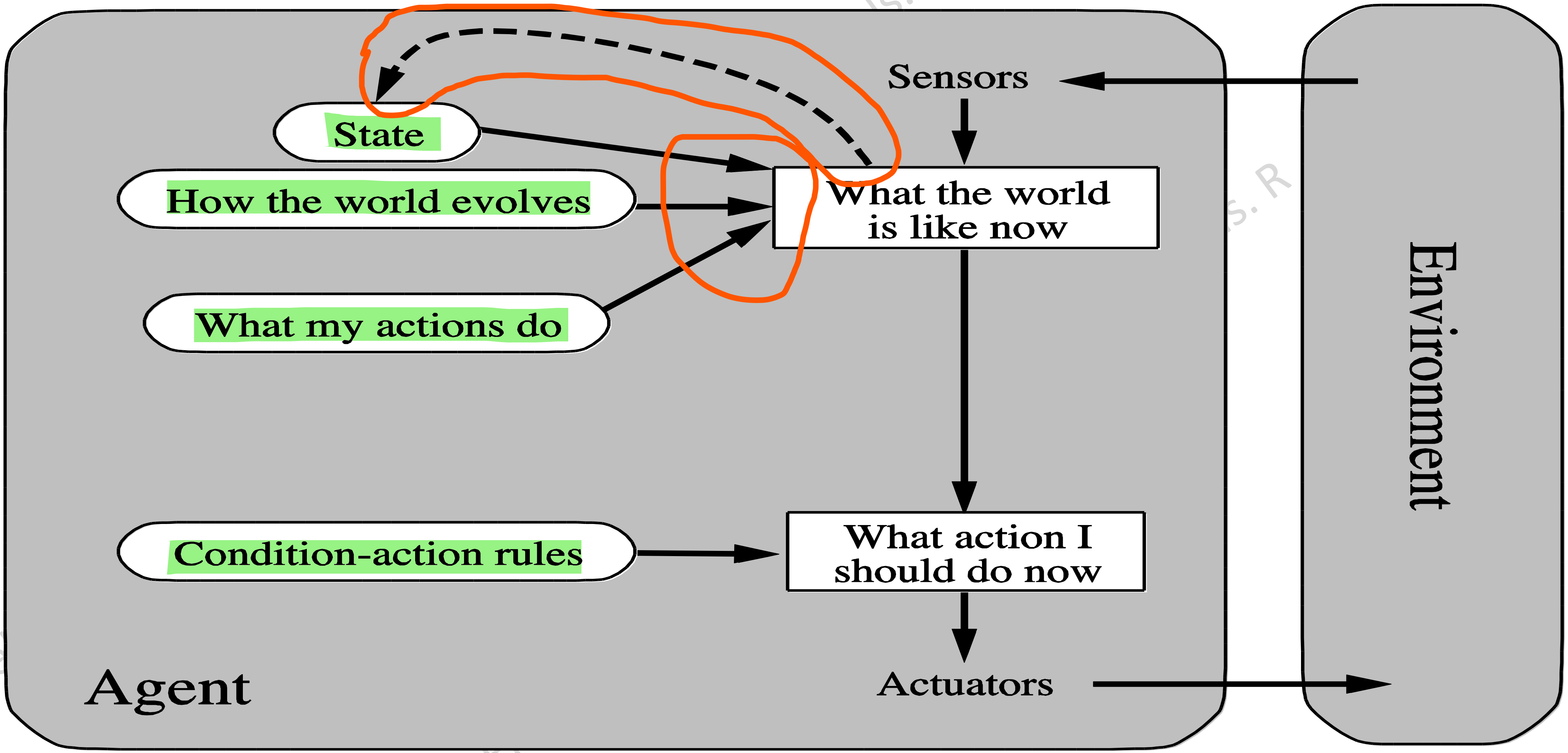
Model Based Agents

- Internal state to track of percept sequence
- Need to know how world works (Model)
  - How world evolves
  - Impact of own actions on world

function MODEL-BASED-REFLEX-AGENT(percept) returns an action
persistent **state**, the agent's current conception of the world state
model, a description of how the next state depends on current state and action
rules, a set of condition—action rules
action, the most recent action, initially none

state ← UPDATE-STATE(state, action, percept, model)
rule. ← RULE MATCH(state, rules)
action ← rule.ACTION
return action

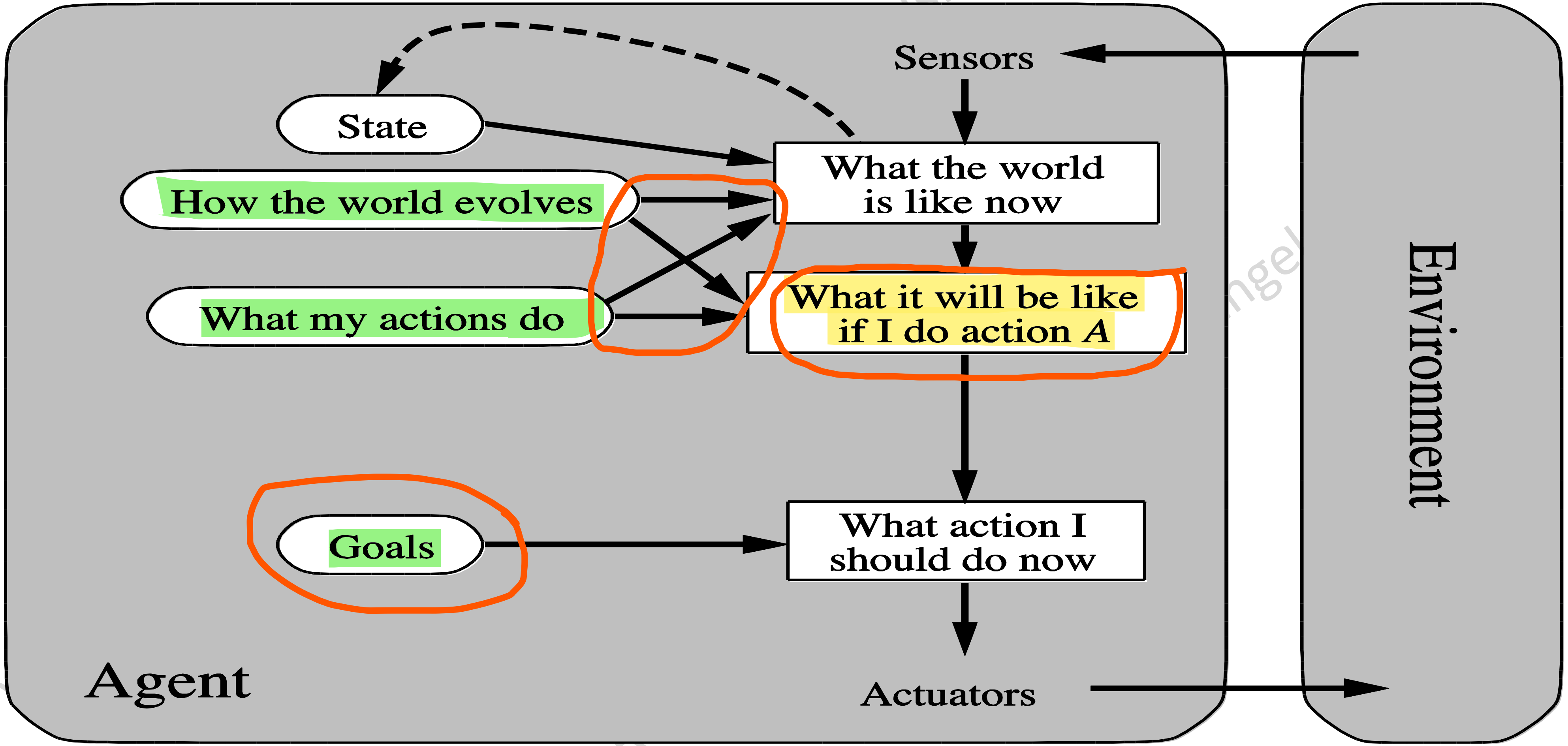
Model-based agent



Goal based agents

- Choose actions that would achieve goals
- Not simple if then else
- Think about future (what would happen if …)
- Slower, but more flexible
  - – knowledge that supports its decisions is represented explicitly and can be modified

Goal-based, model-based agent

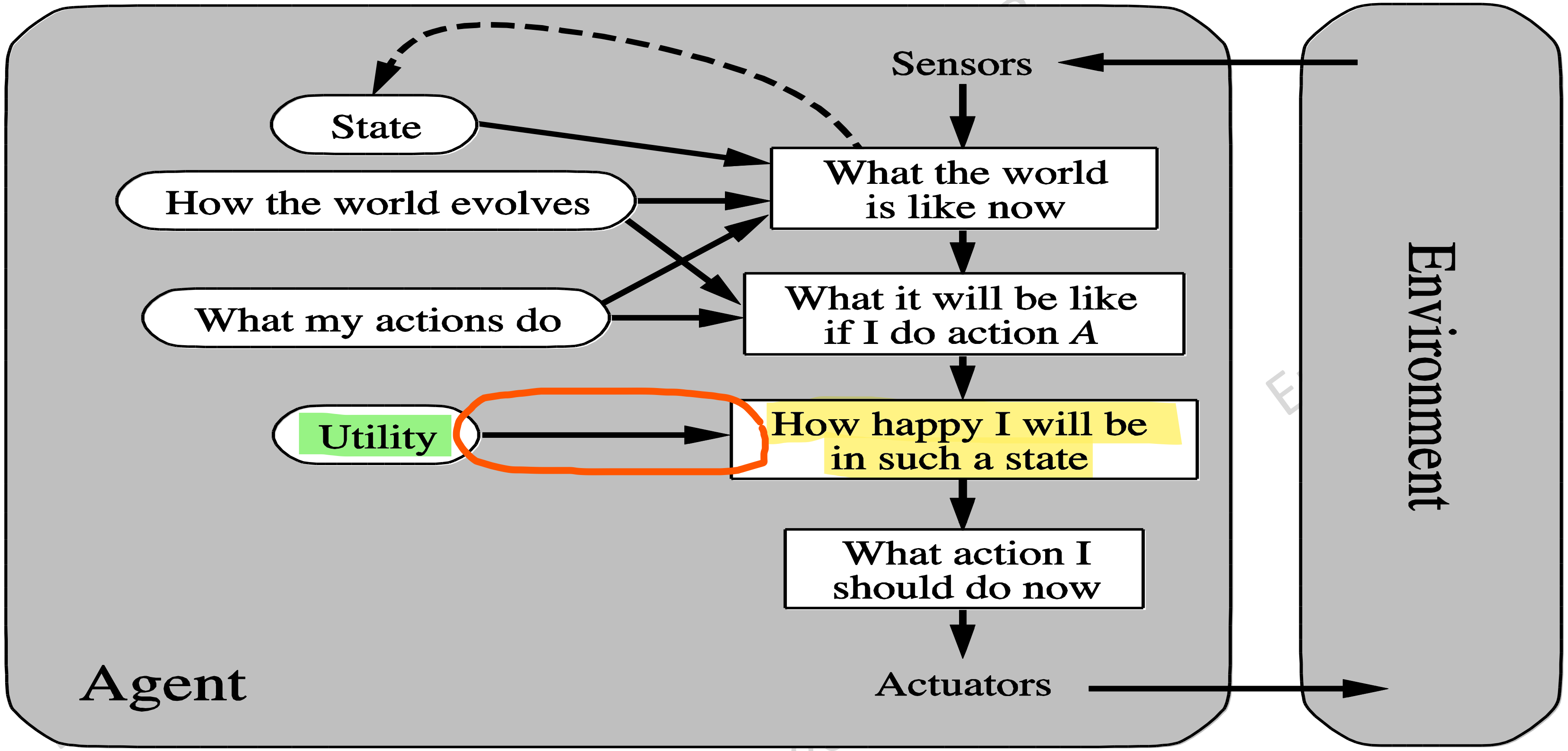


- **A model-based, goal-based agent**
  - keeps track of the world state as well as a set of goals it is trying to achieve, and
  - chooses an action that will (eventually) lead to the achievement of its goals

Utility-based agents

- Goal based agents are binary (ok, not ok)
- Faster, better, reliable actions to reach goals?
- Compare different world states and pick best
  - Utility function – internalization of Performance Measure
  - If internal utility function and external performance measure are in agreement, agent chooses actions to maximize its utility (rational!)

Model-based Utility-based agent



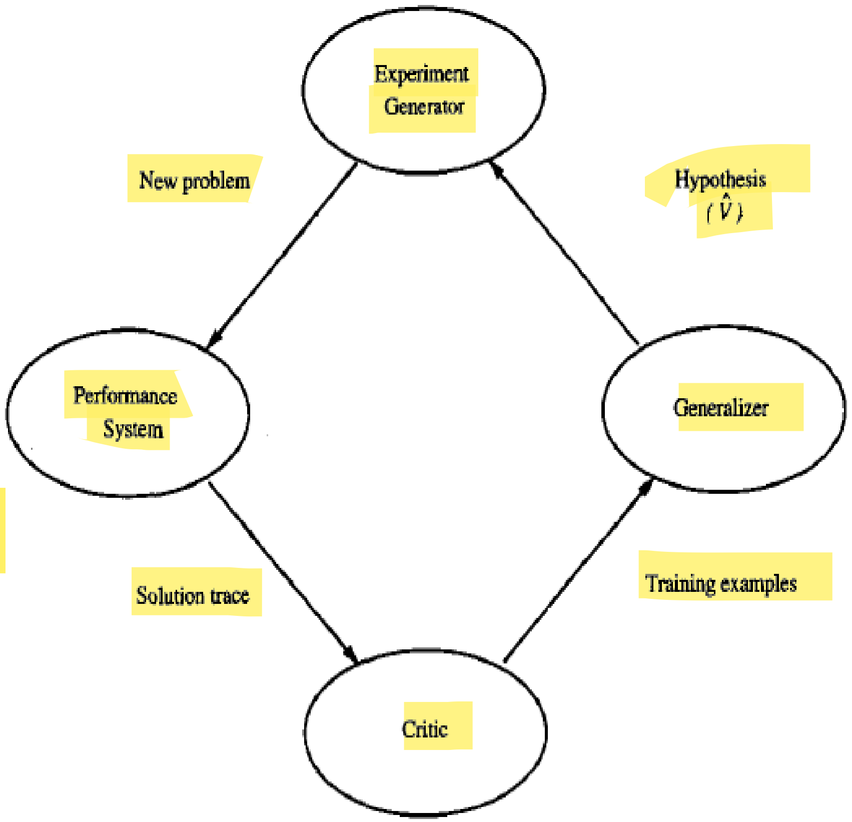
- **A model-based, utility-based agent**
  - uses a model of the world,
  - along with a utility function that measures its preferences among states of the world
- Chooses the action that leads to the best expected utility
  - expected utility is ***computed by averaging over all possible outcome states,*** weighted by the probability of the outcome

Learning agent

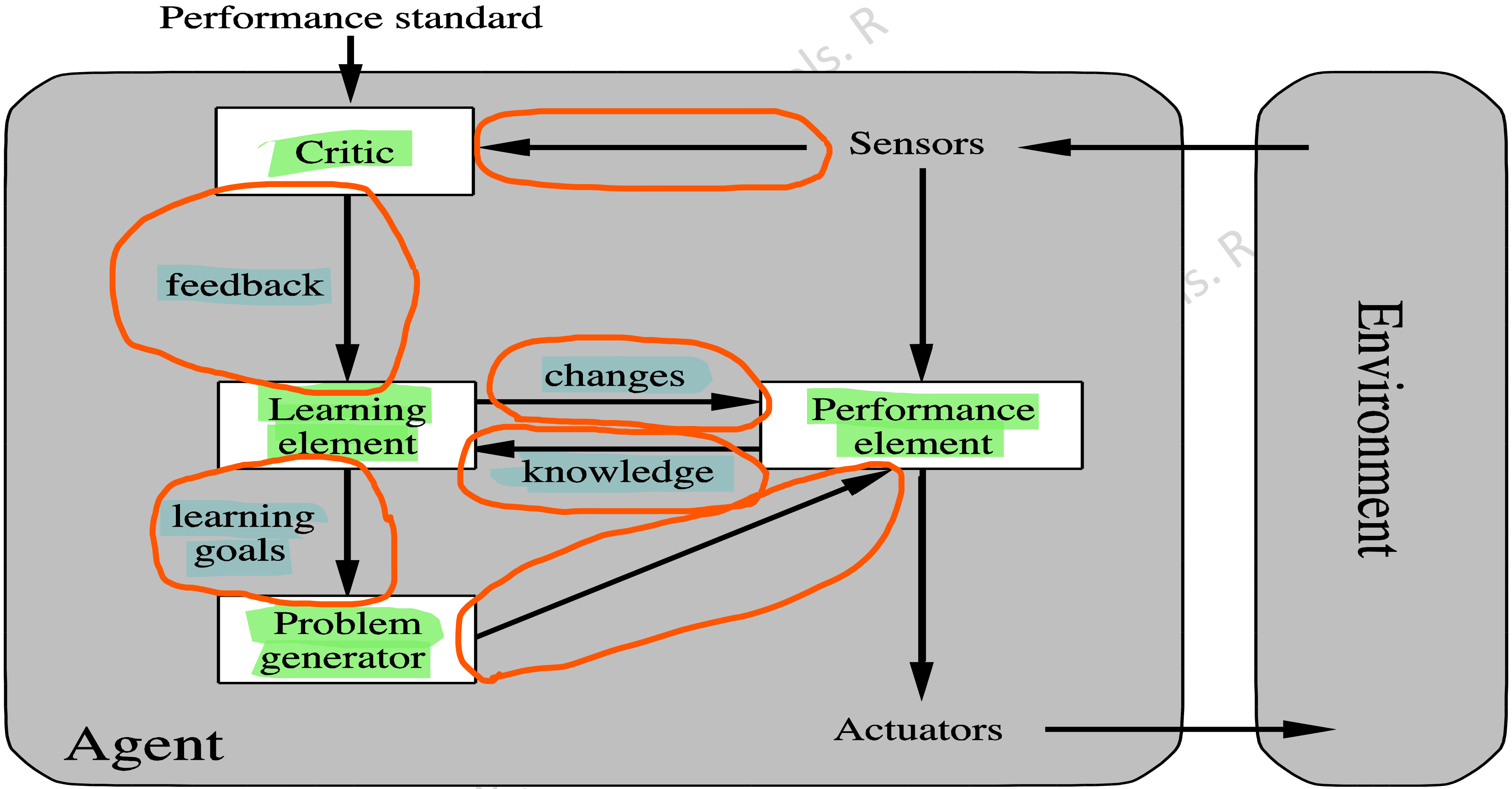
- What is learning in an agent?
  - A process of modification of each component of the agent
  - to bring the components into closer agreement with the available feedback information
  - thereby improv-ing the overall performance of the agent
- Building learning machines and teaching them
  - Allows the agent to operate in initially unknown environments
  - Enables to become more competent than its initial knowledge alone might allow

Learning agent – conceptual Components

- **Problem generator**: Exploratory problems are created for LE
  - Suggests actions that will lead to new and informative experiences
- **Performance Element**: Makes external actions
  - Takes in percepts and selects actions
- **Critic**: Tells LE how PE is doing with respect to a **fixed** performance standard
- **Learning Element** : Learns improvements
  - uses feedback from the **critic** on current performance and determines how to change performance element to do better



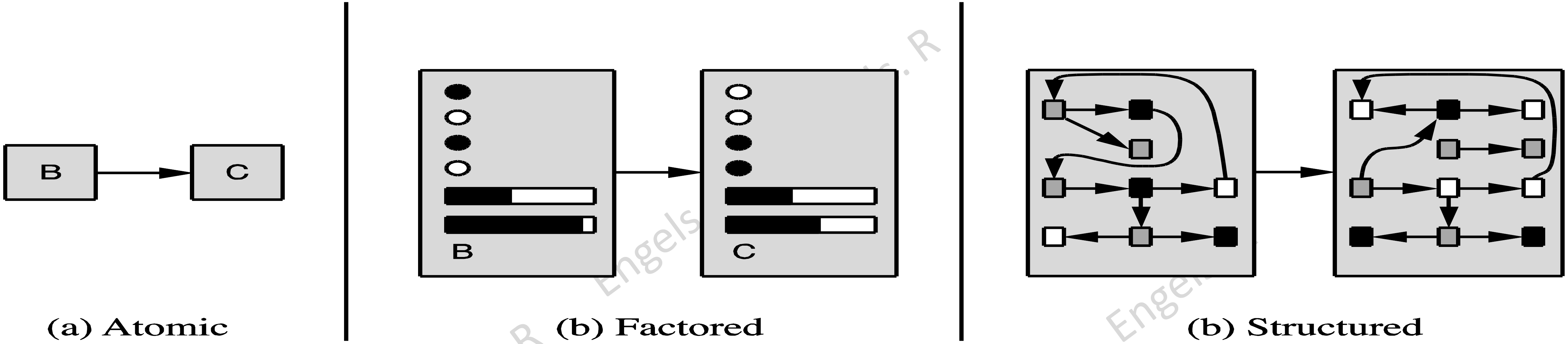
Learning agent



Learning agent for automated taxi

- **Performance Element**
  - consists of whatever **collection of knowledge and procedures the taxi has for selecting its driving actions**
  - Ex: Taxi goes out on the road and drives, using this performance element
- **Critic**
  - **observes the world and passes information along to the learning element**
  - Ex: after the taxi makes a quick left turn across three lanes of traffic, the critic observes the shocking language used by other drivers
- **Learning Element**
  - is able to **formulate a rule saying this (quick left turn) was a bad action** and
  - the **performance element is modified** by installation of the new rule
- **Problem Generator**
  - might identify certain areas of behaviour in need of improvement and
  - **suggest experiments**
  - such as trying out the brakes on different road surfaces under different conditions

Representation of Environments



- Atomic states
  - Each state is indivisible (no internal structure). Ex: search, game playing, Markovian
- Factored states
  - Splits up each state into a fixed set of attributes, with different types of values.
  - Ex: constraint satisfaction algorithms, propositional logic, planning, Machine Learning
- Structured
  - Real world entities are represented as objects with relationships, Ex: Knowledge based learning, FO Probability models, NL understanding

Representation of Environments

- Axis of increasing **expressiveness**
  - **Structured-Representation > Factored-Representation > Atomic**
- More expressive representation can capture
  - everything a less expressive one can capture + plus some more
- Ex: rules of chess

Often, the more expressive language is much more concise

  - Written in 1-2 pages in a structured-representation language (say FOL)
  - Thousands of pages in factored-representation language such as propositional logic
- Reasoning and learning become more complex
  - as the expressive power of the representation increases

2.6 Question

a. Can there be more than one agent program that implements a given agent function? Give an example, or show why one is not possible

b. Are there agent functions that cannot be implemented by any agent program?

c. Given a fixed machine architecture, does each agent program implement exactly one agent function?

d. Given an architecture with n bits of storage, how many different possible agent programs are there?

e. Suppose we keep the agent program fixed but speed up the machine by a factor of two. Does that change the agent function?

2.6 Question

a. Can there be more than one agent program that implements a given agent function? Give an example, or show why one is not possible

- Yes

b. Are there agent functions that cannot be implemented by any agent program?

- Yes. Ex: Agent function always selecting the best solution

c. Given a fixed machine architecture, does each agent program implement exactly one agent function?

- Yes, agent function is fixed for same architecture and program

d. Given an architecture with n bits of storage, how many different possible agent programs are there?

- 2^n

e. Suppose we keep the agent program fixed but speed up the machine by a factor of two. Does that change the agent function?

- If environment is dynamic, agent maybe able to give solution faster, which may be a better solution based on the computation performed

Summary

- Agent
- Agent function
- Agent program
- Rational agent
- Task environment
  - **P**erformance measure, **E**nvironment, **A**ctuators, **S**ensors (PEAS)
- Types of agents
  - TDA, Simplex reflex, model-based, goal-based, utility based, learning agent

References

- AIMA
  - Artificial Intelligence - A Modern Approach 3rd Edition - RUSSELL & NORVIG