INTELLIGENT AGENTS

Dr. Ayesha Hakim

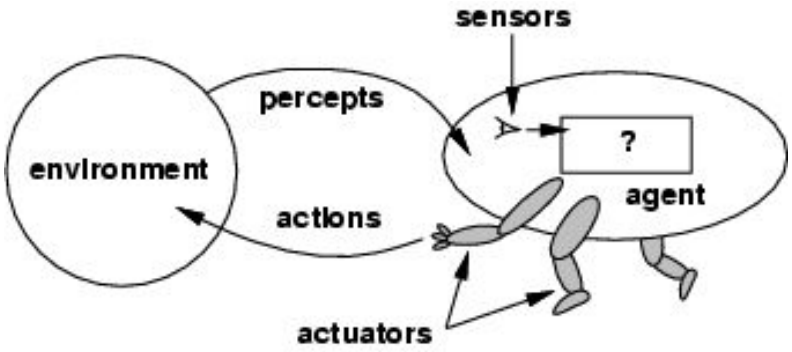
AGENTS

- An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators
- E.g Human Being

SOME TERMS

- Percept
- Percept history
- Environment
- Actuator
- Sensors

AGENTS AND ENVIRONMENTS



- The agent function maps from percept histories to actions:[f: $\mathcal{P}^{\star} \square \mathcal{A}$]
- The agent program runs on the physical architecture to produce f

agent = architecture + program

ELEMENTS OF AI AGENTS

- Environment
- Sensors

  May also include keystrokes, file contents, and network packets, voice commands/wakeup words
- Actuators

  May also include displaying on the screen, writing files, and sending network packets
- Program logic

AGENT FUNCTION VS AGENT PROGRAM

- agent function is an abstract mathematical description;
- the agent program is a concrete implementation, running on the agent architecture.

RATIONAL AGENTS

- An agent should strive to "do the right thing", based on what it can perceive and the actions it can perform.
- The right action is the one that will cause the agent to be most successful
- Performance measure: An objective criterion for success of an agent's behavior
- E.g: performance measure of a vacuum-cleaner

RATIONAL AGENTS

- Rational Agent: 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 whatever built-in knowledge the agent has.

RATIONAL AGENTS

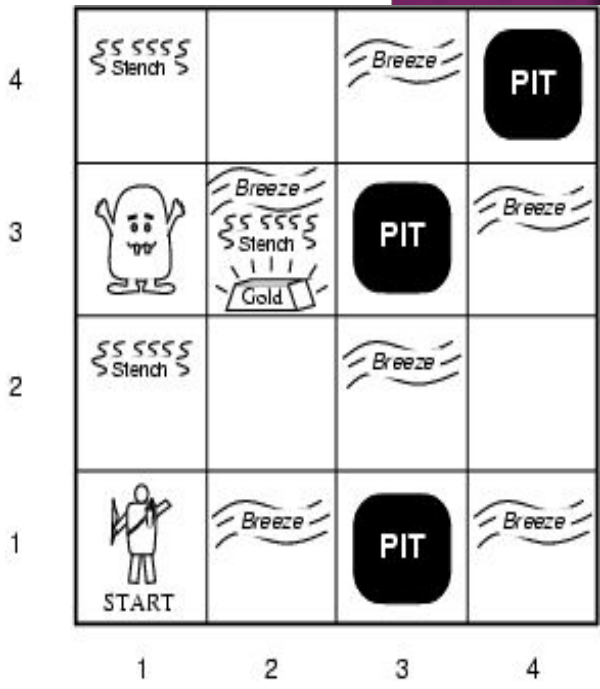
- Rationality is distinct from omniscience
- Agents can perform actions in order to modify future percepts so as to obtain useful information
- An agent is autonomous if its behavior is determined by its own experience

PEAS

- PEAS
- Must first specify the setting for intelligent agent design
- Consider, e.g., the task of designing an automated taxi driver:
  - Performance measure
    Environment
  - Actuators
  - Sensors

PEAS-WUMPUS WORLD

- Performance measure
  - gold +1000, death -1000
  - -1 per step, -10 for using the arrow
- Environment
  - 4 X 4 grid of rooms
  - Agent always starts in square [1,1], facing to right
  - Squares adjacent to wumpus are smelly
  - Squares adjacent to pit are breezy
  - Glitter iff gold is in the same square
  - Shooting kills wumpus if you are facing it
  - Shooting uses up the only arrow
  - Grabbing picks up gold if in same square
  - Releasing drops the gold in same square
- Sensors: Stench, Breeze, Glitter, Bump, Scream
- Actions: Left turn, Right turn, Forward, Grab, Release, Shoot
- Actuators : wheel/feet, arm



PEAS-AUTOMATED TAXI DRIVER

- Performance measure: Safe, fast, legal, comfortable trip, maximize profits
  - Assign measurable points
- Environment: Roads, other traffic, pedestrians, customers, signals
- Actuators: Steering wheel, accelerator, brake, signal, horn, speaker
- Sensors: Cameras, sonar, speedometer, GPS, odometer, engine sensors, keyboard, microphone

PERFORMANCE MEASURE - TAXI DRIVER AGENT

(ON SCALE OF 1 TO 100, REPREHENSIVE REWARDS AND PENALTIES)

- Legal
  - Breaking signal ▫ -40 per signal break
  - Speeding ▫ -20 per speed limit violation
  - Wrong entry ▫ -30 for every wrong entry
  - Wrong parking => -20 for every wrong parking
  - Minor accident ▫ -70 for every minor accident
  - Major accident ▫ -100 for every major accident
- Safe
- Maximize profit
- Comfortable journey
  - Avoiding pothole ▫ +40 per pothole
  - Speeding at speedbreakers ▫ -30
  - Sudden break ▫ -30

PEAS- MEDICAL DIAGNOSIS SYSTEM

- Performance measure: Healthy patient, minimize costs, lawsuits
- Environment: Patient, hospital, staff
- Actuators: Screen display (questions, tests, diagnoses, treatments, referrals)
  Sensors: Keyboard (entry of symptoms, findings, patient's answers), mouse

PEAS- PART-PICKING ROBOT

- Performance measure: Percentage of parts in correct bins, percentage of broken parts during picking, work speed/ minimum displacement of robotic arm
- Environment: Conveyor belt with parts, bins
- Actuators: Jointed arm and hand
- Sensors: Camera, joint angle sensors

KITCHEN CLEANING AGENT

- Performance measure: ?
- Environment: ?
- Actuators:?
- Sensors:?

ENVIRONMENT TYPES

- Fully observable (vs. partially observable): An agent's sensors give it access to the complete state of the environment at each point in time.
- Deterministic (vs. stochastic): The next state of the environment is completely determined by the current state and the action executed by the agent.
- Episodic (vs. sequential): The agent's experience is divided into atomic "episodes" (each episode consists of the agent perceiving and then performing a single action), and the choice of action in each episode depends only on the episode itself.

ENVIRONMENT TYPES

- Static (vs. dynamic): The environment is unchanged while an agent is deliberating.
- Discrete (vs. continuous): A limited number of distinct, clearly defined percepts and actions.
- Single agent (vs. multiagent): An agent operating by itself in an environment.

PROPERTIES OF TASK ENVIRONMENT

- Vacuum cleaning agent?
- Chess playing agent?
- Kitchen cleaning agent ?
- Interactive receptionist?
- Taxi driving agent?
- Wumpus world player?
- Pro-kabaddi referee agent?
- PUBG?

Think-Pair-Share!

AGENT FUNCTIONS AND PROGRAMS

- An agent is completely specified by the agent function mapping percept sequences to actions
- Aim: find a way to implement the rational agent function concisely

TABLE LOOKUP AGENT

Function **TABLE-DRIVEN**(percepts)

returns an action

Input: Static : percepts,

table

append percept to the end of percept sequence

action ▫ LOOKUP(percepts,table)

return action

TABLE-LOOKUP AGENT

Drawbacks:

- Huge table
- Take a long time to build the table
- no agent could ever learn all the right table entries from its experience
- even if the environment is simple enough to yield a feasible table size, the designer still has no guidance about how to fill in the table entries

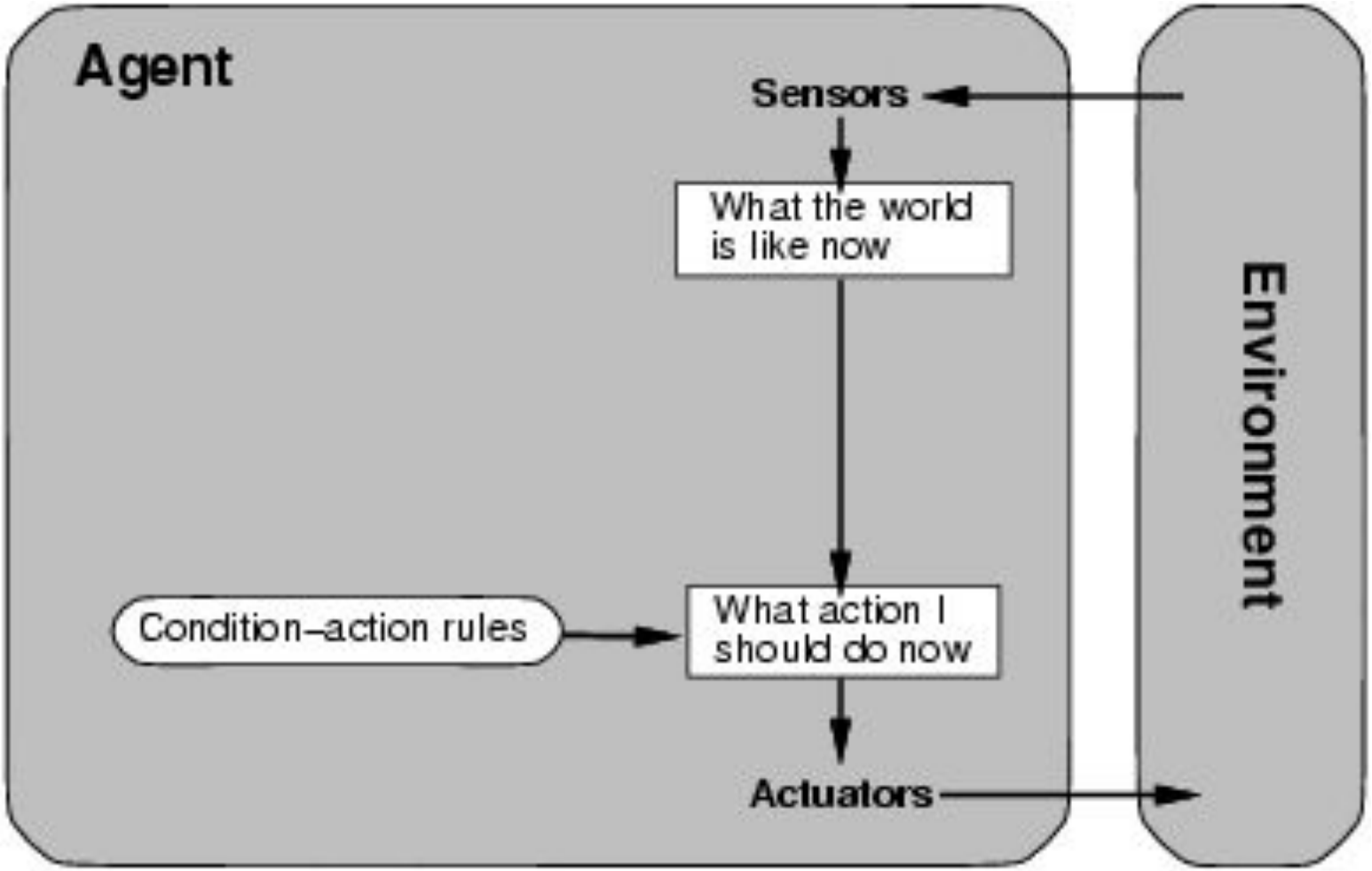
AGENT TYPES

Four basic types in order of increasing levels of intelligence and complexity:

- Simple reflex agents
- Model-based reflex agents
- Model-based Goal-based agents
- Model-based Utility-based agents

- Learning Agent- contains one of the above architectures as a component (Performance Element)

SIMPLE REFLEX AGENTS



SIMPLE REFLEX AGENTS

- Select action based on current percept
- Ignore the rest percept history
- Simple but limited intelligence.

if *car-in-front-is-braking* **then** *initiate-braking*.

AGENT PROGRAM FOR A VACUUM-CLEANER AGENT

Function VACCUM-CLEANER(loc,status) returns an action

If status = dirty then return **suck-Dirt**

else if loc= A then return **Right**

else if loc= B then return **left**

AGENT PROGRAM-SIMPLE REFLEX AGENT

Function **Simple-reflex(**percept**)** returns an action

Input: Static: rules

State ▫ INTERPRET-INPUT(percept)

Rule▫ RULE-MATCH(state,rules);

Action▫ RULE-ACTION(rule);

Return action

SIMPLE REFLEX AGENT: LIMITATIONS

- work *only if the correct decision can be made on the basis of only the current percept, i.e. only if the environment is fully observable.*
- Even a little bit of unobservability can cause serious trouble
  - E.g single image may not be sufficient to know whether the car in front is braking.
- Doesn't act on its own; would always require an external action for the agent to perform a reflex action

Table driven agent

Percept #	Percept	action
1	in A, A clean	Move to B
2	in A, A clean, In B, B dirty	suck dirt
3	in A, A clean, In B, B dirty, B clean	Move to A
4	in A, A clean, In B, B dirty, B clean, in A, A dirty	suck dirt
5	in A, A clean, In B, B dirty, B clean, in A, A dirty, A clean	Move to B

Simple reflex agent

Percept #	Percept	action
1	in A, A clean	Move to B
2	In B, B dirty	suck dirt
3	In B, B clean	Move to A
4	in A, A dirty	suck dirt
5	in A, A clean	Move to B

Cannot diffrentiate in percept 1 and 5

Takes decision on current percept

Drawback for both -

1. single percept has to be enough to take decision
2. if there are two outcomes for single input, both cannot understand it and follow same action which comes first in table.

E.g. condition - In A, A clean

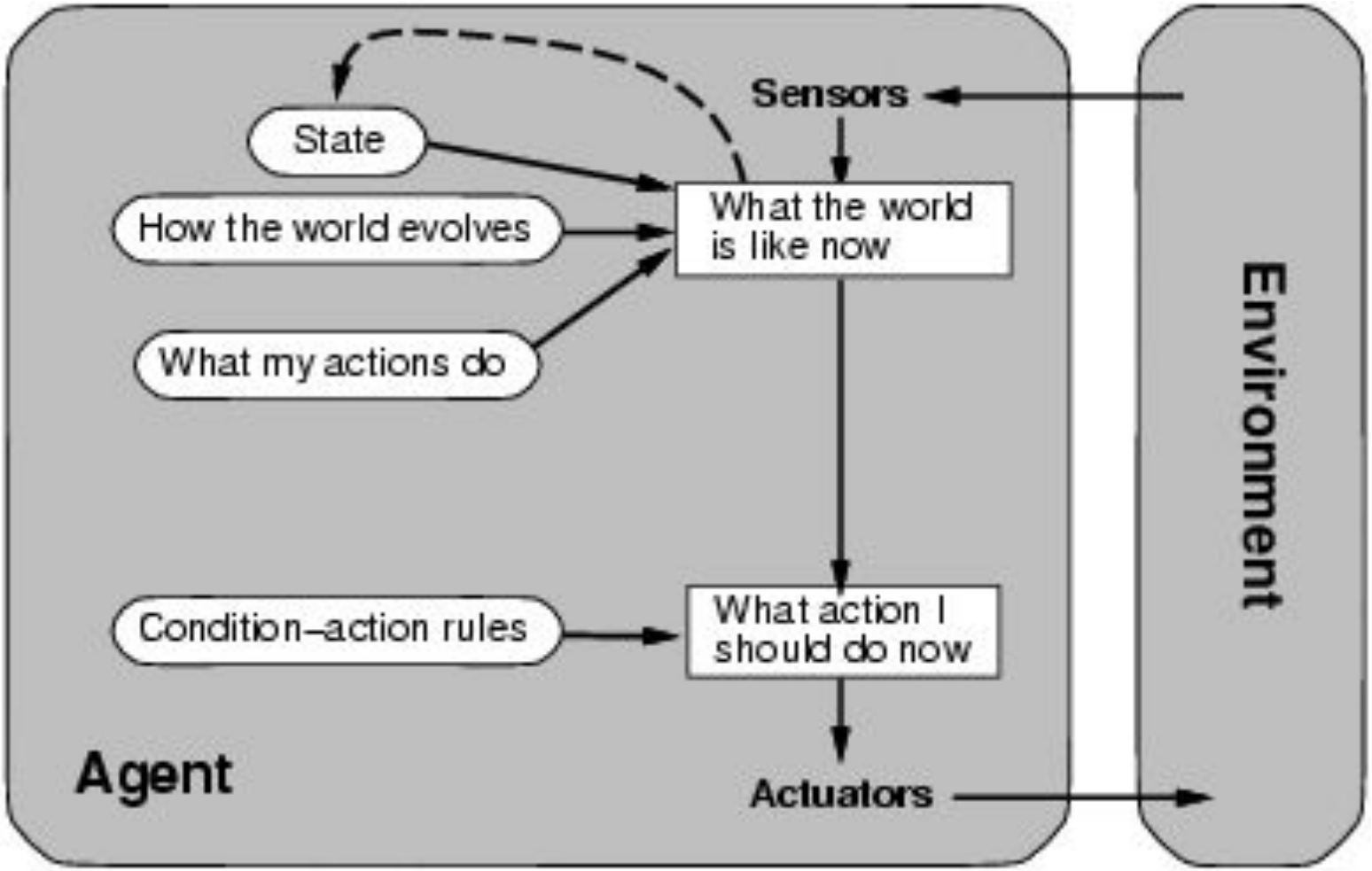
action1- move to B

action 2- no operation (just cleaned both of them)

Sr no	Percept	action
	Baby crying	Give food
	Baby crying	Play
	Baby crying	Read a story
	Baby crying	Put to sleep
	Baby crying	Sing a song

The table driven or simple reflex agent cant take different action for same percept

MODEL-BASED REFLEX AGENTS



HOW THE WORLD EVOLVES?

MODEL-BASED REFLEX AGENTS

- Keep track of the world current state.
- Knows how world evolves
- Keep track of the unseen part of world
- internal state requires two kinds of knowledge
  - Information about how the world evolves independently of the agent
  - How the agent's own actions affect the world

WHY NEED INTERNAL STATE?

- For the braking problem, the internal state is not too extensive just the previous frame from the camera, allowing the agent to detect when two red lights at the edge of the vehicle go on or off simultaneously.
- For other driving tasks such as changing lanes, the agent needs to keep track of where the other cars are if it can't see them all at once.

AGENT PROGRAM FOR MRA

Function **Model-reflex(**percept**)** returns an action

Input: state,rules, action

State ▫ UPDATE-STATE(state,action,percept)

Rule▫ RULE-MATCH(state,rules);

Action▫ RULE-ACTION(rule);

Return action

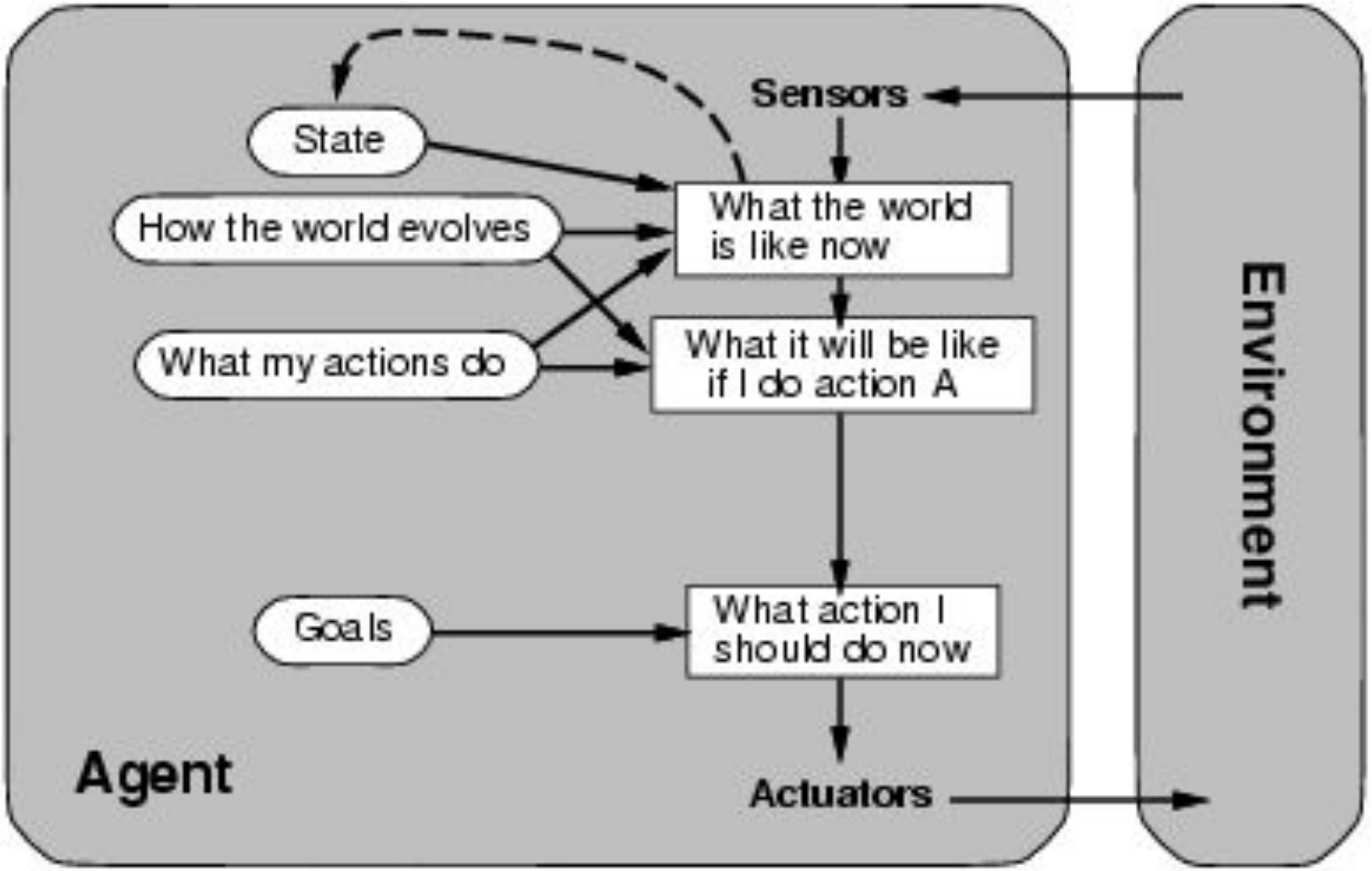
*UPDATE-STATE updates the internal state of the agent function.

MRA

Drawbacks

- No goal achievement

GOAL-BASED AGENTS



GOAL BASED AGENTS

- Agent needs some sort of **goal** information that describes situations that are desirable
  - E.g. automated taxi driver can achieve goal iff knows the passenger destination
- Combines action results with desired goals
- A goal condition may be satisfied just with one action or may take series of actions

GOAL BASED VS REFLEX BASED

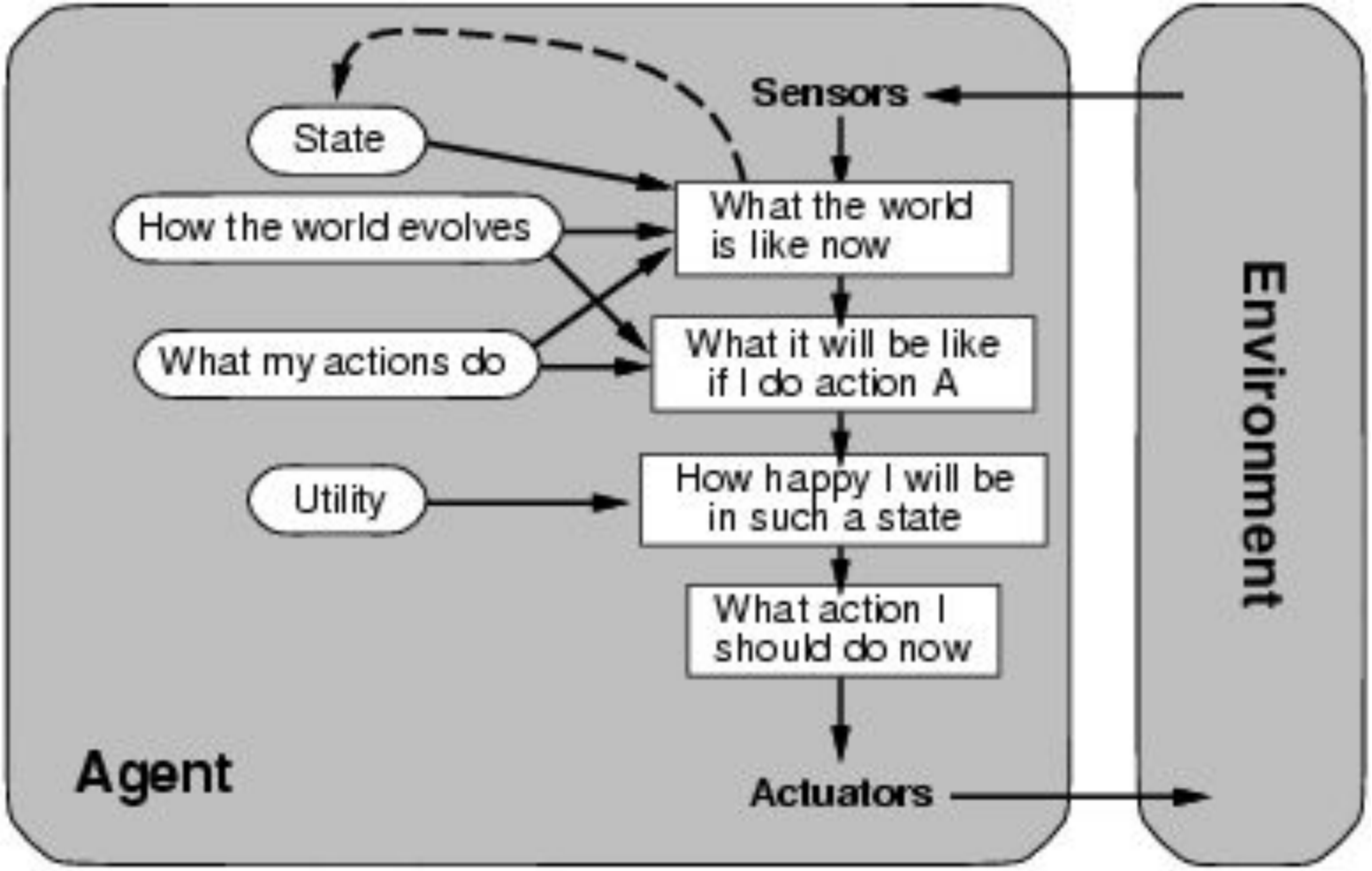
- In reflex based, built-in rules map directly from percepts to actions
  - E.g. reflex agent brakes when it sees brake lights
- goal-based agent could reason given the way the world usually evolves and takes the only action that will achieve the goal
  - it will slow down to achieve the goal of not hitting other cars instead of apply brakes
- Reflex based have to rewrite many condition-action rules to solve new problem.
- The goal-based agent's behavior can easily be changed to go to a different location.

- Efficient,
- More flexible because the knowledge that supports its decisions is represented explicitly and can be modified
  - E.g. in rainy season, the agent can update its knowledge of how effectively its brakes will operate; this will automatically cause all of the relevant behaviors to be altered to suit the new conditions

CATEGORIES OF GOAL BASED AGENTS

- Achieve goal - problem solving through searching
  - Uninformed search –BFS,DFS, DLS, IDS, Uniform Cost Search, Bidirectional search
  - Informed search-best first, A*, IDA*
  - Local search- Hill climbing, genetic algorithms simulated annealing, constraint satisfaction
  - Adversarial search – MaxMin algorithm, alpha-beta pruning
- Achieve goal through planning
  - Partial order planning
  - Total order planning

UTILITY-BASED AGENTS



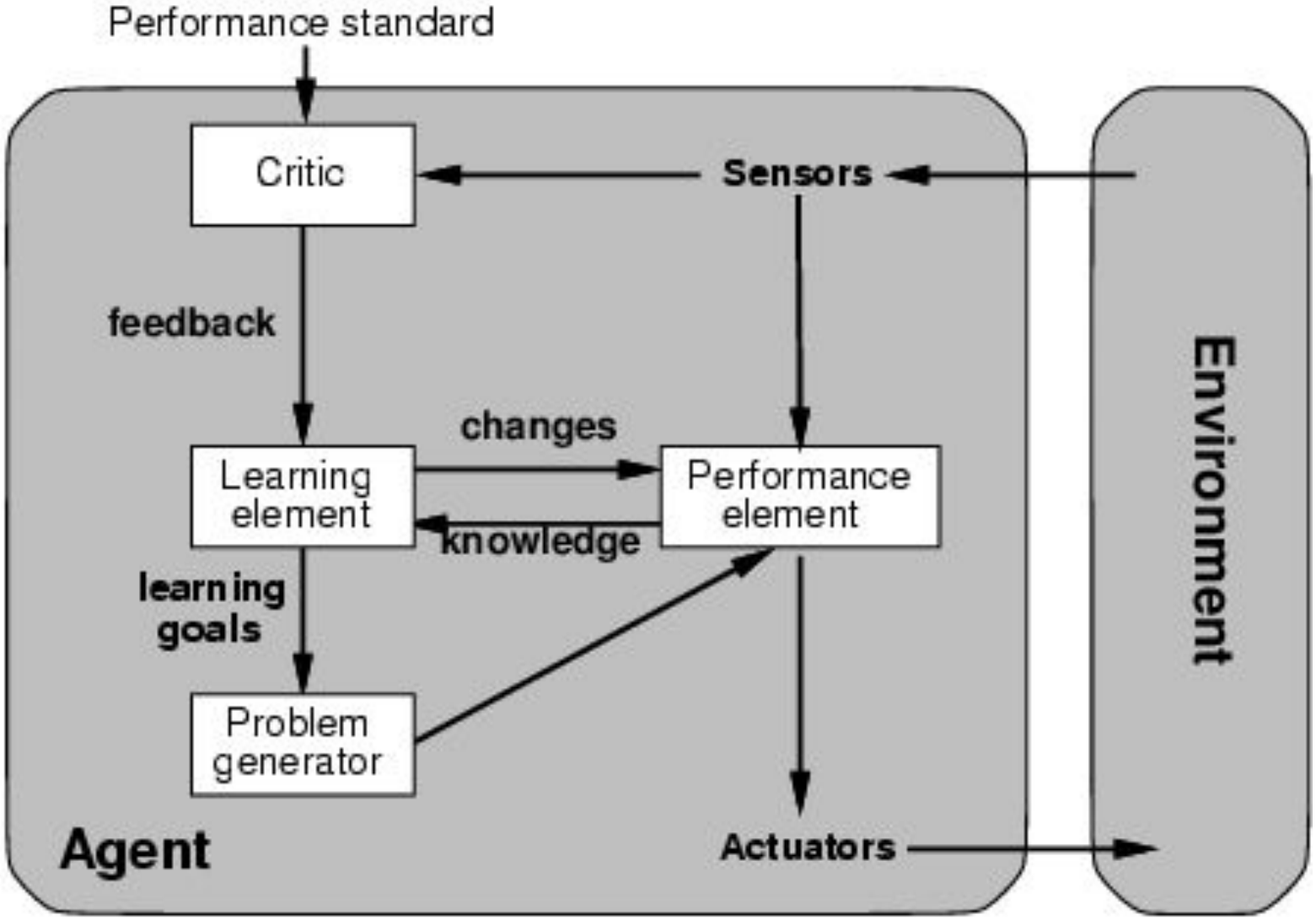
UTILITY BASED AGENTS

- Goals alone are not really enough to generate high-quality behavior in most environments.
- Utility based agents use more than general performance measure that allows a comparison of different world states in different degrees of happiness.
- "happy" does not sound very scientific, the customary terminology is to say that if one world state is preferred to another, then it has higher **utility** for the agent.
- Best suited for decision making agents that must handle the uncertainty inherent in partially observable environments.

UTILITY FUNCTIONS

- A **utility function** maps a state (or a sequence of states) onto a real number, which describes the associated degree of happiness.
- A complete utility function allows rational decisions in two kinds of cases where goals are inadequate.
  - First, when there are conflicting goals, only some of which can be achieved, the utility function specifies the appropriate tradeoff.
  - Second, when there are several goals that the agent can aim for, none of which can be achieved with certainty, utility provides a way in which the likelihood of success can be weighed up against the importance of the goals.

LEARNING AGENTS



- Simple reflex learning agent
  - Mobile keyboard sugestions
- Model based reflex learning agent
  - http://en.akinator.com/personnages/jeu
- Goal based learning agent
  - MYCIN
- Utility based learning agent
  - Taxi Driving agent
  - Any service offering agent

- Turing proposed to build learning machines and then to teach them.
- Learning has another advantage: it allows the agent to operate in initially unknown environments and to become more competent than its initial knowledge alone might allow.

LEARNING VS PERFORMANCE ELEMENT

- learning element is responsible for making improvements, and the performance element is responsible for selecting external actions.
- The performance element is what was previously considered to be the entire agent: it takes in percepts and decides on actions.
- The learning element uses CRITIC feedback and determines how the performance element should be modified to do better in the future.
- The design of the learning element depends very much on the design of the performance element.
- Given an agent design, learning mechanisms can be constructed to improve every part of the agent.

CRITIC

- The critic is necessary because the percepts themselves provide no indication of the agent's success.
  - E.g a percept may receive checkmating opponent, but it needs a performance standard to know that this is a good thing; the percept itself does not say so.

PROBLEM GENERATOR

- problem generator is responsible for suggesting actions that will lead to new and informative experiences.
- The performance element would keep doing the actions that are best, given what it knows.
- But agent may explore a little and do some perhaps suboptimal actions in the short run, it might discover much better actions for the long run.
- It’s job is to suggest these exploratory actions

TAXI DRIVER AS LEARNING AGENT

- The performance element : collection of knowledge and procedures for selecting its driving actions.
  - The taxi goes out on the road and drives, using this performance element.
- The critic: observes the world and passes information along to the learning element.
- learning element: with i/p from critic, formulate a rule saying if actions was good or bad, and then modify performance element by installing the new rule.
- The problem generator might identify certain areas of behavior in need of improvement and suggest experiments.
  - E.g. trying out the brakes on different road surfaces under different conditions

DESIGN OF LEARNING ELEMENT

Three major issues:

- Which *components* of the performance element are to be learned.
- What *feedback* is available to learn these components.

  (*What not to take as a feedback)
- What *representation* is used for the components.

CASE STUDY- AGENT ARCHITECTURE

PROBLEM DEFINITION

- Design a crime investigation agent that goes through incident narration(s) or a question-answer system to give list of suspects and if possible, culprit to the crime.

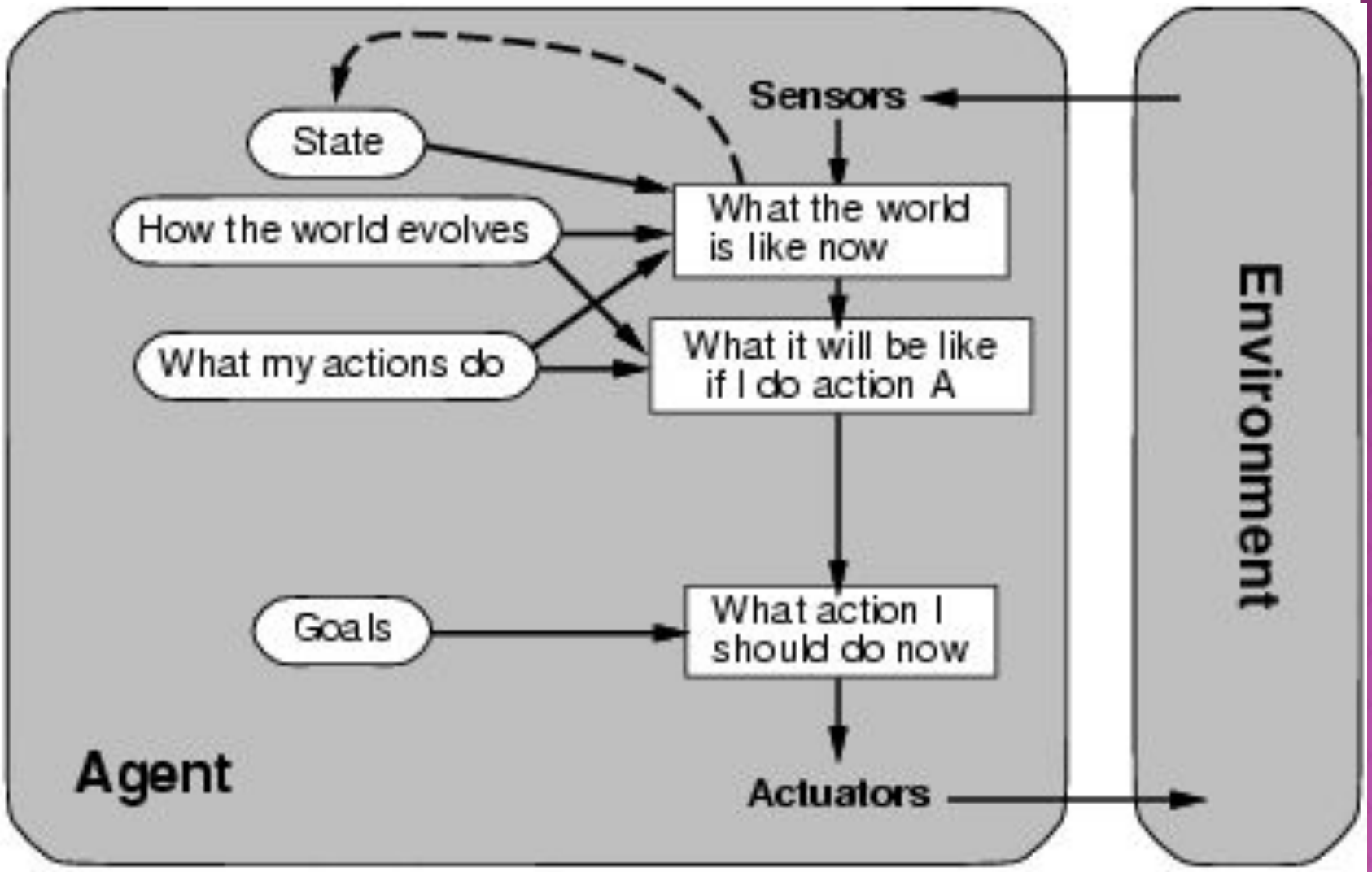
PEAS

- Performance measure-(On the scale of 1 to 10)
  - Clue identification - +8
  - Wrong suspect - -9
  - Resolved case – bonus +10
  - Open case – penalty -10
  - Figured out modus operandi of similar crimes - +10
  - Motive identification - +6 per motive
- Environment- closed room, a data entry person
- Actuators- screen, printer
- Sensors – keyboard, mouse, touchscreen, image reader, video processor

PROPERTIES OF TASK ENVIRONMENT

- Partially observable
- Stochastic
- Sequential
- Dynamic
- Continuous
- Single agent

ARCHITECTURE- GOAL BASED



HOW THE WORLD EVOLVES

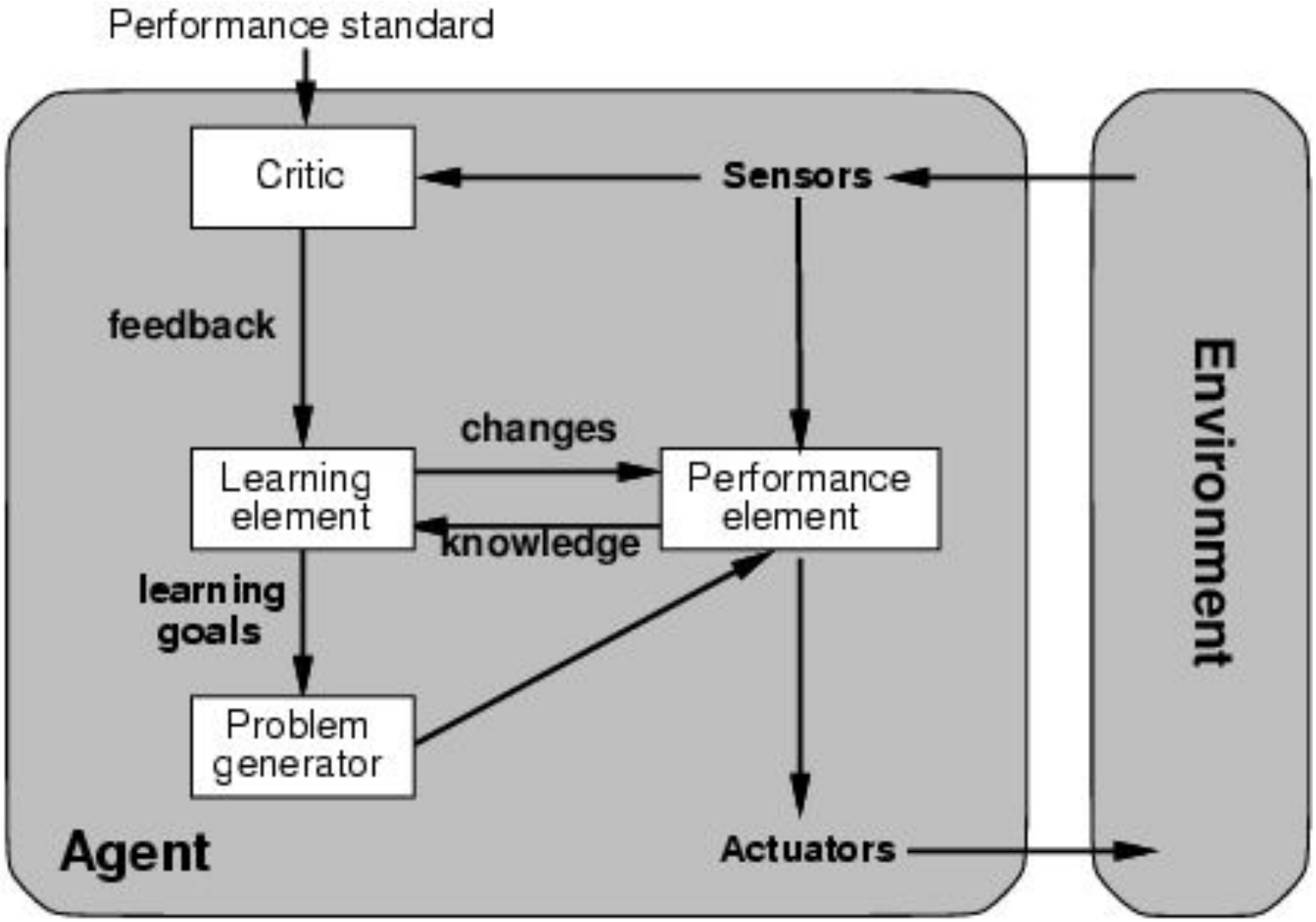
- Basic definitions –crime, suspect, evidence & proof, legal evidence,
- Types of crimes
- Evidence- fingerprint, DNA, belongings, weapons, leaving some trail behind
- Different consequences of different crimes (shoplifting in minor while murder and conspiracies are serious)
- Science of crime investigation
- Human virtues responsible for crime – hate, jealousy, professional crime, psychological issues, money, revenge, challenge, blackmail
- People- economical status, presence or absence of respect or credibility in society with money and/or education, work culture of different professions, availability of tools due to professions, knowledge about Non-tool entities which can act as tool in crime, interpersonal relations-

WHAT MY ACTIONS DO

- Action1: Find_Suspect()
- Action2: Find_Culprit()

- Both actions may have hundreds of rules for finding suspects and culprit based on information provided in “How the world works”
- Not all the rules defined would be applicable to every story, it will vary from story to story
- Information given in “How the world works” and “What my actions do” is generic(Domain knowledge)
- “what the world is like right now” would be case specific

LEARNING AGENTS



GOAL BASED- LEARNING AGENT

- Performance element –
  - Goal based agent

- Learning element
  - Receives new knowledge from PE (E.g. role of mobile location in narrowing suspects in faster and better manner)
  - Creates a rule and adds to “How the world evolves” part so that it can be used for solving any such case; rather than solving only this case
  - Based on this new info, PE may choose different rule to execute in similar situation while solving another crime
  - or may choose to change the sequence of rules to execute while checking on suspects and culprits

GOAL BASED- LEARNING AGENT

- Problem generator
  - May make rules for unseen situations (e.g. solving a case wherein mobile location isnt available)
  - Use it when such a situation arises and based on the effect, may change it or accept it in “How the world evolves” part
  - Will learn better ways to achieve goals or sub-goal(s)
- Critic
  - Sensors give account of situation and rewards/penalties given in performance element tells if its good, better, bad or worst
  - E.g. lots of suspects but no solution/ wrong clues being used etc
  - Learning element updates rules accordingly and changes them in “How the world works” part of performance element

VIDEO RECOMMENDATION SYSTEM??

- Design a video recommendation system on the basis of what the user's watching history and refines the recommendations on the basis of recommendations the user has already chosen to view.
- Design a video recommendation system which takes into consideration the preferences like language, genres and viewing history to display a list of recommended series/movies.

- Design a video recommendation system based on; user views, history, previous view durations and general trends observed.

PROPERTIES OF TASK ENVIRONMENT

- Oservable?
- Deterministic?
- Episodic?
- Static?
- Discrete?
- Single user?

PEAS

- P
- E
- A
- S

PEAS framework to a video recommendation system:

Performance Measure:

- Accuracy: The system should recommend videos that are relevant to the user's interests. This can be measured by the percentage of recommended videos that the user interacts with (e.g., watches, likes, shares).
- Diversity: The system should provide a diverse range of recommendations to avoid monotony. Diversity can be measured by analyzing the variety of genres, topics, or content types in the recommendations.

Environment:

- The environment in this case is the online platform or streaming service where users watch videos. It includes the user interface, the video library, and any other relevant features of the platform.

Actuators:

- Recommendation Engine: The main actuator is the recommendation engine itself. It processes user data, preferences, and historical behavior to generate personalized video recommendations.
- User Interface Controls: The system may have actuators that influence the user interface, such as presenting recommended videos on the homepage or in a dedicated recommendation section.

Sensors:

- User Input: The system gathers user input, including explicit feedback (likes, dislikes) and implicit feedback (watch history, search queries).
- Platform Metrics: The system may also use platform metrics, such as the popularity of videos, to enhance recommendations.
- Contextual Data: Sensors can collect data on the user's current context, such as time of day, location, or device type, to tailor recommendations accordingly.

AGENT ARCHITECTURE(S)?

THANK YOU