Manara University
Faculty of Engineering
Department of IT

Introduction to Artificial Intelligence

Intelligent Agents

Lecture 2

Prof Dr. Eng. Mariam M. Saii

Outline

- Agents and environment
- Rationality
- PEAS (Performance measure, Environment, Actuator, Sensors)
- Environment types
- Agents types

Definition of Agent

- An agent is an entity that perceives and acts

هو كينونة تتلقى مدركات وتنفذ اجراءات

- Abstractly, an agent is a function from percept histories to actions:

العميل هو تابع بموجبه يتم مقابلة عناصر من مجموعة نسميها سلاسل الادراك percept بمجموعة أخرى نسميها مجموعة الإجراءات actions

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

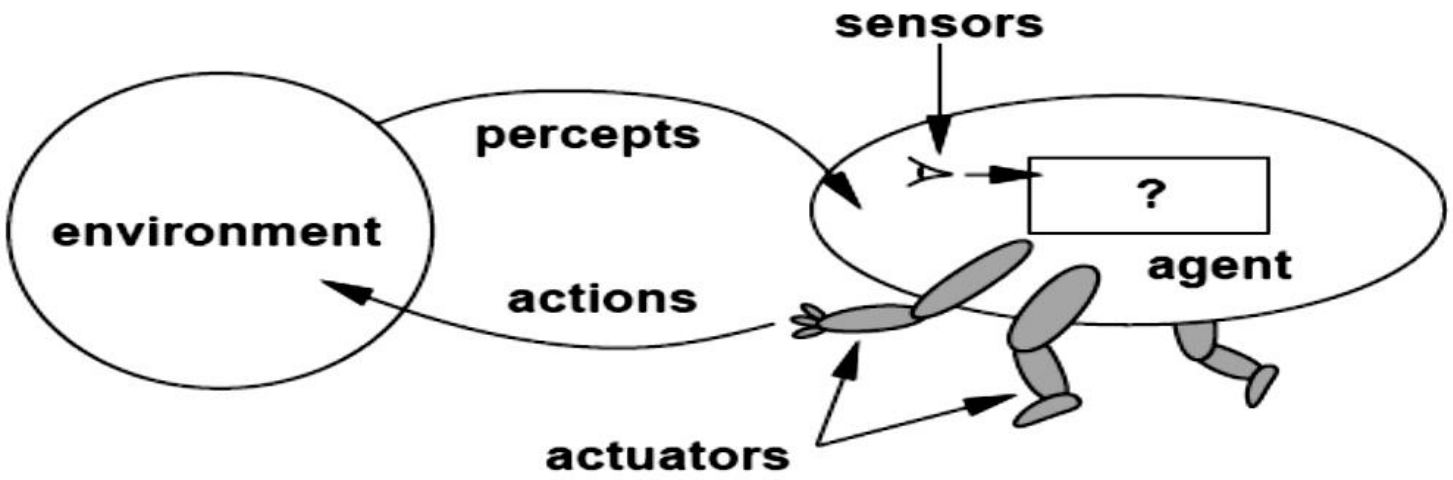
- For any given class of environments and tasks, we seek the agent (or class of agents) with the best performance

من أجل وسط ما ومهمة ما، العميل المناسب (أو مجموعة العملاء) هو ذلك الذي يكون أداؤه الأفضل

- Caveat: computational limitations make perfect rationality unachievable

بسبب محدودية القدرات الحسابية لا يمكن الوصول الى المنطقية الكاملة

Agents and environments



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

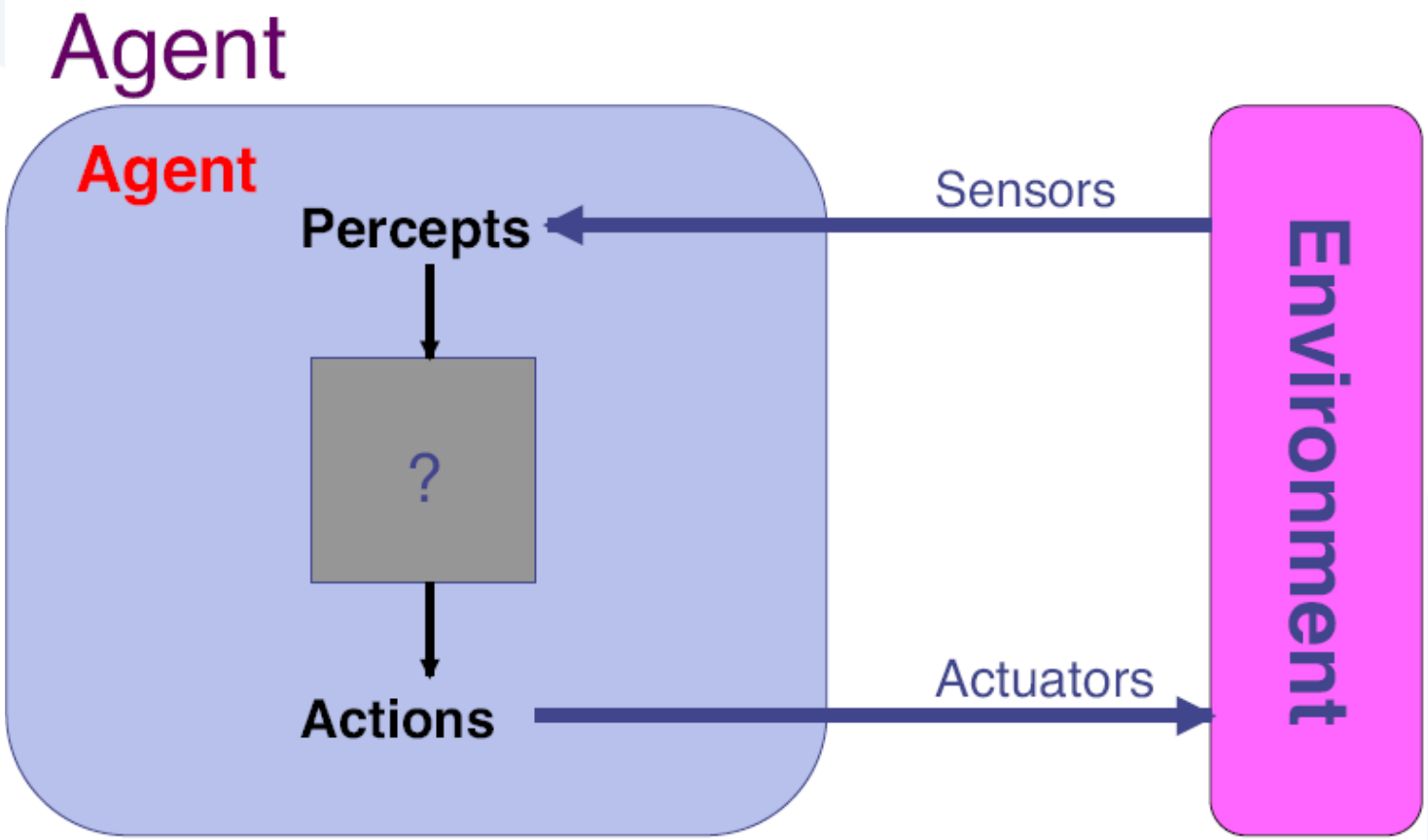
The agent function maps from percept histories to actions:

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

The agent program runs on the physical architecture to produce f

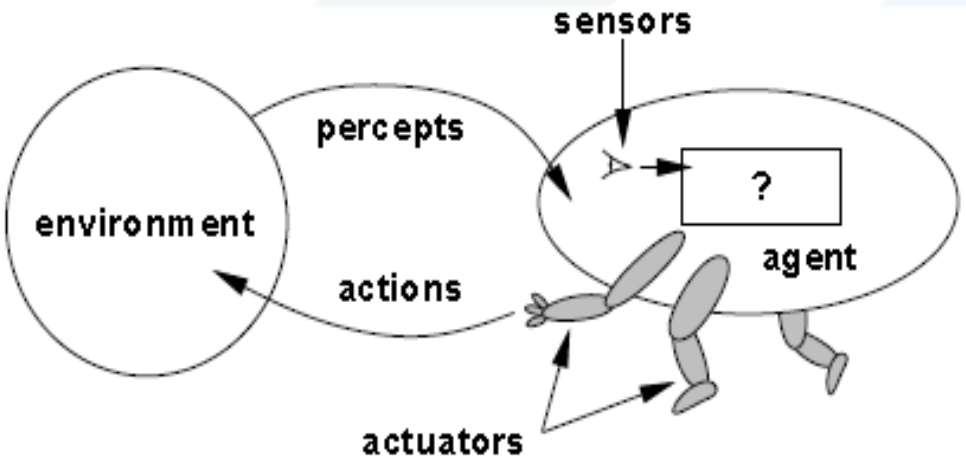
Agent = architecture + program

agent

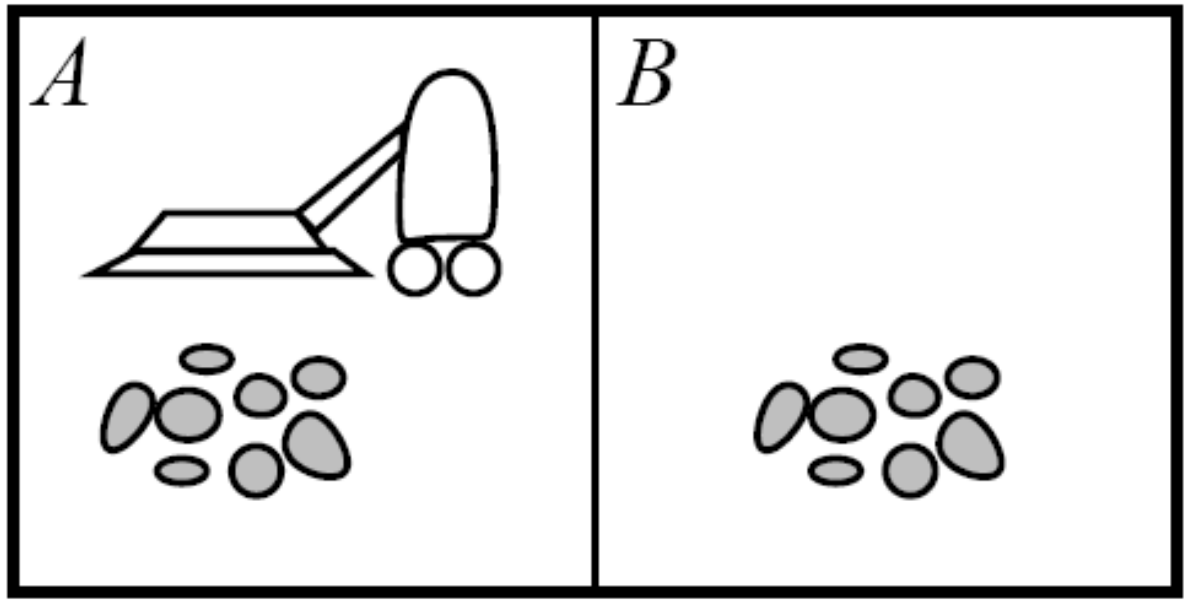

Agent
Agent
Percepts
?
Actions
Sensors
Actuators
Environment

Agents and environments العملاء والبيئة

العميل	الحساسات	المشغلات
انسان	العينان، الأذنان، الأنف، الجلد،	اليدان، الذراعان، الساقان، الفم
روبوت	كاميرا، حساس ضغط، حساس ضوء، ميكروفون.....	ذراع روبوت، محركات، دواليب، مقود،....
برنامج اقلاع	معطيات، سلسلة أحرف، ...	معطيات، سلسلة أحرف، ...



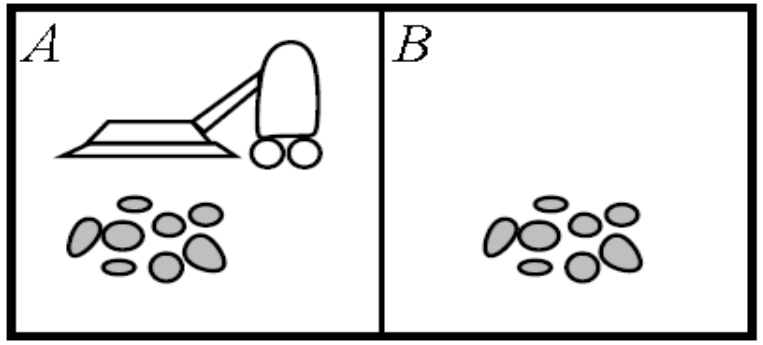
Vacuum –cleaner world



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

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

A vacuum-cleaner agent



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
…	….

- Percepts: location and contents, e.g., [A;Dirty]
- Actions: Left, Right, Suck, NoOp

```
function Reflex-Vacuum-Agent( [ location, status]) returns an action
        if status = Dirty then return Suck
        else if location = A then return Right
        else if location = B then return Left
```

What is rational depends on

– Performance measuring success

– Agents prior knowledge of environment

– Actions that agent can perform

– Agent’s percept sequence to date

-Rational is different from being perfect

Rationality

- A rational agent chooses whichever action maximizes the expected value of the performance measure given the percept sequence to date

  العميل المنطقي هو من يسعى إلى انجاز الشيء الصحيح معتمداً على ما يدركه وعلى ما يستطيع إنجازه من إجراءات والشيء الصحيح هو الاجراء الذي يجعل العميل أكثر نجاحاً في مهمته.

- Rational ≠omniscient العلم بكل شيء
  - percepts may not supply all relevant information
- Rational ≠ clairvoyant مستبصر
  - action outcomes may not be as expected

Hence, rational = successful

Rational ⇨exploration, learning, autonomy مستقل

PEAS

Internet shopping agent

Performance measure : price, quality, appropriateness, efficiency

قياس الأداء : السعر،الجودة، الملاءمة، الكفاءة

Environment : current and future WWW site, vendors, shippers.

البيئة : الموقع الحالي والمستقبلي للشبكة العالمية، والبائعين، والشاحنين

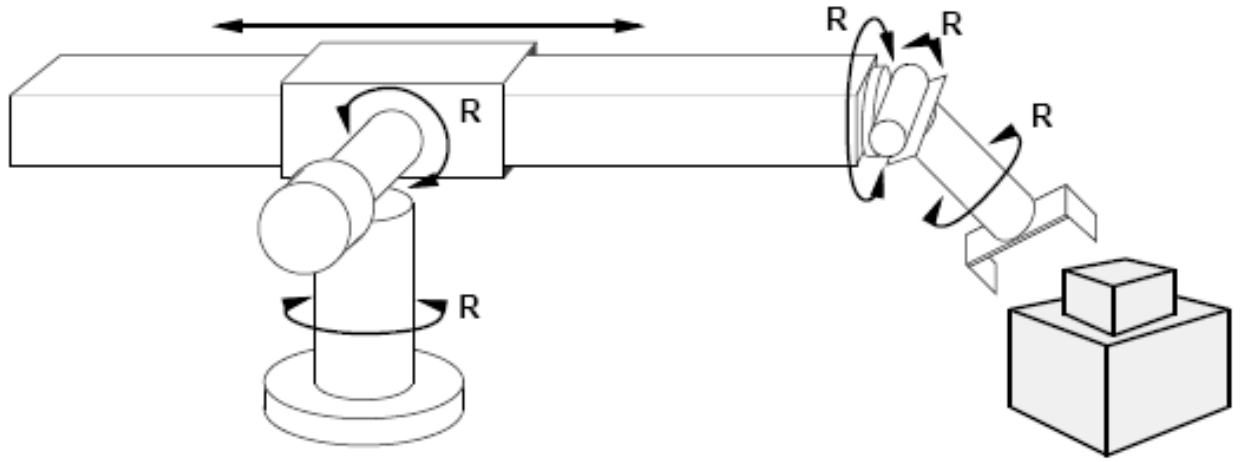
Actuators :display to user, follow URL, fill in form

المحركات : العرض للمستخدم، متابعة عنوانURL ، ملء النموذج

Sensors:HTML pages (text, graphics, scripts)

المستشعرات : صفحات HTML (النصوص والرسومات والبرامج النصية)

PEAS: Part-picking robot



Performance measure: Percentage of parts in correct position
Environment: Parts, robots, workers
Actuators: Jointed arm
Sensors: Camera, point angle sensors

Environment Types

*** Fully observable vs. partially observable.(قابلية الرصد)**

If an agent's sensors give it access to the complete state of the environment at each point in time, then we say that the task environment is fully observable. A task environment is effectively fully observable if the sensors detect all aspects that are *relevant* to the choice of action; relevance, in turn, depends on the performance measure.

- إذا أعطت المستشعرات للوكيل إمكانية الوصول إلى الحالة الكاملة للبيئة في كل نقطة زمنية، فإننا نقول إن بيئة المهمة يمكن ملاحظتها تمامًا. بيئة المهام يمكن ملاحظتها بشكل كامل إذا اكتشفت المستشعرات جميع الجوانب ذات الصلة باختيار الإجراء ؛ الأهمية ، بدورها ، تعتمد على مقياس الأداء

Environment Types

*** Deterministic vs. stochastic.(الحتمية / العشوائية)**

If the next state of the environment is completely determined by the current state and the action executed by the agent, then we say the environment is deterministic; otherwise, it is stochastic.

الحتمية مقابل العشوائية.

إذا تم تحديد الحالة التالية للبيئة تماماً من خلال الحالة الحالية والإجراء الذي ينفذه الوكيل، فإننا نقول إن البيئة حتمية ؛ وإلا فهو عشوائي.

Environment Types

*** Episodic vs. sequential. (العرضية /التسلسلية)**

In an episodic task environment, the agent's experience is divided into atomic episodes.

Each episode consists of the agent perceiving and then performing a single action.

Crucially, the next episode does not depend on the actions taken in previous episodes.

- في بيئة المهام العرضية ، تنقسم خبرة الوكيل إلى حلقات ذرية، تتكون كل حلقة من الوكيل الذي يدرك ثم يقوم بعمل واحد. بشكل حاسم ، الحلقة التالية لا تعتمد على الإجراءات التي تم اتخاذها في الحلقات السابقة

Environment Types

*** Static vs, dynamic.(الثابتة/ الديناميكية)**

If the environment can change while an agent is deliberating, then we say the environment is dynamic for that agent; otherwise, it is static.

- ثابت مقابل ديناميكي. إذا كانت البيئة يمكن أن تتغير أثناء تداول الوكيل ، فإننا نقول إن البيئة ديناميكية لذلك الوكيل ؛ وإلا فإنه ثابت.

Environment Types

*** Discrete vs. continuous.(المستمر/ المتقطع)**

The discrete/continuous distinction can be applied to the *state* of the environment, to the way *time* is handled, and to the *percepts* and *actions* of the agent.

- منفصل مقابل مستمر. يمكن تطبيق التمييز المنفصل / المستمر على حالة البيئة ، والطريقة التي يتم بها التعامل مع الوقت ، وعلى تصورات الوكيل وأفعاله.

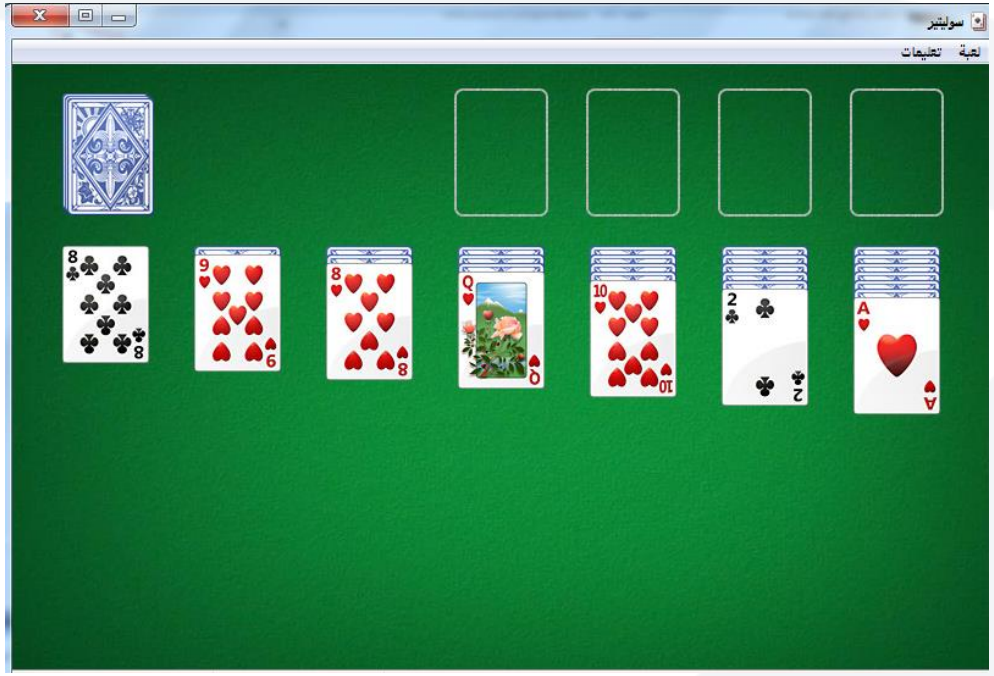
Environment Types

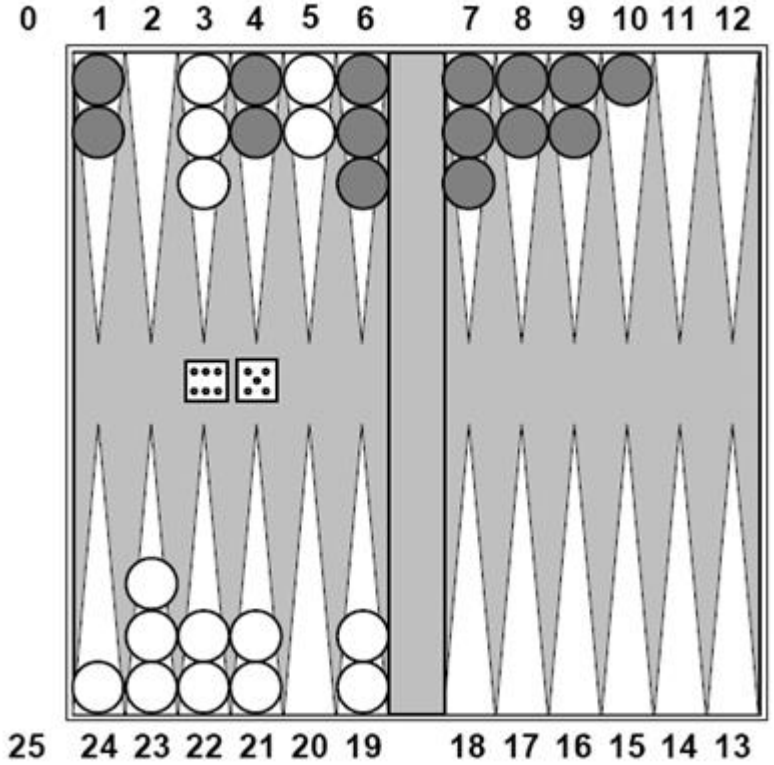
*** Single agent vs. multi agent.(فردي/ متعدد)**

The distinction between single-agent and multi agent environments may seem simple enough

- وكيل واحد مقابل وكلاء متعددين. قد يبدو التمييز بين بيئات الوكيل الفردي والبيئات متعددة الوكلاء بسيطًا بدرجة كافية

backgammon


سوليتير
لعبة تعليمات


0 1 2 3 4 5 6 7 8 9 10 11 12
25 24 23 22 21 20 19 18 17 16 15 14 13

Environment types

	Solitaire	Backgammon	Internet shopping	Taxi
Observable?? Deterministic?? Episodic?? Static?? Discrete?? Single-agent??				

Environment types

	Solitaire	Backgammon	Internet shopping	Taxi
Observable??	Yes	Yes	No	No
Deterministic??				
Episodic??				
Static??				
Discrete??				
Single-agent??				

Environment types

	Solitaire	Backgammon	Internet shopping	Taxi
Observable??	Yes	Yes	No	No
Deterministic??	Yes	No	Partly	No
Episodic??				
Static??				
Discrete??				
Single-agent??				

Environment Types

	Solitaire	Backgammon	Internet shopping	Taxi
Observable??	Yes	Yes	No	No
Deterministic??	Yes	No	Partly	No
Episodic??	No	No	No	No
Static??				
Discrete??				
Single-agent??				

Environment Types

	Solitaire	Backgammon	Internet shopping	Taxi
Observable??	Yes	Yes	No	No
Deterministic??	Yes	No	Partly	No
Episodic??	No	No	No	No
Static??	Yes	Semi	Semi	No
Discrete??				
Single-agent??				

Environment Types

	Solitaire	Backgammon	Internet shopping	Taxi
Observable??	Yes	Yes	No	No
Deterministic??	Yes	No	Partly	No
Episodic??	No	No	No	No
Static??	Yes	Semi	Semi	No
Discrete??	Yes	Yes	Yes	No
Single-agent??				

Environment Types

	Solitaire	Backgammon	Internet shopping	Taxi
Observable??	Yes	Yes	No	No
Deterministic??	Yes	No	Partly	No
Episodic??	No	No	No	No
Static??	Yes	Semi	Semi	No
Discrete??	Yes	Yes	Yes	No
Single-agent??	Yes	No	Yes (except auctions)	No

The environment type largely determines the agent design

The real world is (of course) partially observable, stochastic, sequential, dynamic, continuous, multi-agent

basic kinds of agent programs types

Agent = architecture + program

basic types in order of increasing generality:

- Table-lookup agents العميل المقاد بجدول التقابل
- simple reflex agents العميل المعتمد على رد الفعل
- A model-based reflex agent العميل المعتمد على رد الفعل مع نموذج
- goal-based agents العميل المبني على الهدف
- utility-based agents العميل المعتمد على الخدمة
- Learning- based agents العميل المعتمد على التعلم

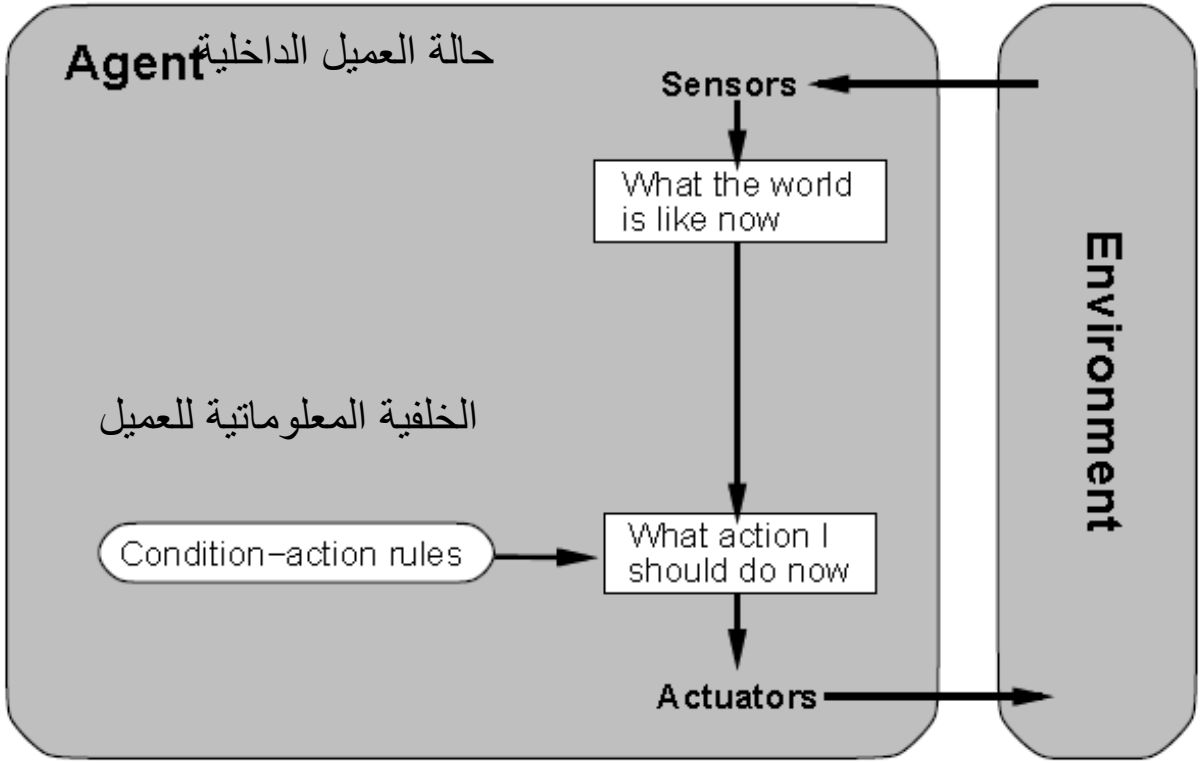
All these can be turned into learning agents

Table- Driven agents العميل المقاد بجدول التقابل

```
function TABLE-DRIVEN-AGENT(percept) returns an action
  persistent: percepts, a sequence, initially empty
              table, a table of actions, indexed by percept sequences, initially fully specified

  append percept to the end of percepts
  action ← LOOKUP(percepts, table)
  return action
```

- Drawbacks:
  - Huge table
  - Take a long time to build the table
  - No autonomy
  - Even with learning, need a long time to learn the table entries

العميل المعتمد على رد الفعل Simple Reflex Agents



قواعد شرط-إجراء (if-then rules)
عند الإنسان هي قواعد غريزية ومكتسبة

- إذا كان الضوء أخضر فيمكنك عبور الطريق
- إذا كان الضوء أحمر إذا قف
- العميل يأخذ قراره بناءا على سلسلة المدركات الحالية فقط.

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

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

العميل المعتمد على رد الفعل Simple Reflex Agents

The simplest kind of agent is the **simple reflex agent.** These agents select actions on the basis of the *current* percept, ignoring the rest of the percept history.
The agent will work *only if the correct decision can be made on the basis of only the current percept-that is, only if the environments fully observable.*

أبسط أنواع الوكلاء هو الوكيل المنعكس البسيط. يختار هؤلاء الوكلاء أفعالهم بناءً على الإدراك الحالي، متجاهلين بقية تاريخ الإدراك. لن يعمل الوكيل إلا إذا أمكن اتخاذ القرار الصحيح بناءً على الإدراك الحالي فقط، أي فقط إذا كانت البيئات قابلة للملاحظة بشكل كامل.

Example

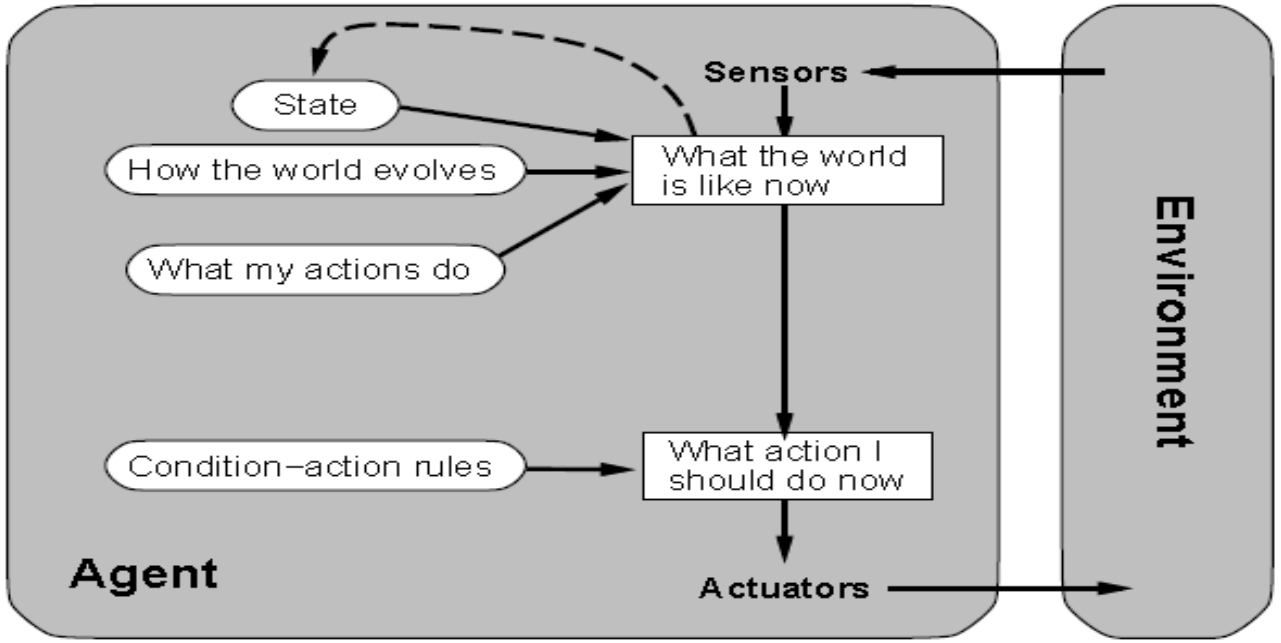
جَامعة المَنارة

function REFLEX-VACUUM-AGENT([*location*,*status*]) **returns** an action

if *status* = *Dirty* **then return** *Suck*
else if *location* = *A* **then return** *Right*
else if *location* = *B* **then return** *Left*

```
(setq joe (make-agent :name 'joe :body (make-agent-body)
                      :program (make-reflex-vacuum-agent-program))

(defun make-reflex-vacuum-agent-program ()
  #'(lambda (percept)
      (let ((location (first percept)) (status (second percept)))
        (cond ((eq status 'dirty) 'Suck)
              ((eq location 'A) 'Right)
              ((eq location 'B) 'Left)))))
```

MODEL based Reflex agents العميل المعتمد على رد الفعل مع ذاكرة



العميل يجد القاعدة التي شروطها تطابق الحالة الراهنة(المعرفة بسلسلة المدركات و بما تم تخزينه في الذاكرة) ثم يقوم بتنفيذ الإجراء الموافق للقاعدة

First, we need some information about how the world evolves independently of the agent.
Second, we need some information about 'how the agent's own actions affect the world

Model-based-Reflex agent

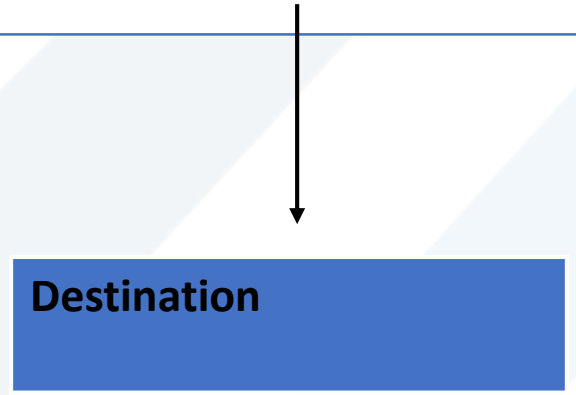
```
function MODEL-BASED-REFLEX-AGENT(percept) returns an action
  persistent: state, the agent's current conception of the world state
              model, a description of how the next state depends on current state and action
              rules, a set of condition–action rules
              action,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

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

The most effective way to handle partial observability is for the agent to *keep track of the part of the world it can't see now.* That is, the agent should maintain some sort of **internal state** that depends on the percept history and thereby reflects at least some of the unobserved aspects of the current state.

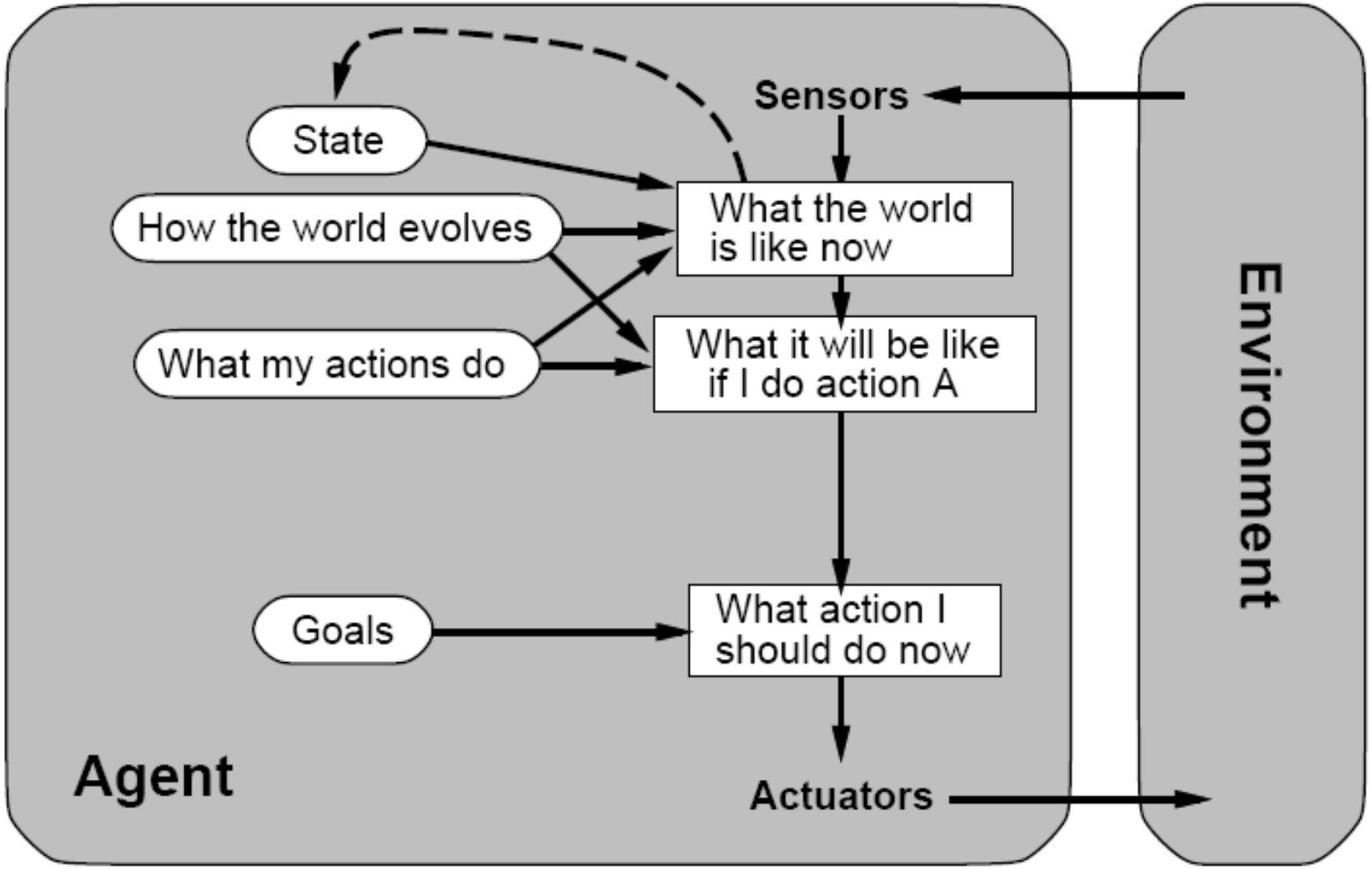
Goal-based agents العميل المبني على الهدف

Knowing about the current state of the environment is not always enough to decide what to do. For example, at a road junction, the taxi can turn left, turn right, or go straight on.

<u>The correct decision depends on where the taxi is trying to get to.</u>



Goal-based agents



Utility-based agents العميل المعتمد على الفائدة

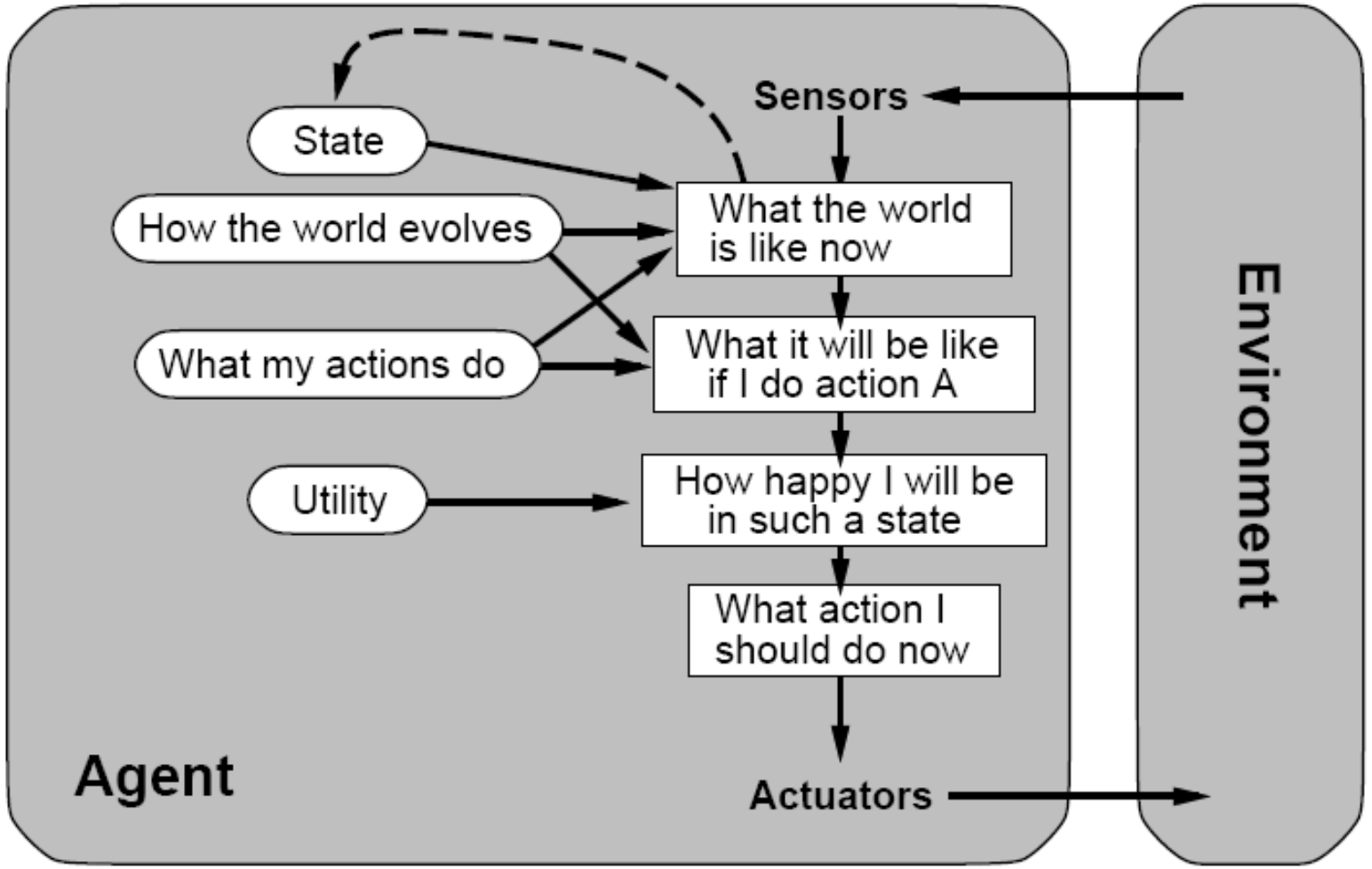
Goals alone are not really enough to generate high-quality behavior in most environments.
For example, there are many action sequences that will get the taxi to its destination (thereby achieving the goal) but some are quicker, safer, more reliable, or cheaper than others.
A utility function maps a state (or a sequence of states) onto a real number, which describes the associated degree of happiness.

الأهداف وحدها لا تكفي لتوليد سلوك عالي الجودة في معظم البيئات. على سبيل المثال، هناك العديد من تسلسلات الإجراءات التي ستوصل سيارة الأجرة إلى وجهتها (مما يحقق الهدف)، ولكن بعضها أسرع وأكثر أمانًا وموثوقية وأقل تكلفة من غيرها. تربط دالة المنفعة حالة (أو سلسلة من الحالات) برقم حقيقي، يصف درجة السعادة المرتبطة بها .

It uses a model of the world, along with a utility function that measures its preferences among states of the world. Then it chooses the action that leads to the best expected utility, where expected utility is computed by averaging over all possible outcome states, weighted by the probability of the outcome.

يستخدم نموذجا للعالم، إلى جانب دالة منفعة تقيس تفضيلاته بين حالات العالم. ثم يختار الإجراء الذي يؤدي إلى أفضل منفعة متوقعة، حيث تُحسب المنفعة المتوقعة بمتوسط جميع حالات النتائج الممكنة، مع ترجيح احتمالية النتيجة .

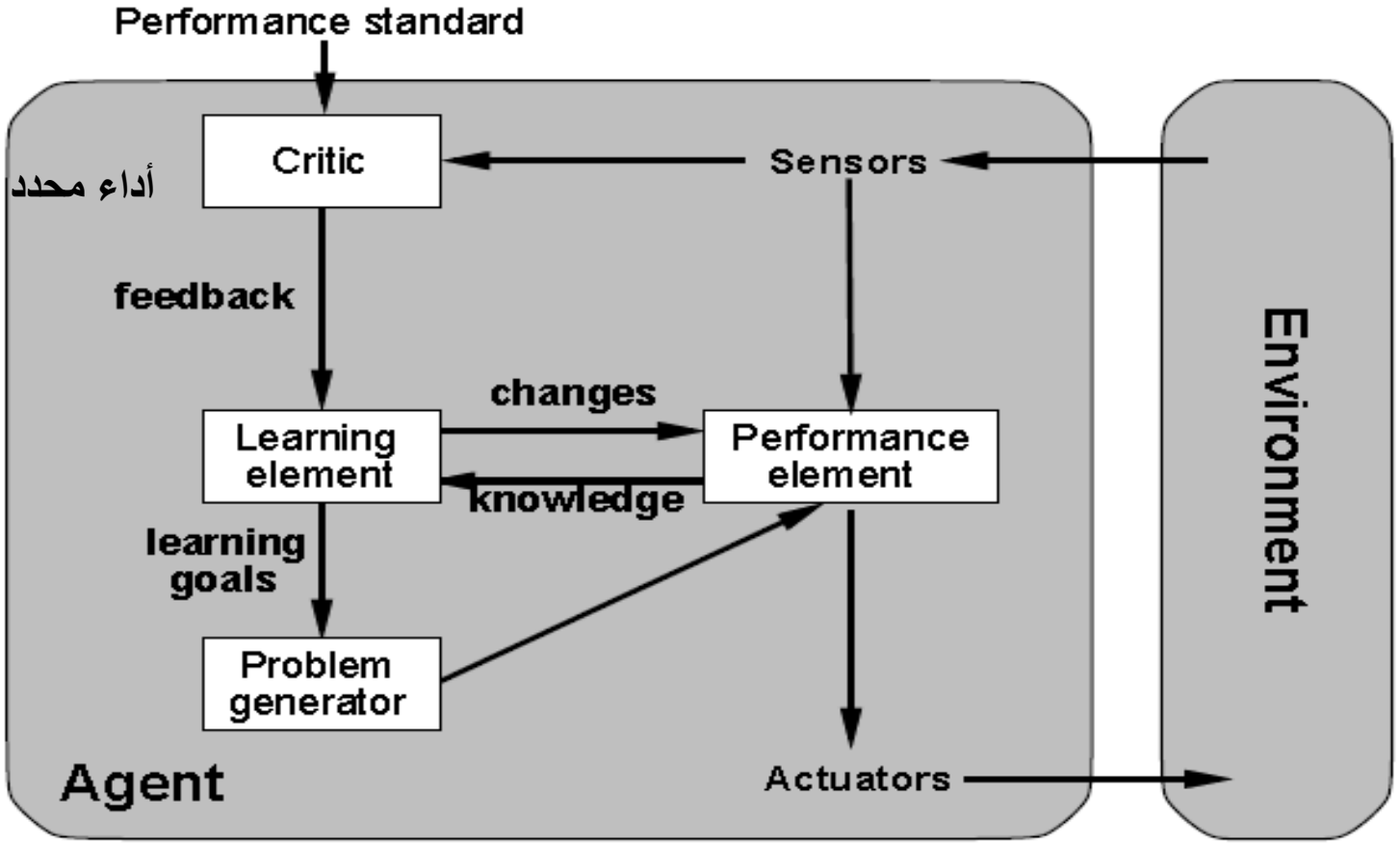
Utility-based agents



Learning agents العميل المتعلم

- يبنى العملاء السابقون على أساس اختيار العميل للإجراء وذلك بفرض أنهم قادرين على الاختيار
- يملك العميل الذكي قابلية تجهيز نفسه بميكانزيم اختيار (يستطيع التعلم)
- الميزة الأساسية لهذا العميل هي القدرة الفائقة على التأقلم مع وسط غير معروف مسبقاً

العميل المعتمد على التعلم Learning agents



Learning agents العميل المتعلم

A learning agent can be divided into four conceptual components

The most important distinction is between the **learning element**, which is responsible for making improvements, and the **performance element**, which is responsible for selecting external actions.

The performance element is what we have previously considered to be the entire agent: it takes in percepts and decides on actions.

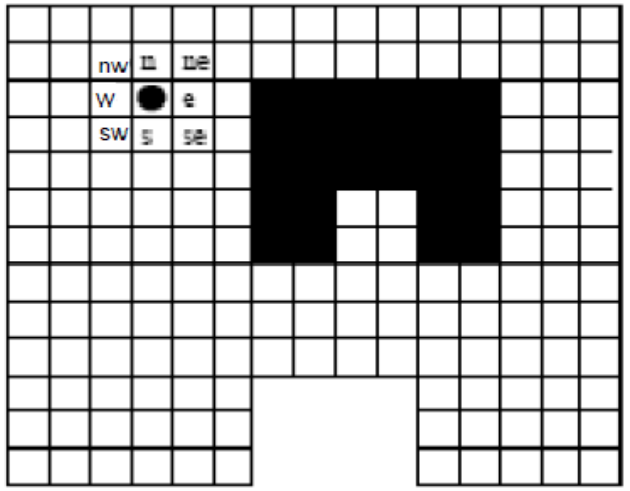
The learning element uses feedback from the **critic** on how the agent is doing and determines how the performance element should be modified to do better in the future.

يمكن تقسيم العميل المتعلم إلى أربعة مكونات مفاهيمية. التمييز الأهم هو بين عنصر التعلم المسؤول عن إجراء التحسينات، وعنصر الأداء المسؤول عن اختيار الإجراءات الخارجية. عنصر الأداء هو ما اعتبرناه سابقًا العميل بأكمله: فهو يستوعب التصورات ويقرر الإجراءات. يستخدم عنصر التعلم تغذية راجعة من الناقد حول أداء العميل، ويحدد كيفية تعديل عنصر الأداء لتحسين أدائه في المستقبل .

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

مُولِّد المشكلات. هو المسؤول عن اقتراح إجراءات تُفضي إلى تجارب جديدة ومفيدة. الفكرة هي أنه إذا ما سارت الأمور على ما يُرام، فسيستمر عنصر الأداء في القيام بأفضل الإجراءات، بناءً على معرفته. ولكن إذا كان العميل مستعدا للاستكشاف قليلا والقيام ببعض الإجراءات التي قد لا تكون مثالية على المدى القصير، فقد يكتشف إجراءات أفضل بكثير على المدى الطويل. تتمثل مهمة مُولِّد المشاكل في اقتراح هذه الإجراءات الاستكشافية .

example



التحكم في روبوت في عالم غرفة فارغة إلا من جدران في وسطها

- **الهدف : يجب على الروبوت الوصول للجدار ومن ثم السير بمحاذاته**
- **الوسط : عالم الغرفة الفارغة إلا من جدران في وسطها**
- **الإجراءات : الروبوت يتحرك في أربع إتجاهات (شرق، غرب ، شمال، جنوب)**
- **المدركات : للروبوت ثماني كميرات (في ثمن إتجاهات) الكميرة التي ترى جدار تعطي قيمة true للمتغير المرافق nw,n,ne,e,se,s,sw,w**
- **انطلاقا من قيم الكميرات يجب تحديد أي إجراء يجب إتخاذه**

لتسهيل وصف القواعد نختار اربع قيم بوليونية توافق ما يجب استخلاصه من المدركات

$$x_1 := n \vee ne$$
$$x_2 := e \vee se$$
$$x_3 := s \vee sw$$
$$x_4 := w \vee nw$$

باستخدام هذه القيم نكتب القواعد المتحكمة في الروبوت او if then roules

$$x_1 \wedge \overline{x_2} \Rightarrow east$$
$$x_2 \wedge \overline{x_3} \Rightarrow south$$
$$x_3 \wedge \overline{x_4} \Rightarrow west$$
$$x_4 \wedge \overline{x_1} \Rightarrow north$$

Summary

Agents interact with environments through actuators and sensors

The agent function describes what the agent does in all circumstances

The performance measure evaluates the environment sequence

A perfectly rational agent maximizes expected performance

Agent programs implement (some) agent functions

PEAS descriptions define task environments

Environments are categorized along several dimensions:
observable? deterministic? episodic? static? discrete? single-agent?

Several basic agent architectures exist:
reflex, reflex with state, goal-based, utility-based