



人工智慧概論

Introduction to AI

第2章 智慧代理人

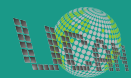
蘇維宗 (Wei-Tsung Su)
suwt@au.edu.tw
564D



歷史版本

版本	說明	日期	負責人
v1.0	初版(智慧代理人)	2020/02/14	蘇維宗

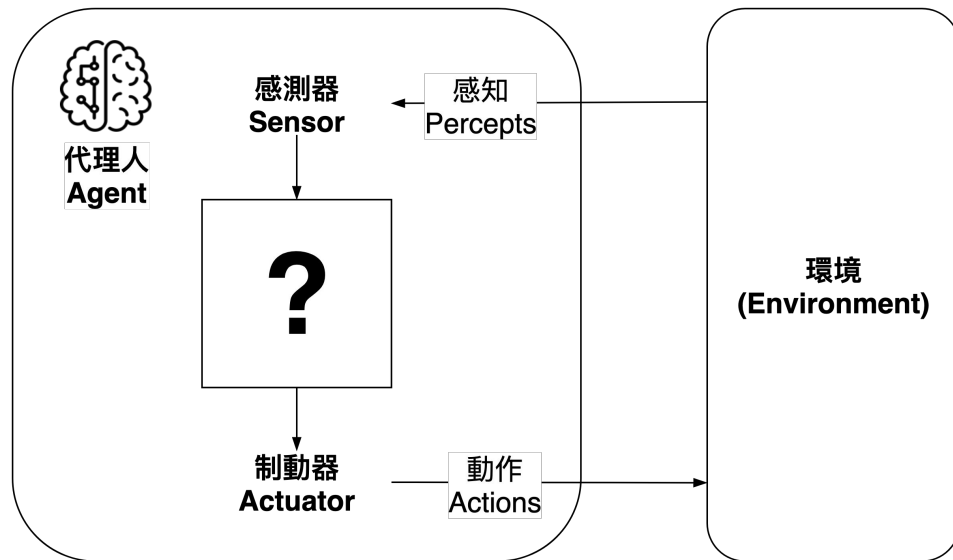
智慧代理人(Intelligent Agent)

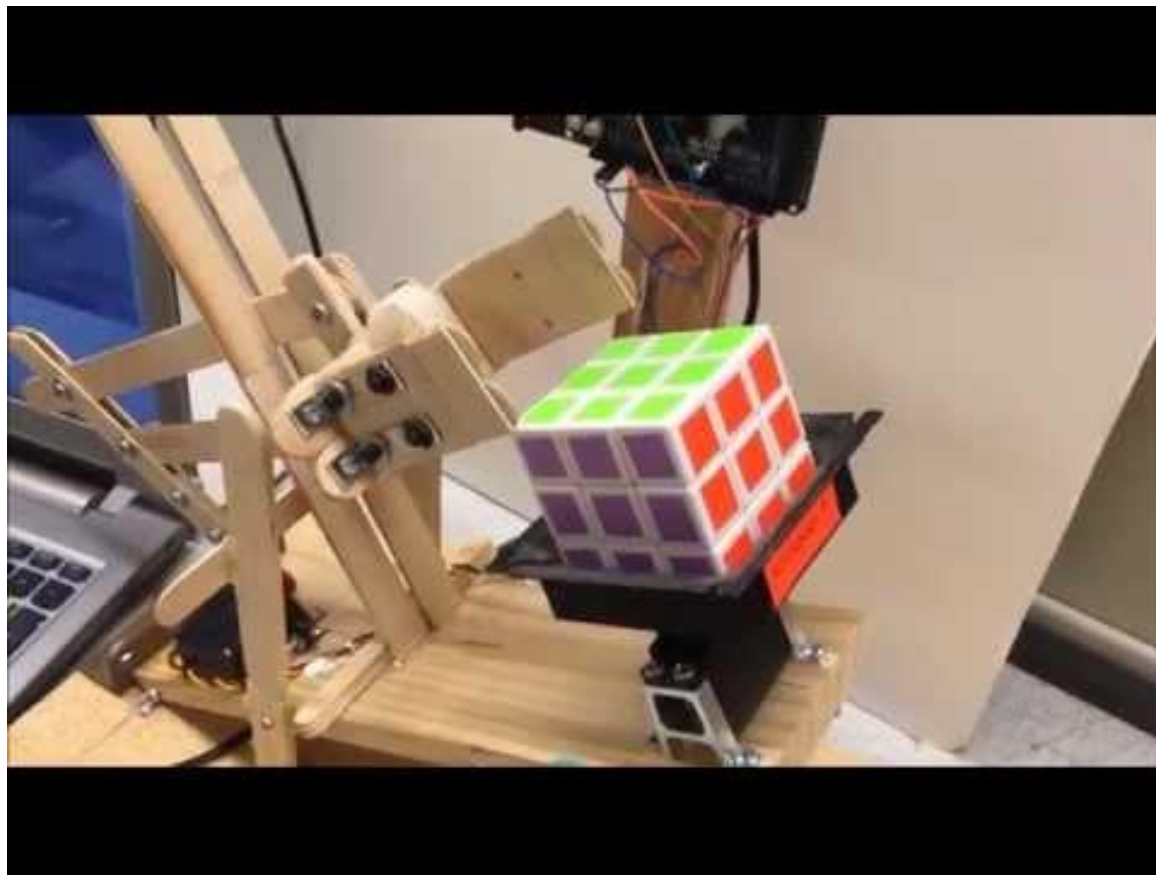


智慧型代理人

代理人(agent)是一種可以透過**感測器(sensors)**與**制動器(actuators)**與環境互動的抽象化實體。

在人工智慧領域中，**智慧型代理人**指的就是可自主運作(autonomy)的代理人。





環境?

感測器?

感知?

制動器?

動作?

AI玩Rubik Cube

代理人功能(Agent Function)

代理人的功能(行為)指的是任意的感知序列與動作的對應。

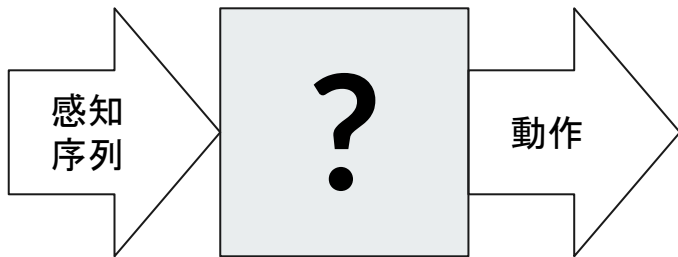


圖. 代理人功能模型

例如,

掃地機器人的功能是根據目前所在範圍**是否骯髒(感知序列)**來決定要**清掃還是移動(動作)**。

AlphaGo的功能是根據**前幾手到目前盤面的順序(感知序列)**來決定要如何下子**(動作)**。

掃地機器人功能

假設掃地機器人在一個只有兩個位置(A與B)的區域中進行清掃。要如何將兩個區域清掃乾淨。

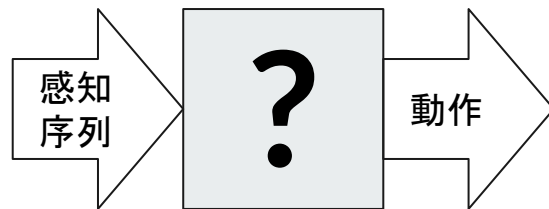
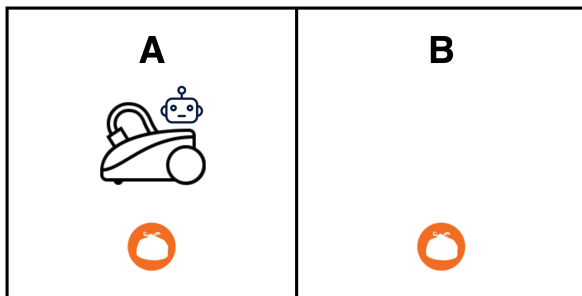


圖. 代理人功能模型

掃地機器人的感知

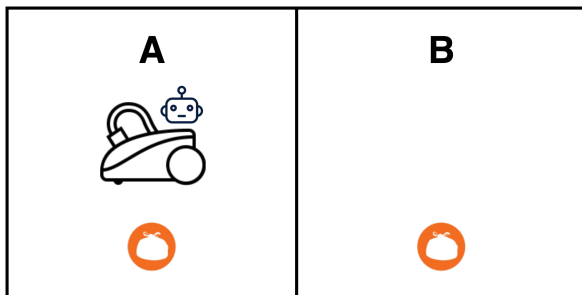
- ???

以及可以擇的動作

- ???

掃地機器人功能(續)

嘗試用**感知序列-動作對應表格**來描述掃地機器人的功能。



感知序列	動作
(A, 乾淨)	右移
(A, 骯髒)	清掃
(B, 乾淨)	左移
(B, 骯髒)	清掃
(A, 乾淨), (A, 乾淨)	右移
(A, 乾淨), (A, 骯髒)	清掃
...	...
(A, 乾淨), (A, 乾淨), (A, 乾淨)	右移
(A, 乾淨), (A, 乾淨), (A, 骯髒)	清掃
...	...

將上面這些對應規則撰寫為代理人程式(agent program)就可以實現掃地機器人, 但**問題是...**



代理人效能評估

如何定義**"好的"**感知序列-動作對應表？

另一方面，什麼叫做**"好的"**感知序列-動作對應表？

所謂**"好的"**，必須透過**效能指標(performance measure)**來評估代理人的一連串動作對環境所造成的狀態改變是否**與期望相符(desirable)**。例如，使用者對掃地機器人的期望會是"在有限動作內將區域中所有位置清掃乾淨"。



代理人效能評估(續)

因此, 合理的代理人定義應該包含

- 用來定義代理人是否成功的效能指標
- 代理人對環境的認知
- 代理人可以執行的動作
- 代理人目前為止擁有的感知序列

對於所有可能的感知序列, 合理的代理人必須根據感知序列所提供的證據與內置的知識(built-in knowledge)選擇一個可以**最大化效能指標**的動作。

範例：定義合理的掃地機器人

- **效能指標**

- 清理一個位置得1分，效能指標為執行1,000個動作後獲得的分數

- **環境認知**

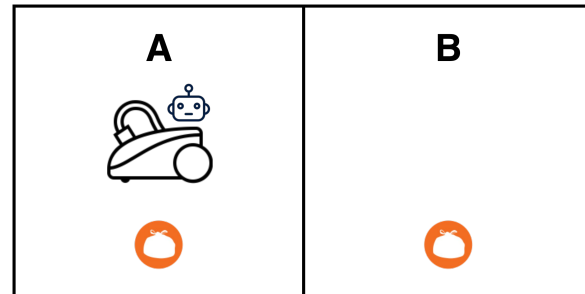
- 環境的地理位置(區域中有哪些位置與位置間的關係為已知，但位置乾淨與否為未知)

- **可採取的動作**

- 左移、右移、與清掃

- **到目前為止擁有的感知序列**

- 目前所在位置以及目前所在位置是否乾淨





練習2-1

練習定義一個合理的代理人

請完成以下工作

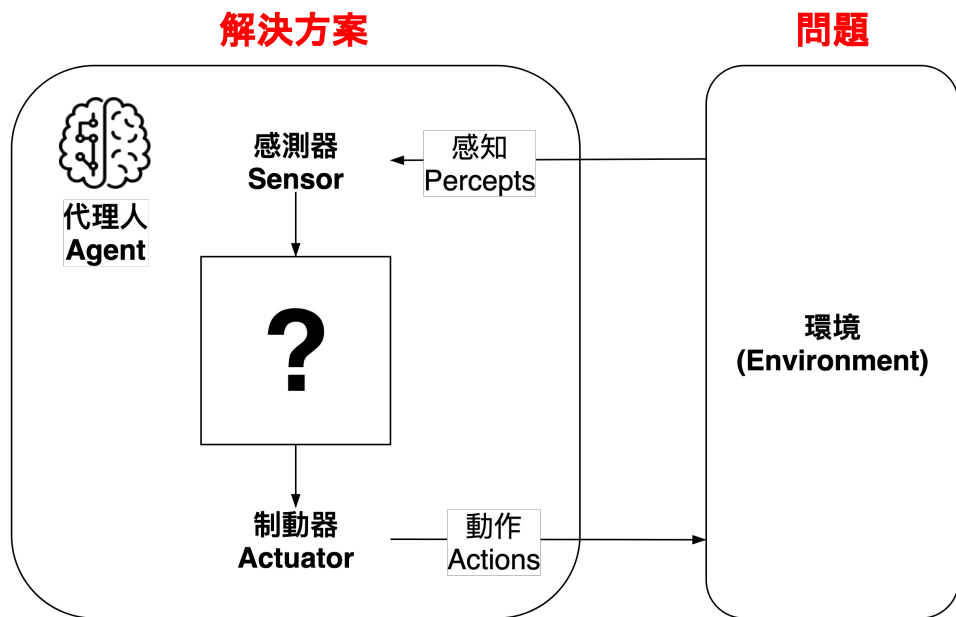
1. 設定一個問題
2. 描述代理人功能
3. 定義合理的代理人
4. 撰寫感知序列-動作對應表

指定工作環境

設計合理的代理人(決解方案), 指定工作環境(問題)也至關重要。

可以透過PEAS來描述工作環境

- Performance measure
- Environment
- Actuators
- Sensors





範例：掃地機器人的工作環境

例如，以PEAS來描述掃地機器人的工作環境如下。

代理人	效能指標	環境	制動器	感測器
掃地機器人	乾淨、速度	區域	清掃器、馬達	灰塵偵測器



範例：自動計程車的工作環境

例如，以PEAS來描述自動計程車的工作環境如下。

代理人	效能指標	環境	制動器	感測器
計程車駕駛	安全、快速、合法、舒服、最大利益	道路、其他車輛、行人、顧客	方向盤、油門、煞車、燈號、喇叭、顯示器	攝影機、聲納、車速表、GPS、哩程表、加速度計、引擎感測器、鍵盤



工作環境的特性

- 完全可觀察 vs. 部分可觀察 (例如, 下棋 vs. 自動駕駛)
- 單代理人 vs. 多代理人 (例如, 下棋 vs. 自動駕駛)
- 決定式 vs. 機率式 (例如, 下棋 vs. 自動駕駛)
- 無前後關係的 vs. 有前後關係的 (例如, 人臉辨識 vs. 下棋)
- 靜態的 vs. 動態的 (例如, 下棋 vs. 自動駕駛)
- 離散的 vs. 連續的 (例如, 下棋 vs. 自動駕駛)
- 已知的 vs. 未知的 (例如, 根據已知結果決定 vs. 必須學習來決定)



練習指定工作環境

工作環境	可完全觀察？	多代理人？	機率式？	有前後關係？	動態？	連續？
撲克牌						
計程車駕駛						
互動英文教學						
填字遊戲						



練習2-2

練習定義工作環境

接續練習2-1, 完成以下工作

1. 以PEAS方式指定工作環境
2. 說明工作環境的特性

代理人程式(Agent Programs)

該是時候想想要如何實作一個代理人程式了。

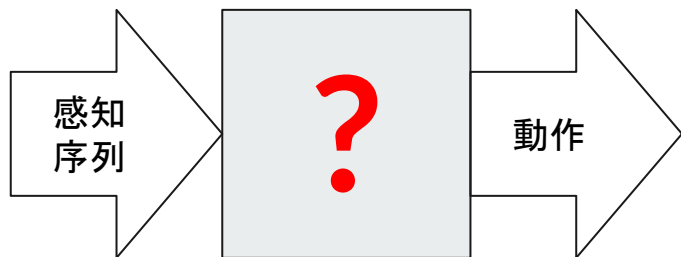


圖. 代理人功能模型

例如, 掃地機器人

感知序列	動作
(A, 乾淨)	右移
(A, 骯髒)	清掃
(B, 乾淨)	左移
(B, 骯髒)	清掃
(A, 乾淨), (A, 乾淨)	右移
(A, 乾淨), (A, 骯髒)	清掃
...	...
(A, 乾淨), (A, 乾淨), (A, 乾淨)	右移
(A, 乾淨), (A, 乾淨), (A, 骯髒)	清掃
...	...

代理人程式(續)

例如，以感知序列-動作對應表來驅動

```
1.  # TABLE-DRIVEN-AGENT
2.  def agent(percepts):
3.      # percepts, 感知序列
4.      # table, 感知序列-動作對應表
5.      action = LOOKUP(percepts, table)
6.      return action
```

例如，掃地機器人

感知序列	動作
(A, 乾淨)	右移
(A, 骯髒)	清掃
(B, 乾淨)	左移
(B, 骯髒)	清掃
(A, 乾淨), (A, 乾淨)	右移
(A, 乾淨), (A, 骯髒)	清掃
...	...
(A, 乾淨), (A, 乾淨), (A, 乾淨)	右移
(A, 乾淨), (A, 乾淨), (A, 骯髒)	清掃
...	...

代理人程式(續)

假設有 n 種可能的感知，且代理人的生命週期是 T ，那感知序列-動作對應表將會有

$$\sum_{t=1}^T n^t$$

個項目。

例如，假設行車記錄器每秒取30幀解析度為640 x 480且顏色深度為24位元的畫面會產生27MB的資料。

如果生命週期為1個小時，那麼感知序列-動作對應表將約會有

$10^{250,000,000,000}$

個項目。

代理人程式的挑戰

因此，人工智慧的挑戰在於如何撰寫小而美的代理人程式而不是參考巨大的表格來讓代理人的績效能夠符合期待？

例如，直到牛頓(Newton)提出計算平方根的方法前都必須透過查表來取得平方根。

SQUARE ROOT CHART 1 TO 100

$\sqrt{1}$	1	$\sqrt{26}$	5.0990	$\sqrt{51}$	7.1414	$\sqrt{76}$	8.7178
$\sqrt{2}$	1.4142	$\sqrt{27}$	5.1962	$\sqrt{52}$	7.2111	$\sqrt{77}$	8.7750
$\sqrt{3}$	1.7321	$\sqrt{28}$	5.2915	$\sqrt{53}$	7.2801	$\sqrt{78}$	8.8318
$\sqrt{4}$	2	$\sqrt{29}$	5.3852	$\sqrt{54}$	7.3485	$\sqrt{79}$	8.8882
$\sqrt{5}$	2.2361	$\sqrt{30}$	5.4772	$\sqrt{55}$	7.4162	$\sqrt{80}$	8.9443
$\sqrt{6}$	2.4495	$\sqrt{31}$	5.5678	$\sqrt{56}$	7.4833	$\sqrt{81}$	9
$\sqrt{7}$	2.6458	$\sqrt{32}$	5.6569	$\sqrt{57}$	7.5498	$\sqrt{82}$	9.0554
$\sqrt{8}$	2.8284	$\sqrt{33}$	5.7446	$\sqrt{58}$	7.6158	$\sqrt{83}$	9.1104
$\sqrt{9}$	3	$\sqrt{34}$	5.8310	$\sqrt{59}$	7.6811	$\sqrt{84}$	9.1652
$\sqrt{10}$	3.1623	$\sqrt{35}$	5.9161	$\sqrt{60}$	7.7460	$\sqrt{85}$	9.2195
$\sqrt{11}$	3.3166	$\sqrt{36}$	6	$\sqrt{61}$	7.8102	$\sqrt{86}$	9.2736
$\sqrt{12}$	3.4641	$\sqrt{37}$	6.0828	$\sqrt{62}$	7.8740	$\sqrt{87}$	9.3274
$\sqrt{13}$	3.6056	$\sqrt{38}$	6.1644	$\sqrt{63}$	7.9373	$\sqrt{88}$	9.3808
$\sqrt{14}$	3.7417	$\sqrt{39}$	6.2450	$\sqrt{64}$	8	$\sqrt{89}$	9.4340
$\sqrt{15}$	3.8730	$\sqrt{40}$	6.3246	$\sqrt{65}$	8.0623	$\sqrt{90}$	9.4868
$\sqrt{16}$	4	$\sqrt{41}$	6.4031	$\sqrt{66}$	8.1240	$\sqrt{91}$	9.5394
$\sqrt{17}$	4.1231	$\sqrt{42}$	6.4807	$\sqrt{67}$	8.1854	$\sqrt{92}$	9.5917
$\sqrt{18}$	4.2426	$\sqrt{43}$	6.5574	$\sqrt{68}$	8.2462	$\sqrt{93}$	9.6437
$\sqrt{19}$	4.3589	$\sqrt{44}$	6.6332	$\sqrt{69}$	8.3066	$\sqrt{94}$	9.6954
$\sqrt{20}$	4.4721	$\sqrt{45}$	6.7082	$\sqrt{70}$	8.3666	$\sqrt{95}$	9.7468
$\sqrt{21}$	4.5826	$\sqrt{46}$	6.7823	$\sqrt{71}$	8.4261	$\sqrt{96}$	9.7980
$\sqrt{22}$	4.6904	$\sqrt{47}$	6.8557	$\sqrt{72}$	8.4853	$\sqrt{97}$	9.8489
$\sqrt{23}$	4.7958	$\sqrt{48}$	6.9282	$\sqrt{73}$	8.5440	$\sqrt{98}$	9.8995
$\sqrt{24}$	4.8990	$\sqrt{49}$	7	$\sqrt{74}$	8.6023	$\sqrt{99}$	9.9499
$\sqrt{25}$	5	$\sqrt{50}$	7.0711	$\sqrt{75}$	8.6603	$\sqrt{100}$	10

© 2019 DADSWORKSHEETS.COM, LLC

DadsWorksheets.com

圖片來源:

<https://www.dadsworksheets.com/charts/square-root-chart.html>



練習2-3

練習撰寫掃地機器人(2 X 1)的代理人程式

請用目前所學撰寫一個掃地機器人程式可以清掃2 X 1的區域



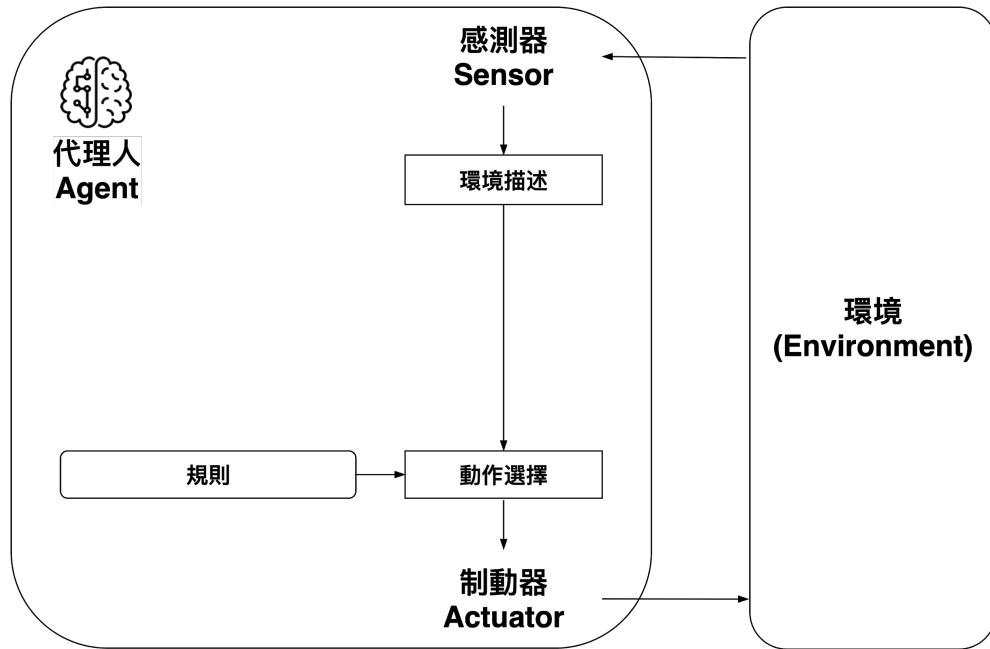
代理人程式的四種基本型態

- 簡單反射型代理人(Simple Reflex Agents)
 - 根據**目前的感知**來選擇動作
- 模型反射型代理人(Model-based Reflex Agents)
- 目標導向型代理人(Goal-based Agents)
- 效益導向型代理人(Utility-based Agents)

簡單反射型代理人

模型反射型代理人虛擬碼

```
1. def SRAgent(precept):  
2.     state = INTERPRET(percept)  
3.     rule = MATCH(state, rules)  
4.     action = rule.ACTION  
5.     return action
```





練習2-4

練習撰寫掃地機器人(2 X 1)的代理人程式

以簡單反射型代理人的方式重新
撰寫可以清掃2 X 1的區域的掃
地機器人程式



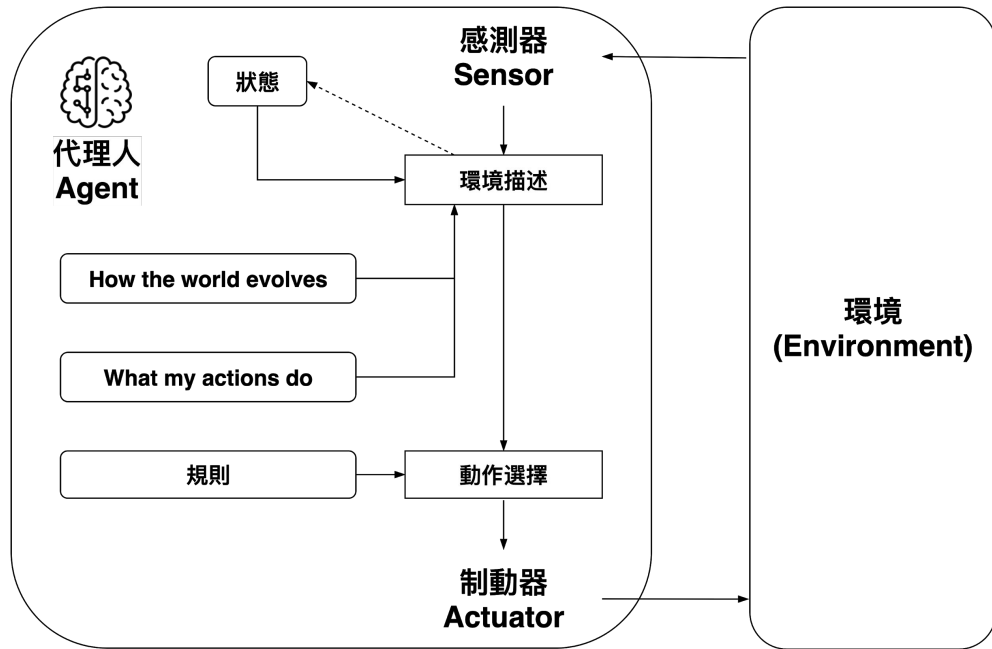
代理人程式的四種基本型態

- 簡單反射型代理人(Simple Reflex Agents)
- 模型反射型代理人(Model-based Reflex Agents)
 - 根據模型(加入考慮環境的變化歷史? 動作造成的影響?)來選擇動作
- 目標導向型代理人(Goal-based Agents)
- 效益導向型代理人(Utility-based Agents)

模型反射型代理人

模型反射型代理人虛擬碼

```
1. def MRAgent(precept):
2.     state = UPDATE-STATE(
3.         state, action,
4.         percept, model)
5.     rule = MATCH(state, rules)
6.     action = rule.ACTION
7.     return action
```





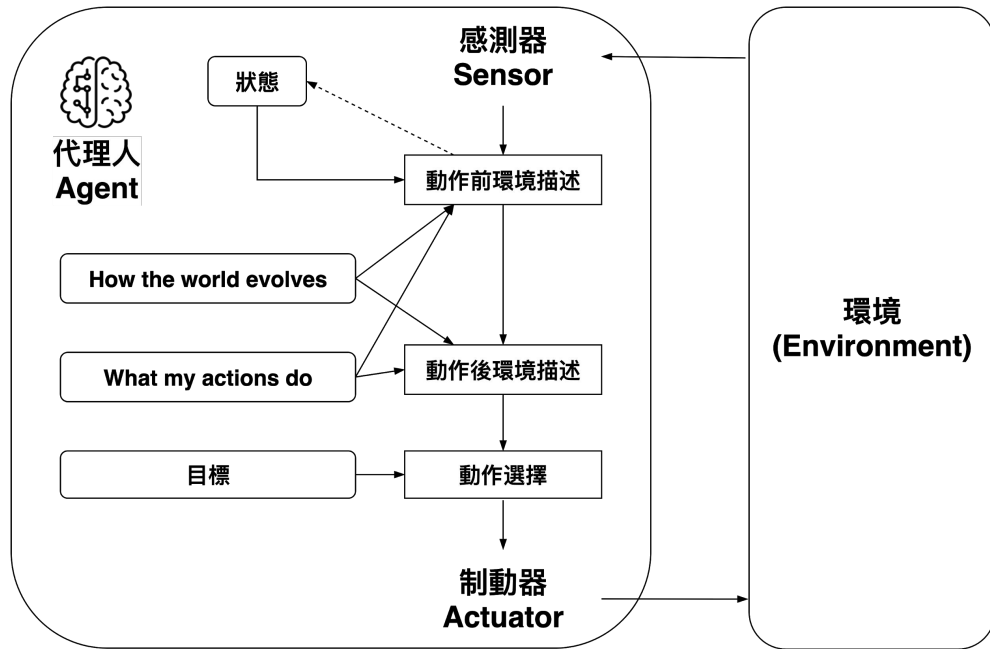
代理人程式的四種基本型態

- 簡單反射型代理人(Simple Reflex Agents)
- 模型反射型代理人(Model-based Reflex Agents)
- 目標導向型代理人(Goal-based Agents)
 - 根據目標(加入考慮未來的變化)來選擇動作
- 效益導向型代理人(Utility-based Agents)

目標導向型代理人

根據事先確立的目標來選擇動作。

例如，搜尋(search)從A點走到B點的路徑。





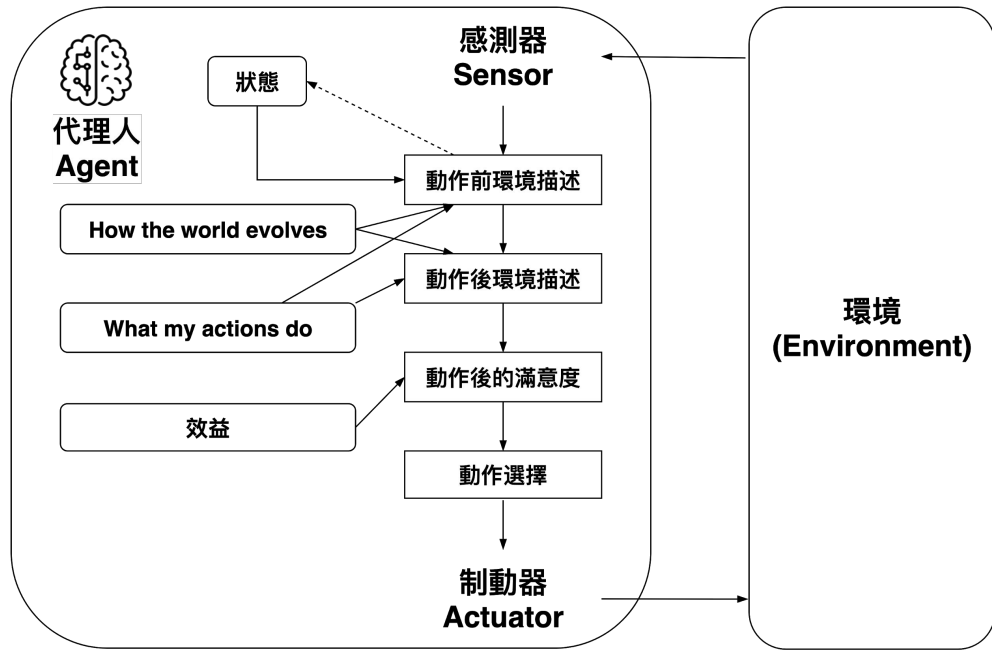
代理人程式的四種基本型態

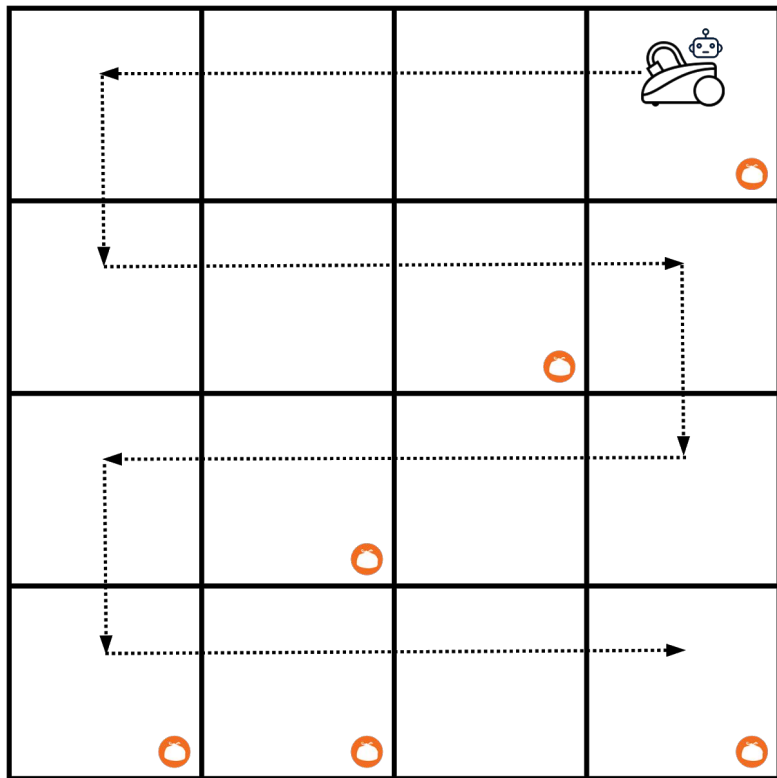
- 簡單反射型代理人(Simple Reflex Agents)
- 模型反射型代理人(Model-based Reflex Agents)
- 目標導向型代理人(Goal-based Agents)
- 效益導向型代理人(Utility-based Agents)
 - 根據**效益(加入考慮可以量化的效能指標)**來選擇動作

效益導向型代理人

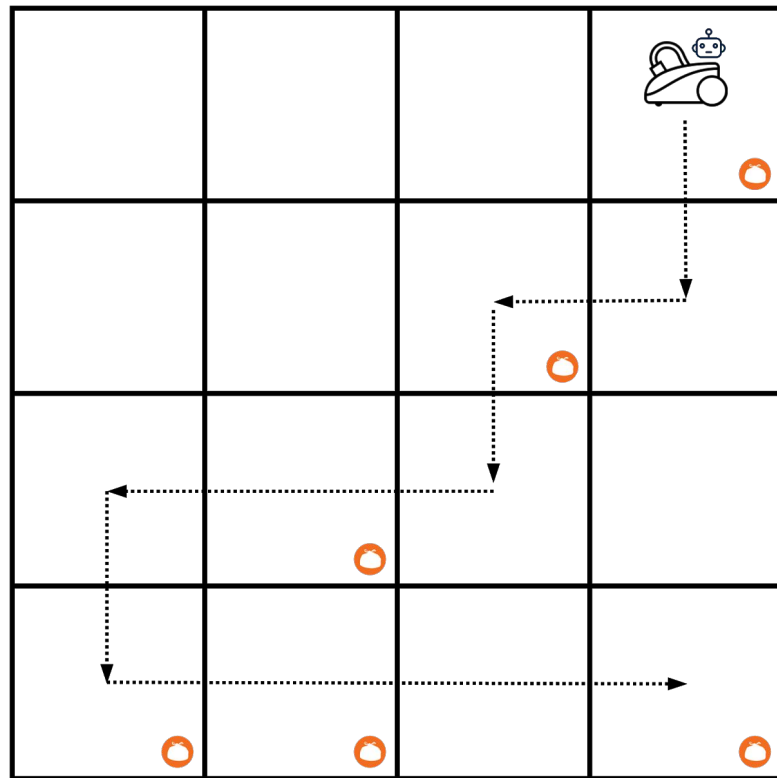
根據事先確立的目標以最大的效益來選擇動作。

例如，搜尋(search)從A點走到B點的**最短路徑**。





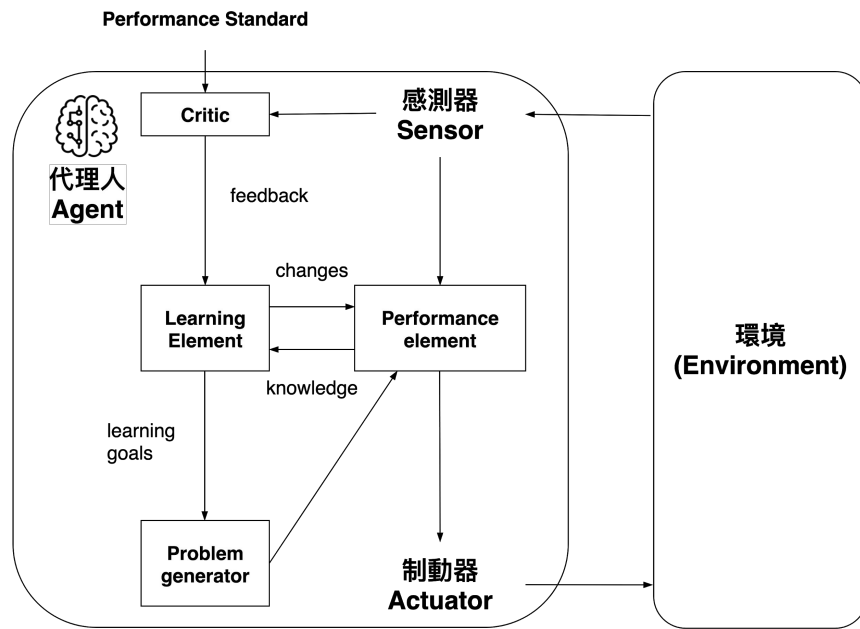
目標導向型代理人



效益導向型代理人

學習型代理人

- Learning element
 - 負責改善
- Performance element
 - 負責選擇行動
- Critic
 - 評估代理人的績效
- Problem generator
 - 建議能夠帶來新的且具有豐富資訊之體驗的動作





作業2-1

練習撰寫掃地機器人(2 X 2)的代理人程式

繳交期限: 03/24/2020

繳交方式: ipynb檔案

請完成以下工作

1. 設定一個問題
2. 描述代理人功能
3. 定義合理的代理人
4. 撰寫感知序列-動作對應表
5. 以簡單反射型代理人的方式
撰寫可以清掃2 X 2的區域
的掃地機器人程式



狀態改變的形式

不管是什麼類型的代理人需要知道

"What the world is like now?"

"What it will be like if do action A?"

所以如何描述環境的狀態非常重要

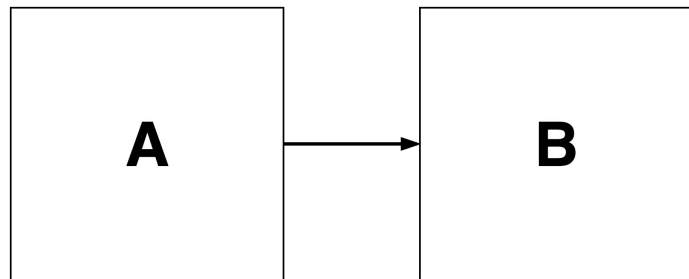
三種描述環境狀態與其改變的方式

- Atomic
- Factored
- Structured



Atomic

環境的狀態是個黑盒子。

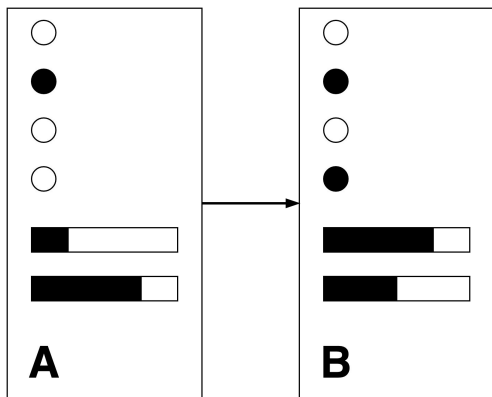


問題範例

- Search
- Game-playing
- Hidden Markov models
- Markov decision processes

Factored

環境的狀態可以表示成一個屬性的向量。

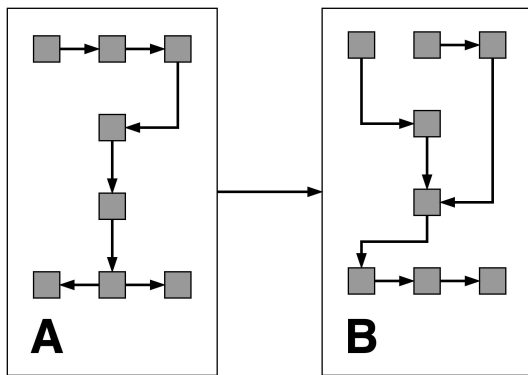


問題範例

- Constraint satisfaction algorithms
- Propositional logic
- Planning
- Bayesian networks
- Machine learning

Structured

環境的狀態包含好幾個物件，每個物件有自己的狀態且物件之間是有關聯性的。



問題範例

- Relational databases
- First-order logic
- First-order probability models
- Knowledge-based learning
- Natural language understanding



作業2-2

接續練習2-1與練習2-2, 撰寫代理人程式來解決自己定義的問題

繳交期限: 04/07/2020

繳交方式: ipynb檔案

請完成以下工作

1. 設定一個問題
2. 描述代理人功能
3. 定義合理的代理人
4. 撰寫感知序列-動作對應表
5. 以簡單反射型代理人的方式
撰寫可以清掃2 X 2的區域
的掃地機器人程式

Q & A



Computer History Museum, Mt. View, CA