

講義 6: 即応/ハイブリッドアーキテクチャ  
*Reactive and Hybrid Architecture*

# 内容

1. 即応アーキテクチャ
2. 包摂アーキテクチャ
  - (a) 状況オートマトン
3. ハイブリッドアーキテクチャ
  - (a) TouringMachines
4. 参考文献

# 1. 即応アーキテクチャ

---

- 記号的 AI には未解決問題 (変換問題, 表現/推論問題など) が多数ある
- これらの問題により, 一部の研究者達は AI パラダイムの生存可能性に疑問を持ち始めた. そして, 即応アーキテクチャの開発に向かった.
- この講義では, AI の主流に対して最も批判的な Rodney Brooks の研究の紹介からはじめる.

## エージェントにおける問題 (講義#2 より)

エージェントを構築する場合，解決すべき二つの重要な問題がある．

1. **変換問題** (transduction problem) 実世界の問題を，有用性を損なわないうちに正確で適切な世界の記号的記述に変換する問題．
2. **表現/推論問題** (representation/reasoning problem) 有用性を損なわないうちに，情報を記号的に表現し，エージェントにそれを操作/推論させる問題．
  - 前者の問題は視覚，音声理解，学習などの研究となった．
  - 後者の問題は知識表現，推論，プランニングなどの研究を生じさせた．

## 2. 包摂アーキテクチャ (1/7)

- Rodney Brooks は以下の 3 点を強調する [Brooks91a,Brooks91b]
  1. 知的な振舞いは，記号的 AI が提唱するような**明示的な記号表現がなくとも**生成されうる
  2. 知的な振舞いは，記号的 AI が提唱するような**明示的な抽象的推論がなくとも**生成されうる
  3. 知能は，ある種の複雑なシステムの**発現特性 (emergent property)** である

## 2. 包摂アーキテクチャ (2/7)

- 彼は自分の研究における 2 つの鍵となる考えを示した
  1. 状況性 (situatedness) と身体性 (embodiment): 実知能は実際の世界の状況に置かれているのであり, 定理証明器やエキスパートシステムのように非身体的システムではない
  2. 知能 (intelligence) と創発性 (emergence): 知的な振舞いはエージェントが環境と相互作用する結果として生じる. さらに, 知能は「エージェントの視点」によるものであり, それは生得的な分離された属性ではない.

## 2. 包摂アーキテクチャ (3/7)

- これらの考え方を例証するために，Brooks は包摂アーキテクチャ (Subsumption Architecture) に基づいたシステムを構築した
- 包摂アーキテクチャは **タスク達成挙動** の階層である
- 各挙動は単純な規則のような構造である
- エージェントの制御を習得するために，各挙動は他の挙動と競合する
- 下位階層は，より原始的な種類の挙動 (例えば，障害物回避) を表し，上位階層よりも優先順位が高い
- 結果として得られるシステムは，必要となる計算という意味において，非常に単純となる
- ロボットは，もしもそれが記号的 AI システムで達成されたとしたら，とても印象的なタスクを行うことができる

## 2. 包摂アーキテクチャ (4/7)

```
Function: Action Selection
1.  function action( $p : P$ ) :  $A$ 
2.  var fired :  $\wp(R)$ 
3.  var selected :  $A$ 
4.  begin
5.      fired  $\leftarrow \{(c, a) \mid (c, a) \in R \text{ and } p \in c\}$ 
6.      for each  $(c, a) \in \textit{fired}$  do
7.          if  $\neg(\exists(c', a') \in \textit{fired}$ 
              such that  $(c', a') \prec (c, a))$  then
8.              return  $a$ 
9.          end-if
10.     end-for
11.     return null
12. end function action
```



## 2. 包摂アーキテクチャ (5/7)

- Steels の火星探査システム [Steels90] は、包摂アーキテクチャを用い、火星における岩石収集タスクのシミュレータで、ほぼ最適な協調動作を達成した。  
このタスクにおける目的は、遠く離れた惑星で探索を行い、貴重な岩石のサンプルを収集することである。サンプルの位置は事前にはわからないが、それらは密集しがちであることは事前に分かっている。

## 2. 包摂アーキテクチャ (6/7)

- 各 (非協調的) エージェントの最低レベルの挙動 (つまり最も高い優先順位を持つ挙動) は, **障害物回避**である

*if detect an obstacle then change direction (1)*

- エージェントが持っているすべてのサンプルは母船に落とされる

*if carrying samples and at the base then drop samples (2)*

- サンプルを持っているエージェントは母船に引き返す

*if carrying samples and not at the base then travel up gradient (3)*

## 2. 包摂アーキテクチャ (7/7)

- エージェントは発見したサンプルを収集する

*if detect a sample then pick sample up (4)*

- 何もすることがないエージェントはランダムに探索する

*if true then move randomly (5)*

これらのルールの前提条件は以下の優先順位で発火する

(1) < (2) < (3) < (4) < (5)

## 2.1. 状況オートマトン (1/2)

- 包摂アーキテクチャに関する洗練されアプローチは Rosenschein と Kaelbling による [Rosenstein86]
- **状況オートマトンパラダイム**では，エージェントは宣言的規則言語で仕様記述され，この仕様は，その宣言仕様を充足するデジタルマシンにコンパイルされる  
このデジタルマシンは**証明可能時間限界**で動作する
- 推論は，実行時にオンラインで行うのではなく，コンパイル時にオフラインで行われる

[An agent]  $x$  is said to carry the information that  $p$  in world state  $s$ , written  $s \models K(x, p)$ , if for all world states in which  $x$  has the same value as it does in  $s$ , the proposition  $p$  is true. [Kaelbling90]

## 2.1. 状況オートマトン (2/2)

- このアプローチの理論限界はよくわかっていない
- 命題仕様のコンパイルは NP 完全問題と等価
- より記述能力があるエージェント仕様言語は，よりコンパイルが困難である

[A] specification of the semantics of the [agent's] inputs ('whenever bit 1 is on, it is raining'); a set of static facts ('whenever it is raining, the ground is wet'); and a specification of the state transitions of the world ('if the ground is wet, it stays wet until the sun comes out'). The programmer then specifies the desired semantics for the output ('if this bit is on, the ground is wet'), and the compiler...[synthesizes] a circuit whose output will have the correct semantics. ... All that declarative 'knowledge' has been reduced to a very simple circuit. [Kaelbling91]

## 2.2. その他の包摂アーキテクチャの例

---

- PENGI [Chapman86]
- Agent Network Architecture [Maes89]

### 3. ハイブリッドアーキテクチャ (1/4)

- エージェント構築に関しては、完全に熟考的なアプローチも、また、完全に即応的なアプローチも、いずれも適していないことを多くの研究者が指摘する
- ハイブリッドシステム (古典的アプローチと他のアプローチの融合) の使用が示唆されている
- 明白なアプローチは2つ (もしくはそれより多くの) のサブシステムからエージェントを構築すること
  - **熟考部**: 記号的な世界モデルを含み、記号的 AI で提案された方法でプランを生成し、意志決定を行う
  - **即応部**: 複雑な推論なしにイベントに即応する

### 3. ハイブリッドアーキテクチャ (2/4)

---

- しばしば，即応部は熟考部に対してある種の優先度を持つ
- この種の構成は自然と階層的アーキテクチャにつながる．  
TouringMachines[Ferguson92] や InterRRaP[Fischer95] がその事例
- そのようなアーキテクチャでは，エージェント制御サブシステムは階層をなし，上位層 (レイヤ) はより抽象性の高いレベルの情報を扱う

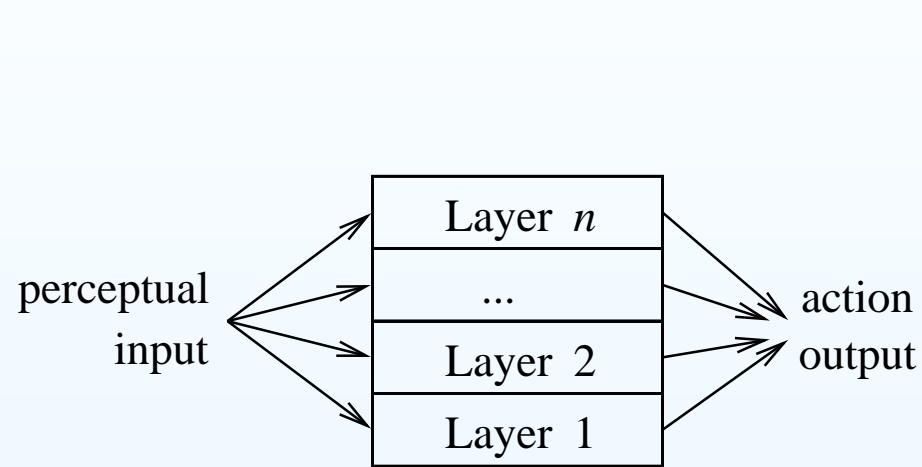


### 3. ハイブリッドアーキテクチャ (3/4)

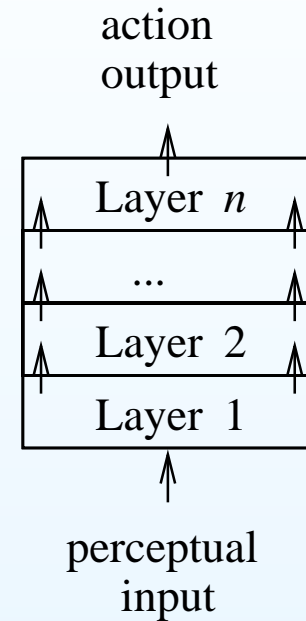
---

- そのようなアーキテクチャにおける重要な問題は，エージェントのサブシステムにどのような制御構造を埋め込むか，そして，レイヤ間の相互作用をどう管理するかである
- **水平層化**: 各レイヤは直接センサ入力と行為出力に接続される
- **垂直層化**: 直接センサ入力と行為出力は最大でも一つのレイヤで処理される

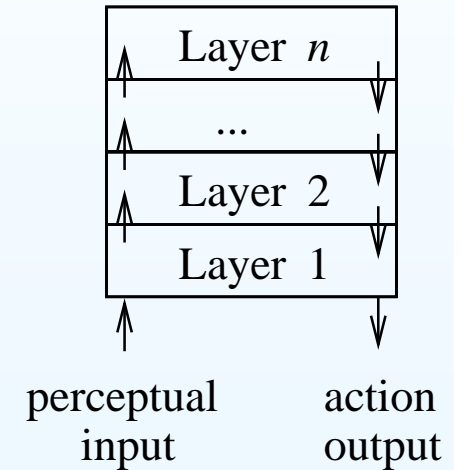
### 3. ハイブリッドアーキテクチャ (4/4)



(a) Horizontal layering



(b) Vertical layering  
(One pass control)



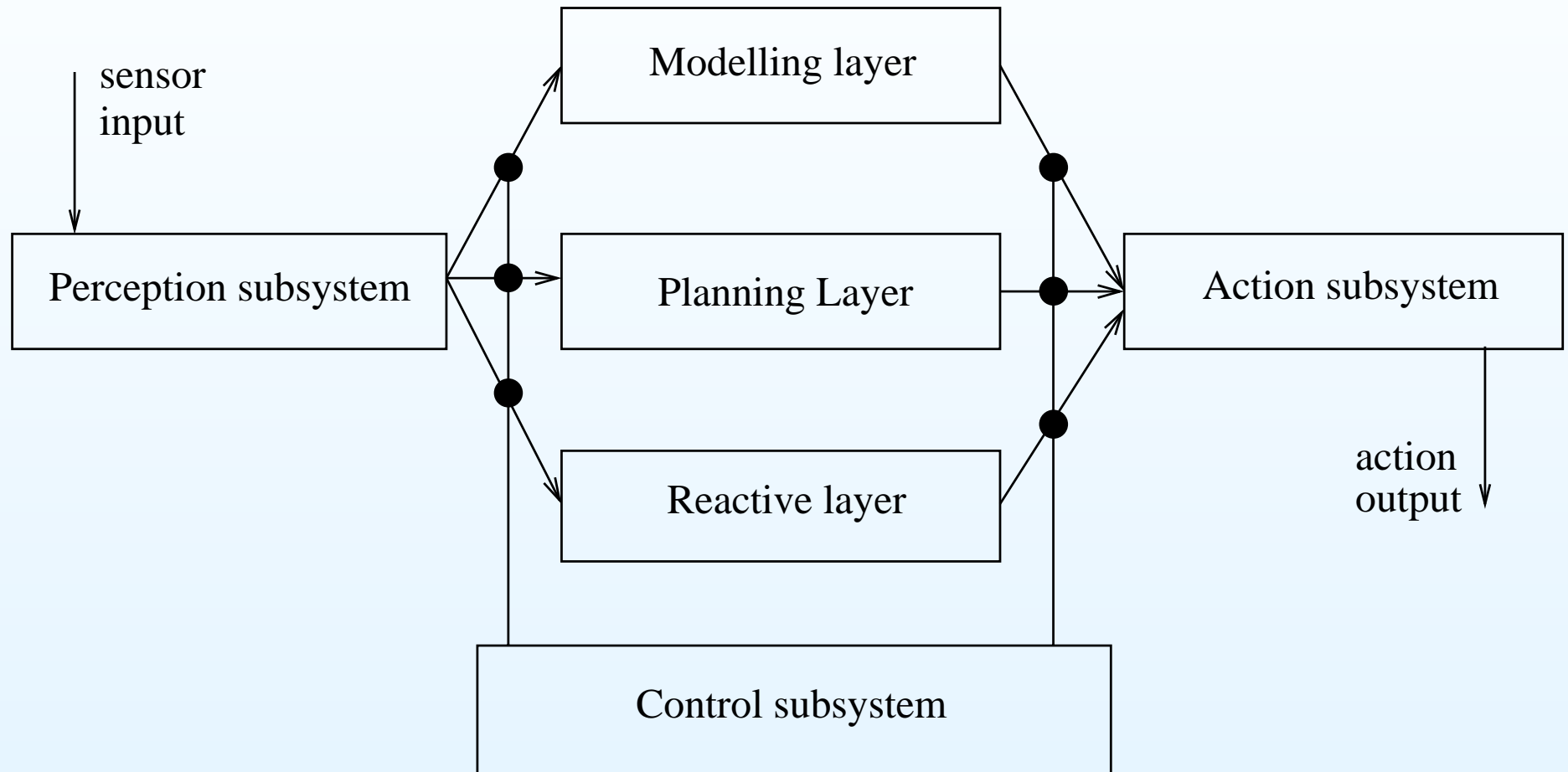
(c) Vertical layering  
(Two pass control)

## 3.1. TouringMachines (1/4)

---

- TouringMachine アーキテクチャ [Ferguson92] は知覚サブシステムと行為サブシステム (これらのサブシステムはエージェントの環境と直接のインタフェースとなる) そして3つの制御層から構成される.

## 3.1. TouringMachines (2/4)



## 3.1. TouringMachines (3/4)

- **即応層**は状況-行為規則の集合により実装されている。  
(例)

```
rule-1: kerb-avoidance
  if
    is-in-front(Kerb, Observer) and
    speed(Observer) > 0 and
    separation(Kerb, Observer) <
      KerbThreshold
  then
    change-orientation(KerbAvoidanceAngle)
```

- **プラン層**はプランを生成し, エージェントの目標を達成するために実行する行為を選択する

## 3.1. TouringMachines (4/4)

- **モデル化層**はエージェントの環境における実体 (entity) の認知的状態に関する記号表現を含んでいる
- 3つの層は互いに通信を行い, またそれは**制御規則**を用いる制御サブシステムに埋め込まれている.

(例)

```
sensor-rule-1:  
  if  
    entity(obstacle-6) in perception-buffer  
  then  
    remove-sensory-record(layer-R,  
                           entity(obstacle-6))
```

## 参考文献 (1/2)

- Brooks91a** Brooks, R. A. (1991a) Intelligence without reason. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI-91), Sydney, Australia*, pp. 569–595.
- Brooks91b** Brooks, R. A. (1991b) Intelligence without representation. *Artificial Intelligence*, **47**, 139–159.
- Steels90** Steels, L. (1990) Cooperation between distributed agents through self organization. In *Decentralized AI – Proceedings of the 1st European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW-89)* (eds Y. Demazeau and J.-P. Müller), pp. 175–196. Elsevier, Amsterdam.
- Rosenschein86** Rosenschein, S. and Kaelbling, L. P. (1986) The synthesis of digital machines with provable epistemic properties. In *Proceedings of the 1986 Conference on Theoretical Aspects of Reasoning About Knowledge* (ed. J. Y. Halpern), pp. 83–98. Morgan Kaufmann, San Mateo, CA.
- Kaelbling90** Kaelbling, L. P. and Rosenschein, S. J. (1990) Action and planning in embedded agents. In *Designing Autonomous Agents* (ed. P. Maes), pp. 35–48. MIT Press, Cambridge, MA.

## 参考文献 (2/2)

- Kaelbling91** Kaelbling, L. P. (1991) A situated automata approach to the design of embedded agents. *SIGART Bulletin*, 2(4), 85–88.
- Ferguson92** Ferguson, I. A. (1992b) Towards an architecture for adaptive, rational, mobile agents. In *Decentralized AI 3 – Proceedings of the 3rd European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW-91)* (eds E. Werner and Y. Demazeau), pp. 249–262. Elsevier, Amsterdam.
- Fischer95** Fischer, K., Müller, J. P. and Pischel, M. (1996) A pragmatic BDI architecture. In *Intelligent Agents, II* (eds M. Wooldridge, J. P. Müller and M. Tambe), LNAI Volume 1037, pp. 203–218. Springer, Berlin.
- Chapman86** Chapman, D. and Agre, P. (1986) Abstract reasoning as emergent from concrete activity. In *Reasoning About Actions and Plans – Proceedings of the 1986 Workshop* (eds M. P. Georgeff and A. L. Lansky), pp. 411–424. Morgan Kaufmann, San Mateo, CA.
- Maes89a** Maes, P. (1989) The dynamics of action selection. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence (IJCAI-89), Detroit, MI*, pp. 991–997.