Agents et Systèmes Multi-agents

Adina Magda Florea

Professeur à l'Université "Politehnica" de Bucarest

adina@cs.pub.ro

URL du cours: **http://turing.cs.pub.ro/auf2/**

Architectures des agents

Plan

- La rationalité des agents
- Structure conceptuelle des agents
- Architectures des agents intelligents
  - Architecture BDI
  - Architecture réactive
  - Architecture hybride
- Programmation orientée agents

1. La rationalité des agents

- Un **agent rationnel** est un agent qui agit d'une manière lui permettant d'obtenir le plus de succès possible dans la réalisation des tâches qu'on lui a assignées.
- *Comment peut-on mesurer le succès d'un agent "rationnel"?*
- Une **mesure de performance**, si possible objective, associée à une certaine tâche que l'agent doit exécuter.

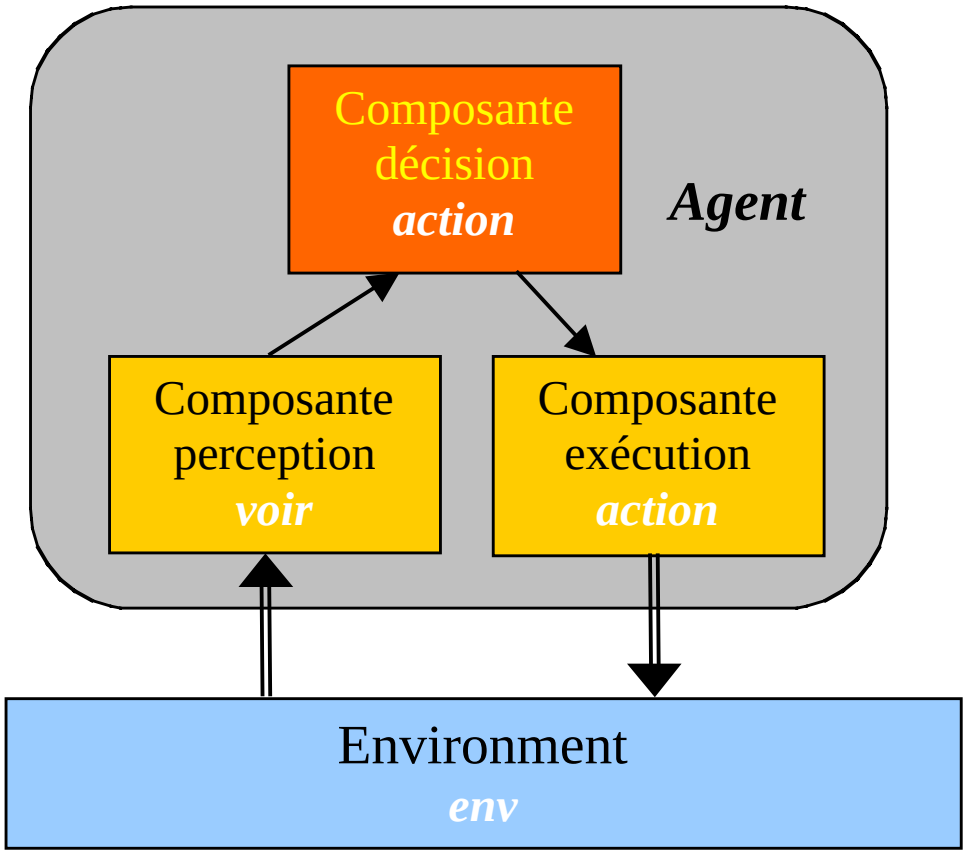
2. Structure conceptuelle des agents

- Un agent est situé dans un environnement.

$E=\{e_1, .., e, ..\}$

- Environnement
  - accessible ou inaccessible
  - déterministe ou non déterministe
  - statique ou dynamique
  - discret ou continu

Modélisation des agents



Agent réactif

voir **: E → P**

action **: P → A**

env **: E x A → P(E)**

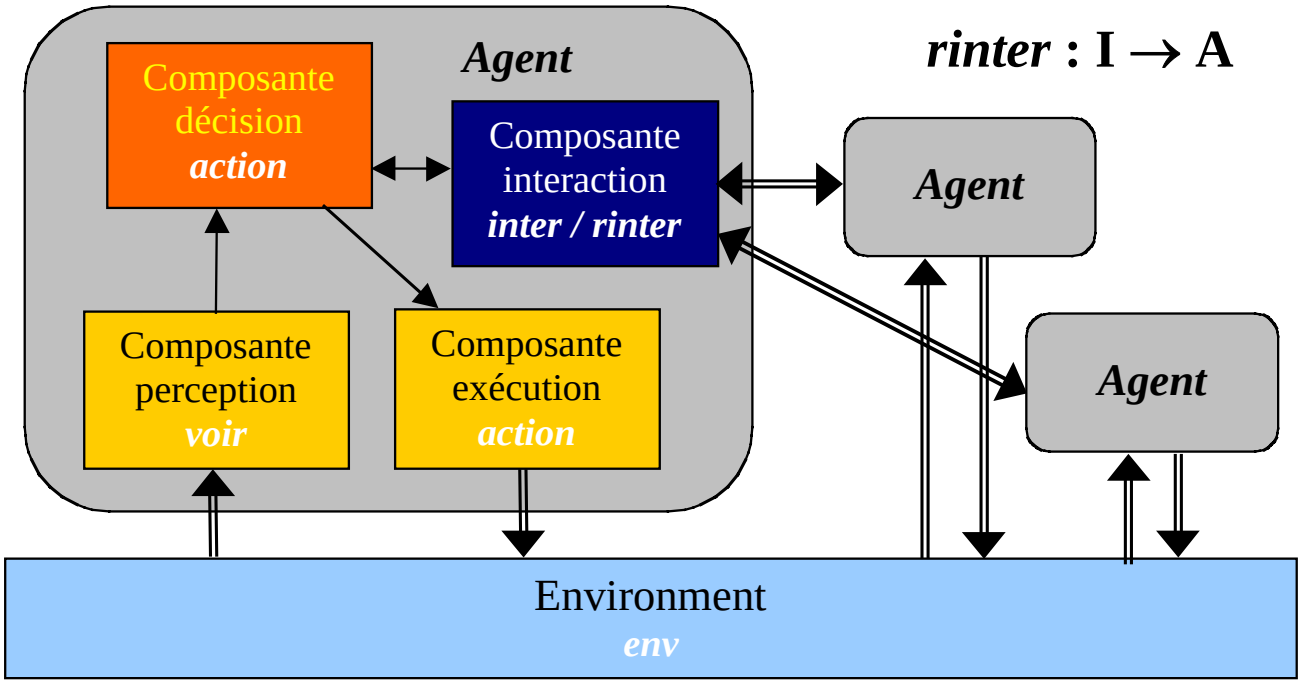
Modélisation des agents

Plusieurs agents réactifs

$env : E \times A_1 \times \ldots A_n \rightarrow P(E)$

$inter : P \rightarrow I$

$rinter : I \rightarrow A$



Modélisation des agents

Agents cognitifs

Agents avec états

- ***action*** **: S → A**
- ***suiv*** **: S x P → S**
- ***inter*** **: S x P → I**
- ***rinter*** **: S x I → A**
- ***voir*** **: E → P**
- ***env*** **: E x A → P(E)**

Modélisation des agents

Agents avec buts

but **: E → {0, 1}**

Agents avec utilité

utilité : E → R

Environnement non déterministe ***env*** **: E x A → P(E)**

la probabilité estimée par l'agent que le résultat de l'exécution de l'action a dans l'état e sera l'état e'

$$\sum_{e' \in env(e, a)} prob(ex(a, e) = e') = 1$$

Modélisation des agents

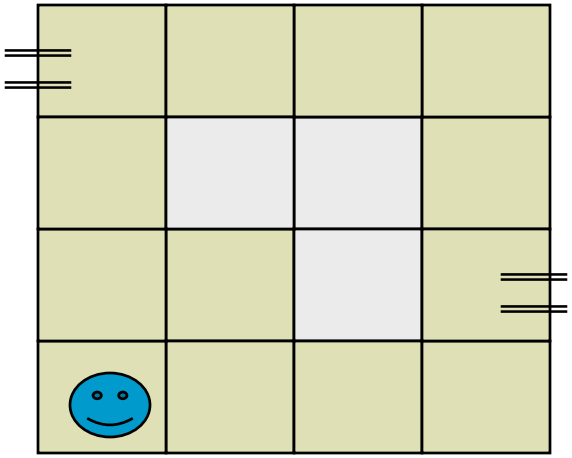
- **L'utilité attendue d'une action dans l'état e du point de vue de l'agent**

$$U(a,e) = \sum_{e' \in env(e,a)} prob(ex(a,e) = e') * utilite(e')$$

- **Le principe de la plus grande utilité attendue - utilité attendue maximale, en anglais *Maximum Expected Utility* (MEU) - dit qu'un agent rationnel devrait choisir l'action qui lui apporte la plus grande utilité attendue.**

Comment modéliser ce probleme

- Agents réactifs
- Agents avec états
- Agents avec buts
- Agents avec utilité



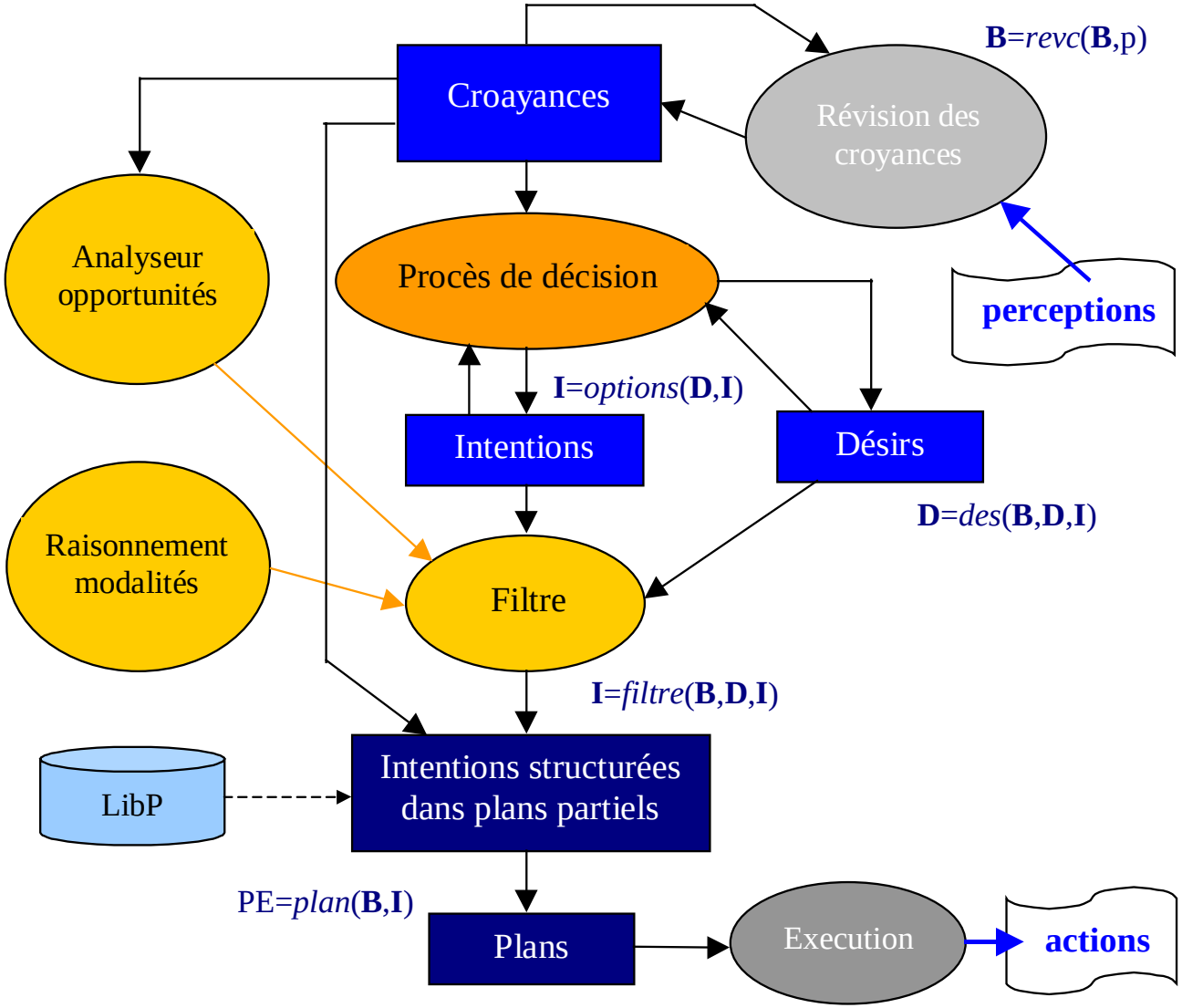
3. Architectures des agents intelligents

3.1 Architecture BDI

Modèle "Croyance-Désir-Intention" de la rationalité d'un agent intelligent

- **Le B = Belief = Croyance**
- Les croyances d'un agent sont les informations que l'agent possède sur l'environnement et sur d'autres agents qui existent dans le même environnement.
- **Le D = Desire = Désir**
- Les désirs d'un agent représentent les états de l'environnement, et parfois de lui-même, que l'agent aimerait voir réalisés.
- **Le I = Intention = Intention**
- Les intentions d'un agent sont les désirs que l'agent a décidés d'accomplir ou les actions qu'il a décidé de faire pour accomplir ses désirs.

Architecture BDI



revc : B x P → B est la fonction de révision des croyances de l'agent lorsqu'il reçoit de nouvelles perceptions sur l'environnement, où P représente l'ensemble des perceptions de l'agent; elle est réalisée par la composante Révision des croyances ;

options : D x I → I est la fonction qui représente le processus de décision de l'agent prenant en compte ses désirs et ses intentions courantes; cette fonction est réalisée par la composante Processus de décision ;

des : B x D x I → D est la fonction qui peut changer les désirs d'un agent si ses croyances ou intentions changent, pour maintenir la consistance des désirs de l'agent (on suppose dans notre modèle que l'agent a toujours des désirs consistants) ; cette fonction est également réalisée par la composante Processus de décision ;

filtre : B x D x I → I est la fonction la plus importante car elle décide des intentions à poursuivre; elle est réalisée par la composante Filtre.

***plan* : B x I → PE** est la fonction qui transforme les plans partiels en plans exécutables, **PE** étant l'ensemble de ces plans ; elle peut utiliser, par exemple, une bibliothèque de plans, représentée par le module **LibP** dans la figure.

- Un **plan** est une séquence d'actions à exécuter dans le temps

Soient $\mathbf{B_0}$, $\mathbf{D_0}$ et $\mathbf{I_0}$ les croyances, désirs et intentions initiales de l'agent.

Algorithme de contrôle d'agent BDI

1 $\mathbf{B} \leftarrow \mathbf{B_0}$

2 $\mathbf{D} \leftarrow \mathbf{D_0}$

3 $\mathbf{I} \leftarrow \mathbf{I_0}$

4 **répéter**

4.1 obtenir nouvelles perceptions **p**

4.2 $\mathbf{B} \leftarrow$ ***revc***(**B**, **p**)

4.3 $\mathbf{I} \leftarrow$ ***options***(**D**, **I**)

4.4 $\mathbf{D} \leftarrow$ ***des***(**B**, **D** ,**I**)

4.5 $\mathbf{I} \leftarrow$ ***filtre***(**B**, **D**, **I**)

4.6 $\mathbf{PE} \leftarrow$ ***plan***(**B**, **I**)

4.7 exécuter(**PE)**

jusqu'à ce que l'agent soit arrêté

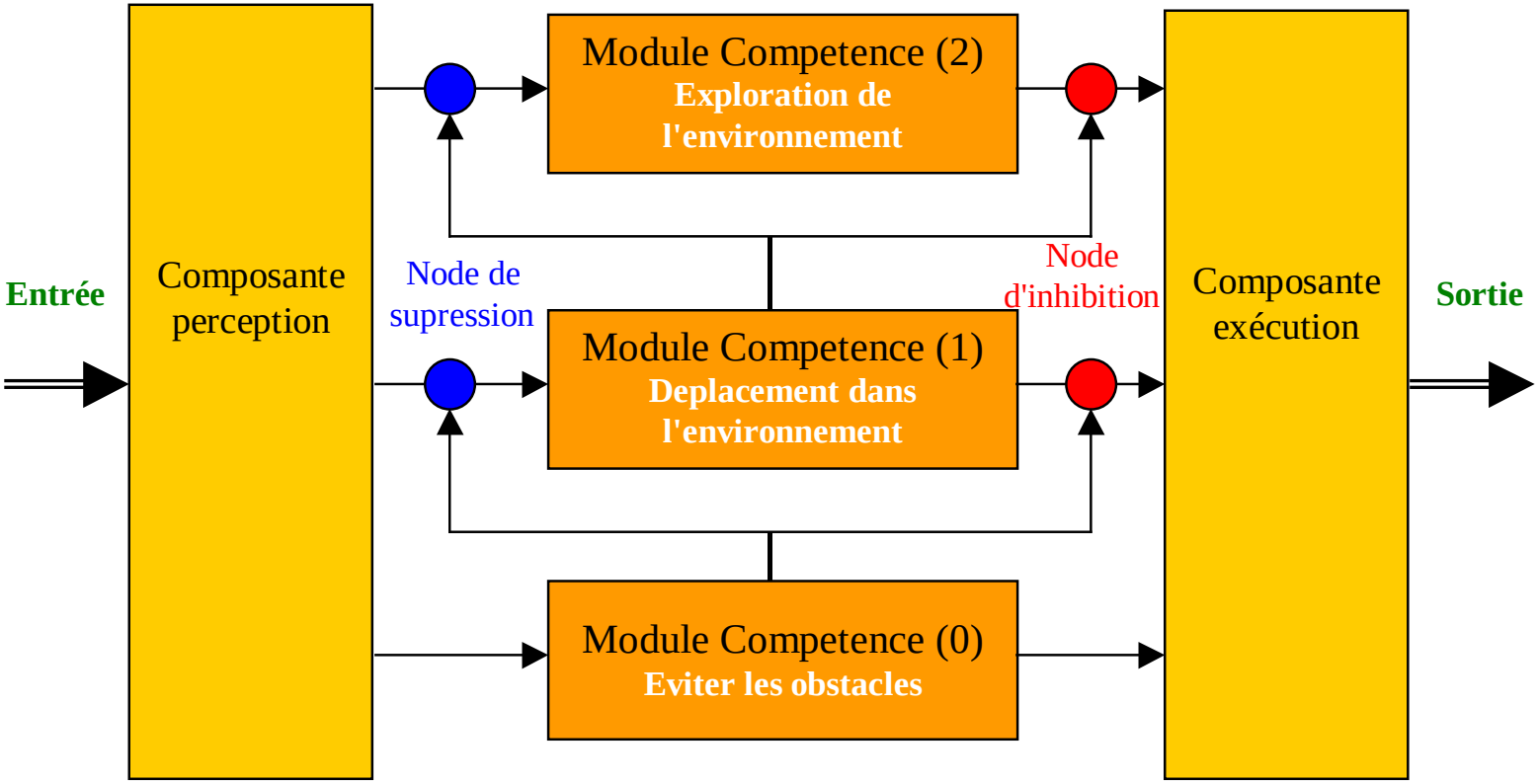
fin

Stratégie d'obligation

- **Obligation aveugle.**Un agent suivant cette stratégie va maintenir ses intentions jusqu'à ce qu'elles soient réalisées, plus précisément jusqu'à ce qu'il croie qu'elles sont réalisées.
- **Obligation limitée**. Cette stratégie dit que l'agent va maintenir ses intentions ou bien jusqu'à ce qu'elles soient réalisées ou bien jusqu'à ce qu'il croie qu'elles ne sont plus réalisables.
- **Obligation ouverte**. Un agent ayant une stratégie d'obligation ouverte maintient ses intentions tant que ces intentions sont aussi ses désirs; une fois que l'agent a conclu que ses intentions ne sont plus réalisables, il ne les considère plus parmi ses désirs.

3.2 Architecture réactive

Architecture réactive de subsomption



- Le fonctionnement de l'agent - règles de comportement, "behaviour rules" en anglais.
- Une **règle de comportement** est semblable à une règle de production et elle a deux parties: une condition **c** et une action **a**.
- L'ensemble de règles de comportement

  Comp = {(**c**, **a**) | **c**∈**P**, **a**∈**A**}
- Chaque module de compétence a associé un sous-ensemble des règles de comportement spécifiques à ses compétences.
- **Relation d'ordre** ou relation inhibitrice totale sur l'ensemble **Comp**, ∠: **Copm** x **Comp** qui établis la priorité des modules de compétence.
- **R -** l'ensembles des règles de comportement qui peuvent être exécutées à un certain moment
- Les règles qui sont effectivement exécutés sont

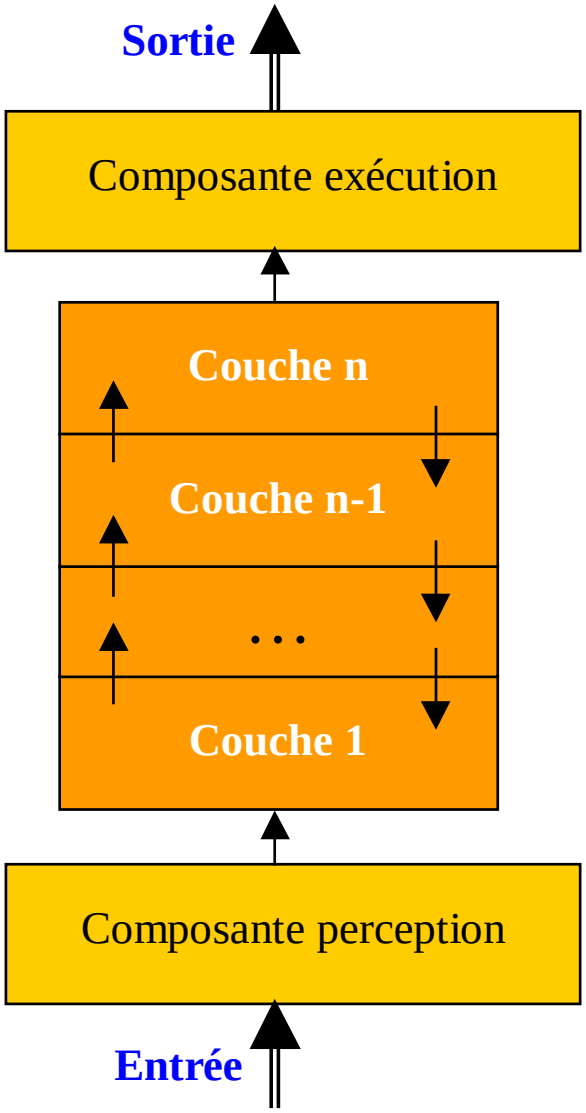
  {(**c**, **a**)| ¬∃(**c'**, **a'**) tel que (**c'**, **a'**) ∠(**c**, **a**)}

3.3. Architecture hybride

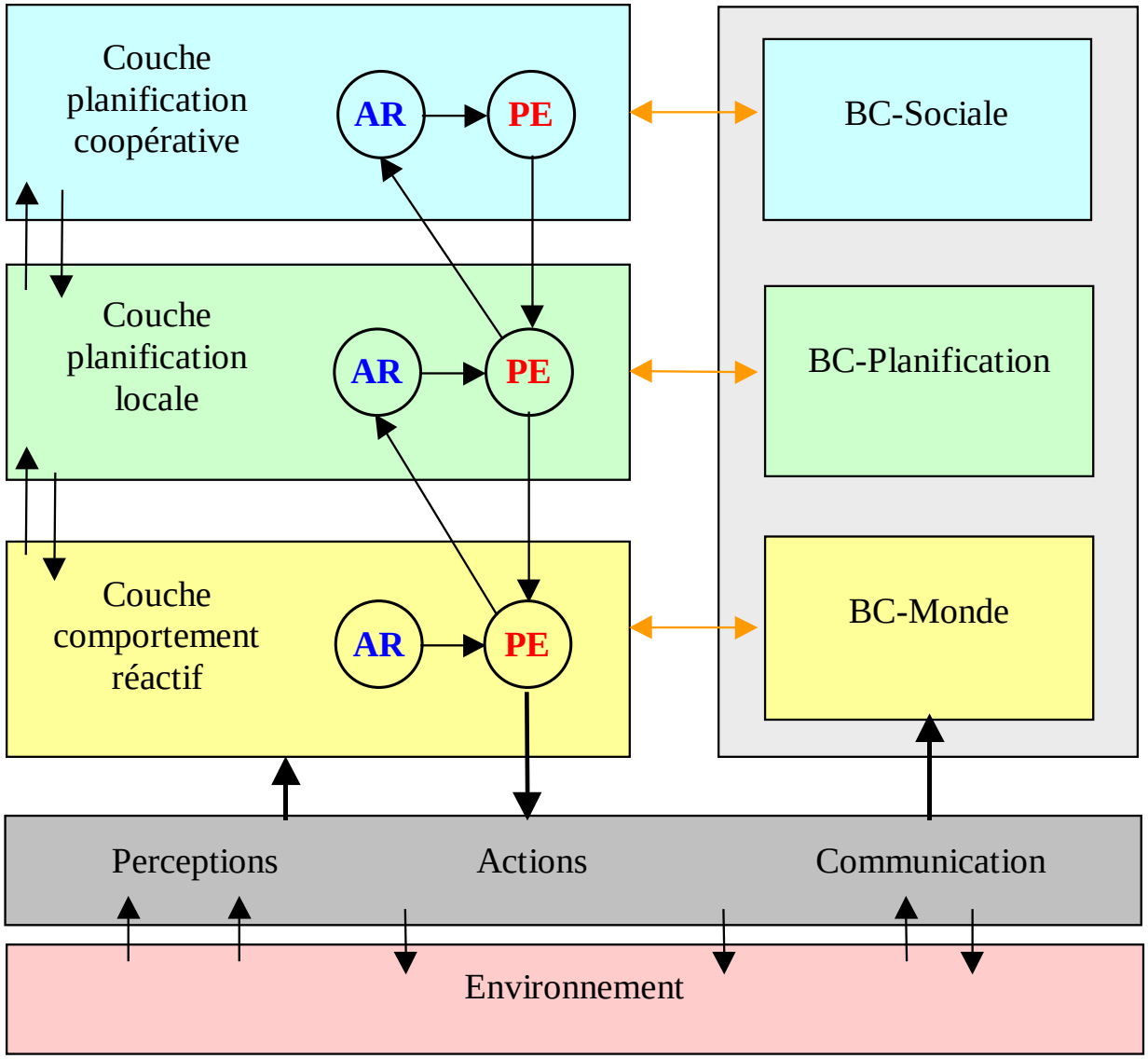
- **Système InteRRaP** ("Integration of Reactive Behavior and Rational Planning" = Intégration du comportement réactif et planification rationnelle).

- L'architecture InteRRaP est une architecture en couches avec des couches verticales où les données d'entrée, notamment les perceptions, passent d'une couche à l'autre.

Architecture hybride
en couches verticales

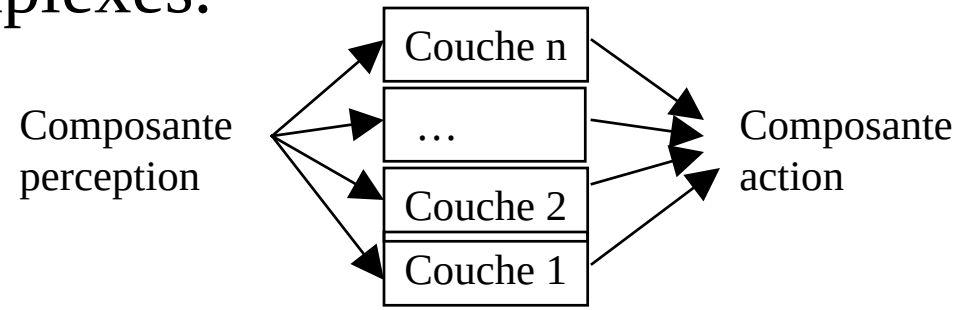


Architecture
InteRRaP



Architecture TuringMachine

- Le système TuringMachine.
- L'architecture du système est une architecture en couches horizontales, semblable à l'organisation de l'architecture de subsomption, où chaque couche est responsable à exécuter des actions de plus en plus complexes.



Architecture horizontale

4. Programmation orientée agents

	POO	POA
Unité de base	**objet**	**agent**
Paramètres définissant l'état de l'unité de base	**pas de contraintes**	**croyances, décisions, obligations, habilités**
Procès de calcul	**envoi des messages et méthodes pour la réponse**	**envoi des messages et méthodes pour la réponse**
Types de messages	**pas de contraintes**	**informer, demander, offrir, promettre, accepter, rejeter, …**
Contraintes sur les méthodes	**pas de contraintes**	**consistance, vérité, …**

Programmation Orientée Objets
versus
Programmation Orientée Agents

AGENT0 – Yoav Shoham

D'autres différences entre POO et POA

- les agents sont autonomes alors que les objets ne le sont pas; un agent va décider par son propre procès de décision s'il exécute ou pas une action requise;
- les agent ont leurs propres buts et ils agissent d'une manière pro-active pour atteindre leurs buts (par exemple saisissent de opportunités) alors que les objets ne le font pas;
- les agents sont capable d'un comportement social et ils peuvent s’engager dans des interactions complexes, par exemple coopération, compétition, négociation, avec d'autres agents; ce n'est pas le cas des objets;
- un système multi-agents est, normalement, un système où les agent correspondent à des fils d’exécution séparés; chaque agent a son propre fil d’exécution alors que les objets, à part les objets concurrents, ne présentent pas cette caractéristique.