

Agents Intelligents (Licence 3 IA)

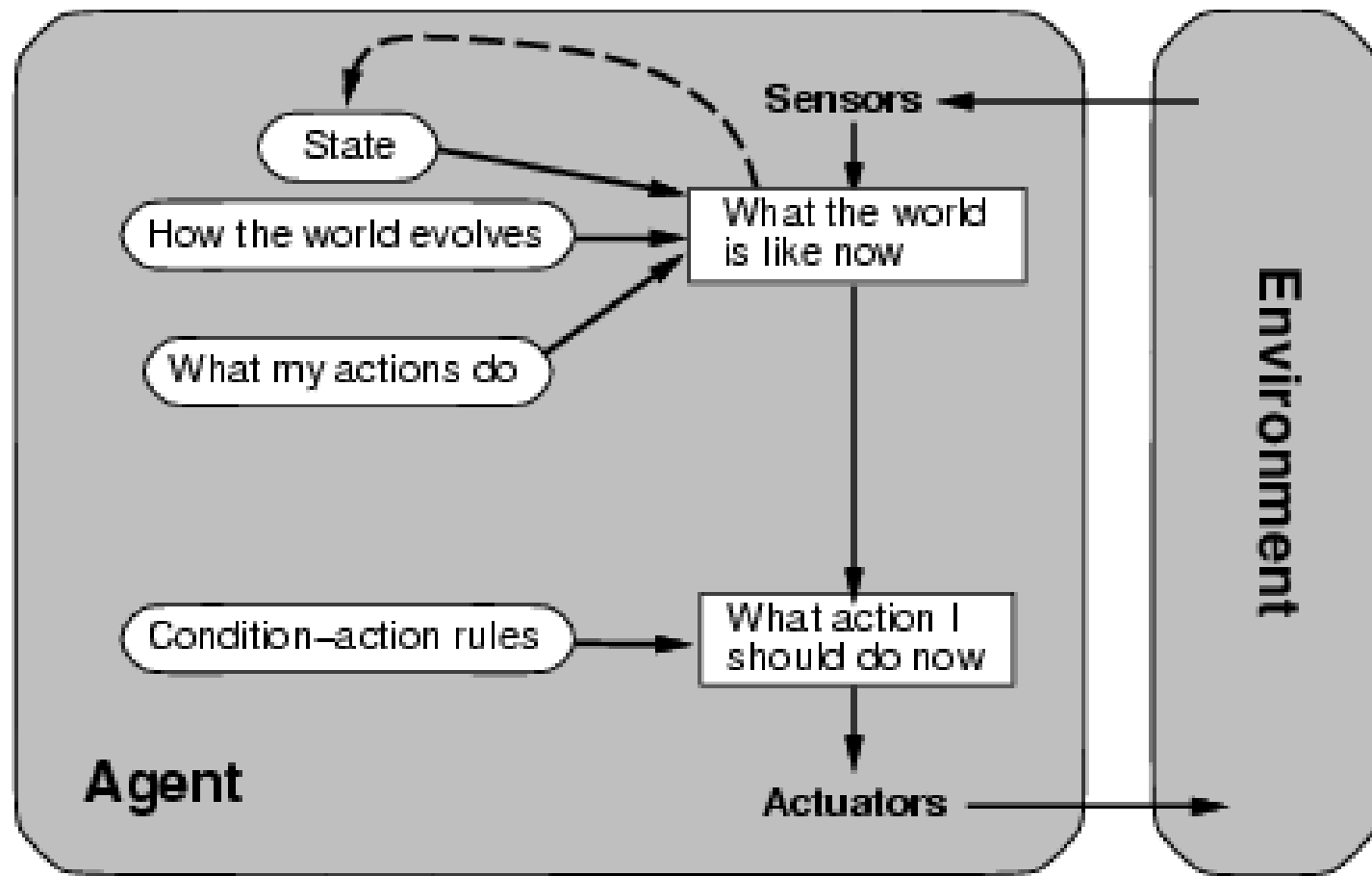
Andrea G. B. Tettamanzi
Laboratoire I3S – Équipe SPARKS
`andrea.tettamanzi@univ-cotedazur.fr`



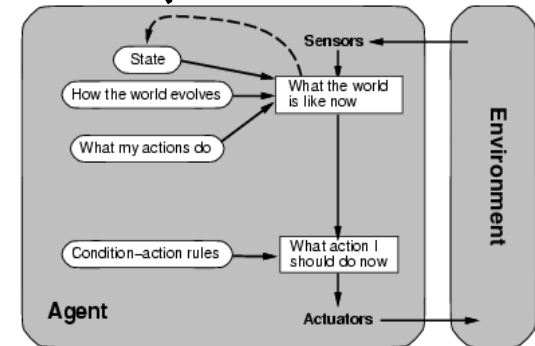
Agents réflexes fondés sur les modèles (ou avec état interne)

- Gérer l'observabilité partielle.
- Doté d'un modèle qui lui permet de suivre l'évolution de la partie du monde qu'il observe au moment présent.
- En utilisant un modèle interne, il choisit une action, de la même manière que l'agent réflexe.
- Ex. aspirateur avec état interne : l'agent a une ``mémoire'' pour enregistrer l'information indiquant qu'il a changé de pièce ($A \rightarrow B$).

Agents réflexes fondés sur les modèles (ou avec état interne)



Agents réflexes fondés sur les modèles (ou avec état interne)



- Comment le monde est maintenant ?
- État ? ``état interne'' décrivant les perceptions passées.
- Comment le monde évolue ? Si X est en A et va à droite alors il se trouvera en B.
- Quel est l'impact de mes actions ? Si je suis en A et je nettoie alors A sera propre.

Agents réflexes fondés sur un modèle

Fonction AGENT-RÉFLEXE-FONDÉ-SUR-MODÈLE(*Perception*) **retourne** une action

Variables statiques :

État : l'idée que l'agent se fait sur l'état courant du monde

Modèle : une description de la façon dont l'état suivant dépend de l'état courant et de l'action

Règles : un ensemble de règles *condition-action*

Action : l'action la plus récente, initialement nulle

État \leftarrow ACTUALISER-ÉTAT(*État*, *Action*, *Perception*, *Modèle*)

Règle \leftarrow TROUVER-RÈGLE(*État*, *Règles*)

Action \leftarrow ACTION-ASSOCIÉE-REGLE[*Règle*]

Retourner *Action*

Agents réflexes fondés sur un modèle

Fonction AGENT-RÉFLEXE-FONDÉ-SUR-MODÈLE(*Perception*) **retourne** une action

Variables statiques :

État : l'idée que l'agent se fait sur l'état courant du monde

Modèle : une description de la façon dont l'état suivant dépend de l'état courant et de l'action

Règles : un ensemble de règles condition-action

Action : l'action la plus récente, initialement nulle

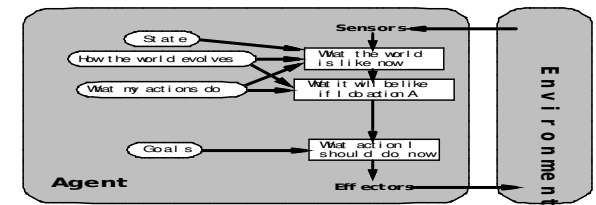
$\text{État} \leftarrow \text{ACTUALISER-ÉTAT}(\text{État}, \text{Action}, \text{Perception}, \text{Modèle})$

$\text{Règle} \leftarrow \text{TROUVER-RÈGLE}(\text{État}, \text{Règles})$

$\text{Action} \leftarrow \text{ACTION-ASSOCIÉE-REGLE}[\text{Règle}]$

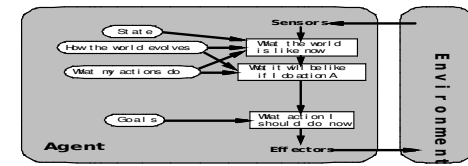
Retourner *Action*

Agents réflexes fondés sur un modèle et des buts

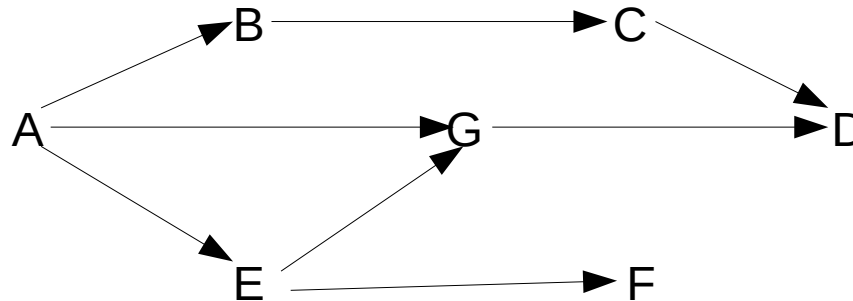


- Connaître l'état actuel du monde ne suffit pas toujours pour décider quoi faire.
- Étendre la représentation des agents avec état interne.
- Outre la connaissance de l'état courant (percept + état interne), l'agent a besoin d'informations concernant le but.
- Ajout de buts (état souhaité du monde).

Agents réflexes fondés sur un modèle et des buts

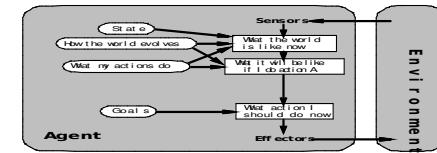


- Ex. conducteur de taxi dans une intersection. Peut tourner à gauche, droite ou continuer tout droit. La décision appropriée dépend de la destination à atteindre (son but).
- Plusieurs actions peuvent être nécessaires (la **planification** en IA).
- Ex. Trois trajectoires (ou plans) possibles pour aller de A à D :



- **But = Être en D (ou le plus proche possible de D)**
- Quelle(s) action(s) permet(tent) au taxi/aspirateur de s'approcher le plus du but ?

Agents réflexes fondés sur un modèle et des buts



Fonction GOAL-BASED-AGENT(Perception) retourne une action

Variables statiques :

État : l'idée que l'agent se fait sur l'état courant du monde

But : une description de l'état souhaité par l'agent

Règles : un ensemble de règles *condition-action*

Action : l'action la plus récente, initialement nulle

État \leftarrow UPDATE-STATE (État, Perception)

Règle \leftarrow RULE-MATCH (État, Règles)

Action \leftarrow Rule_action(Règle)

État \leftarrow UPDATE-STATE (État, Action)

if (État in But) then

 return (Action)

else

 Perception \leftarrow Obtain_Percept(État, But)

 return(GOAL-BASED-AGENT(Perception))

Agents réflexes fondés sur un modèle et des buts

Fonction GOAL-BASED-AGENT(Percept) retourne une action

Variables statiques :

État : l'idée que l'agent se fait sur l'état courant du monde

But : une description de l'état souhaité par l'agent

Règles : un ensemble de règles *condition-action*

Action : l'action la plus récente, initialement nulle

État \leftarrow UPDATE-STATE (État, Percept)

Règle \leftarrow RULE-MATCH (État, Règles)

Action \leftarrow Rule_action(Règle)

État \leftarrow UPDATE-STATE (État, Action)

if (État in But) then

 return (Action)

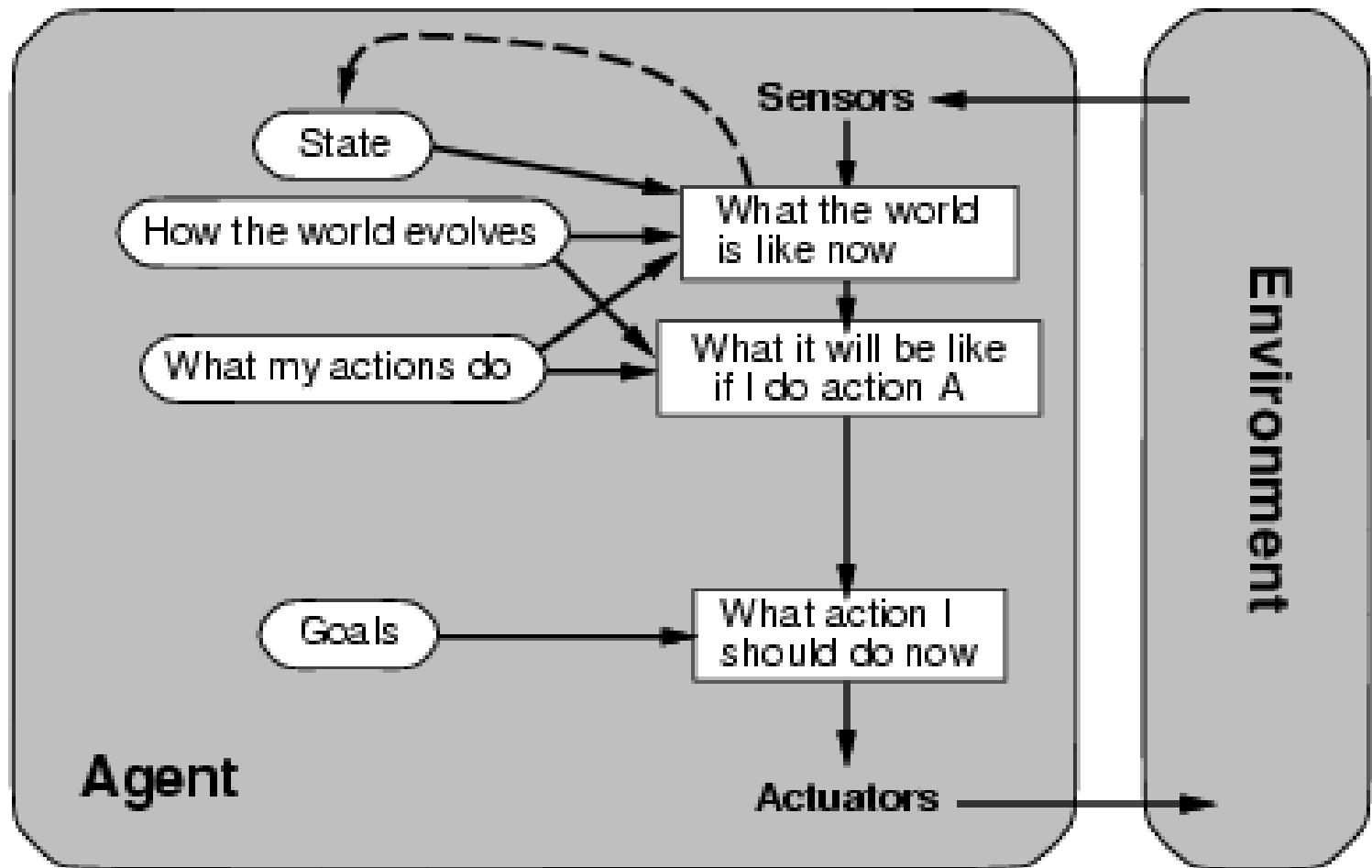
else

 Percept \leftarrow Obtain_Percept(État, But)

 return(Goal-Based-Agent(Percept))

Agents réflexes

fondés sur un modèle et des buts



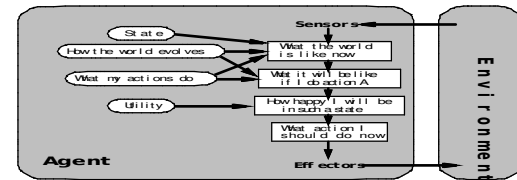
Agents réflexes fondés sur état interne+but+utilité

Il ne suffit pas d'avoir des buts, car souvent nous pouvons avoir plusieurs actions qui satisfont toutes notre but, et nous devons donc trouver un moyen d'exécuter la plus efficace d'entre elles.

Une fonction d'utilité fait correspondre un état à un nombre réel représentant dans quelle mesure on s'approche du but.

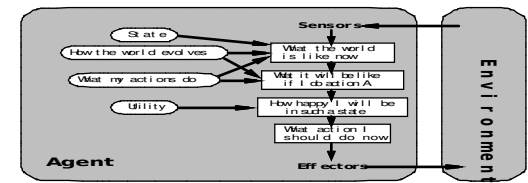
Cette fonction est aussi utile lorsque nous avons plusieurs buts qui peuvent être satisfaits et que nous devons choisir une action pour satisfaire l'un d'entre eux.

Agents réflexes fondés sur état interne+but+utilité



- Ex. conducteur de taxi. Son but est d'amener le taxi à destination sans avoir de préférences sur le trajet.
- Étendre la représentation des agents avec buts où il y a distinction binaire : but *satisfait* ou *non satisfait*.
- *Mesure de performance maximale si le but est satisfait et minimale s'il ne l'est pas.*
- L'utilité = fonction permettant de mesurer le degré de satisfaction d'un état.
- Plusieurs actions/plans possibles → choix de l'action/plan avec la plus grande utilité.
- Outre la connaissance de l'état courant + but, l'agent a besoin d'une fonction qui permet de mesurer son degré de satisfaction par rapport à un état.

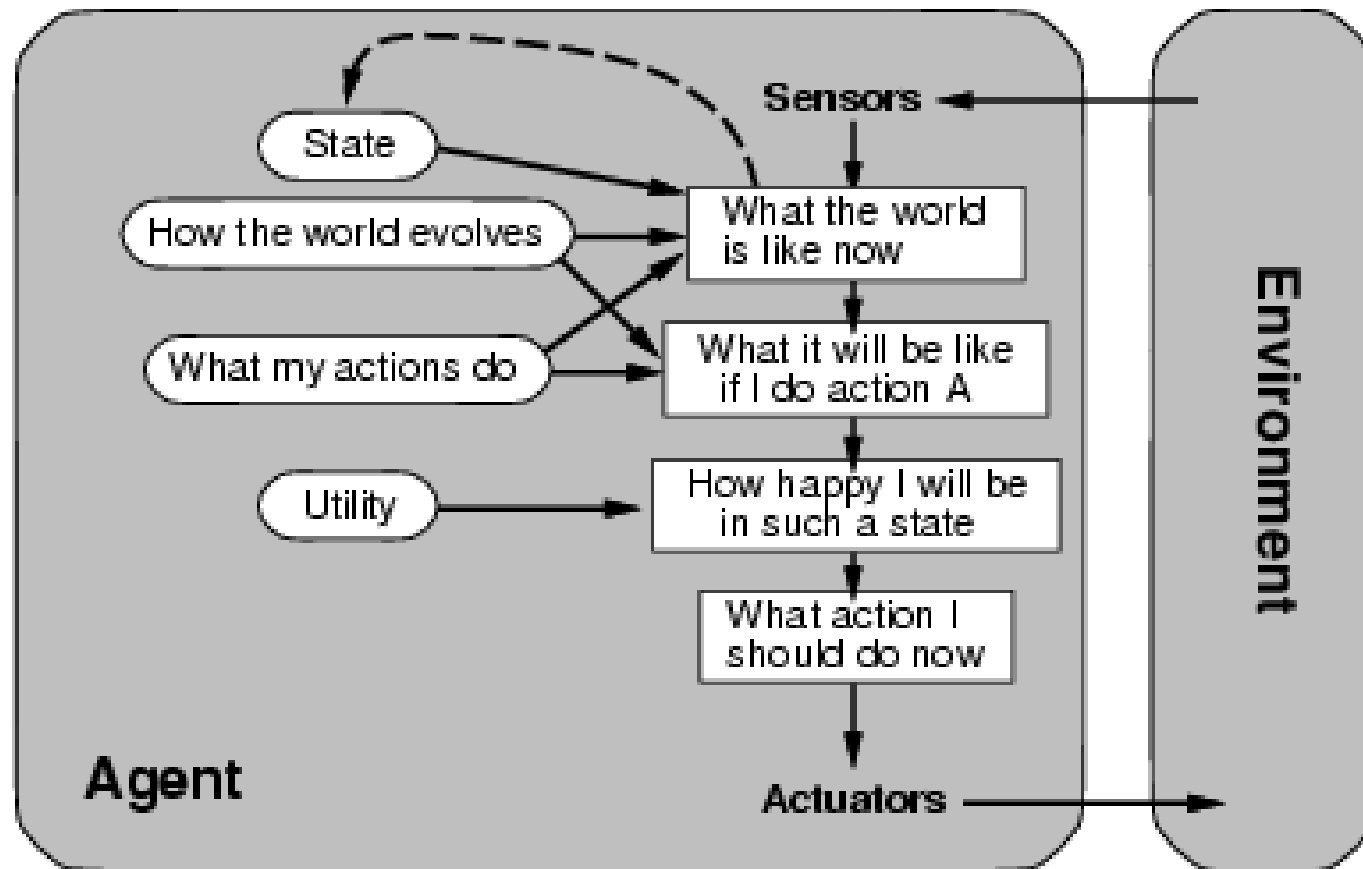
Agents réflexes fondés sur état interne+but+utilité



- Ex. conducteur de taxi (agents fondés sur le but). Son but est d'amener le taxi à destination sans avoir de préférences sur le trajet.
- Avec utilité : trajet le plus court ou avec moins de bouchons, ou avec une meilleure vue du paysage, etc.
- Ex. aspirateur avec deux trajectoires possibles pour arriver à F. Utilité : chemin le plus court ou celui qui permet de nettoyer une quantité plus grande de poussière, etc.
- Mesure de performance plus générale : permet de confronter les états en fonction de l'utilité + modèle.

Agents réflexes

fondés sur état interne+but+utilité



Agents réflexes fondés sur état interne+but+utilité

Fonction Utility-BASED-AGENT(percept) retourne une action

Variables statiques :

État : l'idée que l'agent se fait sur l'état courant du monde

But : une description de l'état souhaité par l'agent

Règles : un ensemble de règles *condition-action*

Action : l'action la plus récente, initialement nulle

État \leftarrow UPDATE-STATE (état, percept)

Règle \leftarrow RULE-MATCH (état, règles)

Action \leftarrow Rule_action(Règle)

État \leftarrow UPDATE-STATE (état, action)

Score \leftarrow Obtain_Score(état)

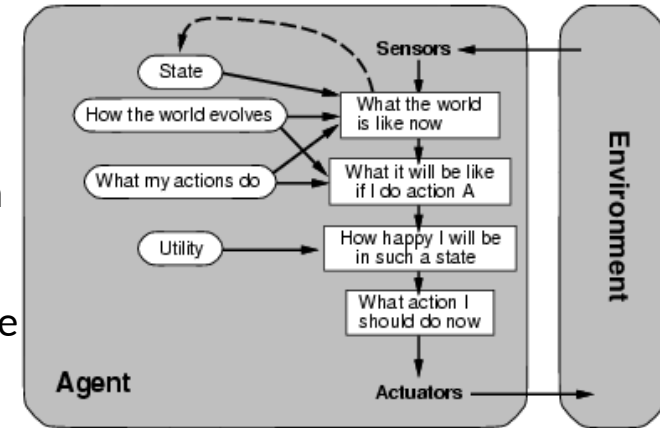
if (État in But) and Best_Score (score) then

return (Action)

else

percept \leftarrow Obtain_Percept(État, But)

return(Utility-Based-Agent(percept))



Agents réflexes fondés sur état interne+but+utilité

Fonction Utility-BASED-AGENT(percept) retourne une action

Variables statiques :

État : l'idée que l'agent se fait sur l'état courant du monde

But : une description de l'état souhaité par l'agent

Règles : un ensemble de règles *condition-action*

Action : l'action la plus récente, initialement nulle

État \leftarrow UPDATE-STATE (état, percept)

Règle \leftarrow RULE-MATCH (état, règles)

Action \leftarrow Rule_action(Règle)

État \leftarrow UPDATE-STATE (état, action)

Score \leftarrow Obtain_Score(état)

if (État in But) and Best_Score (score) then

return (Action)

else

percept \leftarrow Obtain_Percept(État, But)

return(Utility-Based-Agent(percept))

