

Bases de l'Intelligence Artificielle Distribuée

Modèles et Architectures d'agents

Mr. KHEBBACHE Mohib Eddine

2^{ème} année SDIA

30 octobre 2022

mohibeddine-khabache@univ-eloued.dz

Plan

1 Introduction

Plan

- 1 Introduction
- 2 Modèles d'agents

Plan

- 1 Introduction
- 2 Modèles d'agents
- 3 Architectures d'agents

Plan

- 1 Introduction
- 2 Modèles d'agents
- 3 Architectures d'agents

- deux éléments clés :
 - typologie (modèles) des agents \implies répondre à ces questions :
 - les agents disposent-ils d'une représentation de leur environnement ou bien ne font-ils que réagir à des stimuli qu'ils perçoivent ?
 - Les agents sont-ils capables de raisonner à partir de symboles ou bien n'appliquent-ils que des actions pré-établies ?
 - Un agent se caractérise essentiellement par son **architecture** et par son **comportement**
 - L'architecture d'un agent caractérise sa **structure interne**
 - Le comportement d'un agent caractérise sa **fonction** (liant ses perceptions à ses actions)

Plan

- 1 Introduction
- 2 Modèles d'agents
- 3 Architectures d'agents

- **On peut établir une classification des agents selon les critères suivantes :**
 - la représentation (connaissances) de l'environnement dont dispose l'agent
 - le comportement
 - le mode de communication
- généralement, entre autre, deux écoles de pensée des systèmes multi-agents : **réactifs** et **cognitifs**.

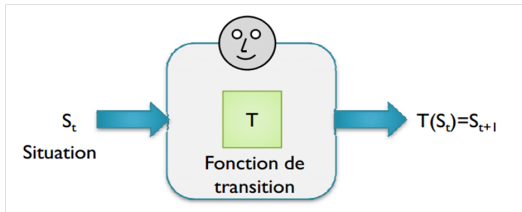
- un agent réactif ne dispose que d'une "représentation sub-symbolique" \implies limitée à ses perceptions
- comportement **dirigé par les perceptions** \implies schéma "réflexe"
 - un agent réactif ne fait que **réagir immédiatement**
 - aux **stimuli** (signaux) de l'environnement
 - suite aux changements (événements) qui surviennent dans l'environnement.
- correspond à l'école qui étudie la possibilité de l'émergence d'un comportement "intelligent" d'un **ensemble d'agents non-intelligents** (type fourmis).

- il s'agit d'acquérir des perceptions (signaux) et de **réagir** à celles-ci
 - selon schéma "stimuli/réaction" en appliquant certaines **règles pré-définies et non ordonnées de type (situation, action)**.

une fonction de transition T

Si condition alors action

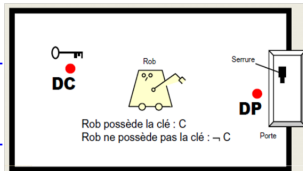
- qui sont exécutées dans une boucle infinie et non déterministe
- sans anticipation, ni planification



rules : règles condition-action
percepts : ensemble de percepts
Repeat
 percept = perceive();
 state := interpret_input(percept) ;
 rule := match(state, rules) ;
 execute(rule[action]) ;
forever

- les agents réactifs ne sont pas "intelligents" individuellement
⇒ ne peuvent atteindre leurs buts que uniquement sur coopération avec autres agents (pas d'autonomie).
- Pas de représentation explicite de leur environnement ou des autres agents.
- leur structure s'articule autour de deux (02) fonctions principales : percevoir et agir ⇒ actions situées (fourmis)
- Communication via l'environnement en propageant de signaux
⇒ l'environnement est fondamental pour la coordination des actions entre plusieurs agents réactifs
- avec leur grand nombre, les agents réactifs peuvent résoudre des problèmes complexes ⇒ modélisation et simulation d'une société d'agents
 - domaines de recherche : **la vie artificielle, de l'éthologie (la fourmilière, la termitière, la ruche d'abeille), etc..**

- Si **DP** and **PO** Alors **SORTIR**
- Si **DP** and **PF** Alors **OUVRIR**
- Si **DP** and **PFC** and **C** Alors **DE-VEROUILLER**
- Si **DP** and **PFC** Alors **RANDOM-WALK**
- Si **DC** and **C** Alors **RANDOM-WALK**
- Si **DC** and not **C** Alors **PRENDRE-CLE**



- **DP** = Devant la PortePosition
- **DC** = Devant la Clé
- **PO**: Porte ouverte :
- **PF**: Porte Fermée
- **PFC**:PFPorte Fermée à clé

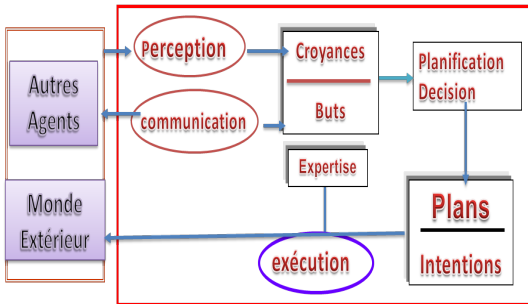
● Un garde dans un jeu

- Tant que je ne vois rien, je suis mon chemin de garde
- Si je vois un ennemi
 - S'il n'est pas menaçant et si je ne suis pas blessé, je l'attaque
 - S'il est menaçant ou si je suis blessé, je sonne l'alarme, et je m'éloigne



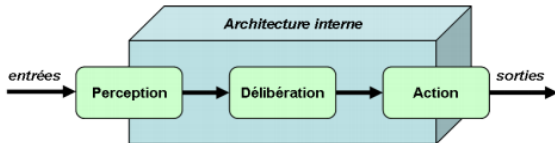
- un agent cognitif dispose d'une "représentation symbolique" de l'environnement,
- et un comportement "intentionnel" **dirigé par les buts** \implies l'agent accomplit ses tâches (rôles) en **poursuivant des buts et des plans explicites**
 - Chaque agent cognitif est doté de capacités (raisonnement, planification (décision) et communication) qui fonctionnent sur
 - une base de connaissance comprenant des
 - **connaissances de domaine** : expertises sur un domaine précis (croyances, buts, plans, intentions)
 - **connaissances de communication**
 - **connaissances de contrôle** : la gestion des interactions avec les autres agents et avec son environnement.
- correspond à l'école qui focalise sur une approche "sociologique" \implies ensemble d'agents "intelligents" et "rationnels" pouvant collaborer.

- à l'aide des mécanismes :
 - de raisonner sur des représentations de leur environnement et des autres agents (analyser les situations et anticiper sur les évènements futurs),
 - et de planifier leur propre comportement (préparer et produire des plans d'actions)
- et ainsi de sous fonctions
 - la révision des croyances,
 - la coopération (négociation, coordination),
 - l'apprentissage.
- L'agent exécute un **plan d'actions "P"** de **manière séquentielle et déterministe**, .
- \implies capacités d'anticipation et de planification permettent à un agent cognitif d'optimiser son comportement



- Les agents cognitifs sont "intelligents" individuellement \implies peuvent atteindre leurs buts sans besoin de coopérer avec autres agents (autonomie et flexibilité).
- Ils possèdent une capacité de mémorisation de lui-même, son environnement et les autres agents.
- chaque agent est spécialisé dans un domaine
- et peut communiquer (directement ou indirectement) avec les autres agents par des modes complexes.
- Les agents sont immergés dans un environnement dans lequel ils interagissent.

- leur structure s'articule autour de trois(03) fonctions principales : percevoir, décider et agir.



- petits groupes d'individus peuvent résoudre des problèmes de coopération \implies problématique d'interaction

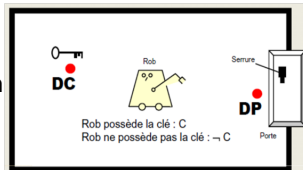
Plan ouvrirPorte

aller jusqu'à l'endroit où se trouve la
clef

prendre la clef

aller jusqu'à la porte

ouvrir la porte avec la clef



- ⊗ **DP** = Devant la PortePosition
- ⊗ **DC** = Devant la Clé
- ⊗ **PO**: Porte ouverte :
- ⊗ **PF**: Porte Fermée
- ⊗ **PFC**:PFPorte Fermée à clé

- Les agents délibératifs sont des agents qui effectuent une certaine **délibération** pour **choisir leurs actions**.
- Une telle délibération peut se faire en se basant :
 - sur **les buts de l'agent** \implies délibération fondée sur buts.
 - ou sur **une certaine fonction d'utilité** \implies délibération fondée sur fonction d'utilité,
- Une telle délibération peut prendre la forme d'un plan d'actions (séquence d'actions).



Modèles d'agents

Agent Cognitif \implies Agent délibératif (rationnels)

Modèles et
Architectures
d'agents

Mr.
KHEB-
BACHE
Mohib
Eddine

Introduction

Modèles
d'agents

Architecture
d'agents

```
State state = initialize();
```

```
while (true) {  
    percepts = perceive();  
    state = updateState(state, percepts);  
    action = deliberate(state, plans);  
    execute(action);  
};
```



- combinaison entre l'agent cognitif et l'agent réactif,
- pour pallier le compromis efficacité/complexité.
- dans des problèmes où ni une architecture complètement réactive, ni une architecture complètement délibérative ou cognitive n'est appropriée \implies systèmes hétérogènes.
 - certains agents doivent pouvoir réagir très rapidement dans certaines situations (comportement réflexe),
 - tandis que dans d'autres doivent avoir un comportement avec plus de raisonnement.

Agents cognitifs	Agents réactifs
Représentation explicite de l'environnement	Pas de représentation explicite
Peut tenir compte de son passé	Pas de mémoire locale
Agents "intelligents" individuellement	non "intelligents" individuellement
Agents complexes (BC + raisonnement+ planification)	architectures minimales (stimulus/réaction)
Communication indépendante de l'environnement	Communication par environnement
Comportement dirigé par buts	Comportement dirigé par perception
nombre d'agents réduit	nombre d'agents élevé

agent purement communicant : \implies entité informatique

- par rapport à la notion d'agent en général, Un agent purement communicant (ou agent logiciel) se distingue :
 - ne possède pas de perception des autres agents,
 - n'agit pas dans un environnement (son contexte d'évolution est naturellement celui des réseaux informatiques),
 - son comportement tient compte :
 - des ressources et des compétences dont il dispose
 - et en fonction des communications qu'il reçoit.

agent purement situé : \implies entité physique

- par rapport à la notion d'agent en général, Un agent purement situé se distingue :
 - se trouve située à une position dans un environnement,
 - capable de percevoir (mais de manière limitée) son environnement,
 - son comportement tient compte :
 - des ressources et des compétences dont il dispose
 - et en fonction des des perceptions qu'il observe.
 - peut communiquer indirectement par le biais des perceptions et de ses actions dans l'environnement.

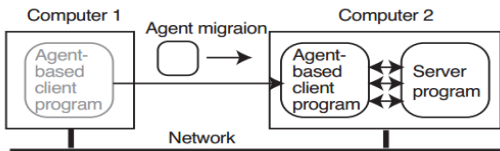
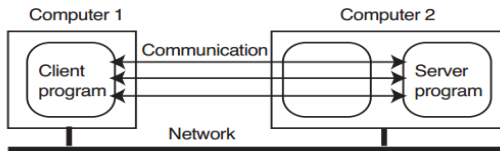
agent mobile : Introduits initialement en 1994 avec l'environnement *Telescript*

- **Le paradigme d'agents mobiles** apportent à les agents dits « **stationnaires** » la capacité de se déplacer (migration de données + code) de manière autonome sur les différents nœuds d'un réseau en travaillant localement sur les ressources disponibles :
 - tout d'abord le code et les données de l'agent mobile sont hébergés dans l'ordinateur A
 - les données et le code sont migrés à l'ordinateur B.
 - Après la migration, le code est exécuté avec les données et les ressources disponibles sur l'ordinateur B.

agent mobile : Introduits initialement en 1994 avec l'environnement *Telescript*

- Ils peuvent fournir un cadre pratique, efficace et robuste pour la mise en œuvre d'applications distribuées et des environnements intelligents tel que les grilles de calcul pour plusieurs raisons :
 - la recherche de performance
 - la facilité d'adaptabilité et de développer d'applications distribuées.
 - coûts de communication réduits
 - réduire la vulnérabilité
 - exécution asynchrone
 - déploiement dynamique de logiciels

agent mobile : Introduits initialement en 1994 avec l'environnement *Telescript*



de communication

réduction de coûts

Plan

- 1 Introduction
- 2 Modèles d'agents
- 3 Architectures d'agents

- Un agent peut toujours être vu comme une **fonction liant ses perceptions à ses actions**.
- Cette fonction est réalisée et implémentée par :
 - programme d'agent qui s'exécute sur une
 - architecture qui organise également l'interface avec l'environnement (perceptions, actions)

Agent = Architecture + Programme (comportement)

- l'architecture :
 - peut être physique (équipement de traitement avec capteurs et effecteurs "actionneurs" physiques) ou virtuelle comme dans le cas des agents logiciels
 - dispose au programme les perceptions disponibles de capteurs (comme entrée), et mis les actions choisis aux effecteurs (sorties).



- le programme d'agent met en œuvre la fonction désirée d'agent .
- Il est invoqué pour chaque nouveau percept des capteurs et retourne une action aux effecteurs.
- Les actions effectuées décrivent le comportement de l'agent en fonction de :
 - ses connaissances,
 - ses compétences,
 - ses perceptions,
 - sa représentation du monde
 - et les communications qu'il possède avec les autres ou avec son environnement.



Russel et Norvig regroupent les programmes d'agents en plusieurs types, à savoir :

- TABLE-DRIVEN "Table-lookup" agent exemple : automated taxi
- Agents réflexes simples
- Agents réflexes fondés sur des modèles (avec modèle interne du monde)
- Agents fondés sur des buts
- Agents complets fondés sur l'utilité

Les types d'agents réflexes sont considérés comme des agents réactifs et les deux derniers types sont considérés comme des agents délibératifs

Agents réflexes simples

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
⋮	⋮

```
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
```

- élément clé dans le processus de développement d'un systèmes multi-agents (phase de conception) :
 - en utilisant une méthodologie particulière pour la spécification et la description de l'organisation interne d'un agent
 - pour répondre à cette question : **comment assembler les différentes parties d'un agent de manière qu'il accom- plisse les actions à partir d'un ensemble d'entrées ?** \implies résumer le point de vue de concepteur.
 - où le choix d'une architecture dépend de :
 - décision du concepteur
 - la manière dont les perceptions sont liées aux actions (types d'agents).
- Pour faire face aux problématiques des SMA en décrivant quelques éléments essentiels de l'architecture d'un système multi-agent.

Architecture d'un agent

Structure générale d'un agent(mono)

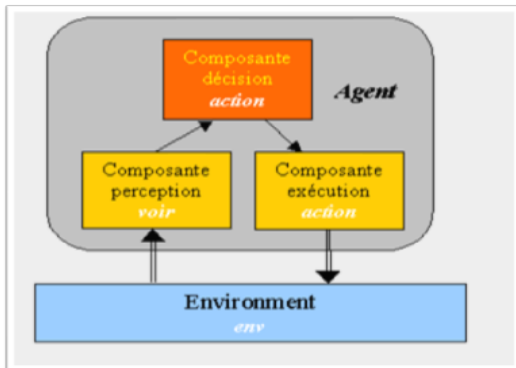
Modèles et
Architectures
d'agents

Mr.
KHEB-
BACHE
Mohib
Eddine

Introduction

Modèles
d'agents

Architecture
d'agents



Architecture d'un agent

Structure générale d'un agent en interaction (communication via env)

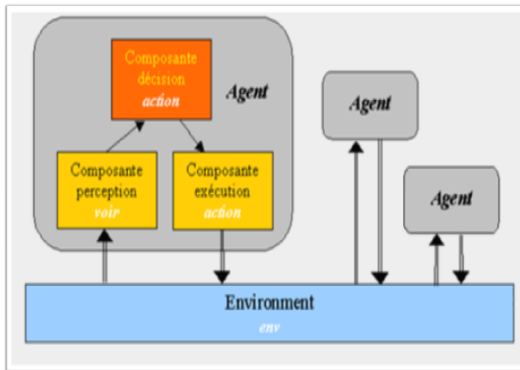
Modèles et
Architectures
d'agents

Mr.
KHEB-
BACHE
Mohib
Eddine

Introduction

Modèles
d'agents

Architecture
d'agents



Architecture d'un agent

Structure générale d'un agent en interaction (communication via env et directe)

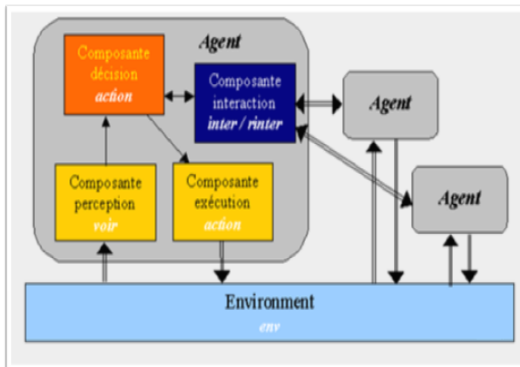
Modèles et
Architectures
d'agents

Mr.
KHEB-
BACHE
Mohib
Eddine

Introduction

Modèles
d'agents

Architecture
d'agents



On distingue deux types

Architectures abstraites

Agent purement réactif

Agent avec état

Architectures concrètes

architectures BDI

modulaires

architectures à base des tableaux noirs

architecture de Subsumption

tâches compétitives

système à base de classifieurs

architectures connexionnistes

- $S = \{s1, s2, \dots\}$ ensemble des états de l'environnement.
- Les compétences d'un agent sont représentés par l'ensemble des actions qu'il peut réaliser $A = \{a1, a2, \dots\}$
- Un agent peut être vu comme une fonction : $Agir : S^* \rightarrow A$

qui fait correspondre une séquence (*) d'états de l'environnement avec des actions.

- Ce sont des agents qui décident ce qu'il faut faire sans référence aux historiques
 - La prise de décision est basée uniquement sur le présent sans aucune référence au passé.
 - parfois appelés agents tropistique car ils répondent simplement directement à leur environnement.
- formellement, Le comportement d'un agent purement réactif peut être représenté par la fonction :

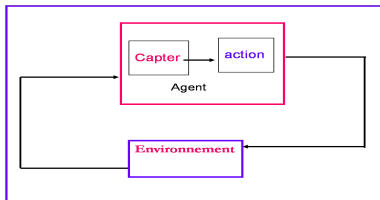
action : S \rightarrow A

exemple : agent thermostat

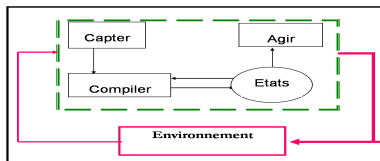
Supposons, sans perte de généralité, que l'environnement du thermostat peut être dans l'un des deux états soit trop froid, ou la température OK. Ensuite, le thermostat est simplement définie comme suit :

$$action(e) = \begin{cases} heateroff & \text{if } e = temperatureOK \\ heateron & \text{otherwise.} \end{cases} \quad (1)$$

- Le modèle abstrait doit être raffiné (décomposer en sous-systèmes)
- La première décomposition fonctionnelle d'un agent est faite entre la perception et son processus de décision sur les actions
- Soit P l'ensemble des perceptions et capter la fonction qui fait correspondre les états de l'environnement à des perceptions
 - capter : $S \rightarrow P$
 - agir : $P^* \rightarrow A$
 - 2 états s_1 et s_2 sont équivalents pour l'agent s'il les perçoit de façon unique : $s_1 \in S$ et $s_2 \in S$ et $\text{capter}(s_1) = \text{capter}(s_2)$.



- Pour utiliser son passé, un agent peut conserver son historique
- Soit I l'ensemble des états internes de l'agent, le processus de décision est exprimé à partir de I ; $agir : I \rightarrow A$
- Ce processus de mémorisation nécessite l'accumulation des connaissances par la fonction : Compiler (délibérer) : $I \times P \rightarrow I$
- donc, Les agents fonctionnent suivant le nouveau cycle : capter -compiler -agir.
 - L'agent perçoit le monde à travers « capter », met à jour son état à travers « compiler », puis choisit une action à exécutée à travers « agir ».



Architecture d'un agent

Architecture concrète (architectures BDI)

- Une architecture BDI est au cœur de la théorie BDI de l'action rationnelle, proposée pour la première fois par Michael Bratman
- l'agent décide des actions à entreprendre à partir de ses états internes qui sont exprimés sous la forme de :
 - **croiances (Belief)** : connaissances du monde ; les informations que l'agent possède sur l'environnement et sur d'autres agents, par sa capacité de perception ou par interaction avec d'autres agents, qui sont utilisées pour calculer les valeurs des désirs (critères).
 - **désirs (Desire)** : buts de l'agent, représentent les états de l'environnement, que l'agent aimerait voir réalisés. sont formalisés sous la forme d'un ensemble de critères qui seront utilisés pour évaluer les plans pour permettre à l'agent d'atteindre ses buts.
 - et **d'intentions (Intention)** : sont les désirs que l'agent a décidé d'accomplir ou les actions qu'il a décidé de faire pour accomplir ses désirs correspond au plan instancié choisi



Architecture d'un agent

Architecture concrète (architectures BDI)

Modèles et
Architectures
d'agents

Mr.
KHEB-
BACHE
Mohib
Eddine

Introduction

Modèles
d'agents

Architecture
d'agents

- Les systèmes d'agents "classiques" qui ont implémenté l'architecture BDI sont : IRMA = Intelligent Resource-bounded Machine Architecture, et PRS = Procedural Reasoning System
- le cycle de contrôle d'un agent BDI :

```
1 B:=B0
2 D:=D0
3 I:=I0
4 répéter
    4.1 obtenir nouvelles perceptions P
    4.2 B:=révision-croyance(B, P)
    4.3 D:=révision-désirs(B, D , I)
    4.5 I:=filtre-intentions(B, D, I)
    4.6 PE:=plan(B, I)
    4.7 exécuter(PE)
jusqu'à ce que l'agent soit arrêté
```

Où exécuter (PE) s'écrit comme :

```
tantque {PE<>{} et nonaccompli(I, B) et possible(I, B)} répéter
    4.7.1 x = première(PE); exécuter(x); PE = reste(PE)
    4.7.2 obtenir nouvelles perceptions p
    4.7.3 B = revc(B, p)
fin tant que
```

Architecture d'un agent

Architecture concrète (architectures BDI)

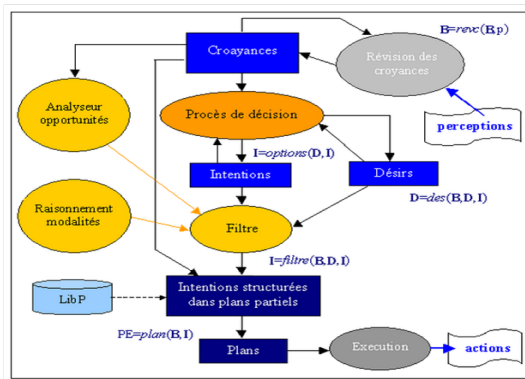
Modèles et Architectures d'agents

Mr. KHEB-BACHE Mohib Eddine

Introduction

Modèles d'agents

Architecture d'agents



Exemple, Comment les agriculteurs améliorent le système de culture en utilisant une architecture basée sur des agents BDI ?

- Solution proposée : Ce modèle comporte deux types d'entités :
 - agents agriculteurs
 - a les variables d'état suivantes : Parcelles, plans, intention, désirs, croyances (mémoire-dernières-productions, météo-derniers-jours, prix-culture, coût-culture), eau, finance.
 - la base de désirs (4 désirs (critères)) : Maximiser le profit, Minimiser les risques financiers, Minimiser la charge de travail, Maximiser les similarités avec le dernier plan choisi
 - La base de croyances (attributs) suivants : mémoire-dernières-productions, météo-derniers-jours, prix-culture, coût-culture, eau.
 - intention : plan d'assolement courant.
 - agents parcelles



Architecture d'un agent

Architecture concrète (architectures modulaires)

Modèles et
Architectures
d'agents

Mr.
KHEB-
BACHE
Mohib
Eddine

Introduction

Modèles
d'agents

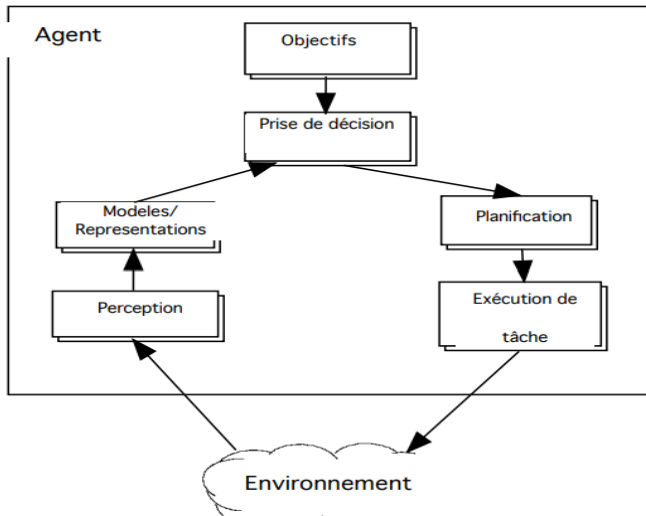
Architectures
d'agents

- Type de composant \implies Module
- architectures proposées pour la définition d'agents cognitifs
- Topologie : information et contrôle de flux entre plusieurs modules
- deux types : horizontales et verticales.

- fondées sur la notion d'ensemble de modules horizontaux liés par des connexions préétablies
 - chaque module **réalisant une fonction horizontale particulière**
 - toutes les liaisons sont fixes \implies flux d'information (circulation des informations) montant puis descendant
 - Les modules les plus courants sont :
 - Les fonctions perceptives (**module de perception** de extérieur par l'intermédiaire de capteurs ou de boîtes aux lettres comprenant des messages) et motrices
 - L'émission et l'interprétation des communications \implies **module d**
 - La base de connaissances (croyances, expertises du domaine, modélisation de l'environnement et des autres agents) \implies modèle cognitif, langages de représentation des connaissances.
 - La gestion des engagements, des buts et la prise de décision \implies **module de raisonnement** \implies les théories de la décision, problème du pathfinding (l'algorithme A*).
 - La planification des actions, etc.

Architecture d'un agent

Architecture concrète (architectures modulaires horizontales)



- réservées pour les SMA cognitifs
- Possibilité de concevoir plusieurs couches logicielles dotés de plusieurs agents/acteurs
 - chaque couche à un niveau d'abstraction de problème
 - les agents dans la même couche ont même nature (homogènes) et même rôles (tâches)
 - les agents entre les # couches sont des agents hétérogènes
 - La communication entre agents (entre plusieurs couches) s'effectue par envois de messages asynchrones \implies langage de communication entre agents.

- utilisée dans les systèmes systèmes mono-agents cognitifs symboliques \implies implémenter la structure interne d'agents cognitifs symboliques.
- Le modèle de tableau noir est fondé sur un découpage en modules indépendants,
 - appelés **sources de connaissance "KS"** (pour Knowledge Source)
 - et qui interagissent (communiquent) indirectement en partageant des informations.

- L'architecture d'un système à base de tableau noir comprend trois sous-systèmes :
 - LES SOURCES DE CONNAISSANCE (KS)
 - LA BASE PARTAGÉE (“TABLEAU NOIR” OU ”BLACKBOARD”)
 - ⇒ comprend toutes les informations que s'échangent les KS
 - ⇒ modélisation du domaine d'application comme l'espace des hypothèses/solutions.
 - UN DISPOSITIF DE CONTRÔLE : ordonne l'ordre dans lequel les KS (modules) seront effectivement activés ⇒ gère les conflits d'accès entre les KS.

Architecture d'un agent

Architecture concrète (à base de tableaux noirs)

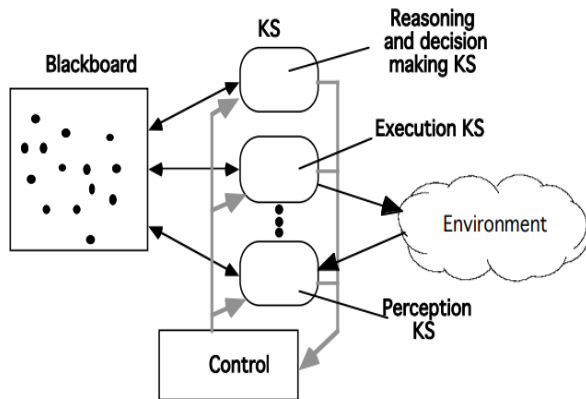
Modèles et
Architectures
d'agents


Mr.
KHEB-
BACHE
Mohib
Eddine

Introduction

Modèles
d'agents

Architecture
d'agents



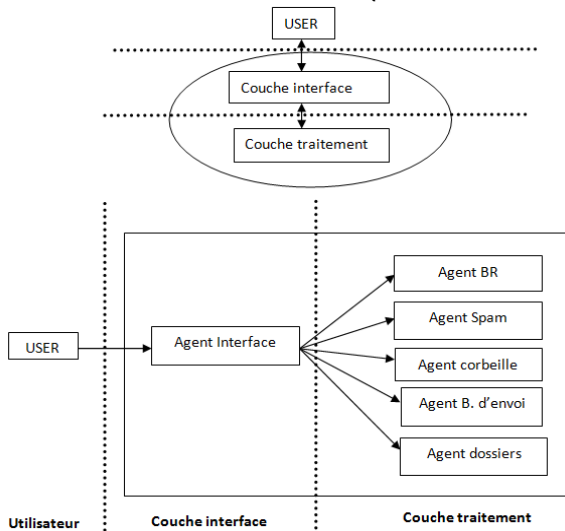
 Control flow
 Data flow

- De point de vue multi-agents, chaque KS pouvant être perçu comme un agent qui interagit avec les autres KS (agents)
 - \implies les KS ne sont être déclenchés effectivement par un système centralisé de contrôle.
- L'architecture de tableau noir présente un avantage de souplesse.
- Elle est utile lors de la phase de prototypage de la réalisation de systèmes
- ou lorsque les temps de réponses ne sont pas trop contraints.
- Son principal inconvénient provient de sa relative inefficacité,

développement d'une messagerie électronique (E-mail) intelligente en exploitant le paradigme agent (Pr. Okba KAZAR).

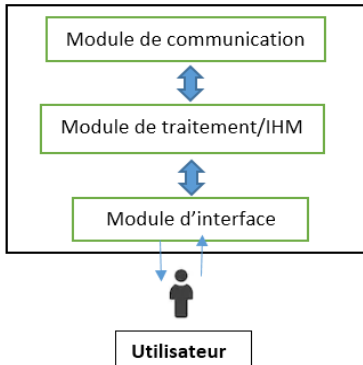
- **motivations** : pour modéliser un tel système, le paradigme agent est utilisé pour les raisons suivantes :
 - le système possède des composantes **autonomes** \implies se comportent sans l'intervention d'un être humain.
 - qui agissent de manière **parallèle**
- **Architecture de SMA** : l'architecture globale du système \implies en couche
- **l'architecture des agents du système** : \implies architecture modulaire.

développement d'une messagerie électronique (E-mail) intelligente en exploitant le paradigme agent (Pr. Okba KAZAR).



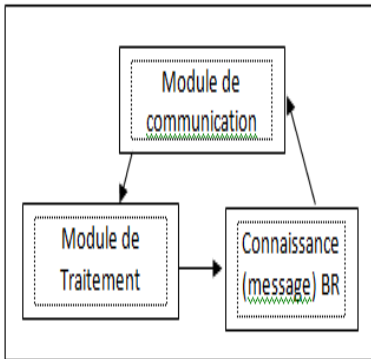
développement d'une messagerie électronique (E-mail) intelligente en exploitant le paradigme agent (Pr. Okba KAZAR).

- l'architecture de l'agents interface :



développement d'une messagerie électronique (E-mail) intelligente en exploitant le paradigme agent (Pr. Okba KAZAR).

- l'architecture des autres agents du système :

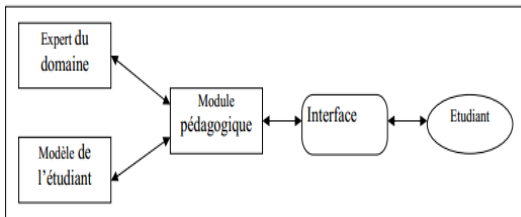


développement d'un tuteur intelligent (système d'enseignement intelligent assisté par ordinateur) (Pr. Okba KAZAR).

- **motivations** : pour modéliser un tel système, le paradigme agent est utilisé pour les raisons suivantes :
 - l'enseignement des principaux concepts de base de données
 - où les différents types de connaissances sont issus de trois sources : **l'expert pédagogique, l'expert du domaine à enseigner et l'étudiant**
 - \implies pbm complexe, distribution et hétérogénéité des différents types de connaissances sur un ensemble d'agents communicants
 - **Architecture de SMA** : l'architecture globale du système \implies à base de tableau noir
 - **l'architecture des agents du système** : \implies architecture modulaire.

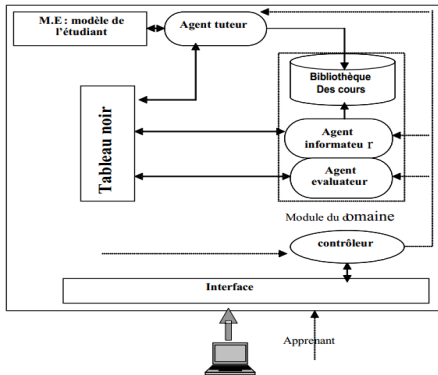
développement d'un tuteur intelligent (système d'enseignement intelligent assisté par ordinateur) (Pr. Okba KAZAR).

- le système d'E.I.A.O (tuteurs intelligents) est organisé autour de trois modules (KS) :
 - le domaine (module de résolution de problème ou d'expertise)
 - le modèle de l'étudiant
 - et l'expert pédagogique (modèle tutorial)



développement d'un tuteur intelligent (système d'enseignement intelligent assisté par ordinateur) (Pr. Okba KAZAR).

⇒ l'architecture globale du système ⇒ à base de tableau noir



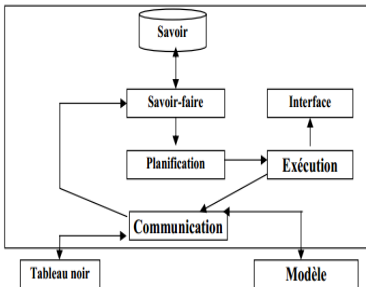


développement d'un tuteur intelligent (système d'enseignement intelligent assisté par ordinateur) (Pr. Okba KAZAR).

- Les agents du système
 - Agent pédagogue
 - Agent informateur
 - Agent évaluateur

développement d'un tuteur intelligent (système d'enseignement intelligent assisté par ordinateur) (Pr. Okba KAZAR).
⇒ l'architecture des agents du système

- Agent pédagogique



développement d'un tuteur intelligent (système d'enseignement intelligent assisté par ordinateur) (Pr. Okba KAZAR).
 ⇒ l'architecture des agents du système

- Agent informateur

