



Introduction aux Systèmes Multi-Agents

C. HANACHI, C. SIBERTIN-BLANC

Université Toulouse I & IRIT.

<nom>@univ-tlse1.fr



Objectifs

■ Comprendre :

- les concepts définissant le paradigme agent
- l'utilité de cette approche pour le développement de certaines applications informatiques ;
- la position de cette technologie par rapport aux Systèmes Experts et à la Programmation Objet.
- La formalisation du comportement des agents et de leurs interactions à l'aide des Réseaux de Petri.



PLAN

- Introduction intuitive
- De l'IA aux Agents en passant par les systèmes experts
- Définition, propriétés et applications des agents.
- Architectures et Fonctionnement d'un agent
- Interactions entre agents
- Organisation des agents
- Applications des SMA



Plus d'information ?

- **Jacques Ferber**, *les systèmes multi-agents*, InterEditions, 1995.
- **J.P. Briot et Y. Demazeau** (Editeurs) *Principes et architecture des Systèmes multi-agents*, Hermes, 2001.
- **Michael Wooldridge**, *An Introduction to Multiagent Systems*, John Wiley & Sons Editor, 2002. ISBN 0 47149691X
- fipa.org : <http://www.fipa.org>
- agentlink.org : <http://www.agentlink.org>
- multiagent.com : <http://www.multiagent.com>
- Outils: www.agentbuilder.com/AgentTools/



1. Introduction intuitive



Un agent

- « On appelle **agent** un système mécanique, biologique ou logiciel qui interagit avec son environnement ». Anne Nicole.
- Par exemple :
 - une imprimante peut être vue comme *un agent mécanique* qui réagit à des commandes et produit des actions en retour.
 - Les plantes, les animaux, les humains sont des *agents biologiques* ayant plus d'autonomie, qui absorbent des nutriments, qui respirent, qui se transforment et qui transforment leur environnement.
 - Les *agents logiciels* sont des programmes autonomes, mis en route sur une machine, qui perçoivent certains éléments de leur environnement par les flots d'entrée (clavier, souris, capteurs) et agissent par leur flot de sortie (affichages sur l'écran, commande de machine physique, contrôle de processus).



Un système multi-agents

- « Un **Système Multi-Agents** (SMA) comporte plusieurs agents qui interagissent entre eux dans un environnement commun. Certains de ces agents peuvent être des personnes ou leurs représentants (avatars), ou même des machines mécaniques. S'il y a moins de trois agents, on parle plutôt d'interaction homme/machine, ou machine/machine que de systèmes multi-agents.» Anne Nicole.



Pour concevoir un SMA, il faut définir

- un modèle de SMA
 - le modèle de chacun des agents qui vont entrer en action (niveau microscopique);
 - définir leur environnement et leurs interactions (niveau macroscopique) ;
 - définir les organisations sociales (niveau macro) qui les structurent.
- un modèle concret de SMA :
 - qui crée, initialise les agents,
 - installe leur organisation et
 - lance les agents qui doivent intervenir pour une exécution particulière.



Rôles des SMA

- Résoudre un problème de manière distribuée : systèmes multi-experts.
 - Les actions des agents sont des transformations d 'objets liées à la description d 'un problème.
 - agents plutôt rationnels
- Simulation de phénomènes complexes.
 - Les agents simulent des actions physiques, biologiques ou sociales qui produisent des modifications du monde représenté. Ex : simulation de la pêche dans le delta du Niger, des épidémies, écosystèmes (proies / prédateurs).
 - Agents plutôt réactifs.



Rôles des SMA

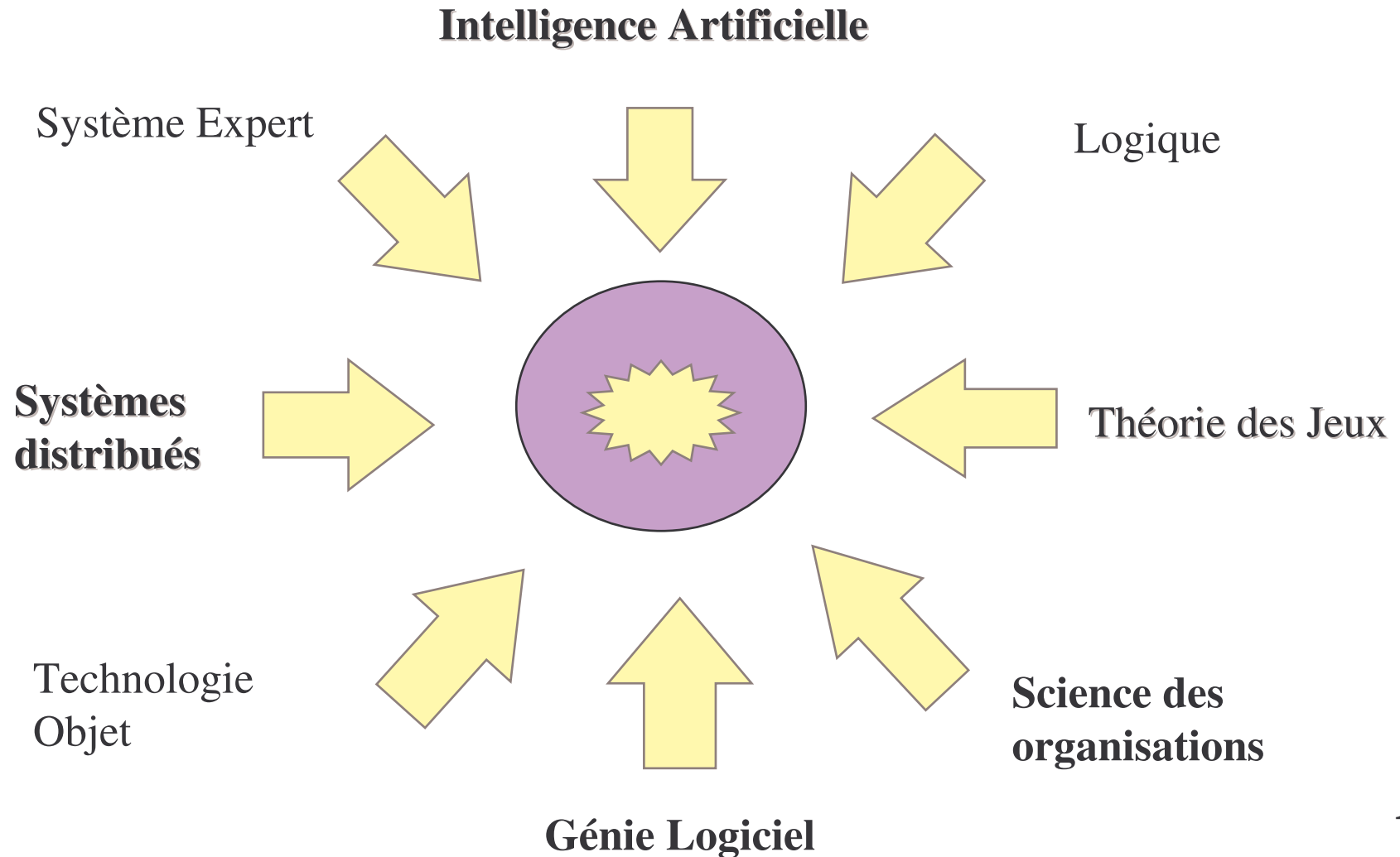
- Gérer et maintenir un environnement de travail.
 - Les actions physiques ou sociales réalisées par les agents sont des actions réelles, elles évoluent dans le temps et modifient le monde : robots footballeurs, agents négociant un rendez-vous au profil de l'utilisateur.
 - Agents plutôt cognitifs et sociaux.



Le rôle central de l'interaction

- « Une interaction est la mise en relation dynamique de deux ou plusieurs agents par le biais d'un ensemble d'actions réciproques... Les interactions sont non seulement la conséquence d'actions effectuées par plusieurs agents en même temps, mais aussi l'élément nécessaire à la constitution d'organisation sociales » (Ferber).
- Interagir pour gérer, communiquer, se coordonner, coopérer, négocier...
- Interaction passive par modification de l'environnement ;
- Interaction intentionnelle par action visible sur l'environnement ou par l'intermédiaire de messages que les agents s'envoient les uns aux autres.
- Nécessité de modèles formels d'interactions.

Disciplines de Recherche Contribuant aux SMA





1. De l'IA aux Agents en passant par les Systèmes Experts.

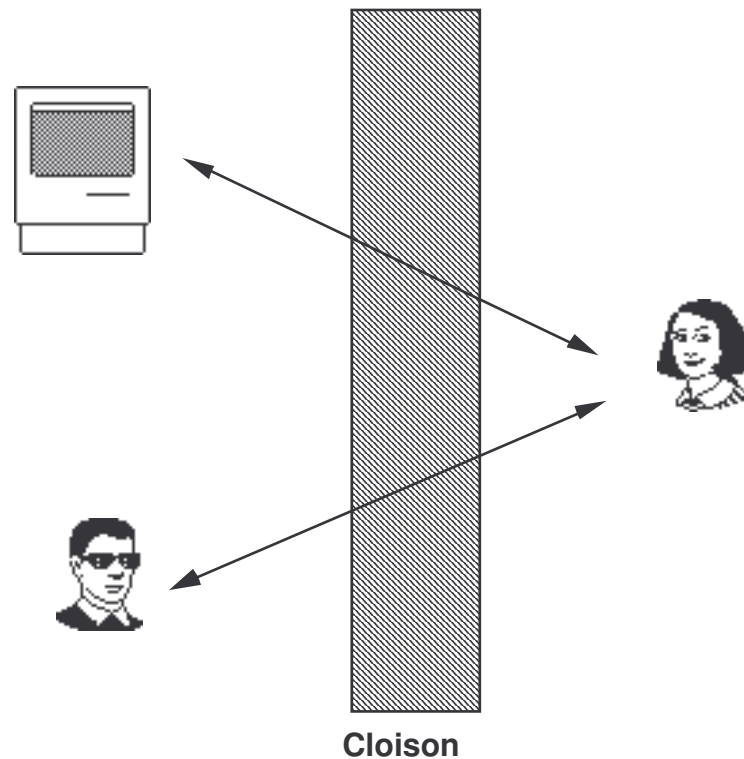


Rappels sur l'Intelligence Artificielle ?

- **BUT DE L'IA** : doter les machines de capacités habituellement attribuées à l'intelligence humaine :
 - jouer aux échecs,
 - parler,
 - traduire un texte,
 - conduire un vélo,
 - amener le petit-déjeuner au lit,
 - reconnaître un ami sur une photo.

Qu 'est-ce qu 'une machine intelligente ?

Le test de A. Turing





Caractéristiques de l'IA

- Manipulation d'**informations symboliques** (concepts, règles, objets) identiques à ceux habituellement pris en compte par un être humain lorsqu'il raisonne.
- Basée sur des **méthodes heuristiques** qui :
 - guident la résolution vers les solutions les plus vraisemblables,
 - économisent le temps de calcul car élaguent l'ensemble des chemins de résolution possibles,
 - peuvent être remises en cause.
⇒ permettent d'envisager des problèmes complexes.
- Basée sur une grande quantité de connaissances.

Recherche heuristique sur l'exemple du taquin.

1	4	2
8	7	6
	5	3



1	2	3
4	5	6
7	8	



Domaines d'application de l'IA

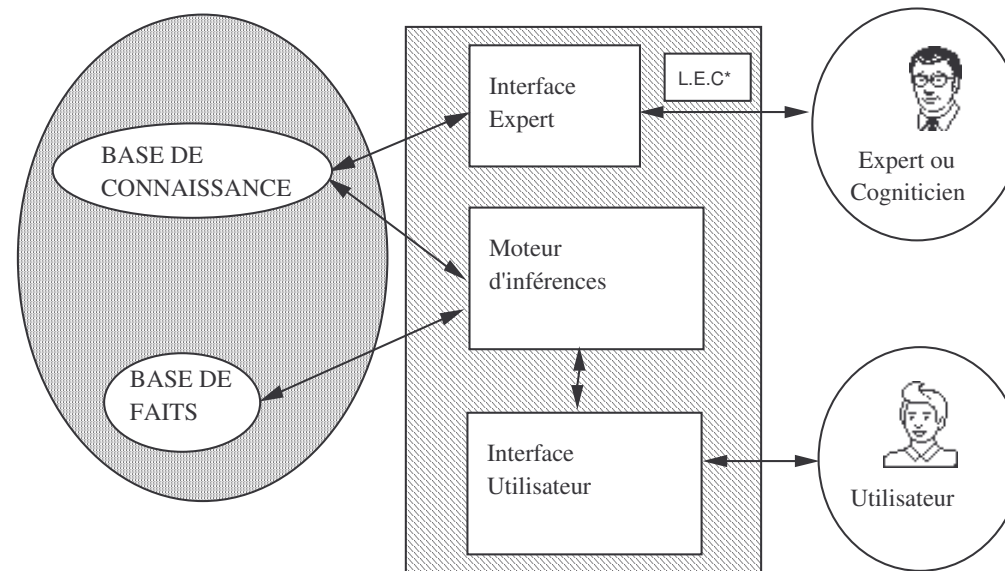
- Démonstration automatique de théorèmes
- Traitement du langage naturel écrit
- Traitement automatique de la parole :
compréhension, synthèse
- Interprétation d'images et vision par ordinateur
- Robotique
- Jeux
- Systèmes Experts
- Apprentissage



Qu'est-ce qu'un système expert ?

- "Les systèmes experts sont des logiciels, bientôt aussi des matériels, destinés à remplacer ou assister l'homme dans des domaines où est reconnue une expertise humaine :
 - insuffisamment structurée pour constituer une méthode de travail précise, sûre, complète, directement transposable sur ordinateur,
 - sujette à révisions ou compléments (selon l'expérience accumulée) » H. FARRENY
- "Un SE est un programme ordinaire écrit dans un style particulier »
J. FELDMANN

Architecture d'un système expert.



*: Langage d'Expression des
Connaissances



Exemple d'une base de connaissances

LA BASE DE FAITS

/*faits de la forme activites (*Entreprise, *Secteur) */

activites (boby-oil, petrole).

activites (boby-oil, alimentation).

activites (yakachauffer, alimentation).

activites (vroom, automobile).

activites (vroom, alimentation).

activites (glouton, alimentation).

activites (glouton, automobile).

activites (petrole-jr, petrole).

/* faits de la forme participations(*Entreprise1,*Entreprise2,*Pourcentage)*/

participations(boby-oil, yakachauffer,60).

participations (vroom, yakachauffer,20).

participations (glouton, vroom, 60).

LA BASE DE REGLES

R1 : **SI** participations (*X1, *X2, *Z)

ALORS controle (*X1, *X2).

R2 : **SI** participations (*X1, *X3, *Z) **ET** controle (*X3, *X2)

ALORS controle (*X1, *X2).

R3 : **SI** controle (*X1,*X2) **ET** activites (*X2,*Y2)


ALORS activites (*X1,*Y2).

Questions ? : controle (*X,*Y) ? controle (*X,yakachauffer) ? controle (vroom,yakachauffer) ?




Pourquoi distribuer le contrôle et/ou la connaissance

- Distribution fonctionnelle :
 - multi-expertise / points de vue multiples ;
 - décomposition de problèmes
- Distribution physique :
 - Problèmes intrinsèquement distribués : réseaux, contrôle aérien.
 - Robotique.
- Distribution informatique :
 - développement de machines parallèles ;
 - développement des langages à objets.
- Faire coopérer des systèmes pré-existants, hétérogènes, distribués, autonomes : sources d'information dans le web, workflow inter-organisationnel.



Quelques modèles et systèmes précurseurs (1)

- Hearsay II (Hayth-Roth 1973) :
 - reconnaissance de la parole ;
 - modèle du tableau noir.
- DVMT (Distributed Vehicle Monitoring Testbed)
 - Modèle d'organisation répartie ;
 - Repérage du mouvement de véhicules à partir d'un réseau de capteurs.
- Acquisition Coopérative d'Information (Lesser) :
 - Modèle de résolution distribuée de problèmes ;
 - Recherche d'information dans des sources réparties, hétérogènes et autonomes.



Quelques modèles et systèmes précurseurs (2)

- Contrôle/ordonnancement d'Ateliers Flexibles (Sycara).
- ...
- Vers des entités autonomes capables de :
 - coopérer (contrôle) ;
 - collaborer (allocation de tâches) ;
 - négocier (résolution de conflits) ;
 - se coordonner (synchronisation).



Éco-résolution : les principes selon Alexis Drogoul

<http://www.poleia.lip6.fr/~drogoul/projects/eco/npuzzle.html>

Eco-Résolution : propose de remplacer les méthodes classiques de résolution de problèmes par une approche multi-agents dans laquelle la solution correspond à l'obtention de configurations stables d'une population d'agents au travers de l'interaction.

Chaque éco-agent est défini de la manière qui suit :

- un *but* représenté par un autre éco-agent avec lequel il entretient une relation de satisfaction ;
- un *état interne*. Trois états sont possibles : *satisfait*, en *recherche de satisfaction*, en *recherche de fuite* ;
- des *actions élémentaires*, qui dépendent du domaine d'application. Ce sont des actions de satisfaction ou de fuite ;
- une *fonction de perception* des agents gêneurs (*i.e.* les agents qui empêchent l'agent courant d'être satisfait ou de fuir) ;
- une *liste de dépendances* (*i.e.* les agents dont l'agent courant est le but et ses dépendances ne pourront être satisfaits que si l'agent courant se trouve lui-même satisfait).

Éco-résolution : l'algorithme général

Le comportement élémentaire d'un éco-agent est indépendant du domaine d'application et répond aux principes suivants :

La volonté d'être satisfait. Un éco-agent cherche à atteindre un état où il est satisfait. S'il ne peut pas y arriver, parce qu'empêché par des gêneurs, il les agresse pour les faire fuir.

```
procédure trySatisfaction(x : ecoAgent) =  
  si le but de x est satisfait alors  
    pour tous les agents y qui gênent x  
      tryFlee(y, x, but(x))  
  dès qu'il n'y a plus de gêneurs doSatisfaction(x)
```

doSatisfaction est dépendant du domaine d'application et se charge d'exécuter l'action dans laquelle l'agent vérifie sa condition de satisfaction. Cette action est possible car les gêneurs sont partis.



Éco-résolution : l'algorithme général

L'obligation de fuir. Lorsqu'un éco-agent est agressé, il doit obligatoirement fuir. Il doit chercher une situation où la contrainte *c* passé en paramètre de la procédure fuir soit satisfaite.

// x fuit y avec la contrainte *c*

procédure tryFlee(*x* : ecoAgent, *y*: ecoAgent, *c*: ecoAgent)

si *x* était satisfait, il devient insatisfait

soit *p* = trouverPlacePourFuir(*x*, *y*, *c*)

si *p* est nil alors ``pas de fuite possible",

sinon pour tous les agents *z* qui gênent *x* dans sa fuite vers *p*

tryFlee(*z*, *x*, *p*)

dès qu'il n'y a plus de gêneurs pour fuir, alors **doFlee**(*x*, *p*)

Taquin & éco-résolution

1	2	3
4	6	8
	7	5



1	2	3
4	5	6
7	8	



2. Définition, propriétés et applications des agents et des SMA ?



Définitions

- « The term "agent" is used (and misused) increasingly to describe a broad range of computational entities. This tends to obscure the differences between radically different approaches.
 - Some agents performs tasks individually ... others need to work together
 - Some are mobile ... some static
 - Agents communicate via messages ... some don't communicate at all
 - Some learn and adapt ... others don't
 - Despite this diversity, we can identify some common properties. »



Définitions

- “Un agent est une entité réelle ou virtuelle évoluant dans un environnement, capable de le percevoir, qui peut communiquer avec d’autres agents, qui exhibe un comportement autonome, lequel peut être vu comme la conséquence de ses connaissances, de ses interactions avec d’autres agents et des buts qu’il poursuit” (Ferber).
- “Intelligent agents continuously perform three functions: perception of dynamic conditions in the environment; action to affect condition in the environment; and reasoning to interpret perceptions, solve problems, draw inferences, and determine actions.” (Hayes-Roth).



Définitions

- « On appelle **système multi-agent** (ou SMA), un système composé des éléments suivants :
 - Un environnement **E**, c'est à dire un espace disposant généralement d'une métrique.
 - Un ensemble d'objets **O**. Ces objets sont situés, c'est à dire que, pour tout objet, il est possible à un moment donné, d'associer une position dans E. Ces objets sont passifs, c'est à dire qu'ils peuvent être perçus, créés, détruits et modifiés par les agents.
 - Un ensemble **A** d'agents, qui représentent les entités actives du système.
 - Un ensemble de relations **R** qui unissent les objets et les agents entre eux.
 - Un ensemble d'opérations **Op** permettant aux agents A de percevoir, produire, consommer, transformer et manipuler les objets de O.
 - Des **opérateurs** chargés de représenter l'application de ces opérations et la réaction du monde à cette tentative de modification, que l'on appellera les lois de l'univers. »Ferber.



Caractéristiques des SMA

- Chaque agent a des *informations* ou des *capacités* de résolution de problèmes *limitées*, ainsi chaque agent a un point de vue partiel ;
- Il n'y a *aucun contrôle global* du SMA ;
- Les données sont *décentralisées* ;
- Le calcul est *asynchrone*.

Caractéristiques de différentiation des SMA

	Attributs	Valeurs
Agents	Nombre	Deux et plus...
	Uniformité	Homogénéité...hétérogénéité
	Buts	Contradictaires...complémentaires
	Architecture	Réactive...délibérative
	Compétences	Simple...avancée
Interaction	Fréquence	Faible...Elevée
	Persistance	Court terme...Long terme
	Niveau	Envoi de signal...manipulation de connaissances
	Contrôle et Flux de données	Décentralisée...hiérarchique
	Variabilité	Fixe...variable
	Propos	Compétitif...coopératif
Environnement	Prédiction	Possible...impossible
	Accessibilité et connaissance	Illimitées...limitées
	Dynamique	Fixe...variable
	Variété	Pauvre...riche
	Disponibilité des ressources	Restrictives...abondante



Dichotomie entre agent réactif et agent cognitif

- **Agents cognitifs** : chaque agent est spécialisé dans un domaine et sait communiquer avec les autres. Ils possèdent des buts et des plans explicites leur permettant d'accomplir leurs buts. (analogie avec les groupes en sociologie).
 - Représentation explicite de soi, environnement et les autres agents.
 - Organisation explicite.
 - Interaction explicite et élaborée.
- Exemple : aptitude à construire un plan du type :
 - Plan ouvrir la porte :
 - aller jusqu'à l'endroit où se trouve la clef
 - prendre la clef
 - aller jusqu'à la porte
 - ouvrir la porte avec la clef



Dichotomie entre agent réactif et agent cognitif

- **Agents réactifs** : agents sans intelligence (sans anticipation, sans planification) qui réagissent par stimulus-réponse à l'état courant de l'environnement. Des comportements intelligents peuvent émerger de leur association.
 - Pas de représentation explicite.
 - Organisation implicite/induite
 - Communication via l'environnement.
- Exemple de comportement réactif pour ouvrir une porte :
 - R1 : si je suis devant la porte et que j'ai une clef alors l'ouvrir
 - R2 : si je suis devant la porte et sans clef alors essayer de l'ouvrir
 - R3 : si la porte ne s'ouvre pas et que je n'ai pas la clef alors aller chercher la clef
 - R4 : si je cherche une clef et qu'il y a une clef devant moi alors prendre la clef et aller vers la porte.



Propriétés clés des Agents

- **Autonomie** : un agent peut agir sans l'intervention directe d'un tiers (agent ou humain) et contrôler ses actions ainsi que son état interne.
- **Réactivité** : un agent perçoit son environnement et répond aux changements qui s'y produisent en un temps raisonnable. L'environnement peut être le monde physique, un utilisateur via une interface graphique, d'autres agents, un système d'information (internet), ...
- **Communication** : un agent peut communiquer avec d'autres agents ainsi qu'avec des utilisateurs humains.
- **Aptitude sociale** : un agent peut interagir avec d'autres agents de façon coopérative ou compétitive pour atteindre ses objectifs.
- **Pro-activité** : l'agent est capable, sur sa propre initiative, de se fixer des buts pour atteindre ses objectifs (opportuniste).



Propriétés clés des agents

- **Agent Intelligent** : un agent est « intelligent » s'il est capable de réaliser des actions *flexibles* et *autonomes* pour atteindre les objectifs qui lui ont été fixés. La flexibilité correspond aux propriétés de Réactivité, Pro-activité, Aptitudes sociales.



Propriétés additionnelles

- **Apprentissage** : un agent peut mémoriser ses expériences et adapter son comportement en conséquence.
- **Mobilité** : un agent peut se déplacer d'une machine à une autre et éventuellement se dupliquer.

Cet agenda électronique est-il un agent ?

Janvier 2002 - Microsoft Internet Explorer

View Favorites Tools Help

Forward Stop Refresh Home Search Favorites History Mail Print Edit Messenger

http://www.calendars.net:8194/dessihm/d01/01/2002?display=M&style=B&positioning=A

UT1 - ENAC

Navigate: 2001 Jan Feb Mar Avr Mai Jun Jul Aou Sep Oct Nov Dec 2003

Janvier 2002

Lundi	Mardi	Mercredi	Jeudi	Vendredi
31 [1] Dec	1 Jan	2	3	4
Vacances de Noël	Vacances de Noël 10:00 - 12:00 Anglais ENAC bât C, deuxième étage 14:00 - 18:00 Chef d'oeuvre New Year's Day (civic)	09:00 - 12:00 Chef d'oeuvre	Vacances de Noël	Vacances de Noël
7 [2]	8	9	10	11
	10:00 - 12:00 Anglais ENAC bât C, deuxième étage 13:30 - 16:30 Cours 11 : Technologies Internet Rémi Bastide 14:00 - 18:00 Chef d'oeuvre	09:00 - 12:00 Chef d'oeuvre 09:15 - 12:15 Cours n°11 : Modèles et Technologie Composants COM Philippe Palanque 13:30 - 16:30 Cours 11 : Technologies Internet Rémi Bastide	10:00 - 12:00 Anglais en remplacement du cours du 18/12/01 14:00 - 16:00 Visualisation d'information Usages et systèmes Stéphane Chatty	09:00 - 12:00 Cours 7 : Ergonomie Irène Gaillard
14 [3]	15	16	17	18
13:30 - 15:00 Présentation technique Chef d'oeuvre : Portail vocal 15:00 - 16:30 Présentation	10:00 - 12:00 Anglais ENAC bât C, deuxième étage 13:30 - 15:00 Présentation technique Chef d'oeuvre :	09:00 - 12:00 Chef d'oeuvre 14:00 - 17:00 Cours 2 : Interaction gestuelle Nadine Vigouroux	09:00 - 12:00 Cours 2 : Interaction vocale en entrée Philippe Truillet 14:00 - 16:00 Visualisation	09:00 - 12:00 Cours 12 : Approches Multi-Agent Chihab Hanachi Salle MH 204



Cet agenda électronique est-il un agent ?

- Il rend service à des utilisateurs.
- Est-il réactif ? : oui, quand un utilisateur modifie le calendrier, le manager est informé.
- Est-il pro-actif ? Non. Quand un rendez-vous approche, il n'en informe pas l'intéressé.
- Est-il autonome ? Non. Ces décisions d'agir sont programmées, il ne raisonne pas et ne planifie pas.
- Pour être un agent intelligent, il est nécessaire :
 - d'anticiper les besoins des utilisateurs (regarder le calendrier des vacances, questionner le secrétariat, ...) ;
 - communiquer avec d'autres calendriers pour arranger des rendez-vous ;
 - s'adapter et apprendre les préférences des utilisateurs.



Domaines d 'application des agents

- **Systèmes ouverts** (Exemple Internet).
Exigences : négociation, coopération, entremetteur, annuaire, appels d 'offres.
- **Systèmes distribués par nature** : (géographique, fonctionnelle, humaine).
- **Systèmes Complexes** : société d 'entités coopérantes, robotique collective. Les agents apportent modularité, abstraction, possibilité de simulation.
- **Assistant expert** (délégation). Exigences :
 - autonomie : à partir d 'une spécification vague et imprécise, il doit déterminer comment le problème peut être résolu et ensuite le résoudre sans une intervention constante de l 'utilisateur.
 - Pro-activité : il ne doit pas attendre la fin de chaque action utilisateur pour agir mais faire des suggestions à l 'utilisateur.
 - Adaptabilité : il doit connaître les préférences de l 'utilisateur et leur évolution et les intégrer dans son comportement.
- **Système légataire** (agentification).



Comparaison avec d 'autres technologies

■ **Systèmes Experts**


- fournissent des conseils
- raisonnent, mais n 'agissent pas en perception de ..., ni en action sur ...
- pas pro-actifs
- pas autonome : nécessitent des interventions et des instructions.
- Pas adaptatifs.
- Pas distribués.
- Les agents peuvent être dotés de règles inférentielles utilisées dans les SE.
- L 'évolution des SE vers les systèmes multi-experts fut à l 'origine des SMA.



Comparaison avec d 'autres technologies

■ Objets

- Contrôlent leur état mais pas leur comportement (répondent directement aux requêtes correspondant à leurs méthodes). L 'agent, contrairement à l 'objet, a le pouvoir de négocier ou refuser les demandes de services.
- flexibilité absente.
- pas nécessairement concurrents alors que l 'agent possède son propre fil d 'exécution (thread). La notion d '*objets actifs* permet de faire des agents mais pas à les rendre intelligents (flexibles et autonomes).
- Agents intègrent la notion d '*action persistante* qui tente de manière répétée d 'accomplir quelque chose (pas la peine de contrôler explicitement le succès, l 'échec, les alternatives, ...).
- Organisation et interaction simples.
- Mais les langages à objets peuvent aider à construire des agents...



3. Architecture et fonctionnement des agents ?



Architecture abstraite

- $S=\{s_1,s_2,\dots\}$ l'ensemble des états de l'environnement.
- Les compétences d'un agent sont représentées par l'ensemble des actions qu'il peut réaliser : $A=\{a_1,a_2,\dots\}$
- 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.
- Agent purement réactif : agir : $S \rightarrow A$



Architecture abstraite

- Soit P l'ensemble des perceptions et *capter* la fonction qui fait correspondre les états de l'environnement à des perceptions qu'en a l'agent :
 - $\text{capter} : S \rightarrow P$
 - $\text{agir} : P^* \rightarrow A$
 - 2 états $s1$ et $s2$ sont équivalents pour l'agent s'il les perçoit de façon unique : $s1 \in S$ et $s2 \in S$ et $\text{capter}(s1) = \text{capter}(s2)$.
- Agent avec état :
 - I l'ensemble des états internes
 - $\text{agir} : I \rightarrow A$
 - Compiler (délibérer) : $I \times P \rightarrow I$
 - nouveau cycle : capter - compiler - agir.



Architecture

- **Les agents logiques** : fonctionnement basé sur des déductions logiques.
- **Les agents réactifs** : fonctionnement basé sur une simple correspondance entre les situations et les actions (interagissent avec leur environnement mais sans raisonner dessus).
- **Les agents BDI** : l'agent décide des actions à entreprendre à partir de ses états internes qui sont exprimés sous la forme de croyances (Belief), de désirs (Desire) et d'intentions (Intention).
- **Les agents multi-niveaux** : les connaissances internes de ce type d'agents sont organisées en différents niveaux d'abstractions, permettant ainsi différents niveaux de traitement.



Architecture : agent logique

- Exemples d 'agents logique « robots nettoyeurs »
 - Modélisation de l 'environnement :
 - Position (X,Y) : l 'agent est à la position (X,Y).
 - Poussière(X,Y) : il y a une poussière en (X,Y).
 - Orientation (D). : l 'agent est dans la direction D.
 - Actions
 - avancer, tourner : pour se déplacer.
 - Aspirer : pour aspirer la poussière.
 - Règles de comportement (priorité décroissante)
 - R1 : si position (X,Y) et poussière (X,Y) alors faire (aspirer).
 - R2 : si position (0,0) et orientation(nord) et non poussière (0,0) alors faire (avancer).
 - ...

Architecture : agent logique

- Action similaire à un résolveur de théorèmes.
- Principe : Si une action peut être dérivée de l'état courant la faire, sinon faire une action qui est consistante avec l'état courant.

Fonction agir (UnEtat : Etats) : ActionsPossibles

Début

Pour chaque action **a** ∈ ActionsPossibles faire
 Si UnEtat |-Faire(a) alors ' faire(a) se déduit de l'état courant.
 Renvoyer(a)

 Finsi

FinPour

Pour chaque action **a** ∈ ActionsPossibles faire
 Si UnEtat (**non** |-) (**non Faire(a)**) 'a est consistante avec son état interne
 Renvoyer(a)

 Finsi

FinPour.

Renvoyer null

FinFonction agir.



Agents réactif (subsumption architecture)

■ Principe :

- Interagissent simplement avec leur environnement plutôt que de se le représenter et de raisonner dessus.
- Basés sur des règles de type (situation, action).
- Plusieurs comportements peuvent être lancés.
- Sélection de comportements non contradictoires.
- Niveaux comportementaux.

■ Formalisation :

- Un comportement est une paire (c,a) , $c \subseteq P$ est *un ensemble de perceptions* appelées conditions, et $a \in A$ est *une action*.
- Un comportement (c,a) sera potentiellement jouable ssi l'environnement est dans un état $s \in S$ avec $\text{capter}(s) \in c$.
- On associe à l'ensemble des règles comportementales R de l'agent une relation binaire d'inhibition I . $I \subseteq R \times R$. C'est une relation d'*ordre total*. On écrit $b1 \ I \ b2$ si $(b1,b2) \in I$, et on prononce $b1$ inhibe $b2$, ce qui signifie que $b1$ est prioritaire sur $b2$.

Agents réactifs

- Fonction agir ($p: P$) : A
var activable : desrègles
début
 $\text{activable} := \{(c,a) / (c,a) \in R \text{ et } p \in c\}$
 pour chaque $(c,a) \in \text{activable}$ faire
 si $\neg (\exists (c',a') \in \text{activable} \text{ tel que } (c',a') \text{ Inhibe } (c,a))$
 alors
 renvoyer a
 finsi
 fin pour
 renvoyer null
Fin fonction agir.



Agents réactifs : exemple

- **Exemple** (selon Steels) : l'objectif est de rapporter des pierres rares localisées sur une planète distante, sur laquelle seul les robots pourront aller. On ne sait pas où se trouvent ces pierres, mais on sait qu'elles y sont en tas. Les robots n'ont pas de cartes de cette planète et en raison du terrain très accidenté, ils ne pourront pas communiquer directement entre eux. Enfin, ils devront être capables de retourner au vaisseau pendant et après avoir accompli leur mission. Le vaisseau émet un signal continuellement et dans toutes les directions, son intensité permet de se guider.

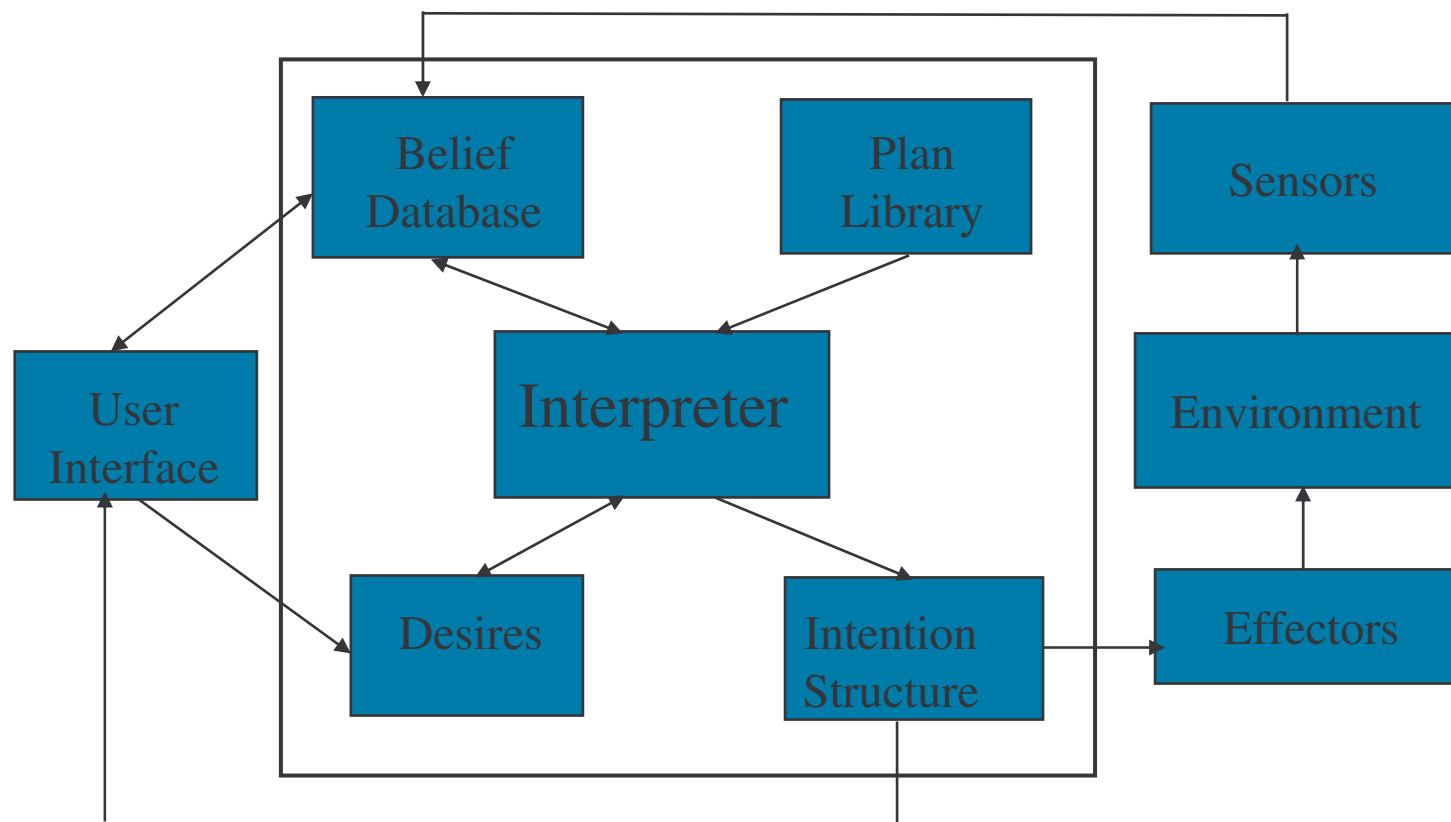
-



Architecture BDI

- **Believes** : informations(faits) courantes qu 'un agent possède à propos de l 'environnement.
- **Desires** : buts à réaliser si possible. C'est la description du comportement désiré de l'agent en termes de satisfaction, de maintenance ou de vérification de propriétés.
- **Plans** : connaissances qui déterminent comment certaines séquences de tests et d 'actions permettent d 'atteindre des buts ou bien réagir à certaines situations. Plan=(condition d 'invocation, condition sur le contexte, plan : when event if condition then action).
- **Intentions** : plans (et donc but) instancié choisis pour une (éventuelle) exécution.

Architecture BDI.



BDI agent architecture (selon Kiny).



Architecture BDI.

- Fonction agir (p:P):A
début
 B:=réviser_les_croyances (B, p)
 D:=déterminer_de_nouveaux_buts (B,D,I)
 I:=sélectionner_les_buts_a_tenter (B,D,I)
 renvoyer (action(unplan (I)))
Fin fonction agir.

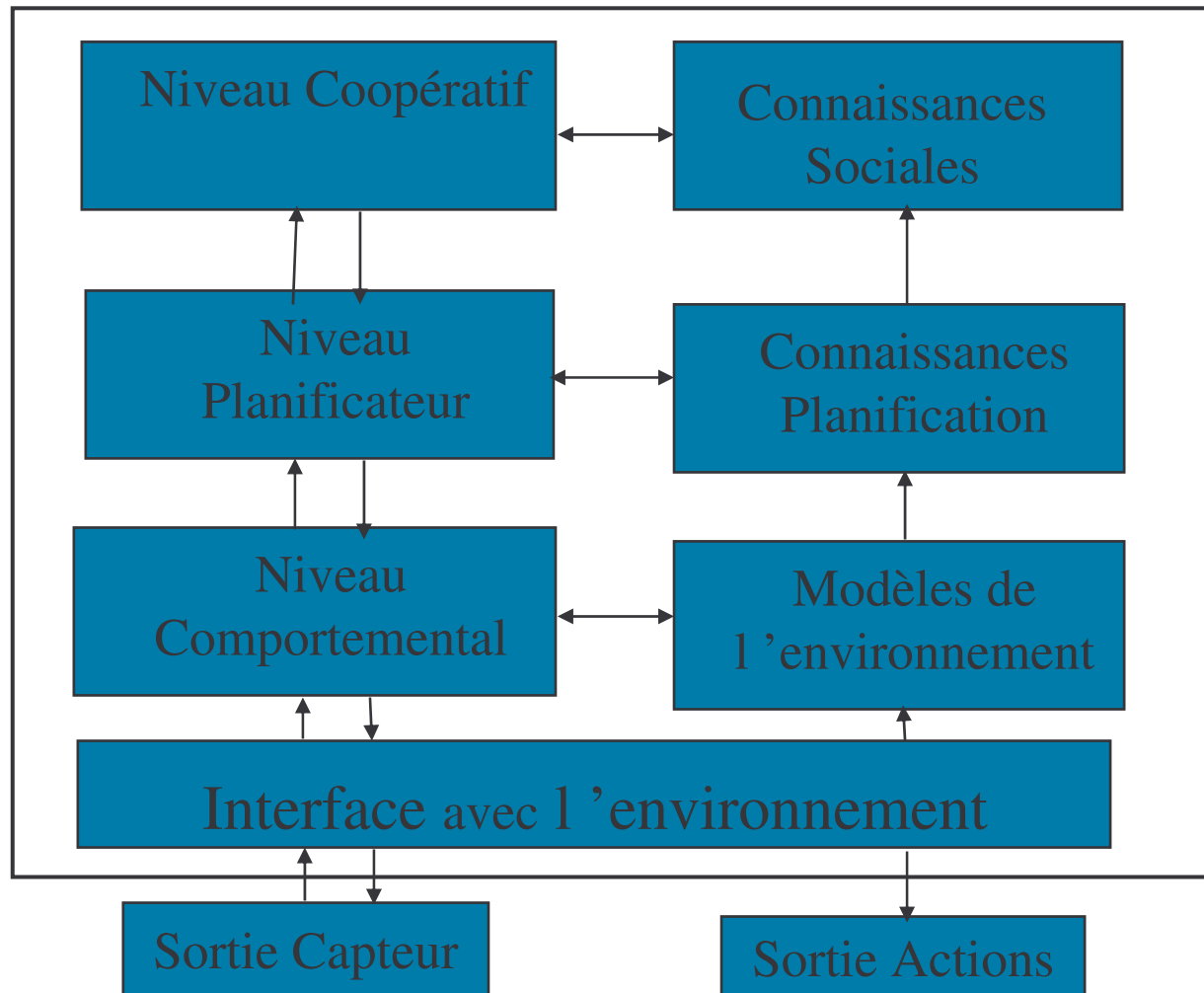


Architecture multi-niveaux

synthèse des approches réactives et pro-actives

- Hiérarchie de niveaux qui interagissent entre eux.
- Deux modes de contrôle :
 - Contrôle horizontal : tous les modules sont directement connectés aux capteurs externes et à la sortie qui déclenche des actions ; chaque module interne à l'agent se comporte comme un agent en proposant des actions à faire.
 - Contrôle vertical : seul un modèle gère les entrées (capteurs) et un autre les sorties (actions à faire).
- Exemple : le modèle Interrap (architecture à contrôle vertical à deux passes).

Architecture Multi-niveaux





Quelques plateformes

- **AgentBuilder**, Resticular Systems, Inc., Java.
- **Aglets**, IBM Japan, Java, Mobile Agents.
- **Concordia**, Mitsubishi Electric, Java, Mobile Agents.
- **Intelligent Agent Factory**, Bits & Pixels, Java.
- **Jack Intelligent Agents**, Agent Oriented Software, Jack agent language.
- **Madkit**, Madkit Developement Group, Java, Scheme et Jess.