

Systemes multi-agents

Cours 4 – Architectures cognitives: Belief–Desire–Intention

Douae AHMADOUN

`douae.ahmadoun@gmail.com` - `douae.ahmadoun@safrangroup.com`

Ingénieure de recherche en IA

Rappel : Agents réactifs

- Pas de représentation explicite (ni d'eux-mêmes, ni des autres, ni de leur environnement)
 - Pas de raisonnement abstrait
- Emergence d'une organisation et d'un comportement "intelligent" d'un ensemble d'agents non-intelligents. L'intelligence n'est explicitée à aucun niveau dans le système

Rappel : Agents réactifs

- Pas de représentation explicite (ni d'eux-mêmes, ni des autres, ni de leur environnement)
 - Pas de raisonnement abstrait
- Emergence d'une organisation et d'un comportement "intelligent" d'un ensemble d'agents non-intelligents. L'intelligence n'est explicitée à aucun niveau dans le système

Avantages de l'approche réactive :

- simplicité
- pas de communication directe
- approche comportementale
- exposition de comportements "intelligents"

Limitations de l'approche réactive :

- intelligence chez le concepteur
- complexe/long à programmer
- gestion des cas imprévus
- généralisation complexe
- gestion des cas nécessitant de la mémoire

INTRODUCTION

Agents cognitifs

Caractéristiques :

- Représentation du monde
- Capabilité de raisonner
- Architecture de type sense \rightarrow plan \rightarrow act

Architectures communes :

- Belief Desire Intention (BDI)
- Agents logiques
- Processus markoviens

Agents cognitifs

Caractéristiques :

- Représentation du monde
- Capabilité de raisonner
- Architecture de type sense \rightarrow plan \rightarrow act

Architectures communes :

- Belief Desire Intention (BDI)
- Agents logiques
- Processus markoviens

Principes de l'architecture BDI

- Basé sur un modèle de la psychologie humaine [1]
- Composé de 4 composants :
 - Beliefs** Croyances de l'agent sur l'état du monde, sur son propre état, sur les autres agents, etc.
 - ▶ Ce que je sais du monde.
 - Desires** Buts de l'agent, ses objectifs généraux.
 - ▶ Ce que je souhaite.
 - Intentions** Buts de l'agent à court terme, objectifs qu'il essaie d'atteindre actuellement.
 - ▶ Ce que je fais.
Plans : Ensemble des plans accessibles à l'agent, composés d'une suite d'actions atomiques.

Logique modale épistémique

- Logique servant à représenter les connaissances/croyances d'un agent
Modalité \neq Fonction de vérité
On ajoute un opérateur représentant la modalité
- $K_i\varphi$: « l'agent i croit à la proposition φ »
Opérateur K : Je sais que (ou je désire, j'ai l'intention)
- Croyance commune d'un groupe d'agents : $K_{i,j}\varphi$

Logique modale épistémique

Axiomatique¹ sur un ensemble de propositions (ou modèle) M :

axiome de distribution **K** $(K_i\phi \wedge K_i(\phi \implies \psi)) \implies K_i\psi$
Modus ponens

axiome de cohérence **D** $\neg(K_i\phi \wedge K_i\neg\phi)$
Je ne me contredis pas

axiome d'introspection positive **4** $K_i\phi \implies K_i(K_i\phi)$
Je sais ce que je sais

axiome d'introspection négative **5** $\neg K_i\phi \implies K_i(\neg K_i\phi)$
Je sais ce que je ne sais pas

→ Système KD54

1. \wedge : et logique ; \implies implication logique ; \neg : non logique

Logique modale épistémique et agent BDI

Un agent BDI possède :

- Des croyances (Beliefs) sur lui-même et le monde
- Des désirs (Desires) éventuellement contradictoires,
- Un ensemble d'intentions (Intentions) non conflictuelles,
- Un mécanisme de raisonnement pour mettre à jour ses croyances, choisir ses désirs et générer de nouvelles intentions.

Modèles B, D et I :

- B : KD45
- D : KD
- I : KD

Logique modale épistémique

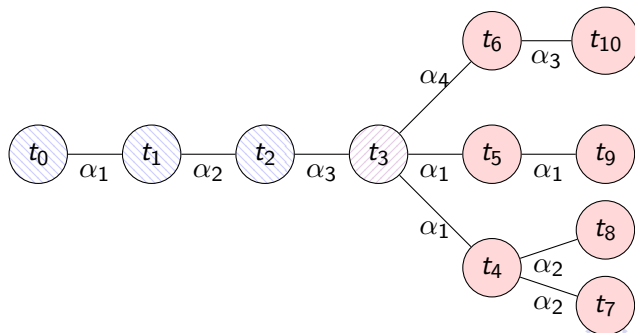
Attention aux confusions !

- « Je désire avoir une bonne note à ce module et avoir une bonne note à ce module implique de travailler donc je désire travailler »
- « Je désire que mes étudiants aient une bonne note et je désire qu'avoir une bonne note veuille dire que l'étudiant a bien travaillé donc je désire que mes étudiants travaillent bien. »
- « Je désire avoir une bonne note à ce module et avoir une bonne note à ce module implique de finir le TP ce soir. Je désire aussi aller au concert de Vadim Repin, et aller à ce concert implique de ne pas finir mon TP ce soir. »

Logic Of Rational Agents (Wooldridge)

Composition de plusieurs types de logiques :

- logique modale
- logique temporelle (ou dynamique) :
 - ▶ temps représenté comme un arbre
 - ▶ passé linéaire
 - ▶ futur représenté par des branchements de possibilités
 - ▶ transitions du monde annotées par des actions



ARCHITECTURES

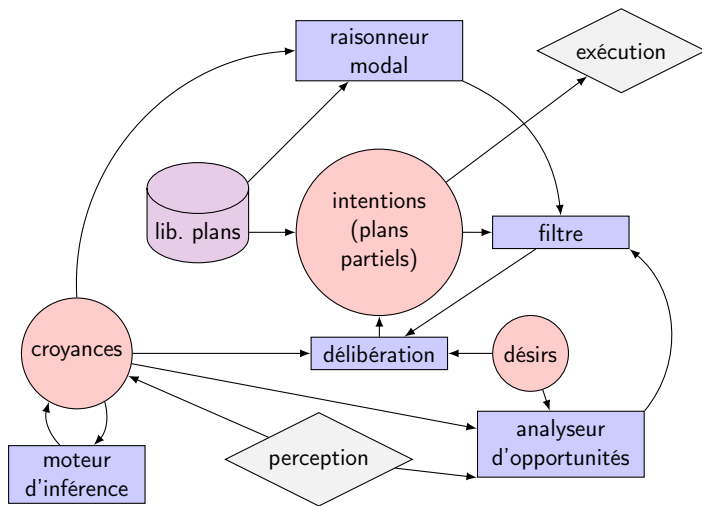
BDI

IRMA (Bratman *et al.*)

Intelligent Resource-bounded Machine Architecture

- Première architecture BDI
- Pas implémentée
- Basée sur des composants et des interconnexions

IRMA (Bratman et al.)



IRMA (Bratman *et al.*)

Composants

révision de croyances met à jour la base de croyances de l'agent en fonction des perceptions reçues. S'assure que la base de croyances reste KD45

délibération génère un ensemble d'intentions et de désirs cohérents à partir des croyances, des désirs et des intentions. S'assure des propriétés de ces ensembles

analyseur d'opportunités détermine les intentions les plus à même d'aboutir, et celles qui n'aboutiront pas

raisonneur modal transforme des intentions en plans partiels

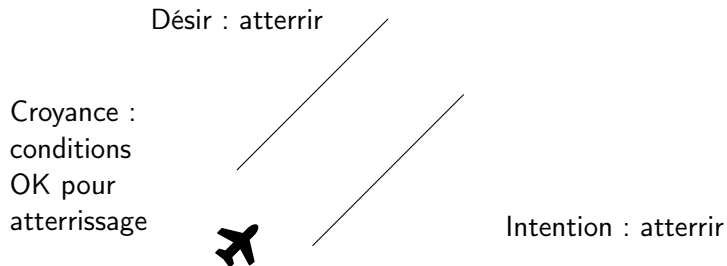
filtre utilise les désirs et les intentions et s'appuie sur l'analyseur d'opportunités et le raisonneur modal pour générer des plans partiels

librairie de plans librairie de plans préétablis ; permet de transformer les plans partiels en plans exécutables

IRMA (Bratman *et al.*)

Intelligent Resource-bounded Machine Architecture

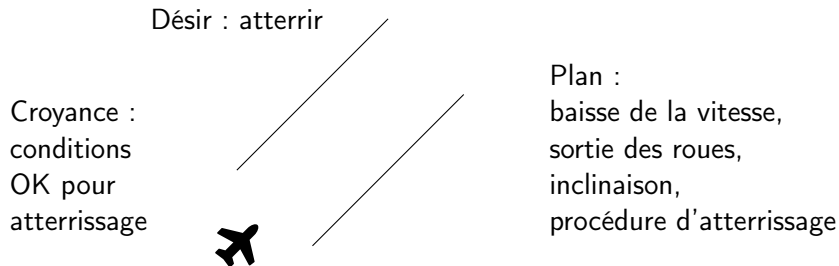
Exemple :



IRMA (Bratman *et al.*)

Intelligent Resource-bounded Machine Architecture

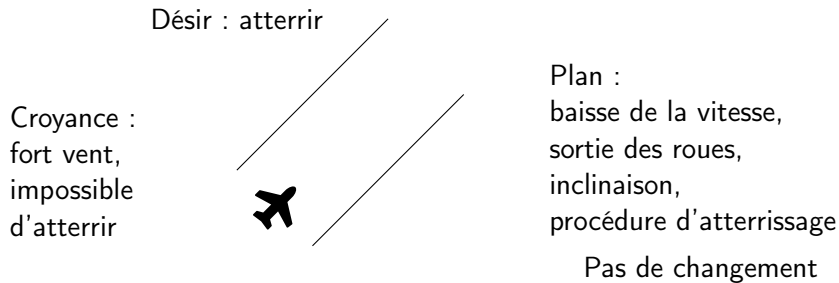
Exemple :



IRMA (Bratman *et al.*)

Intelligent Resource-bounded Machine Architecture

Exemple :

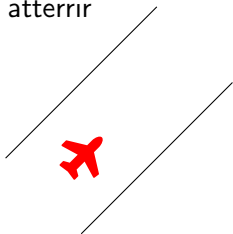


IRMA (Bratman *et al.*)

Intelligent Resource-bounded Machine Architecture

Exemple :

Désir : atterrir



Plan :
baisse de la vitesse,
sortie des roues,
inclinaison,
procédure d'atterrissage

Engagement

Quel problème ? Quelles solutions ?

Engagement

Quel problème ? Quelles solutions ?

- problème : l'agent est « obstiné »
- solution : révision des plans/intentions selon la notion d'*engagement*
- 3 niveaux d'engagement :
 - engagement aveugle** (*blind commitment*) révision lorsque l'intention est atteinte
 - engagement obstiné** (*single-minded commitment*) idem + remise en cause des intentions si l'agent ne les croit plus réalisables
 - engagement ouvert** (*open-minded commitment*) idem + remise en cause des intentions lorsque les désirs ont changé

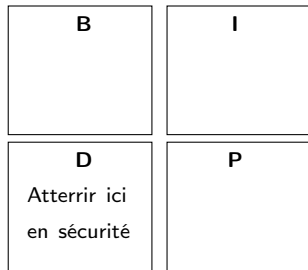
Engagement : la boucle BDI (Wooldridge)

```
1:  $B \leftarrow B_0; D \leftarrow D_0; I \leftarrow I_0$   
2: loop  
3:   Obtenir nlls perceptions  $p$   
4:    $B \leftarrow \text{recv}(B, p)$   
5:    $I \leftarrow \text{options}(D, I)$   
6:    $D \leftarrow \text{des}(B, D, I)$   
7:    $I \leftarrow \text{filtre}(B, D, I)$   
8:    $PE \leftarrow \text{plan}(B, I)$   
9:   Exécuter(PE)  
10: end loop
```

Engagement : la boucle BDI (Wooldridge)

Engagement aveugle

```
 $B \leftarrow B_0; D \leftarrow D_0; I \leftarrow I_0$   
loop  
  Obtenir nlls perceptions  $p$ ●  
   $B \leftarrow \text{recv}(B, p)$   
   $I \leftarrow \text{options}(D, I)$   
   $D \leftarrow \text{des}(B, D, I)$   
   $I \leftarrow \text{filtre}(B, D, I)$   
   $PE \leftarrow \text{plan}(B, I)$   
  Exécuter(PE)  
end loop
```



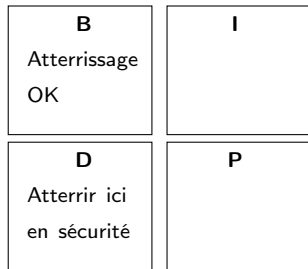
« Att.
OK » ;
Temps clair
Pas de vent



Engagement : la boucle BDI (Wooldridge)

Engagement aveugle

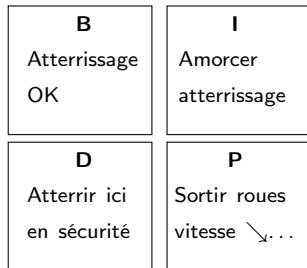
```
 $B \leftarrow B_0; D \leftarrow D_0; I \leftarrow I_0$   
loop  
  Obtenir nles perceptions  $p$   
   $B \leftarrow \text{recv}(B, p) \bullet$   
   $I \leftarrow \text{options}(D, I)$   
   $D \leftarrow \text{des}(B, D, I)$   
   $I \leftarrow \text{filtre}(B, D, I)$   
   $PE \leftarrow \text{plan}(B, I)$   
  Exécuter(PE)  
end loop
```



Engagement : la boucle BDI (Wooldridge)

Engagement aveugle

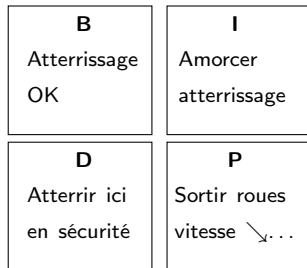
```
 $B \leftarrow B_0; D \leftarrow D_0; I \leftarrow I_0$   
loop  
  Obtenir nles perceptions  $p$   
   $B \leftarrow \text{recv}(B, p)$   
   $I \leftarrow \text{options}(D, I) \bullet$   
   $D \leftarrow \text{des}(B, D, I) \bullet$   
   $I \leftarrow \text{filtre}(B, D, I) \bullet$   
   $PE \leftarrow \text{plan}(B, I) \bullet$   
  Exécuter(PE)  
end loop
```



Engagement : la boucle BDI (Wooldridge)

Engagement aveugle

```
 $B \leftarrow B_0; D \leftarrow D_0; I \leftarrow I_0$   
loop  
  Obtenir nlls perceptions  $p$   
   $B \leftarrow \text{recv}(B, p)$   
   $I \leftarrow \text{options}(D, I)$   
   $D \leftarrow \text{des}(B, D, I)$   
   $I \leftarrow \text{filtre}(B, D, I)$   
   $PE \leftarrow \text{plan}(B, I)$   
  Exécuter( $PE$ )•  
end loop
```



« Att.
ailleurs »



Engagement : la boucle BDI (Wooldridge)

Engagement obstiné

$B \leftarrow B_0; D \leftarrow D_0; I \leftarrow I_0$

loop

Obtenir nlls perceptions p

$B \leftarrow \text{recv}(B, p)$

$I \leftarrow \text{options}(D, I)$

$D \leftarrow \text{des}(B, D, I)$

$I \leftarrow \text{filtre}(B, D, I)$

$PE \leftarrow \text{plan}(B, I)$

while ($PE \neq \emptyset \wedge \text{possible}(I, B)$) • **do**

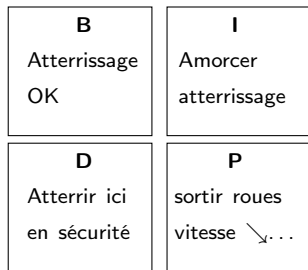
$x \leftarrow PE.\text{pop}(); \text{exec}(x)$

obtenir nlls perceptions p

$B \leftarrow \text{recv}(B, p)$

end while

end loop



« Att.
ailleurs »



Engagement : la boucle BDI (Wooldridge)

Engagement obstiné

$B \leftarrow B_0; D \leftarrow D_0; I \leftarrow I_0$

loop

Obtenir nlls perceptions p

$B \leftarrow \text{recv}(B, p)$

$I \leftarrow \text{options}(D, I)$

$D \leftarrow \text{des}(B, D, I)$

$I \leftarrow \text{filtre}(B, D, I)$

$PE \leftarrow \text{plan}(B, I)$

while ($PE \neq \emptyset \wedge \text{possible}(I, B)$) **do**

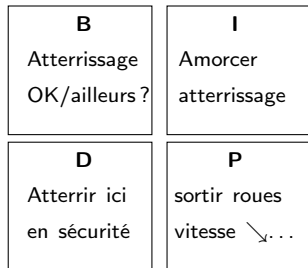
$x \leftarrow PE.\text{pop}(); \text{exec}(x)$ •

obtenir nlls perceptions p •

$B \leftarrow \text{revc}(B, p)$ •

end while

end loop



Engagement : la boucle BDI (Wooldridge)

Engagement obstiné

$B \leftarrow B_0; D \leftarrow D_0; I \leftarrow I_0$

loop•

Obtenir nlls perceptions p

$B \leftarrow \text{recv}(B, p)$

$I \leftarrow \text{options}(D, I)$

$D \leftarrow \text{des}(B, D, I)$

$I \leftarrow \text{filtre}(B, D, I)$

$PE \leftarrow \text{plan}(B, I)$

while ($PE \neq \emptyset \wedge \text{possible}(I, B)$)• **do**

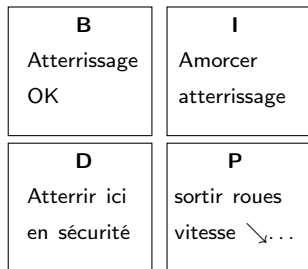
$x \leftarrow PE.\text{pop}(); \text{exec}(x)$

obtenir nlls perceptions p

$B \leftarrow \text{recv}(B, p)$

end while

end loop



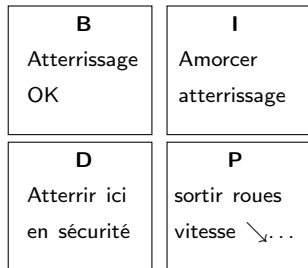
Fort vent
orage



Engagement : la boucle BDI (Wooldridge)

Engagement ouvert

```
while ( $PE \neq \emptyset \wedge possible(I, B)$ ) do●  
   $x \leftarrow PE.pop()$ ;  $exec(x)$   
  obtenir nlls perceptions  $p$   
   $B \leftarrow revc(B, p)$   
   $D \leftarrow des(B, D, I)$   
   $I \leftarrow filtre(B, D, I)$   
   $PE \leftarrow plan(B, I)$   
end while
```



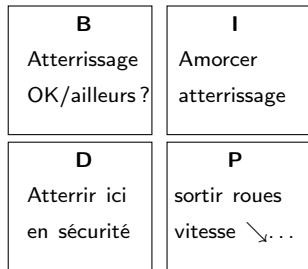
« Att.
ailleurs »



Engagement : la boucle BDI (Wooldridge)

Engagement ouvert

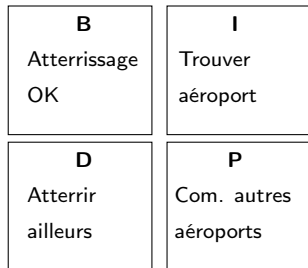
```
while ( $PE \neq \emptyset \wedge possible(I, B)$ ) do
   $x \leftarrow PE.pop()$ ;  $exec(x)$ •
  obtenir nlls perceptions  $p$ •
   $B \leftarrow revc(B, p)$ •
   $D \leftarrow des(B, D, I)$ 
   $I \leftarrow filtre(B, D, I)$ 
   $PE \leftarrow plan(B, I)$ 
end while
```



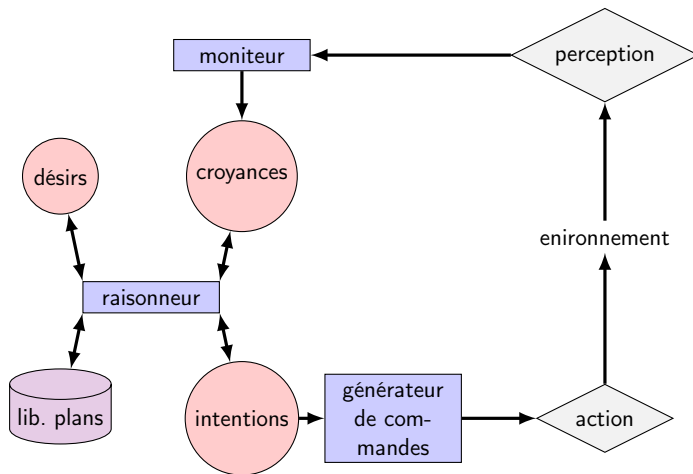
Engagement : la boucle BDI (Wooldridge)

Engagement ouvert

```
while ( $PE \neq \emptyset \wedge possible(I, B)$ ) do  
   $x \leftarrow PE.pop()$ ;  $exec(x)$   
  obtenir nlls perceptions  $p$   
   $B \leftarrow revc(B, p)$   
   $D \leftarrow des(B, D, I)$ •  
   $I \leftarrow filtre(B, D, I)$ •  
   $PE \leftarrow plan(B, I)$ •  
end while
```



Engagement : Procedural Reasoning System (Georgeff & Lansky)



Langages orientés BDI I

AgentSpeak langage de programmation *abstrait* basé sur PRS

dMARS Distributed Multi-Agent Reasoning System

- Langage de développement BDI basé sur PRS,
- Développé en C++,
- Plus supporté aujourd'hui (*cf.* JACK).

JACK Plate-forme propriétaire de Agent Oriented Software Ltd. Caractéristiques :

- Inspiré de PRS et dMars
- Domain Specific Language (DSL) compilé en Java
- IDE fourni (facilités de conception)
- Noyau avec fonctionnalités de base (messages, raisonnement...)
- Extensions : FIPA, déploiement sur serveur...

Jason Implémentation Java de AgentSpeak

- Opensource (GPL/LGPL)
- Extension de AgentSpeak plus pratique

Langages orientés BDI II

- Interconnexion possible avec d'autres frameworks (JADE)

Jadex Extension de JADE implémentant le modèle BDI

- Développement Java + XML
- Embarque une interface et un IDE
- Opensource LGPL
- Inclut toutes les fonctionnalités de JADE

2APL A Practical Agent Programming Language

- Domain Specific Language + Java
- IDE (plugin Eclipse)
- Opensource (GPL 3)

GOAL Langage orienté but

- Domain Specific Language (semblable à PROLOG)
- IDE (plugin Eclipse)
- Opensource







Conclusion

- Avantages de l'approche BDI
 - ▶ pratique : agent capable de raisonner
 - ▶ « proche de l'humain » (basé sur une théorie de la cognition humaine)
 - ▶ explicable, prédictible
 - ▶ basé sur une axiomatique riche
 - ▶ modèle générique de haut niveau
- Inconvénients de l'approche BDI
 - ▶ pas de notion d'apprentissage
 - ▶ beaucoup de calibration à la main
 - ▶ beaucoup de calcul
 - ▶ comment l'implémenter efficacement ?

Exemple avec GAMA

https://www.youtube.com/watch?v=9Feb_pVfnpg&ab_channel=GamaPlatform




Références I

-  Michael E Bratman. *Intention, Plans, and Practical Reason*, Harvard University Press, 1987.
-  Michael Wooldridge. *Reasoning about rational agents* MIT Press, 2000.
-  Rafael H Bordini, et Jomi F Hübner. “BDI agent programming in AgentSpeak using Jason.” *International Workshop on Computational Logic in Multi-Agent Systems*, pp. 143-164, 2005.
-  Michael E Bratman, David J Israel et Martha E Pollack. “Plans and resource-bounded practical reasoning”. *Computational Intelligence* 4.4 pp 349-355, 1988.
-  Mehdi Dastani. “2APL : a practical agent programming language.” *Autonomous agents and multi-agent systems* 16.3, pp 214-248, 2008.
-  Michel P Georgeff, Amy L Lansky. “Procedural knowledge”. *Proceedings of the IEEE* 74.10 pp 1383-1398, 1986.

Références II

-  Michel P Georgeff et François-Félix Ingrand. "Decision making in an embedded reasoning system". *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence* pp 972-978, 1989.
-  Koen V. Hindriks "Programming rational agents in goal." *Multi-Agent Programming*, pp. 119-157, 2009.
-  Nick Howden, Ralph Rönquist, Andrew Hodgson, et Andrew Lucas. "JACK intelligent agents-summary of an agent infrastructure." *5th International conference on autonomous agents*, 2001.
-  Mark d'Inverno, David Kinny, Michael Luck et Michael Wooldridge. "A formal specification of dMARS." *International Workshop on Agent Theories, Architectures, and Languages*, pp 155-176, 1997.
-  Alexander Pokahr, Lars Braubach, et Winfried Lamersdorf. "Jadex : A BDI reasoning engine." *Multi-agent programming* pp 149-174, 2005.

Références III

-  Anand S Rao et Michael P Georgeff. "BDI agents : from theory to practice". *Proceedings of the first international conference on multiagent systems* pp 312-319, 1995.
-  Anand S Rao "AgentSpeak (L) : BDI agents speak out in a logical computable language." *European Workshop on Modelling Autonomous Agents in a Multi-Agent World*, pp 42-55, 1996.
-  M S Smitha Rao, A M Jyothsna. "BDI : Applications and Architectures". *IJERT 2013 2.2*, 2013.