

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/>

Communication entre agents



Plan

- Comment communiquer ?
- Communication indirecte
- Langages de communication entre agents: KQML, FIPA
- Le contenu de la communication
- Protocoles d'interaction

1. Comment communiquer ?

Communication humaine

La communication est l'échange intentionnel des informations obtenues par la production et perception des signes appartenant à un système partagé de signes conventionnels (AIMA, Russell&Norvig) → *langage*

La communication est vue comme une action (acte de communication) et comme une attitude intentionnelle

Composantes de la communication

Émetteur

- ⇔ Intention
- ⇔ Génération
- ⇔ Synthèse

Récepteur

- ⇒ Perception
- ⇒ Analyse
- ⇒ Désambiguïsation
- ⇒ Incorporation

→ { Syntaxe
Sémantique
Pragmatique

Communication Artificielle

- langages de bas niveau vs langages de haut niveau
- communication directe vs. communication indirecte

Communication au niveau des ordinateurs

- mémoire partagée
- transmissions des messages

Communication entre agents / Communication SMA

- ➔ Communication de bas niveau: signaux simples, traces, langages de bas niveau
- ➔ Communication de haut niveau – agents cognitifs, vus comme des systèmes intensionnels
- ➔ Communication dans les SMA = plus que la simple communication, implique l'interaction
- ➔ L'environnement offre une infrastructure où les interactions se déroulent. Cette infrastructure comprend des protocoles d'interaction

Protocoles de communication = permettent aux agents d'échanger et comprendre les messages

Protocoles d'interaction = permettent aux agents les échanges structurés des messages

But → La communication permet aux agents de:

- coordonner leurs actions et comportement, une propriété d'un SMA avec des agents agissant dans un environnement commun
- essayer de modifier les états des autres agents
- essayer de persuader les autres agents de faire des certaines actions

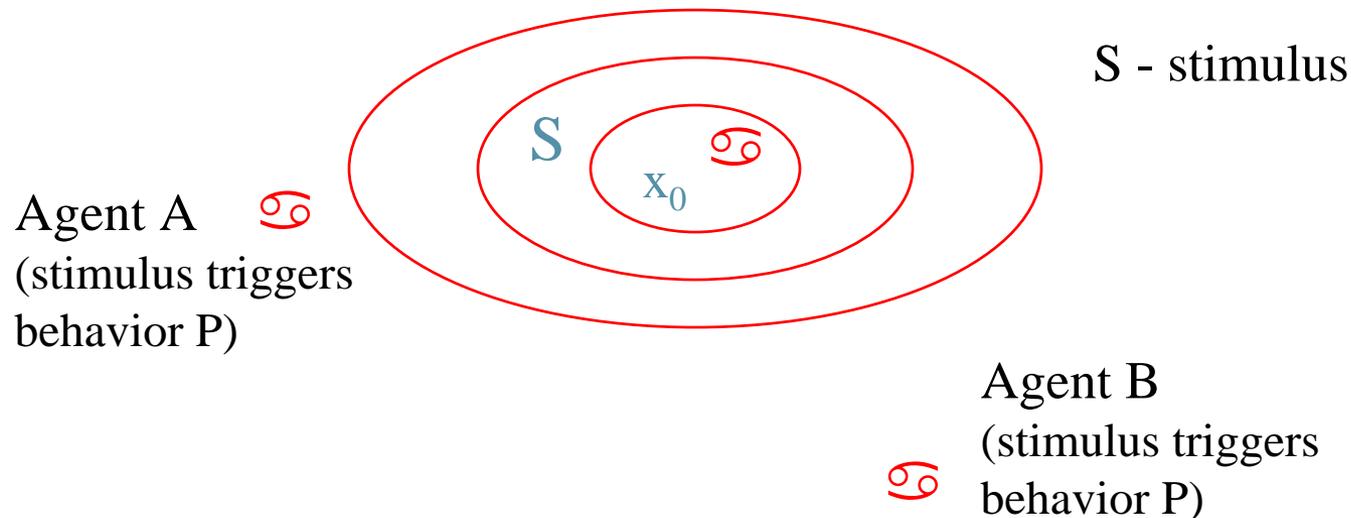
2. Communication indirecte

2.1 Propagation des signaux - Manta, A. Drogoul 1993

- Un agent émet un signal qui est transmis dans l'environnement; l'intensité du signal décroît par rapport à la distance
- Dans un point x , un signal peut avoir une des intensités suivantes:

$$V(x) = V(x_0) / \text{dist}(x, x_0)$$

$$V(x) = V(x_0) / \text{dist}(x, x_0)^2$$

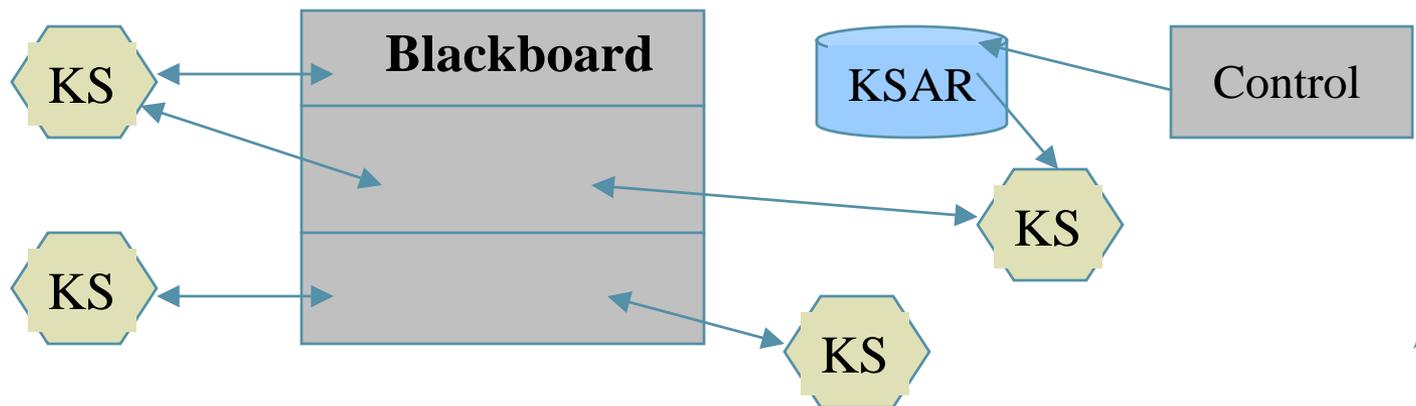


2.2 Traces - L. Steels, 1995

- un agent laisse tomber des "morceaux radioactifs" en faisant ainsi des traces
- un agent suit une trace fait diminuer la trace jusqu'à la disparition de la trace

2.3 Systèmes Tableaux Noirs, Barbara Hayes-Roth, 1985

- "Blackboard" = une zone commune (mémoire partagée) dans laquelle les agents peuvent échanger des informations et connaissances
- Les agents commencent la communication en écrivant des informations sur le tableau noir
- Les agents cherchent des nouvelles infos sur le "blackboard", et peuvent filtrer ces infos
- Les agents doivent s'inscrire à un site central pour recevoir l'autorisation d'écrire sur le tableau noir
- Blackboard = un paradigme de calcul puissant pour la résolution distribuée des problèmes
- Agents = Sources de connaissances (KS = 'Knowledge Sources')



3. Communication directe

Envoie des messages

- ❑ appel des méthodes – Le langage Acteurs ("Actors")
- ❑ échange des plans partiels – coordination des agents coopératifs

Langages de communication entre agents

ACL = Agent Communication Languages

La communication est vue comme une action –
actes de communication

3.1 Le model Acteur

Proposé par Agha de l'Université de Illinois at Urbana-Champaign.

- Acteur
 - Acteur = un objet avec son propre fil d'exécution
 - La communication est réalisée par transmission asynchrone des messages

- **Primitives de Acteurs**

send(a,v)

- a = récepteur
- v = contenu

newactor(e)

- e = expression à évaluer
- retourne l'adresse du nouvel acteur

ready(b)

- b = comportement (état nouveau)

Peut être ajouté à plusieurs langages: Lisp, Java, etc.

Primitives "Actor"

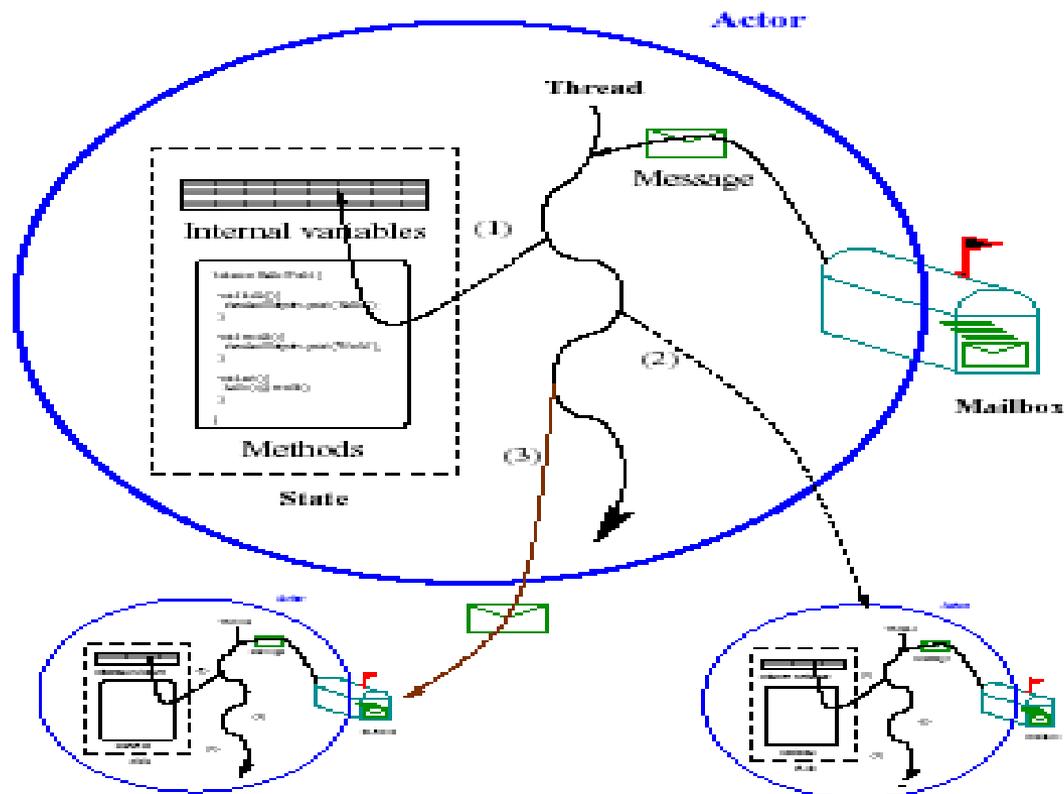


Fig. 1.1. In response to a message, an actor can: (1) modify its local state, or (2) create new actors, or (3) send messages to acquaintances.

3.2 Langages de communication entre agents

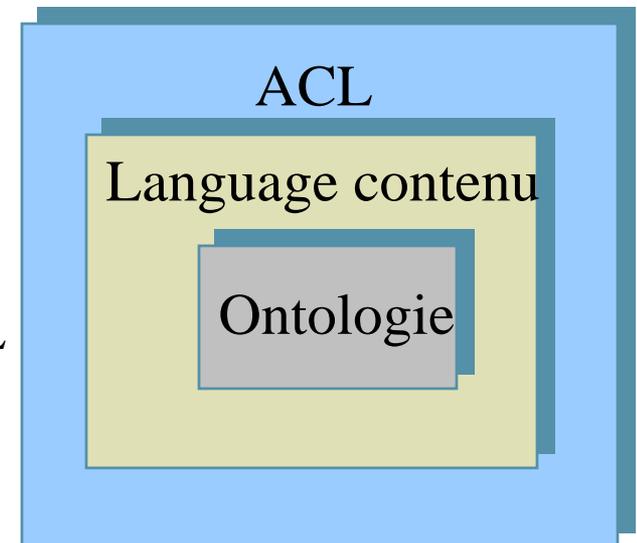
Concepts (qui font la différence par rapport aux langages comme RPC, RMI or CORBA, ORB):

- ACLs peuvent communiquer des propositions, des règles et des actions et pas uniquement des objets sans une sémantique associée
- Un message ACL décrit un état désiré dans un langage déclaratif et n'est pas une simple invocation de méthode
- Les ACLs ne sont fondés sur la théorie BDI: les agents BDI essaient de communiquer leurs états BDI et de modifier les états BDI des autres agents
- Les ACLs partent de la théorie des actes de langage (ou de discours) ("Speech Act Theory")
- La communication peut être structurée dans des conversations

Origine des ACLs

Knowledge Sharing Effort - DARPA, 1990

- External Interface Group - interaction entre KBS - KQML
- Interlingua - langage commun de la BC - KIF
- Shared, Reusable Knowledge Bases - Ontolingua



La théorie des actes de langage

(Theory of Speech Acts)

J. Austin - How to do things with words, 1962, J. Searle - Speech acts, 1969

Un acte de langage a trois aspects

actes locutoires = accompli quand on formule correctement un énoncé

actes illocutoires = expriment une intention du locuteur (performatifs)

actes perlocutoires expriment les effets sur l'allocutaire visée par les actes locutoires et illocutoires

Jacqueline a dit à Paul: "Tu dois fermer la porte"



locution



illocution



contenu

prelocution: porte fermée (on espère!)

La théorie des actes de langage

Austin a classé les performatifs dans les catégories suivantes :

- **verdictifs** (*acquitter, condamner, prononcer, décréter, classer, évaluer, ...*)
 - **exercitifs** (*commander, dégrader, ordonner, léguer, pardonner, ...*)
 - **promissifs** (*promettre, faire vœu de, garantir, parier, jurer de, ...*)
 - **comportatifs** (*s'excuser, remercier, déplorer, critiquer, braver, ...*)
 - **expositifs** (*affirmer, conclure, nier, postuler, remarquer, ...*)
-
- Les verbes illustrant ces catégories n'ont qu'une valeur indicative - performatifs implicites - dire *je viendrai demain* équivaut souvent à dire : *je promets de venir demain*
 - Les utiliser ne suffit pas à conférer une valeur performative – il faut vérifier les conditions dites de félicité: certaines conditions doivent être remplies pour que l'énoncé de ces phrases emporte leur vérité.
 - Les verbes peuvent être ordonnés selon leur force illocutoire : p.ex. *suggérer* < *préconiser* < *demander* < *réclamer* < *ordonner*.

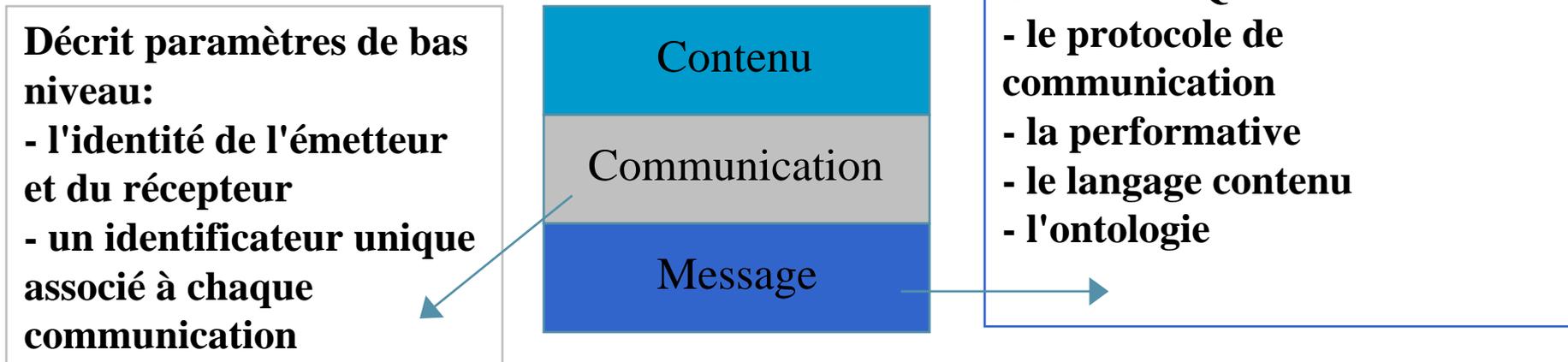
3.3 KQML - Knowledge Query and Manipulation Language

Un langage de communication de haut niveau orienté messages et un protocole pour l'échange des informations qui est indépendant de la syntaxe du contenu (KIF, SQL, Prolog,...) et de l'ontologie de l'application

KQML sépare:

- la sémantique du protocole de communication (indépendante de domaine)
- la sémantique du message (dépendante de domaine)

3 couches conceptuelles



Syntax des S-expressions utilisées en LISP

(**ask-one** :sender joe
 :receiver ibm-stock
 :reply-with ibm-stock
 :language PROLOG
 :ontology NYSE-TICKS
 :content (price ibm ?price))

(**tell** :sender willie
 :receiver joe
 :reply-with block1
 :language KIF
 :ontology BlockWorld
 :content (AND (Block

A)(Block B)

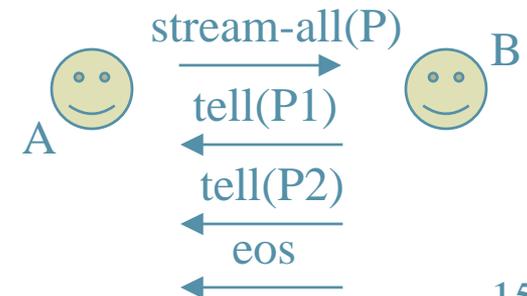
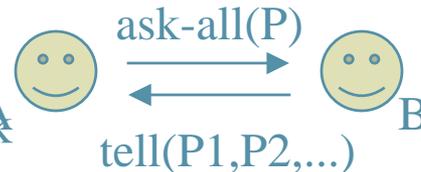
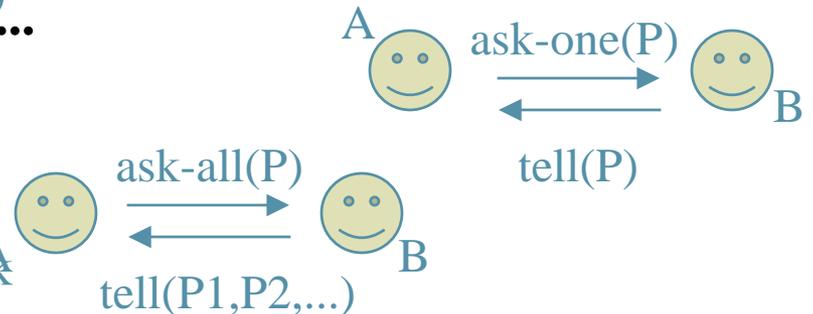
1. Performatives de question:

ask-one, ask-all, ask-if, stream-all,...

(On A

(**stream-all** :sender willie
 :receiver ibm-stock
 :content (price ?VL ?price))

(**standby** :content (**stream-all**
 :content (price ?VL
 ?price))



2. Performatives de génération:

standby, ready, next, rest, discard, generate,...

3. Performatives de réponse:

reply, sorry ...

4. Performatives expositives:

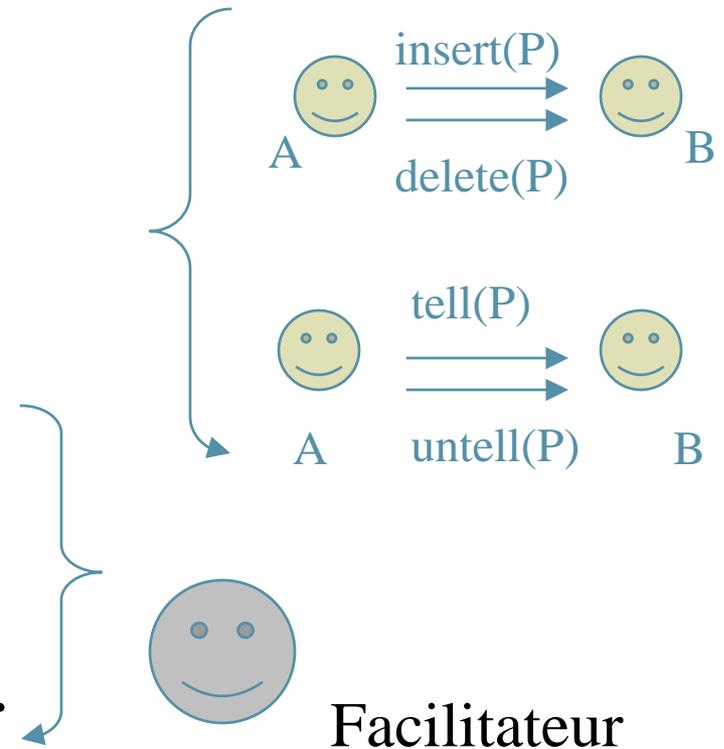
tell, untell, insert, delete, ...

5. Performatives de capacité:

advertise, subscribe, recommend...

6. Performatives de réseau:

register, unregister, forward, route, ...



KQML – 2 types d'actes illocutoires:

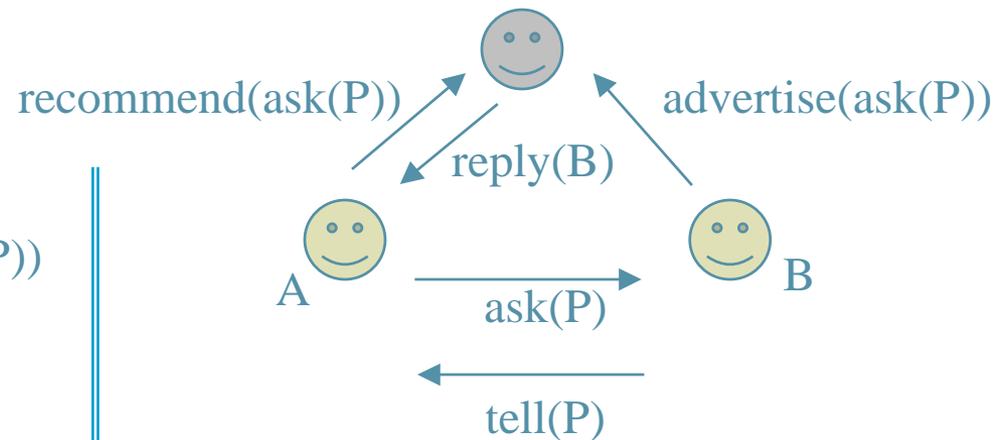
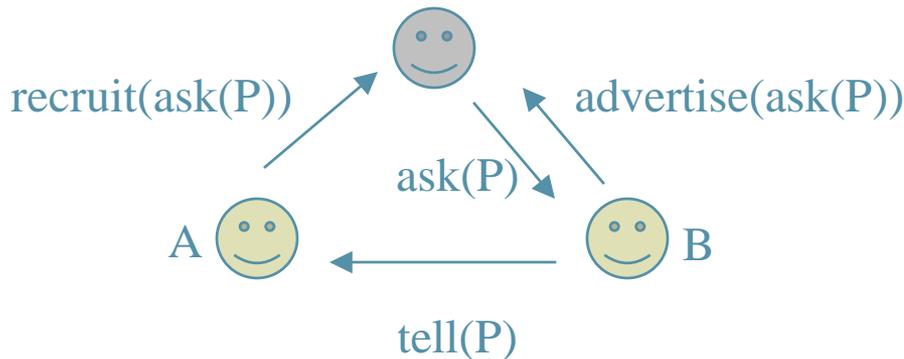
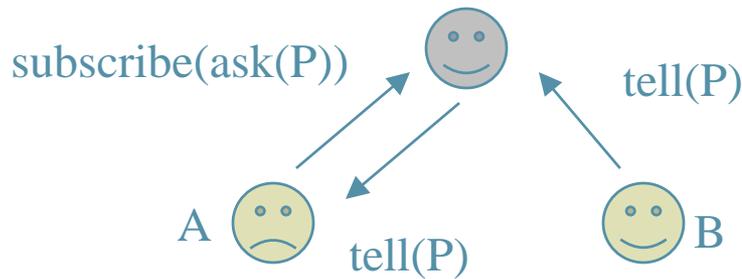
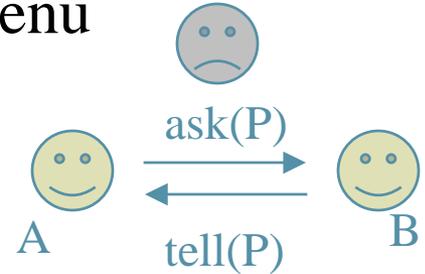
expositifs et exercitifs

+ performatives liées au facilitateur et au réseau

Agent Facilitateur



- Agent Name Server
- transmettre les messages to autres agents par ANS
- transmettre les messages to autres agents par contenu
- couplage des offres et requêtes
- services de médiation et de translation



Sémantique de KQML (Labrou & Finin)

tell(A,B,X)

A states to B that A believes the content X to be true, $\text{Bel}(A,X)$

Pre(A): $\text{Bel}(A,X) \wedge \text{Know}(A, \text{Want}(B, \text{Know}(B,S)))$

Pre(B): $\text{Int}(B, \text{Know}(B,S))$

where S may be any of $\text{Bel}(B,X)$ or $\neg\text{Bel}(B,X)$

Post(A): $\text{Know}(A, \text{Know}(B, \text{Bel}(A,X)))$

Post(B): $\text{Know}(B, \text{Bel}(A,X))$

Completion: $\text{Know}(B, \text{Bel}(A,X))$

advertise(A,B,M)

A states to B that A can and will process the message M from B, if it receives one

$\text{Int}(A, \text{Proc}(A,M))$

Pre(A): $\text{Int}(\text{Proc}(A,M))$

Pre(B): NONE

Post(A): $\text{Know}(A, \text{Know}(B, \text{Int}(A, \text{Proc}(A,M))))$

Post(B): $\text{Know}(B, \text{Int}(A, \text{Proc}(A,M)))$

Completion: $\text{Know}(B, \text{Int}(A, \text{Proc}(A,M)))$

3.4 FIPA - ACL

Foundation for Intelligent Physical Agents, 1996

- But de FIPA = offrir une spécification qui permet l'interopérabilité maximale entre agents et SMA
- Les comités de FIPA: ACL, agent specification, agent-software interaction
- FIPA est basée sur la théorie des actes de langage
- actes de communication (CA) au lieu de performatives
- sémantique différente

(inform :sender Agent1
:receiver Agent2
:content (price good2 150)
:in-reply-to round-1
: reply-with bid03
: language S1
:ontology hp-auction
:reply-by 10
:protocol offer
:conversation-id conv-1)

FIPA actes de communication

Informatives

- query_if, subscribe, inform, inform_if
confirm, disconfirm, not_understood

Distribution des tâches

- request, request_whenver, cancel, agree,
refuse, failure

Negotiation

- cfp, propose, accept_proposal
reject_proposal

Sémantique de FIPA-ACL

SL (*Semantic Language*) - une logique modale

croyances

croyances incertaines

désirs

intentions

B φ - belief **D** φ - desire **U** φ - uncertain belief

PG φ - intention

Bif φ - express whether an agent has a definite opinion one way or another about the truth or falsity of φ

Uif φ - the agent is uncertain about φ

La sémantique d'un acte de communication (CA) est spécifiée comme un ensemble des formules en SL qui décrivent:

- La precondition de faisabilité – les conditions nécessaires pour l'émetteur; l'émetteur n'est pas obligé à faire le CA
- L'effet rationnel – l'effet que l'agent peut s'attendre à suite de la CA; indique les conditions qui doivent être vraies pour le récepteur
- Le récepteur n'est pas obligé à assurer l'effet
- L'émetteur ne peut pas être sur que l'effet sera accompli

$\langle i, \text{inform}(j, \varphi) \rangle$

Pre: $B_i \varphi \wedge \neg B_i (Bif_j \varphi \vee Uif_j \varphi)$

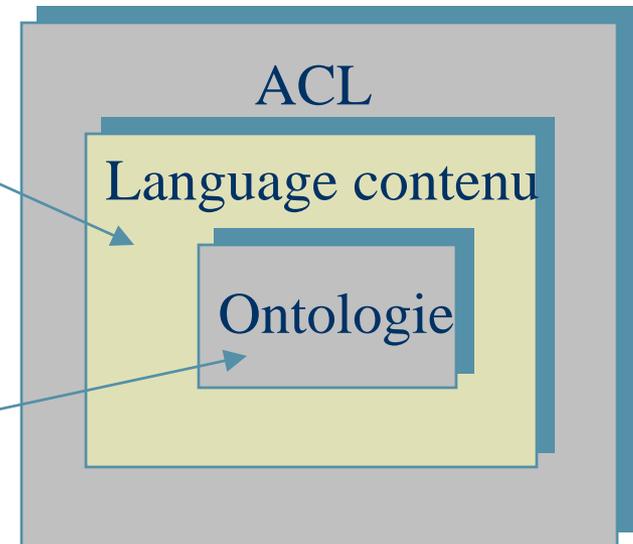
Post: $B_j \varphi$

4. Contenu de la communication

■ Langages pour le contenu

- KIF
- Prolog
- Clips
- SQL
- FIPA-SL, FIPA-CCL, FIPA-KIF
- DAML

■ Ontologies



DARPA Agent Markup Language

The DAML Program officially began in August 2000. The goal of the DAML effort is to develop a language and tools to facilitate the concept of the Semantic Web.

4.1 Langages pour le contenu

- **KIF - Knowledge Interchange Format**

(salary ?x ?y ?z)

(interested joe '(salary ,?x ,?y ,?z))

- **FIPA-SL**

(**request** :sender (agent-identifier :name i)

:receiver (set (agent-identifier :name j)

:content ((action (agent-identifier :name j)

(deliver box7 (loc 10 15))))

:protocol fipa-request

:language fipa-sl

:reply-with order56)

(**agree** :sender (agent-identifier :name j)

:receiver (set (agent-identifier :name i)

:content ((action (agent-identifier :name j)

(deliver box7 (loc 10 15))) (priority

order56 low))

:protocol fipa-request

:language fipa-sl

:in-reply-to order56)

4.2 Ontologies

Ontology = une spécification des objets, concepts et relations dans un domaine particulier;

- une ontologie comprend un schéma conceptuel pour décrire l'organisation et l'interprétation du domaine

- une ontologie est plus qu'une taxonomie des classes – elle peut comprendre en plus des axiomes

$$\forall x (\text{Block } x) \Rightarrow (\text{PhysicalObject } x)$$

Au lieu de (Block A)

(InstanceOf A Block)

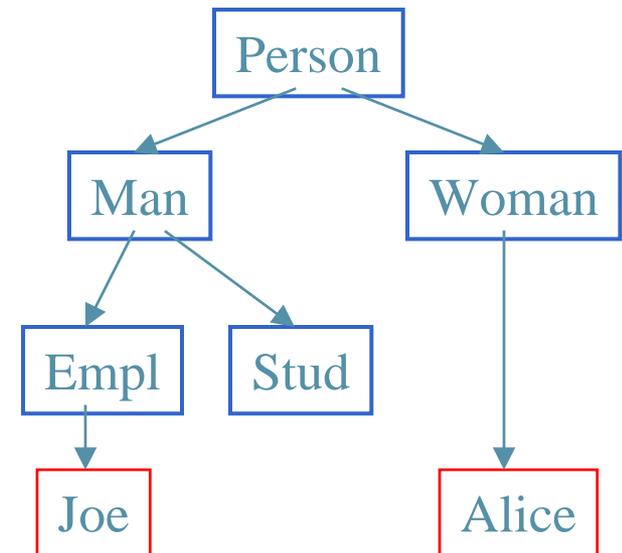
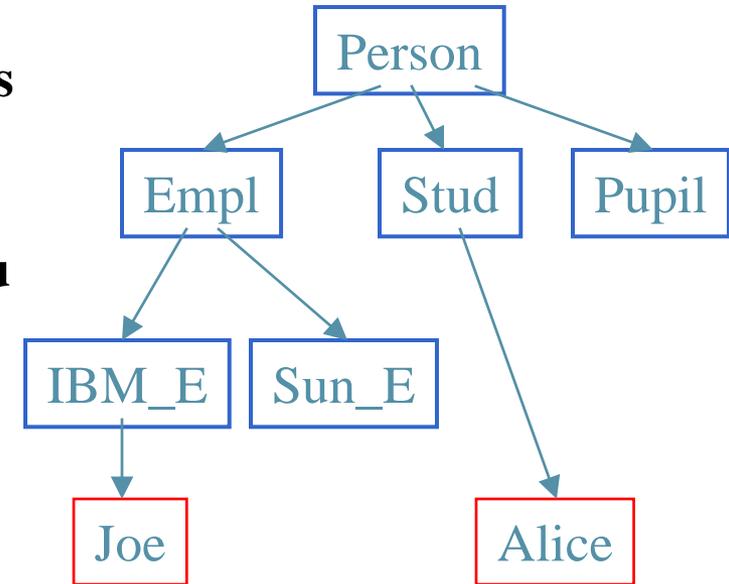
(Class Block)

Hiérarchie

(Class PhysicalObjects)

(SubclassOf Block PhysicalObjects)

$$\forall x,y,z (\text{InstanceOf } x y) \wedge (\text{SubclassOf } y z) \Rightarrow (\text{InstanceOf } x z)$$



Ontologies



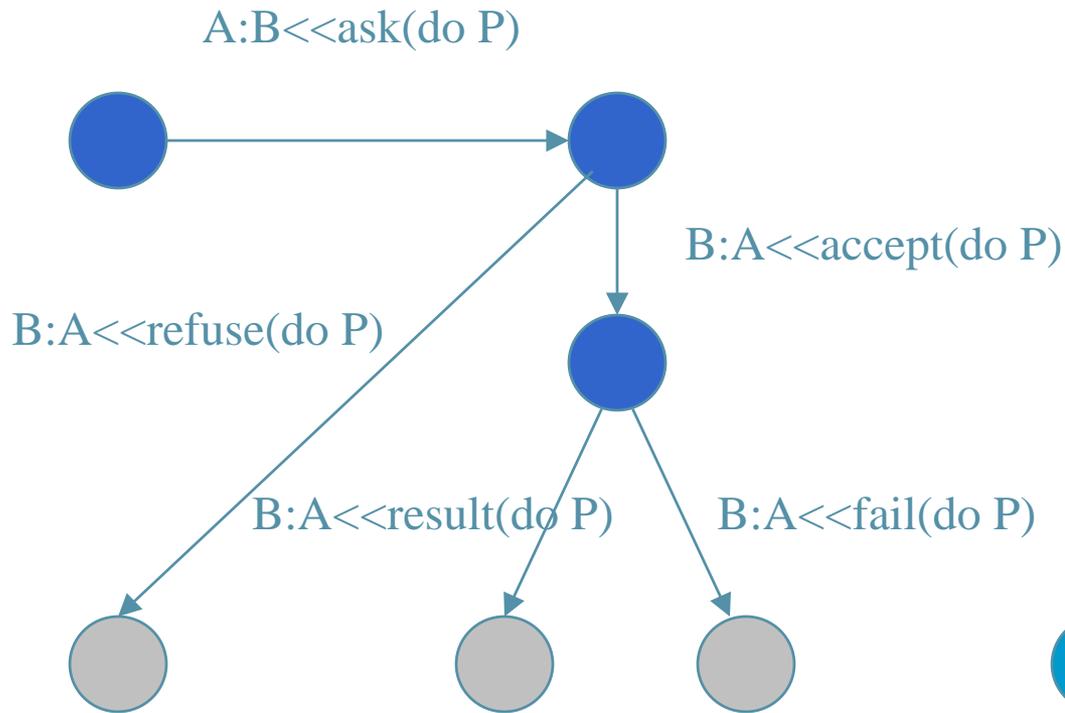
- Ontologies lexicalisées (WordNet, EuroWordNet, BalkanNet, FrameNet, MikroKosmos).
- Ontologies pour la représentation des connaissances
- Ontologies implicites, par exemple les librairies de classes dans les LOO

5. Protocoles d'interaction

Protocoles d'interaction = permettent aux agents d'avoir des conversations (échange structuré des messages)

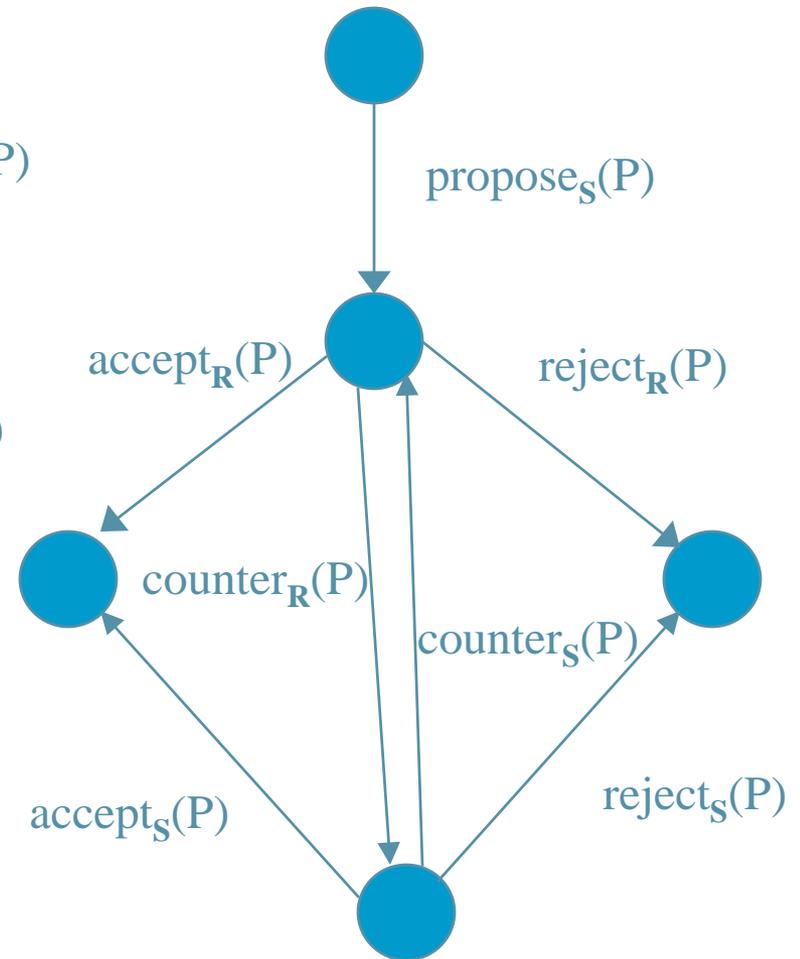
- **Automates finis**
- **Conversations en KQML**
- **Réseaux de Petri**
- **FIPA IP standards:**
 - **FIPA-query, FIPA-request, FIPA-contract-net, ...**

5.1 Automates finis



Winograd, Flores, 1986

COOL, Barbuceanu, 95



5.2 Conversations en KQML

Definite Clause Grammars (DCG)

- utilisées pour la spécification des conversations en KQML
- DCGs étendent les grammaires indépendantes de contexte non-terminaux peuvent être des termes composés
- la règle peut contenir des appels des fonctions ou des conditions à tester

Ex: $\text{noun}(N) \rightarrow [W], \{\text{RootForm}(W,N), \text{is_noun}(N)\}$

$S \rightarrow \mathbf{s}(\text{Conv}, P, S, R, \text{inR}, R_w, \text{IO}, \text{Content}), \{\text{member}(P, [\text{advertise}, \text{ask-if}])\}$

$\mathbf{s}(\text{Conv}, \text{ask-if}, S, R, \text{inR}, R_w, \text{IO}, \text{Content}) \rightarrow$

$[\text{ask-if}, S, R, \text{inR}, R_w, \text{IO}, \text{Content}] \mid$

$[\text{ask-if}, S, R, \text{inR}, R_w, \text{IO}, \text{Content}], \{\text{OI is inv}(\text{IO})\},$

$\mathbf{r}(\text{Conv}, \text{ask-if}, S, R, _, R_w, \text{OI}, \text{Content})$

$\mathbf{r}(\text{Conv}, \text{ask-if}, R, S, _, \text{inR}, \text{IO}, \text{Content}) \rightarrow$

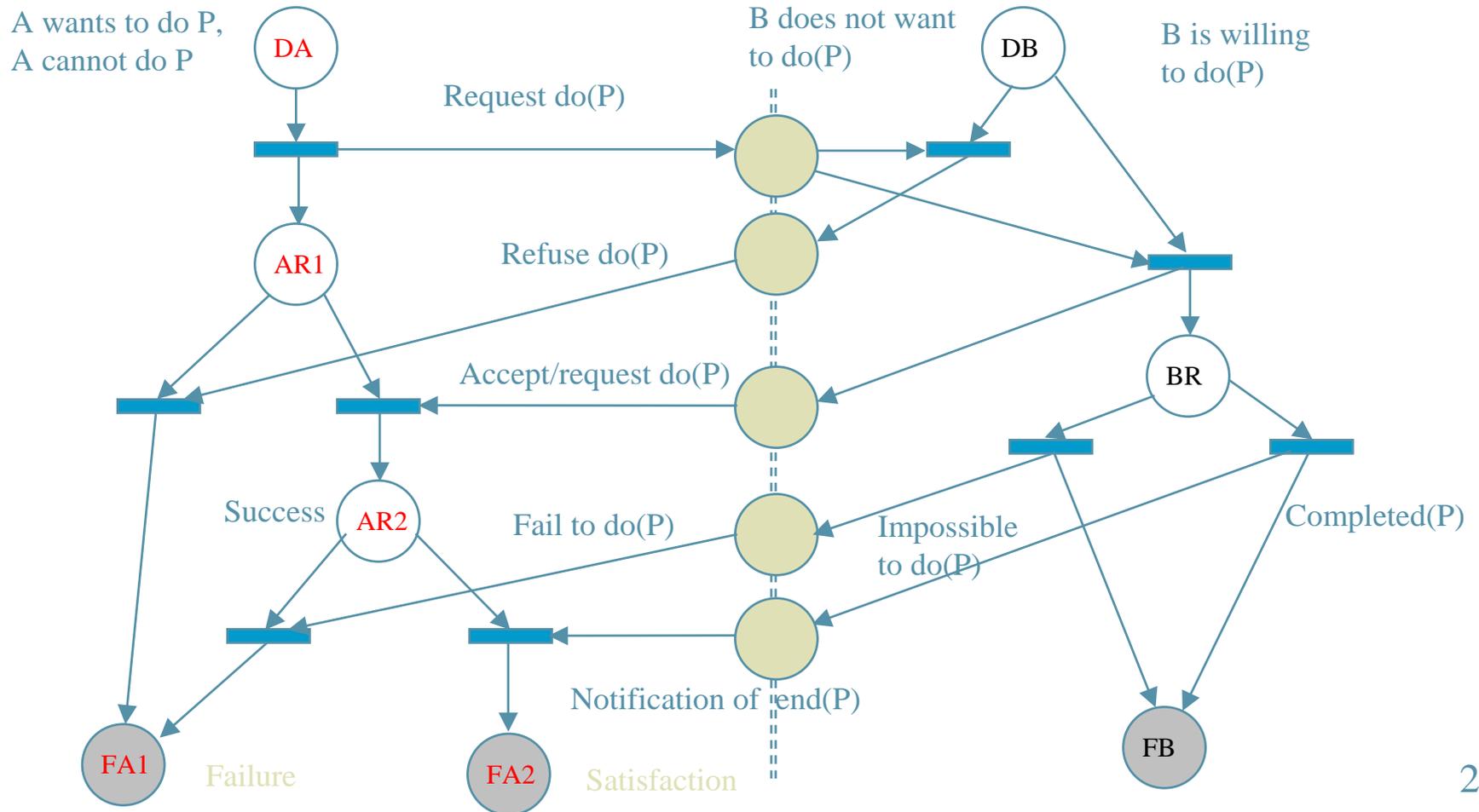
$[\text{tell}, S, R, \text{inR}, R_w, \text{IO}, \text{Content}] \mid$

$\text{problem}(\text{Conv}, R, S, \text{inR}, _, \text{IO})$

5.3 Réseaux de Petri

Ferber, 1997

- Un graphe orienté avec deux types de noeux: places et transitions
- Une transition est activée si toutes les places d'entrée P de T possèdent des tokens



Bibliographie

- M. Huhns, L. Stephens. Multiagent systems and societies of agents. In Multiagent Systems - A Modern Approach to Distributed Artificial Intelligence, G. Weiss (Ed.), The MIT Press, 2001, p.79-120.
- M. Wooldridge. Reasoning about Rational Agents. The MIT Press, 2000, Chapter 7
- Y. Labrou, T. Finin. Semantics and conversations for an agent communication language. In Readings in Agents, M. Huhns & M. Singh (Eds.), Morgan Kaufmann, 1998, p.235-242.
- J. Ferber - Multi-Agent Systems. Addison-Wesley, 1999, Chapter 6
- T. Finin, R. Fritzson - KQML as an agent communication language. In Proc. of the Third International Conference on Information and Knowledge Management (CIKM'94), ACM Press, 1994.
- M. Singh. Agent communication languages: Rethinking the principles. IEEE Computer, Dec. 1998, p.40-47.
- Y. Labrou, T. Finin, Y. Peng. Agent communication languages: The current Landscape. IEEE Computer, March/April 1999, p. 45-52.
- FIPA97. "Agent Communication Language" Specification FIPA, 11/28/97

Webliographie

- DARPA KSE** <http://www-ksl.stanford.edu/knowledge-sharing/>
KQML <http://www.cs.umbc.edu/kqml/>
KIF <http://logic.stanford.edu/kif/>
Ontolingua <http://www-ksl-svc.stanford.edu:5915/&service=frame-editor>
FIPA <http://www.fipa.org/>
DAML <http://www.daml.org/>

Gruber, T., What is an Ontology, <http://www.kr.org/top/definitions.html>
<http://www.w3.org/2001/sw/WebOnt/>
<http://www.cs.man.ac.uk/~horrocks/Slides/index.html>