Multi-Agent Oriented Programming
- Interaction Oriented Programming -

Olivier Boissier

ENS Mines Saint-Etienne
http://www.emse.fr/~boissier
Olivier.Boissier@emse.fr

Web Intelligence Master – Nov 2012

Plan

- Introduction
- Fondements
- Langage de Communication Agent
- Protocoles d'interaction entre agents
- Conclusion

Définition

Définition

- Interaction = mise en relation **dynamique** de deux ou de plusieurs agents par le biais d'un ensemble d'actions **réciproques**

  Existence d'une interaction lorsque la dynamique propre d'un agent est perturbée par les influences des autres

- Interaction est le moteur d'un système multi-agents.
- Selon les agents et les systèmes, l'interaction prend diverses formes :
  - actions sur l'environnement,
  - inférences,
  - communication,

Motivations

Caractéristiques Multi-Agents

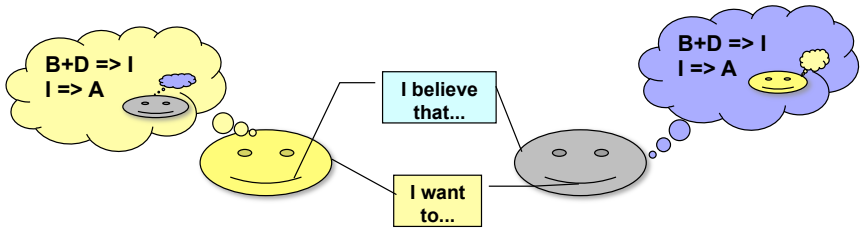
Besoins :

- Un comportement global cohérent au niveau du système (coopération, collaboration, …) répondant aux objectifs globaux à satisfaire
- Les agents doivent comprendre un langage commun

Grâce ou malgré

- Autonomie des agents : les agents agissent de manière autonome selon leurs buts, compétences, préférences, de manière non prévisible
- Hétérogénéité des agents et ouverture du SMA : impossible de prévoir a priori tous les comportements des agents
- Délégation/adoption de tâches entre les agents

Caractéristiques Agent



- Un agent est une entité autonome capable de **raisonner sur des connaissances**,
- qui doit partager :
  - Des connaissances afin de révéler une partie de son état mental aux autres agents
  - Des intentions afin de modifier l'état mental des autres agents

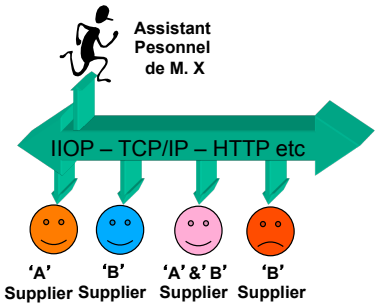
Plan

- ✓ Introduction
- Fondements
- Langage de Communication Agent
- Protocoles d'interaction entre agents
- Conclusion

Plusieurs couches
Couche système distribué (1)

Fournir immédiatement un lot de 10 pièces 'A' et lot de 20 pièces 'B', au meilleur prix.

Mister X

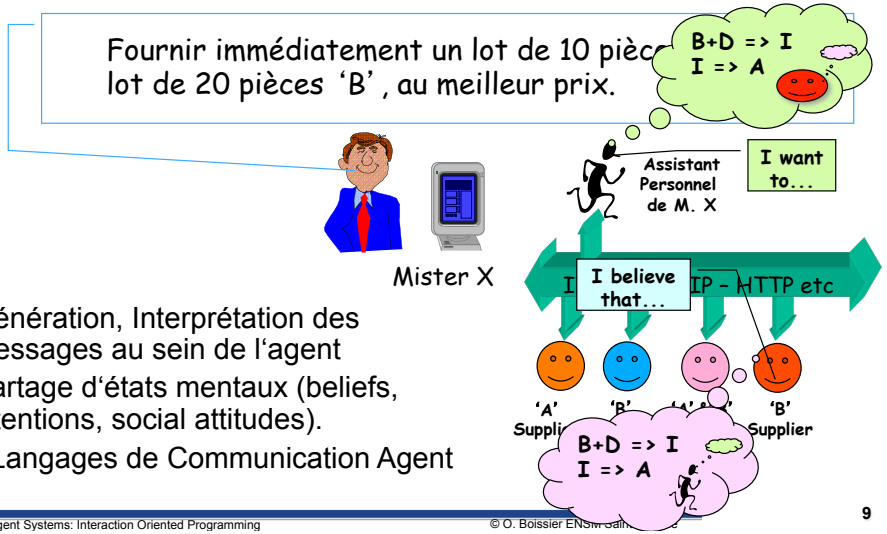


Couche logique au dessus des services fournis par les niveaux bas et intermédiaires des systèmes distribués (Protocole de transport pour le routage des messages SMTP, TCP/IP, HTTP, IIOP, etc)

Plusieurs couches
Couche système distribué (2)

- S'appuie sur des mécanismes assurant :
  - Identification de l'émetteur,
  - Respect de l'ordre des messages en émission et en réception,
  - Fiabilité de la couche de transport.
- Utilise des mécanismes pour :
  - des échanges directs par messages :
    - point à point / diffusion,
    - synchrone / asynchrone,
    - désignation des agents.
  - des échanges par mémoire partagée :
    - Associativité,
    - ex : architecture de tableaux noirs.

Plusieurs couches
Couche multi-agent (1)



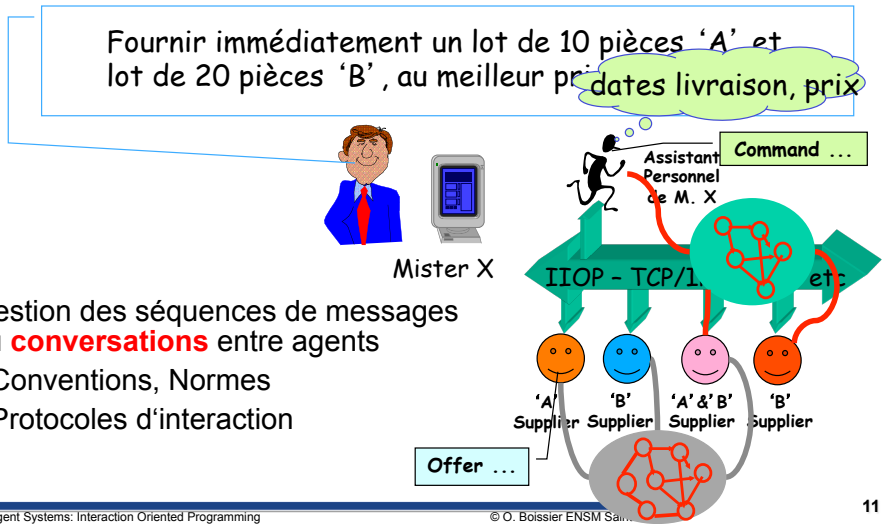
- Génération, Interprétation des messages au sein de l'agent
- Partage d'états mentaux (beliefs, intentions, social attitudes).

→ Langages de Communication Agent

Plusieurs couches
Couche multi-agent (2)

- Communication dans un langage commun (hypothèse courante),
  - Dans un système ouvert, un langage commun constitue une interface entre les agents
  - ➔ Les agents doivent avoir des capacités à manipuler un langage commun
- Dimensions de la communication :
  - Syntaxe : manière dont les symboles sont structurés
  - Sémantique : ce que le symbole désigne
  - Pragmatique : manière dont les symboles sont interprétés et dont la communication est mise en œuvre
  - → Compréhension du sens des termes (à quoi ils font référence, sémantique + pragmatique)
  - Utiliser efficacement ce vocabulaire, dans un temps et des ressources de calcul limitées, pour :
    - exécuter des tâches, satisfaire des buts, éviter des conflits, partager des connaissances, modifier l'environnement et/ou état mental d'un autre agent

Plusieurs couches
Couche multi-agent (3)



- Gestion des séquences de messages ou **conversations** entre agents

→ Conventions, Normes

→ Protocoles d'interaction

Sources multiples

- Linguistique, Philosophie du langage, Psychologie cognitive et sociale, sociologie
- Linguistique informatique, Intelligence Artificielle
- Pragmatique conversationnelle [Habermas]
- Intention dans les communications
  - Prise en compte des états mentaux (ex: BDI) :
    - croyances, intentions, désirs, ...

➢ Théorie des actes de langage (Speech Acts) [Austin 62, Searle 72, Vanderveken 88],

➢ Interactions au sein de conversations

Théories des actes de langage

- Théories des actes de langages sont des théories relatives à l'utilisation du langage (pragmatique)
- "Quand dire c'est faire" [Austin 62]
  - ➡ Toute communication est faite avec l'objectif de satisfaire un but ou une intention
  - ex : "il pleut!", "ferme la porte, s'il te plait." ...
- Importance du caractère communautaire, social et institutionnel du langage

Dimensions

- Acte locutoire :
  - production d'une suite de signes selon les règles syntaxiques d'un langage donné , contexte, références (➡ mode de production).
- Acte illocutoire :
  - acte accompli en produisant cette suite de signes dans un contexte donné, exprimant une intention (➡ intention du locuteur)
- Acte perlocutoire:
  - acte accompli par le fait d'énoncer la suite de signes (➡ effet sur l'allocutaire)
- exemple
  - "Tu ne dois pas faire cela" (locutoire), je conseille ou j'ordonne (illocutoire), dissuade ou importune (perlocutoire).

Exemple (Emetteur)

Marie dit à Jean : "S' il te plaît, ferme la fenêtre".

force illocutoire

contenu propositionnel

Marie considère ce message soit comme une **demande**, soit comme un **ordre** (composante illocutoire)

Marie émet des sons (composante locutoire)

Exemple (Récepteur)

Marie dit à Jean : "S' il te plaît, ferme la fenêtre".

force illocutoire

performatif (verbe) qui identifie la force illocutoire

Jean considère ce message soit comme une **demande**, soit comme un **ordre** (composante illocutoire)

Si tout va bien (pour l' intention illocutoire de Marie) Jean ferme la fenêtre (composante perlocutoire).

Force illocutoire

Acte de langage = Force illocutoire (F) “composante intentionnelle” + contenu propositionnel (p) “composante représentationnelle”

- 6 composantes [Vanderveken 88]:
  - but illocutoire,
  - mode d'accomplissement,
  - conditions :
    - ◆ contenu propositionnel ◆ préparatoires ◆ sincérité
  - degré de puissance.
- 5 classes d'actes illocutoires “performatifs” :
  - assertifs, directifs, commissifs, déclaratifs, expressifs.

Succès et satisfaction

- **Succès** et **satisfaction** d’un acte de discours :
  - sont exprimés par les conditions de succès et conditions de satisfaction
  - participent à la définition de la sémantique de l’acte.
- Succès :
  - conditions devant être remplies dans le contexte d'énonciation.
- Satisfaction :
  - conditions devant être satisfaites dans l'état du monde résultant de la composante perlocutoire de l'acte.

Phase 1

Sans communication

Ex: sémaphores

Signaux

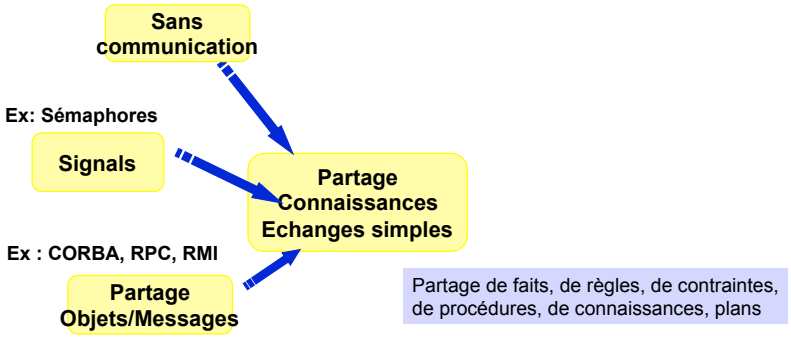
Ex : CORBA, RPC, RMI

Partage Objets/Messages

Partage d‘objets, Appels de procédures
Structures de données

Phase 2

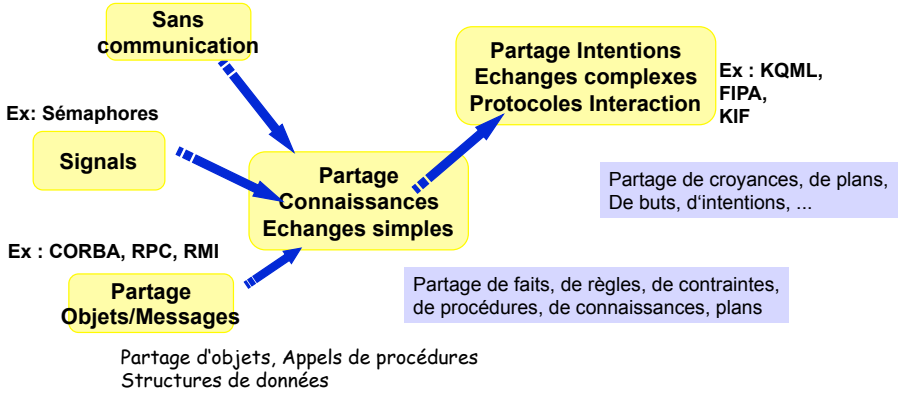
- Langages Ad hoc
- Standardisation : Knowledge Sharing Effort (KSE)



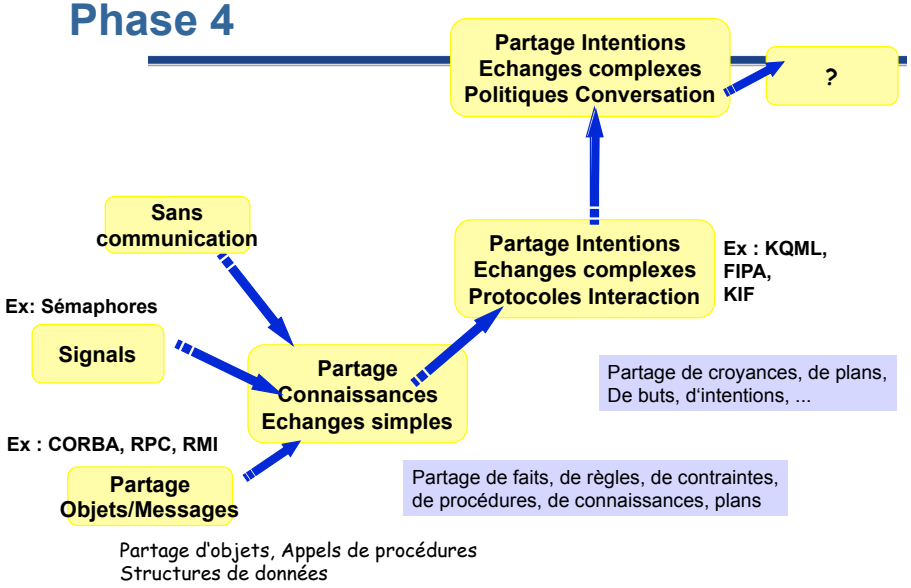
Partage d‘objets, Appels de procédures
Structures de données

Phase 3

- Langages Ad hoc
- Standardisation : KQML [Labrou 99] (in KS Effort) , FIPA ACL [FIPA 97,99,00]



Phase 4

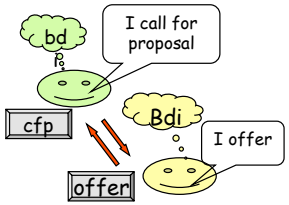


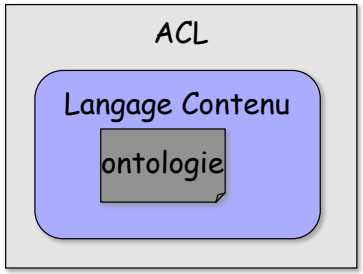
Plan

- ✓ Introduction
- ✓ Fondements
- Langages de communication Agent
- Protocoles d'interaction entre agents
- Conclusion

Agent Communication Language

- Langage de haut-niveau pour l'échange d'attitudes propositionnelles et établir collaboration, négociation, ...
  - Inform, request, cfp, agree, understood
  - FIPA-ACL, KQML, etc.
- Sémantique basée sur les états mentaux
- Intègre plusieurs sous langages
  - Sender, Receiver,
  - Langage de contenu
    - Action, objects, propositions
    - Ex: KIF, FIPA-SL, FIPA-CCL, etc.
- Ontologie : Vocabulaire commun, avec des définitions précises, relatif à un domaine
  - Wheather-ontology, cinema-ontology, etc.





KQML [Labrou 96]

- Originellement développé au sein du “Knowledge Sharing Effort” (ARPA, NSF, ...) pour le partage de **connaissances,** d’**informations,** de **services** entre différents systèmes à base de connaissances
  - KSE : organisé en trois groupes de travail :
    - Interlingua ➡ KIF
    - SRKB ➡ Ontolingua
    - External Interfaces ➡ KQML
- Du fait de sa généralité, utilisé ensuite comme ACL
- Historique :
  - 1993: Spécification de KQML ACL avec des exemples d‘architectures
  - 1997: proposition d‘une nouvelle spécification de KQML par Y. Labrou et T. Finin
- www.agents.edu/kqml

Syntaxe (1)

- Définition d’une syntaxe avec 3 niveaux :
  - communication, message, contenu

```
(ask-if                    performatif
    :ontology industrial   infos. utiles pour
    :language prolog       l'interprétation
   :sender A               et le routage du
   :receiver B             message
   :reply-with id1
   :content "start(process,i)" )
```


Syntaxe (2)

Reserved Parameter Keywords	
:sender	The actual sender of the performative
:receiver	The actual receiver of the performative
:from	The origin of the performative in *:content* when *forward* is used
:to	The final destination of the performative in *:content* when *forward* is used
:in-reply-to	The expected label in a response to a previous message
:reply-with	The expected label in a response to the current message
:language	The name of the representation language of the *:content*
:ontology	The name of the *:ontology* assumed in the *:content* parameter
:content	The information about which the performative expresses an attitude

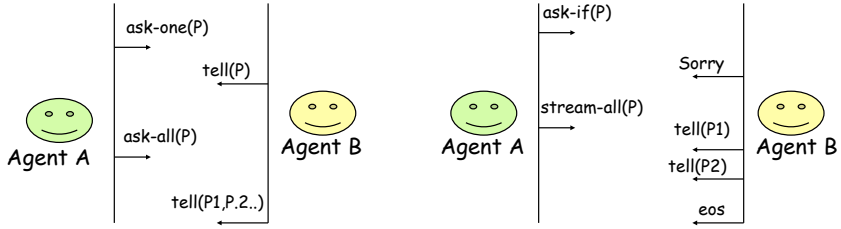
Performatifs (1)

- Bibliothèque de 36 **performatifs** répartis en 3 catégories :
  - discours : échanges d’informations et de connaissances (tell, deny, ask-if, stream-all, …)
  - interconnexion : entre l’agent et les *facilitators* (register, unregister, broadcast, broker, …)
  - exception : changer le déroulement des échanges (error, sorry, standby, …)
- Sémantique a été spécifiée par la suite (93->98).
  - ambiguïté et imprécision des performatifs,
  - certains sont inutiles et incohérents,
  - absence de certains performatifs (promissif).

Performatifs (2)

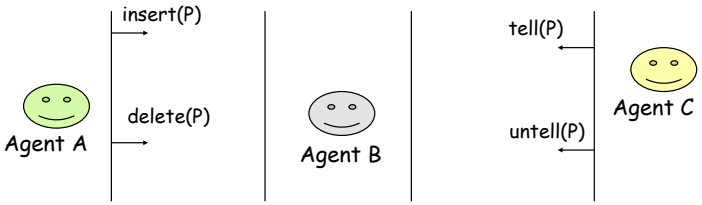
- **Inform family**:
  - Tell, untell,
  - Database: insert, uninsert, delete-one, delete-all, undelete
- **Request family**:
  - Network: broadcast, forward
  - Goal: achieve, unachieve
  - Facilitation: broker-one, recommend-one, recruit-one, broker-all, recommended-all, recruit-all
- **Promise**: advertise, unadvertise
- **Meta**: deny, subscribe
- **Reply**: stream,eos
- **Query family**:
  - Basic: ask-if, ask-one, ask-all
  - Stream: stream, eos
  - Cursor: standby, ready, next, rest, discard

Performatifs (3) : Query



- Les performatifs ask-one, ask-all, ask-if et stream-all fournissent un mécanisme de base de demande
- Ex: quand agent A envoie ask-all(P) à agent B cela signifie que A veut connaître de B toutes les instantiations qui rendent P vraie

Performatifs (4) Tell, Untell, Insert, Delete



- tell(P) signifie que l'agent C croît que P est vraie. Untell(P) signifie que l'émetteur, agent C, ne croît pas que P
- insert(P) est une demande de l'agent A vers l'agent B d'ajouter P et delete(P) de l'enlever

Sémantique (1)

- ***Preconditions*** (Pre) indiquent les états nécessaires dans les agents pour que le message puisse être créé
- ***Postconditions*** (Post) décrivent les états des agents après le traitement du message
- ***Completion*** décrit le résultat attendu du message

Sémantique (2) : Exemple Tell

Tell(A,B,X)

- Pre(A): bel(A,X) ∧ know(A,want(B,know(B,bel(A,X))∨know(B,¬bel(A,X))))
- Pre(B): intend(B, know(B,bel(A,X))∨know(B,¬bel(A,X)))
- Post(A): know(A,know(B,bel(A,X)))
- Post(B): know(B,bel(A,X))
- Completion: know(B,bel(A,X))

Sémantique (3) : Problèmes

1. Deny(A,B,Tell(B,A,X))
   - Est-ce que A nie le fait que B ait dit ou
   - Est-ce que A nie le fait X ?
2. Uninsert(A,B,X) (possible seulement après insert(A,B,X))
   - Post(B): ¬bel(B,X)
   - Que se passe-t' il si B peut inférer X à partir d' autre sources?

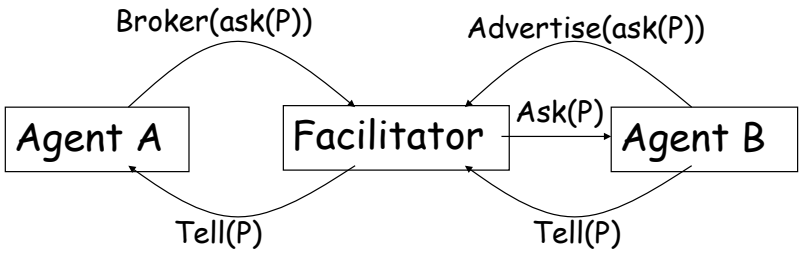
Autres caractéristiques

- Nommage des Agents
  - Gestion du nommage au sein d'une infrastructure Multi-Agent
- Infrastructure de mise en relation (brokering) : Facilitator
  - agents particuliers qui échangent des meta-informations sur les autres agents et fournissent des services de communication tels que :
    - Message forwarding
    - Matchmaking
    - Resource discovery
  - Ils peuvent être seulement des pages jaunes ou des agents intelligents
- ✓ Plusieurs performatifs ' 'administratifs' ' liés à la gestion du système

Nommage des agents

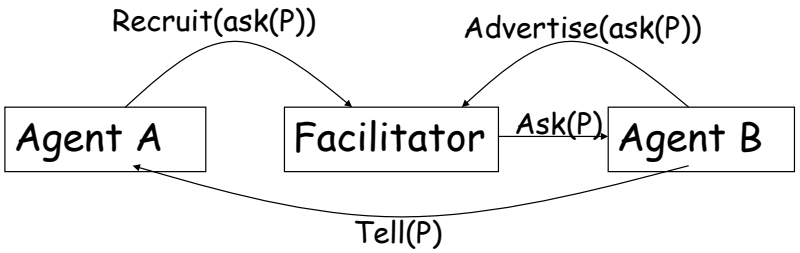
- Performatif : register.
- Hypothèse que les noms sont locaux
  - Agent A peut s'enregistrer auprès de l'agent B sous le nom de Coltrane
  - Agent A peut s'enregistrer auprès de l'agent C sous le nom de Rollins
- N'exclue par l'utilisation d'un *Agent Name Server* central
- Informations nécessaires pour l'enregistrement d'un agent dans l'ANS :
  - name(davis,tcpip,[davis.jazz.com, 8080])
  - name(adrian,smtp,[davis@jazz.com])
  - name(adrian, http,[www.jazz.com:8090/people/davis])

Facilitator : Performatifs



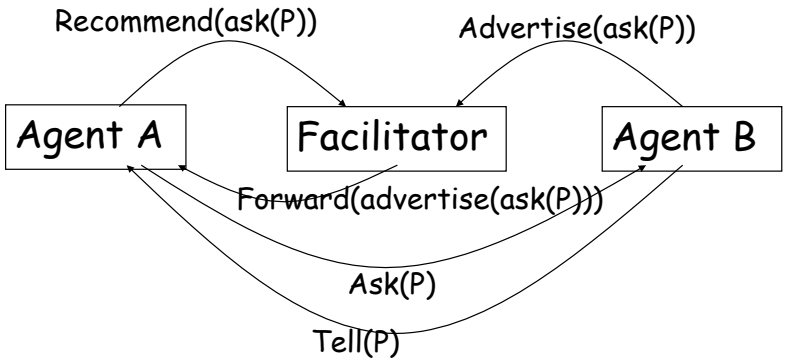
Broker

Facilitator : Performatifs (2)



Recruit

Facilitator : Performatifs (3)



Recommend

ACL-FIPA

- KQML, premier essai de standardisation d ' un ACL **MAIS**
  - pas de sémantique formelle, pas d 'infrastructure pour la gestion des agents, …

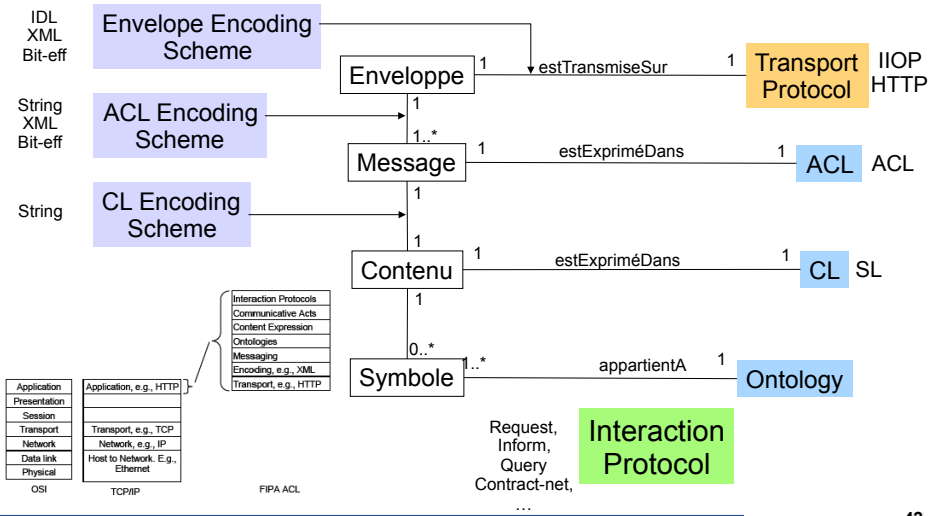
  ➔ **MULTIPLES** dialectes **incompatibles**, évoluant **rapidement**
- Standardisation d' un ACL au sein de la FIPA
  - FIPA (Foundation for Intelligent Physical Agents)
    - Création en Avril 1996 (50 membres)
    - Identification et sélection d 'applications
      - Personal Travel Assistance, Personal Assistant, Audio Visual Entertainment & Broadcasting, Telecommunication Management
    - Sélection d 'axes technologiques
      - Gestion des agents, Communication entre agents, Intégration Agent/SW
    - de nombreuses spécifications

  ➔ **UN SEUL standard** ACL-FIPA évoluant **lentement**

ACL-FIPA [FIPA 97, 99]

- Syntaxe similaire à celle de KQML
- Ensemble d'actes de communication de base,
  - les autres actes peuvent être définis par composition d'actes de base
- Ensemble de messages prédéfinis que tout agent doit être capable de traiter :
  - not-understood : si l'agent reçoit un message qu'il ne peut pas comprendre. Tout agent doit être capable de traiter un tel message.
- Primitives d'administration et de facilitation hors de l'ACL
  - Elles sont définies dans le contenu !
- Sémantique basée sur ARCOL [sadek]
  - Exprimée au sein de Feasability Preconditions (FP) et Rational Effect (RE) faisant référence à des états dont le contenu est exprimé via une logique multi-modale (Semantic Language)
  - ✓ Modèle trop lourd, difficilement utilisable
    - ➔ Stratégies de simplifications (Conversation Policies)

Vision d'ensemble



Transport des messages

- Message Transport Protocol (MTP) :
  - réalise le transfert physique des messages entre deux MTS
- Message Transport Service (MTS) :
  - appelé également ACC (Agent Communication Channel)
  - fourni par la plateforme sur laquelle l'agent s'exécute.
  - assure le transport de messages en FIPA ACL entre des agents d'une même plateforme ou d'agents entre différentes plateformes
- ACL :
  - Charge utile des messages transportés par le MTS et MTP

Enveloppe d'un message

	Parameter	Description
*	to	If no intended-receiver parameter is present, then the information in this parameter is used to generate intended-receiver field for the messages the ACC subsequently forwards.
*	from	If required, the ACC returns error and confirmation messages to the agent specified in this parameter.
	comments	None.
*	acl-representation	None. This information is intended for the final recipient of the message.
	payload-length	The ACC may use this information to improve parsing efficiency.
	payload-encoding	None. This information is intended for the final recipient of the message.
*	date	None. This information is intended for the final recipient of the message.
	intended-receiver	An ACC uses this parameter to determine where this instance of a message should be sent. If this parameter is not provided, then the first ACC to receive the message should generate an intended-receiver parameter using the to parameter.
	received	A new received parameter is added to the envelope by each ACC that the message passes through. Each ACC handling a message must add a completed received parameter. If an ACC receives a message it has already stamped, it is free to discard the message without any need to generate an error message.
	transport-behaviour	Reserved for future use.

* Champ obligatoire

Structure d'un message ACL

	Parameter	Category of Parameters
*	performative	Type of communicative acts
	sender	Participant in communication
*	receiver	Participant in communication
	reply-to	Participant in communication
	content	Content of message
	language	Description of Content
	encoding	Description of Content
	ontology	Description of Content
	protocol	Control of conversation
	conversation-id	Control of conversation
	reply-with	Control of conversation
	in-reply-to	Control of conversation
	reply-by	Control of conversation

* Champ obligatoire

Structure d'un message ACL

- **Language** : langage dans lequel le champ contenu est exprimé.
  - **Notes:** Le champ *content* de l'ACL est exprimé dans un langage formel (content language). Ce champ peut être omis si les agents recevant le message connaissent le langage dans lequel l'expression du contenu est écrite.
  - **Notes:** Le contenu peut être encodé de différentes manières. L'élément d'encodage est utilisé de manière optionnelle pour spécifer l'encodage pour l'agent récepteur. Si l'encodage n'est pas spécifié, il sera précisé dans l'enveloppe du message.
- **ontology** : ce champ désigne le(s) ontologie(s) utilisée(s) pour donner un sens aux symboles utilisés dans l'expression du contenu.
  - **Notes:** Les ontologie(s) est/sont utilisées en conjonction avec le champ language pour aider l'interprétation du message au sein de l'agent récepteur.

Exemple

(***inform***	type d'acte de communication
:sender A	infos. utiles pour le routage
:receiver B	infos. utiles pour le routage
:content (price (bid goood02) 150)	proposition ou action
:in-reply-to round-4	
:reply-with bid04	
:encoding 1000	
:language fipa-sl1	langage utilisé dans content
:ontology hpl-auction	
:reply-by 10	deadline pour la réponse
:protocol offer	protocole d'interaction utilisé
:conversation-id conv02	conversation en cours
)	

Performatifs FIPA

- accept-proposal: the action of accepting a previously submitted proposal to perform an action
- agree: the action of agreeing to perform some action, possibly in the future
- cancel: the action of cancelling some previously requested action which has temporal extent
- cfp: the action of calling for proposals to perform a given action
- confirm: the sender informs the receiver that a given proposition is true, where the receiver is known to be uncertain about the proposition
- disconfirm: the sender informs the receiver that a given poposition is false, where the receiver is known to believe, or believe it likely that, the proposition is true.
- failure: the action of telling another agent that an action was attempted but the attempt failed.
- inform: the sender informs the receiver that a given proposition is true.
- inform-if: a macro-action for the agent of the action to inform the recipient whether or not a proposition is true.
- inform-ref: a macro-action for the sender to inform the receiver the object which corresponds to a descriptor (e.g., a name)
- not-understood: the sender A informs the receiver B that it perceived that B performed some action, but A did not understand what B did.

Performatifs FIPA

- propagate: the sender intends that the receiver treats the embedded message as sent directly to it, and wants the receiver to identify the agents denoted by the given descriptor and send the received propagate message to them
- propose: the action of submitting a proposal to perform a certain action, given certain preconditions
- proxy: the sender wants the receiver to select target agents denoted by a given description and to send an embedded message to them
- query-if: the action of asking another agent whether or not a given proposition is true.
- query-ref:the action of asking another agent for the object referred to by a referential expression
- refuse: the action of refusing to perform a given action and explaining the reason for the refusal.
- reject-proposal: the action of rejecting a proposal to perform some action during a negotiation.
- request: the sender requests the receiver to perform some action.
- request-when: the sender wants the receiver to perform some action when some given proposition becomes true.
- request-whenever: the sender wants the receiver to perform some action as soon as some proposition is true and thereafter each time the proposition becomes true again.
- subscribe: the act of requesting a persistent intention to notify the sender of the value of a reference, and to notify again whenever the object identified by the reference changes.

Performatifs (1)

- Information (contenu : proposition)
  - query_if, query_ref, subscribe, inform, inform_if, inform_ref, confirm, disconfirm, not_understood
- Distribution de tâches (contenu : action)
  - request, request_whenever, cancel, agree, refuse, failure
- Négociation (contenu : action et proposition)
  - cfp, propose, accept_proposal, reject_proposal

Sémantique (1)

- Sémantique basée sur les attitudes mentales (belief, intention, etc.)
- La sémantique d'un acte est définie en termes de *Feasibility Preconditions* (FPs) et *Rational Effects* (REs)
- Les Feasibility Preconditions d'un AC définissent les conditions qui doivent être vraies avant que l'agent planifie l'exécution de l'AC
- Le Rational Effect est l'effet qu'un agent espère produire en exécutant l'AC (Il n'y a aucune garantie que l'effet sera atteint)
- L'utilisation de FPs et REs implique la description des états des agents

Sémantique (2)

- Belief : $B_i p$ « l'agent *i* croit que la proposition *p* est vraie »
- Uncertain : $U_i p$ « l'agent *i* est incertain à propos de *p* mais pense que *p* est plus vraisemblable que non *p*
- Choice : $C_i p$ « l'agent *i* souhaite que la proposition *p* soit vérifiée
- $a_1;a_2$ (séquence), $a_1|a_2$ (choix)
- Feasable(a), Done(a), Agent(i,a)
- But persistant $PG_i p$, Intention $I_i p$

Sémantique (3) : exemple INFORM

INFORM : *i* informe *k* que la proposition *p* est vraie

- contenu du message : proposition
- <i, inform(k,p)>

  FP : $B_i p \wedge \neg B_i(Bif_k p \vee Uif_k p)$

  RE : $B_k p$

Sémantique (4) : exemple REQUEST

<i, request(j,a)>

FP: $FP(a)[i\backslash j] \wedge B_i Agent(j,a) \wedge \neg B_i I_j Done(a)$

RE: Done(a)

Agent i requests agent j to open a file

```
(request_   :sender i
            :receiver j
            :content "open \"db.txt\" for input"
            :language vb)
```


Sémantique (5) : exemple QUERY-IF

<i,query-if(j,X)≡<i, request(j,<j,inform-if(i,X)>)>

FP: $\neg B_j X \wedge \neg B_j \neg X \wedge \neg U_j X \wedge \neg U_j \neg X \wedge$
$\neg B_i I_j Done(<j,inform\text{-}if(i,X)>)$

RE: Done (<j,inform(i,X)>|<j,inform(i, ¬X)>)

Agent i asks agent j if j is registered with server d1

```
(query-if_  :sender i
            :receiver j
            :content registered(j,d1)
            :language Prolog)
```


Représentation des connaissances

- Besoin d'être capable de représenter et de raisonner sur:
  - Modèles des autres agents, modèles des utilisateurs, beliefs, desires, intentions, plans, perceptions, etc.
  - Tâches, architectures de tâches, plans, etc.
  - Meta-data sur des documents, sur des collections de documents
- De plus, les agents devront pouvoir aussi communiquer sur leur représentation des connaissances elle-même

Multiples langages de contenu

- Proposition d'une grande diversité de langages de contenu pour les ACL
  - KIF, Prolog, CLIPS, SQL
  - FIPA-SL :
    - Le plus utilisé dans les spécifications de la FIPA (obligatoire pour la gestion des agents par exemples)
    - Trois sous-ensembles : SL0 (sous-ensemble minimal de SL), SL1 (forme propositionnelle), SL2 (restriction pour la décidabilité)
  - FIPA-CCL : langage de choix de contrainte
  - FIPA-RDF0 : langage basé sur RDF
  - FIPA-KIF
- Intérêt particulier pour les langages de contenu qui peuvent servir de langage pivot suffisamment expressif, pour de multiples systèmes

FIPA-SL

- **Proposition** :
  - expression à qui peut être affectée une valeur de vérité dans un contexte donné.
  - formule suivant un ensemble de règles de formation
  - utilisée comme contenu d'un performatif de type `inform`
- **Action** qui peut être exécutée :
  - Action simple ou action composite construite par mise en séquence ou alternative d'actions
  - Utilisée comme contenu d'un performatif de type `request`.
- **Expression de référence** (IRE) :
  - identification d'un objet dans le domaine.
  - Opérateur de référence
  - Utilisée comme contenu d'un performatif de type `inform-ref`.

FIPA-SL : exemple information

- Agent *i* confirms to agent *j* that it is, in fact, true that the shark is a mammal.

```
(confirm
    :sender (agent-identifier :name i)
    :receiver (set (agent-identifier :name j))
    :content ((is mammal shark))
    :language fipa-sl
)
```


FIPA-SL : exemple d'action

```
(request
    :sender (agent-identifier :name i)
    :receiver (set (agent-identifier :name j))
    :content ((action (agent-identifier :name j)
                  (deliver box017 (loc 12 19)))
              )
    :protocol fipa-request
    :language fipa-sl
    :reply-with order567
)
(agree
    :sender (agent-identifier :name j)
    :receiver (set (agent-identifier :name i))
    :content ((action (agent-identifier :name j)
 (deliver box017 (loc 12 19)))
              (priority order567 low)
             )
    :in-reply-to order567
    :protocol fipa-request
    :language fipa-sl
)
```

Agent *i* (a job-shop scheduler) requests *j* (a robot) to deliver a box to a certain location. *J* answers that it agrees to the request but it has low priority.

FIPA-SL : exemple d'IRE

```
(request
  :sender (agent-identifier :name i) :receiver (set(agent-identifier :name j))
  :content
    ((action (agent-identifier :name j)
      (inform-ref
       :sender (agent-identifier :name j)
       :receiver (set (agent-identifier :name i))
       :content ((iota ?x (UKPrimeMinister ?x)))
       :ontology world-politics
       :language fipa-sl
      )
     ))
  :reply-with query0  :language fipa-sl
)
(inform
    :sender (agent-identifier :name j)
    :receiver (set (agent-identifier :name i))
    :content ((= (iota ?x (UKPrimeMinister ?x)) "Tony Blair"))
    :ontology world-politics  :in-reply-to query0
)
```

Agent *i* requests *j* to tell it the current Prime Minister of the United Kingdom

KIF

- KIF = Knowledge Interchange Format
- Langage logique, version préfixée du calcul des prédicats du premier ordre avec des extensions pour la non-monotonie.
  - codage de tuples
  - opérateurs de comparaison
  - opérateurs logiques
  - connaissance sur des connaissances ‘ et ,
    (interested joe ‘ (salaire ,?x ,?y ,?z))
  - codage de procédures

Ontologies

- Ontologie : spécification des objets, des concepts et des relations dans un domaine particulier
- analogue à un schéma de base de données et non pas au contenu de la base elle-même
  - représenter les classes, les relations entre classes mais pas les instances.
- Cf. Web Sémantique

Programmation des ACL (1)

- Agent Centered Programming
  - ⇔ Linkage of ACL semantic and Agency Theory model of Agent's behavior
  - Use of a simplified version of Natural Language Speech Act Theory
    - Agent communication as a kind of communicative action
      - ⇔ Executed in service of intention
      - ♦ Change the belief of the parties involved in communication
- NO System Centered Programming

Programmation des ACL (2)

- Problems with relating ACL semantic with agency theory :
  - Agents are almost never actually programmed using the concepts that are used in an agency theory.
  - Agents being autonomous, an agent can **never** directly change the beliefs of another agent. Thus the effect of a speech act cannot be guaranteed.
  - Difficulty to express sequencing of acts, to express obligations on the receiving agent,

Programmation des ACL (3)

- Lots of implementations
- That rely on several *implicit* assumptions

  on :
  - Shared ontology, nonstandard meanings, …
  - Agents that use them,
  - Environments in which they are executed, …

  In order to improve the efficiency, for developer convenience, etc.

  ➔ Difficulty to generalize and extend

Plan

- ✓ Introduction
- ✓ Fondements
- ✓ Langage de Communication Agent
- Protocoles d'interaction entre agents
- Conclusion

Introduction

- Combinatoire des messages selon les ACL en se basant uniquement sur la sémantique des performatifs est trop importante
- Existence de schémas typiques structurant les échanges : attente de séquences de messages en réponse à d'autres messages.

→ **Protocoles d'interaction, Politiques de conversation**

- Autonomie des agents : utilisation propre de ces schémas globaux

→ **Stratégies d'interaction**

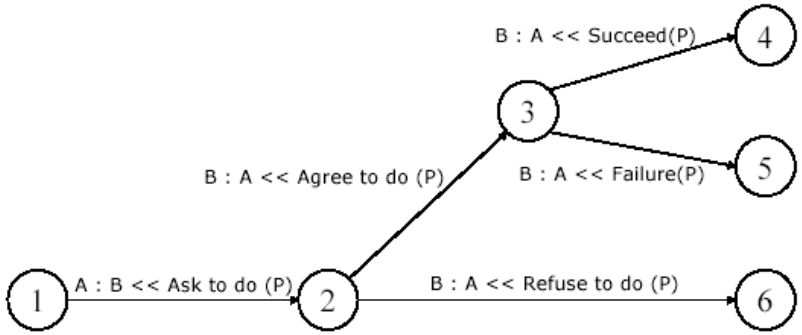
- Les agents peuvent engager de nombreux dialogues ou conversation selon un ou plusieurs de ces schémas, avec un ou plusieurs agents simultanément ou en séquence.

→ **Coordination des conversations**

Définition

- **Définition**
  - schéma commun de conversation utilisé pour exécuter une tâche.
  - Stratégie de haut niveau gouvernant les interactions entre agents permettant de faciliter et structurer leur dialogue.
  - Un protocole précise qui peut dire quoi à qui et les réactions possibles à ce qui est dit.
- Basé sur des actes de communication
  - Actes différents selon les étapes de la conversation, restriction des actes possibles
  - Actes agissent sur l'état mental de l'agent, sur la suite de la conversation
- Les sémantiques peuvent être définies au niveau du protocole d 'interaction au lieu de considérer la sémantique des actes individuels de communication.

Exemple



Ingénierie des Protocoles

- Modélisation avec différents formalismes :
  - graphes état-transition, graphe de raisonnement, réseaux de pétri, langages formels (Z), langages de descriptions de protocoles, ... AUML, …
- Méthodologie de définition de protocoles
- Existence d 'un ensemble de base de protocoles d 'interaction pré-définis, cf. Standards FIPA mais possibilité de définir des protocoles d 'interaction Ad-Hoc

Exemples de protocoles non-agent:
- TCP/IP, SNMP, Hand-shaking, Hello!

Agent vs Système Multi-Agent

- Distinction claire entre la description des conversations et la description des agents
  - Deux niveaux : conversations et agents
  - Spécification explicites de schémas de conversation

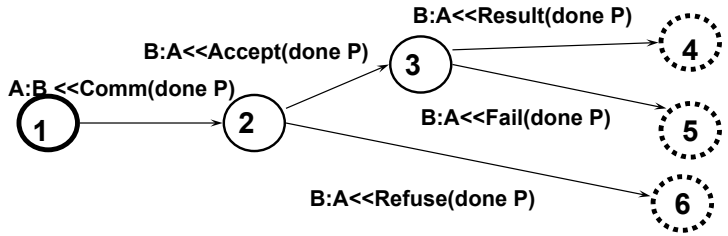
➔ Agents sont des entités autonomes qui interagissent à l' intérieur de schémas de conversations
  - Les schémas de conversation contraignent le comportement d'interaction des agents
  - Les conversations sont le résultat des interactions des agents

Langages de spécification

- Existence de plusieurs langages de spécification de protocoles d'interaction
  - Transition Nets (FSM, Petri Nets)
  - AUML
  - Spécification basées sur la Logique
  - Arbres de sous buts
  - Systèmes de spécification de protocoles réseaux
- Variation en rigidité, concurrence, complexité,
- Disponibilité d'outils et techniques.

Automates à états finis

- Description d'une conversation par une suite d'**états** liés par des **transitions** (interactions entre les agents).
- Point de vue global aux agents ou point de vue local à un agent



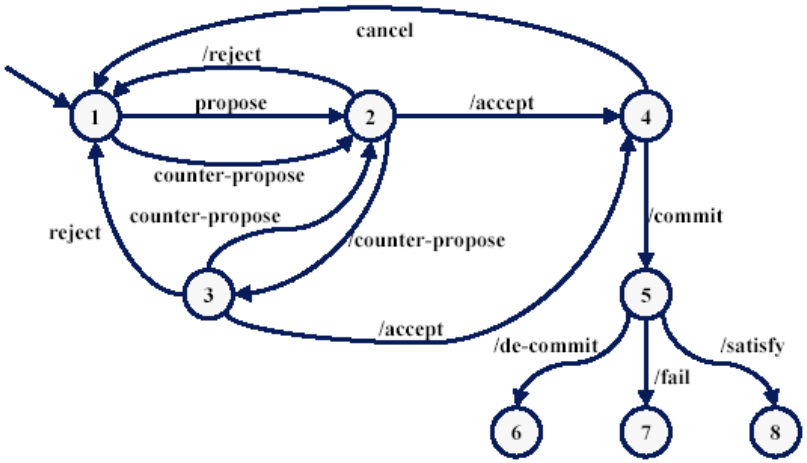
Cool [Barbuceanu 95]

- Fondé sur KQML
- Distinction de trois niveaux dans les interactions entre agents :
  - Contenu de l'interaction (abordé par les ontologies KIF)
  - Intention de l'interaction (abordé par KQML)
  - Conventions d'interaction (abordé par un langage de coordination COOL)

Cool : Notation

- **Automate à état fini**
  - **états** : états possibles d'une conversation (état accord, état refusé),
  - **transitions** : messages (performatifs) possibles,
    - <acte reçu> /
    - / <acte envoyé>
- **Un message reçu sans correspondance avec une transition valide de l'automate décrit déclenche une règle de reprise d'erreur**
- **Description selon le point de vue d'UN agent : besoin de deux automates pour une conversation.**

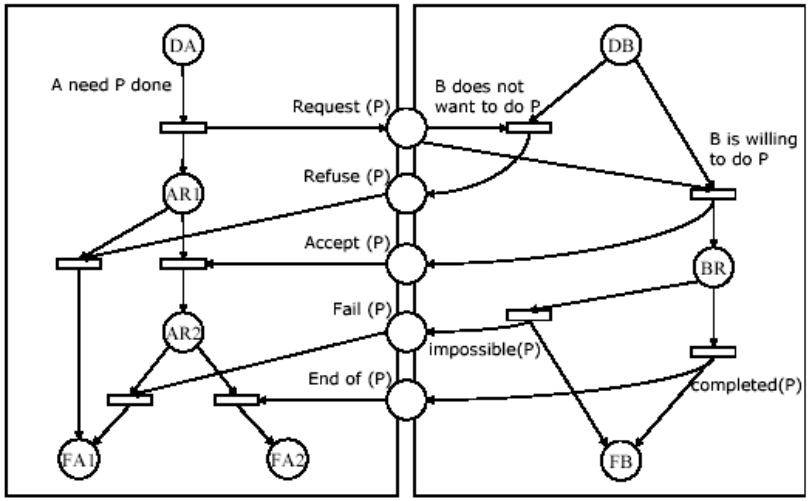
Cool : exemple



Réseaux de Petri

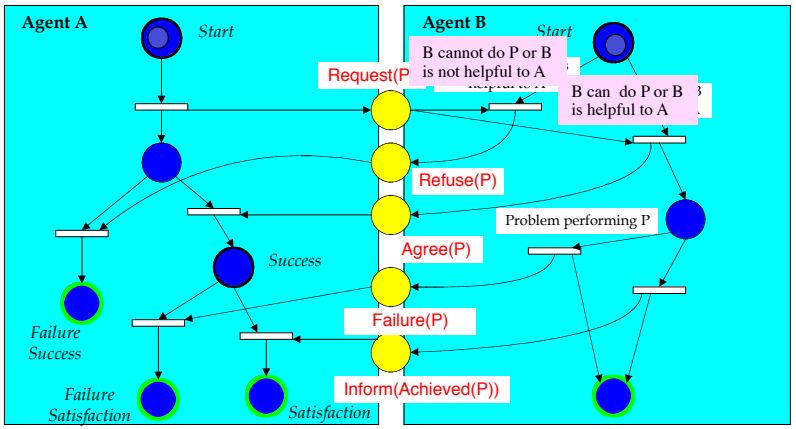
- Places
  - état interne de l'agent,
  - message en cours d'acheminement,
- Transitions
  - synchronisation due à la réception d'un message
  - conditions d'application des actions.

Réseaux de Petri



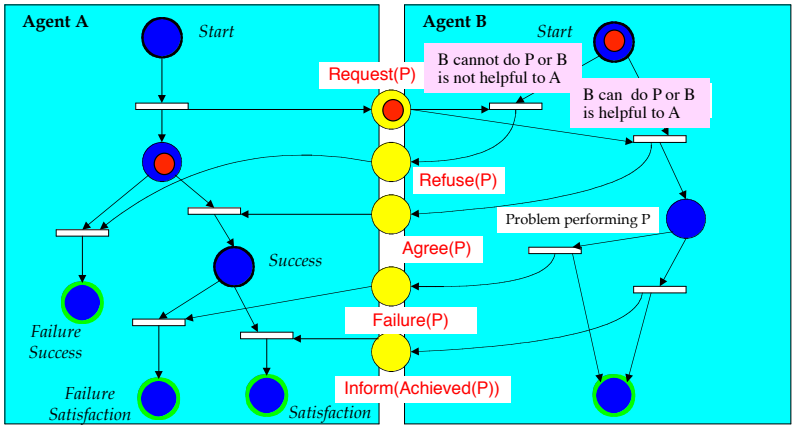
Réseaux de Petri

- Example: requesting to do something



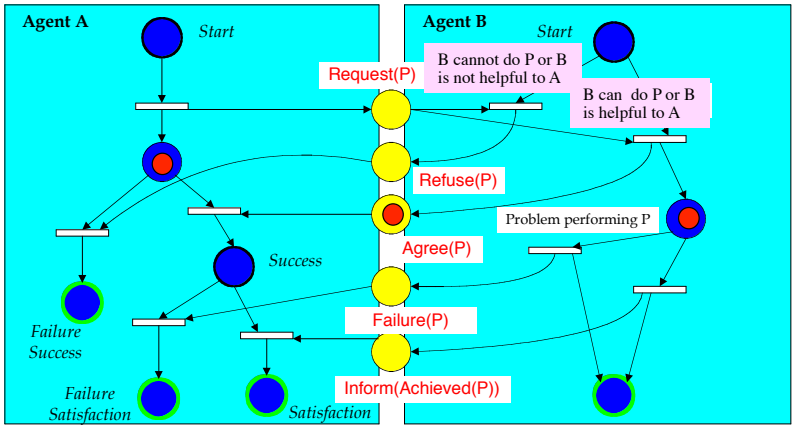
Réseaux de Petri

Example: requesting to do something



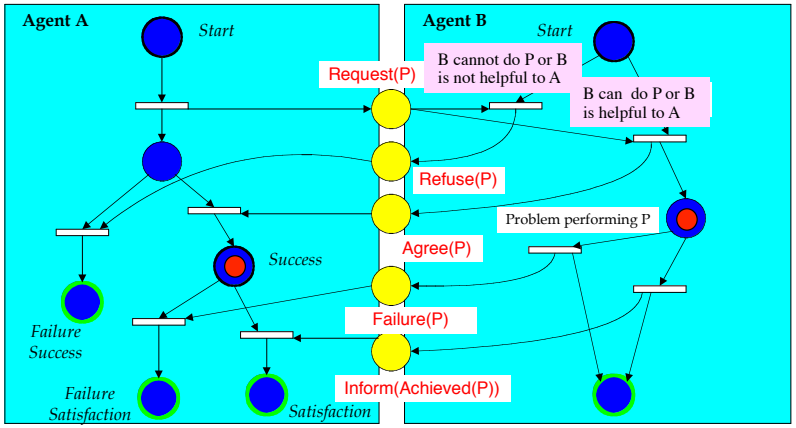
Réseaux de Petri

Example: requesting to do something



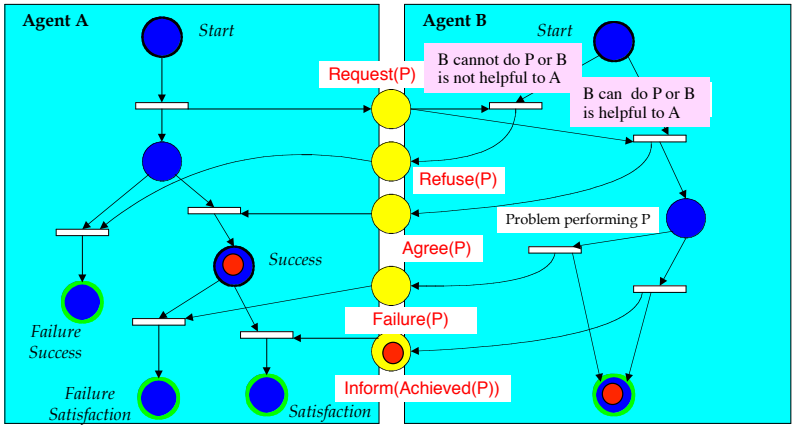
Réseaux de Petri

Example: requesting to do something



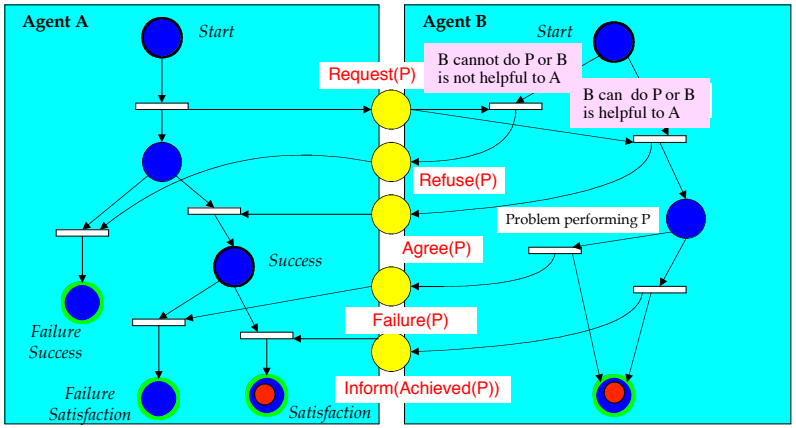
Réseaux de Petri

Example: requesting to do something



Réseaux de Petri

Example: requesting to do something

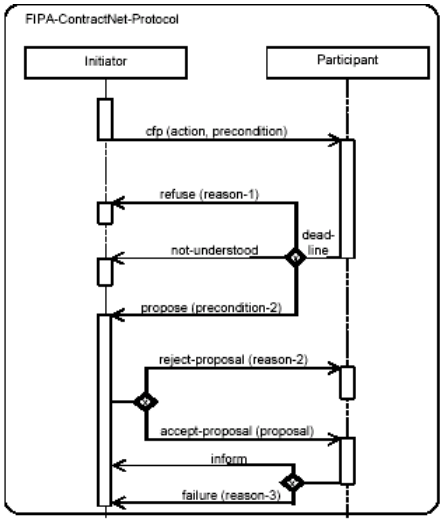


AUML (Agent UML)

- Semantics
  - A protocol description represents an interaction . a set of messages exchanged among different agent roles within a collaboration to effect a desired behavior of other roles or agent instances.
- Notation
  - Vertical dimension represents time,
    - Time proceeds down the page
    - Only time sequences are important
  - Horizontal dimension represents different agent roles
    - No significance to the horizontal ordering of the roles
- Various labels can be shown either in the margin or near the lifelines or messages that they label
  - E.g., timing marks, generated goals depending on the received message, etc.

Issu du cours de Scot De Loach

AUML : Notation

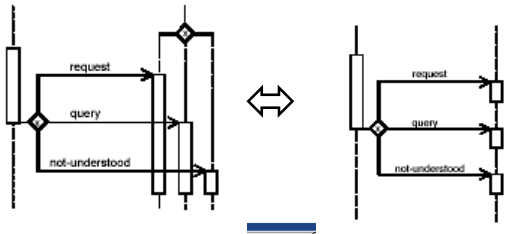


AUML : Vie d'un rôle

- The lifeline may split into two or more lifelines to show AND/OR parallelism and decisions
- Each separate track corresponds to a branch in the protocol
- The lifelines may merge together at some subsequent point
  - AND parallelism starts at a horizontal heavy bar,
  - OR parallelism (inclusive-or) starts at a horizontal heavy bar w a non-filled diamond, and,
  - Decision (exclusive-or) starts at a horizontal heavy bar with a non-filled diamond with "x" inside the diamond
- And is continued with a set of parallel vertical lifelines connected to the heavy bar.

AUML : Threads d'interaction

- A tall thin rectangle whose top is aligned with its initiation time and whose bottom is aligned with its completion time.
- Drawn over the lifeline of an agent role
- Represents a task being performed
  - May be labeled as text next to the thread or in the left margin
  - The incoming message may indicate the task



AUML : Messages (1)

- Message - communication from one agent role to another that conveys information with the expectation that the receiving agent role would react according to the semantics of the communicative act
- Shown as a horizontal solid arrow from one thread of interaction to another thread of interaction
  - May be specified using AND/OR/XOR
- Each arrow is labeled using the following syntax:

```
predecessor guard-condition sequence-expression
  communicative-act argument-list
```


AUML : Messages (2)

- predecessor - consists of at most one natural number followed by a slash (/) defining the sequencing of a parallel construct
- guard-condition - a usual UML guard condition - the message is sent iff the guard is true
- sequence-expression - a constraint, especially with n..m which denotes that the message is sent n up to m times
  - the keyword broadcast denotes the broadcast sending of a message
- communicative-act - the name of a communicative act, e.g., inform
- argument-list - a comma-separated list of arguments enclosed in parentheses - the parentheses can be omitted if the list is empty

Protocoles d'interaction

- Notation : AUML
- Sémantique
  - Une description de protocole représente un schéma d'interaction, ensemble de messages échangés entre différents rôles d'agent à l'intérieur d'une collaboration pour mettre en place un comportement désiré sur les autres rôles d'agents ou agents.
- Protocoles existants :
  - fipa-query: the receiver agent is requested to perform some kind of inform action
  - fipa-request: the receiver is requested to perform some action.
  - fipa-contract-net: an agent (manager) solicits proposals from other agents (contractors) by specifying the task and the conditions placed by the manager upon the execution of the task. The contractor's proposal includes the preconditions that the contractor is setting out for the task (e.g., price, time, etc.)
  - …

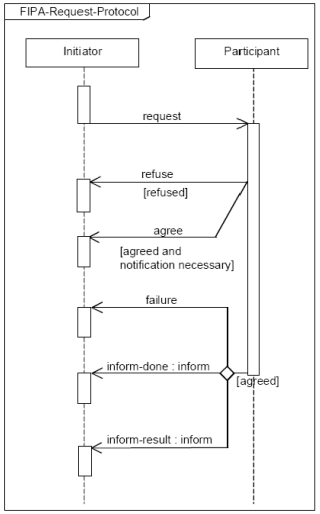
```
(request  :sender A :receiver B  :content (some act)
          :protocol fipa-request )
```

Protocoles spécifiés

Interaction Protocol name:	FIPA doc. name:	Modification date	Status
FIPA-Request	IP00102A	October 2000	Experimental
FIPA-Query	IP00103A	October 2000	Experimental
FIPA-Request-When	IP00104A	October 2000	Experimental
FIPA-Contract-Net	IP00105A	October 2000	Experimental
FIPA-Iterated-Contract-Net	IP00106A	October 2000	Experimental
FIPA-Auction-English	IP00107A	October 2000	Experimental
FIPA-Auction-Dutch	IP00108A	October 2000	Experimental
FIPA-Brokering	IP00109A	October 2000	Experimental
FIPA-Recruiting	IP00110A	October 2000	Experimental
FIPA-Subscribe	IP00111A	October 2000	Experimental
FIPA-Propose	IP00112A	October 2000	Experimental

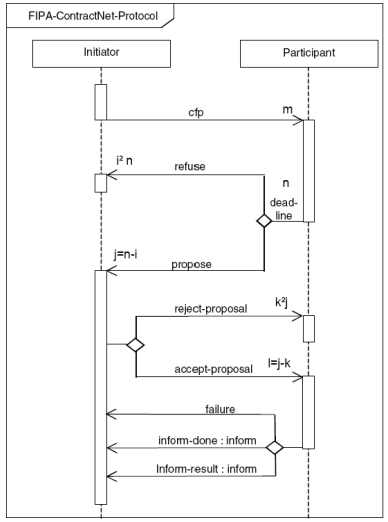
Protocole d'interaction de requête

- Identifié par `fipa-request` dans le champ `protocol` d'un message ACL
- A tout point du protocole un message de performatif `not-understood` peut être envoyé
- A tout point du protocole un protocole d'annulation peut être déroulé `fipa-cancel-meta-protocol`



Protocole d'interaction Contract-Net

- Identifié par `fipa-contract-net` dans le champ `protocol` d'un message ACL
- A tout point du protocole un message de performatif `not-understood` peut être envoyé
- A tout point du protocole un protocole d'annulation peut être déroulé `fipa-cancel-meta-protocol`



Pour aller plus loin …

- [FIPA0001] Specification No. SC00001, FIPA Abstract Architecture Specification.
- [FIPA0008] Specification No. SC00008, FIPA SL Content Language Specification.
- [FIPA0014] Specification No. SI00014, FIPA Nomadic Application Support Specification.
- [FIPA0023] Specification No. SC00023, FIPA Agent Management Specification.
- [FIPA0026] Specification No. SC00026, FIPA Request Interaction Protocol Specification.
- [FIPA0037] Specification No. SC00027, FIPA Query Interaction Protocol Specification.
- [FIPA0028] Specification No. SC00028 , FIPA Request When Interaction Protocol Specification
- [FIPA0029] Specification No. SC00029, FIPA Contract Net Interaction Protocol Specification.
- [FIPA0030] Specification No. SC00030, FIPA Iterated Contract Net Interaction Protocol Specification..
- [FIPA0034] Specification No. SC00033, FIPA Brokering Interaction Protocol Specification.
- [FIPA0001] Specification No. SC00034, FIPA Recruiting Interaction Protocol Specification.
- [FIPA0035] Specification No. SC00035, FIPA Subscribe Interaction Protocol Specification.
- [FIPA0036] Specification No., SC00036, FIPA Propose Interaction Protocol Specification.

Pour aller plus loin …

- [FIPA0037] Specification No. , SC00037, FIPA Communicative Act Library Specification.
- [FIPA0061] Specification No., SC00061, FIPA ACL Message Structure Specification.
- [FIPA0067] Specification No., SC00067, FIPA Agent Message Transport Service Specification. Available
- [FIPA0069] Specification No., SC00069, FIPA ACL Message Representation in Bit-Efficient Specification
- [FIPA0070] Specification No., SC00070, FIPA ACL Message Representation in String Specification.
- [FIPA0071] Specification No., SC00071, FIPA ACL Message Representation in XML Specification.
- [FIPA0075] Specification No. SC00075, FIPA Agent Message Transport Protocol for IIOP
- Specification.
- [FIPA0084] Specification No., SC00084, FIPA Agent Message Transport Protocol for HTTP Specification.
- [FIPA0085] Specification No. SC00085, FIPA Agent Message Transport Envelope Representation in XML Specification.
- [FIPA0088] Specification No. SC00088, FIPA Agent Message Transport Envelope Representation in Bit Efficient Specification.
- [FIPA0009] Specification No. SI0009, FIPA Device Ontology Specification.
- [FIPA0094] Specification No. SC00094, FIPA Quality of Service Specification.

FIPA Interaction Protocols [XC00025D]

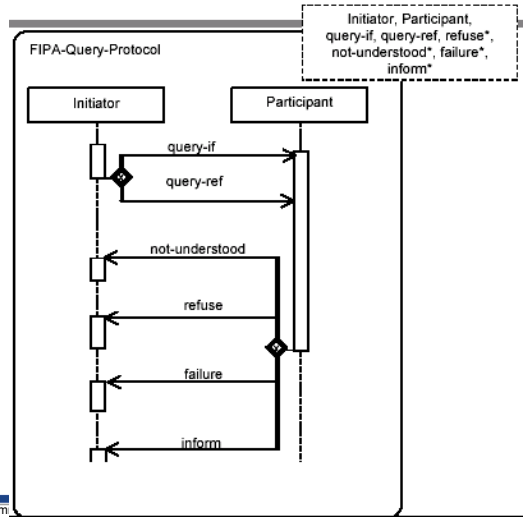
- Notation : AUML
- Sémantique
  - Une description de protocole représente un schéma d'interaction, ensemble de messages échangés entre différents rôles d'agent à l'intérieur d'une collaboration pour mettre en place un comportement désiré sur les autres rôles d'agents ou agents.
- Protocoles existants :
  - **fipa-query**: the receiver agent is requested to perform some kind of inform action
  - **fipa-request**: the receiver is requested to perform some action.
  - **fipa-contract-net**: an agent (manager) solicits proposals from other agents (contractors) by specifying the task and the conditions placed by the manager upon the execution of the task. The contractor's proposal includes the preconditions that the contractor is setting out for the task (e.g., price, time, etc.)
  - ..some more

```
(request  :sender A :receiver B  :content (some act)
          :protocol fipa-request )
```

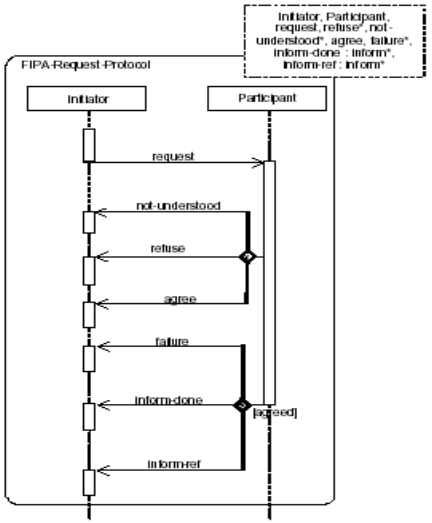

Protocoles spécifiés

Interaction Protocol name:	FIPA doc. name:	Modification date	Status
FIPA-Request	IP00102A	October 2000	Experimental
FIPA-Query	IP00103A	October 2000	Experimental
FIPA-Request-When	IP00104A	October 2000	Experimental
FIPA-Contract-Net	IP00105A	October 2000	Experimental
FIPA-Iterated-Contract-Net	IP00106A	October 2000	Experimental
FIPA-Auction-English	IP00107A	October 2000	Experimental
FIPA-Auction-Dutch	IP00108A	October 2000	Experimental
FIPA-Brokering	IP00109A	October 2000	Experimental
FIPA-Recruiting	IP00110A	October 2000	Experimental
FIPA-Subscribe	IP00111A	October 2000	Experimental
FIPA-Propose	IP00112A	October 2000	Experimental

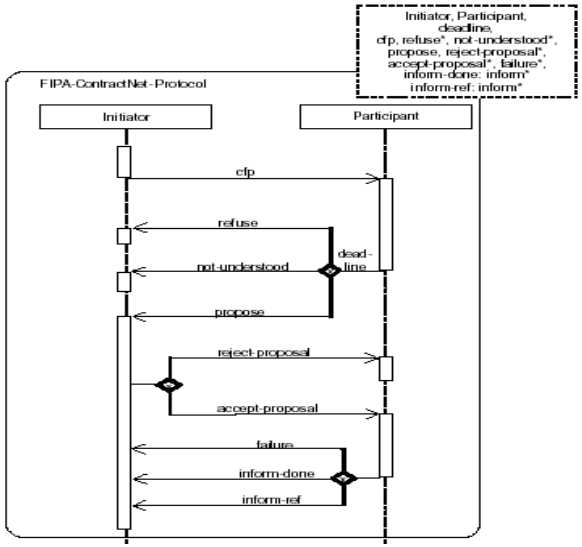
Query



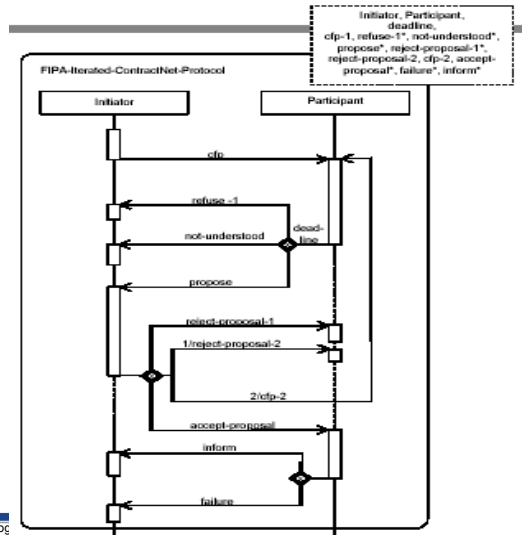
Request [XC00026E]



Contract Net Protocol [XC00029E]



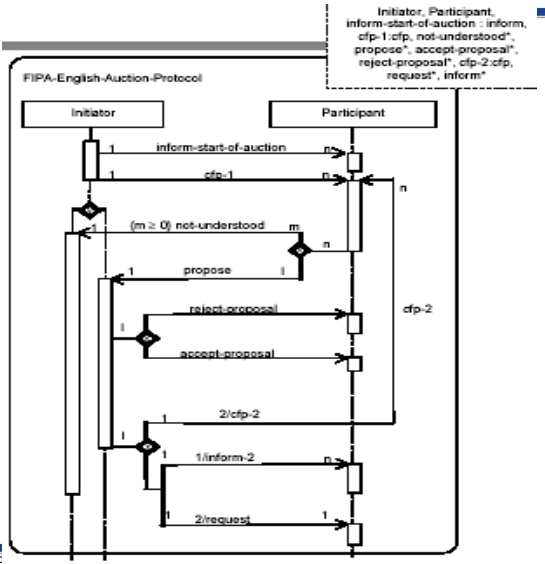
Iterated Contract Net (1)



Iterated Contract Net (2)

The FIPA Iterated Contract Net Interaction Protocol (IP) is an extension of the basic FIPA Contract Net IP (see [FIPA00029]), but it differs by allowing multi-round iterative bidding. As with the FIPA Contract Net IP, the manager issues the initial call for proposals with the *cfp* act (see [FIPA00037]). The contractors then answer with their bids as *propose* acts (see [FIPA00037]) and the manager may then accept one or more of the bids, rejecting the others, or may iterate the process by issuing a revised *cfp*. The intent is that the manager seeks to get better bids from the contractors by modifying the call and requesting new (equivalently, revised) bids. The process terminates when the manager refuses all proposals and does not issue a new *cfp*, accepts one or more of the bids or the contractors all refuse to bid.

English Auction (1)



English Auction (2)

In the FIPA English Auction Interaction Protocol (IP), the auctioneer seeks to find the market price of a good by initially proposing a price below that of the supposed market value and then gradually raising the price. Each time the price is announced, the auctioneer waits to see if any buyers will signal their willingness to pay the proposed price. As soon as one buyer indicates that it will accept the price, the auctioneer issues a new call for bids with an incremented price. The auction continues until no buyers are prepared to pay the proposed price, at which point the auction ends. If the last price that was accepted by a buyer exceeds the auctioneer's (privately known) reservation price, the good is sold to that buyer for the agreed price. If the last accepted price is less than the reservation price, the good is not sold. In Figure 1, the auctioneer's calls, expressed as the general cfp act (see [FIPA00037]), are multicast to all participants in the auction. For simplicity, only one instance of the message is portrayed. Note also that in a physical auction, the presence of the auction participants in one room effectively means that each acceptance of a bid is simultaneously broadcast to all participants and not just the auctioneer. This may not be true in an agent marketplace, in which case it is possible for more than one agent to attempt to bid for the suggested price. Even though the auction will continue for as long as there is at least one bidder, the agents will need to know whether their bid (represented by the propose act - see [FIPA00037]) has been accepted. Hence the appearance in the IP of the accept-proposal (see [FIPA00037]) and reject-proposal acts (see [FIPA00037]), despite this being implicit in the English Auction process that is being modelled. Note that the proposals that are submitted by the bidders primarily concern the bidding process. In response to a cfp to submit bids to purchase a good X, a proposal would be something of the order: "I propose that the bidding level be raised to purchase price Z and I assert that I am able to pay Z for X." This allows the auctioneer to be confident that the bidder can indeed pay the price without committing to actually paying it until the auctioneer specifically requests X (at price Z) from the winning bidder. At the end of the IP, the auctioneer will typically enter a request IP (see [FIPA00026]) with the winning bidder to complete the auction transaction.

Plan

- ✓ Introduction
- ✓ Fondement
- ✓ Langage de Communication Agent
- ✓ Protocoles d'interaction
- ➢ **Conclusion**

Synthèse

- De nombreuses formes d'interaction,
- Pas de formalisme unique pour leur définition,
- Contrôle des interactions par la définition de protocoles d'interaction,
- Définition de systèmes de gestion de conversation,
- Des liens à établir avec les systèmes répartis pour la mise en oeuvre.

Défis

- Meaning negotiation
- Ontology
- Dialogism – Dialogue Construction
- Conversation Policies

Bibliographie

- [Barbuceanu 95] M. Barbuceanu, M.S. Fox "Cool : a language for describing coordination in multi-agent systems", ICMAS 95, p 17-24
- [Boissier 97] O. Boissier, Y. Demazeau, "Une architecture multi-agent pour des systèmes de vision ouverts et décentralisés", TSI, Octobre 97
- [Burmeister 93] B. Burmeister, A. Haddadi, K. Sundermeyer, "Generic Configurable Cooperation Protocols for Multi-Agent Systems", MAAMAW'93
- [Demazeau 93] Y. Demazeau, "La plate-forme PACO et ses applications", 2ème journée nationale du PRC-IA sur les Systèmes Multi-Agents, PRC-IA, Montpellier, Décembre 1993.
- [Drogoul 93] A. Drogoul, "De la simulation multi-agent à la résolution collective de problèmes. Une étude de l'émergence de structures d'organisation dans les systèmes multi-agents", Thèse de l'Université Paris 6
- [Durfee 87] E.H. Durfee, V.R. Lesser, D.D. Corkill, "Cooperation through Communication in a Distributed Problem Solving Network", Distributed Artificial Intelligence, M. Huhns editor, Pitman.
- [Erman 80] L.D. Erman and al. "The Hearsay II Speech Understanding System : Integrating Knowledge to resolve Uncertainty" ACM, Computing Surveys, vol 12, p 212-253, 1980.
- [Ferber 95] J. Ferber, "Les systèmes multi-agents, vers une intelligence collective", InterEditions, 1995.

Bibliographie (suite)

- [FIPA 97] FIPA, ‘‘Agent Communication Language’’ Specification FIPA, 28/11/97
- [Georgeff 83] M.P. Georgeff, "Communication and Interaction in Multi-Agent Planning", AAAI 83, p 125-129
- [Greaves:00] M. Greaves, H. Holmback, J. Bradshaw « What is a Conversation Policy ? », Issues in Agent Communication, F. Dignum, M. Greaves editors, LNAI 1916, p. 118—131, Springer Verlag, 2000.
- [Hayes-Roth 85] B. Hayes-Roth, "A Blackboard Architecture for Control", Artificial Intelligence, 26(3), p 251-321, 1985
- [Hewitt 77] "Viewing Control Structures as Patterns of Message Passing", Artificial Intelligence, 8(3), p. 323-374, 1977
- [Labrou 96] Y. Labrou, "Semantics for an agent communication language", Phd, University of Maryland, Septembre 1996
- [Labrou 99] Y. Labrou, T. Finin, Y. Peng, “Agent Communication Languages : the Current Landscape”, IEEE Intelligent Systems, p. 45-52, March/April 1999.
- [Pesty 00] S. Pesty, J.L. Koning, “Modèles de communication”, Systèmes Multi-Agents, J.P. Briot, Y. Demazeau eds, A paraître.
- [Rosenschein 85] J.S.Rosenschein, "Rational Interaction : cooperation among intelligent agents", SRI Technical Report STAN-CS-851081 1985
- [Searle 72] J.R. Searle, "Les actes de langage", 1972
- [Singh 98] M.P. Singh, “Agent communication languages : rethinking the principles”, IEEE computer, p. 40-47, December 1998
- [Vanderveken 88] D. Vanderveken, "Les actes de discours", P. Mardaga Ed. 1988.