

Interaction ;
Conversation entre les agents

Définition

□ Interaction

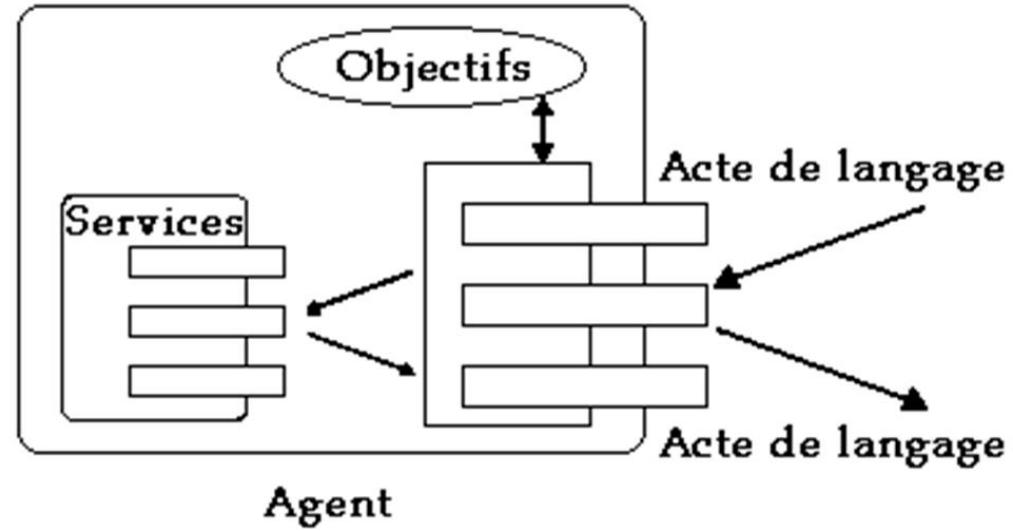
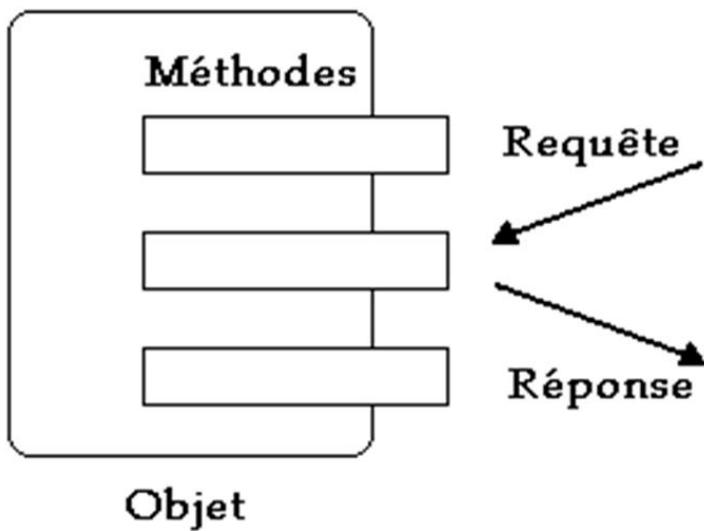
- Mise en relation dynamique de deux ou de plusieurs agents par le biais d'un ensemble d'actions réciproques.
 - Il y a interaction lorsque la dynamique propre d'un agent est perturbée par les influences des autres
- L'interaction est le moteur d'un système multi-agents.
- Selon les agents et les systèmes, l'interaction prend diverses formes: actions sur l'environnement, communication, inférences,....

Agent communicant

Découplage entre la requête et la réponse

Objet : L'appel d'une méthode est synchrone

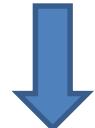
Agent : L'appel d'un service est asynchrone



Communication dans les SMAs

Pourquoi ?

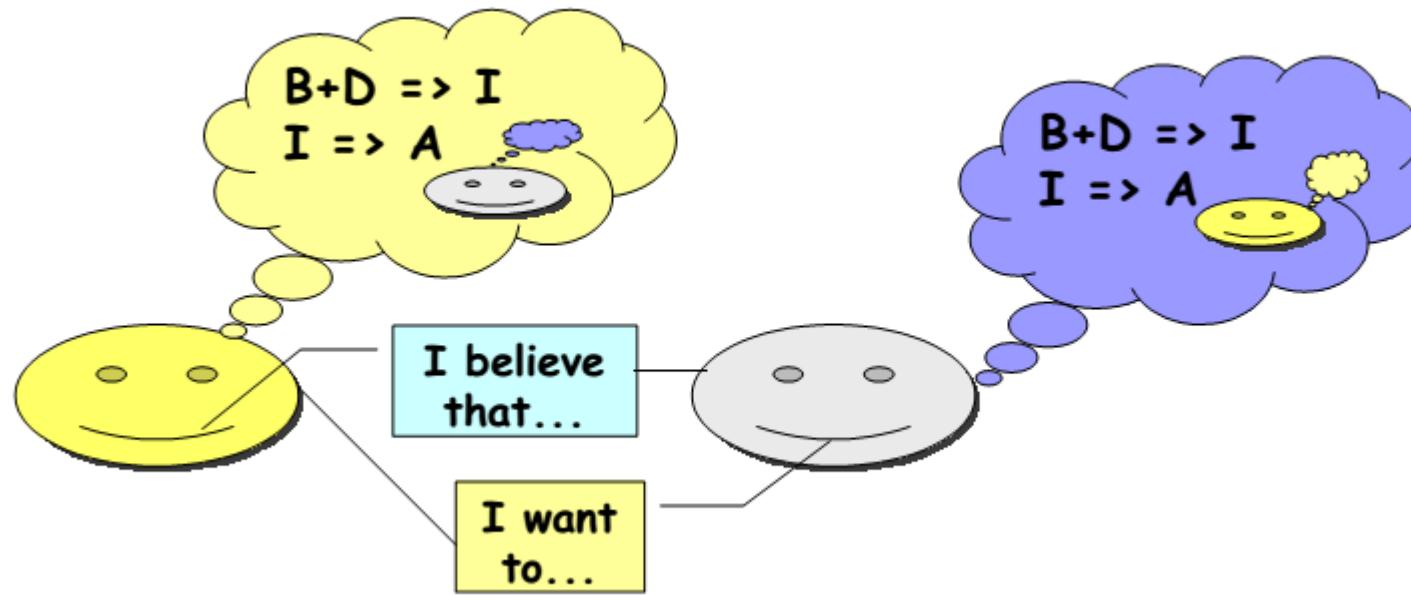
- ❑ Coordination entre les agents; un *SMA* peut réaliser plus de choses qu'un seul agent
- ❑ Plus riche et plus efficace que d'observer des changements dans l'environnement
- ❑ Diversité et hétérogénéité des agents
- ❑ Autonomie, Intérêts propres des agents, Evolution du monde rendent les comportements imprévisibles et conduire à des solutions inacceptables.



Les agents communicants doivent comprendre un langage commun.

Communication dans les SMAs

Pourquoi ?



□ Communiquer pour :

- Révéler aux autres son état *BDI*
- Essayer de modifier l'état *BDI* des autres

Interactions directes

Qui? Quoi? Comment?

Principe

- Approche classique de communication adressée : un émetteur envoie un message à un récepteur localisé par son adresse.
- Interaction dirigée par l'émetteur
- Modalités
 - Point à point, diffusion.

Interactions directes

Qui ? Problème de connexion

Problématique:

Avec quel agent interagir pour obtenir un service, une ressource ...?

Solutions :

- Gestion des connaissances sociales.
- Protocoles.
- Service offert par la plateforme.

Aspects techniques (1)

- Ne concerne pas les services des niveaux bas et intermédiaires des logiciels distribués (protocoles réseaux, transport,)
- Elle suppose l'existence de :
 - Protocole de transport pour le routage des messages (*SMTP, TCP/IP, HTTP, IIOP, etc*)
 - Identification de l'émetteur
 - Respect de l'ordre des messages en émission et réception
 - Fiabilité de la couche de transport
 - Mécanisme pour conduire et contrôler la communication au sein des agents

Aspects techniques (2)

- Échanges directs par messages
 - Point à point / diffusion
 - Synchrone / asynchrone
 - Désignation des agents (facilitateurs)

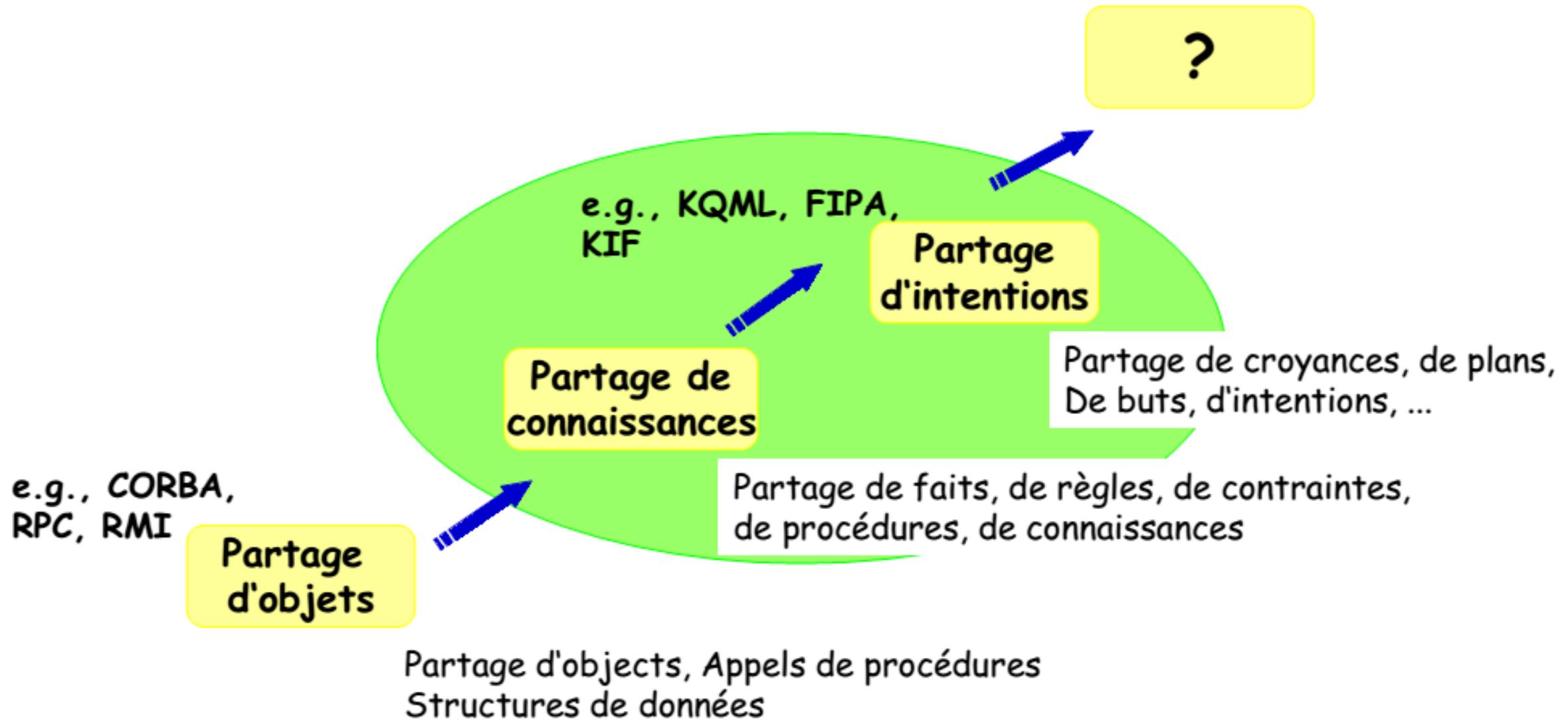
- Échanges indirects exemple: par mémoire partagée.
 - Ex: architecture de tableaux noirs

○ Point à point / diffusion

- Quelle est la relation qui unit le receveur d'une communication à son émetteur?
- Lorsque le destinataire est connu de l'émetteur, ce dernier peut lui adresser des messages en particulier et ainsi instaurer une communication individuelle.
- ✓ On dit alors que les communications sont effectuées selon un mode *point à point*.
- ✓ C'est ce type de communication qui est généralement le plus employé dans les agents cognitifs. On le notera ainsi:

(id) <émetteur> : <destinataire> << <énoncé>
- Si le destinataire n'est pas connu de l'émetteur, le message est envoyé sous un **mode** appelé **diffusion** (*broadcasting*) à tout un ensemble d'agents, ces derniers étant liés à l'émetteur par une relation de voisinage (proximité dans l'environnement, connexion dans un réseau, etc.).

Panorama



Sources

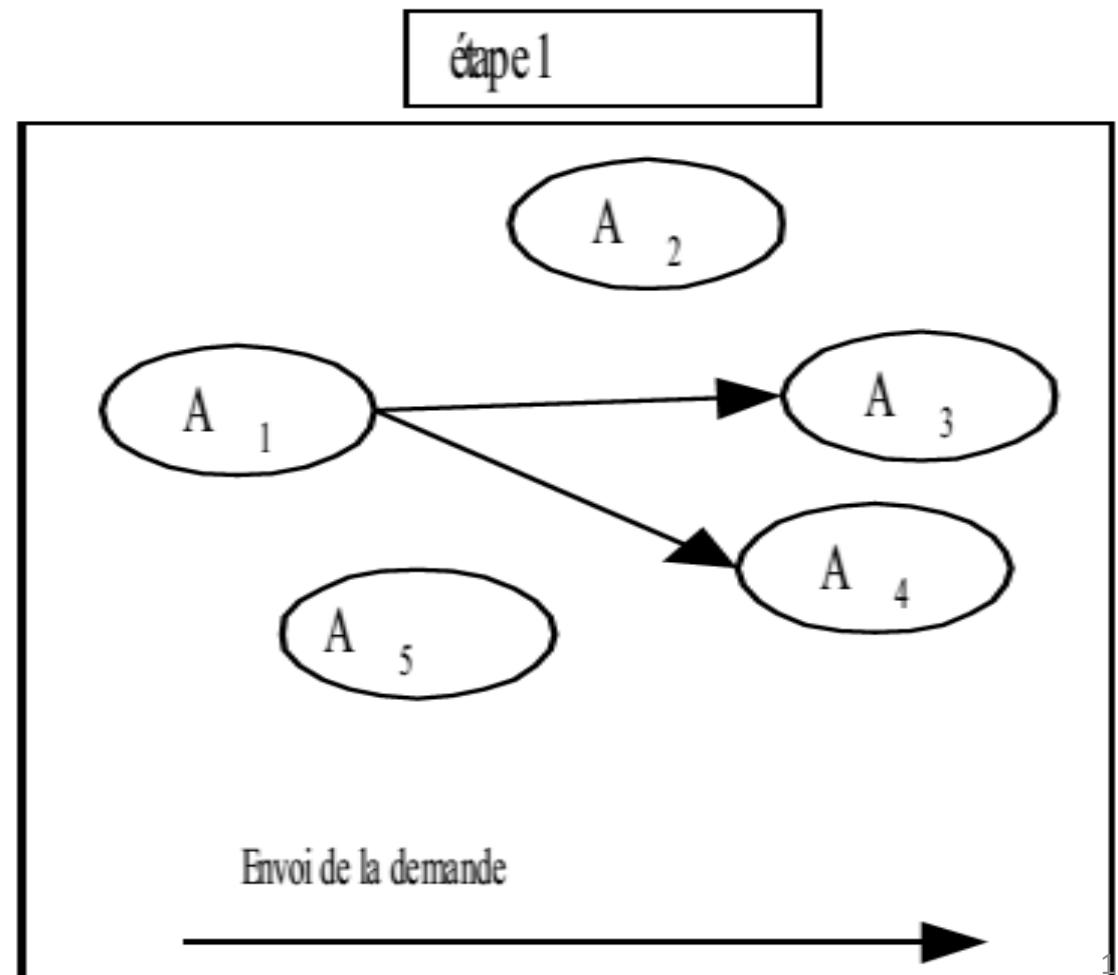
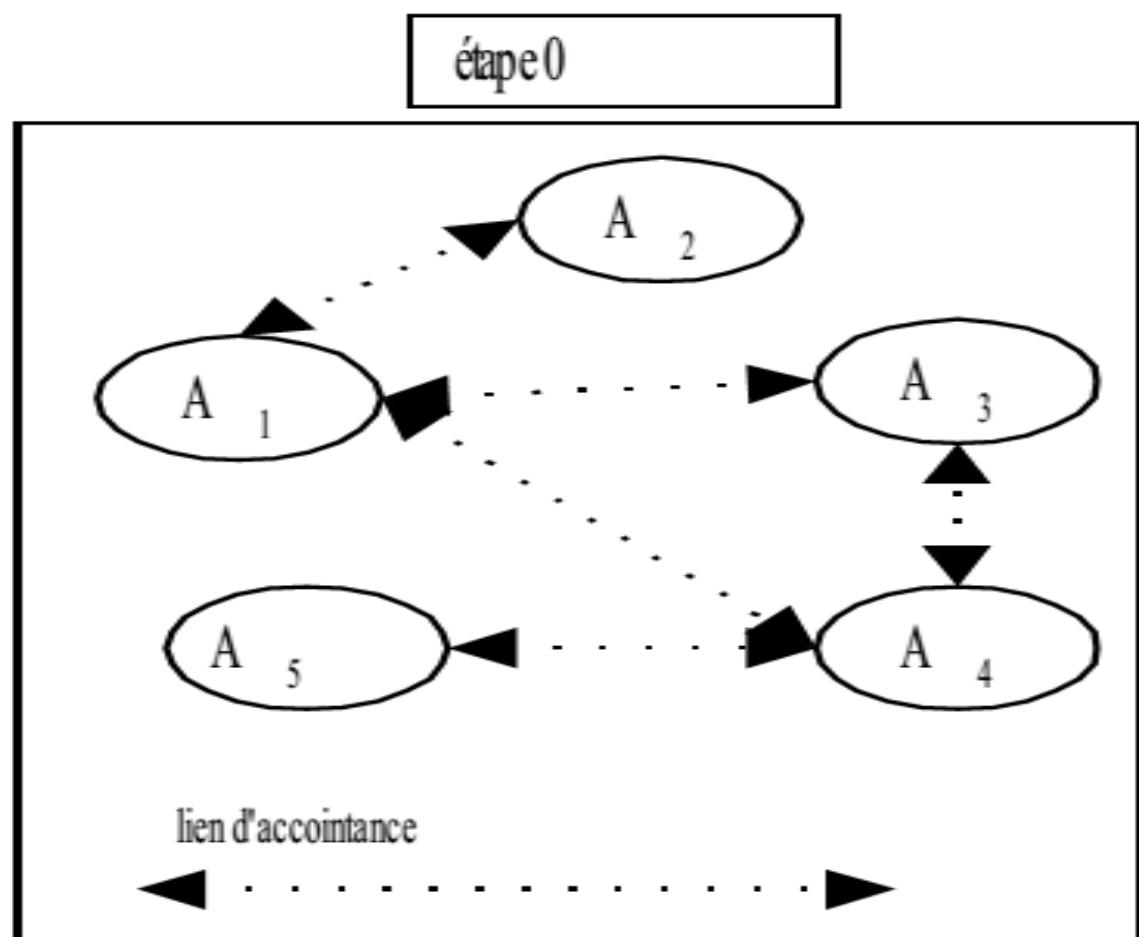
- Recherches en traitement des langues naturelles, en pragmatique conversationnelle
Sur les intentions dans les communications,...
- Prise en compte des états mentaux (ex: *BDI*):
 - Croyances, intentions, désirs,...
- ✓ Utilisation de la théorie des actes de langage (*Speech Acts*)[*Austin 62, Searle 72, Vanderveken 88*]
- ✓ Interactions au sein de conversations

Interaction directe

Accointances

Principe :

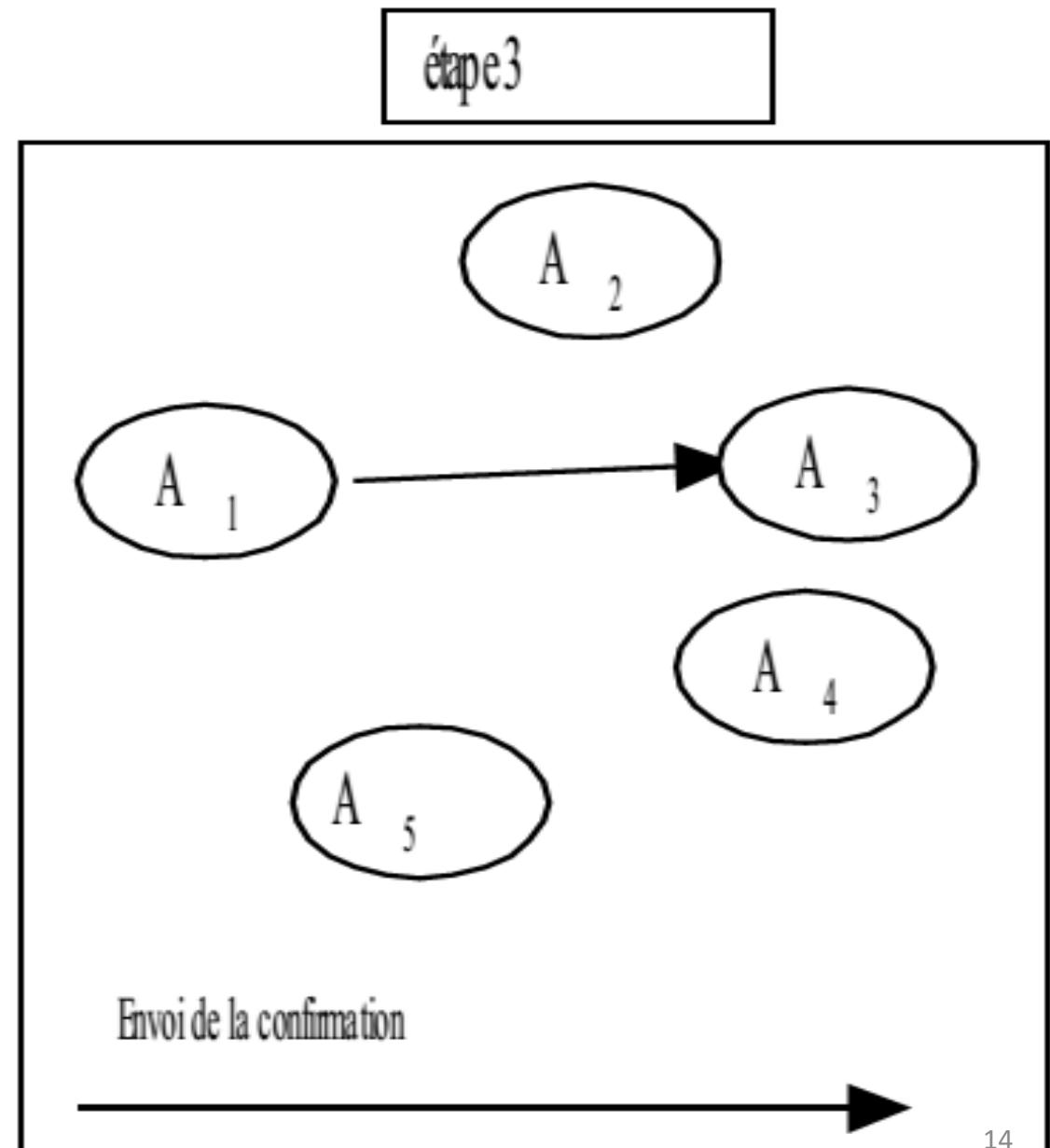
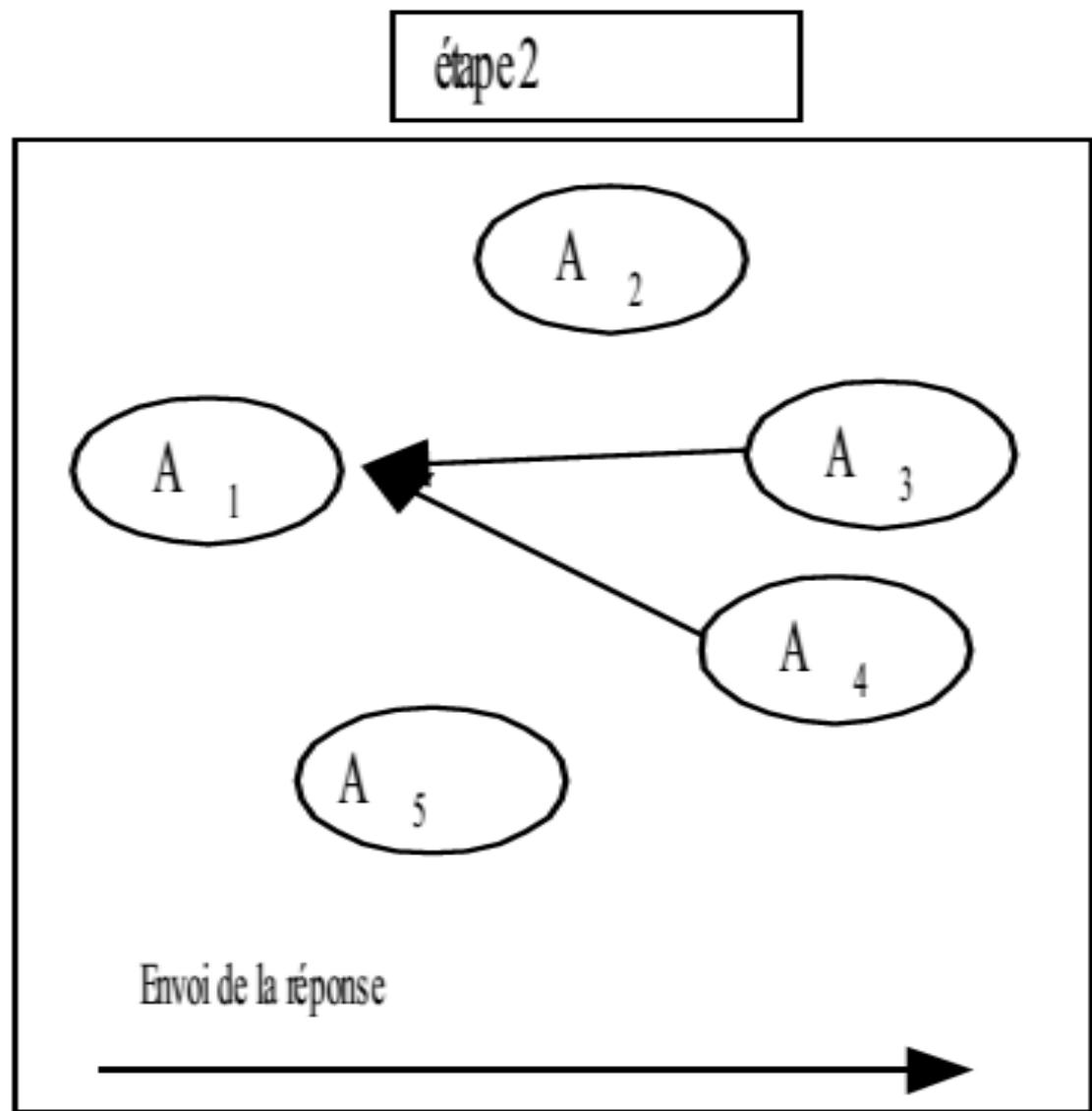
Les connaissances sociales d'un agent représentent ses accointances avec les autres agents du système.



Interaction directe

Accointances

Principe :



Facilitateurs (Middle-Agent)

□ Principe :

stocker la connaissance sur les compétences dans des agents spécialisés.

Modèle de coordination fondé sur les compétences (capability-based coordination)

□ Exemple *KQML Facilitateurs*

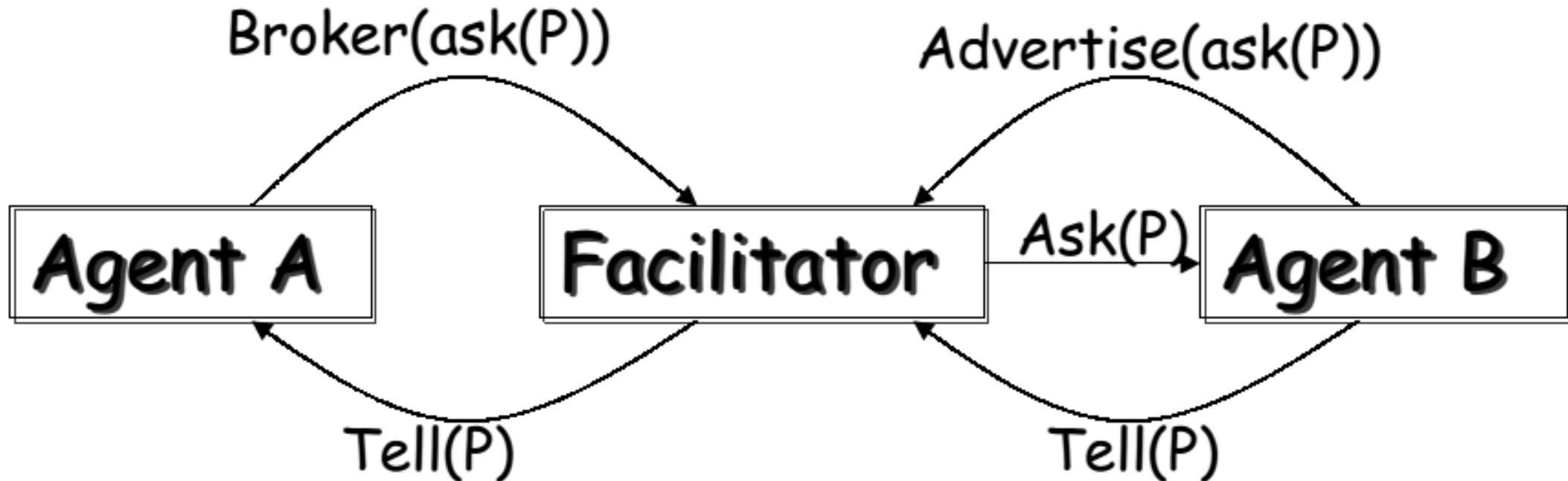
Facilitateurs sont des agents particuliers qui

- Échangent des métainformations sur les autres agents
- Fournissent des services de communication tels que :
 - *Message forwarding*
 - *Matchmaking*
 - *Resource discovery*

□ Facilitateurs peuvent être seulement des pages jaunes ou des agents intelligents

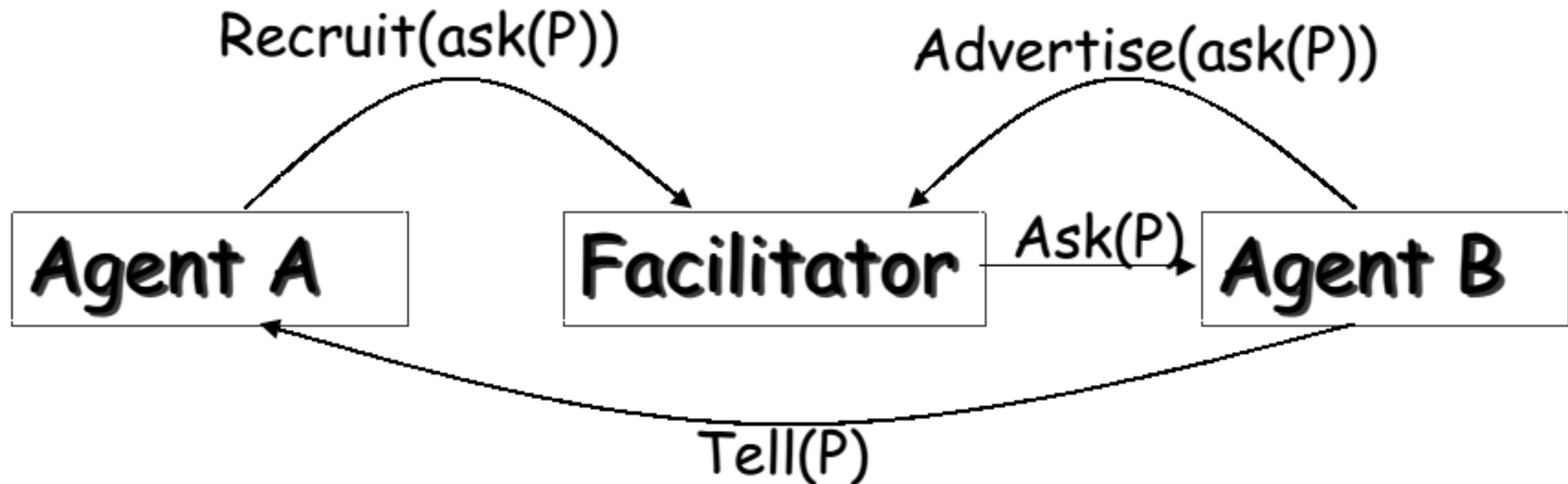
ACL : KQML Facilitateurs (2)

Performatifs : Broker



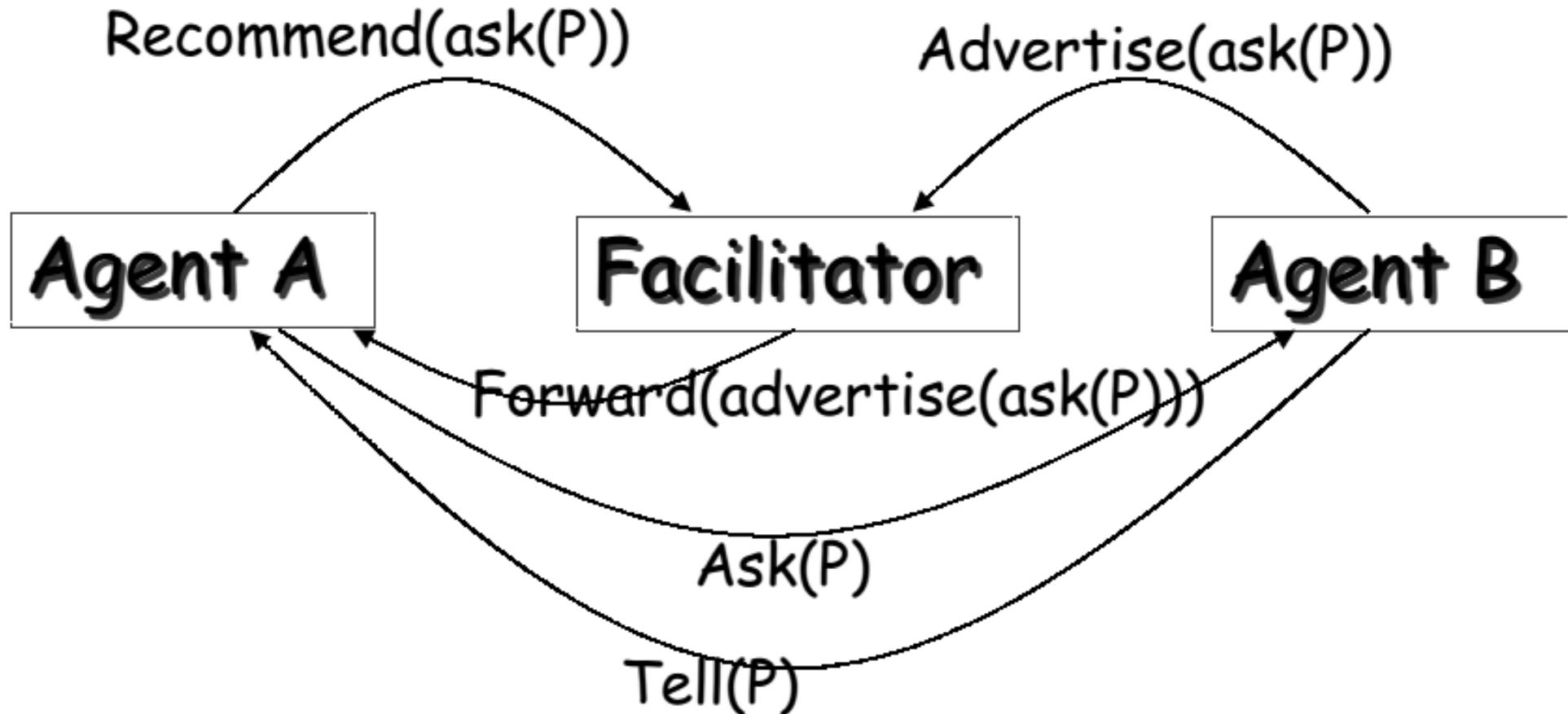
ACL : KQML Facilitateurs (3)

Performatifs : Recruit



ACL : KQML Facilitateurs (4)

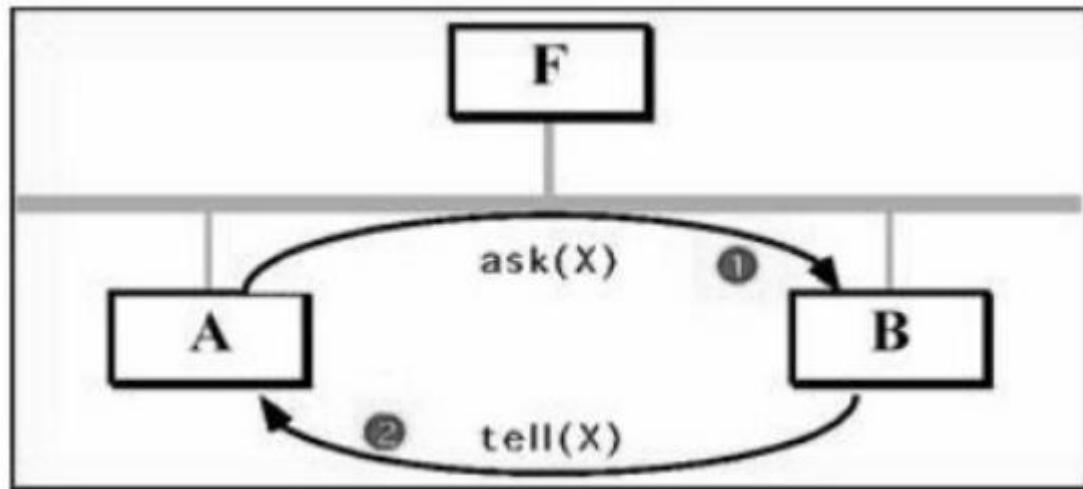
Performatifs : *Recommend*



Exemples de patrons d'interaction *KQML* (1)

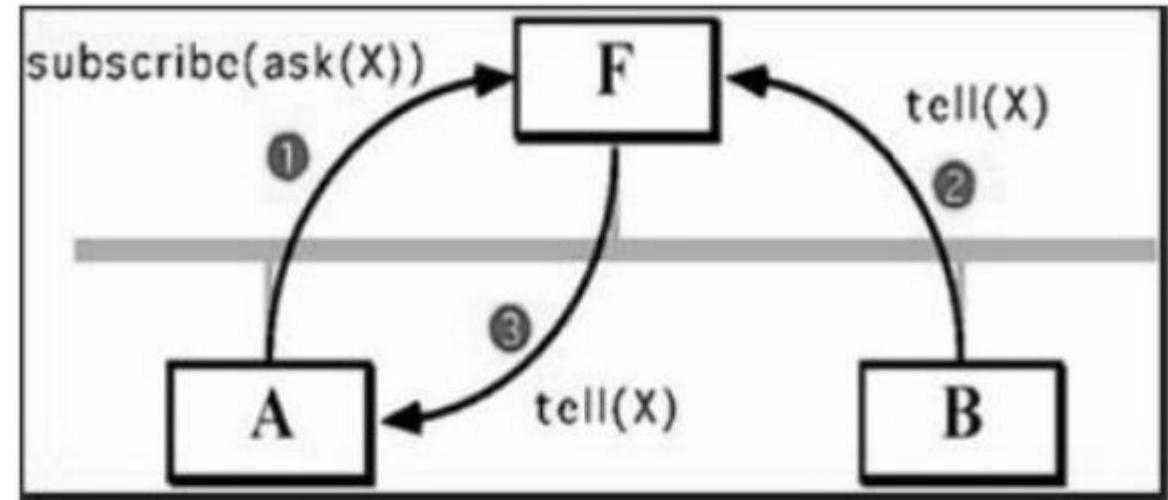
Ask & Tell

A sait que *B* existe et lui demande *X*



Subscribe

A demande à l'agent *F* de l'informer lorsqu'il aura la connaissance de *X*

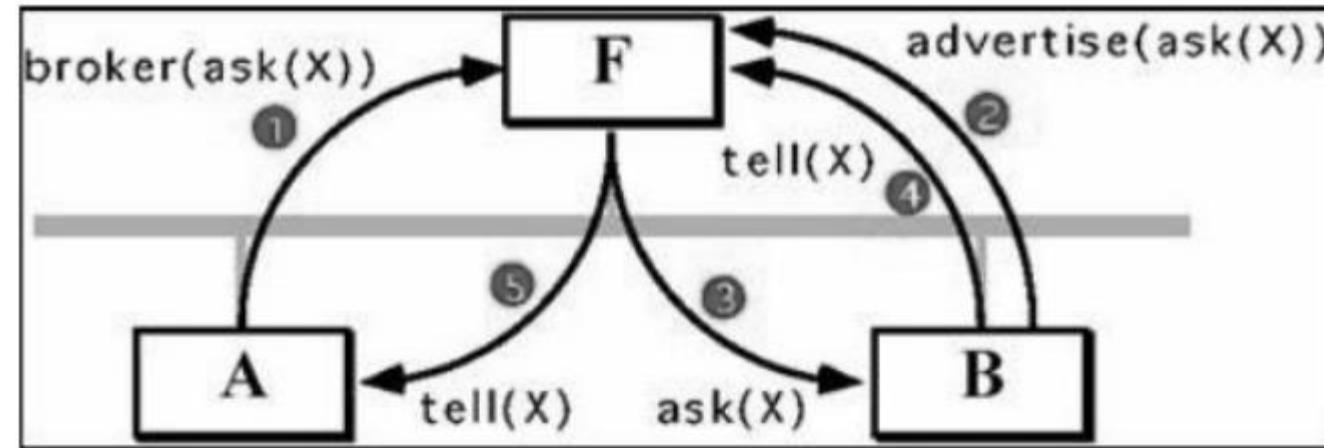


Exemples de patrons d'interaction KQML (2)

- A demande à F si il peut lui fournir X par son réseau d'accointances

Broker & Advertise

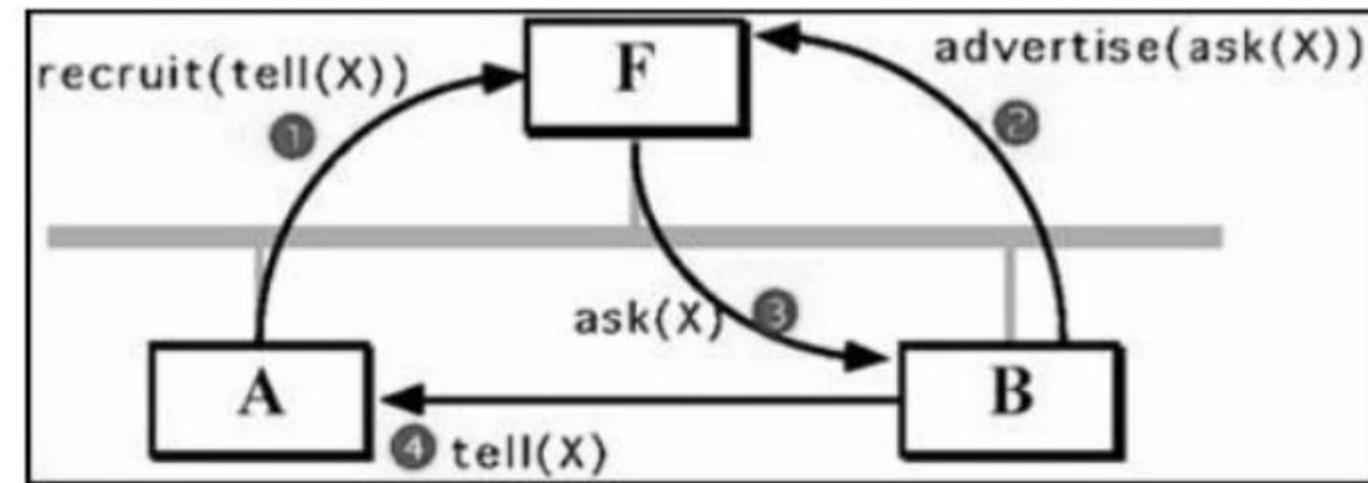
F demande X à un agent B mais est utilisé comme intermédiaire



- A obtient X directement par un agent du réseau d'accointance de F

Recruit

F demande à un agent B de donner X directement à A



ACL : KQML Sémantique (1)

- **Préconditions**: indiquent les états nécessaires dans les agents pour que le message puisse être créé
- **Post conditions**: décrivent les états des agents après le traitement du message
- **Complétion**: décrit le résultat attendu du message

Les limites sémantique de *KQML*

Sémantique a été spécifiée par la suite (93 → 98)

- **Ambiguité et imprécision des performatifs:** La signification des performatifs est donnée sous la forme de simples descriptions en langage naturel, ce qui les rend souvent vagues et confus
- **Certains sont inutiles et incohérents,** particulièrement les performatifs *achieve*, *broker* et *stream-all*, qui ne peuvent être compris comme des actes de langage indirects puisqu'ils satisfont en fait des buts appartenant à d'autres agents.
- **Absence de certains performatifs (promissif):** Par exemple, il n'est pas possible en *KQML* de dire que l'on s'engage, auprès d'un tiers, à accomplir une action.

ACL : KQML Sémantique (2)

Exemple

Tell(A,B,X)

- Pre(A): $\text{bel}(A,X) \wedge \text{know}(A, \text{want}(B, \text{know}(B, \text{bel}(A,X)) \vee \text{know}(B, \neg \text{bel}(A,X))))$
- Pre(B): $\text{intend}(B, \text{know}(B, \text{bel}(A,X)) \vee \text{know}(B, \neg \text{bel}(A,X)))$
- Post(A): $\text{know}(A, \text{know}(B, \text{bel}(A,X)))$
- Post(B): $\text{know}(B, \text{bel}(A,X))$
- Completion: $\text{know}(B, \text{bel}(A,X))$

ACL : KQML Sémantique (3)

□ Problèmes (Ambigüité)

- *Deny* ($A, B, Tell(A, B, X)$)
 - Does A deny the telling or
 - Does A deny the fact X ?
 - *Uninsert* (A, B, X) //only possible after *insert* (A, B, X)
 - $Post(B)$: $\neg bel(B, X)$
 - What if B can infer X from other sources ?
- Pas de sémantique formelle, pas d'infrastructure pour la gestion des agents, ...

- **Non-prédicibilité de la réponse** : La sémantique de l'interaction ne décrit pas la réponse, seulement les conditions et effets du point de vue de l'agent émetteur !
- **L'agent peut répondre ce qu'il veut**
 - définition de protocoles en particulier, *SMA* faiblement couplés
- **L'agent peut ne pas répondre**
 - attente et timeout en particulier, *SMA* ouverts

Protocoles d'interaction

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

- Un protocole d'interaction spécifie des règles qui doivent être respectées par les agents durant une conversation, et définit ainsi pour chaque étape les types de messages qui peuvent être envoyés.

Les conversations

- La théorie des actes de langage telle qu'elle a été développée par *Austin, Searle, Vanderveken, Récanati* et d'autres ne prend en compte que l'acte isolé,
- Elle ne traite absolument pas de la séquence d'interactions qui s'établit entre les interlocuteurs lors de leurs communications ni de leurs attentes réciproques.
- une extension de la théorie des actes de langage qui s'appuie sur la définition de protocoles de communication, en considérant que tout acte de langage suppose un certain enchaînement possible des énonciations et qu'il engendre certaines modifications dans l'état mental des interlocuteurs.
- Dans ce cadre, les actes de langage sont les unités élémentaires permettant d'analyser les conversations.

Les conversations

En effet, un acte de langage n'est pratiquement jamais accompli isolément, et il se trouve souvent à l'origine d'autres actes.

Par exemple

- la demande (*M5*) $A: B \ll DemanderFaire(P)$ est un acte de demande d'action et se trouve à l'origine de toute une série d'actions ultérieures: l'acceptation ou le refus de B d'effectuer P , éventuellement la réalisation de la tâche P et sa signalisation à A .
- Ces conséquences sont importantes car elles entraînent des attentes de la part du locuteur.

Exemple :

- si l'agent B accepte de faire P , A peut supposer que l'action P sera terminée à la date $D+T$ où D est la date de début de l'action P et T est le temps normal de réalisation de P .

Par exemple

- si A est le propriétaire d'une maison, B un peintre et P l'action de peindre le salon de la maison de A , le message ***M3***, s'il est accepté par B peut amener A à penser que son salon sera repeint après un certain temps. La définition de ce temps peut lui même faire l'objet de transactions ou bien se référer au temps normal d'exécution, c'est-à-dire faire référence à une conception standard, partagée par les interlocuteurs, des caractéristiques normales de la réalisation de la tâche en question.

En effet, un agent A qui pose une question à un agent B , c'est-à-dire qui émet un message de la forme:

(M6) A: B << Questionner(quel est l'ensemble $\{x \mid P(x)\}$)

A attend l'une des trois réactions suivantes:

1. La réponse à la question:

(M6) B: A << Réponse(M6, a₁, ..., a_n) où les a₁, ..., a_n sont les réponses à la question précédente, c'est-à-dire les valeurs de x qui satisfont $P(x)$;

2. Un refus de répondre à la question, éventuellement adjoint d'une explication:

(M7') B: A << RefuserIncompétentDemande(M6)

3. Une demande de renseignements complémentaires (en passant au niveau méta, puisqu'il est nécessaire de parler de la communication):

(M7") B: A << MétaQuestionner(M6, Arguments(P))

De la communication à la conversation : «Protocole d'interaction »

- ❑ 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.
 - Protocole 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

Définitions

Conversation

séquence de messages échangés entre plusieurs agents, conforme à un protocole d'interaction.

Protocole d'interaction

un ensemble de règles plus ou moins génériques qui sont partagées (conventions sociales) par les agents.

- ❑ Un protocole a un **nom**.
- ❑ Un protocole affecte à un participant un **rôle**.
- ❑ Un protocole a un comportement qui décrit la séquence des messages qui peuvent être échangées spécifiée dans un langage (par ex : *AFD*, *AUML*, *RdP*, . . .)

Protocole d'interaction

La modélisation de ces conversations passe notamment par la définition de protocoles, c'est-à-dire de séquences valides de messages. Il existe plusieurs manières de décrire des protocoles, mais les plus courantes utilisent des automates à états finis ou des réseaux de Pétri.

Protocole d'interaction : Automates à états finis

L'intérêt de la notion d'automate à états finis est, d'une part, de pouvoir décrire très facilement le comportement d'un agent capable de mémoriser l'état dans lequel il se trouve

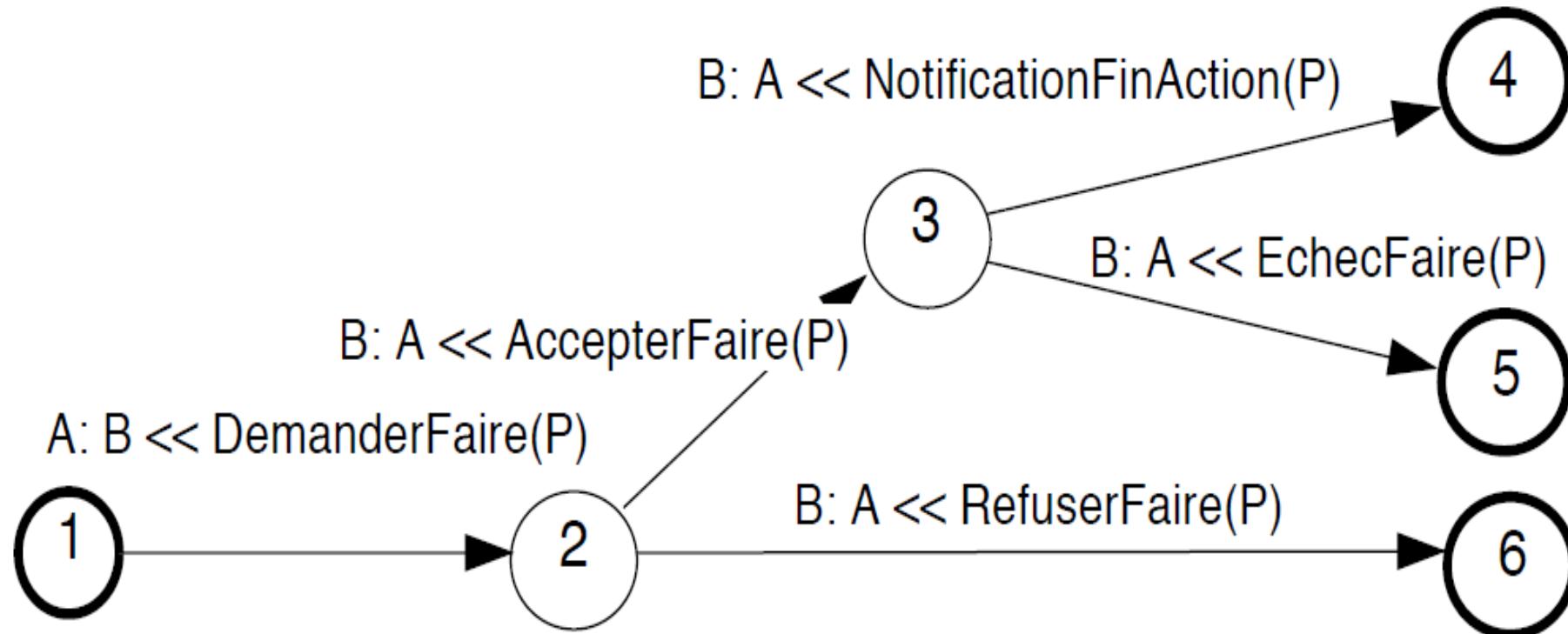
Etats et transitions

- Un automate à états finis est décrit par l'ensemble de ses états et par ce que l'on appelle sa fonction de transition qui spécifie l'état suivant en fonction de l'état actuel et des informations qu'il reçoit en entrée.
- On représente un automate à états finis sous la forme d'un graphe orienté dont les nœuds sont les états de l'automate et les arcs ses transitions.
- Sur les arcs, on indique à la fois les évènements qui le font passer d'un état à un autre et les actions qui doivent être entreprises (entreprendre) lors de cette transition.

Protocole d'interaction : Automates à états finis

Description d'une conversation par une suite d'états liés par des transitions (interactions entre les agents).

Exemple Automate de description d'une conversation commençant par un exercitif (ordonner)



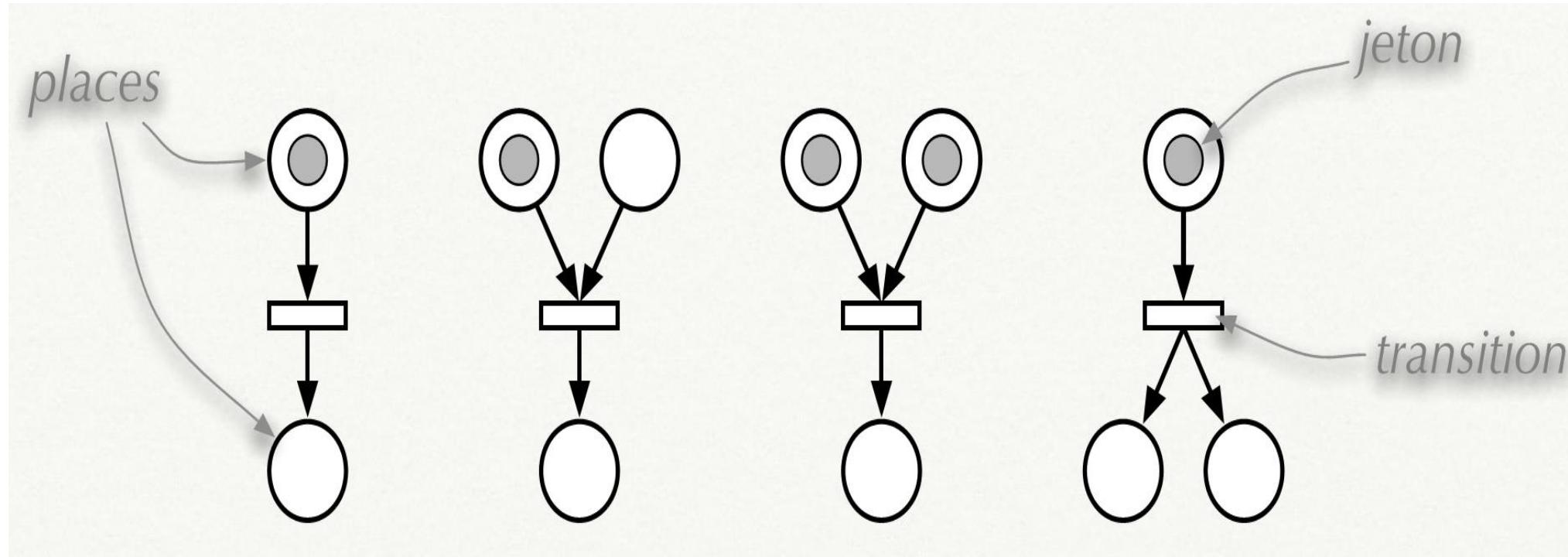
Protocole d'interaction : Automates à états finis

- Initialement la conversation se trouve dans l'état 1.
- Puis A , en demandant à B d'accomplir une action P , entame (يبدأ) la conversation, qui passe alors dans l'état 2.
- Plusieurs possibilités s'ouvrent alors:
 - B peut accepter, ce qui l'engage auprès de A , ou
 - B peut refuser (par exemple en se considérant incomptétent pour accomplir cette action), ou
 - Il est aussi possible pour B de passer au niveau méta en demandant des précisions supplémentaires, mais ceci n'est pas pris en compte dans le diagramme.
- En fonction des réponses de B , la conversation passera dans l'état 3 ou 6, déterminant ainsi les suites de communications possibles.
- La conversation est considérée comme terminée lorsque l'automate atteint l'un de des états terminaux 4, 5 et 6.

Protocole d'interaction : Réseau de Pétri

- Un réseau de Pétri est défini comme un graphe orienté comprenant deux sortes de nœuds: des places et des transitions.
- Ce graphe est constitué de telle sorte que les arcs du graphe ne peuvent relier que des places aux transitions ou des transitions aux places, suivant en cela les règles de formation des réseaux de transitions augmentés définis précédemment.
- On représente graphiquement les places par des cercles et les transitions par des barres.
- Les réseaux de Pétri, à l'encontre des automates, permettent la représentation des aspects dynamiques des processus en déplaçant des marques de places en places.
- ces marques ne sont que des jetons, c'est-à-dire des éléments simples et indiscernables.

Protocole d'interaction : Réseau de Pétri



Protocole d'interaction : Réseau de Pétri

Une conversation est une suite d'états liés par des transitions.

Places

- État interne de l'agent
- Message en cours d'acheminement

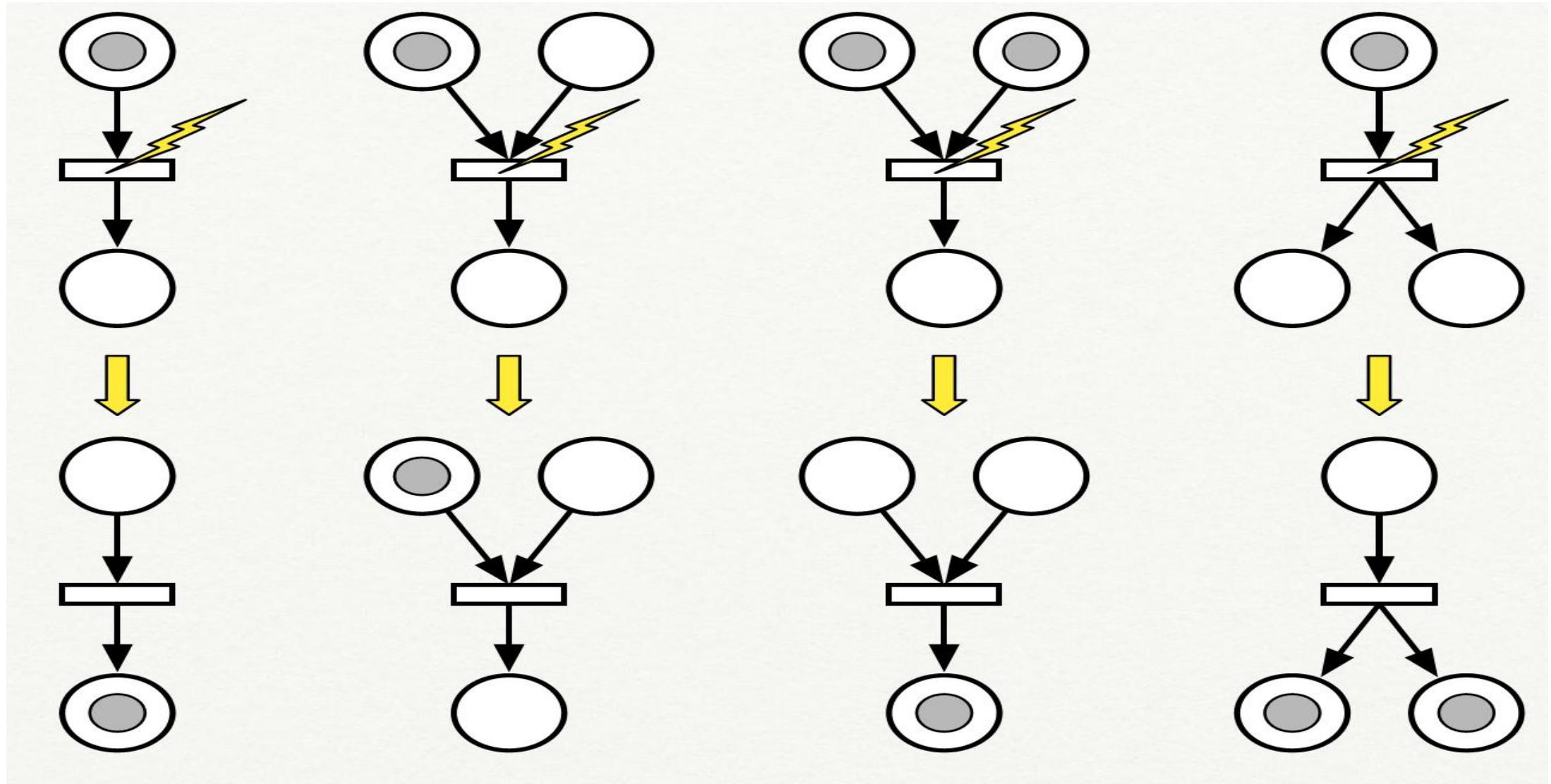
Transitions

- Synchronisation due à la réception de messages,
- Conditions d'application des actions.

Protocole d'interaction : Réseau de Pétri

- Les marques évoluent de place en place en suivant ce que l'on appelle des règles de franchissement.
- On dit qu'une transition T peut être franchie, ou encore qu'une transition est valide, si toutes ses places d'entrées P pour lesquelles il existe un arc orienté de P vers T possèdent une marque.
- Parmi toutes les places validées on en prend une au hasard qui sera effectivement déclenchée, et on dira que les marques franchissent cette transition.
- Franchir une transition revient à enlever une marque sur chaque place d'entrée et à en ajouter une sur chaque place de sortie (les places P pour lesquelles il existe un arc de T vers P).

Protocole d'interaction : Réseau de Pétri

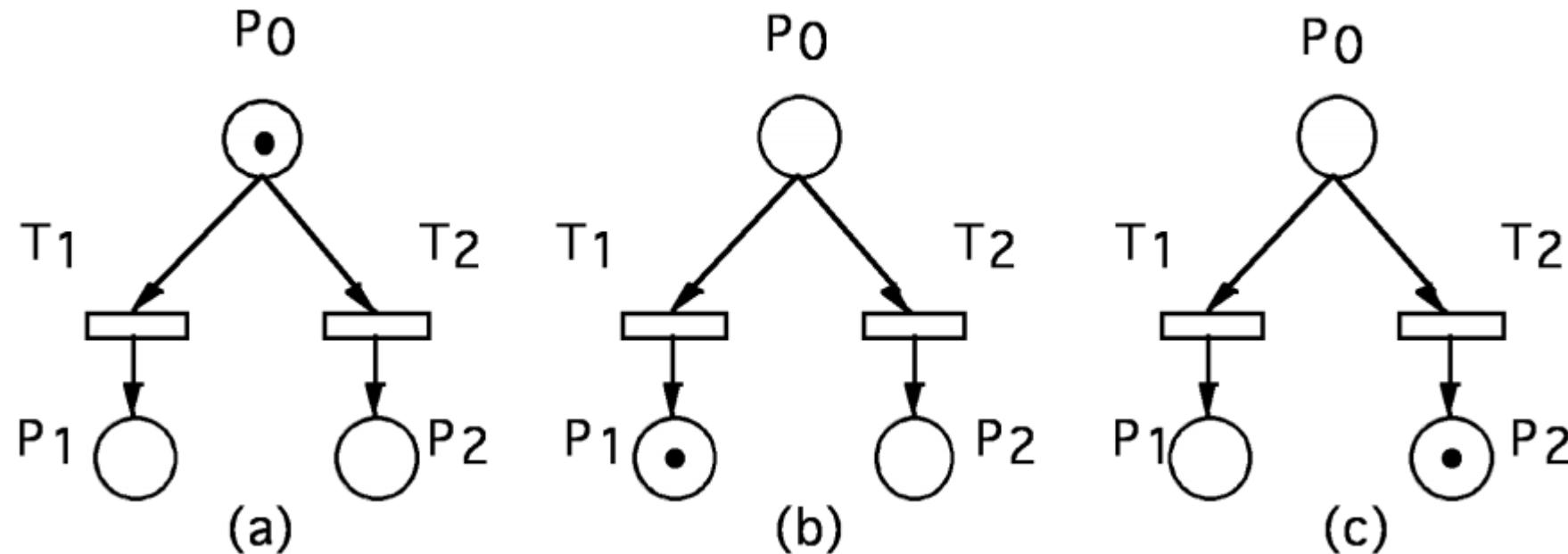


Protocole d'interaction : Réseau de Pétri

- Lorsqu'il y a plusieurs transitions possibles en sortie d'une place, la règle de tirage au sort permet de déterminer laquelle des transitions doit être franchie, comme le montre la figure suivante.
- On appelle *marquage* une distribution de jetons sur les places, et marquage initial la distribution initiale de jetons dans le réseau.

Protocole d'interaction : Réseau de Pétri

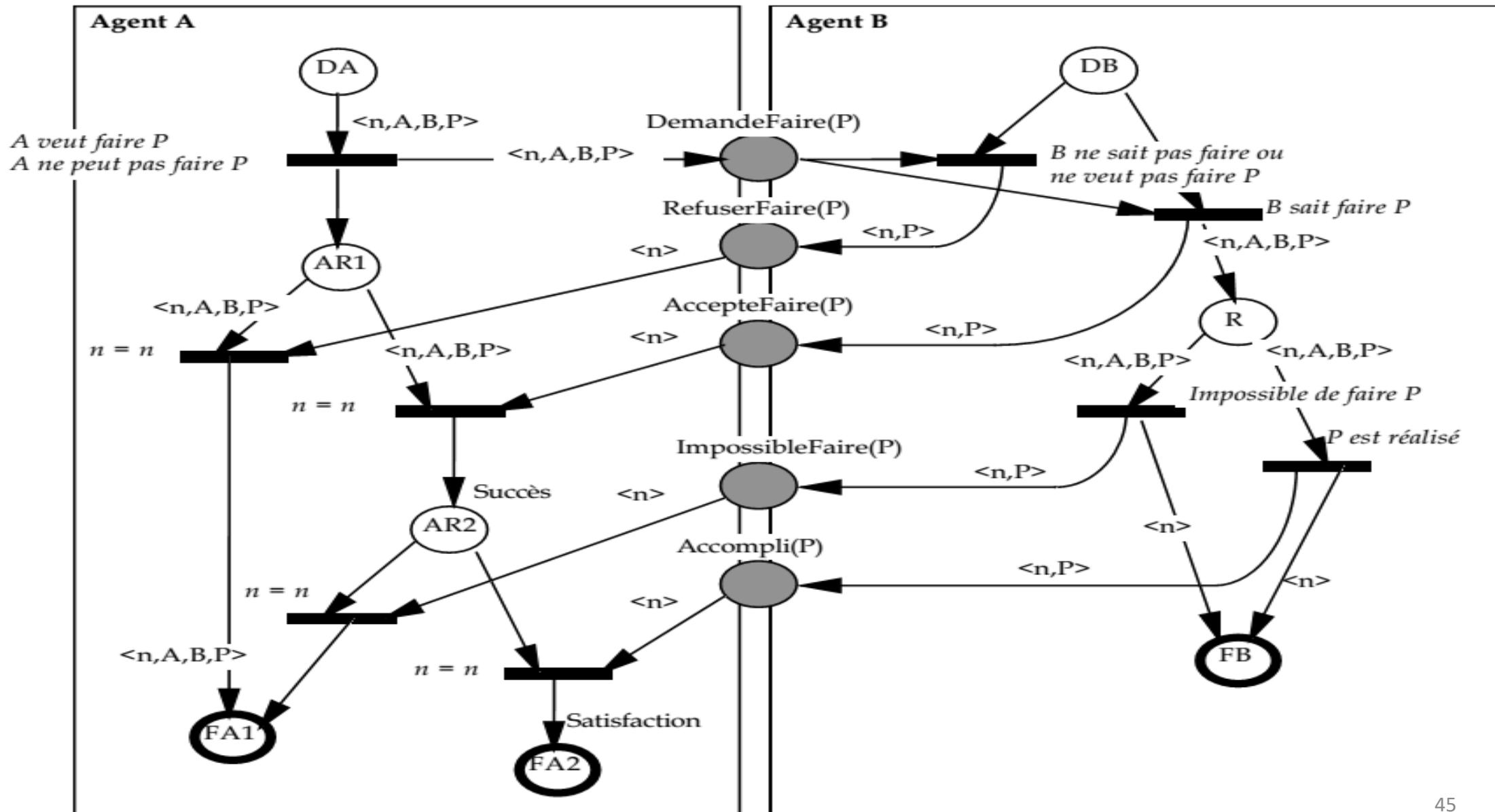
Les transitions $T1$ et $T2$ étant validées en (a), le tirage au sort des transitions à franchir produit le marquage (b) ou (c)



Protocole d'interaction : Conversations et réseaux de Pétri

- Les automates à états finis sont très pratiques pour préciser la structure des conversations lorsqu'elles apparaissent de manière isolée, c'est-à-dire lorsqu'on peut ramener une conversation à un seul processus.
- Cependant, les agents sont parfois engagés dans plusieurs conversations à la fois et il leur faut gérer ces conversations multiples.
- Il est alors plus facile de décrire la nature de ces interactions en utilisant les réseaux de Pétri.

Protocole d'interaction : Réseau de Pétri : Exemple



Protocole d'interaction : Réseau de Pétri : Exemple

- Un réseau de Pétri correspondant au traitement d'un exercitif.
- Chaque agent est décrit par un sous-réseau de Pétri, dont les places correspondent aux états internes de l'agent (ou de la tâche en cours, comme nous le verrons par la suite).
- Les transitions correspondent soit à des synchronisations dues à la réception de messages, soit à des conditions d'applications des actions, telles que les conditions essentielles ou préparatoires des actes de langage associés.
- Les messages en cours d'acheminement sont représentés par des places supplémentaires qui raccordent les agents A et B de manière à ne plus former qu'un seul réseau.
- Les places DA et DB décrivent les états initiaux dans lesquels se trouvent les agents avant le début de la conversation et les places $FA1$, $FA2$ et FB représentent les états de fin de conversation.

Protocole d'interaction : Réseau de Pétri : Exemple

- A partir de l'état DA , l'agent A envoie une requête à B et passe dans l'état ARI qui représente une attente de réponse.
- Si B ne sait pas faire P , il renvoie un refus à A , qui passe alors dans l'état $FA1$, ce qui indique que A doit chercher ailleurs la réalisation de P .
- Si B peut faire P , il envoie un message d'acceptation à A , qui se place de nouveau en attente de réponse.
- Pendant ce temps, B se trouve dans l'état BR en essayant d'accomplir l'action P .
- S'il peut la faire, alors il envoie une notification de fin d'accomplissement, ce qui place l'agent A en état $FA2$, sinon il indique qu'il ne lui est pas possible de faire P , mais dans les deux cas il passe dans l'état terminal FB .

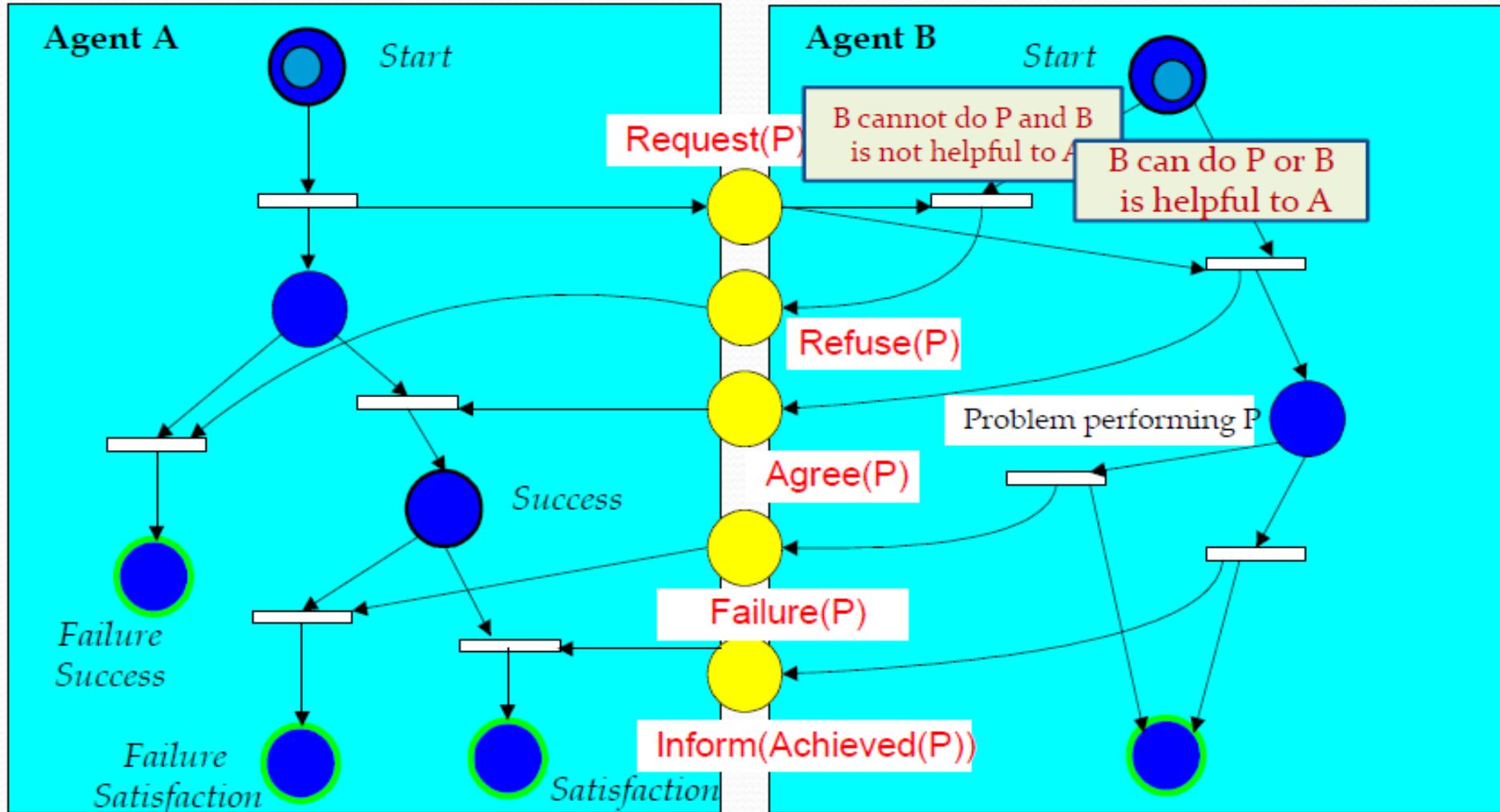
Protocole d'interaction : Réseau de Pétri : Exemple

- L'intérêt de cette approche est de pouvoir non seulement expliquer les échanges entre les agents, mais aussi de préciser les états internes dans lesquels ils se trouvent, et donc de décrire plus finement ce qui se passe au cours d'une conversation.

- On notera d'ailleurs que toutes les caractéristiques liées aux actes de langage se retrouvent ici. Par exemple les états de succès et de satisfaction de la demande initiale sont figurés respectivement aux places *AR2* et *FA2*.

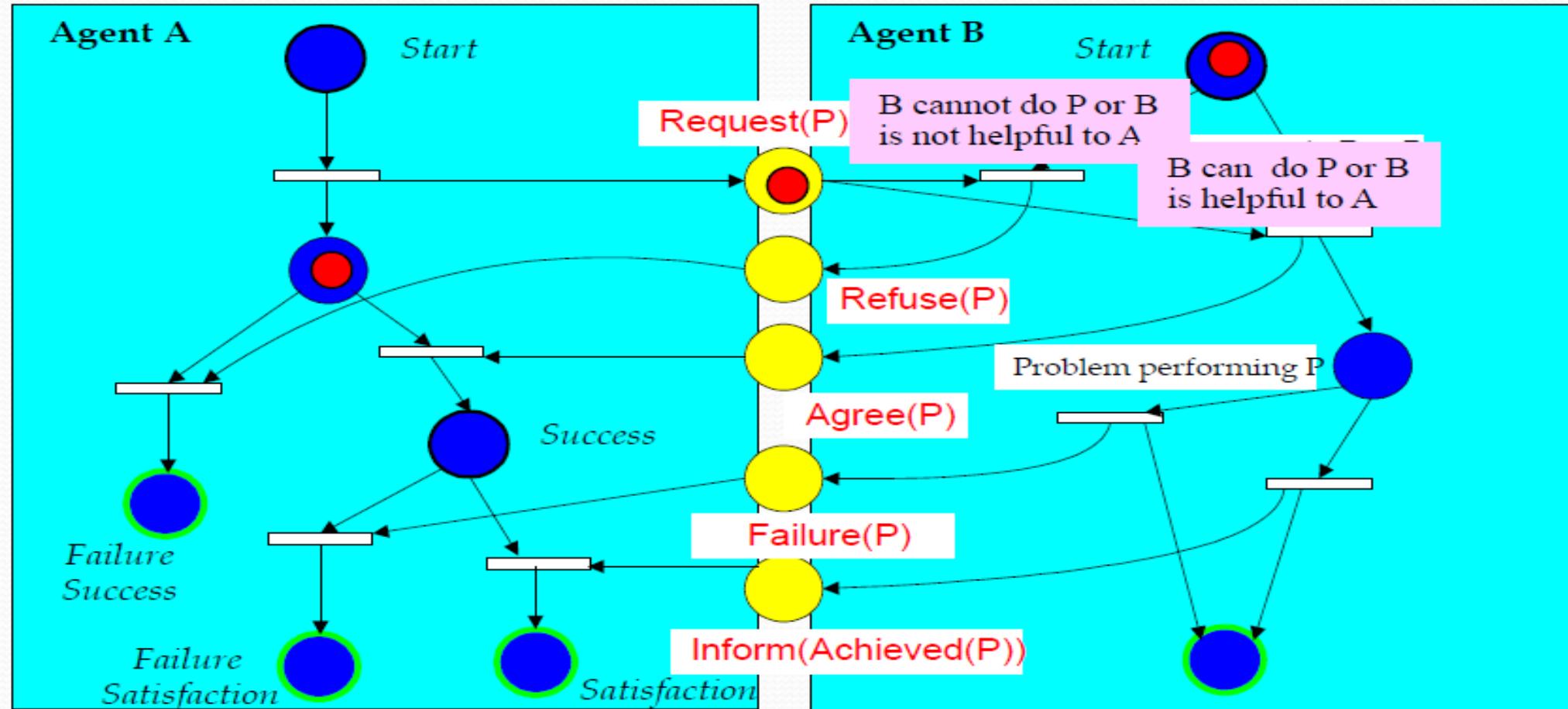
Réseaux de Petri

- Example: requesting to do something



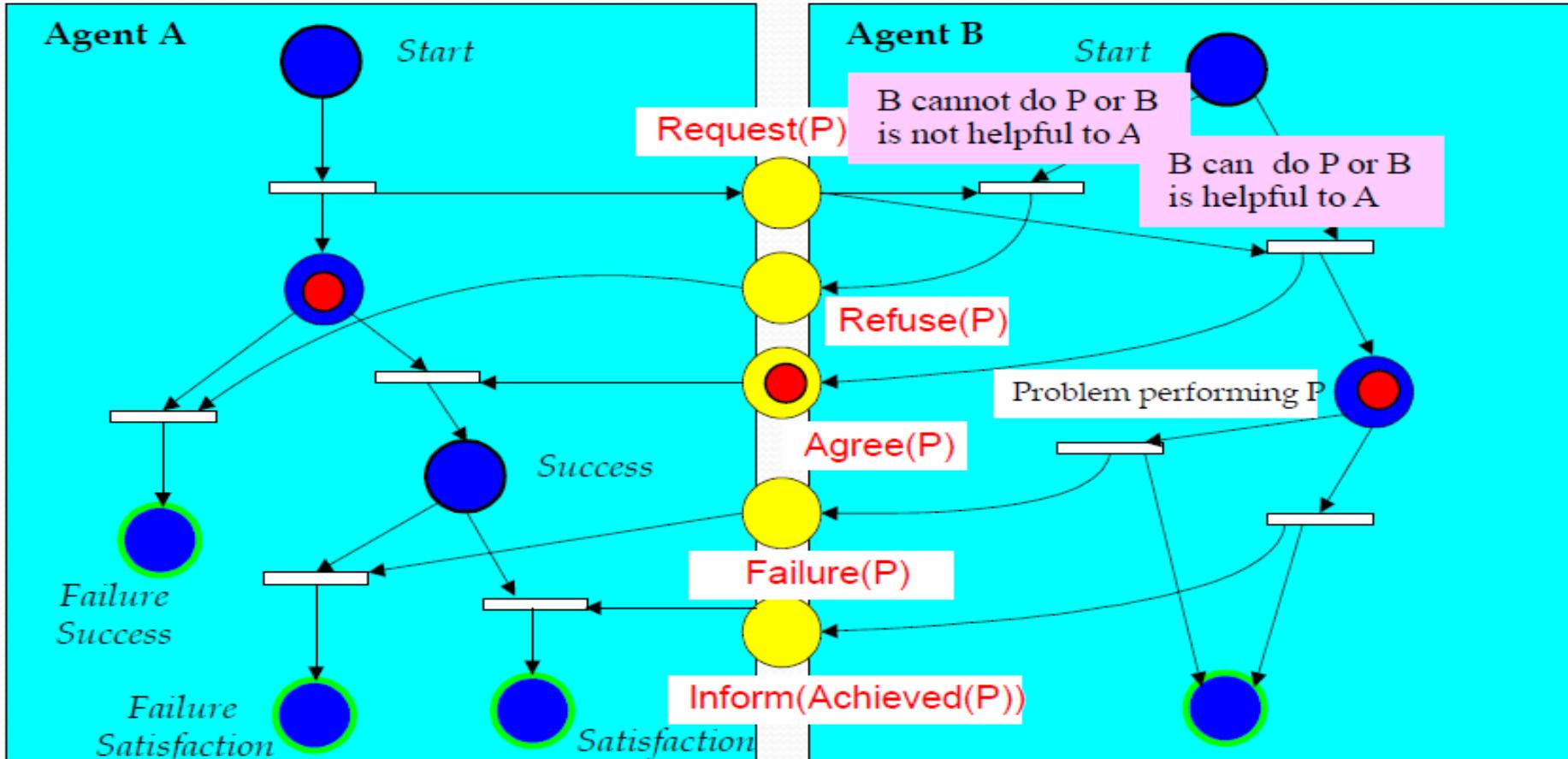
Réseaux de Petri

Example: requesting to do something



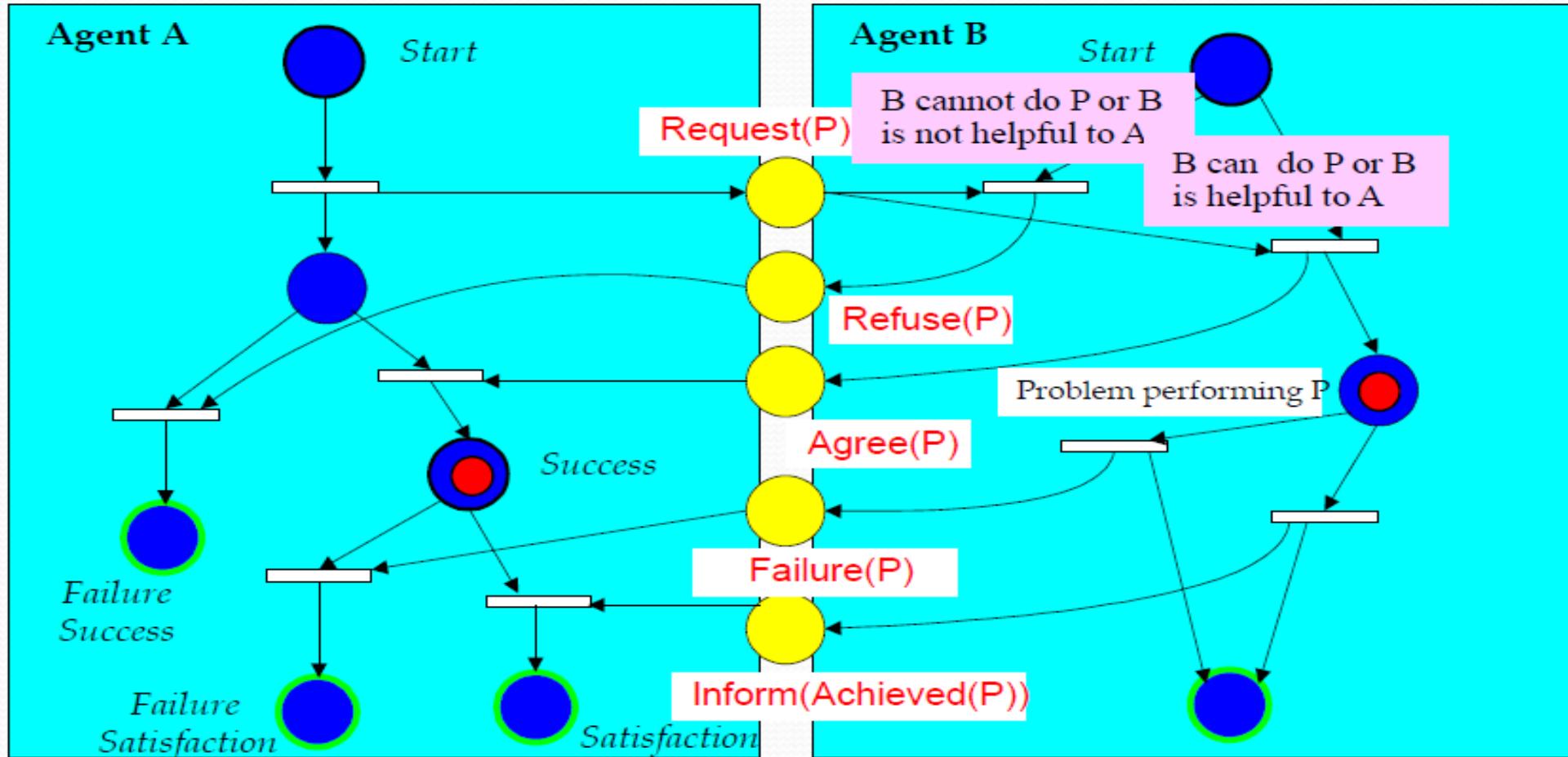
Réseaux de Petri

Example: requesting to do something



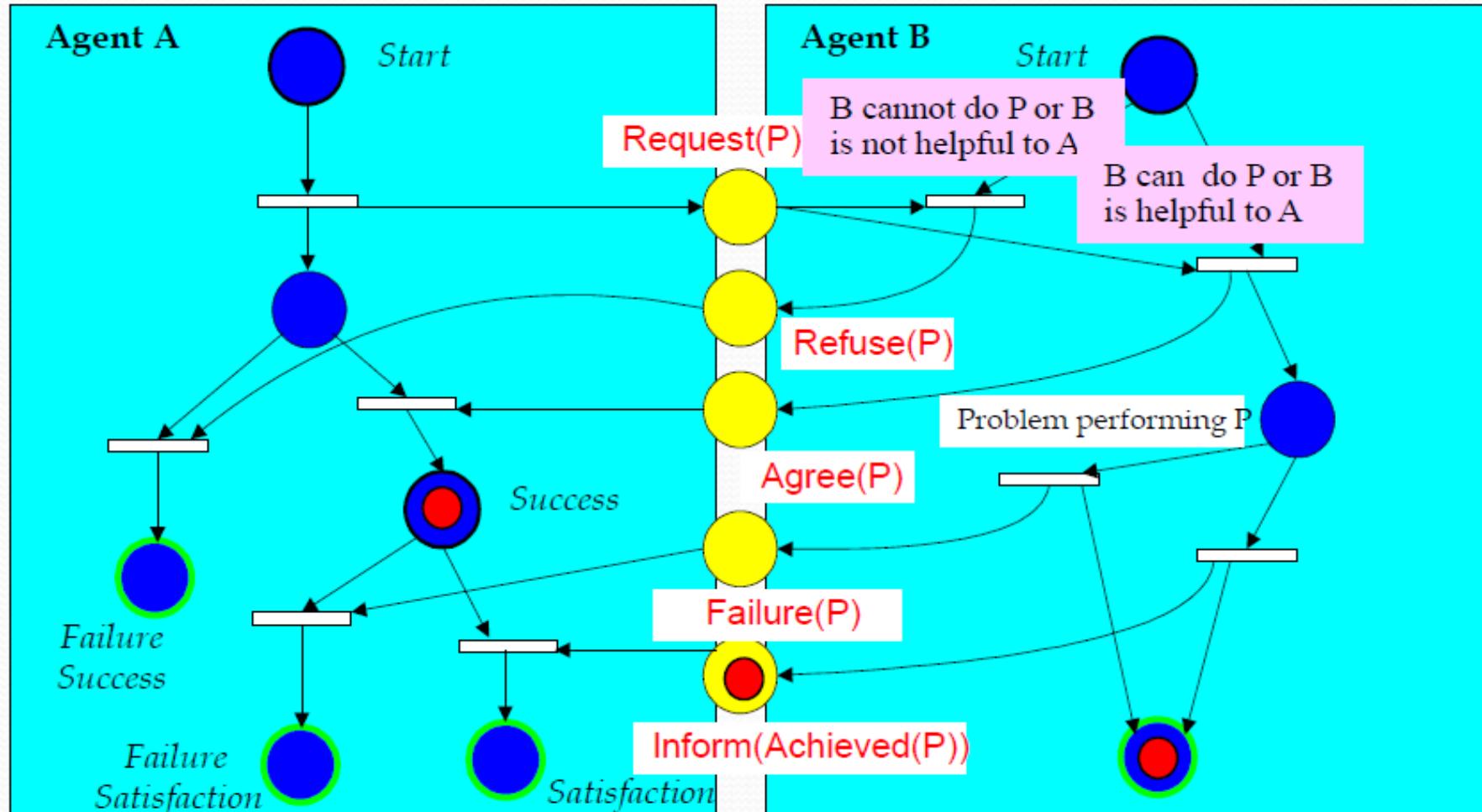
Réseaux de Petri

Example: requesting to do something



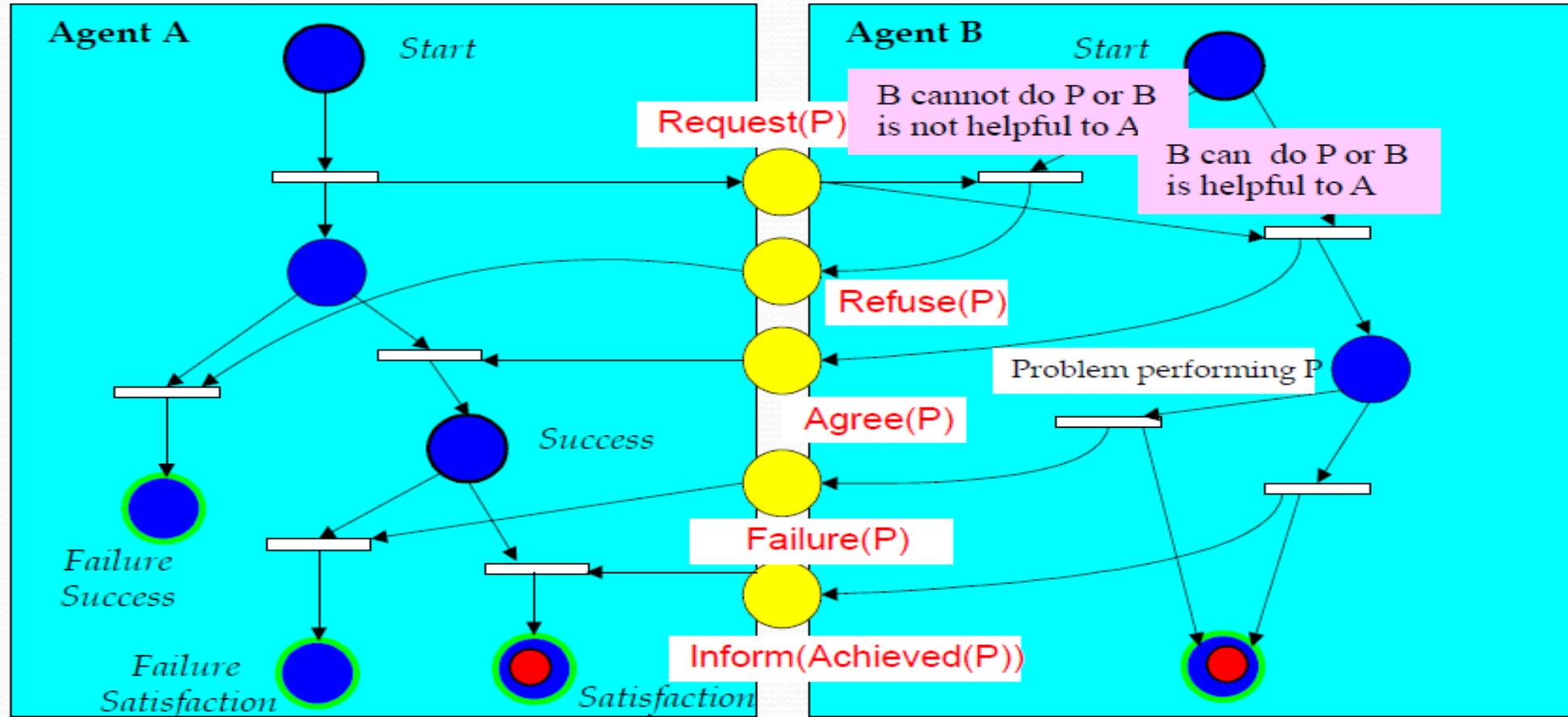
Réseaux de Petri

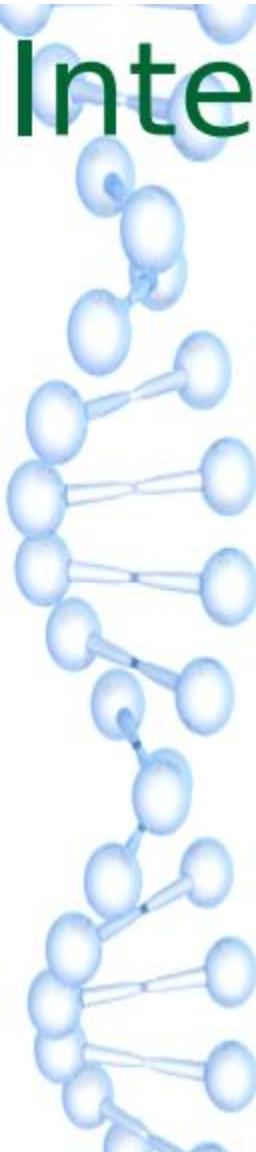
Example: requesting to do something



Réseaux de Petri

Example: requesting to do something



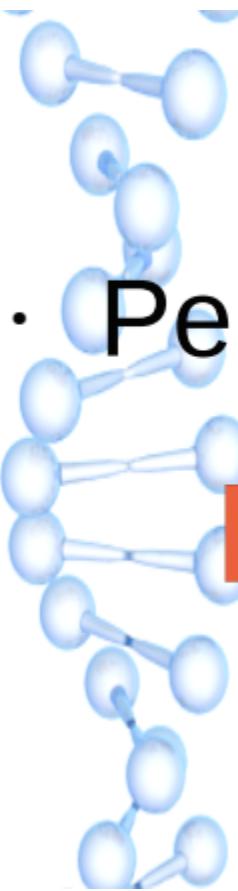


Interaction Formalisme AUML

- Sémantique : un protocole est un ensemble de messages échangés entre des rôles
- Notation
 - Une dimension verticale qui représente le temps.
 - Une dimension horizontale qui représente les différents rôles.

- Exemple :
 - Rôles : étudiant, enseignant
 - Agents : Anne, Bob, Charles
 - Anne → étudiant, Bob → étudiant, enseignant
- Fil d'interaction
 - à partir des rôles

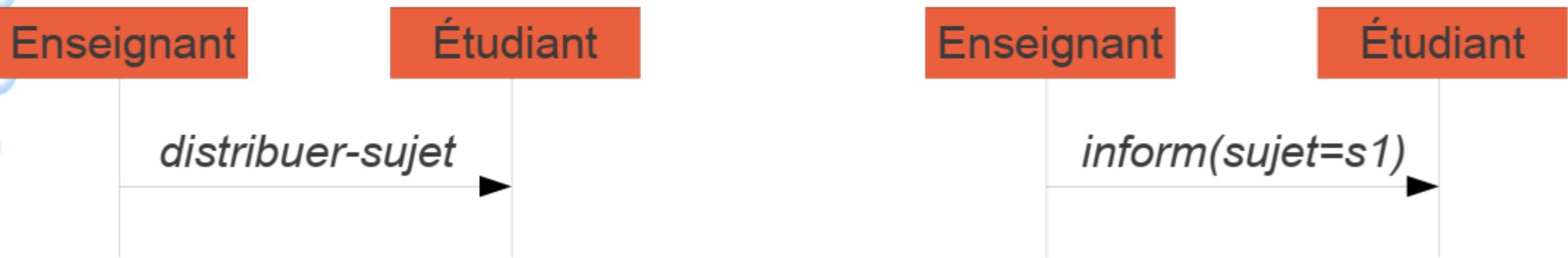


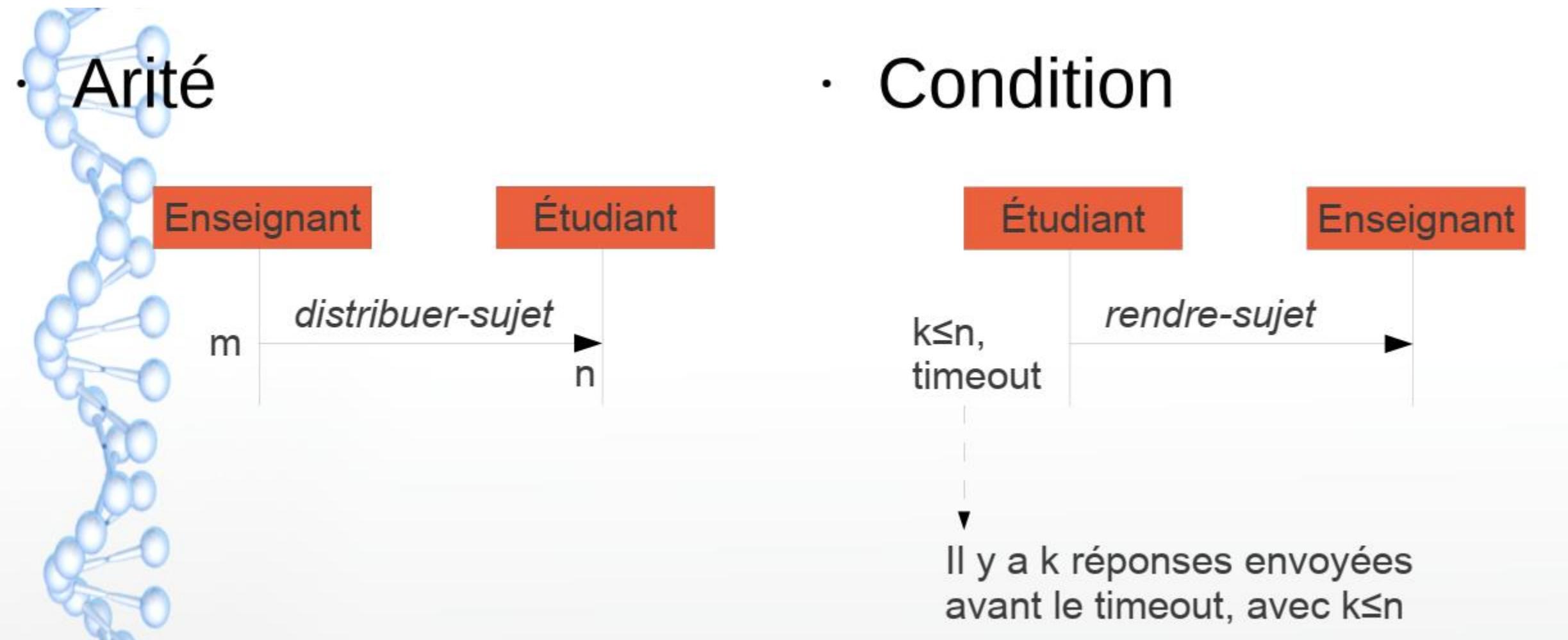


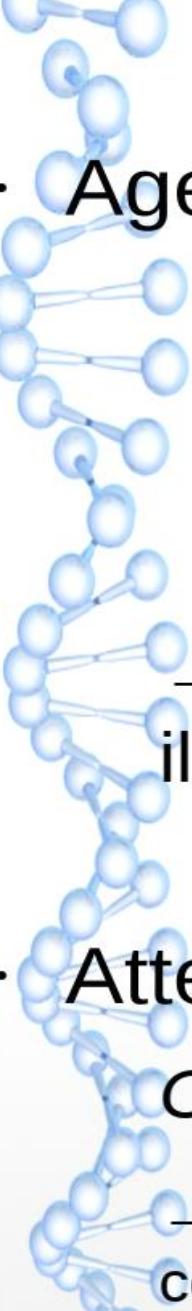
Protocole : message

- Performatif

- Performatif + Contenu







Protocole : messages

- Agencements



→ si enseignant envoie un « distribuer-sujet » alors il recevra un « rendre-sujet »

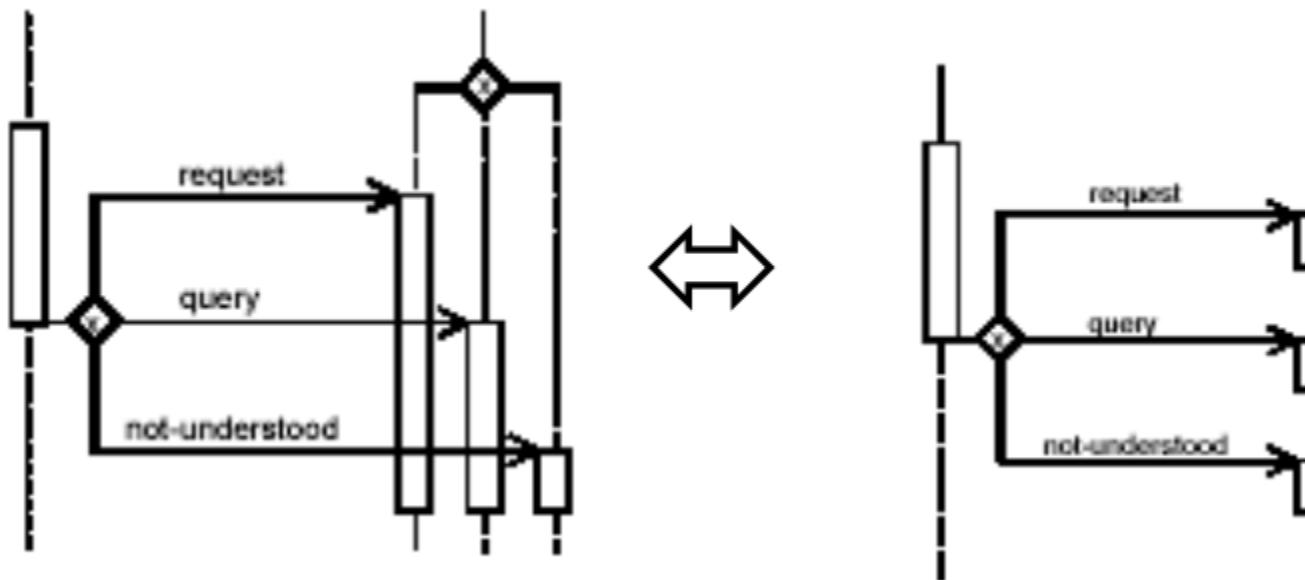
- Attention (rappel)

Ce sont des rôles, pas des agents !

→ Les agents qui prennent un rôle s'engagent à interagir conformément au protocole

- 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 with 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.

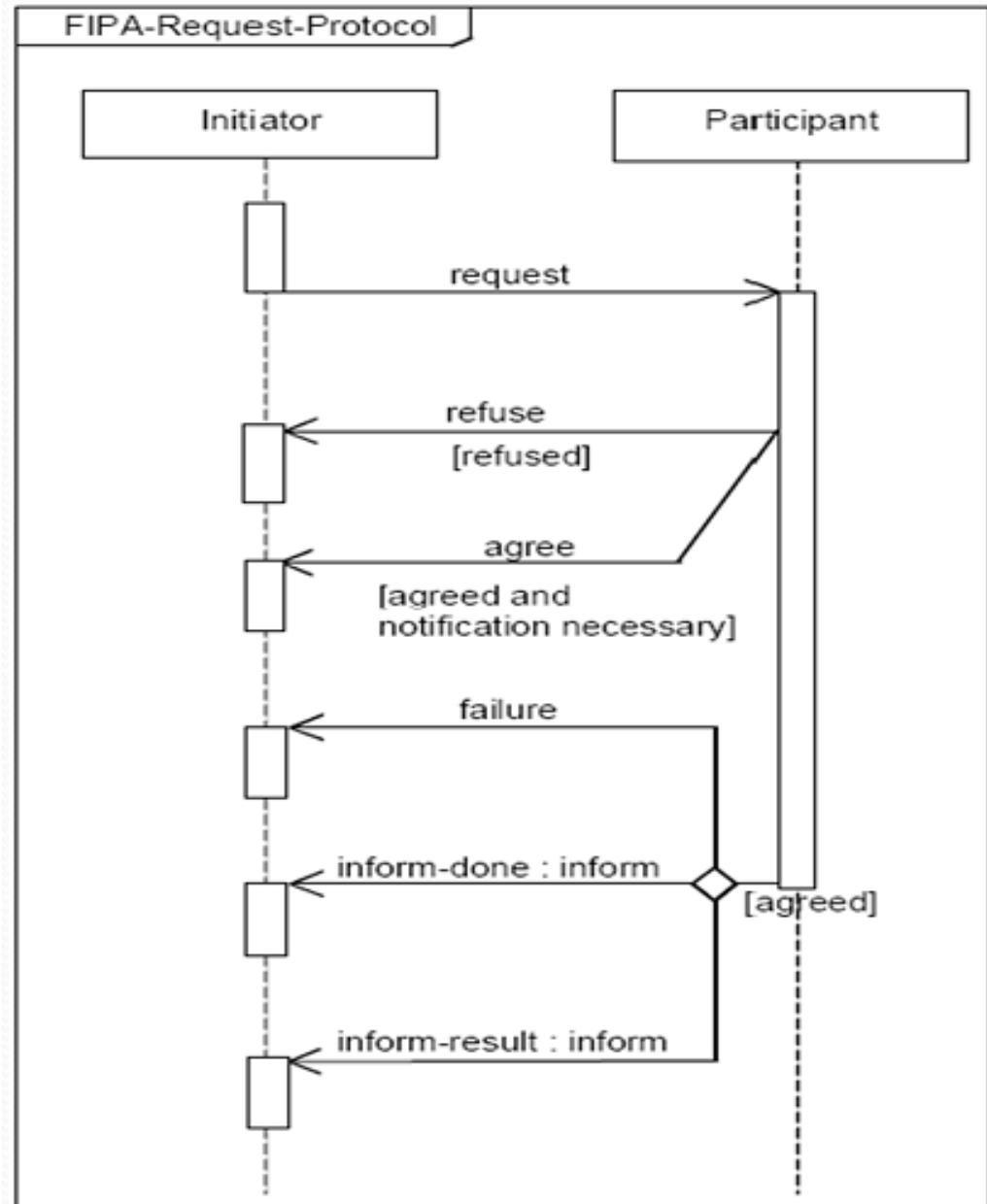
- A tall thin rectangle whose top is aligned with its initiation time and whose bottom is aligned with its completion time.
- Drawn over the lifeline of an agent role
- Represents a task being performed
 - May be labeled as text next to the thread or in the left margin
 - The incoming message may indicate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



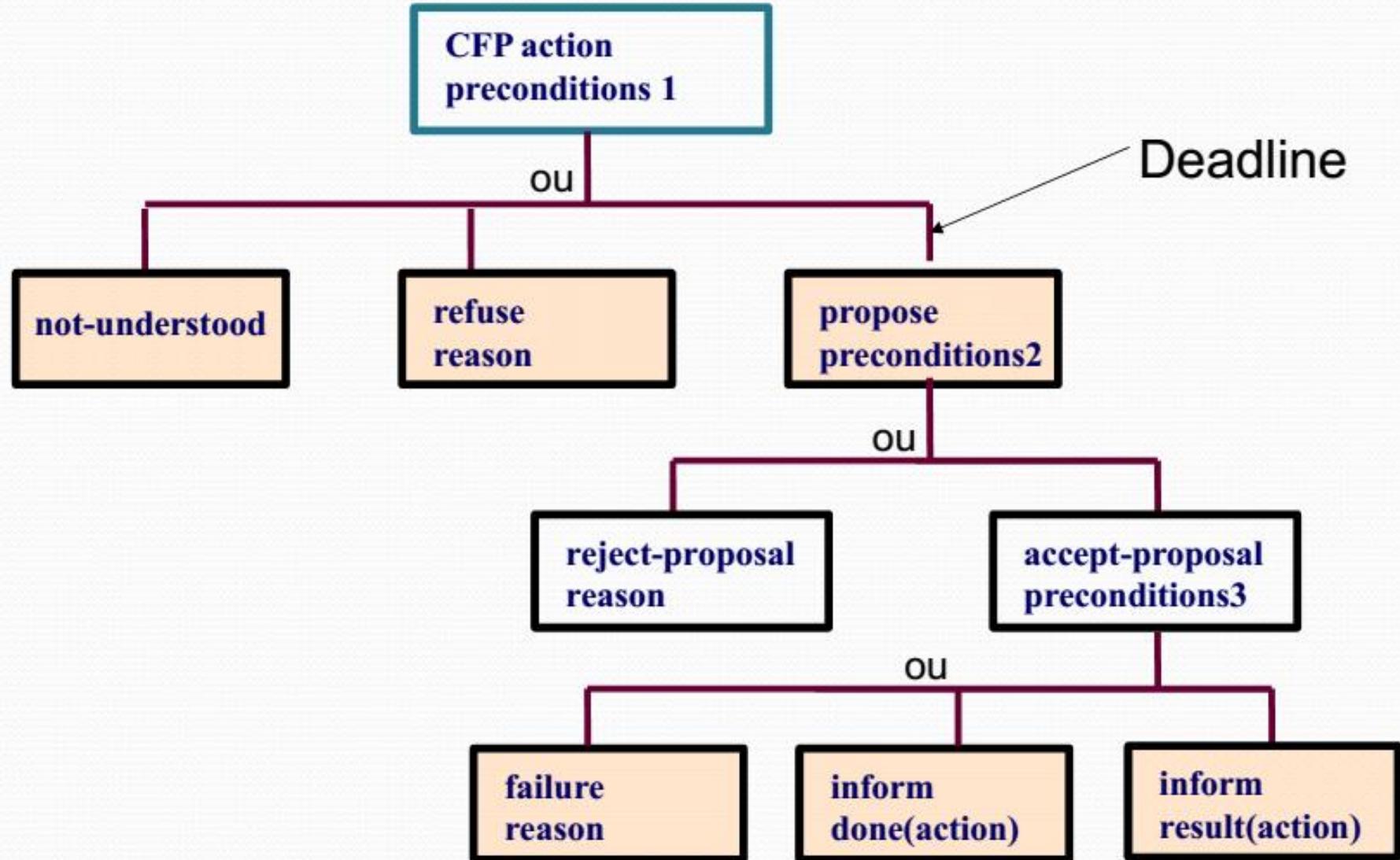
Exemples de protocoles *ACL-FIPA*

Request

- Identifié par *fipa-request* dans le champ protocole 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*



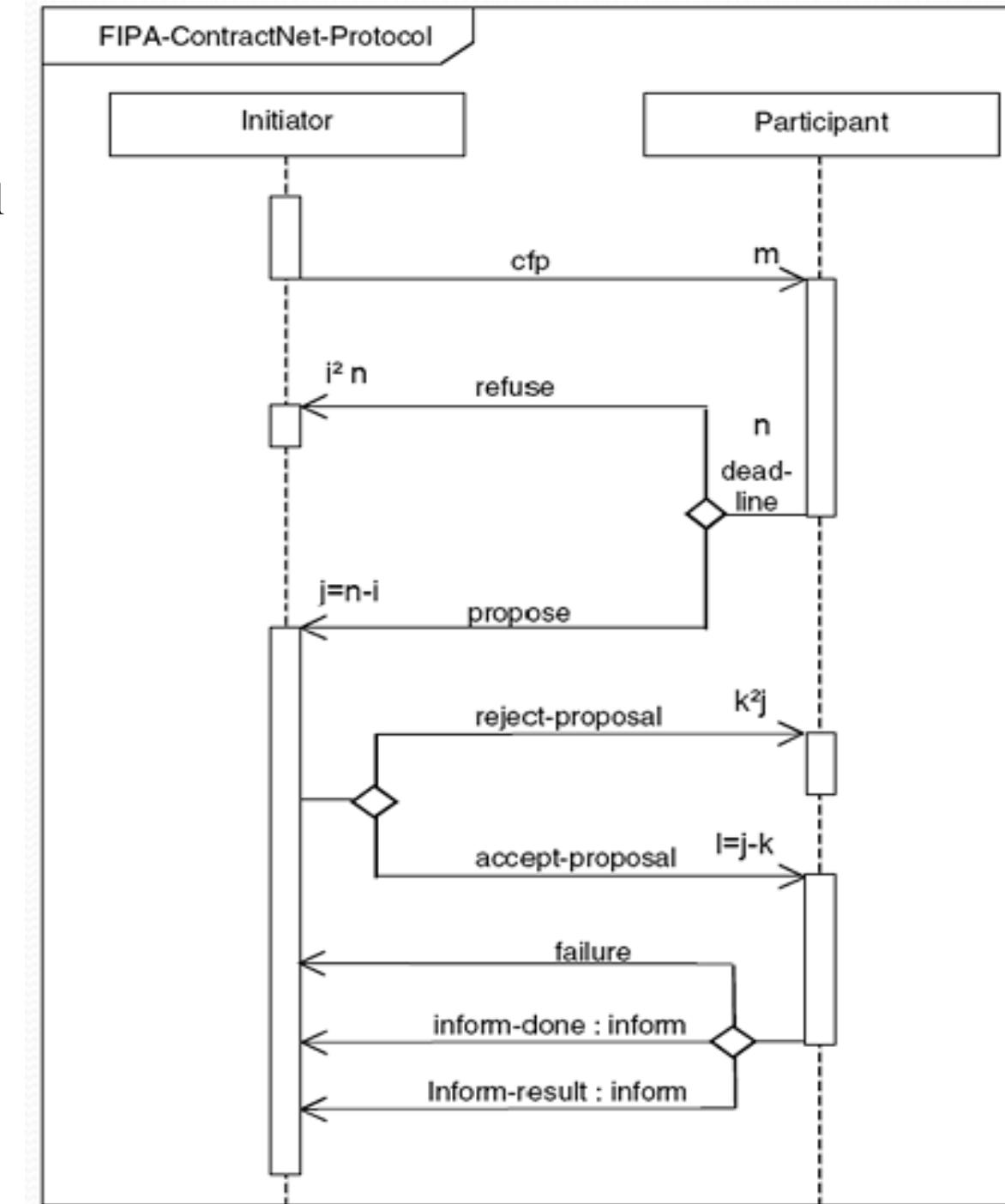
Contract-net



CFP : call for proposal (un appel à proposition)

Contract-Net (réseau contractuel)

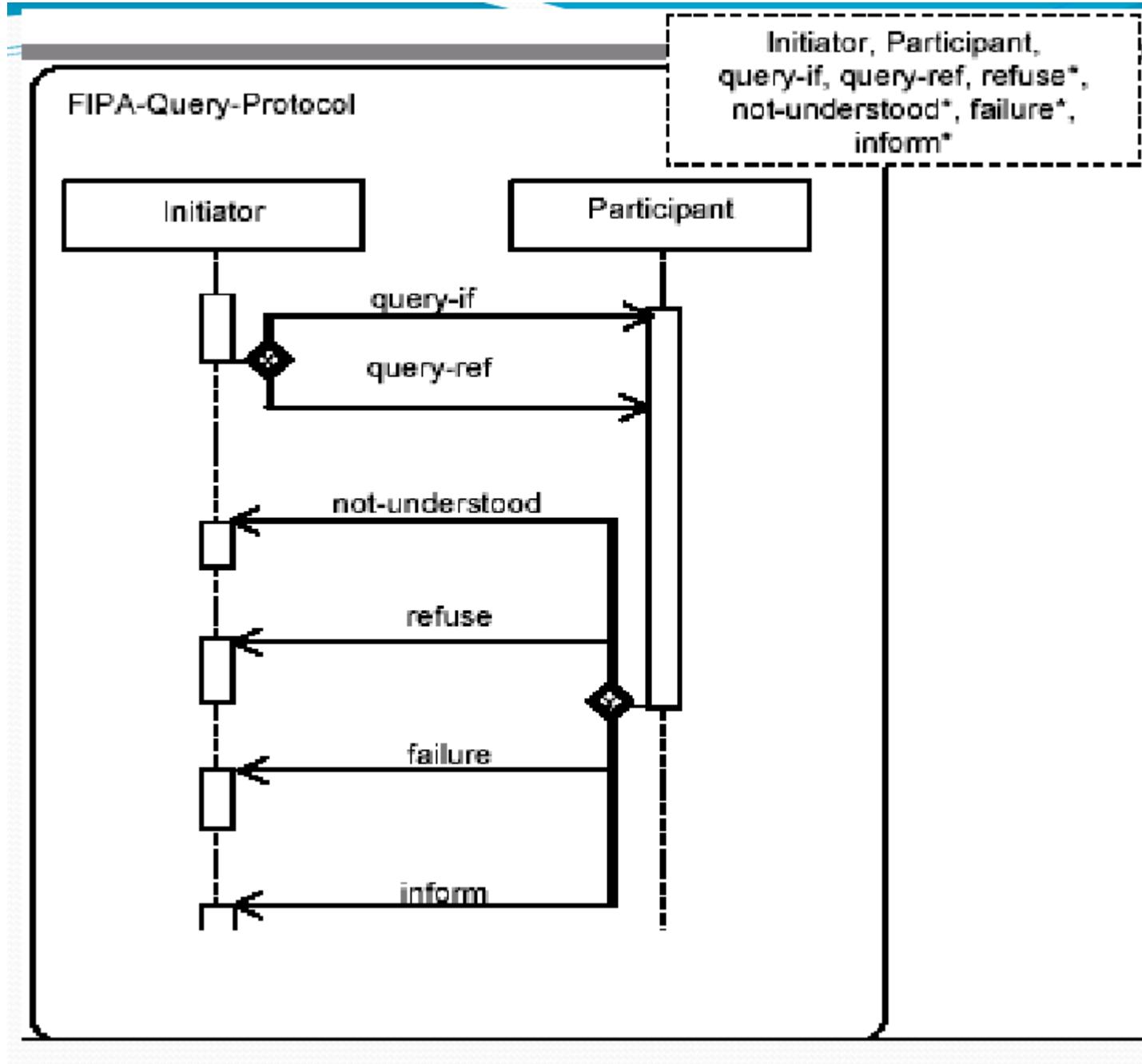
- Identifié par *fipa-contract-net* dans le champ protocol d'un message *ACL*
- A tout point du protocole un message de performatif *not-understood* peut être envoyé
- A tout point du protocole un protocole d'annulation peut être déroulé *fipa-cancel-meta-protocol*



Contract-net

- Le protocole s'appuie sur la formalisation des interactions
- Les agents peuvent jouer deux rôles, manager (المُسْؤُلُ عَنِ الْمَزَادِ) ou contractor (المُشَارِكُ فِي الْمَزَادِ) :
- le protocole est initialisé par le manager, qui envoie un appel à proposition (en anglais call for proposal) ;
- les contractors génèrent des propositions (en anglais proposals) indiquant tous les éléments pertinents pour que le manager fasse son choix et le lui envoient ;
- le manager choisit la proposition qu'il préfère et envoie une notification d'acceptation (accept) au contractor ayant emporté l'enchère (المزاد). Il envoie aussi une notification de rejet (reject) aux autres contractors ;
- enfin, une fois la tâche exécutée, le contractor en informe le manager. Si le contrat comprenait un résultat à communiquer, il lui envoie également ce résultat (inform). Si le contrat a échoué, il l'en informe (cancel).

Query



Request

