

# Un modèle d'interaction pour la négociation

Marie-Hélène Verrons

Equipe SMAC - LIFL  
Université de Lille 1

Présentation du 24 septembre 2004  
à la réunion MFI

# Plan de la présentation

- Présentation de l'équipe
- Contexte et objectifs
- La négociation
- GeNCA : un modèle en trois couches
- Applications
- Conclusions
- Perspectives

# Présentation de l'équipe

- L'équipe SMAC du LIFL est dirigée par M. le Professeur Philippe Mathieu et comporte 10 personnes.
- Quatre thèmes de recherche :
  - Étude des architectures systèmes multi-agents (SMA)
  - Étude de la modélisation du comportement
  - Étude des mécanismes de négociation
  - Agents situés et interactions entre agents

# Contexte et Objectifs

## Contexte :

- Négociation entre agents informatiques
- Interactions formelles entre agents
- Génie logiciel

## Objectifs :

- Proposer un protocole générique de négociation : un noyau pur, PGCD des applications de négociation
- Réaliser une API polyvalente, qui sera réutilisée dans différentes applications de négociation.

# Définitions

- La négociation : C'est un processus qui permet de faire évoluer la satisfaction de chacun.
- Un contrat : C'est la proposition d'une ou plusieurs ressources par un initiateur .
- Une référence : le Contract Net Protocol.  
Proposé par Smith en 1980, il est basé sur les lois qui régissent le commerce : l'offre et la demande. Un agent nommé “manager” veut déléguer une tâche à un “contracteur”.

# Différentes négociations

- Les enchères
  - Les systèmes de vote
  - Le *Take it or leave it offer*
  - Les négociations multi-attributs
- 
- Les négociations multi-étapes
  - Les négociations combinées
  - Les négociations par argumentation

# Constat

- Toutes les négociations sont sensiblement identiques.
- Formellement, on a des ressources  $r_i$  et des participants  $p_j$ .
- Un initiateur propose un contrat sur un ensemble de ressources à un ensemble de participants et une discussion s'ensuit afin d'arriver à une solution acceptable pour tout le monde.
  - => Réification de la notion de ressource et de la notion de contrat
  - => Ce qui change, c'est la stratégie et le moyen de communiquer

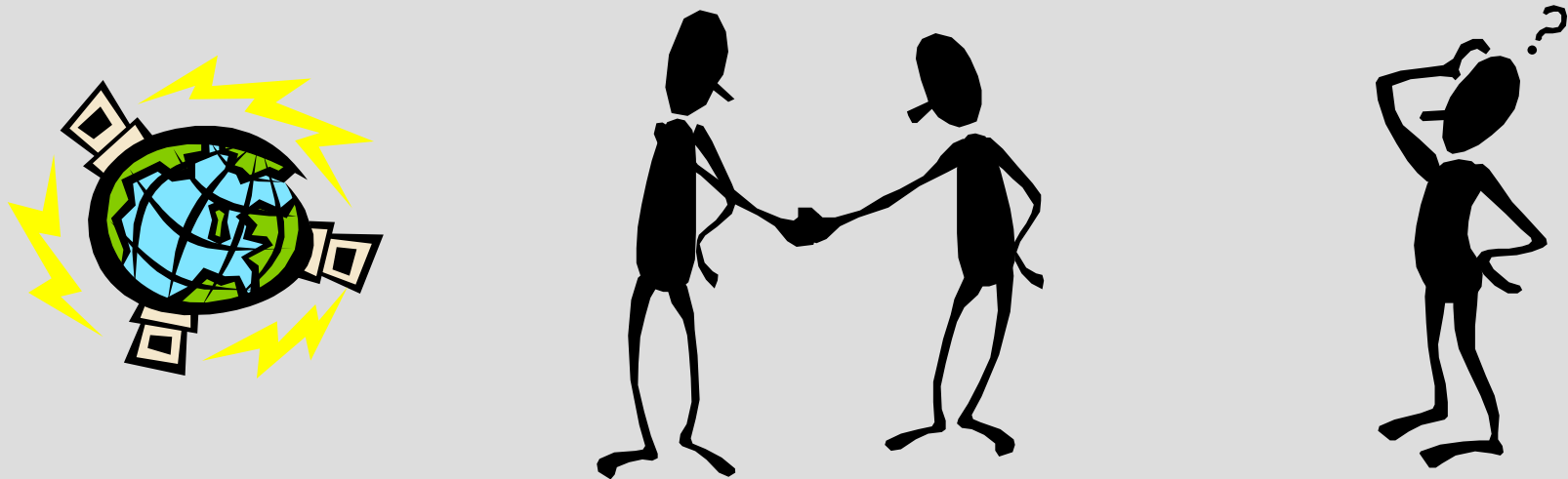
# Propriétés de négociation

- cardinalité : 1 vers 1, 1 vers n, n vers m
- type de ressources (individuelles ou communes)
- synchrone/asynchrone
- simultanéité des négociations
- nombre minimal d'accords pour que le contrat soit confirmé
- nombre de tours de parole
- possibilité de se rétracter
- nombre de renégociations



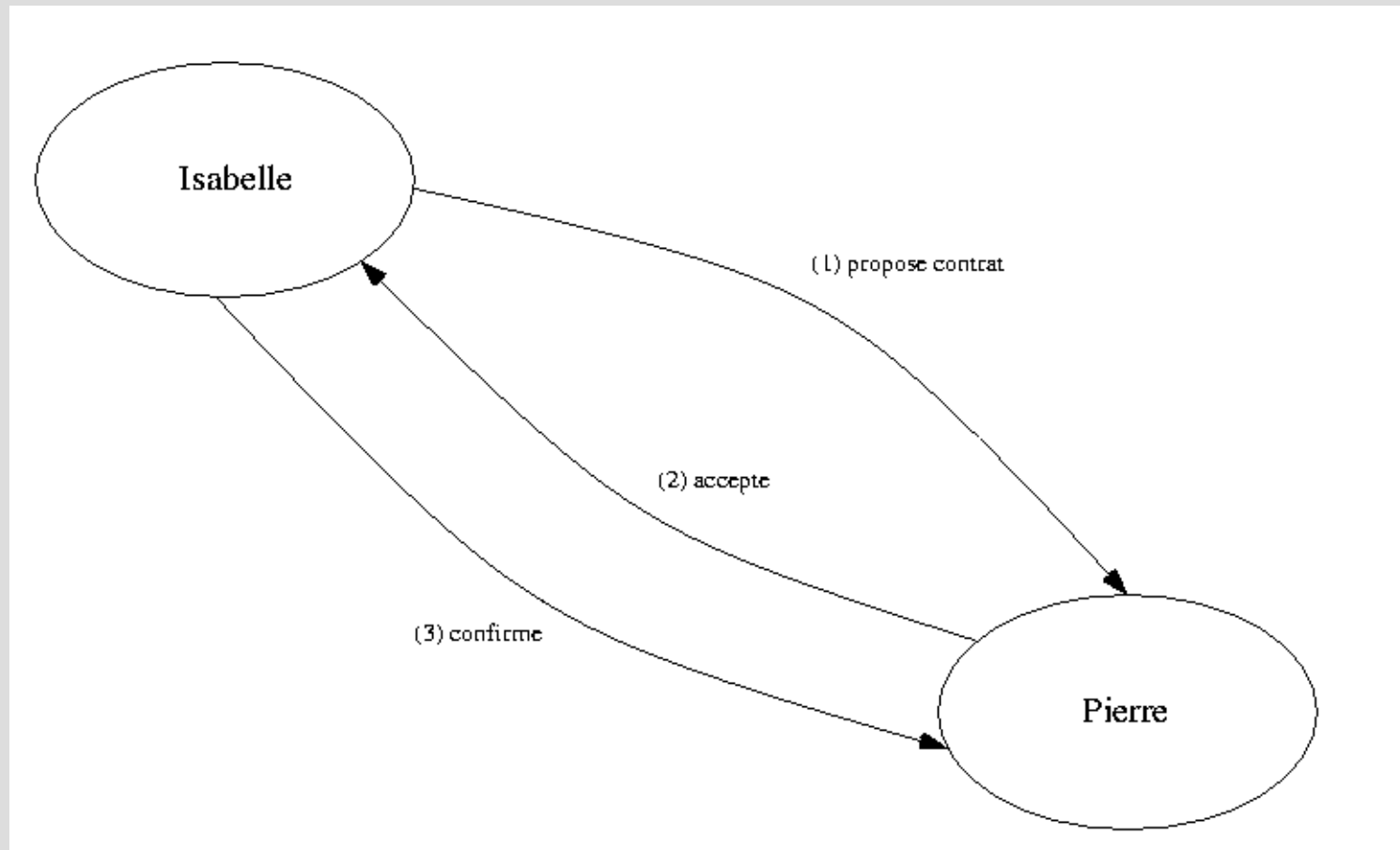
# GeNCA : un modèle en trois couches

- Nous soutenons que toute négociation automatique peut se décomposer en trois couches permettant de réaliser des négociations de contrats sous la forme la plus générique :

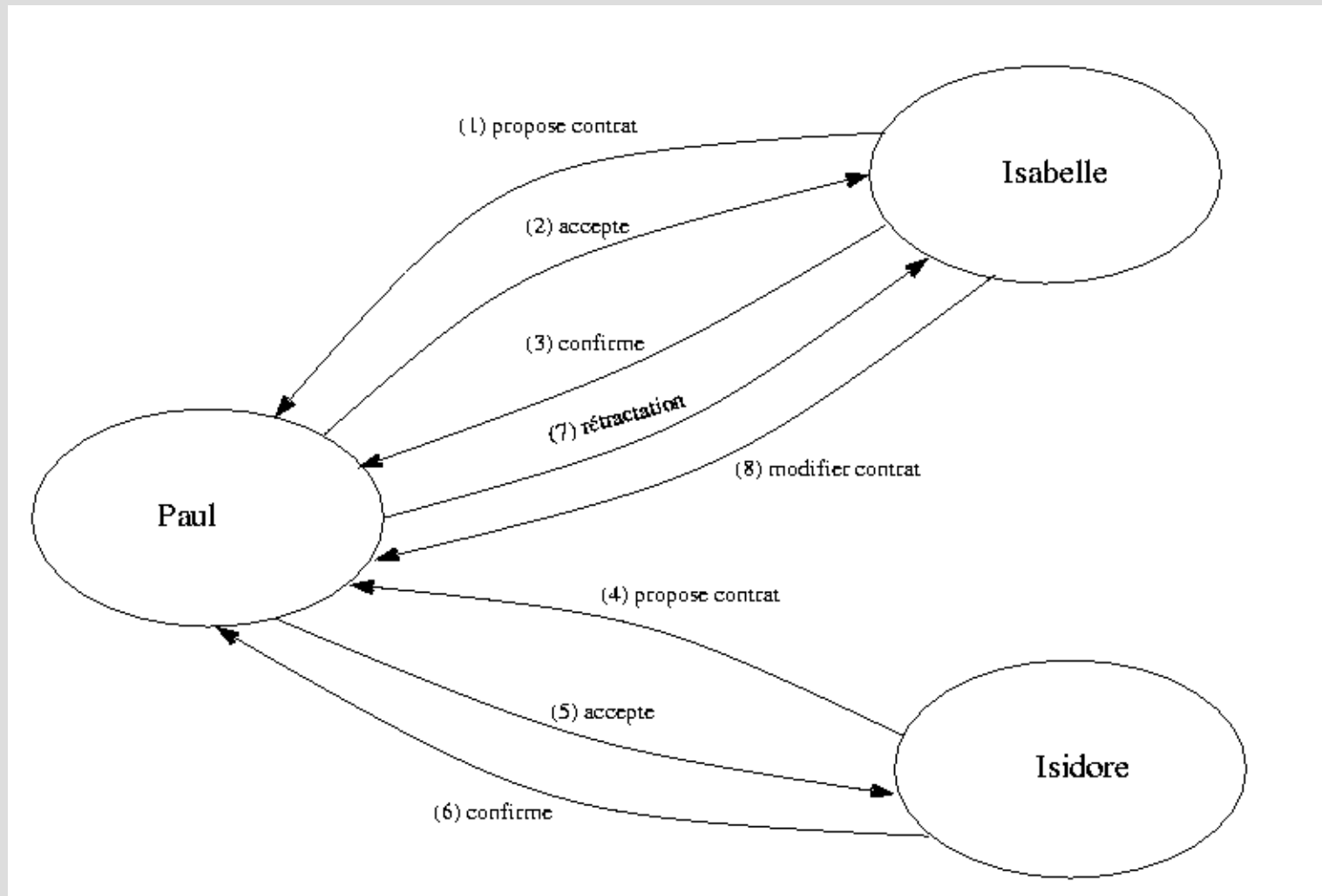


=> Création de l'API Java GeNCA

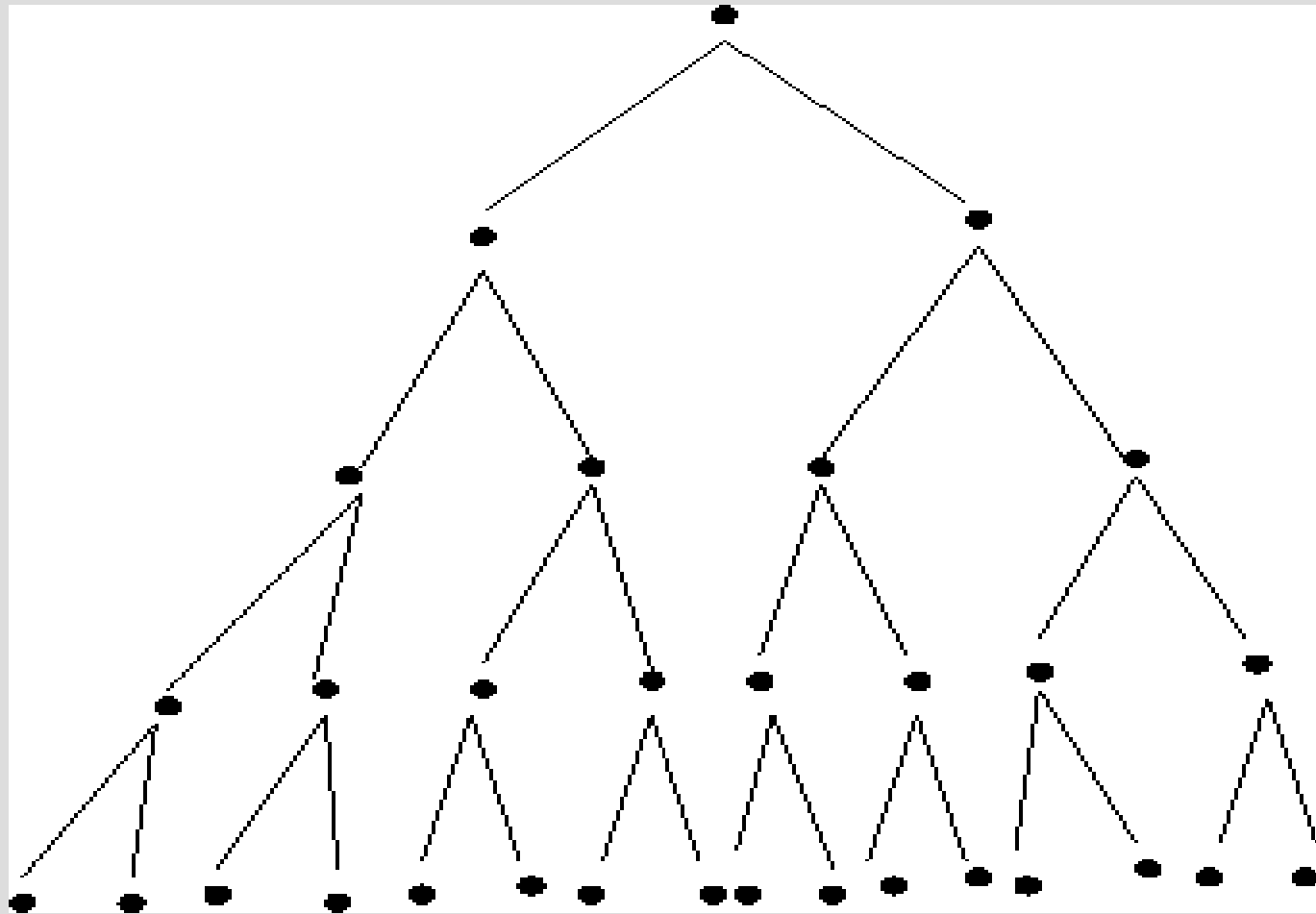
# Exemple simple de négociation



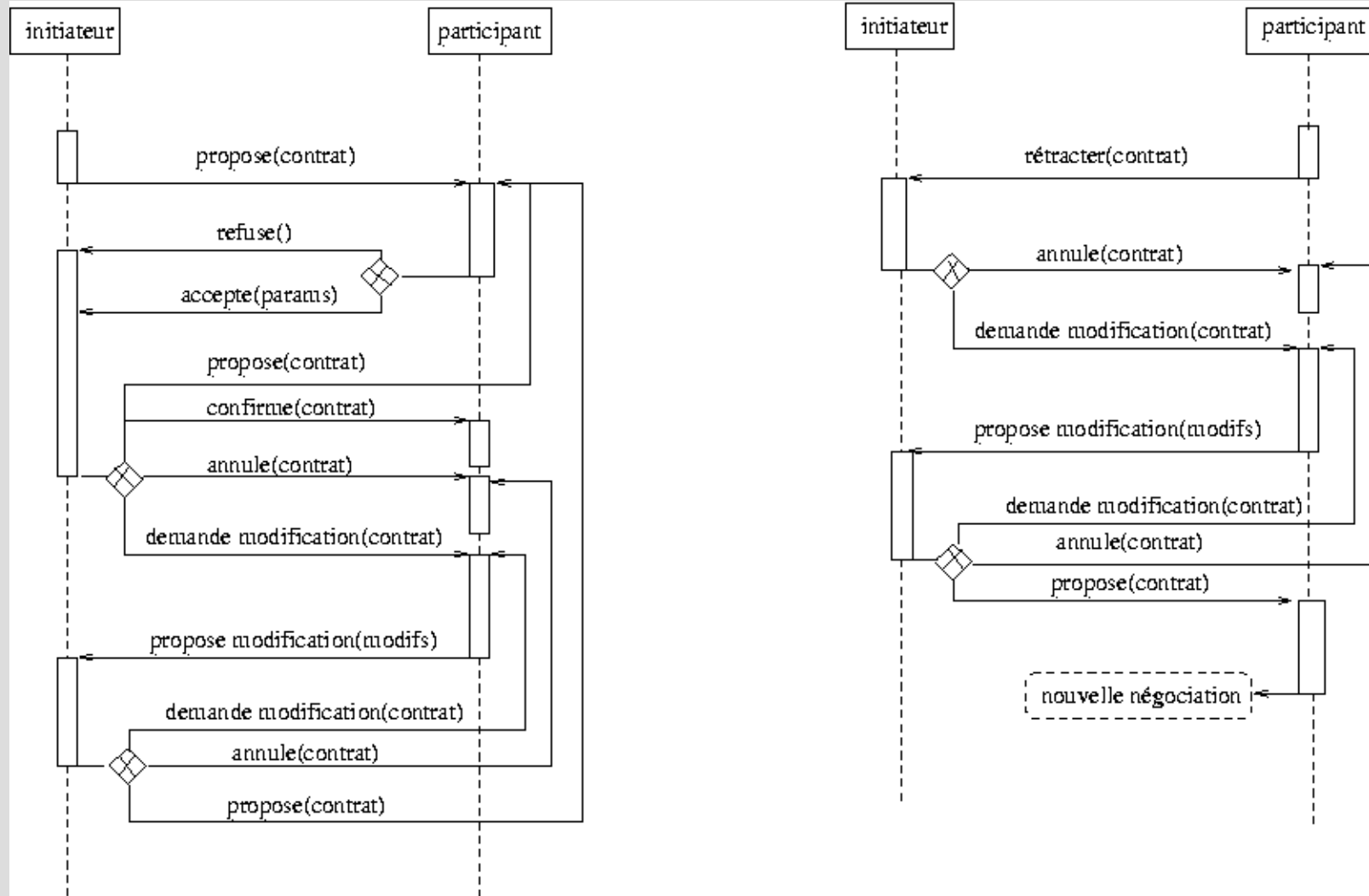
# Exemple avec renégociation



# Exemple plus complexe



# Le protocole de négociation



# Paramétrage du protocole

```
<!ELEMENT protocol (answer-delay,default-answer,  
minAgreements,nbRounds,nb-modifications-by-round,  
retraction-allowed,nbRenegotiations)>  
<!ELEMENT answer-delay (#PCDATA)>  
<!ELEMENT default-answer EMPTY>  
<!ATTLIST default-answer value (accept | refuse) "refuse">  
<!ELEMENT minAgreements (#PCDATA)>  
<!ELEMENT nbRounds (#PCDATA)>  
<!ELEMENT nb-modifications-by-round (#PCDATA)>  
<!ELEMENT retraction-allowed EMPTY>  
<!ATTLIST retraction-allowed value (true | false) "true">  
<!ELEMENT nbRenegotiations (#PCDATA)>
```

# Propriétés de notre protocole

- Paramétrage du protocole par fichier
- Renégociation automatique
- Cardinalité  $n$  vers  $m$
- Gestion des deadlocks
- Gestion des négociations conflictuelles soit en mode séquentiel, soit en mode parallèle
- Convient entre autres pour :
  - le *take it or leave it offer*
  - les enchères
  - les négociation multi-attributs
  - la prise de rendez-vous
  - la création d'emplois du temps
  - le choix d'un restaurant

# Modélisation des négociations

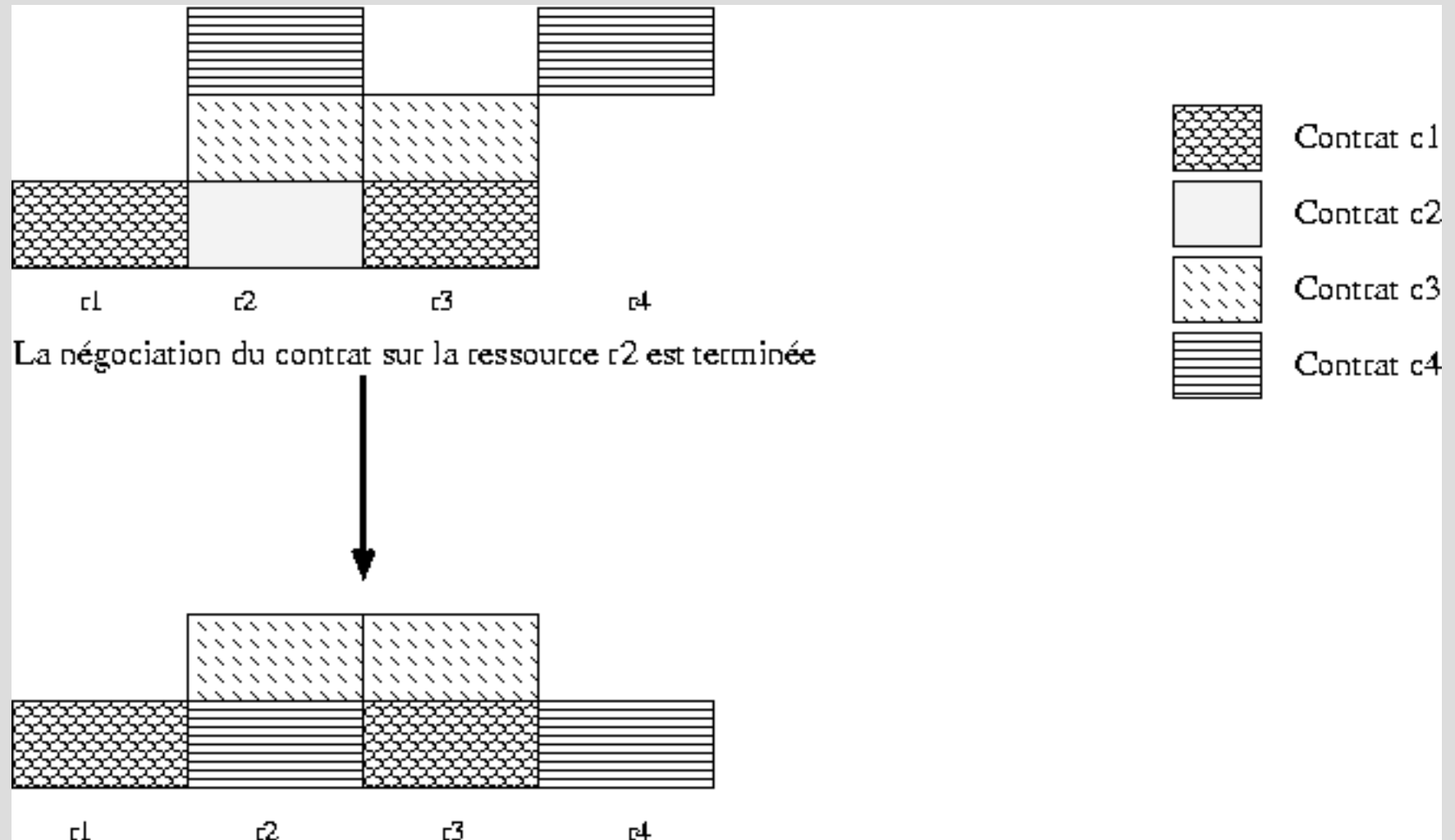
- Deux types de structures de données : celles pour la description de la négociation et celles pour la dynamique de la négociation.
- Description de la négociation :
  - ressources : ce qu'il est possible de négocier (espace de définition)
  - contrats : l'initiateur et les ressources à négocier
  - propriétés du contrat (connues de l'initiateur uniquement) : paramètres de la négociation



# Modélisation des négociations

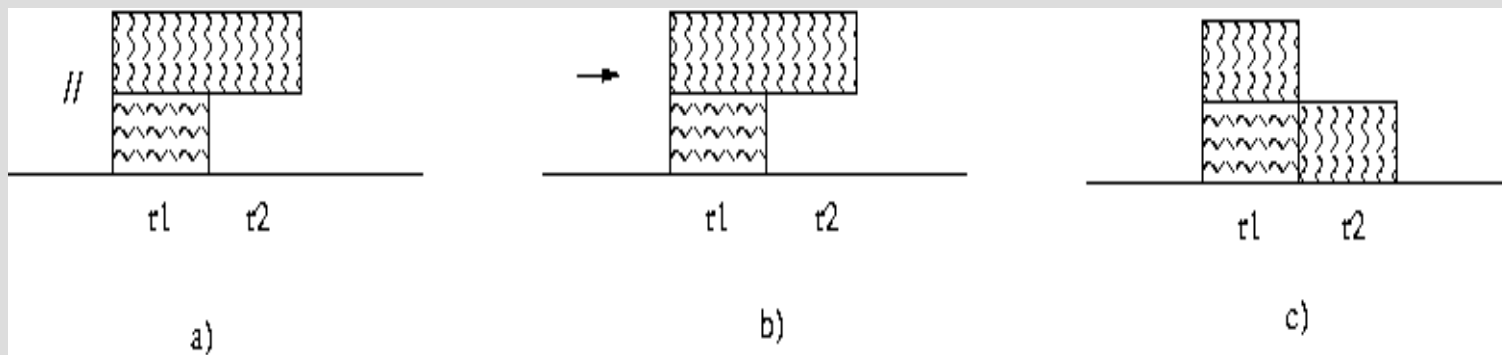
- Dynamique des négociations :
  - le négociateur : s'occupe de la gestion des micro-agents
  - les micro-agents (buts et engagements) : codent l'état des négociations de l'agent.
  - la matrice des négociations en attente
  - les contrats pris

# La gestion des négociations

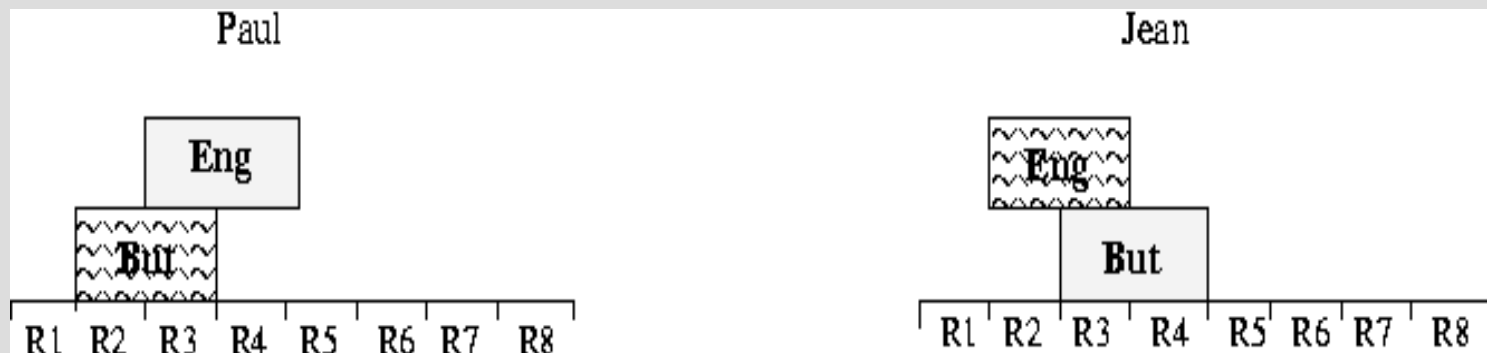


# Modes de gestion et situation de deadlock

- Modes de gestion :



- Situation de deadlock :



# La couche communication

- Définit les actes de langage nécessaires pour la communication entre des agents négociant avec GeNCA
- Utilise un mécanisme d'abonnement
- Interface Java :

```
public interface Communicator {  
  
    sendToServer(ServerMessage msg);  
  
    sendToAgent(Object to, ServerMessage msg);  
  
}
```

# La couche stratégique

- Une bonne stratégie nécessite un expert du domaine et de l'entreprise.
- 2 rôles => 2 stratégies : l'une pour les initiateurs, l'autre pour les participants.
- 2 listes de priorités : une pour les ressources, une pour les participants.
- Nous proposons une stratégie générique pour chacun des rôles, qui se base sur les listes de préférences fournies par l'utilisateur sur les ressources et sur les autres participants. Ces stratégies fournissent ainsi un comportement utilisable par défaut.

# La couche stratégique

- ```
public interface InitiatorStrategy {  
    void receiveAccept(String from, Object[] params);  
    void decide();  
    void defaultDecision(Vector whoHasAnswered);  
    void initModification();  
    void judge(String from, Object[] params);  
    void decideModification();  
    void defaultModification();  
}
```
- ```
public interface ParticipantStrategy {  
    void whatDoIAnswer(Contract c);  
    Object[] proposeModification();  
}
```

# Outils pour la création de stratégies

- Ressources :
  - libres, libres et dans l'ordre de priorité donné par l'utilisateur, prises par un initiateur moins important et dans l'ordre de priorité
- Contrats passés :
  - liste des contrats proposés par un initiateur donné, le précédent contrat proposé par un initiateur donné
  - les ressources qui ont été proposées par un initiateur donné, la ressource la plus/moins proposée pour un initiateur donné, le nombre de fois qu'une ressource a été proposée par un initiateur donné
  - les initiateurs qui ont proposé une ressource donnée.
- Résultats passés :
  - taux de succès avec un participant et une ressource donnés
  - taux de rétractation d'un participant et une ressource donnés

# La stratégie par défaut de l'initiateur

1. SI assez accords => confirme
2. SINON SI nbTours < nbToursMax => demande modification

$$\text{note}(r_i) = \text{priorité}(r_i, \text{init}) * \text{priorité}(\text{init}, \text{init}) + \sum_{j=1}^n \text{priorité}(r_i, p_j) * \text{priorité}(p_j, \text{init})$$

- (a) SI  $\exists$  nouvelle possibilité => propose nouveau contrat
- (b) SINON SI nbTours < nbToursMax => demande modification
- (c) SINON annule

3. SINON annule



# Stratégie par défaut du participant

- La stratégie du participant consiste à accepter une proposition de contrat lorsque les ressources sont libres ou prises par un initiateur possédant une priorité inférieure à celle de l'initiateur du nouveau contrat.
- En cas de demande de modification, le participant propose des ressources libres dans l'ordre de préférence qu'il leur a donné.

# Le paquetage GeNCA

- Il fournit la couche de négociation, le serveur de noms de la couche communication et des implémentations des interfaces de communication et de stratégies.
- Les implémentations de la couche communication permettent d'utiliser n'importe quelle plate-forme d'agents distribués, des agents threadés centralisés et la communication par mail. Pour tout autre mode de communication, il faut implémenter l'interface Communicator et créer un agent jouant le rôle de serveur de noms.
- Les implémentations de la couche stratégique sont celles décrites auparavant.
- Il fournit une interface graphique de base pour la communication avec l'utilisateur.
- Chaque agent possède deux modes de fonctionnement : automatique ou manuel.

# Pour utiliser GeNCA, il faut :

- implémenter les interfaces des couches communication et stratégie spécifiques à l'application si les implémentations par défaut ne conviennent pas.
- définir les fichiers XML de configuration du système (ressources communes, paramètres de négociation par défaut ...) et de l'agent (stratégie à utiliser, paramètres de négociation ...).
- écrire ses agents en incluant le négociateur de la couche négociation.
- étendre la classe Contract et modifier l'interface graphique si d'autres paramètres de négociation sont nécessaires.

# Applications

- Plusieurs applications réalisées avec GeNCA :
  - Un système de prise de rendez-vous
  - Un système d'enchères à offres scellées
  - Un système d'enchères hollandaises
  - Un système de création d'emplois du temps
  - Un système de choix d'un restaurant
- Disponibles sur le site de GeNCA :

<http://www.lifl.fr/SMAC/projects/genca>

# Un système de prise de rendez-vous

- Chaque agent doit être capable de négocier des rendez-vous pour le compte de l'utilisateur.
- Chaque utilisateur possède un agenda privé avec des plages horaires libres ou non.
- Chaque utilisateur peut négocier un rendez-vous pour une ou plusieurs tranches horaires avec un ou plusieurs participants.
- Les ressources à négocier sont les tranches horaires et sont communes à tous les participants.
- Les stratégies par défaut conviennent.
- Pour la communication, utilisation d'une implémentation fournie (Madkit par exemple).
  - => Seul le fichier de configuration doit être écrit.

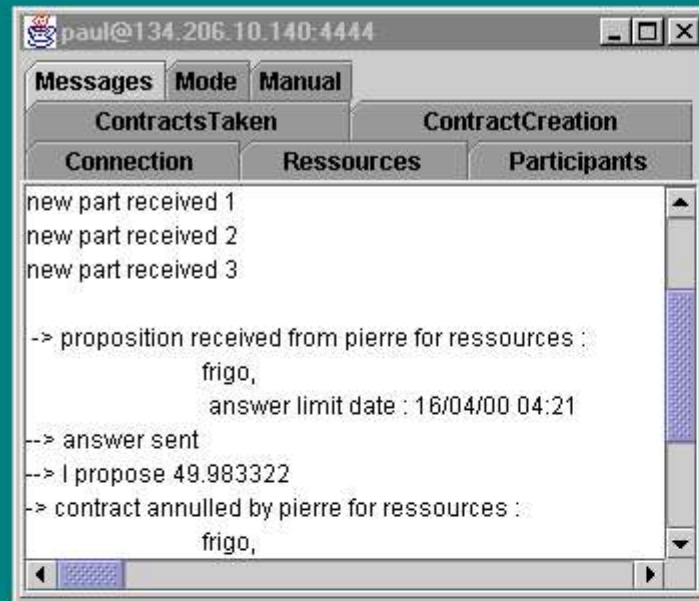
```
<?xml version="1.0"?>
<!DOCTYPE genca SYSTEM "genca.dtd" >
<genca>
  <application-type>rdv</application-type>
  <resources-list>
    <resource>14h-15h</resource>
    <resource>15h-16h</resource>
    <resource>16h-17h</resource>
    <resource>17h-18h</resource>
  </resources-list>
  <default-initiator-strategy>
    fr.lifl.genca.strategy.DefaultInitiatorStrategy
  </default-initiator-strategy>
  <default-participant-strategy>
    fr.lifl.genca.strategy.DefaultParticipantStrategy
  </default-participant-strategy>
```

```
<protocol>
  <minAgreements>100%</minAgreements>
  <answer-delay>10</answer-delay>
  <default-answer value="refuse"/>
  <nbRounds>20</nbRounds>
  <nb-modifications-by-round>5</nb-modifications-by-round>
  <retraction-allowed value="true"/>
  <nbRenegotiations>3</nbRenegotiations>
</protocol>
<simultaneity value="deferred"/>
<window value="true"/>
<default-priority value="5"/>
</genca>
```

# Avantages

- Contrairement à l'outil Microsoft qui collecte les agendas des utilisateurs afin de trouver un créneau horaire libre pour tous, GeNCA permet aux utilisateurs de garder leur agenda privé.
- Echange de créneaux horaires libres/préférés par l'utilisateur.
- Prise en compte des préférences de l'utilisateur pour les horaires mais aussi pour les participants.
- Renégociation automatique d'un rendez-vous qui doit être déplacé.





# Conclusion

- Notre proposition :
  - Architecture en 3 couches qui sépare le coeur de la négociation des considérations spécifiques aux applications telles que la communication et la stratégie de négociation.
  - Propriétés de la négociation paramétrables (agent/ système/ contrat)
  - Renégociation automatique
  - Notion étendue de l'idée de contrat
  - Système de gestion des négociations simultanées
  - Mécanismes de gestion des deadlocks
  - Stratégies générales

# Conclusion

- Avantages :
  - Les couches de base d'un système de négociation n'ont pas à être réécrites.
  - Un comportement par défaut permet d'effectuer tout de suite une négociation basique
  - Le contrat peut être affiné
  - Possibilité de prendre en compte les résultats des négociations antérieures
  - La description du protocole est abstraite
  - Différents problèmes peuvent être traités (applicatifs, stratégiques, environnementaux )
  - Mode de réponse manuel ou automatique

# Perspectives

- Extraire le protocole afin de pouvoir en changer (par réification et méta-interprétation). Fournir une bibliothèque de protocoles
- Intégrer d'autres types de négociations (multi-step, combinée, persuasion/argumentation ...)
- Analyser (et réaliser) d'autres applications de négociations
- Amélioration du niveau stratégique

# Pour plus d'infos

consulter le site web de GeNCA :

<http://www.lifl.fr/SMAC>

rubrique GeNCA

ou m'écrire :

[mailto:verrons@lifl.fr?subject=\[GeNCA\]](mailto:verrons@lifl.fr?subject=[GeNCA])