





Agent UML



Plan

- ❑ Introduction aux agents
- ❑ Programmation orientée agent(AOP)
- ❑ Introduction à l'Agent UML
- ❑ Représentation des classes d'agent
- ❑ Modélisation des interactions entre agents



Qu'est ce qu'un agent

Un agent est un objet qui peut prendre des initiatives (an objet that can say `go'...) et refuser de répondre à une sollicitation (... and `no').

Il est caractérisé par les propriétés suivantes :

- o Autonomie
- o Réactivité
- o Capacité à agir
- o Sociabilité



Quand est ce que la notion d'agent est utile ?

- La notion d'agent est adaptable à de nombreux HW et SW systèmes, mais elle est particulièrement utile dans certains contextes où la complexité est assez élevée en rendant la notion d'objet insuffisante pour décrire le système.
- L'accent est mis sur le comportement des agents et non sur le contenu des objets (attributs et méthodes)



Programmation orientée agent (1)

- La programmation orientée agent (AOP) est une spécialisation de la programmation orienté objet (POO) dont la façon ou le système est conçu :

“Un système informatique est vu comme étant composé de modules de communications , chacun avec sa propre manière de gérer les messages.” [Shoham, 1993]

- L'état mental de modules (agents) se compose d'éléments tels que les croyances, les capacités et les intentions.



Programmation orientée agent (1)

- Dans un système multi_agent, chaque agent peut:
 - Informer les autres agents sur les faits
 - Offrir et demander des services
 - Accepter ou refuser des propositions
 - Concurrencer pour accéder aux ressources
 - Collaborer pour atteindre des objectifs communs



Qu'est ce qu'un Agent UML

- ❑ Agent UML est une notation de soutien pour les systèmes de développement orientés agent.
- ❑ Il consiste à utiliser le langage de modélisation UML et de son extension afin de représenter les agents, leurs comportements et leurs interactions entre eux.
- ❑ AUML n'est pas limité à l'utilisation d'UML, il est adapté même par d'autres approches.



Qui s'intéresse à AUML

- **OMG Special Interest Group:** recommande des normes pour la technologie des agents. (www.omg.org)
- **FIPA Modeling Technical Committee:** chargé d'élaborer une norme AUML. (www.auml.org)
- **Autres méthodes:** MESSAGE, Gaia, Tropos, Prometheus, MaSE, ...



Caractéristiques communes des Agents

Les agents partagent certaines caractéristiques communes :

- **Identificateur**

- Identifie chaque agent dans un sys multi-agents.

- **Role**

- Définit le comportement d'un agent dans une société (ex : vendeur,..)

- **Organization**

- Définit les relations entre les rôles (similaire aux organisations humaines ou animales telles que les hiérarchies, les groupes d'intérêt ou de troupes)

- **Capacité**

- Précise ce qu'est un agent est capable de faire et dans quelles conditions

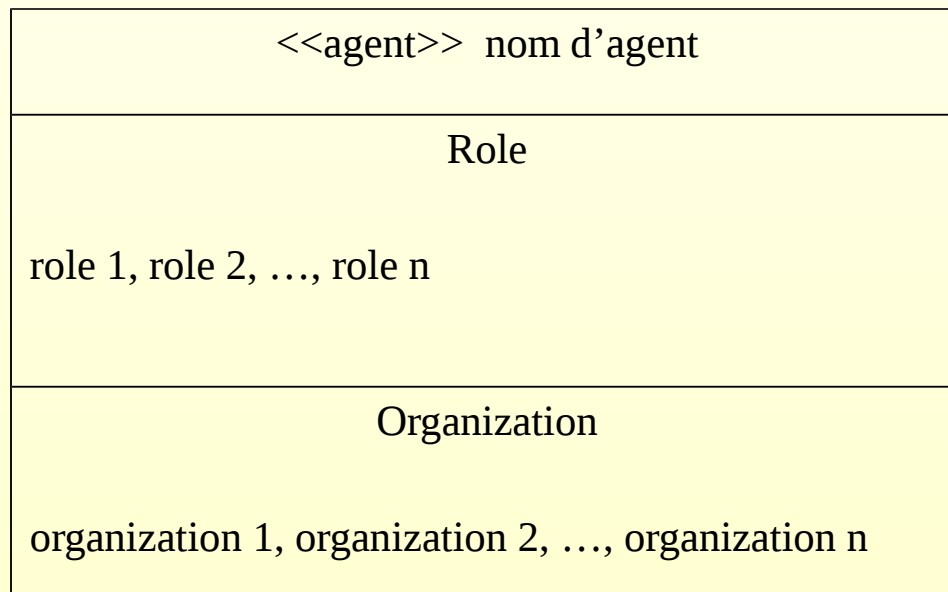
- **Service**

- Décrit une activité qu'un agent peut exercer, et est fourni à d'autres agents



Représentation des agents

- Les diagrammes de classes UML peuvent être utilisé pour représenter la vision statique des agents.



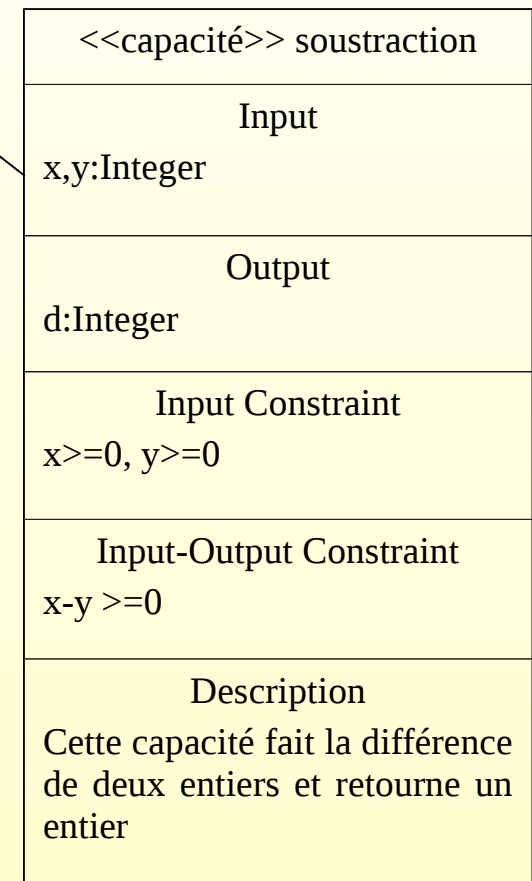
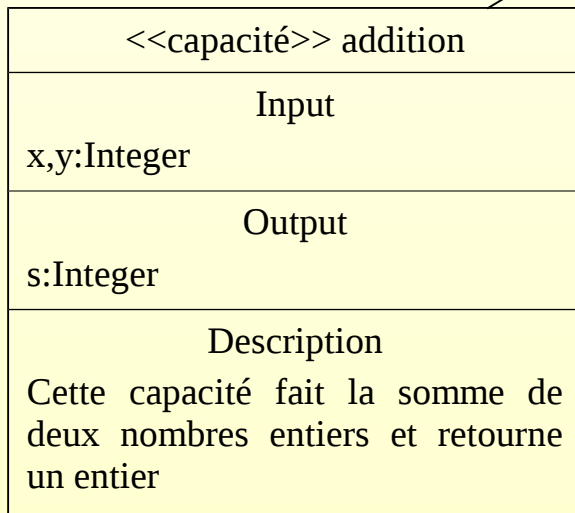
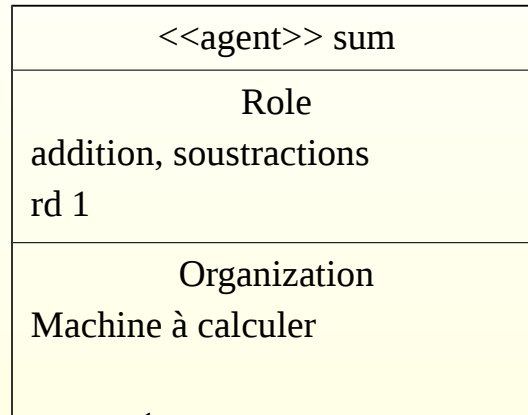
Représentation des capacités (1)

- Une capacité est composée des éléments suivants:
 - **Input (Entrée)**
 - Ce que l'agent doit recevoir en entrée pour réaliser sa tâche
 - **Output (Sortie)**
 - Ce que la capacité génère à la suite des travaux
 - **Input constraints (Contraintes d'entrée)**
 - Contraintes qui sont attendus à tenir avant que l'action définie par la capacité peut être réalisée
 - **Output constraints (Contraintes de sortie)**
 - Contraintes qui sont attendus à tenir dans la situation après l'action définie par la capacité a été réalisée
 - **Input-output constraints (Les contraintes d'entrées-sorties)**
 - Les contraintes qui doivent être titulaires aux situations d'entrée et de sortie
 - **Description**
 - Une description en langage naturel de la capacité

Représentation des capacités (2)

- Exemple d'une représentation de capacités :

L'agent «sum» a deux capacités d'exprimer le fait qu'il est en mesure de faire des additions et soustractions





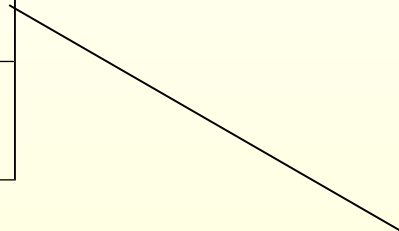
Représentation des services (1)

- Un service est composé des éléments suivants:
 - **Name**
 - Nom du service
 - **Description**
 - Description en langage naturel du service
 - **Type**
 - Type de service
 - **Protocol**
 - Liste de protocoles d'interaction supporter par le service
 - **Agent communication language**
 - Langues de communication utilisées dans ce service
 - **Ontology**
 - Liste des ontologies (info + connaissances) chargé par le service
 - **Properties**
 - Une liste de propriétés caractérise le service

Représentation des services (2)

<<agent>> sum
<p>Role addition, soustraction rd 1</p>
<p>Organization Machine de calcul</p>

<<service>> computation
<p>Description Ce service fait une addition à la demande du protocole Demande-ajout et rend une soustraction à la demande du protocole Demande-soustraction</p>
<p>Type computation</p>
<p>Protocol Demande-ajout demande-soustraction</p>
<p>Agent Communication Language FIPA ACL</p>
<p>Ontology Computation ontologie</p>

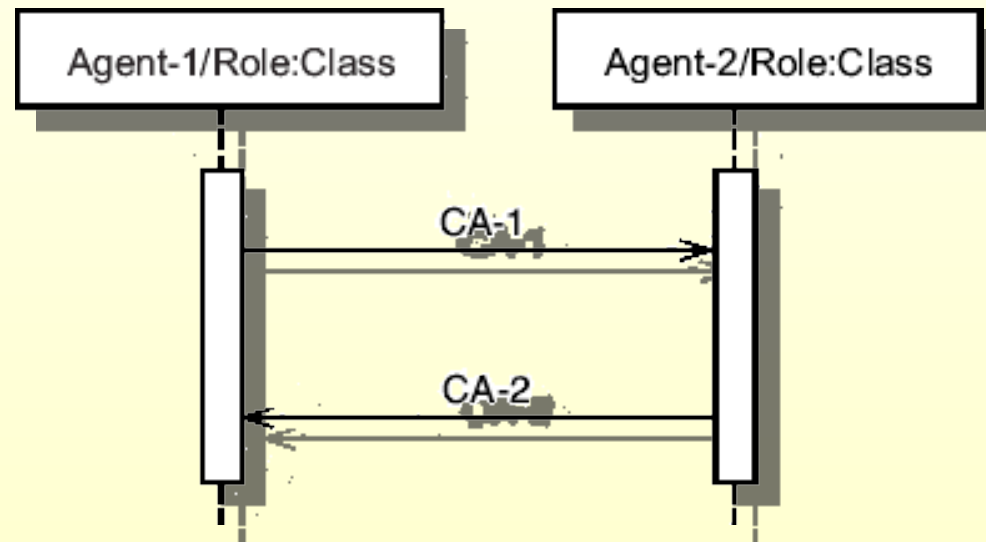


- Exemple d'une représentation d'un service :

L'agent «sum» exporte ou fournit un service qui fait des additions ou des soustractions sur demande

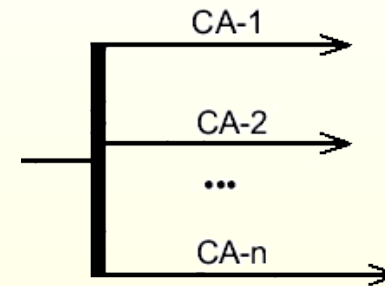
Représentation des interactions

- Interactions agent peuvent être représenté dans le standard UML en utilisant des diagrammes de séquence

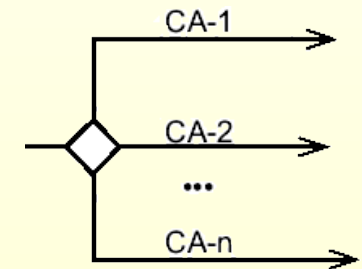


Intéractions simultanées(1)

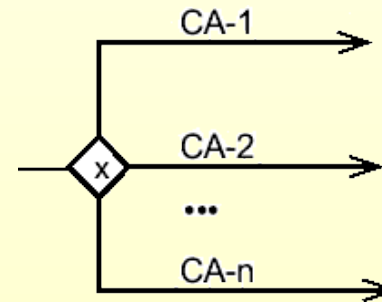
- UML a été étendu afin de représenter la communication simultanée des actes envoyés de l'expéditeur au destinataire.
 - a) Les actes de communication simultanées de CA-1 à CA-n sont envoyés en parallèle.
 - b) Une sélection de n actes est envoyé en parallèle (zéro ou plus).
 - c) Choix exclusif: un seul des actes de communication est envoyé.



(a)



(b)



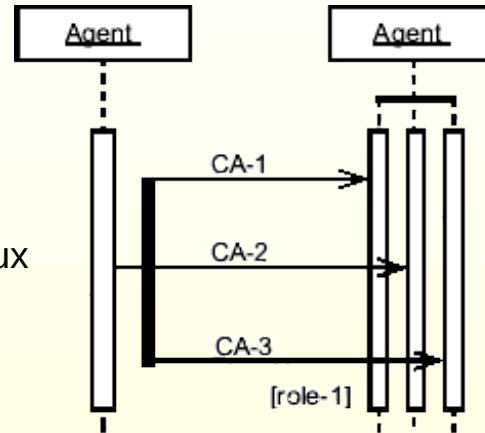
(c)

Intéractions simultanées (2)

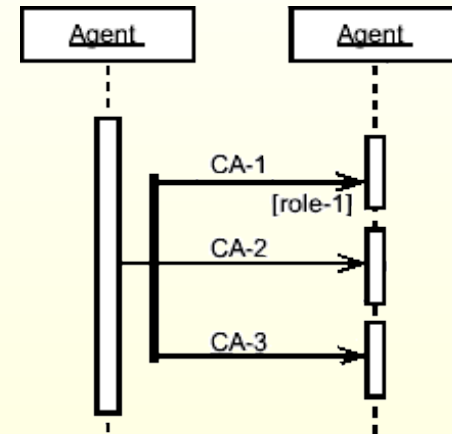
- a) Un agent envoie 3 Actes de communications parallèlement à un autre agent.

Le diagramme peut être interprété de deux manières différentes:

- Chaque CA est obtenue à partir du même agent / rôle par un thread différent de l'exécution
- Chaque CA est attribué à un rôle différent de l'agent (dans ce cas messages peuvent être annotés en précisant le rôle)

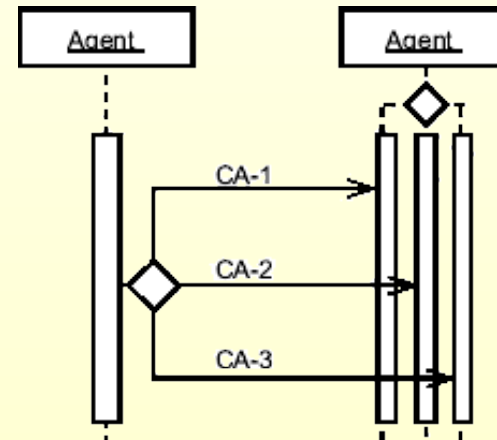


(a)



(b)

- b) Même sémantique de (a), mais avec une simple notation
- c) Choix de trois différents acte de communication reçu par trois différents threads (ou rôles)

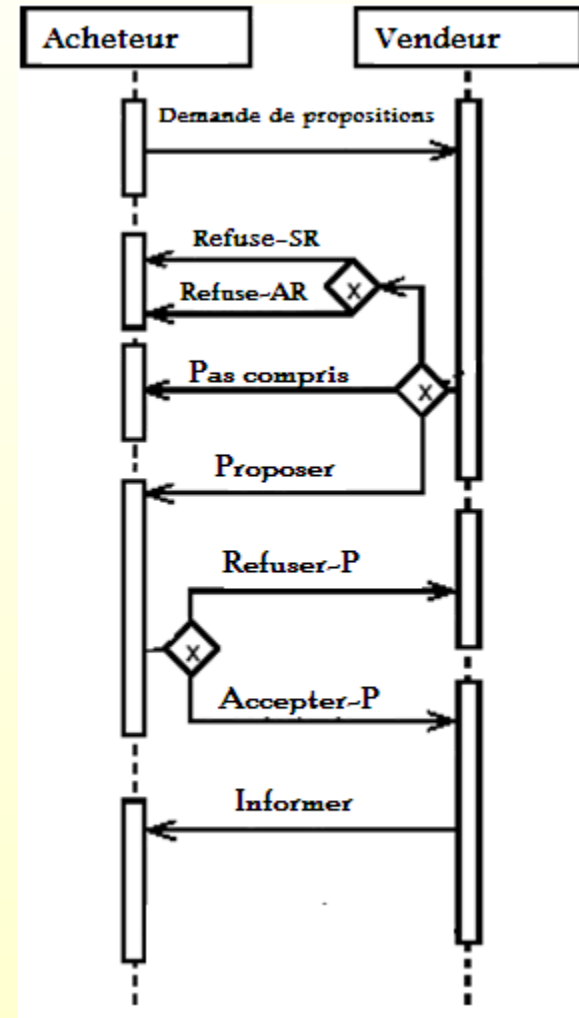


(c)

- **NOTE:** chaque CA simultanées peuvent être envoyés à différents agents

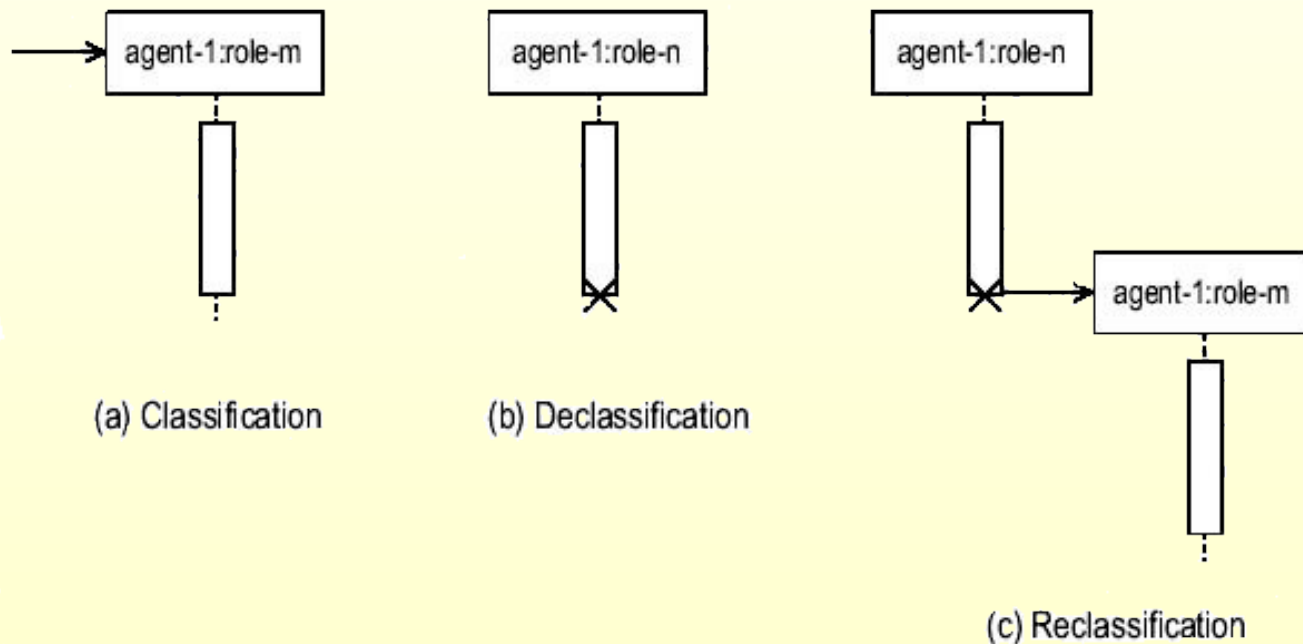
Exemple d'interaction

- L'acheteur envoie une demande de propositions pour le vendeur
- Le vendeur a trois options à choisir :
 - refuser (avec ou sans raisons)
 - dire qu'il ne comprenait pas
 - faire une proposition
- Si le vendeur a fait une proposition, l'acheteur a le choix de refuser ou d'accepter
- Lorsqu'il accepte, le Vendeur lui informe sur l'état de sa proposition



Gestion de rôles

- Diagrammes de séquence UML peuvent être utilisés pour représenter l'évolution du rôle des agents.



Rôle de l'objet dans AOP

- L'objet a la possibilité d'être inclus dans un système orienté-agent et peut communiquer avec les agents de passage de messages utilisant des méthodes.

