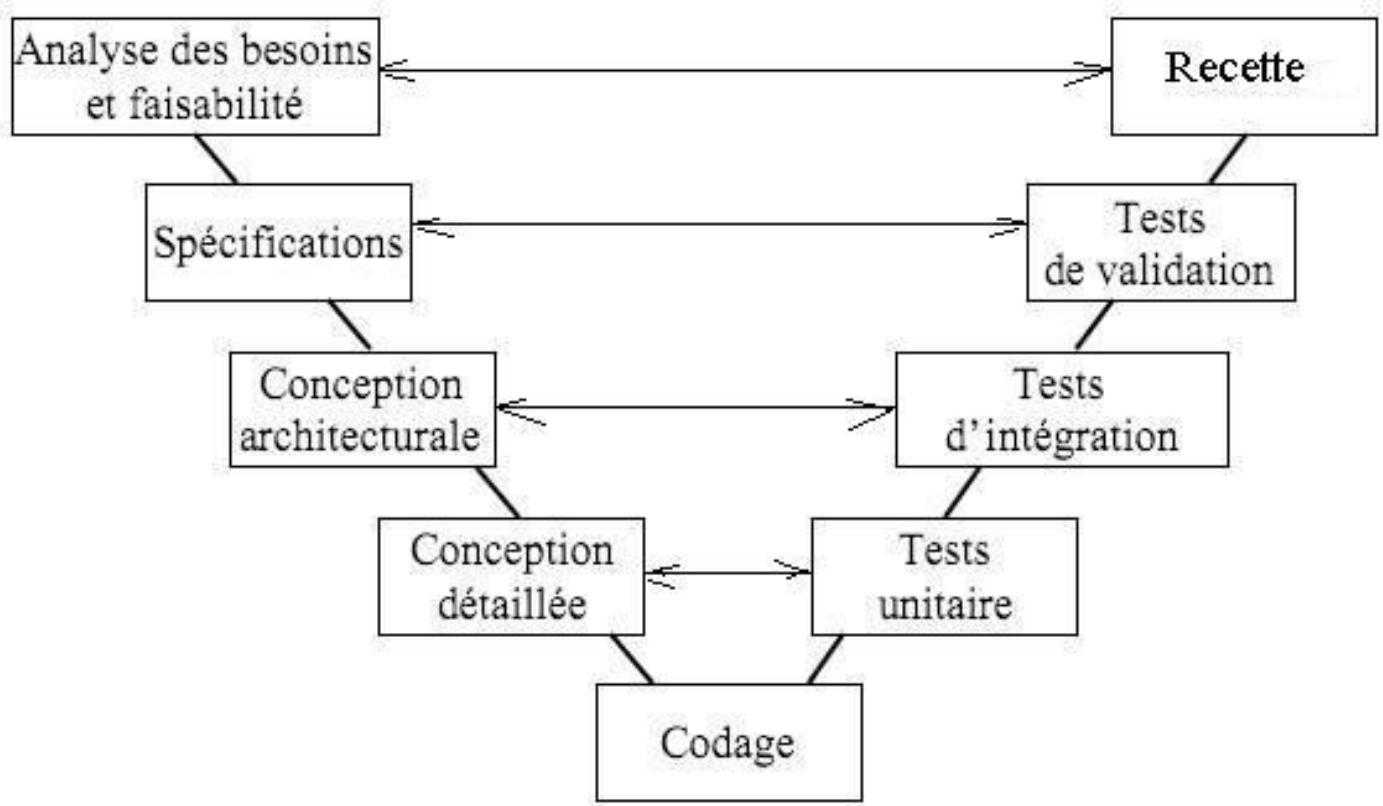
Analyse Orientée Objet

Cours 1 : Introduction

Diagramme de cas d'utilisation

Cycle de vie d'un projet logiciel

- Cycle en V :



Définitions

- Analyse : de la perception réelle vers la représentation
  - L'analyse consiste sur la base d'un cahier de charge de modéliser les acteurs et le système en ne s'intéressant pas à l'implémentation.
  - On parle d'approche métier :

    QUI (acteurs) fait QUOI (système)

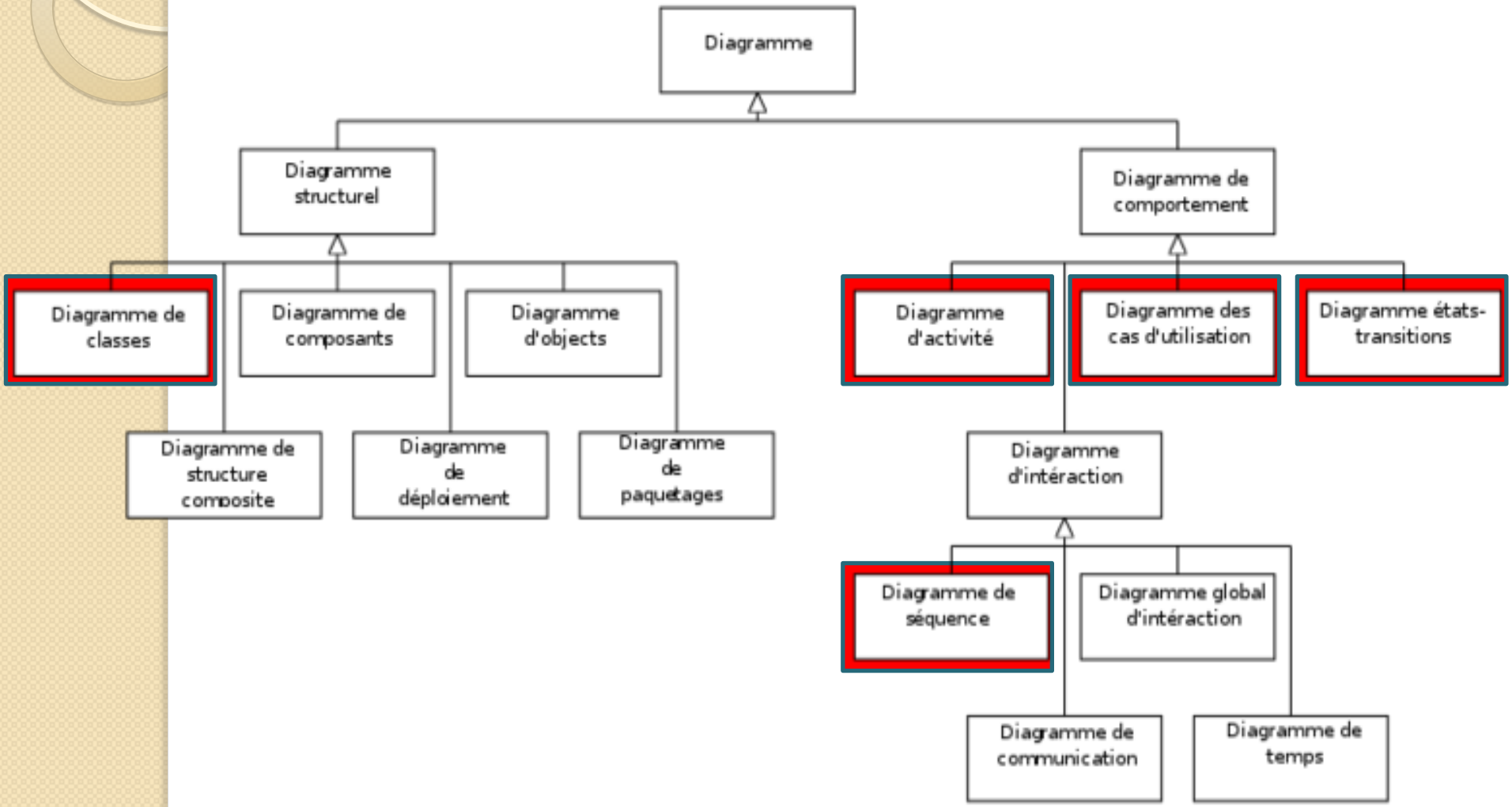
- Conception : de la représentation vers la représentation du prototype
  - La conception consiste sur la base de l'analyse à modéliser la réalisation informatique.
  - On parle d'approche informatique :

    COMMENT ?

Historiques d'UML

- UML : Unified Modeling Language, standard d'OMG
- Fusion de 3 langages de modélisation OO (Booch, OMT, OOSE) en 1995.
- Dernière version UML 2.4.1 août 2011
- UML offre un standard de modélisation pour spécifier, visualiser, modifier et construire les documents nécessaires au bon développement d'un logiciel orienté objet

Taxonomie des diagrammes



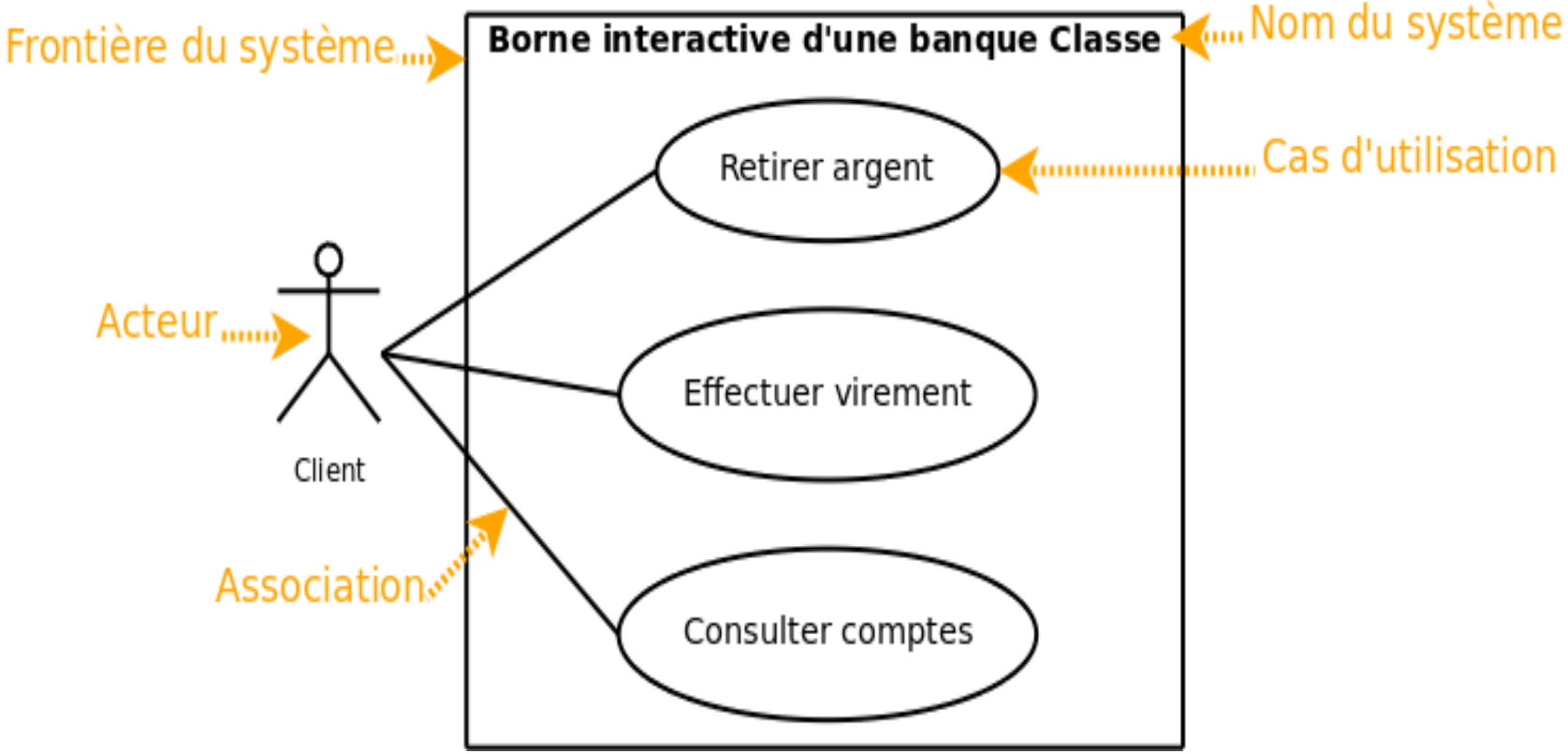
PHASE D'ANALYSE

- On commence par identifier :
  - Le système
  - Les acteurs qui interagissent avec le système
  - Les actions des acteurs sur le système

=> diagramme de cas d'utilisation

Éléments de diagramme de cas d'utilisation

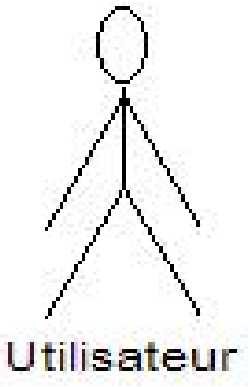
- Acteurs
- Cas d'utilisation



http://laurent-audibert.developpez.com/Cours-UML

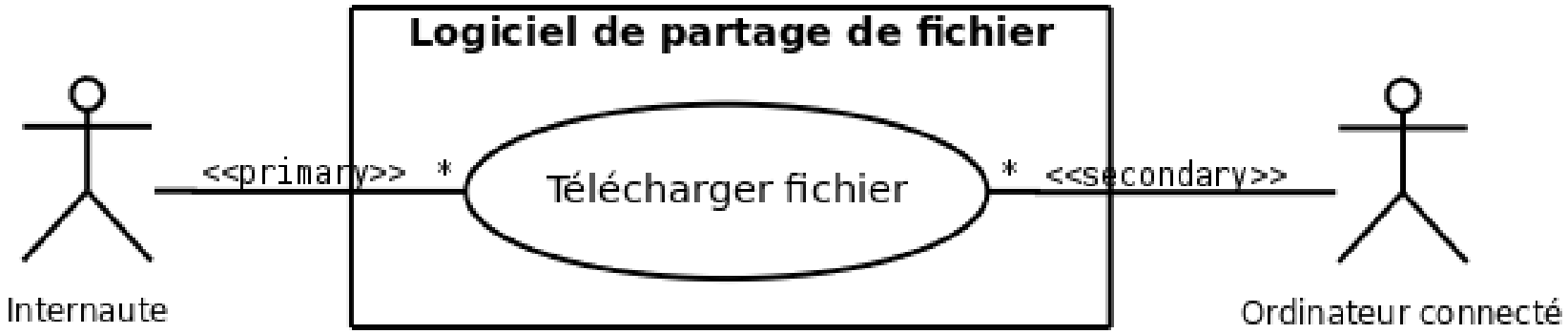
Les acteurs

- Un acteur est une entité externe (personne, imprimante, serveur, SGBD, …) qui interagit avec le système.
- Un même humanoïde peut être plusieurs acteurs.
- On définit donc un acteur par un ensemble de rôles qu'il a sur le système.

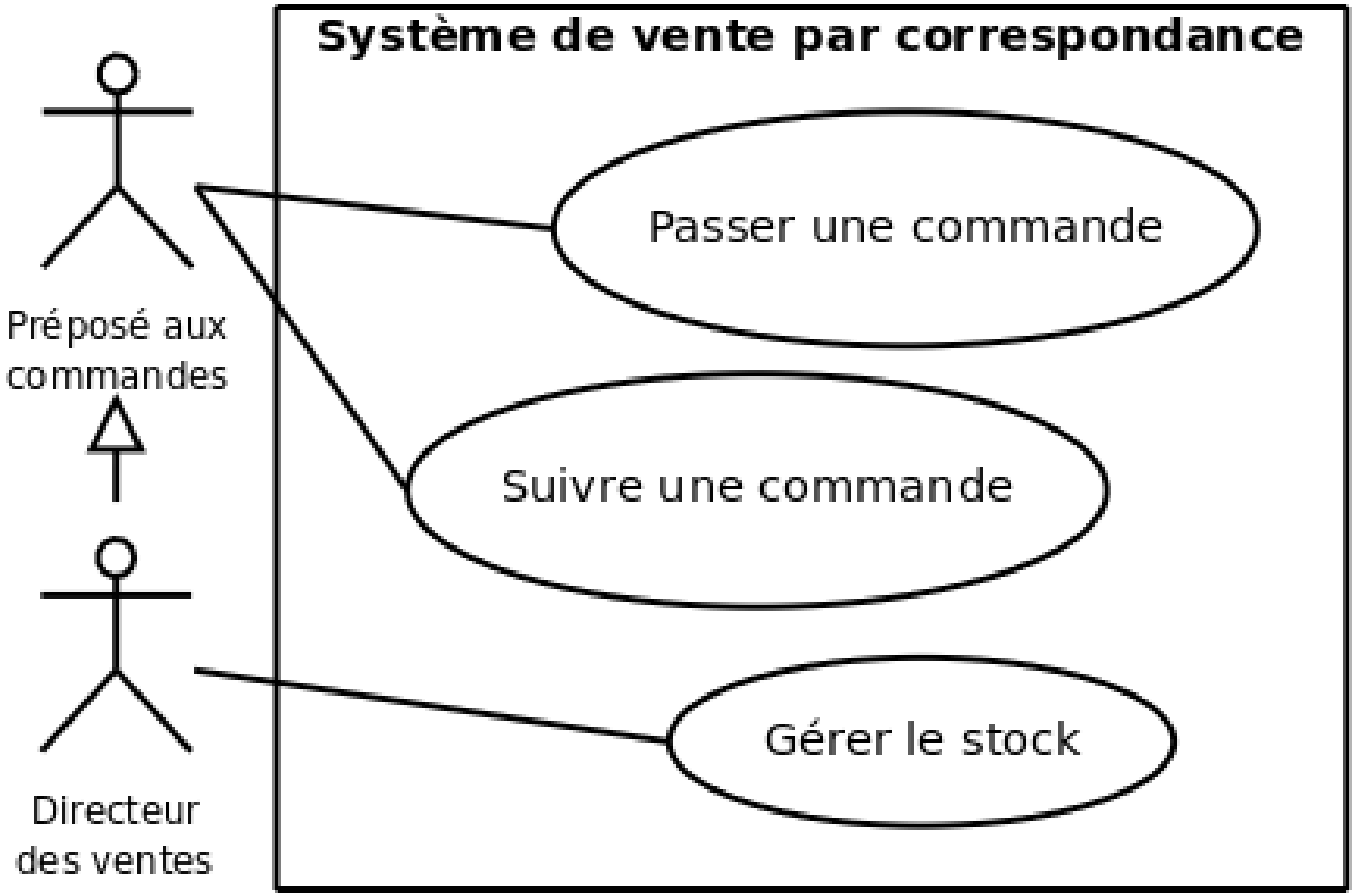


Type d'acteurs

- Acteur principal : le cas d'utilisation rend service à cet acteur
- Acteur secondaire



Relation entre acteurs : généralisation



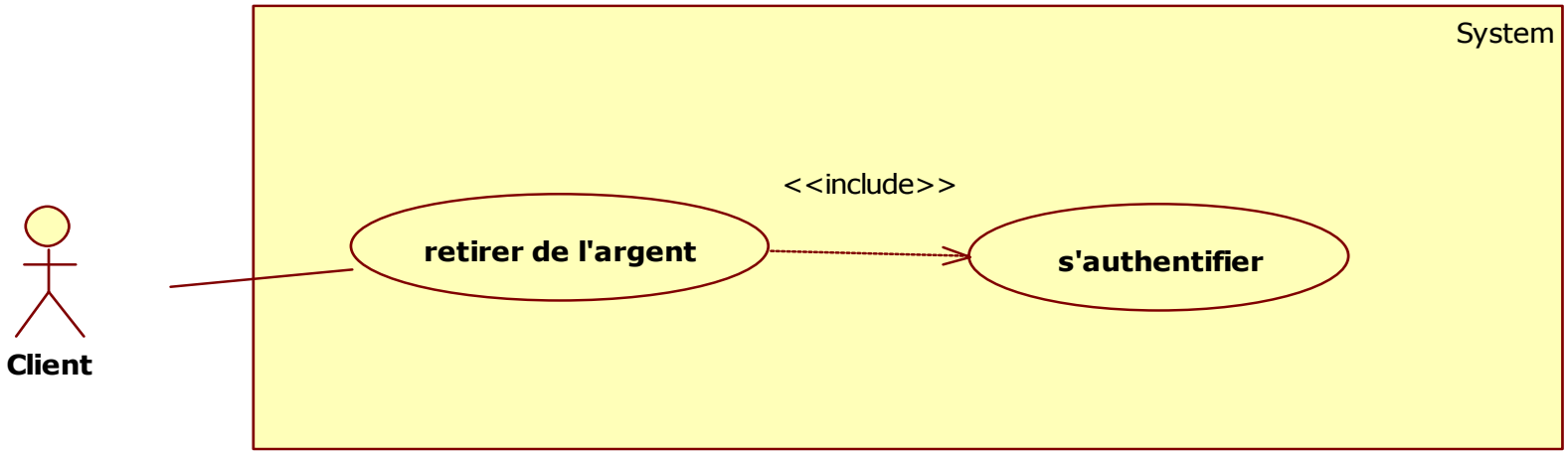
http://laurent-audibert.developpez.com/Cours-UML

Relations entre cas d'utilisation

1. Relation d'inclusion : <<include>>
2. Relation d'extension : <<extend>>
3. Relation de généralisation

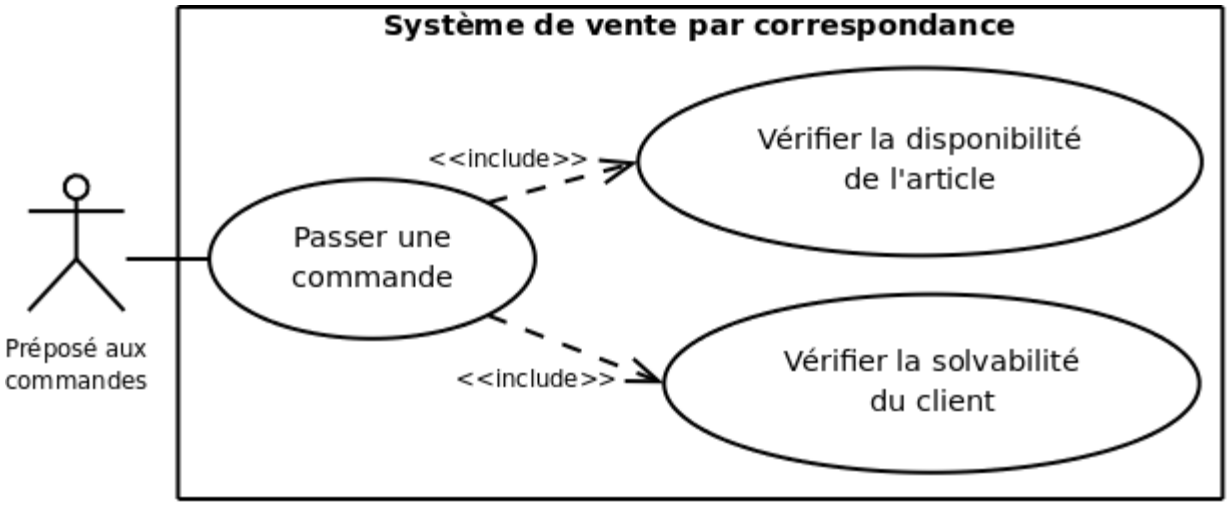
La relation <<include>>

- Un cas A inclut un cas B si le comportement décrit par le cas A **inclut systématiquement** le comportement du cas B
- Exemple :
  - accès aux informations d'un compte bancaire inclut nécessairement une phase d'authentification avec un identifiant et un mot de passe



La relation <<include>>

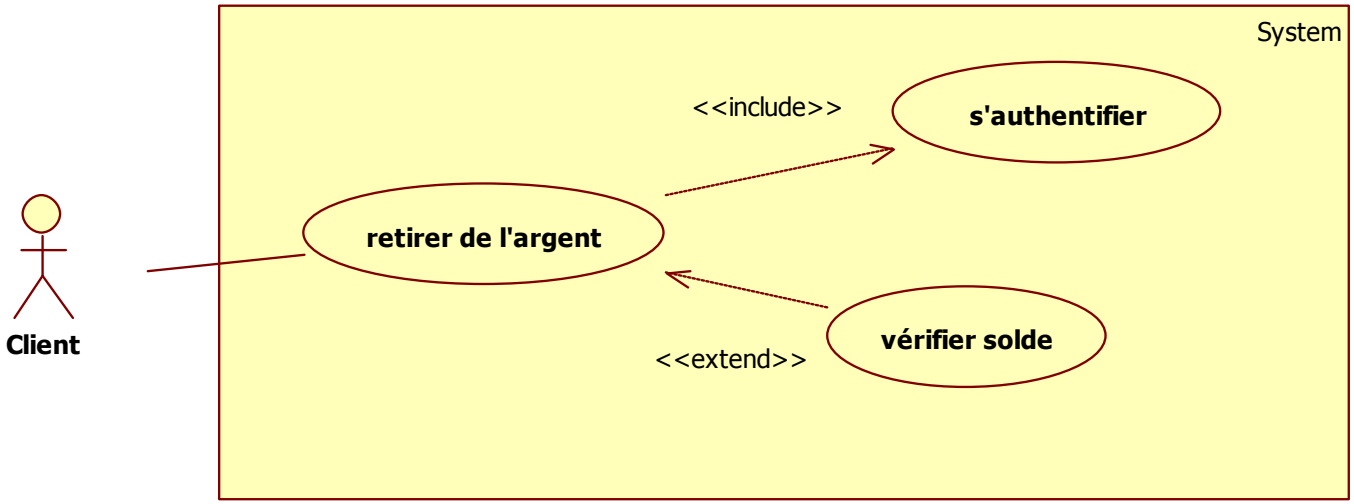
- L'inclusion permet de :
  - factoriser la description d'un cas d'utilisation qui est commune à d'autres cas d'utilisation
  - décomposer un cas complexe en sous-cas plus simples (mais à ne pas trop abuser !)



http://laurent-audibert.developpez.com/Cours-UML

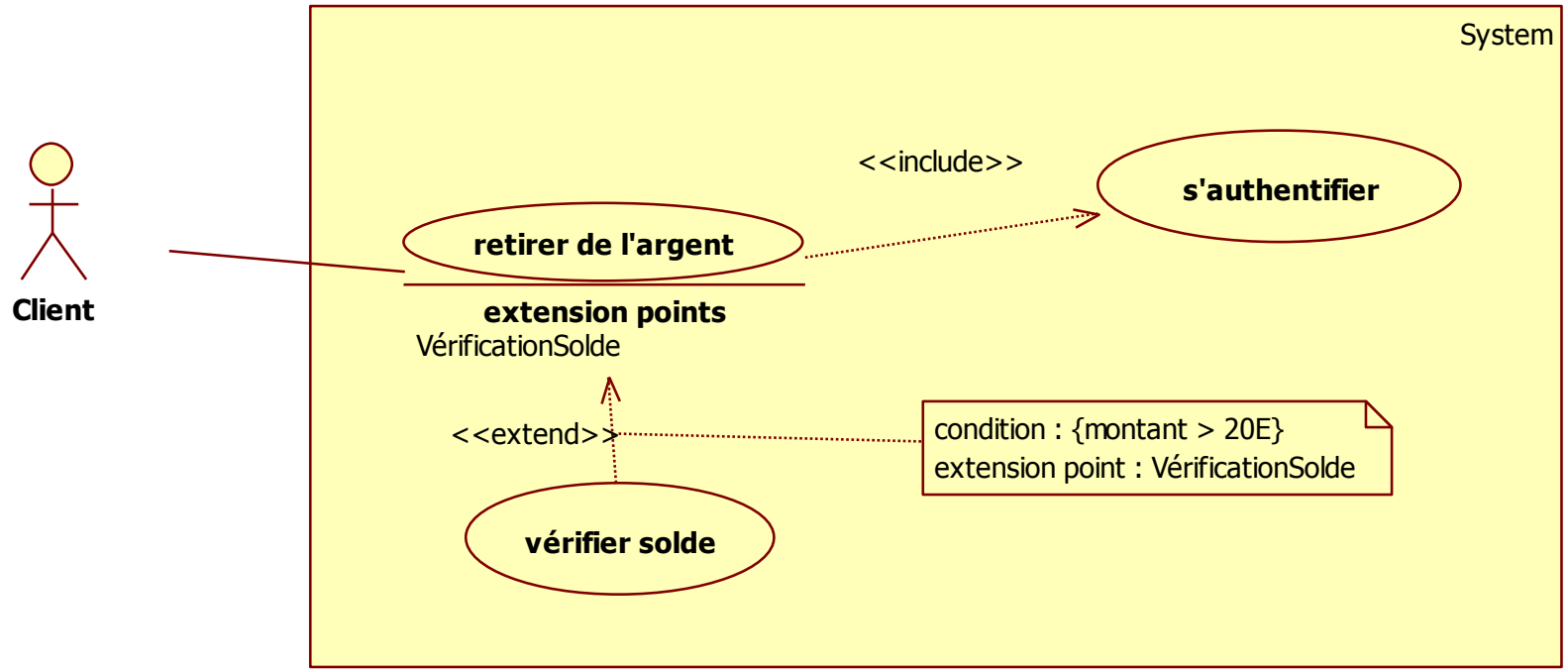
La relation <<extend>>

- Un cas A étend un cas B lorsque le cas A **peut être** appelé au cours de l'exécution du cas B.
- Exemple :
  - quand la demande de retrait dépasse 20 euros, on fait une vérification du solde du compte



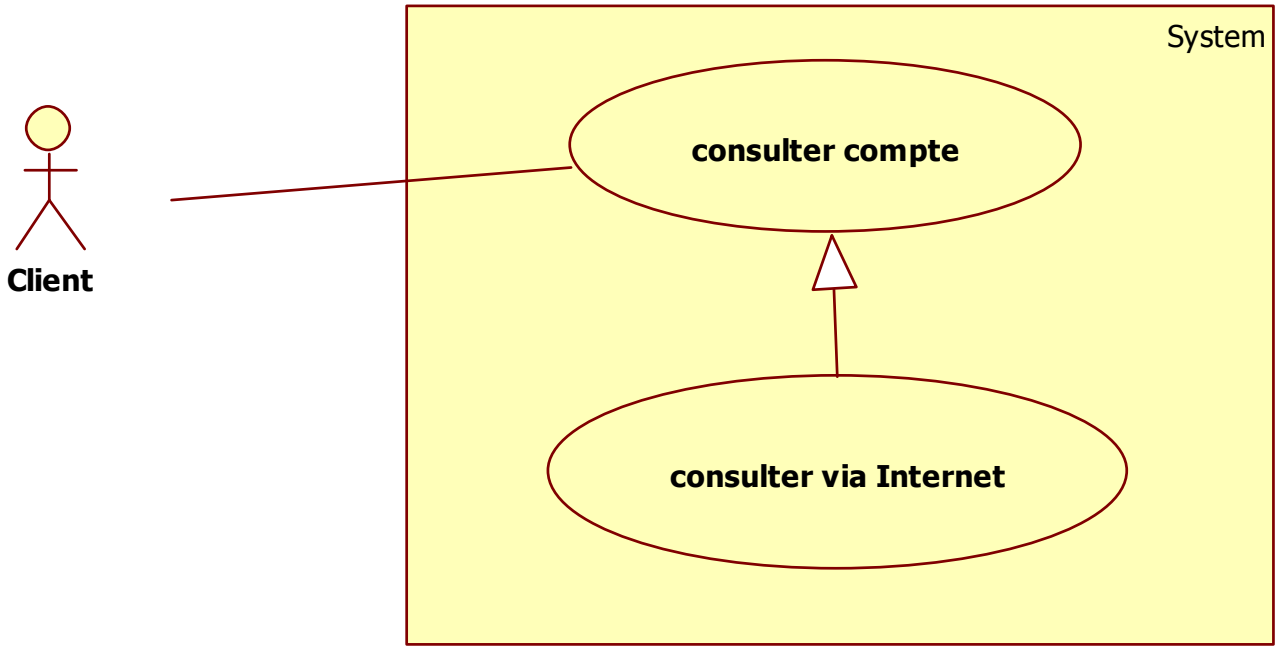
La relation <<extend>>

- L'extension peut intervenir à un point précis du cas étendu : **point d'extension**.
- **Condition** d'extension : contrainte dans une note



La relation <<generalize>>

- Un cas A est une généralisation d'un cas B si B est un cas particulier de A.
- Exemple :
  - la consultation d'un compte *via* Internet est un cas particulier de la consultation.



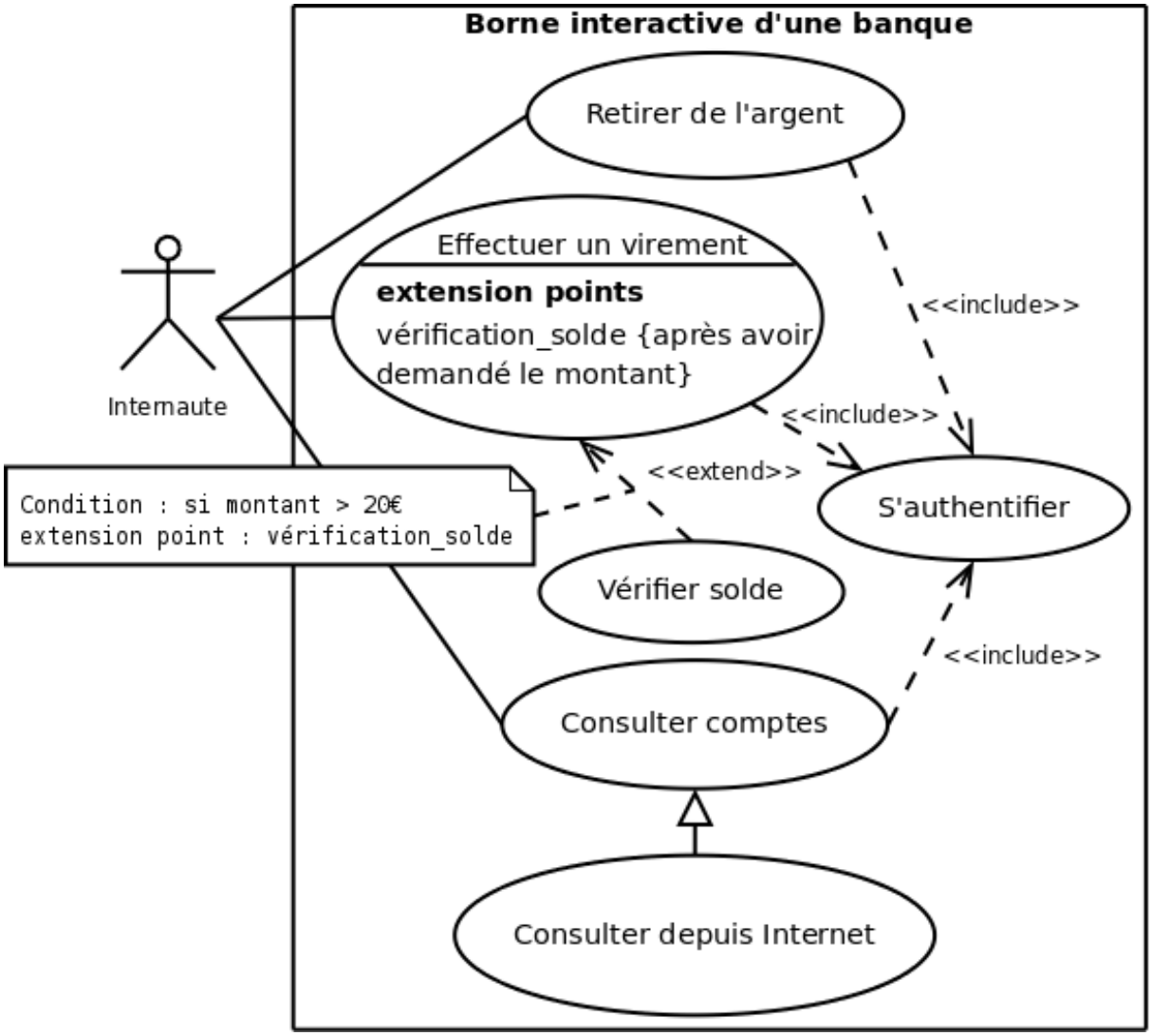
Fiche descriptive d'un cas d'usage

- A chaque cas d'usage, on associe une fiche descriptive.
  1. une description du cas d'usage
  2. règle d'initiation : qu'est ce qui déclenche la transaction ?
  3. règle de terminaison : qu'est ce qui termine la transaction ?
  4. règle d'exception : qu'est ce qui déclenche le cas d'usage quand la règle d'initiation n'est pas vérifiée ?
  5. les relations avec d'autres cas d'usage : extension, inclusion et généralisation

Fiche descriptive d'un cas d'usage

- Exemple : cas Retirer l'argent
  1. Description : il s'agit du cas d'usage qui permet à un client de retirer une somme d'argent à partir d'un distributeur
  2. Règle d'initiation : le client doit avoir sa carte bancaire valide et le code
  3. Règle de terminaison : le solde du compte bancaire est mis à jour
  4. Règle d'exception :
  5. Relations : ce cas inclut le cas s'authentifier

Exemple de diagramme de cas d'utilisation complet

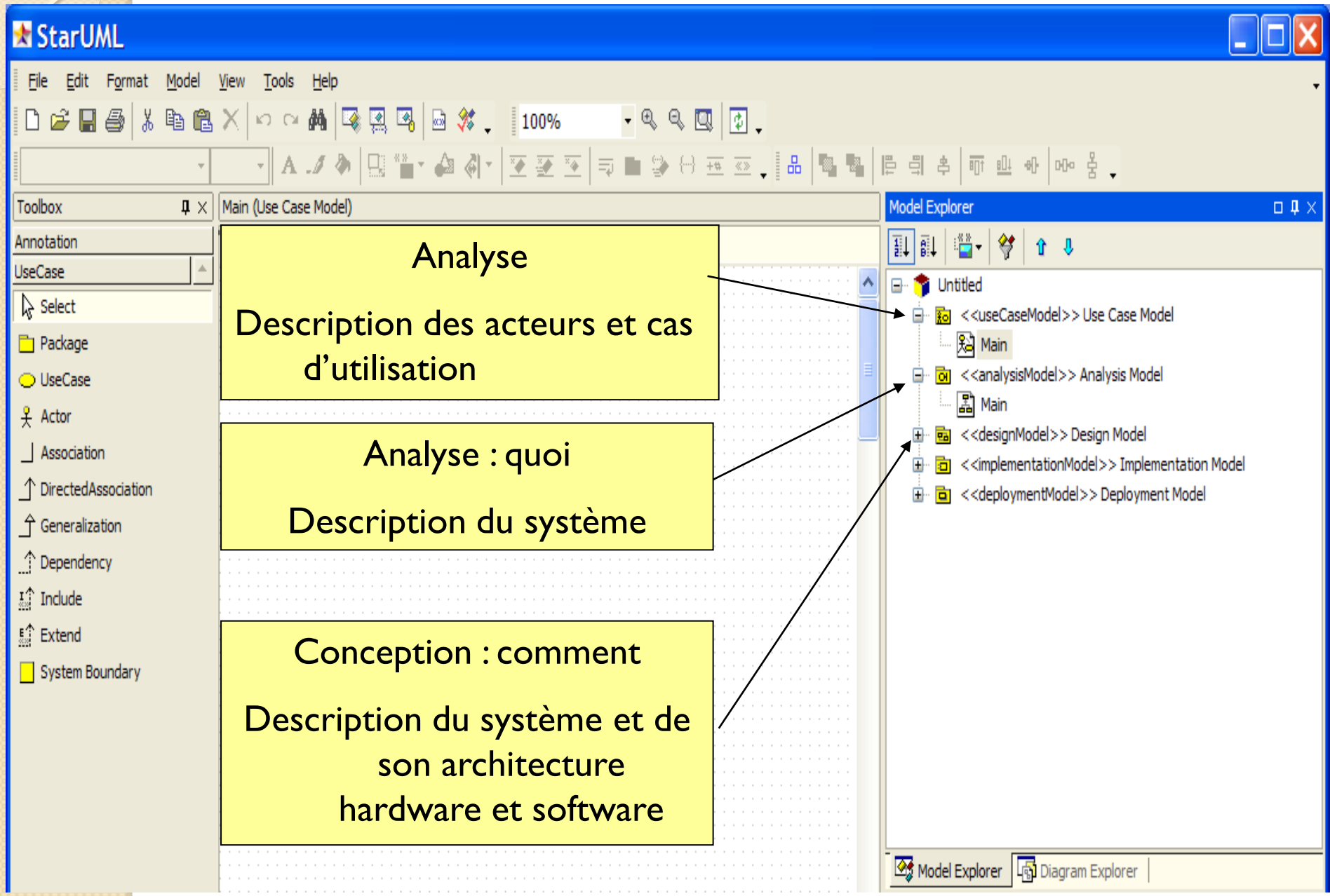


http://laurent-audibert.developpez.com/Cours-UML

Résumé

- Bien identifier les acteurs :
  - extérieurs du système et dialoguent avec lui
  - humains, systèmes informatiques, hardware, …
- Bien recenser les cas d'utilisation :
  - cas d'utilisation = fonction métier du système
  - limiter le nombre de cas : règle 5-7 (ne pas retomber dans une décomposition fonctionnelle descendante hiérarchique)
  - nommer avec un verbe à l'infinitif suivi d'un complément (point de vue de l'acteur)
- Bien organiser les diagrammes (par thème, par niveau d'abstraction, …) pour les systèmes complexes
- Attention : pas de notion temporelle dans un diagramme de cas d'utilisation !

Démonstration StarUML


StarUML
File Edit Format Model View Tools Help
100%
Toolbox
Annotation
UseCase
Select
Package
UseCase
Actor
Association
DirectedAssociation
Generalization
Dependency
Include
Extend
System Boundary
Main (Use Case Model)
Analyse
Description des acteurs et cas d'utilisation
Analyse : quoi
Description du système
Conception : comment
Description du système et de son architecture hardware et software
Model Explorer
Untitled
<<useCaseModel>> Use Case Model
Main
<<analysisModel>> Analysis Model
Main
<<designModel>> Design Model
<<implementationModel>> Implementation Model
<<deploymentModel>> Deployment Model
Model Explorer
Diagram Explorer