Conception Orienté Objet
Cas d'Utilisation / Use Case

INSPIRÉ DU COURS DE FRÉDÉRIC MALLET

AMOSSE EDOUARD

UML2 – Use Case

Objectifs :

- Montrer comment capturer les exigences des utilisateurs
- Lire et interpréter les Cas d'Utlisation (CU).

Le comportement du système

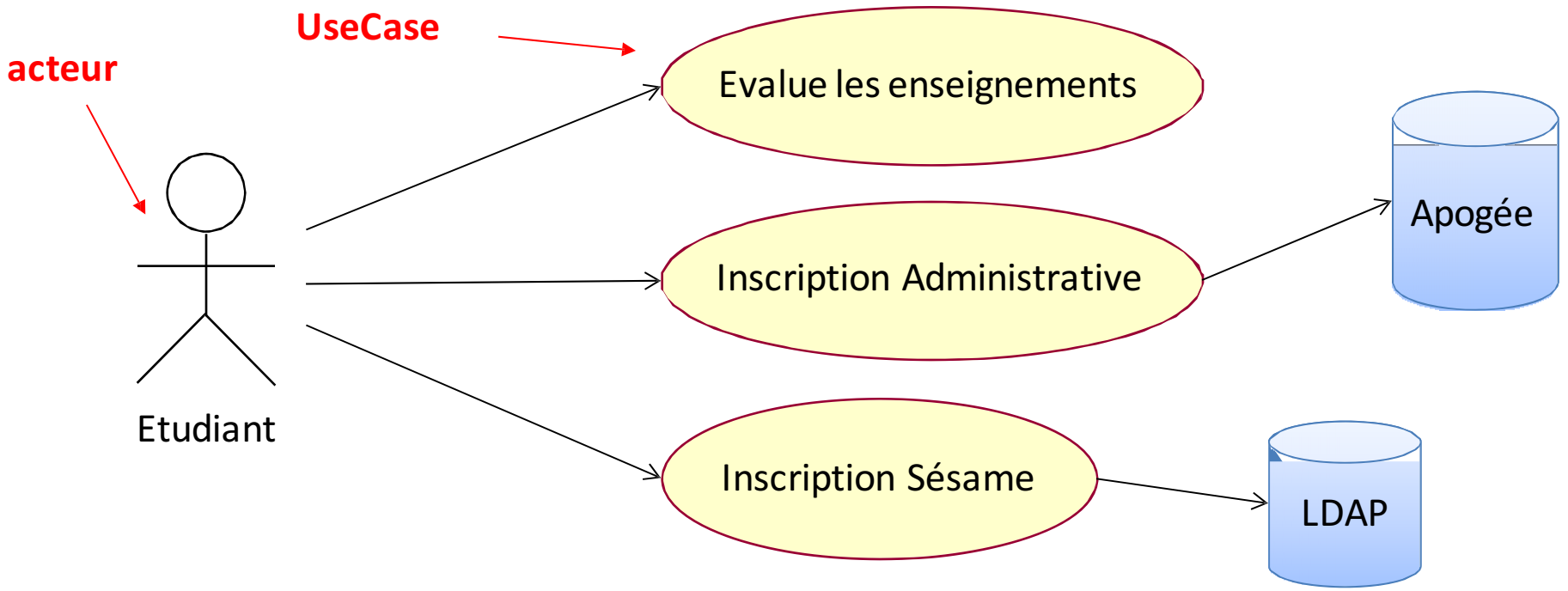
- C'est l'ensemble des actions et réactions du système
  - L'ensemble de la fonctionnalité et des responsabilités du **système** et de son **environnement** sont capturés par un diagramme de *UseCase*.
  - L'environnement est l'ensemble des **acteurs** (personnes, logicielles, machines) qui interagissent avec le système et ne sont **PAS** à concevoir !
  - Doit être approuvé par les acteurs (y compris le client) !
- Les diagrammes d'activités
  - Capture l'ensemble des actions à réaliser, l'algorithme à réaliser

Exigences Fonctionnelles et Non-Fonctionnelles

- **Exigences Fonctionnelles**
  - Définir une fonction du système à développer.
  - Décrit ce que le système doit faire.
  - Examplef:
    - Inscrire étudiant,
    - Caluler Moyenne, Imprimer bulletin
- **Exigences Non fonctionnelles**
  - Exigence caractérisée par une propriété désirée du système
    - Performance, Robustesse, Conviviabilité, Maintenabilité
  - Exemples
    - Le temps de réponse à une inscription ne doit pas dépasser 30 secondes…
    - Le coût du projet ne doit pas être supérieur à 150K

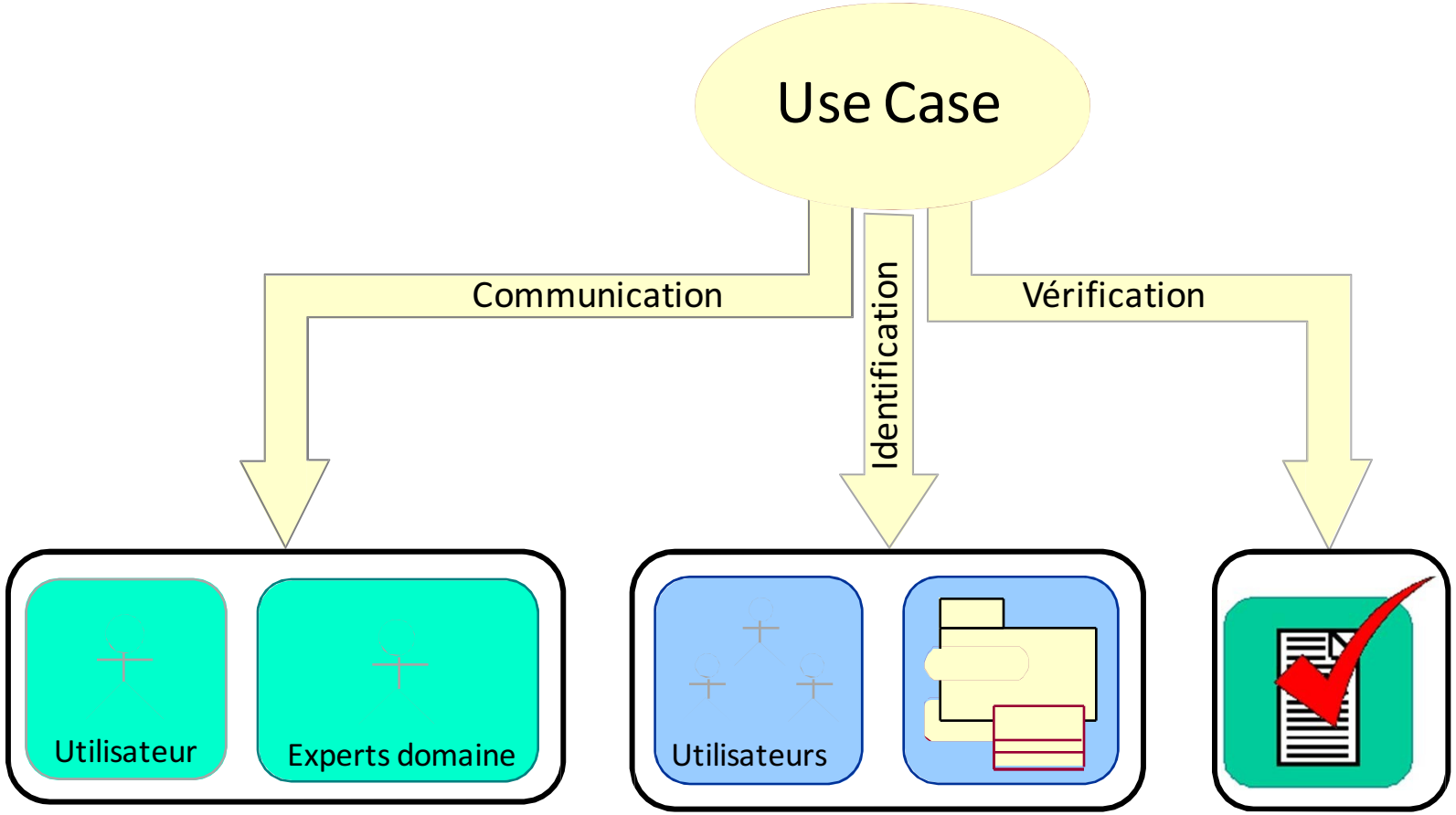
Qu’est ce qu’un CU?

- Un modèle de CU décrit les exigences fonctionnelles du système.
- Décrit la façon dont le système fonctionne.



Intérêts des CUs

- Communication
- Identification
- Vérification



Composantes d’un diagramme de CU

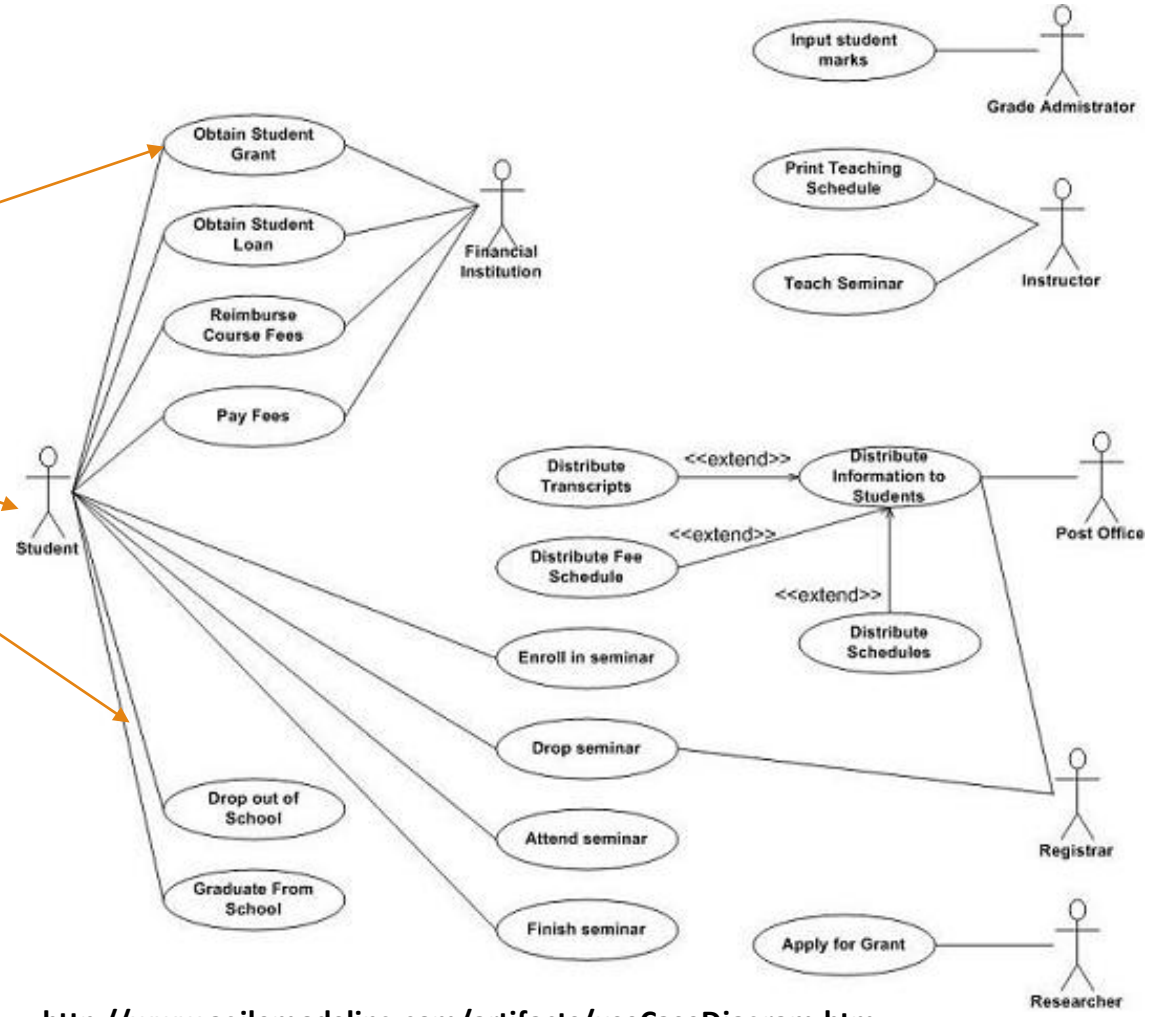
- **Les cas d’utilisation** : Sequence d’actions représentant une valeur mesurable pour un acteur du système. Les CUs sot representés en utilisant des ellipses.
- **Acteurs**: Les acteurs sont des personnes, des organisations, autres systèmes jouant un rôle dans une ou plusieurs activités du système. Ils sont modélisés par des icones (semblables) à l’acteur. Ex: image d’une personne, d’une organisation etc…
- **Associations** : Les associations representent les interactions entre les acteurs et les CUs. Les associations sont modélisés par des lignes continues.
- **Frontières** (optionnelles) : Ce sont des rectangles permettant de représenter les limitations du système.
- **Packages** (Optionnelles) : En UML, les packages sont utilisés pour grouper les CUs en groupe en fonction d’un découpage donné du système.

Diagramme de CU – Exemple (1)

Use Case

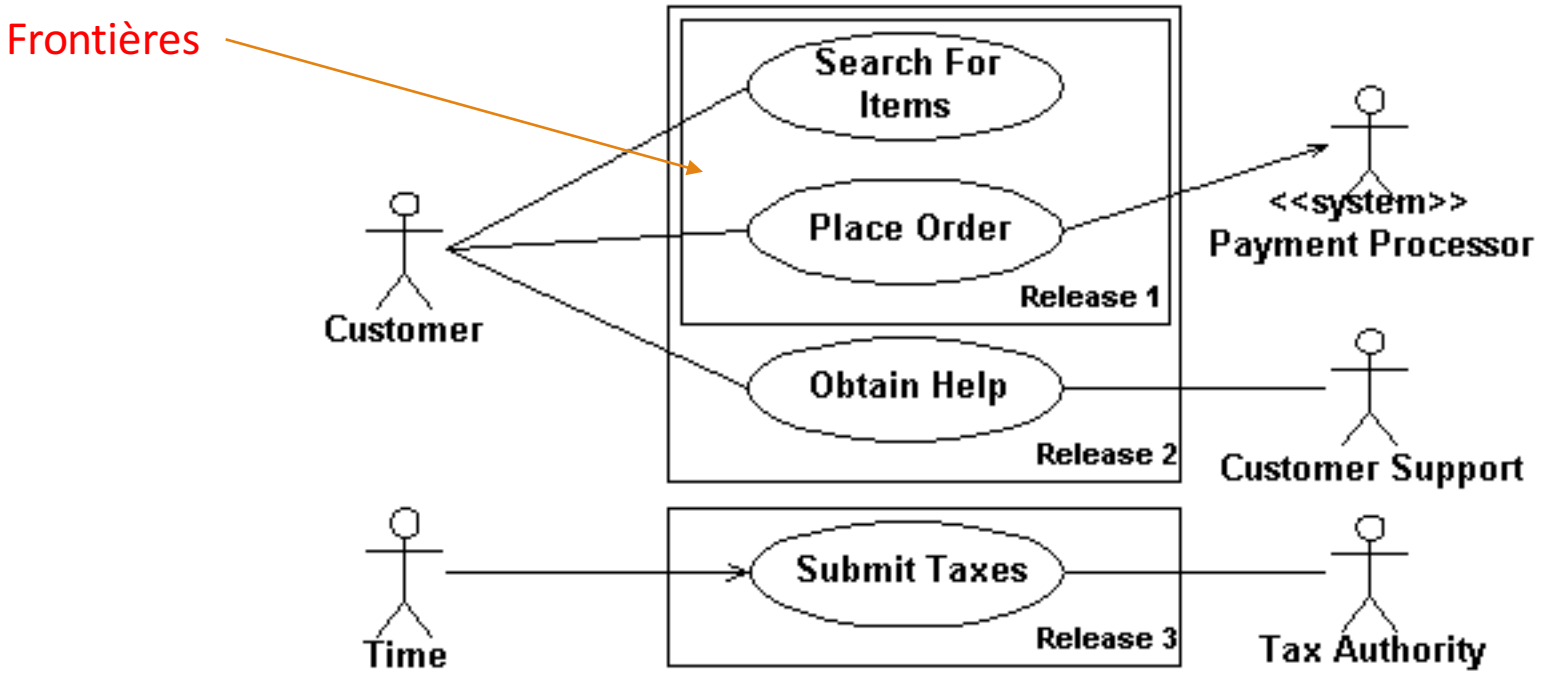
Acteur

Associations



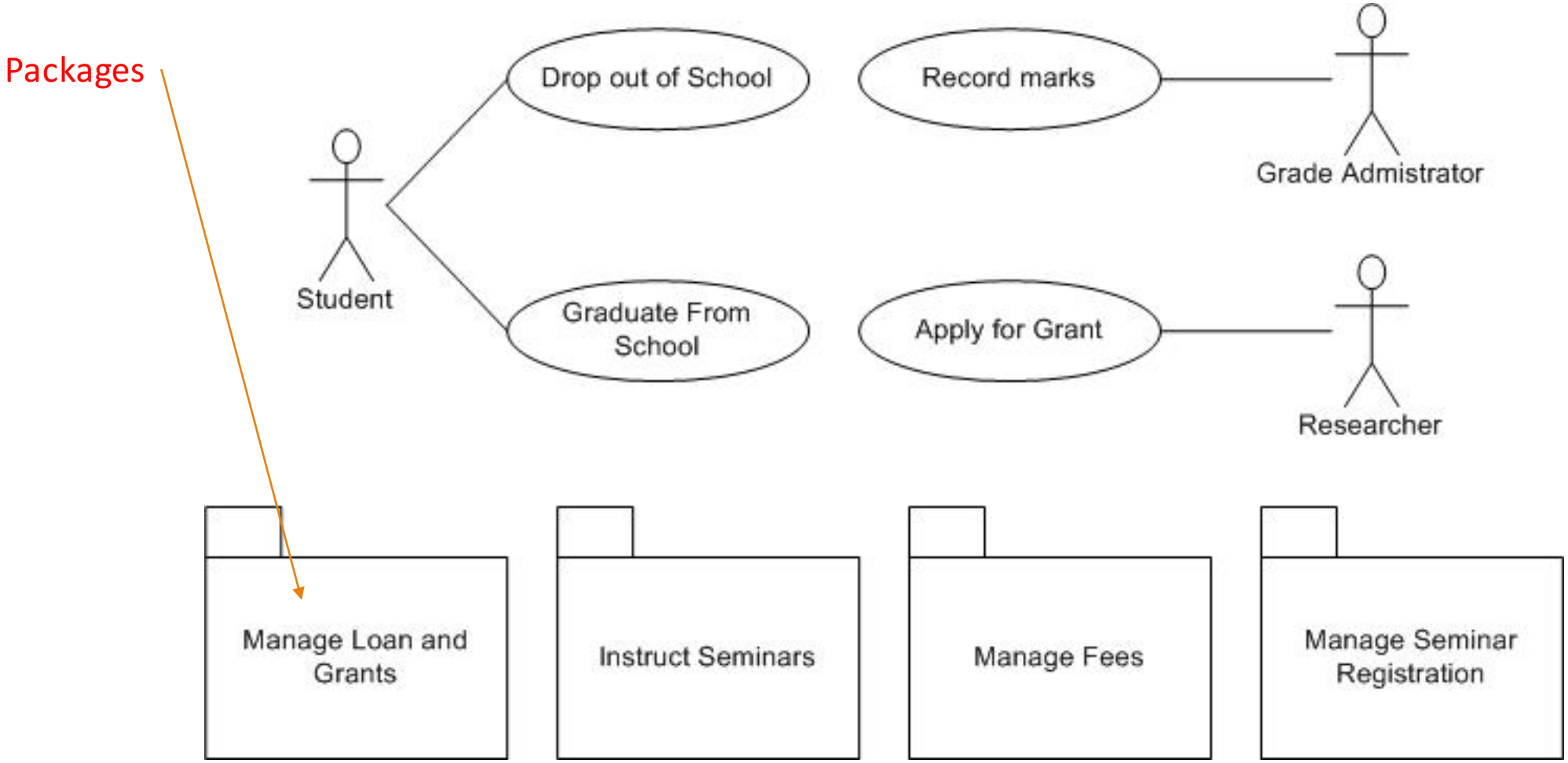
http://www.agilemodeling.com/artifacts/useCaseDiagram.htm

Diagramme de CU – Exemple (2)



http://www.agilemodeling.com/artifacts/useCaseDiagram.htm

Diagramme de CU – Exemple (3)



http://www.agilemodeling.com/artifacts/useCaseDiagram.htm

Qu’est qu’un Acteur?

- Un acteur représente tout ce qui peut réagir avec le système
  - On peut adapter l’icône à la nature de l’acteur

- Décrit le rôle que peut jouer l’utilisateur d’un système
  - Une entité peut jouer plusieurs rôles
    - On peut être à la fois professeur et responsable de département
    - Conseiller et Client dans une banque

Acteurs (2)

- Interagit activement avec le système:
  - Fournir des informations
    - Exemple: Remplir le formulaire d'inscription
  - Recevoir passivement des informations
    - Recevoir un mail de confirmation

- Les acteurs sont externes au système

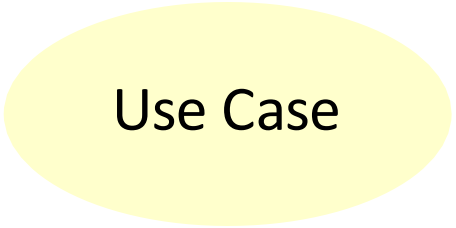
Comment identifier les acteurs

- Un acteur n'est pas forcément utilisateur
  - Oublier un acteur => se tromper sur l'interface
  - Le nom de l'acteur reflète son **rôle**
- Trouver les différents rôles des utilisateurs
  - Responsable clientèle, responsable d'agence, administrateur…
- Identifier les autres systèmes (imprimantes, logiciel)
- Vérifier que les acteurs interagissent avec le système
  - Les clients d'un magasin ne sont pas des acteurs pour le système de la caisse
  - La caissière est l'acteur

Les CUs

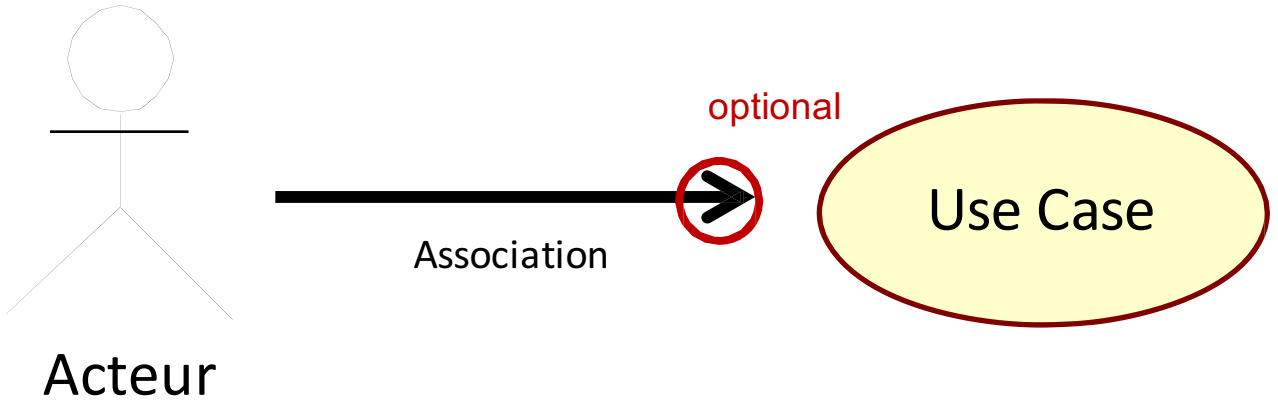
❖ Un ensemble d'instance de CUs

- Chaque instance est une séquence d'actions qu'un système réalise et qui produit un résultat visible par au moins un acteur.
- Un *Use Case* représente un dialogue entre un ou plusieurs acteurs et le système.
- Un *Use Case* décrit les actions prises par le système pour délivrer un résultat à un acteur.



CUs et Acteurs

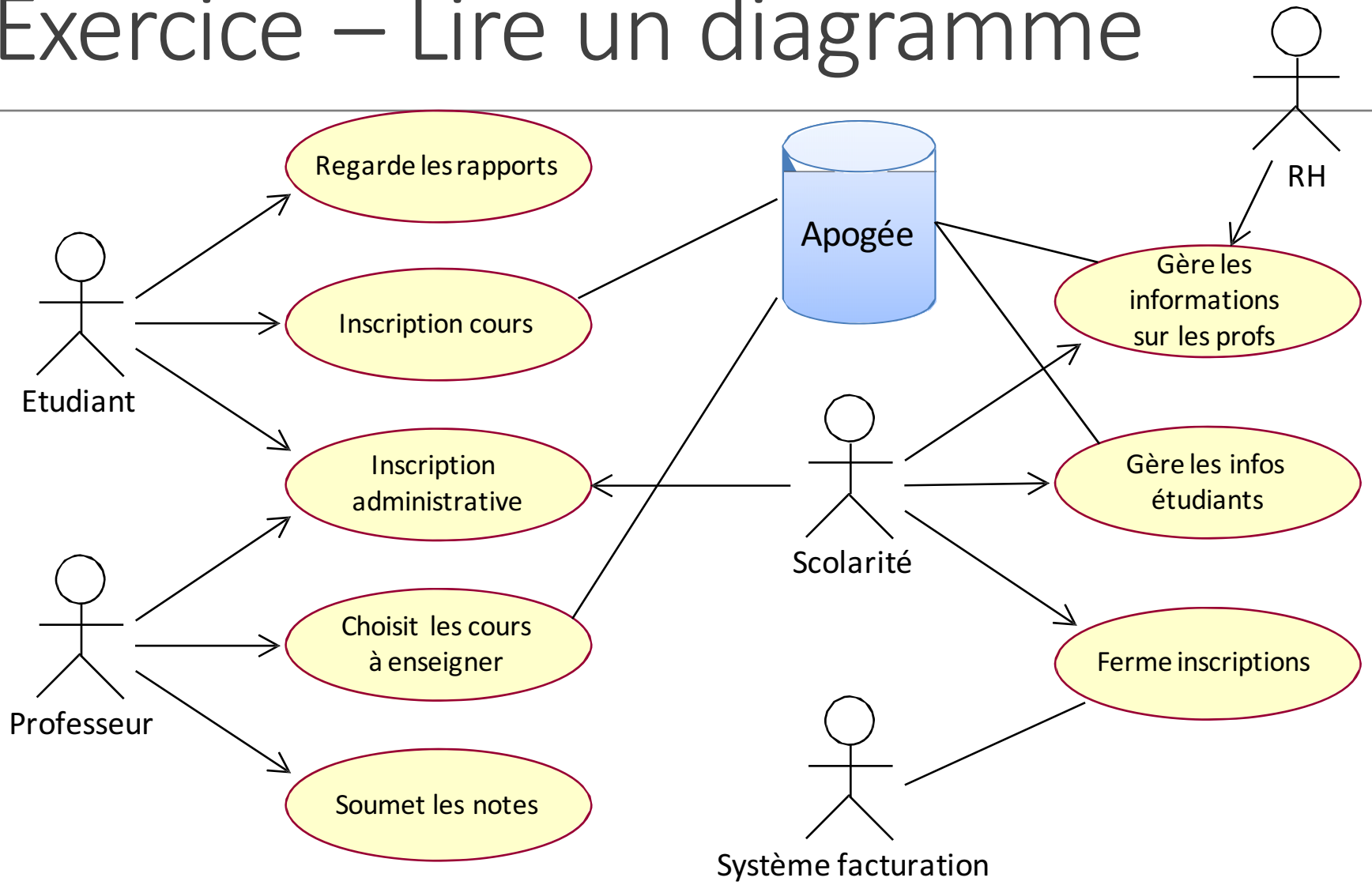
- Un CU représente un dialogue entre un acteur et le système
- Un CU est initié par un acteur



Comment identifier les CUs?

- Description exhaustive des exigences fonctionnelles
  - Se placer du point de vue de chaque acteur
  - Comment et pourquoi il se sert du système
- Bon niveau d'abstraction
  - Ni trop détaillé, ni trop grossier
- Règle de nommage
  - Verbe à l'infinitif + complément
    - Retirer de l'argent
    - Distribuer de l'argent
  - Du point de vue de l'acteur (pas du système)

Exercice – Lire un diagramme



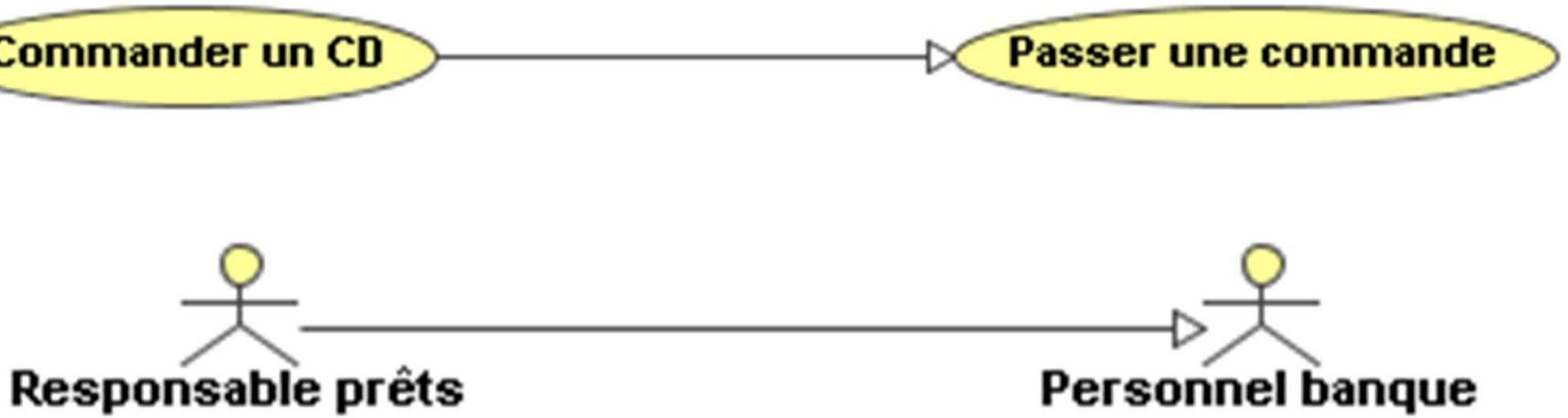
Aucune relation temporelle dans ce type de diagramme !

Exercice (2)

- Lister les acteurs du système
- Lister les cas d'utilisation
- Décrire en quelques mots chaque cas d'utilisation.
- Quel est le rôle de l'Apogée dans ce diagramme?

Généralisation des CUs

- Il est possible de généraliser les acteurs ainsi que les CUs.
- La généralisation se traduit par une flèche orientée partant de l'entité la plus spécifique vers l'entité la plus générique

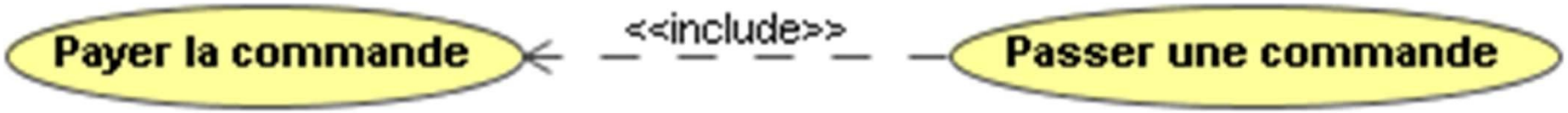


spécifique

générique

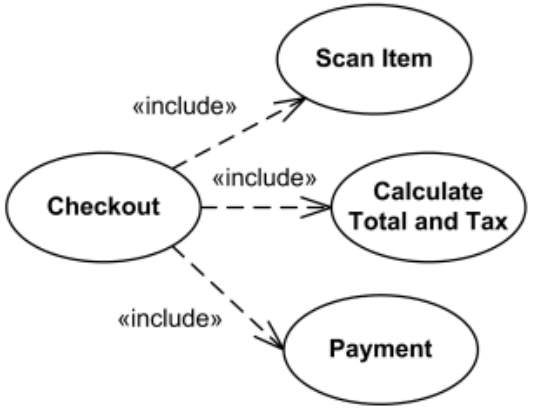
L’inclusion des CUs

- Un CU inclus est une sous fonction obligatoire du CU dans lequel il est inclus
  - On ne peut pas passer de commandes sans payer
  - Permet de décomposer la complexité des CUs
- L’inclusion se traduit par un trait discontinu partant de la sous fonction et orienté vers le CU l’incluant. La flèche est labélisée par le mot clef <<include>> traduisant l’inclusion.

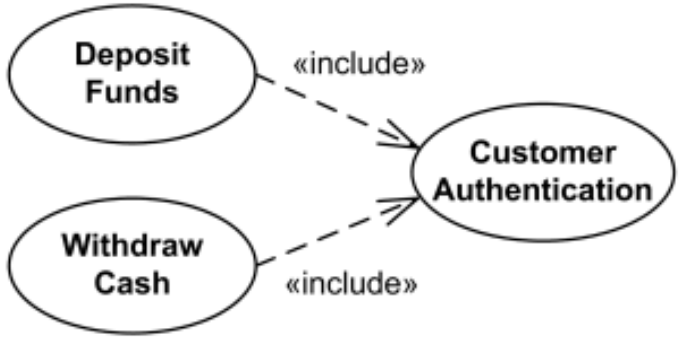


L’inclusion des Cus (2)

- Un CU peut inclure un ou plusieurs CUs

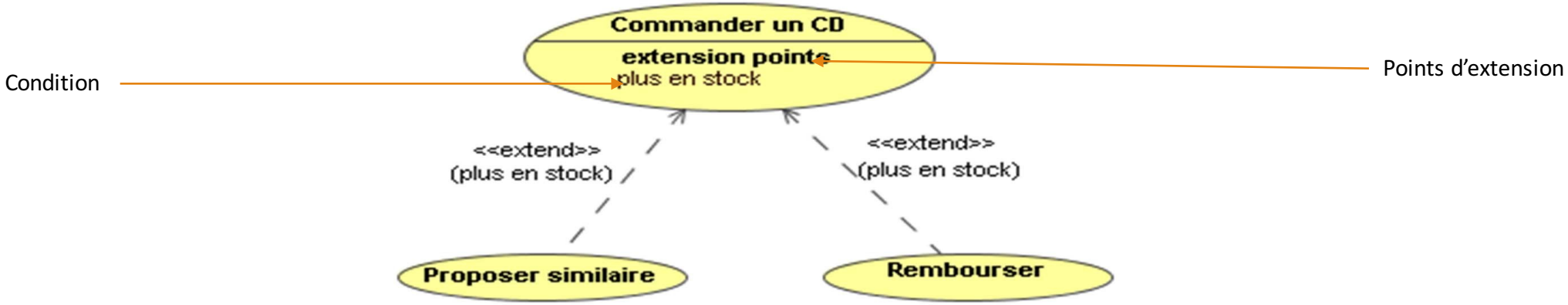


- Plusieurs CUs peuvent inclure un même CU



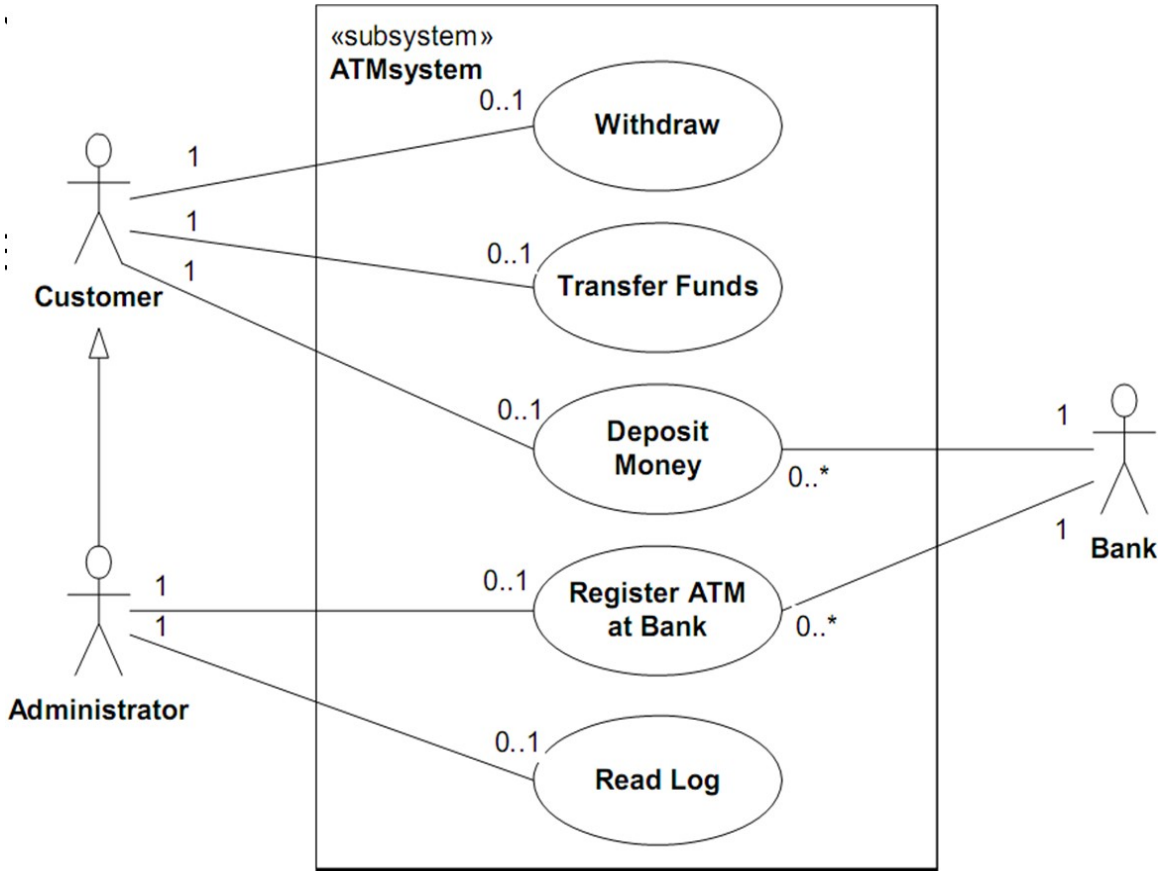
L'extension des CUs

- Les CUs peuvent définir des points d'extension optionnels
  - Un *Use Case* peut être exécuté sans que ses points d'extensions ne soient réalisés.
  - Un Use Case peut avoir plusieurs extensions.
  - Plusieurs *Use Case* peuvent étendre un point d'extension donné.
- Les extensions sont traduits par une ligne discontinu partant du CU à étendre vers le CU sur lequel il est étendu. La ligne discontinu est labélisée par le mot clef « extend ».



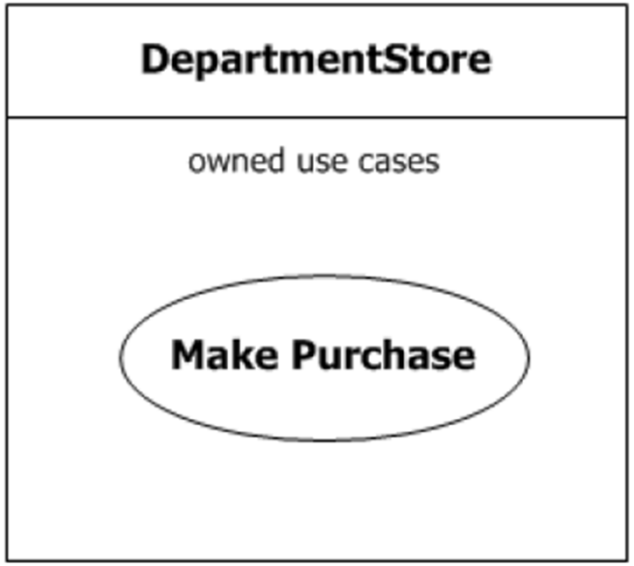
Systèmes et Sous Systèmes

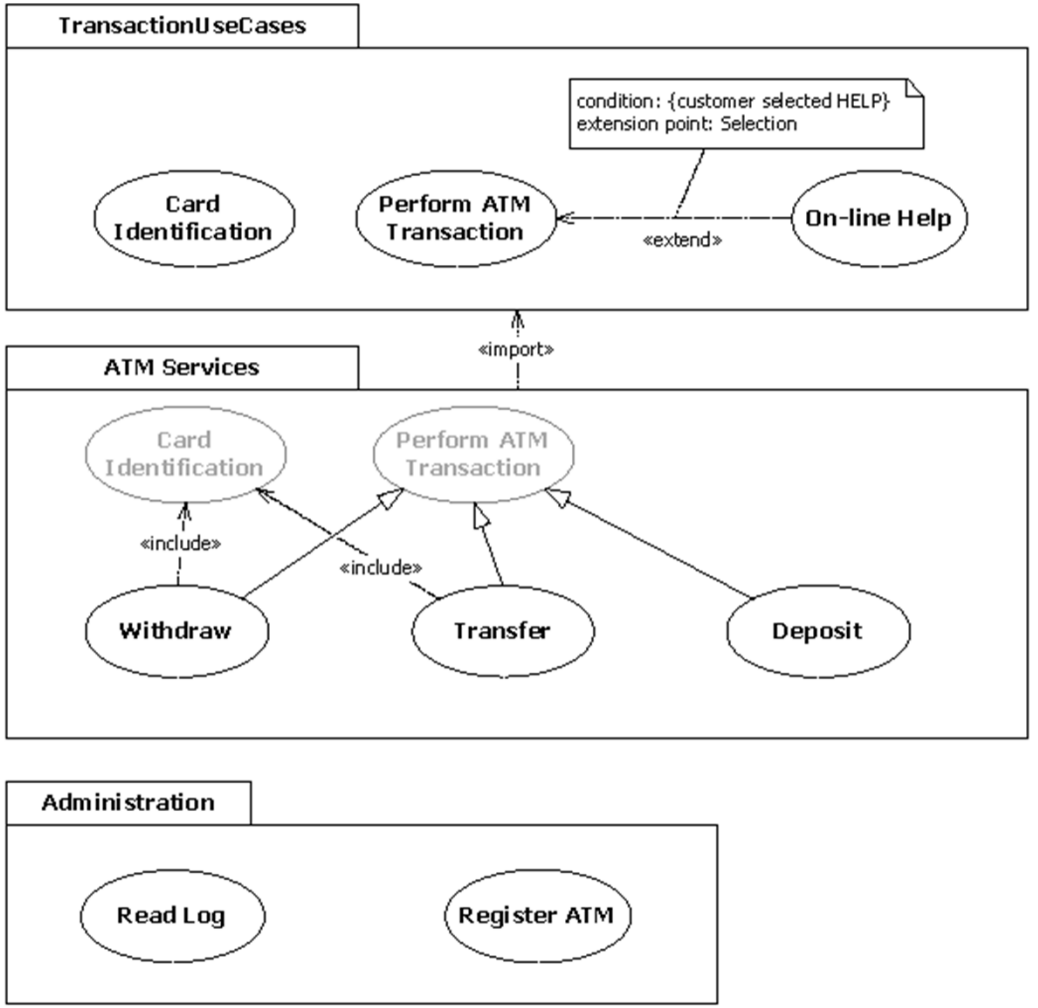
- Un diagramme de CU permet aussi de définir:
  - Les frontières du système: Ce qu'il contient et ce qu'il ne contient pas
  - Le qui fait quoi



Paquetage

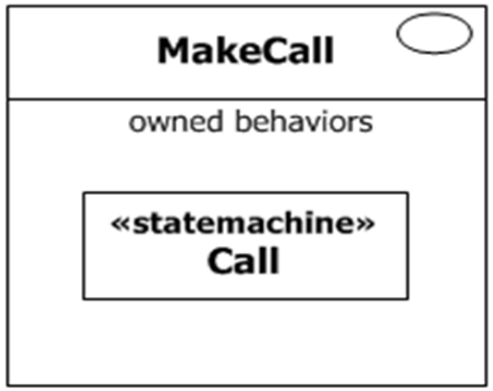
- Permet de regrouper les CUs

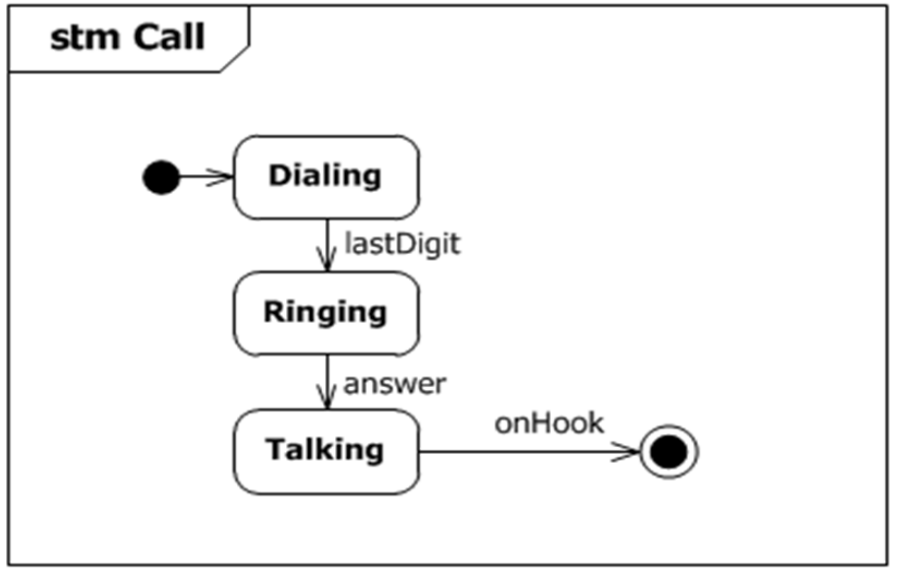




Comportement des CUs

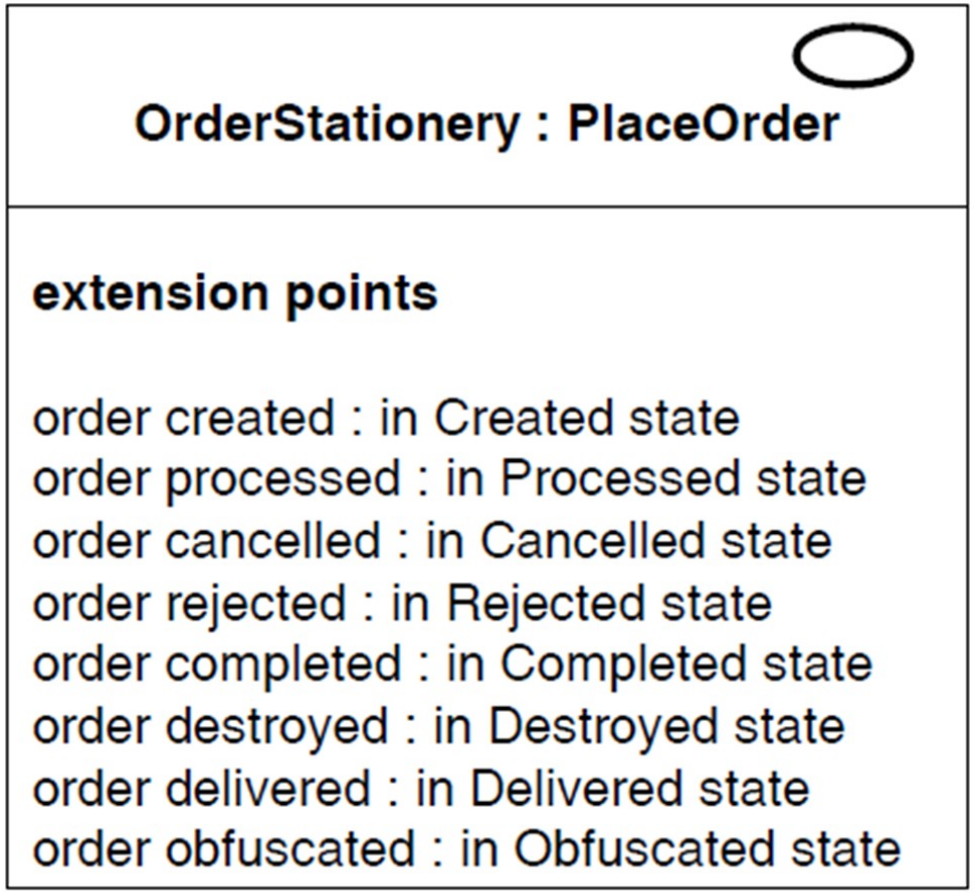
- Il est aussi important de modéliser le comportement des CUs
  - Machines à états
  - Diagrammes d'activités ou d'interaction
- Représente l'enchainement des actions dans la réalisation d'un CU





Points d’extension et états

- Les points d’extension sont définis pour un état donné



Description textuelle (1/2)

- Première partie: générale
  - Nom: verbe infinitif + complément
  - Objectif:
    - Description résumée
    - Renseignée au début du projet
  - Acteurs principaux: « primary »
    - Ceux qui réalisent le cas d'utilisation
  - Acteurs secondaires: « secondary »
    - Ceux qui ne font que recevoir de l'information
  - Dates et version: création et mises à jour
  - Responsables

Description textuelle (2/2)

- Deuxième partie: fonctionnement nominal et dégradé
  - Préconditions: état du système pour déclencher
  - Ensemble de scénarios: (cf. diagrammes d'interactions)
    - Séquence d'échanges
    - Scénario nominal (sans erreurs)
    - Autres scénarios potentiels
  - Postconditions:
    - L'état du système à l'issue des différents scénarios
- Troisième partie
  - Spécifications non fonctionnelles (coût, durée des actions)
  - Interface graphique (maquette de l'IHM pour chaque use case)