

UML 2 – Diagramme de cas d'utilisation & Notions transversales d'UML

Laurent Audibert

Institut Universitaire de Technologie de Villeteuse
Département Informatique

8 février 2011

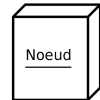
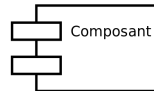
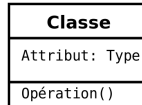
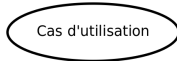
- 1 Notions transversales du langage UML
- 2 Éléments des diagrammes de cas d'utilisation
- 3 Relations dans les diagrammes de cas d'utilisation
- 4 Modélisation des besoins avec UML

- 1 Notions transversales du langage UML
 - Classeur
 - Stéréotype
 - Note
 - Paquetage
 - Espace de noms
- 2 Éléments des diagrammes de cas d'utilisation
- 3 Relations dans les diagrammes de cas d'utilisation
- 4 Modélisation des besoins avec UML

Classeur



Acteur

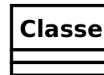


Classeur

Élément de modèle doté d'une identité, pouvant être instancié, et qui possède des caractéristiques structurales et comportementales

- **Représentation générique** : rectangle, en trait plein, contenant éventuellement des compartiments
- Un classeur peut revêtir différentes formes graphiques
- **Exemple de classeur** : acteurs, cas d'utilisation, **classe**, interface, signal, nœud, composant...

Stéréotype – Définition

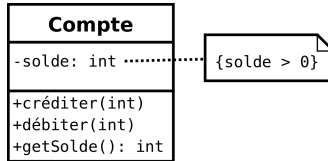


Stéréotype

Un stéréotype est une annotation s'appliquant sur un élément de modèle pour mieux caractériser des variétés d'un même concept

- **Représentation** : chaînes de caractères entre guillemets (`<< >>`)

Note

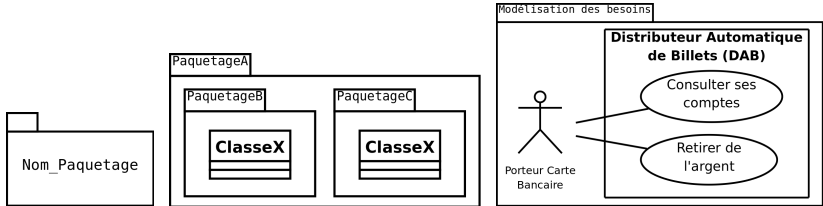


Note

Contient une information textuelle comme un commentaire, une contrainte...

- **Représentation** : rectangle dont l'angle supérieur droit est plié et relié à l'élément décrit grâce à une ligne en pointillés
- Le texte n'est pas contraint par UML

Paquetage



Paquetage

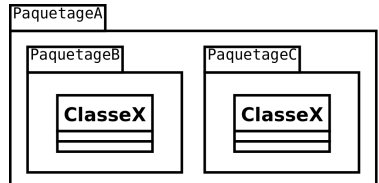
Regroupement d'éléments de modèle et de diagrammes

- Permet d'organiser une modélisation UML
- Conseil : la structuration doit respecter la *cohérence* et l'*indépendance*
- Peut contenir tout type d'élément de modèle, y compris des paquetages
- Un paquetage racine détient la totalité des éléments de la modélisation

Espace de noms

Espace de noms

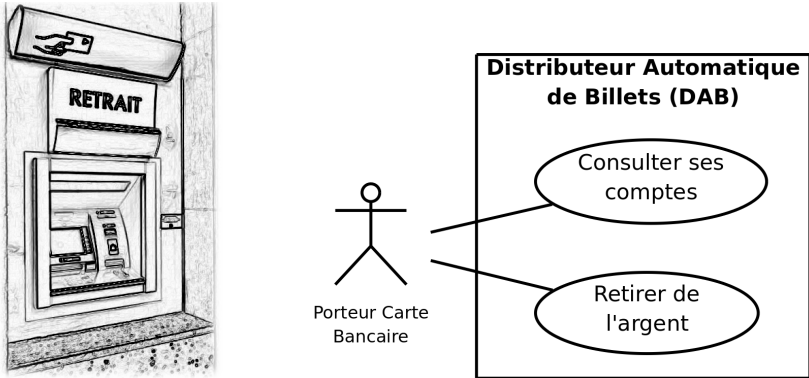
Les paquetages, les classeurs...
sont des espaces de noms



- Tout élément peut être identifié de façon unique par son **nom qualifié**
- **Nom qualifié** : constitué des espaces de noms traversés depuis la racine jusqu'à l'élément en question
- **Exemple** :
 - `PaquetageA::PaquetageB::ClasseX`
 - `PaquetageA::PaquetageC::ClasseX`

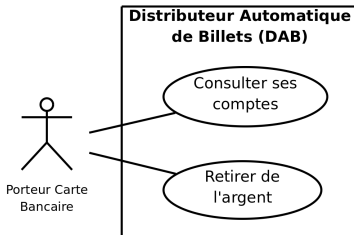
- 1 Notions transversales du langage UML
- 2 Éléments des diagrammes de cas d'utilisation
 - Qu'est-ce qu'un diagramme de cas d'utilisation ?
 - Acteur
 - Cas d'utilisation
 - Représentation d'un diagramme de cas d'utilisation
- 3 Relations dans les diagrammes de cas d'utilisation
- 4 Modélisation des besoins avec UML

Qu'est-ce qu'un diagramme de cas d'utilisation ?



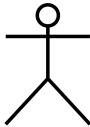
Le diagramme de cas d'utilisation est
la première étape UML d'analyse d'un système.

Qu'est-ce qu'un diagramme de cas d'utilisation ?



- Moyen simple et facilement compréhensible pour exprimer les besoins des utilisateurs
- Permet de recenser les grandes fonctionnalités d'un système
- Ne pas négliger cette étape pour produire un logiciel conforme aux attentes des utilisateurs
- L'élaboration de ce diagramme se fonde sur des entretiens avec les utilisateurs

Acteur



Porteur Carte Bancaire

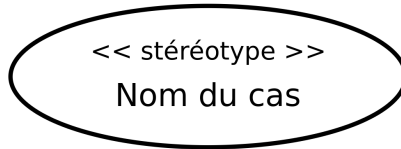
`<<actor>>`

Porteur Carte Bancaire

Acteur

Un acteur est l'idéalisation d'un rôle joué par une personne externe, un processus ou une chose qui interagit avec le système modélisé

Cas d'utilisation

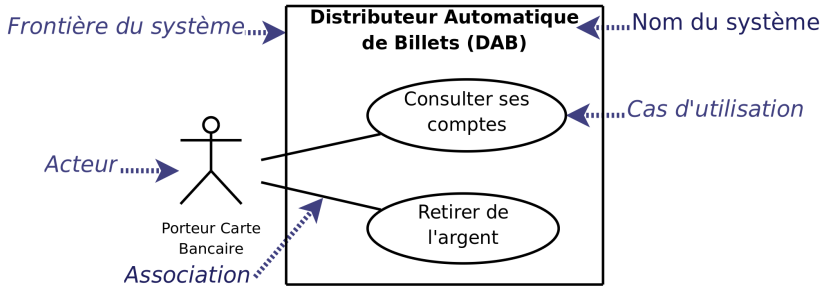


Cas d'utilisation

Un cas d'utilisation est une unité cohérente représentant une fonctionnalité visible de l'extérieur

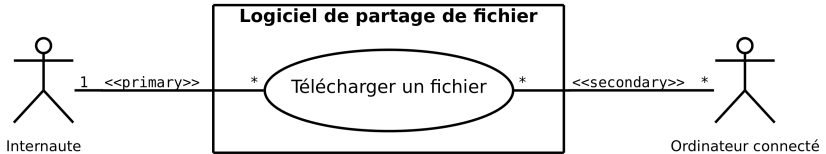
- Un cas d'utilisation réalise un service de bout en bout, avec un déclenchement, un déroulement et une fin
- Il ne dit rien sur la manière de réaliser ce service

Représentation d'un diagramme de cas d'utilisation



- 1 Notions transversales du langage UML
- 2 Éléments des diagrammes de cas d'utilisation
- 3 Relations dans les diagrammes de cas d'utilisation**
 - Relations entre acteurs et cas d'utilisation
 - Relations entre cas d'utilisation
 - Relations entre acteurs
 - Exemple de diagramme de cas d'utilisation
- 4 Modélisation des besoins avec UML

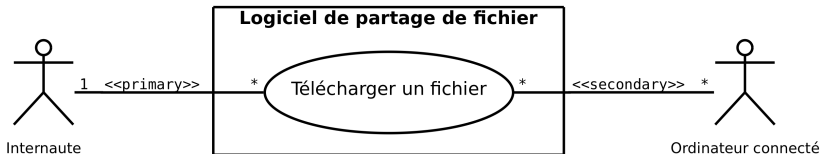
Relation d'association



Relation d'association

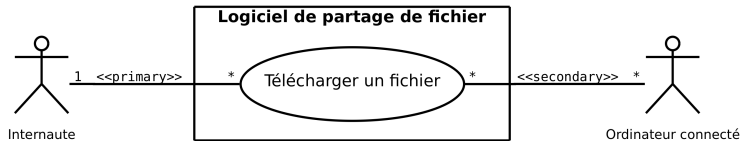
Chemin de communication entre un acteur et un cas d'utilisation, représenté par un trait continu

Multiplicité



- Utilisation :
 - lorsqu'un acteur peut interagir plusieurs fois avec un cas d'utilisation
 - lorsque plusieurs acteurs peuvent interagir avec le même cas d'utilisation
- * signifie *plusieurs* fois
- *n* exactement *n* fois
- *n..m* signifie entre *n* et *m* fois

Acteurs principaux et secondaires



Acteur principal

Celui à qui le cas d'utilisation rend service (stéréotype << *primary* >>)

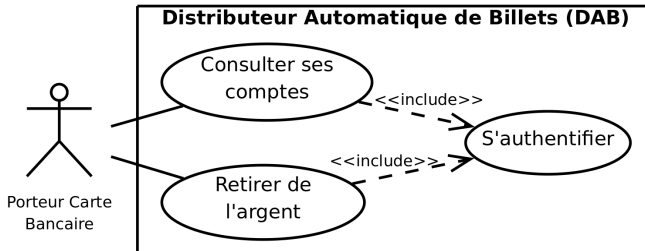
- Un cas d'utilisation a au plus un acteur principal
- Un acteur principal obtient un résultat observable du système
- En général, l'acteur principal initie le cas d'utilisation par ses sollicitations

Acteur secondaire

Tous les autres acteurs (stéréotype << *secondary* >>)

- Un acteur secondaire est sollicité pour des informations complémentaires

Cas d'utilisation interne



Cas d'utilisation interne

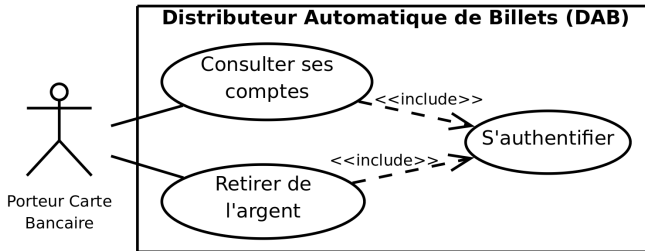
Cas non directement relié à un acteur

Relations entre cas d'utilisation : présentation

Il existe principalement deux types de relations :

- ① les dépendances stéréotypées, qui sont explicitées par un stéréotype. Les plus utilisés sont
 - l'**inclusion**
 - et l'**extension**
- ② la **généralisation/spécialisation**

Relation d'inclusion : définition



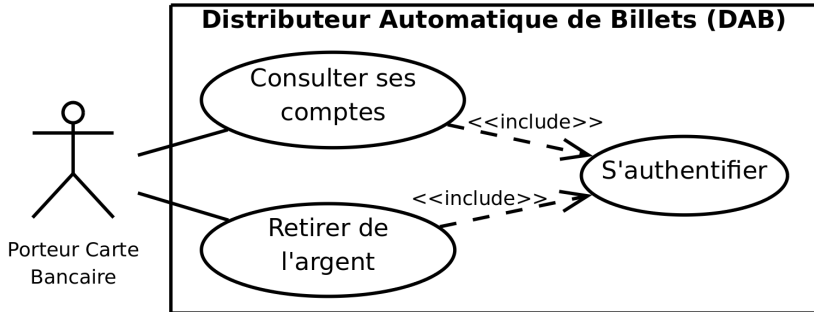
Relation d'inclusion

Un cas A inclut un cas B si le comportement décrit par A inclut le comportement de B : le cas A dépend de B

- Représentation : flèche avec un trait pointillé stéréotypée `<< include >>`, si A inclut B, flèche dirigée de A vers B

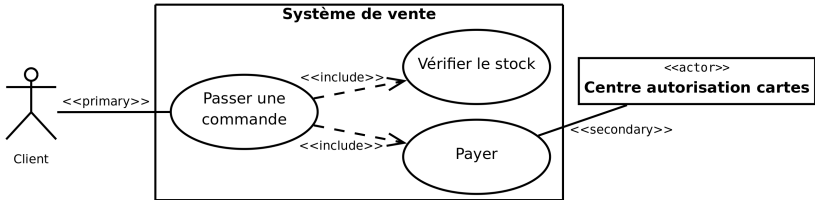
- Lorsque A est sollicité, B l'est obligatoirement

Relation d'inclusion : factorisation



- Factoriser une partie commune à plusieurs cas d'utilisation

Relation d'inclusion : décomposition



- Décomposer un cas complexe en sous-cas plus simples

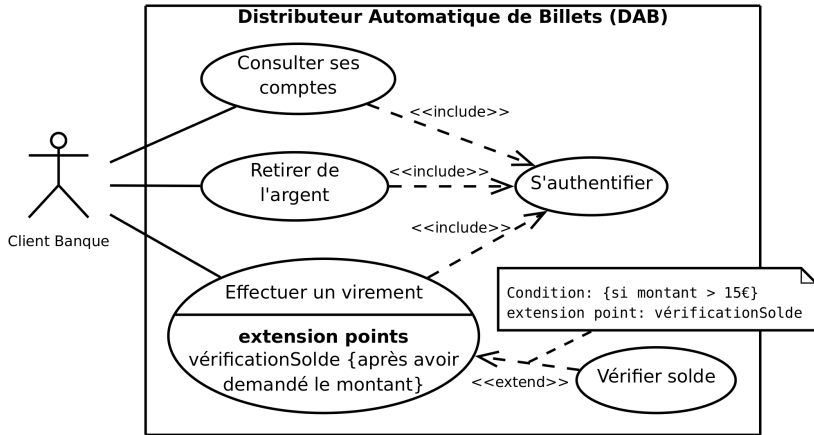
Relation d'extension : définition

Relation d'extension

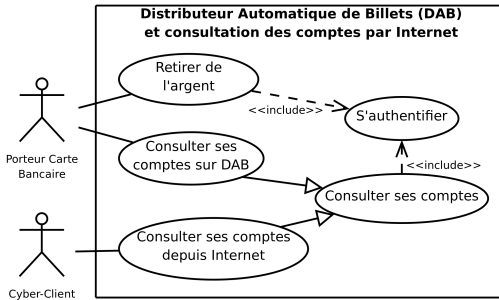
Un cas d'utilisation A étend un cas d'utilisation B lorsque A peut être appelé au cours de l'exécution de B

- Représentation : flèche avec un trait pointillé stéréotypée « *extend* », si A étend B, flèche dirigée de A vers B
- Exécuter B peut **éventuellement** entraîner l'exécution de A
- **La relation la plus utile** car elle a un sens du point de vue métier (les autres sont des artifices d'informaticiens)

Relation d'extension : exemple



Relation de généralisation

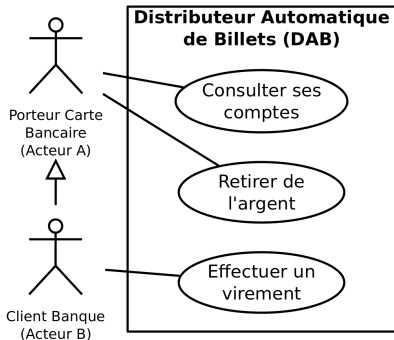


Relation de généralisation

A est une généralisation
d'un cas B si B est un
cas particulier de A

- Représentation : flèche avec un trait plein dont la pointe est un triangle fermé désignant le cas le plus général
- Cette relation se traduit par le concept d'héritage dans les langages orientés objet

Relations entre acteurs



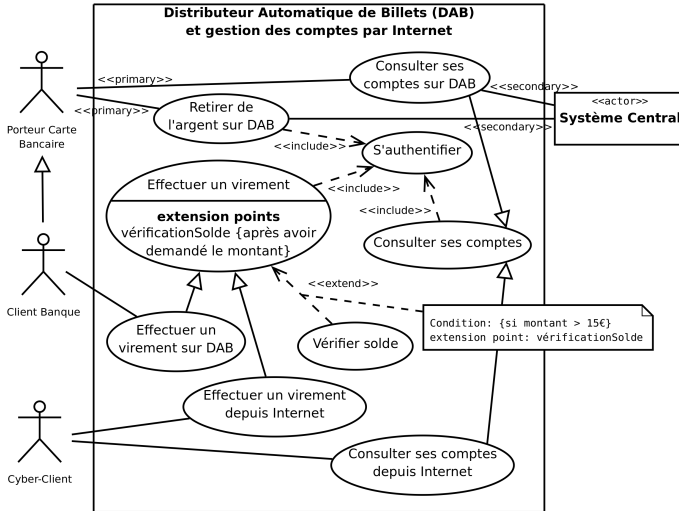
La seule relation possible entre deux acteurs est la généralisation

Relations entre acteurs

Un acteur A est une généralisation d'un acteur B si l'acteur A peut être substitué par l'acteur B

- Représentation : flèche avec un trait plein dont la pointe est un triangle fermé désignant l'acteur le plus général
- Si A généralise B, tous les cas d'utilisation accessibles à A le sont aussi à B

Exemple de diagramme de cas d'utilisation



- 1 Notions transversales du langage UML
- 2 Éléments des diagrammes de cas d'utilisation
- 3 Relations dans les diagrammes de cas d'utilisation
- 4 **Modélisation des besoins avec UML**
 - Comment identifier les acteurs ?
 - Comment recenser les cas d'utilisation ?
 - Description textuelle des cas d'utilisation
 - Remarques

Comment identifier les acteurs ?

- Les acteurs d'un système sont les entités externes à ce système qui interagissent avec lui
- Un acteur représente un ensemble cohérent de **rôles** joués vis-à-vis du système
- Chaque acteur doit être nommé, le nom doit refléter le rôle
- Plusieurs utilisateurs peuvent avoir le même rôle, et donc correspondre à un même acteur
- Une même personne physique peut jouer des rôles différents vis-à-vis du système et donc correspondre à plusieurs acteurs
- Exemples : responsable clientèle, responsable d'agence, administrateur, imprimante, logiciel...

Comment recenser les cas d'utilisation ?

- Se placer du point de vue de chaque acteur et déterminer comment il utilise le système
- Ne pas faire apparaître les détails des cas d'utilisation (ne pas réduire un cas à une action)
- Rester au niveau des grandes fonctions du système
- Garder à l'esprit qu'il n'y a pas de notion temporelle dans un diagramme de cas d'utilisation

Il est recommandé de rédiger une description textuelle en trois parties :

- ❶ Identifier le cas : Nom, Objectif, Acteurs principaux, Acteurs secondaires, Dates, Responsable, Version
- ❷ Description du fonctionnement du cas
Les préconditions : État du système avant que ce cas d'utilisation puisse être déclenché
Des scénarios : Décrits sous la forme d'échanges d'évènements entre l'acteur et le système
 - Scénario nominal (quand il n'y a pas d'erreur)
 - Scénarios alternatifs (variantes du scénario nominal)
 - Scénarios d'exception (cas d'erreurs)**Des post-conditions :** État à l'issue des différents scénarios
- ❸ Rubrique optionnelle (interface graphique. . .)

Remarques

- Une utilisation poussée des relations n'est pas primordiale
→ L'effort doit se porter sur l'identification :
 - des acteurs
 - des cas d'utilisation
 - des associations simples entre acteurs et cas d'utilisation
- Les diagrammes de cas d'utilisation ne peuvent être qualifiés de modélisation à proprement parler : de nombreux éléments descriptifs sont en langage naturel
- Ils ne correspondent pas *stricto sensu* à une approche objet : capturer les besoins, les découvrir, les réfuter, les consolider, etc., correspond plus à une approche fonctionnelle classique