



LICENCE 2 INFORMATIQUE AMSI

Introduction au langage UML

Fabien PEUREUX



Le langage UML

Objectifs généraux

- 🎯 UML est un langage de modélisation pour :
- Comprendre et décrire les besoins
 - Spécifier des systèmes simples ou complexes
 - Concevoir et construire des solutions
 - Documenter un système
 - Communiquer entre les membres de l'équipe projet



Le langage UML

Mode d'usage

- ③ UML est un langage graphique (semi-formel) à usage général, quels que soient :
 - Le type de système – logiciel, matériel, entreprise,...
 - Le domaine métier – gestion, ingénierie, télécoms,...
 - Le processus de développement – cascade, cycle en V,...



Le langage UML

Mise en œuvre

- ② UML 2.0 comporte une douzaine de diagrammes représentant autant de vues distinctes pour représenter des concepts particuliers du système d'information.
- ② Ils se répartissent en deux grands groupes :
 - **Diagrammes structurels ou diagrammes statiques (UML Structure)**
 - **Diagrammes comportementaux ou diagrammes dynamiques (UML Behavior)**



Les digrammes UML

Vues statiques (1)

⊙ Diagramme de cas d'utilisation

- Représente la structure des grandes fonctionnalités nécessaires aux utilisateurs du système

⊙ Diagramme de classes

- Représente l'architecture conceptuelle du système

⊙ Diagramme d'objets

- Instancie le diagramme de classes pour illustrer une situation précise



Les digrammes UML

Vues statiques (2)

⊙ Diagramme de composants

- Représente la structure physique du logiciel à développer

⊙ Diagramme d'architecture

- Représente l'architecture logiciel du système

⊙ Diagramme de déploiement

- Représente la disposition physique des différents matériels logiciels



Les digrammes UML

Vues dynamiques

⊙ Diagramme de séquences / collaboration

- Représente la succession chronologique ou spatiale des opérations réalisées par un acteur pour un scénario donné

⊙ Diagramme d'activités

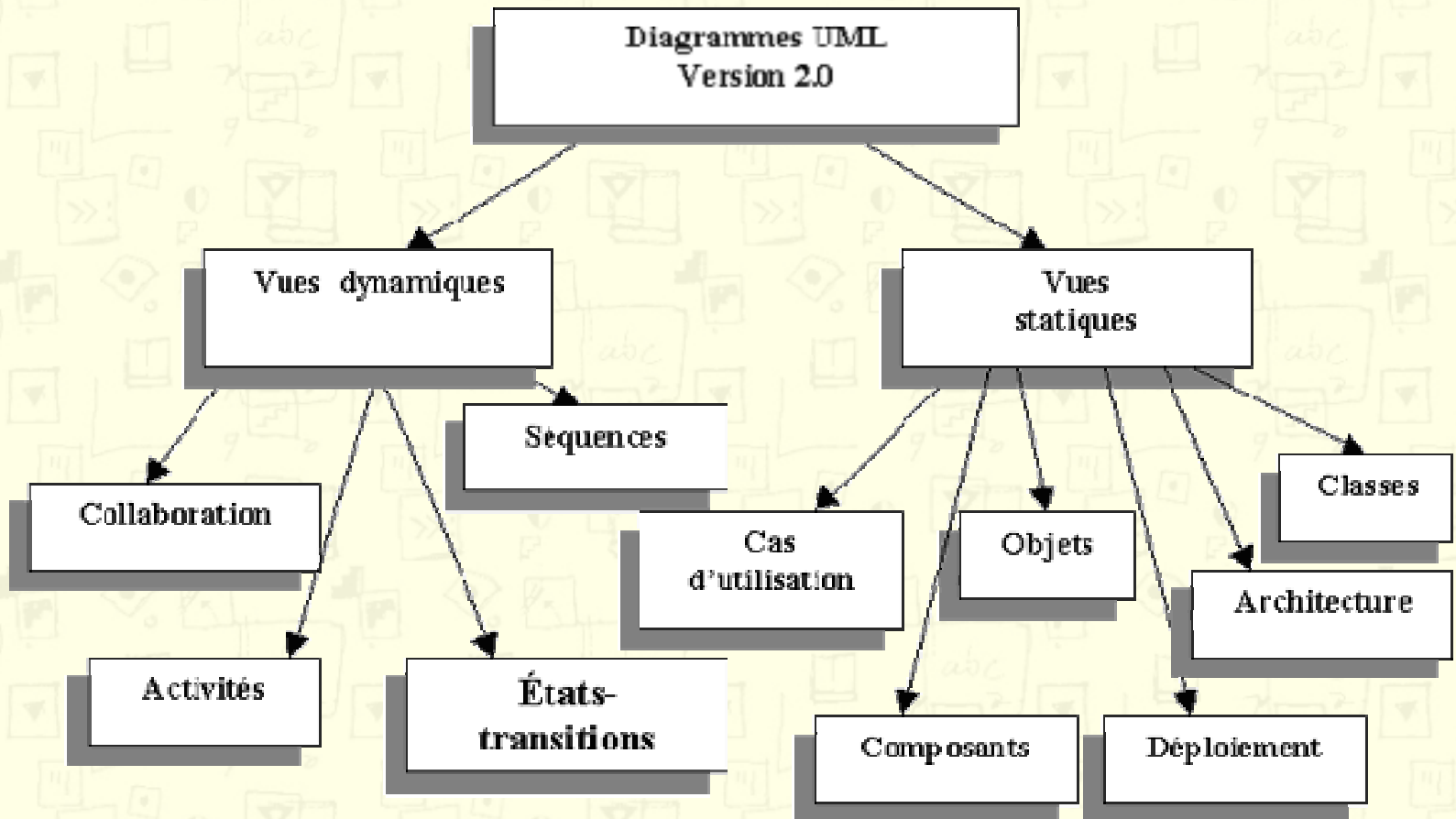
- Représente l'enchaînement des activités qui concourent au processus d'un point de vue fonctionnel

⊙ Diagramme d'états/transitions

- Représente la façon dont évoluent (i.e. cycle de vie) les objets du système

Diagrammes UML

Synthèse





LICENCE 2 INFORMATIQUE AMSI

L'expression des besoins avec UML

Fabien PEUREUX

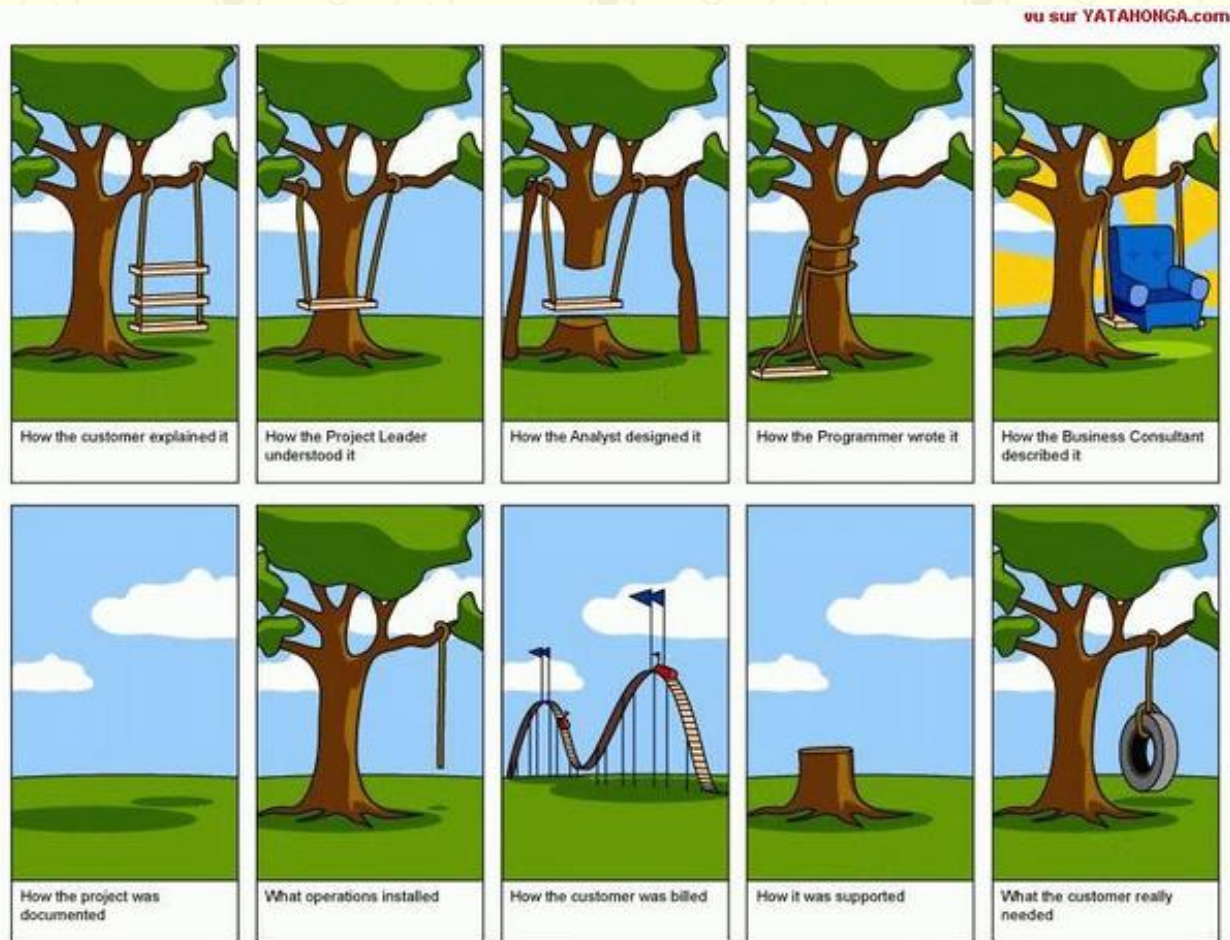


L'expression des besoins Problématique

- ⊙ Des besoins mal spécifiés conduisent inévitablement à des systèmes inutilisables ou inutiles
- ⊙ Une étude récente montre que l'origine de défaut logiciel provient dans 56% des cas à une mauvaise capture des besoins
- ⊙ Parmi les symptômes les plus fréquents sur les projets informatiques ayant échoué :
 - La mauvaise compréhension des besoins des utilisateurs
 - L'incapacité à traiter des exigences évolutives

Les exigences

Différents points de vue...





La gestion des exigences Approche

⊙ La gestion des exigences ne signifie pas :

- Avoir des exigences correctes et définitives dès le démarrage du projet
- C'est une pensée irréaliste du cycle en cascade

⊙ La gestion des exigences signifie :

- Ne pas être négligent
- Les recueillir efficacement
- Enregistrer, tracer et organiser
- Maîtriser les changements



Les exigences

Objectifs généraux (1)

- ① Maintenir un accord avec le client et les autres parties prenantes sur ce que le système doit faire
- ② Fournir à l'équipe projet une meilleure compréhension des exigences système
- ③ Définir les frontières du système (son périmètre)



Les exigences Objectifs généraux (2)

- ② Fournir une base pour planifier le contenu technique des itérations de conception et de développement
- ② Fournir une base pour l'estimation du coût et du temps de développement du système
- ② Définir une interface utilisateur pour le système, en se focalisant sur les besoins et les objectifs des futurs utilisateurs



La gestion des exigences avec UML

⊙ Diagramme de cas d'utilisation (capture des besoins)

- Cerner les objectifs fonctionnels du point de vue des utilisateurs finaux

⊙ Diagramme de séquences (analyse des besoins)

- Identifier et comprendre des scénarii d'utilisation

⊙ Diagramme d'activités (analyse des besoins)

- Comprendre d'un point de vue global les enchaînements fonctionnels des scénarii d'utilisation

Le diagramme UML de cas d'utilisation (Use Case)

Fabien PEUREUX



Diagramme de cas d'utilisation Objectifs

- ① Capturer le comportement désiré du système.
- ① Spécifier ce que le système fait (fonctions), mais pas comment il le fait (implantation)
- ① Créer une entente entre les développeurs, les utilisateurs et les experts métier.
- ① Servir à valider l'architecture et à guider l'évolution du système



Diagramme de cas d'utilisation Description

- ② Formalisation de **Ivar Jacobson**.
- ② Description du comportement d'un système du point de vue de l'utilisateur sous la forme d'actions et de réactions.
- ② Caractérise les fonctions du système déclenchées en réponse à la stimulation d'un acteur externe.
- ② Définition des limites du système et des relations entre le système et l'environnement.



Diagramme de cas d'utilisation Intérêts (1)

- ③ La détermination et la compréhension des besoins sont difficiles.
- ③ Les cas d'utilisation recentrent l'expression des besoins sur les utilisateurs : un système est avant tout construit pour ses utilisateurs.
- ③ Les cas d'utilisation permettent aux utilisateurs de structurer et d'articuler leurs besoins.

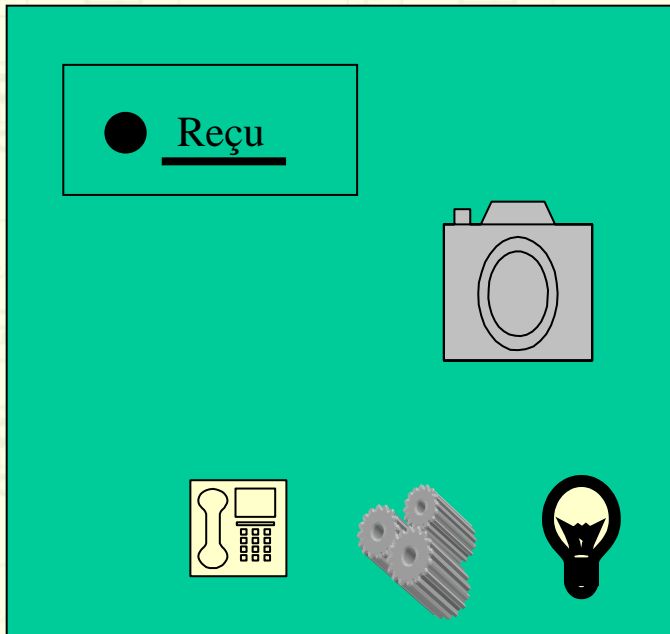


Diagramme de cas d'utilisation Intérêts (2)

- ③ On structure la démarche par rapport aux interactions d'une seule catégorie d'utilisateurs à la fois; réduit la complexité de la détermination des besoins.
- ③ Le formalisme des cas d'utilisation est basé sur le langage naturel; il est accessible sans formation particulière des utilisateurs.
- ③ Ils concrétisent le futur système dans une formalisation proche de l'utilisateur.

Diagramme de cas d'utilisation

Exemple de besoins



🌀 Machine à recycler :

- Recevoir et vérifier les objets introduits par les utilisateurs,
- Imprimer et sortir un reçu pour les objets introduits
- Imprimer la liste de tous les objets reçus pour l'opérateur
- Mettre à jour les données du système
- Déclencher un signal d'alarme en cas de problème.

Diagramme de cas d'utilisation

Concepts

- Représente les acteurs, le système et les cas d'utilisation eux-mêmes

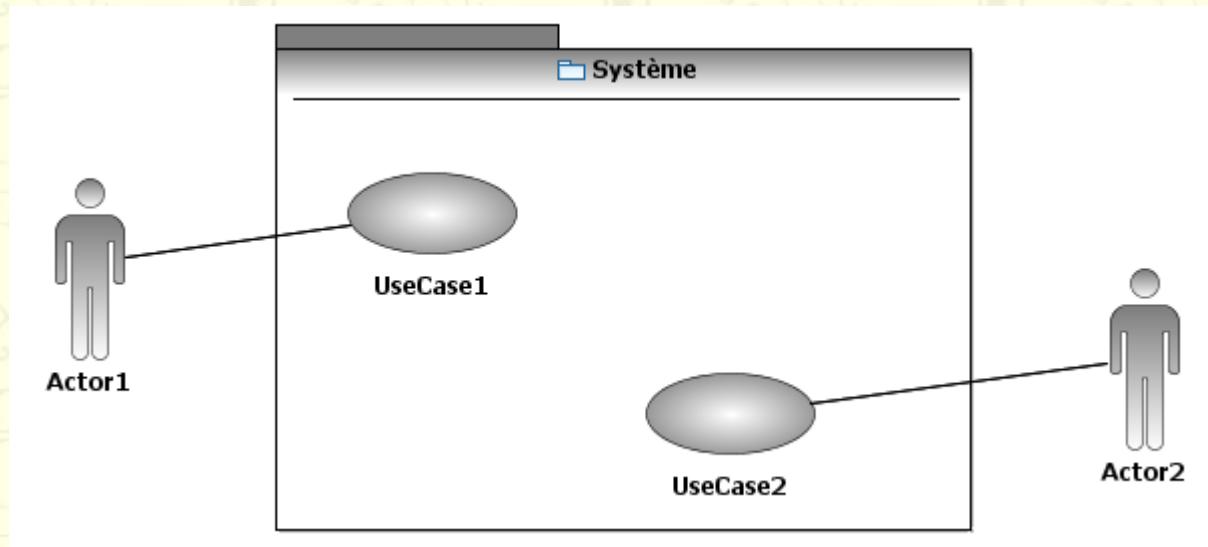


Diagramme de cas d'utilisation

Les acteurs

- 🌀 L'acteur est un agent externe qui utilise ou interagit avec le système
- 🌀 Les acteurs peuvent être :
 - Des rôles joués par des humains
 - D'autres systèmes informatiques





Diagramme de cas d'utilisation

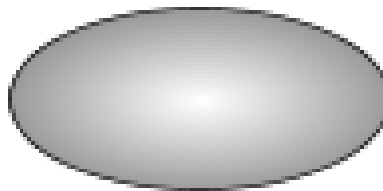
Les acteurs

- Représente tout ce qui est externe au système, humain ou non, qui interagit avec le système et qui correspond à une catégorie d'utilisateurs (plus précisément à un rôle).
- Se détermine en observant les utilisateurs directs du système, ainsi que les autres systèmes qui interagissent avec le système en question.
- Le nom de l'acteur décrit le rôle joué par l'acteur.

Diagramme de cas d'utilisation

Les cas d'utilisation

- ② Modélise un service rendu par le système, sans imposer le mode de réalisation
- ② Est représenté par une ellipse contenant le nom du cas (un verbe à l'infinitif), et optionnellement, au-dessus du nom, un stéréotype



«Stéréotype»
UseCase1



Diagramme de cas d'utilisation Frontière (package)

- ① La frontière du système est représentée par un cadre. Le nom du système figure à l'intérieur du cadre, en haut.
- ① Les acteurs sont à l'extérieur et les cas d'utilisation à l'intérieur.
- ① Le diagramme comprend une collection de cas d'utilisation.
- ① Il caractérise le comportement de l'ensemble du système et des acteurs externes (dans leur interaction).

Diagramme de cas d'utilisation

Exemple 1

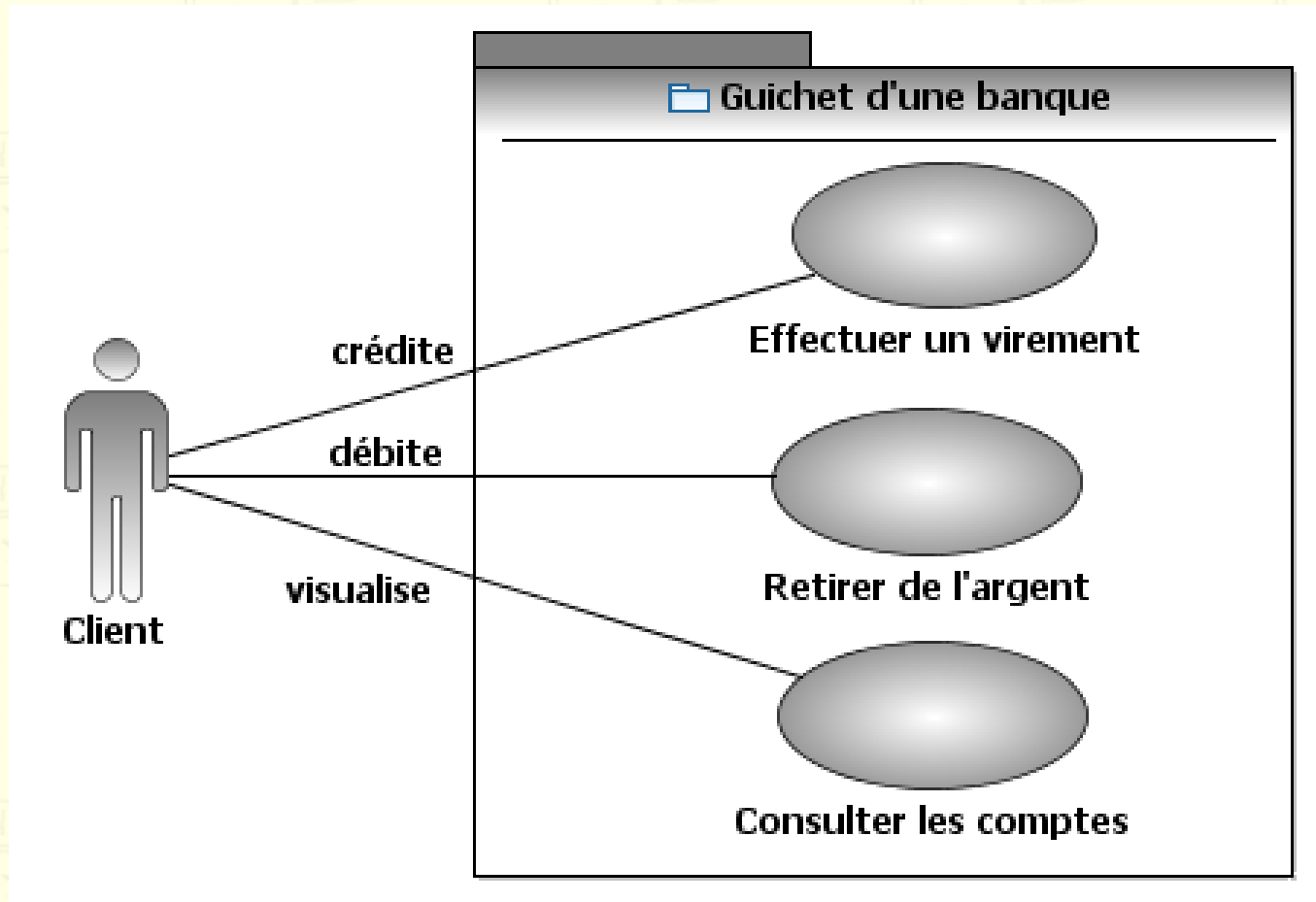
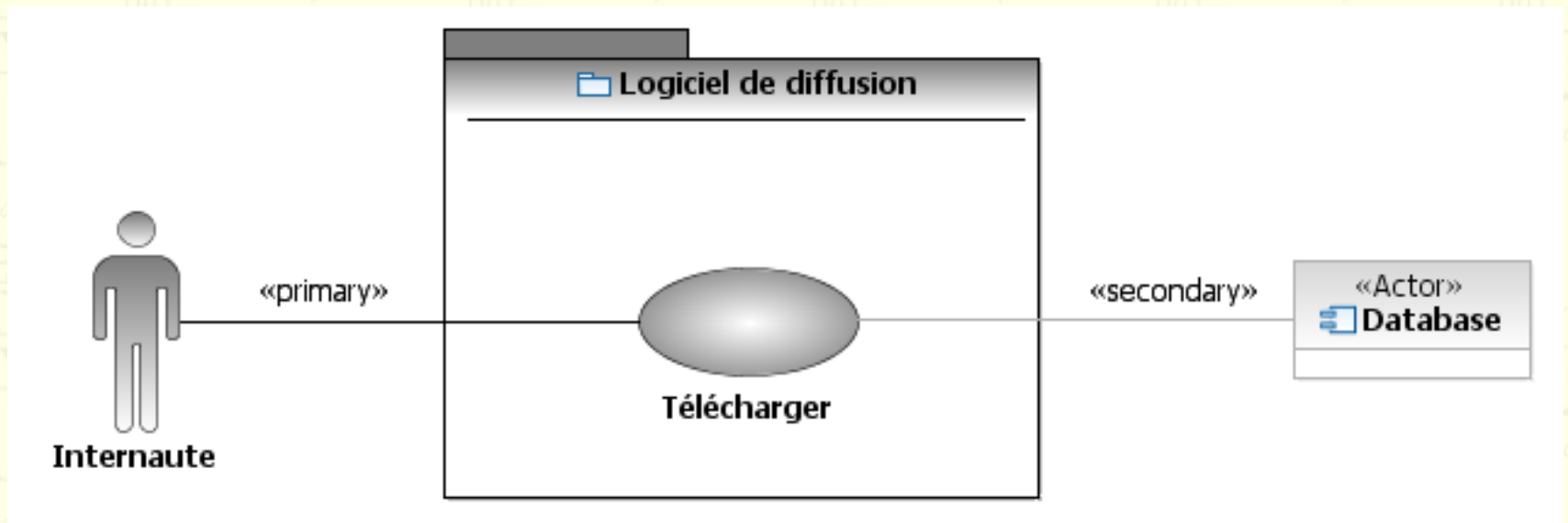


Diagramme de cas d'utilisation

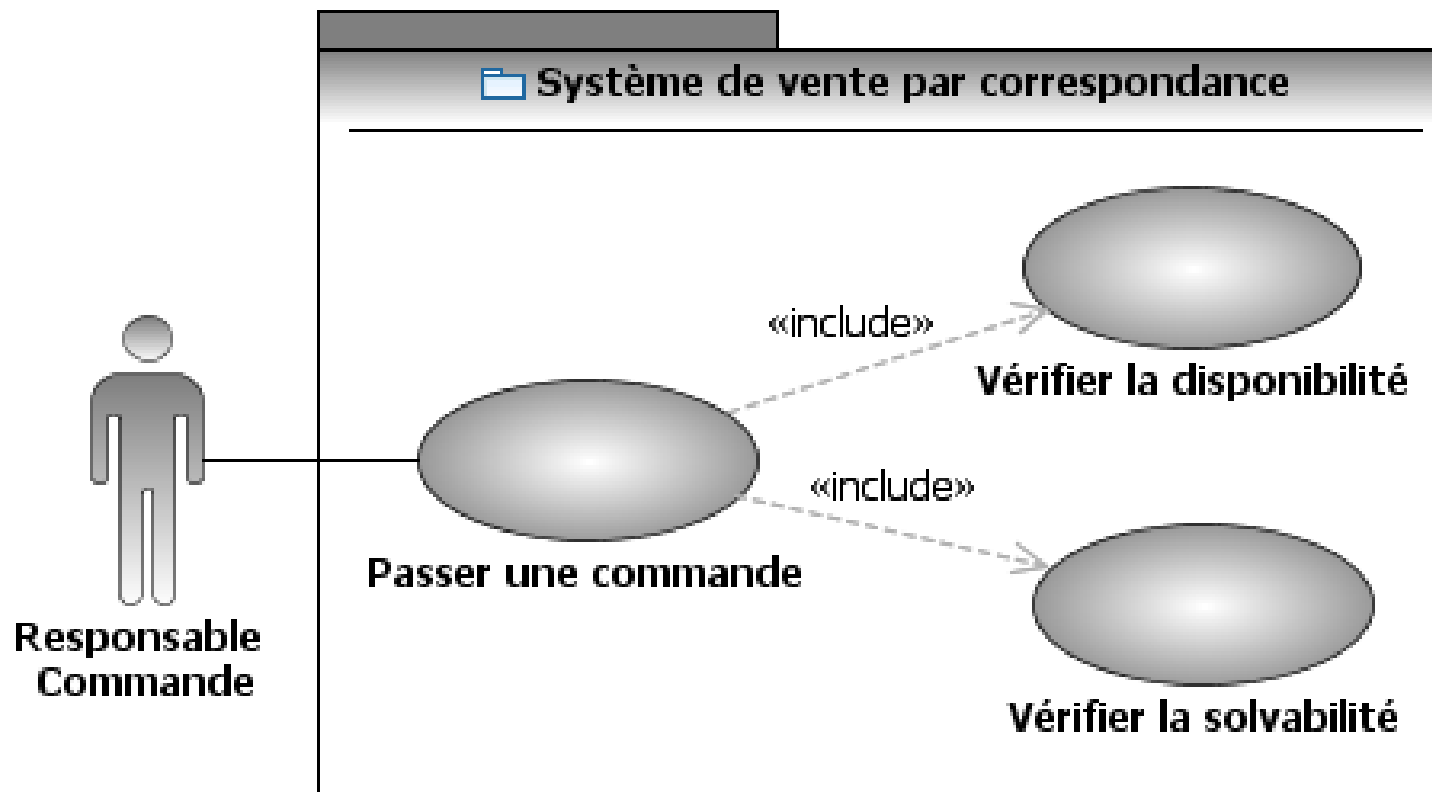
Exemple 2



Raffinement des cas d'utilisation : la relation « include »

- ③ Une relation d'inclusion entre cas d'utilisation signifie qu'une instance du cas d'utilisation source comprend également le comportement décrit par le cas d'utilisation destination.
- ③ Elle est à utiliser lorsqu'un ensemble d'actions peut être utilisé dans plusieurs cas d'utilisation et que l'on ne souhaite pas répéter cet ensemble.
- ③ Un tel ensemble est alors décrit dans un cas d'utilisation séparé et est lié au cas d'utilisation qui l'utilise par un lien <<use>> ou <<include>>.

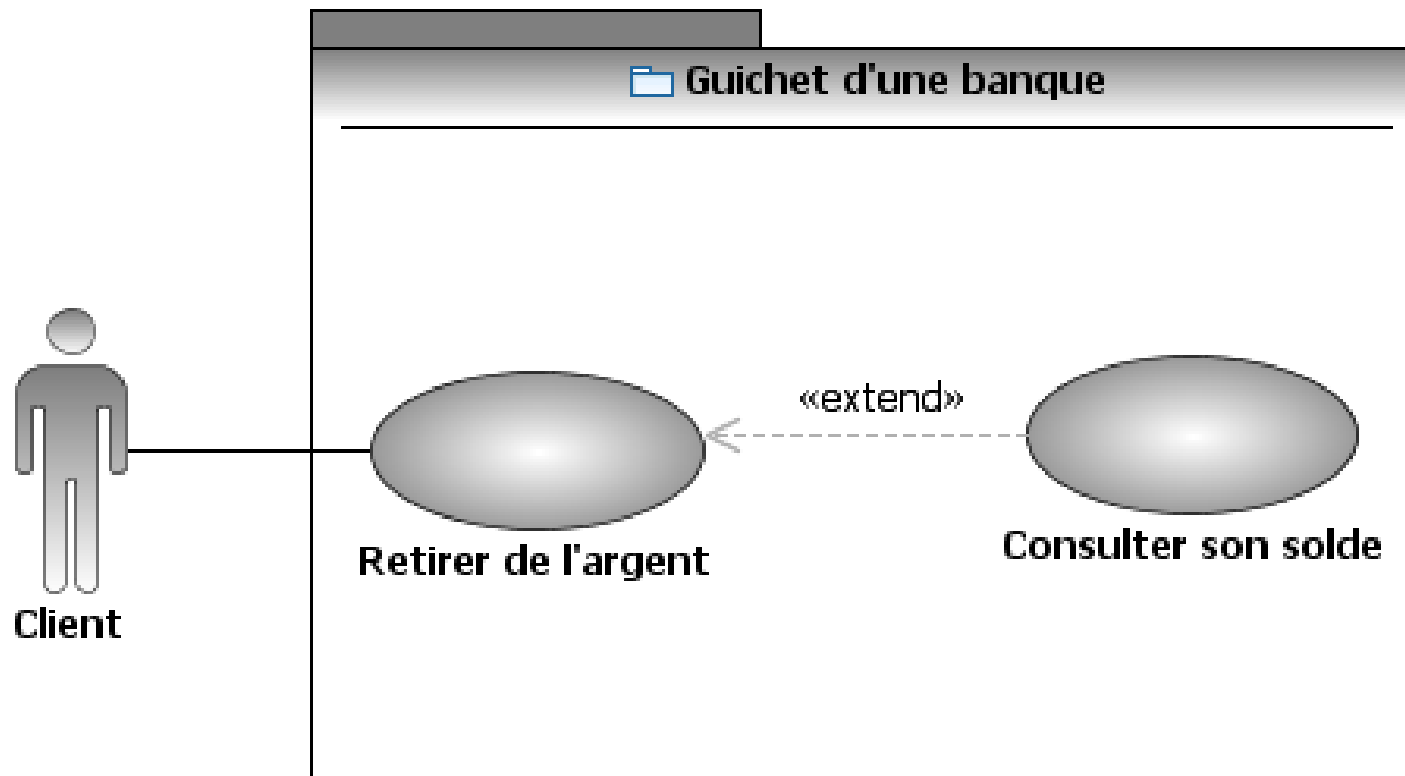
Raffinement des cas d'utilisation : la relation « include »



Raffinement des cas d'utilisation : la relation « extend »

- ② Une relation d'extension entre cas d'utilisation signifie que le cas d'utilisation source peut étendre le comportement du cas d'utilisation destination.
- ② A utiliser lorsqu'un cas d'utilisation est similaire à un autre cas d'utilisation à l'exception d'une petite variation.
- ② Une telle variation est décrite dans un cas d'utilisation à part, les deux cas d'utilisation sont ensuite liés par une relation d'extension.

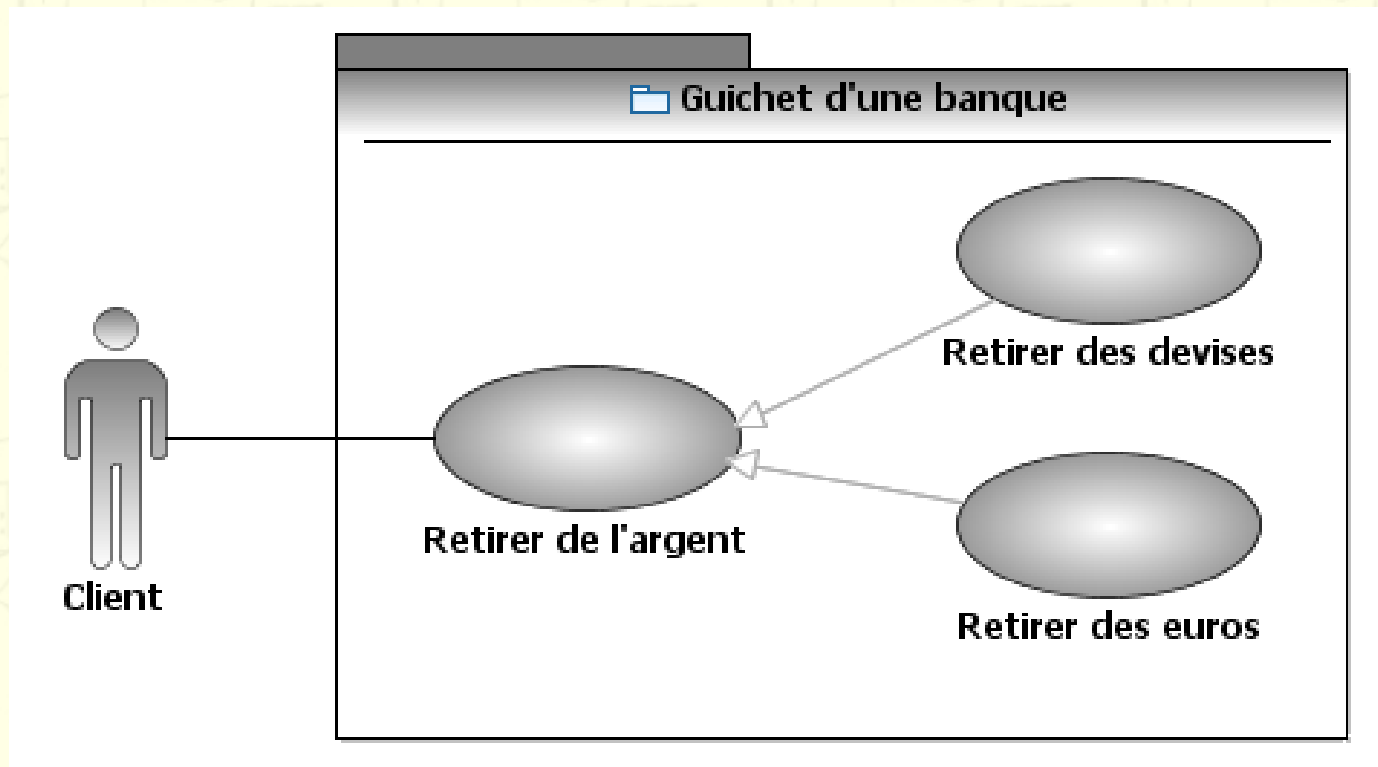
Raffinement des cas d'utilisation : la relation « extend »



Raffinement des cas d'utilisation : la relation de généralisation (cas)

- ④ Les cas d'utilisation descendants héritent de la description de leur parent.
- ④ Les cas d'utilisation descendants définissent un cas concret tandis que le parent définit un cas abstrait.
- ④ Pour une instance du cas d'utilisation parent, un et seul un cas d'utilisation descendant est applicable.

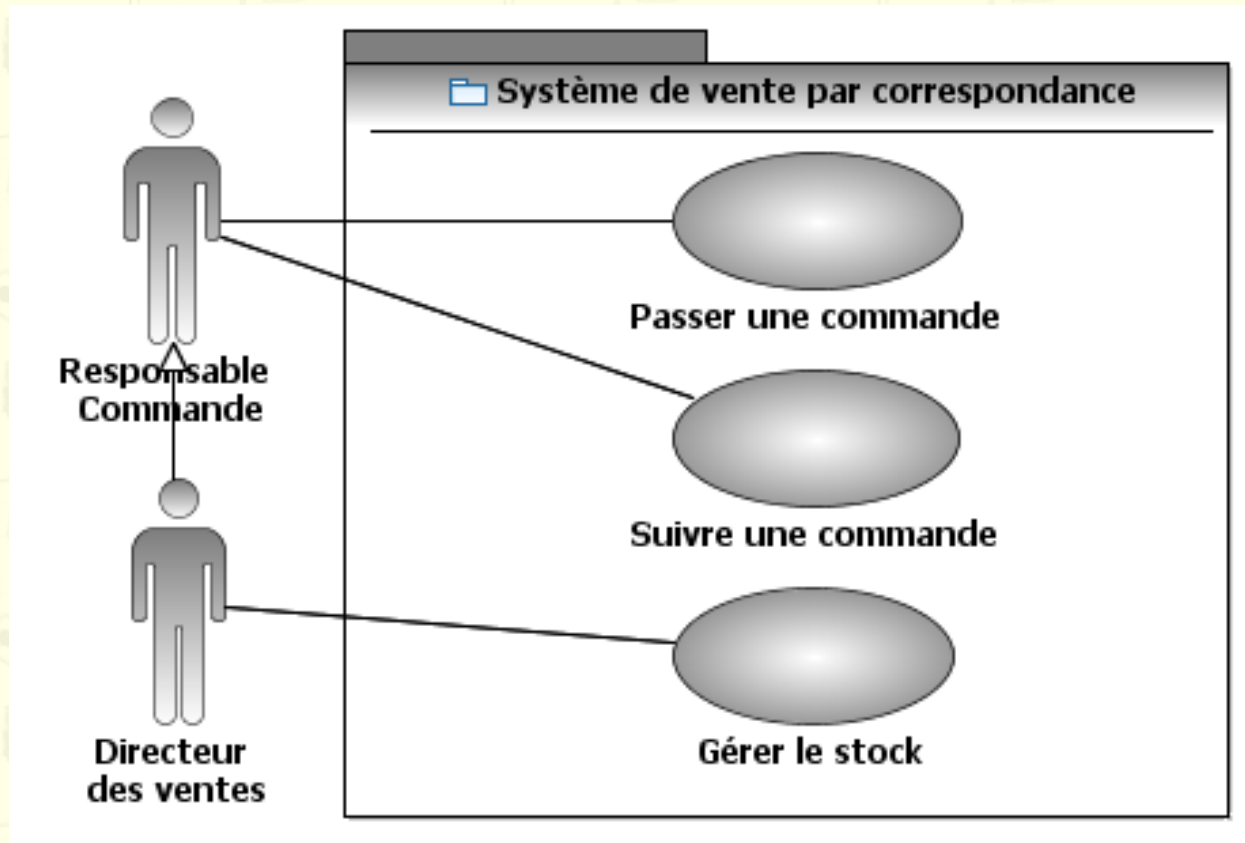
Raffinement des cas d'utilisation : la relation de généralisation (cas)



Raffinement des cas d'utilisation : la relation de généralisation (acteur)

- Seule relation possible entre deux acteurs
- Un acteur A est une généralisation d'un acteur B si l'acteur A peut être substitué par l'acteur B.
- Les cas d'utilisation accessibles à A le sont aussi à B, mais l'inverse n'est pas vrai.

Raffinement des cas d'utilisation : la relation de généralisation (acteur)



Identification des cas d'utilisation

Méthode basée sur les acteurs

- ① Identifier les acteurs liés à un système ou à une organisation.
- ② Pour chaque acteur, identifier les processus qu'il initie ou auxquels il participe.

Identification des cas d'utilisation

Méthode basée sur les évènements

- ① Identifier les évènements externes auxquels un système doit répondre.
- ② Relier ces évènements à des acteurs et à des cas d'utilisation.

Identification des cas d'utilisation

Méthode basée sur les cas

- ① Déterminer dans le cahier des charges les services fonctionnels attendus du système.
- ② Relier ces services aux personnes qui les utilisent.

Synthèse

