

SEANCE 2

ANALYSE DU DOMAIN ET MODÉLISATION DES EXIGENCES

SUJETS

Analyse du domaine

- Modélisation de domaine

Exigences

- Les exigences des clients
- Exigences fonctionnelles
- Exigences non-fonctionnelles
- Cas d'utilisation
- Diagrammes de cas d'utilisation
- Relations entre les cas d'utilisation
- Exemples de cas d'utilisation

Histoires d'utilisateurs des processus agiles

ANALYSE DU DOMAINE

L'analyse de domaine nous permet de comprendre le domaine du problème

On est concerné par le domaine dans lequel les clients s'attendent à utiliser le logiciel

Analyse de domaine: recueillir l'information sur le domaine des experts, de livres, de logiciels existants et de sa documentation ...

- L'objectif est de comprendre le domaine du problème indépendamment du système particulier que nous avons l'intention de développer
- Nous ne cherchons pas à délimiter le système du restant de l'environnement
- Nous nous concentrons sur les concepts et la terminologie du domaine d'application avec une portée plus large que le futur système.

ANALYSE DU DOMAINE

Vous n'êtes **pas censé devenir un expert de domaine**

- Mais vous devez rassembler suffisamment d'informations pour comprendre le problème

Avantages:

- Développement plus rapide: communiquer plus efficacement avec les intervenants
- Une meilleure compréhension des besoins des utilisateurs
- Anticipation des extensions

ANALYSE DE DOMAINE

1. **Glossaire des termes définissant la terminologie et les concepts communs du domaine du problème**
2. **Connaissance générale du domaine (information de base bien connus par les experts)**
3. **Les clients et les utilisateurs (qui achèteront ou achèteront le logiciel)**
4. **Environnement (le nouveau système interagira-t-il avec les systèmes existants?)**
5. **Modèle de domaine**
 - Diagramme de classes UML pour modéliser la relation entre les entités
 - Modèle de relation d'entité
6. **Tâches et procédures actuellement exécutées**
7. **Les applications des compétiteurs**

MODÈLE DE DOMAINE

Il n'existe pas de vue unifiée en ingénierie logicielle concernant ce qu'est un modèle de domaine

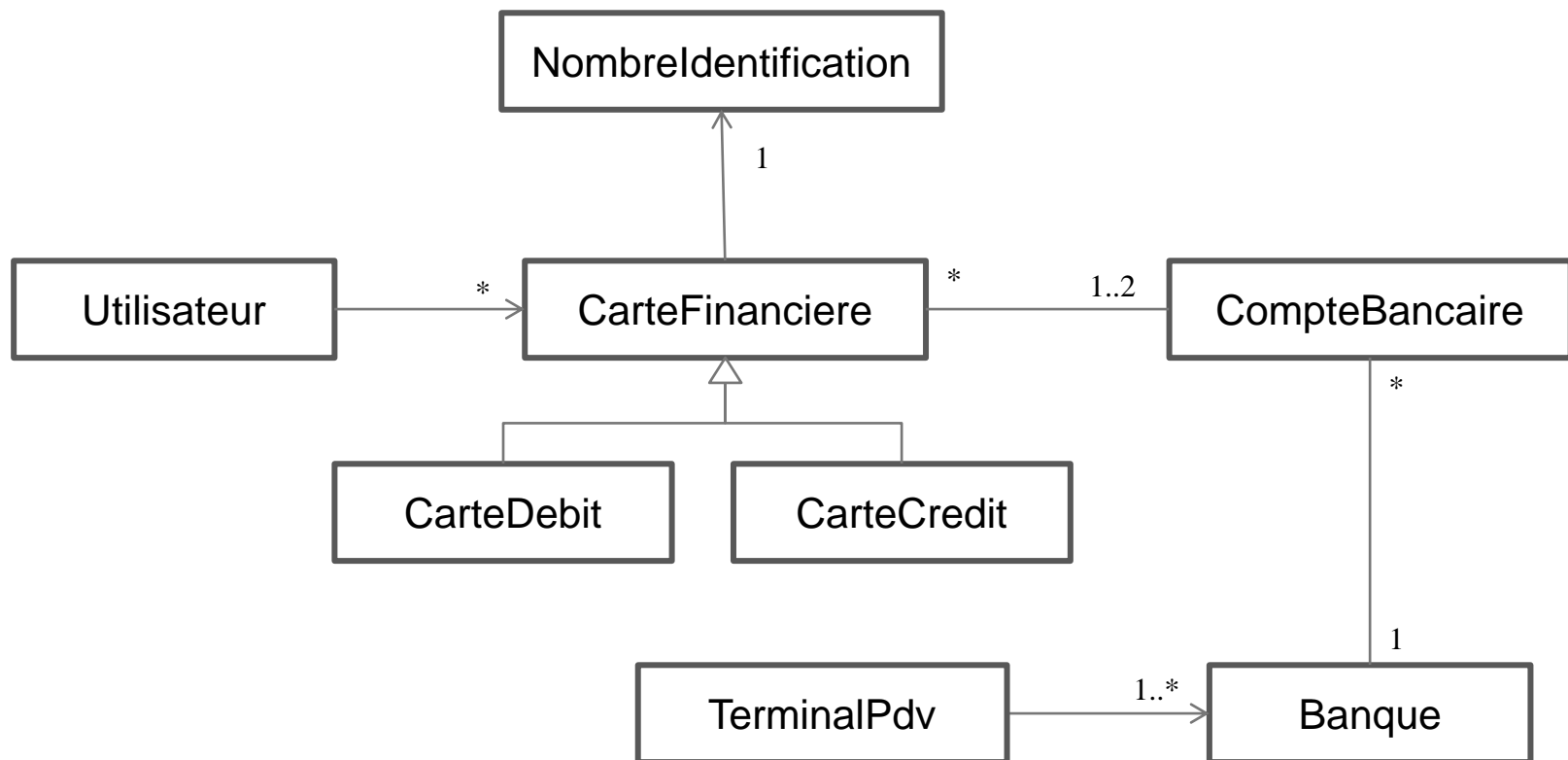
Cependant, en général, la modélisation du domaine est la décomposition d'un domaine en ses entités individuelles

C'est une façon de décrire et de modéliser les entités de domaine et les relations entre elles

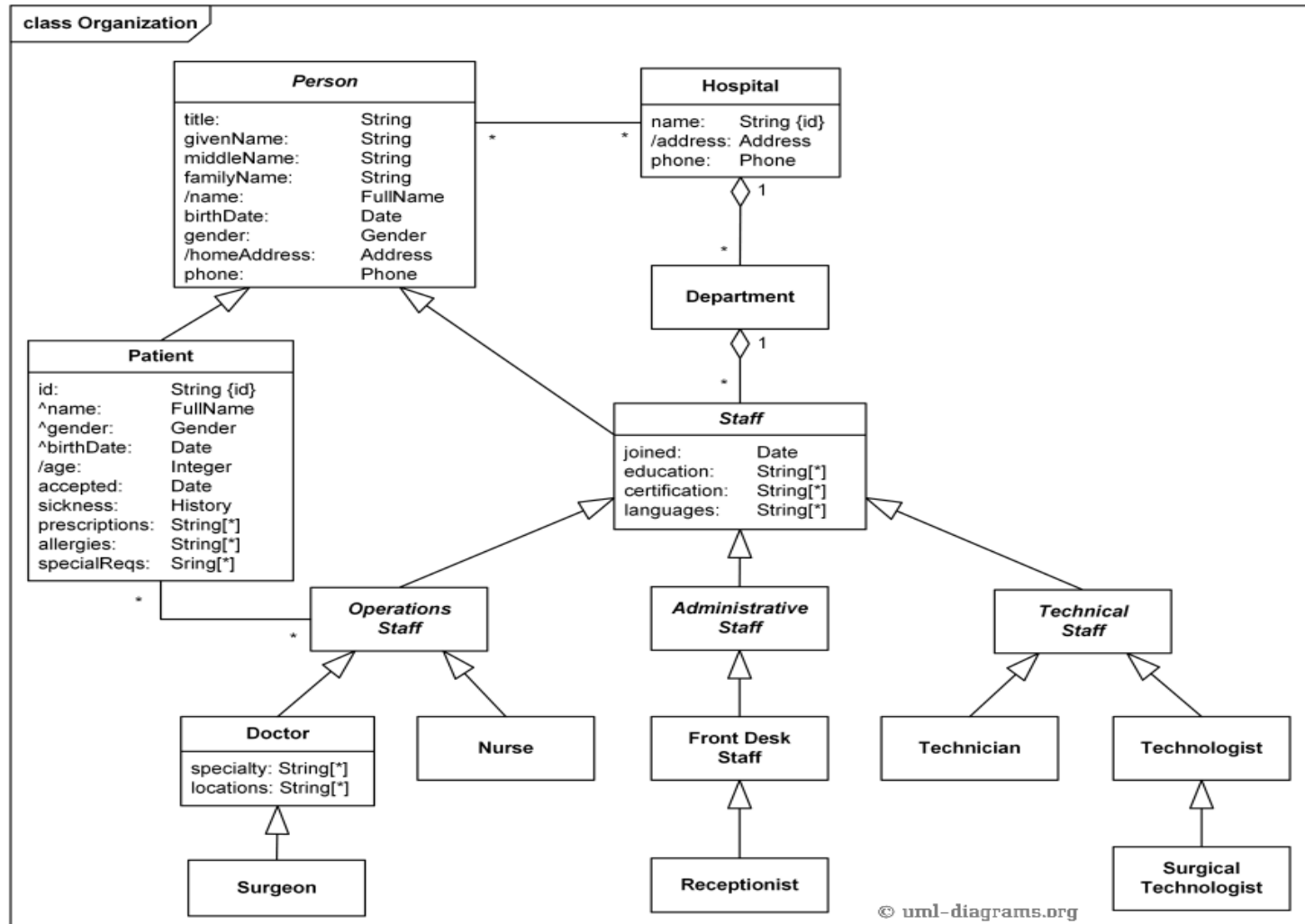
- Ces entités décrivent collectivement l'espace de domaine

En générale, le modèle de domaine pour toutes les entreprises du même domaine doit être presque similaire

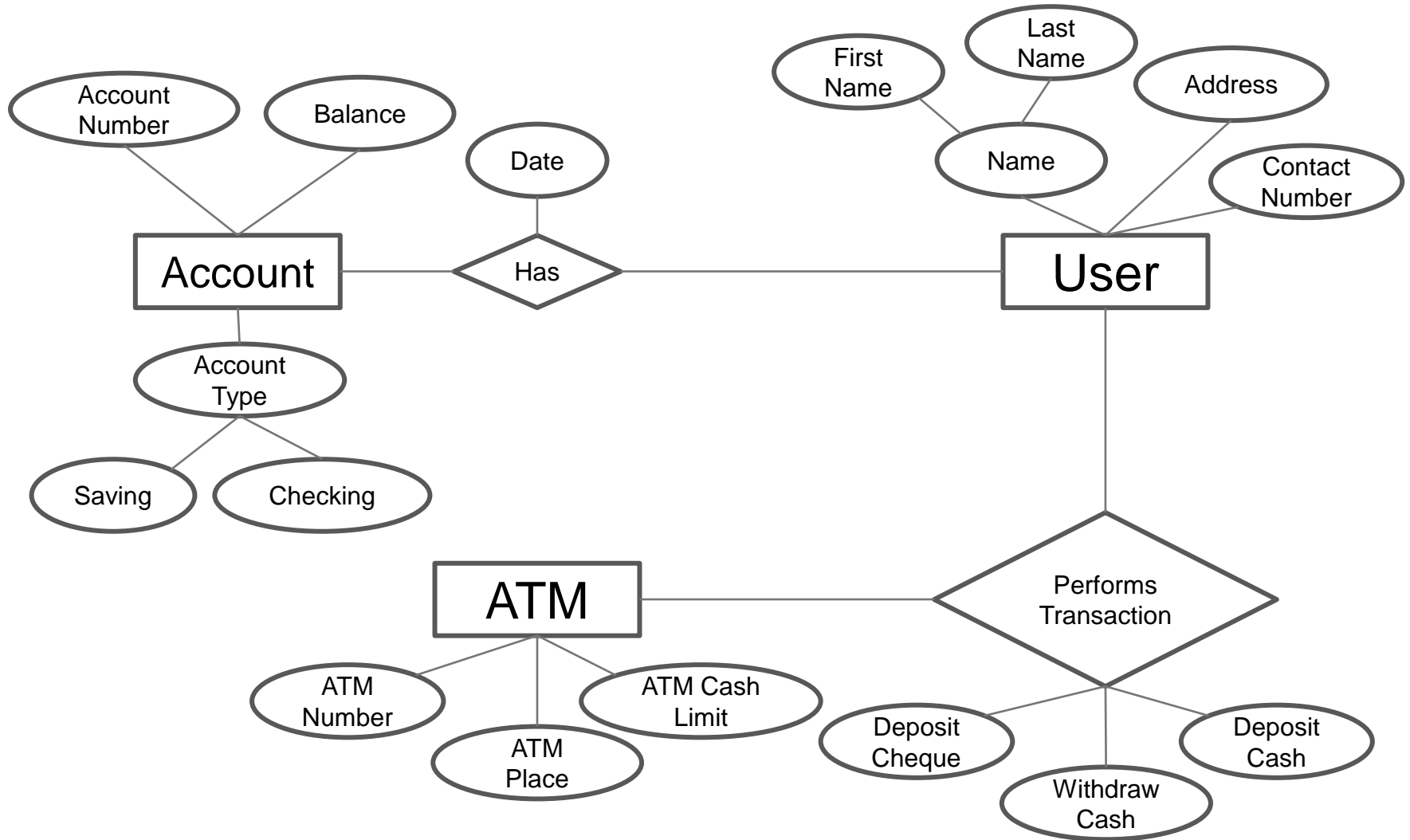
EXEMPLE - MODÈLE DE DOMAINE - DIAGRAMME DE CLASSE UML



EXEMPLE - MODÈLE DE DOMAINE - DIAGRAMME DE CLASSE UML



EXEMPLE - MODÈLE DE DOMAINE - MODÈLE DE RELATION D'ENTITÉ



EXIGENCES

Nous allons décrire trois types d'exigences:

- Exigences du client (aussi appelé informelle)
- Exigences fonctionnelles
- Exigences non-fonctionnelles

EXIGENCES DU CLIENT

Nous avons complété l'analyse du domaine, nous sommes prêts à passer à la prochaine étape

Nous devons savoir exactement ce que le client a besoin:

Exigences du client

- C'est l'endroit où les exigences du client sont collectées

Typiquement composé de déclarations non-techniques de haut niveau

Exemple

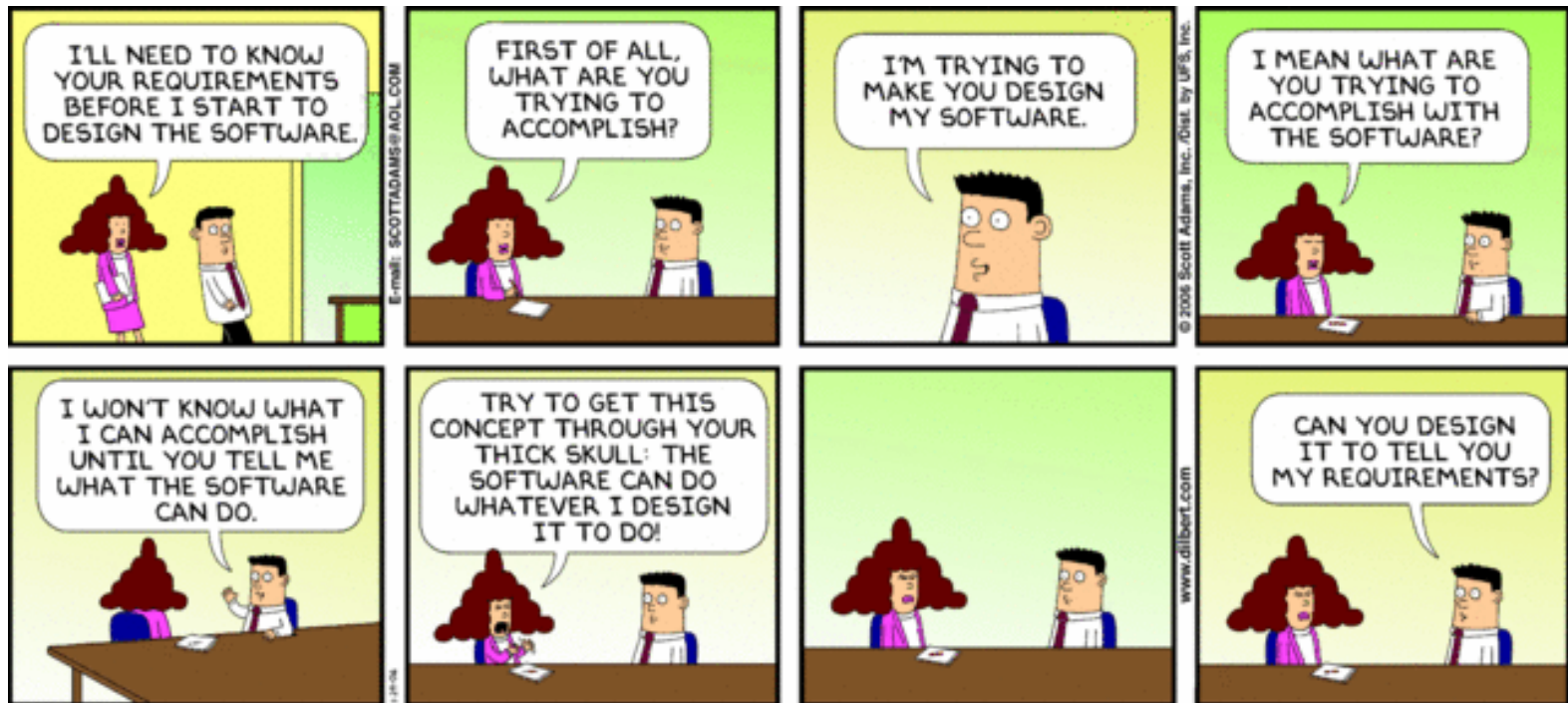
- Exigence 1: « Nous devons développer un portal de client en ligne »
- Exigence 2: « Le portal doit lister tous nos produits »



uOttawa

L'Université canadienne
Canada's university

EXIGENCES DU CLIENT



EXIGENCES FONCTIONNELLES

Capturer le comportement prévu du système

- Peut être exprimé comme services, tâches ou fonctions que le système effectue

Les cas d'utilisation sont rapidement devenus une pratique répandue pour capturer les exigences

- Ceci est surtout vrai dans la communauté objet-orientée d'où ces notions originaires
- Leur application n'est pas limitée aux systèmes objet-orientés

CAS D'UTILISATION

Un *cas d'utilisation* définit un ensemble d'interactions but-orienté entre les acteurs externes et le système en considération

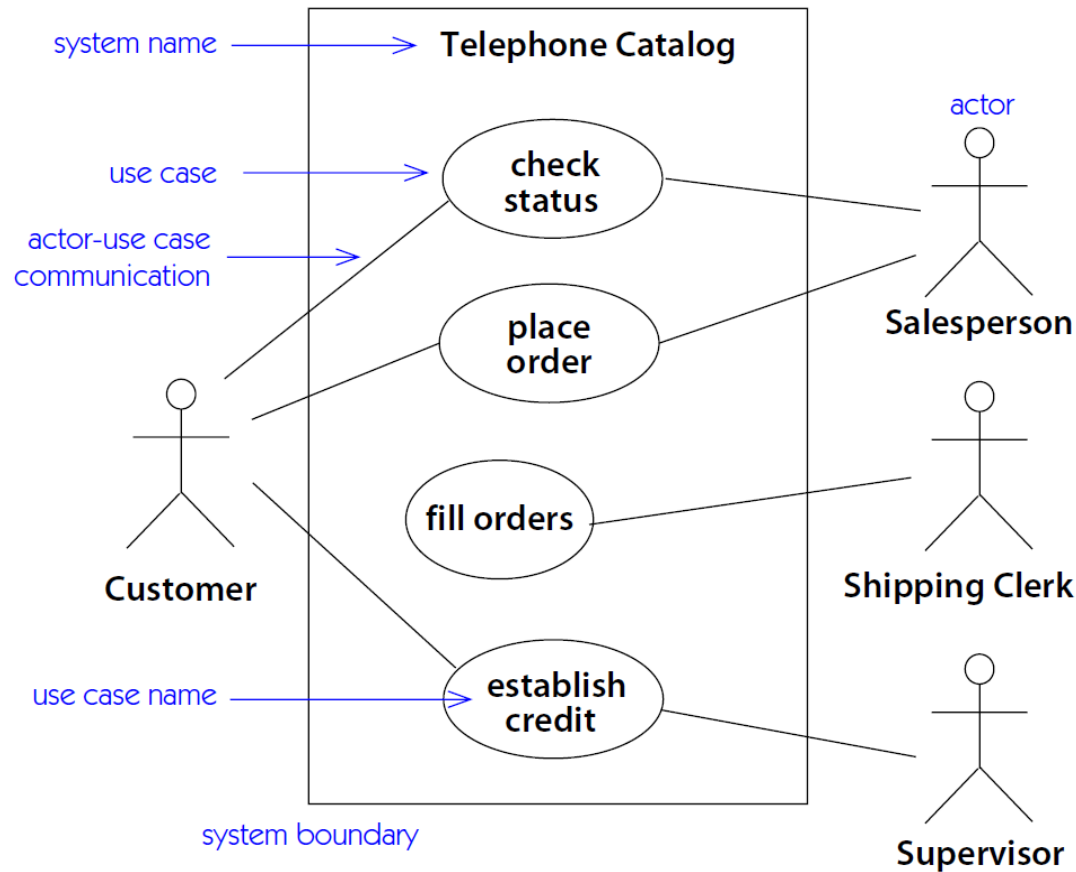
***Les acteurs* sont des composantes à l'extérieur du système qui interagissent avec le système**

- Un acteur peut être une classe d'utilisateurs ou d'autres systèmes

Un cas d'utilisation est initié par un utilisateur qui possède un but particulier en tête, et est complété lorsque le but est atteint

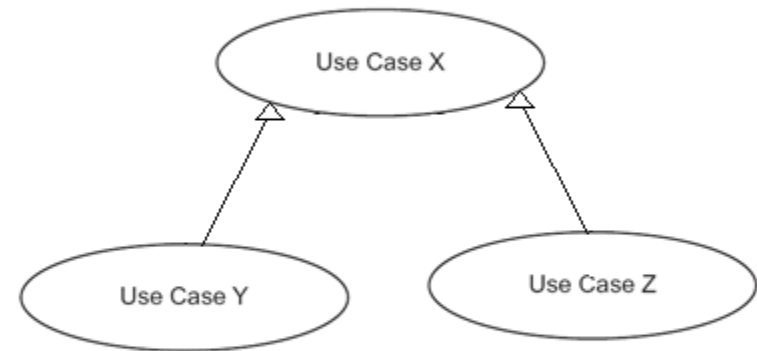
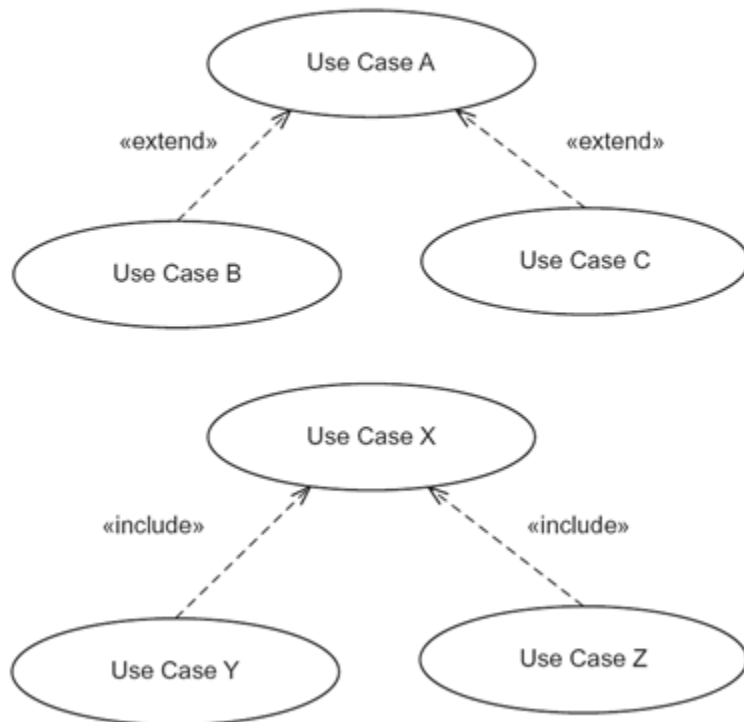
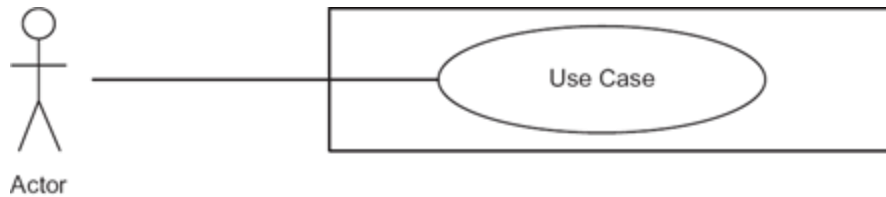
Il décrit la séquence d'interactions entre les acteurs et le système qui sont nécessaires pour livrer le service qui satisfait au but

DIAGRAMMES DE CAS D'UTILISATION



Dessiner UML

DIAGRAMMES DE CAS D'UTILISATION

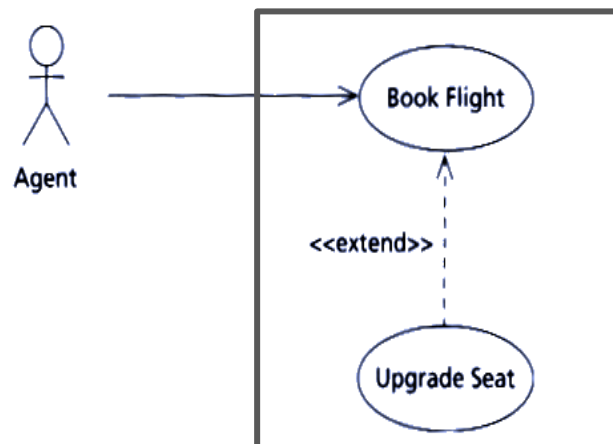


RELATIONS ENTRE LES CAS D'UTILISATION

Relation d'inclusion: un fragment de cas d'utilisation qui est *dupliqué* en multiples cas d'utilisation

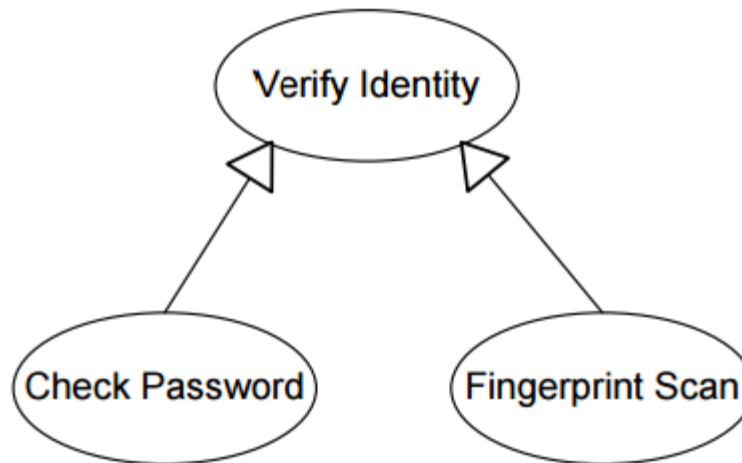
Relation d'extension: cas d'utilisation qui ajoute conditionnellement des étapes à un cas d'utilisation de première classe

- Exemple:

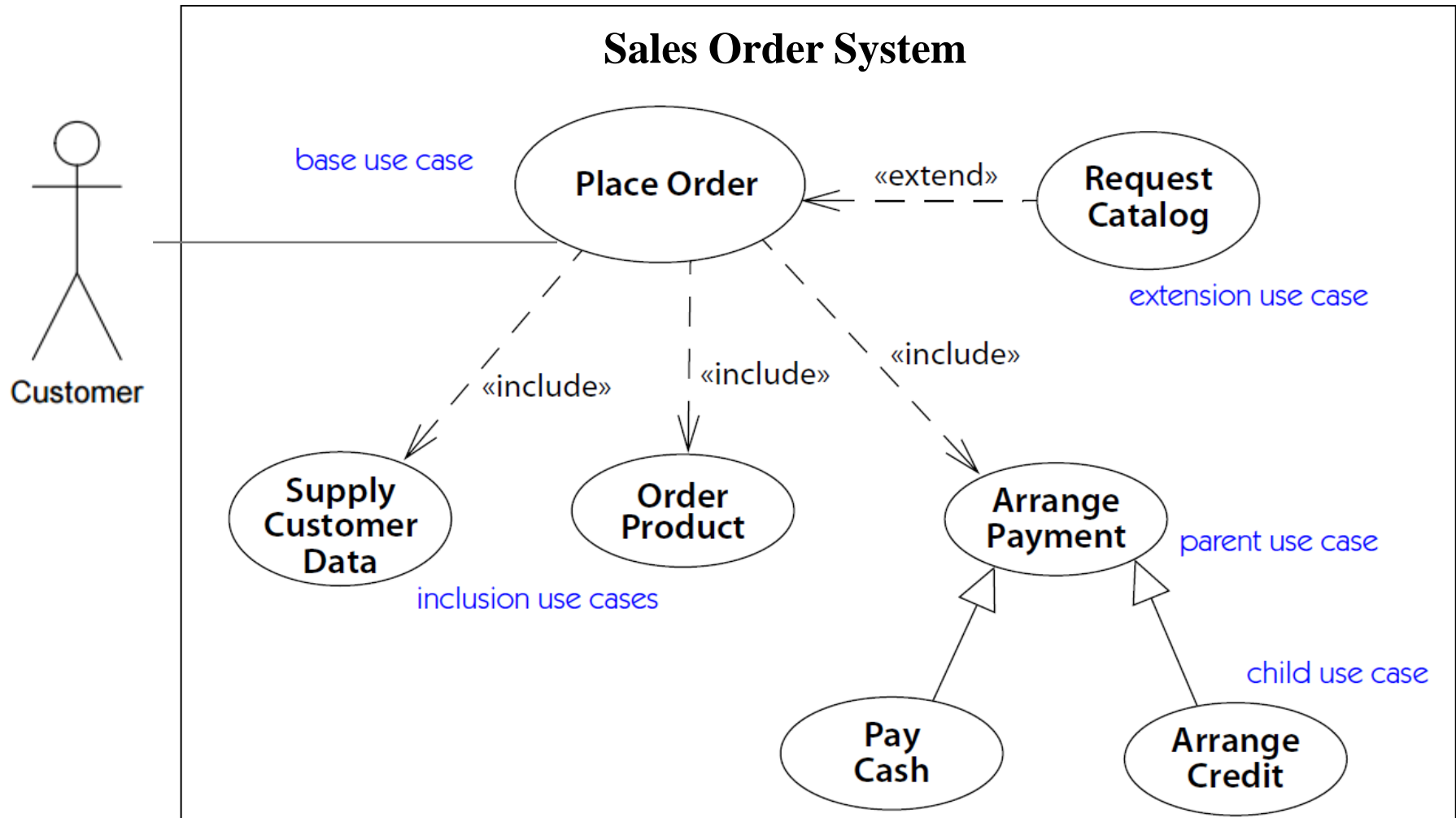


RELATIONS ENTRE LES CAS D'UTILISATION

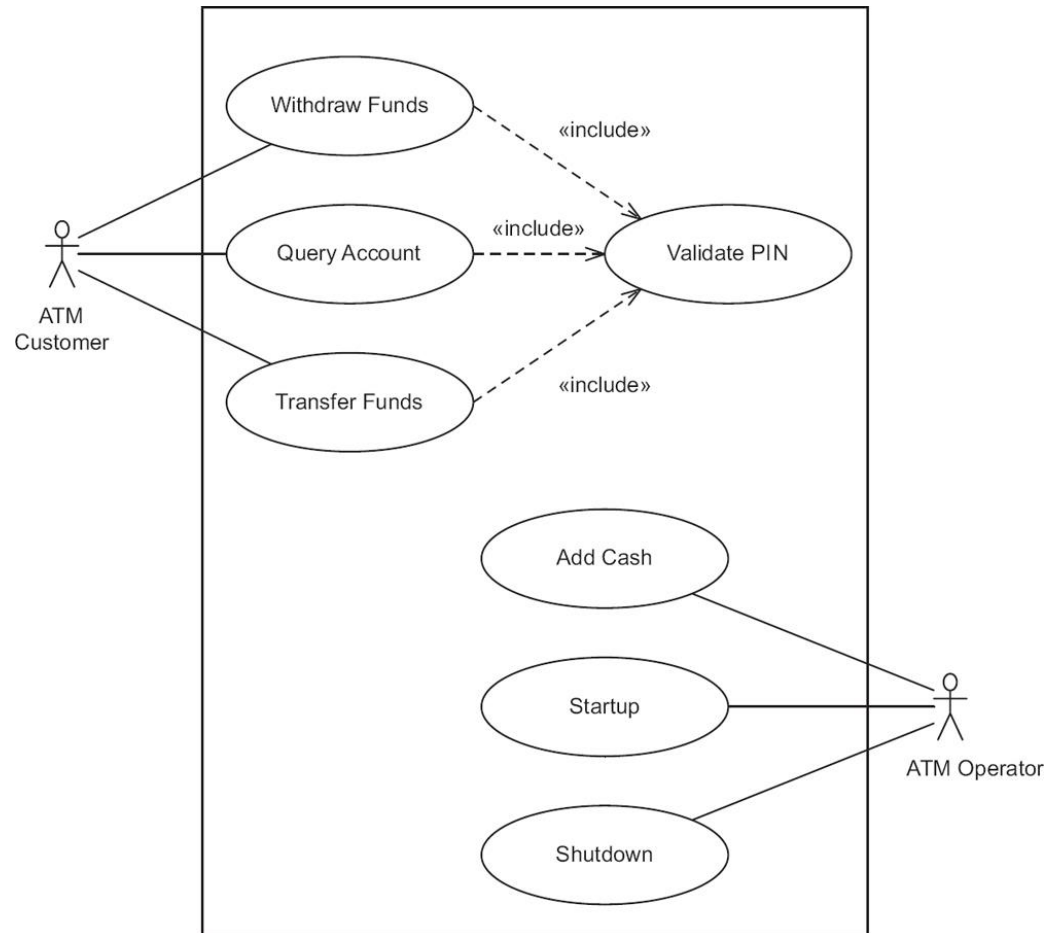
Relation Généralisation: un des cas d'utilisation enfant sera exécuté



RELATIONS ENTRE LES CAS D'UTILISATION



CAS D'UTILISATION – EXEMPLE DE L'ATM



CAS D'UTILISATION – EXEMPLE DE GUICHET AUTOMATIQUE BANCAIRE (ATM)

Acteurs:

- Client de l'ATM
- Opérateur de l'ATM

Cas d'utilisation:

- Le client peut:
 - Retirer des fonds d'un compte chèque ou épargne
 - Voir la balance du compte
 - Transférer des fonds d'un compte à un autre
- L'opérateur de l'ATM peut:
 - Éteindre la machine ATM
 - Remplir le distributeur de monnaie de la machine
 - Démarrer la machine ATM

CAS D'UTILISATION – EXEMPLE DE L'ATM

Valider le NIP est une cas d'utilisation *d'inclusion*

- Il ne peut pas être exécuté tout seul
- Doit être exécuté en faisant partie du *cas d'utilisation concret*

Cependant, un *cas d'utilisation concret* peut être exécuté

CAS D'UTILISATION – ATM – VALIDER LE NIP (1)

Nom du cas d'utilisation: Valider le NIP

Sommaire: Le système valide le NIP du client

Acteur: Client de l'ATM

Précondition: L'ATM est inoccupé, affichant le message de bienvenue.

CAS D'UTILISATION – ATM

– VALIDER LE NIP (2)

Séquence principale:

1. Le client insert sa carte ATM dans le lecteur.
2. Si le système reconnaît la carte, il lit le numéro de la carte.
3. Le système demande au client son NIP.
4. Le client rentre son NIP.
5. Le système vérifie la date d'expiration de la carte et si la carte a été déclarée perdue ou volée.
6. Si la carte est valide, le système vérifie si le NIP rentré correspond au NIP de la carte maintenu par le système.
7. Si les NIP matchent, le système vérifie quels comptes sont accessibles avec la carte ATM.
8. Le système affiche les comptes du client et demande au client quel type de transaction il veut: retirer, voir ou transférer.

CAS D'UTILISATION – ATM

– VALIDER LE NIP (3)

Séquences alternatives:

- Étape 2: Si le système ne reconnaît pas la carte, il éjecte la carte
- Étape 5: Si le système détermine que la carte est expirée, le système confisque la carte.
- Étape 5: Si le système détermine que la carte a été déclarée perdue ou volée, le système confisque la carte.
- Étape 7: Si le NIP rentré par le client ne matche pas le NIP de cette carte, le système redemande le NIP.
- Étape 7: Si le client rentre un NIP incorrect trois fois, le système confisque la carte.
- Étapes 4-8: Si le client rentre « Annuler », le système annule la transaction et éjecte la carte.

Poste-condition: Le NIP du client a été validé.

CAS D'UTILISATION – ATM - RETIRER DES FONDS (1)

Nom du cas d'utilisation: Retirer des fonds

Sommaire : Le client retire un montant spécifique de fonds d'un compte bancaire valide .

Acteur : Le client de l'ATM

Dépendance: Inclure le cas d'utilisation pour valider le NIP.

Condition: ATM est inactif , l'affichage d'un message de bienvenue.

CAS D'UTILISATION – ATM - RETIRER DES FONDS (2)

Séquence principale :

1. Inclure le cas d'utilisation pour valider le NIP.
2. Le client sélectionne « retrait », rentre le montant, et sélectionne le numéro de compte .
3. Le système vérifie si le client dispose de suffisamment de fonds dans le compte et si la limite quotidienne ne sera pas dépassée .
4. Si toutes les conditions sont vérifiées, le système autorise la distribution de l'argent.
5. Le système distribue le montant d'argent requis .
6. Le système imprime un reçu avec le numéro de transaction , type de transaction, le montant retiré , et le solde du compte .
7. Le système éjecte la carte .
8. Le système affiche le message de bienvenue.

CAS D'UTILISATION – ATM - RETIRER DES FONDS (3)

Séquences alternatives :

- Étape 2: Si le système détermine que le numéro de compte n'est pas valide, il affiche un message d'erreur et éjecte la carte.
- Étape 3: Si le système détermine que les fonds sont insuffisants dans le compte du client, il affiche une excuse et éjecte la carte .
- Étape 3: Si le système détermine que le montant de retrait quotidien maximal autorisé a été dépassé, il affiche une excuse et éjecte la carte .
- Étape 5: Si les fonds de l' ATM sont épuisées, le système affiche une excuse, éjecte la carte, et arrête le guichet automatique.

Poste-condition : les fonds du client ont été retirés .

EXIGENCES NON FONCTIONNELLES

Les exigences fonctionnelles définissent ce que le système doit faire

Les exigences non fonctionnelles définissent comment le système est supposé être.

- Généralement, décrit les attributs du système tels que sécurité, fiabilité, maintenance, d'évolutivité, utilisabilité...

EXIGENCES NON FONCTIONNELLES

Les exigences non-fonctionnelles peuvent être spécifiées dans une section distincte de la description de cas d'utilisation

- Dans l'exemple précédent, pour le cas d'utilisation « valider le NIP », il peut y avoir une exigence de sécurité que le numéro de la carte et le NIP doivent être encryptés.

Les exigences non fonctionnelles peuvent être spécifiées pour un groupe de cas d'utilisation ou pour le système entier.

- **Exigence de sécurité:** Le système doit encrypté le numéro de carte ATM et le NIP.
- **Exigence de performance:** Le système doit répondre aux sélections de l'acteur dans un délai de 5 secondes.

EXIGENCES DANS LES PROCESSUS AGILES

Exigences dans SCRUM et programmation extrême sont décrites en utilisant des histoires d'utilisateurs

Une histoire d'utilisateur est une définition de haut niveau d'une exigence

- Descriptions simples d'une fonction dit du point de vue de la personne qui désire la nouvelle capacité
- Au cours de la population du backlog, ils contiennent juste assez d' informations pour que les développeurs peuvent produire une estimation raisonnable de l'effort
- Plus tard, plus de détails peuvent être ajoutés en consultation avec le client

Ils suivent généralement un format simple:

Comme un <type d'utilisateur>, je veux <un but> de sorte que <une raison quelconque>

EXIGENCES DANS LES PROCESSUS AGILES

Les histoires d'utilisateurs peuvent être écrits à différents niveaux de détail

- Une histoire d'utilisateur qui couvre une grande quantité de fonctionnalités est généralement connu comme une histoire épique
- Une histoire épique ne peut pas être complétée en une seule itération
- Il est décomposé en histoires d'utilisateurs qui peuvent être implémentées en une seule itération

Exemple

- Histoire épique: En tant qu'utilisateur, je peux sauvegarder tout mon disque dur.
- Historique des utilisateurs:
 - En tant qu'utilisateur, je peux spécifier des fichiers ou des dossiers à sauvegarder selon la taille du fichier, la date de création et date de modification.
 - En tant qu'utilisateur, je peux indiquer les dossiers à ne pas sauvegarder afin que mon lecteur de sauvegarde ne soit pas rempli de choses que je n'ai pas besoin de sauvegarder

MERCI!

QUESTIONS?