
Cours 4 – Expression de Besoins

MODULE INTRODUCTION AU GÉNIE LOGICIEL

Objectifs du Cours

Permettre à l'étudier de
pouvoir exprimer les
besoins d'un logiciel d'une
manière formelle

Apprendre le modèle de
spécifications bien
formées des spécifications

Utiliser les diagrammes
de cas d'utilisation pour
modélisation les
spécifications
fonctionnelle

Utiliser les techniques
avancées des diagrammes
de cas d'utilisation pour
produire des diagrammes
représentatifs

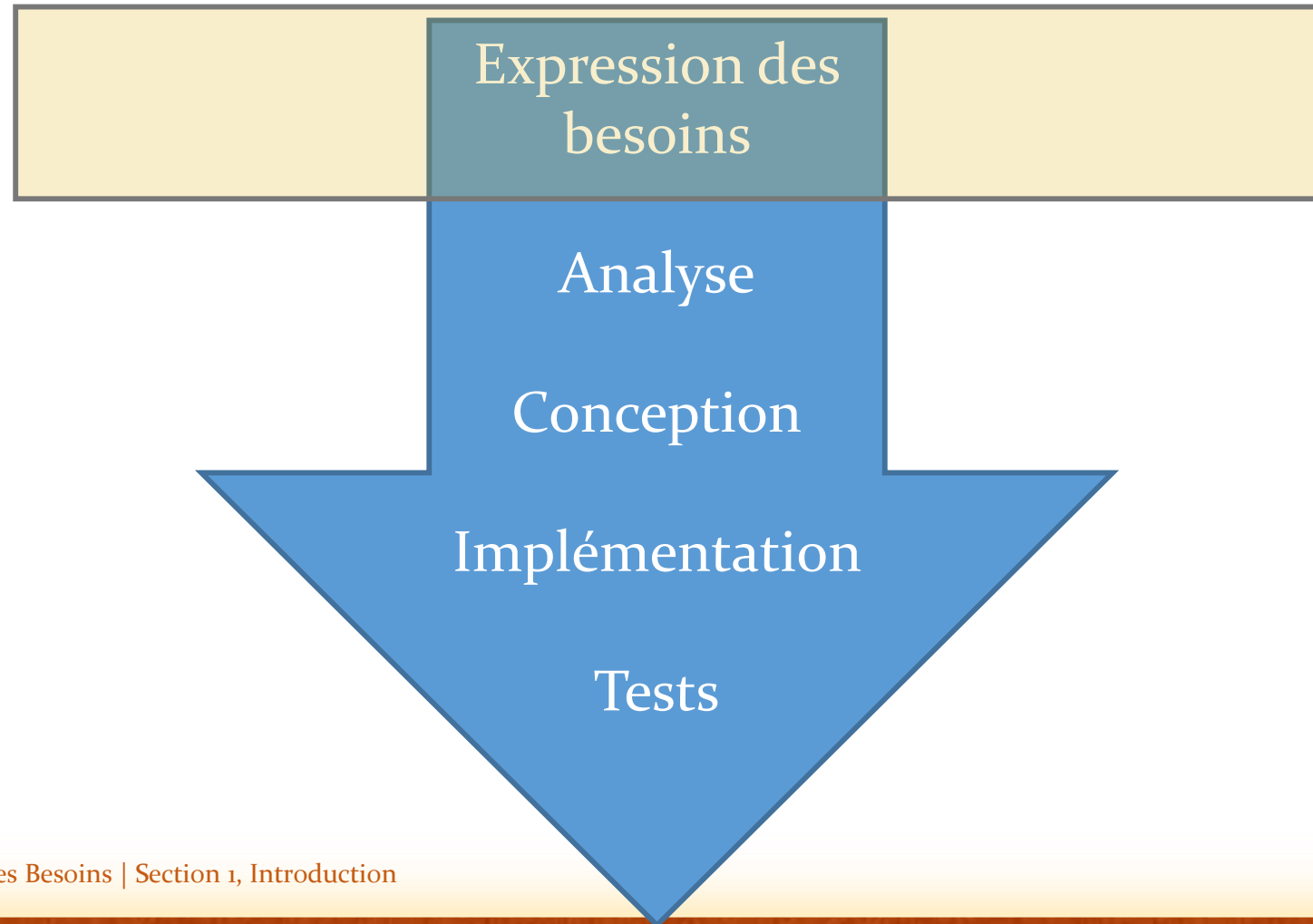
Plan du Cours



Introduction

SECTION 1

Cycle de Vie



Motivations

Quoi et Pourquoi
développer ?

Besoin de comprendre
le client

Besoin de formaliser
les attentes du client

Mieux les besoins sont
capturés et bien
formalisés, plus le
projet a des chances de
réussir

Aider le client à
exprimer ce qu'il sait et
ce qu'il veut

Types de besoins

Besoins fonctionnels

- Que doit faire le système
- Le système du point de vue de son utilisateur
- Quoi ?

Besoins non fonctionnels

- Contraintes
- Exigence ou choix techniques
- Comment ?

Spécifications

- Les spécifications sont l'expression formelle des besoins
- Une spécification fonctionnelle exprime comment est le système du point de vue utilisateur
- Une spécification technique exprime comment est le système d'un point de vue interne (technique, technologie,...etc.)
- Le langage naturel peut être utilisé pour les spécifications fonctionnelles et / ou techniques

Problèmes de Recensement

Problèmes de compréhension

- Les développeurs et le client ne parlent pas le même langage

Problèmes de volatilité

- Une spécification « valide » peut ne plus l'être après une courte période de temps

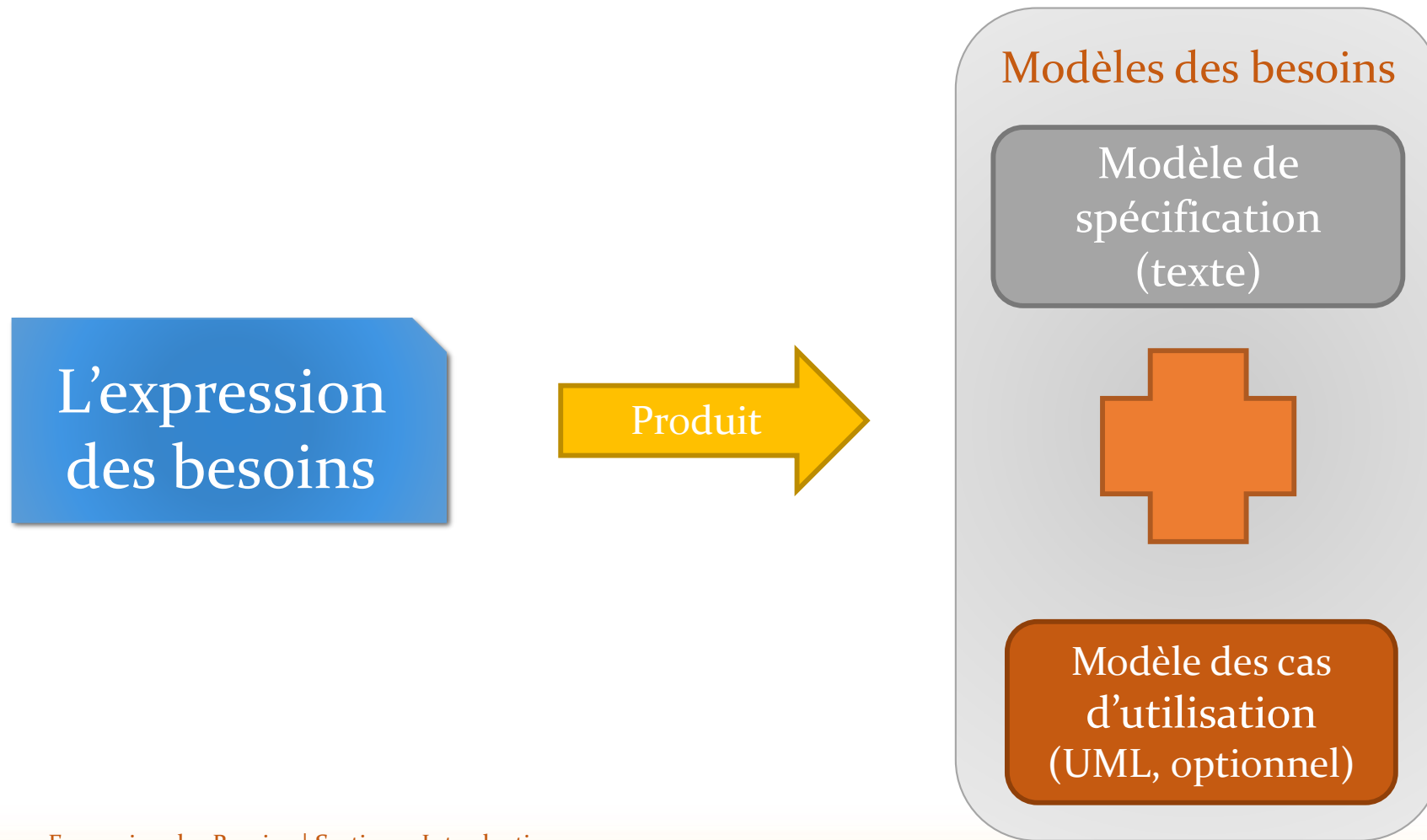
Problèmes humains

- Conflits
- Rétention d'information
- ...etc.

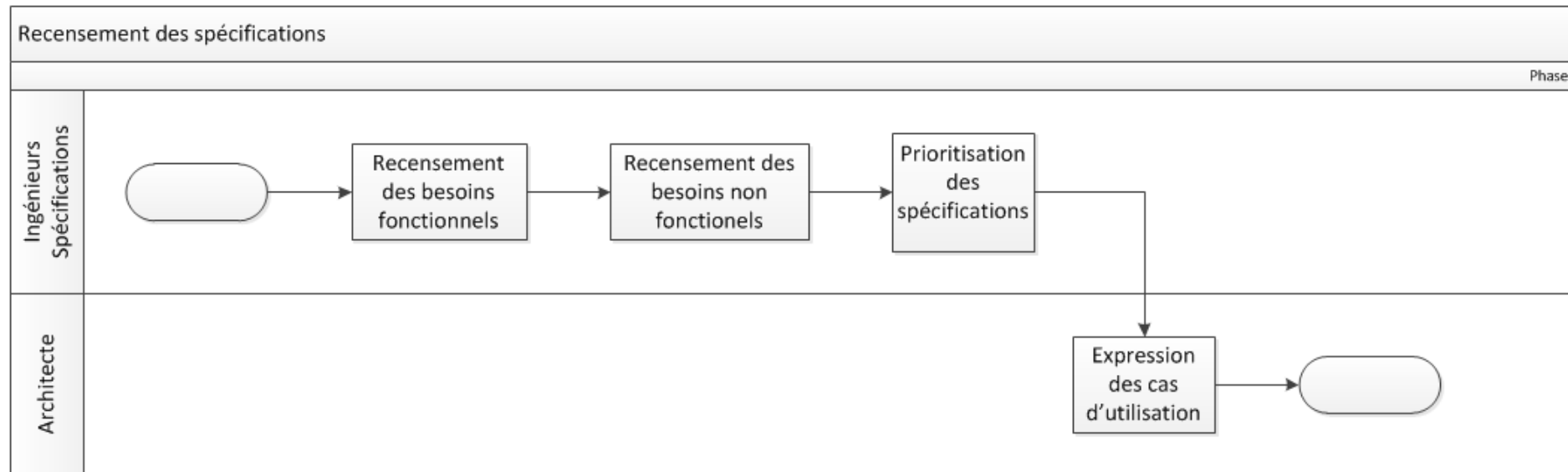
Problèmes de portée

- Savoir l'étendue d'une spécification
- À quel sous-système elle appartient ?

Modèles de Besoins



Processus d'Expression des Besoins



Expression de besoins dans UP

- UP propose deux modèles pour l'expression des besoins : le *modèle de spécifications* et le *modèle des cas d'utilisation*
- Le modèle des spécifications convient aussi bien pour les *spécifications fonctionnelles* que les *spécifications non fonctionnelles*
- Le modèle des cas d'utilisation est basé sur les *diagrammes de cas d'utilisation d'UML*
- Le modèle des cas d'utilisation s'adapte mieux aux *spécifications fonctionnelles*
- Le modèle de spécifications peut être créé en utilisant un éditeur de texte, un bloc-notes ou un outil dédié

Introduction

• ————— •
SECTION 1 – DÉBAT (10 MNS)

Modèle de Spécifications



SECTION 2

Modèle de spécifications

Un ensemble de phrases bien formées

Phrases numérotées

Formulation Uniforme

Chaque spécification a un *numéro unique*

Une spécification décrit *une seule fonction* du système

Une spécification peut être *fonctionnelle* (décrivant un aspect métier) ou *non-fonctionnelle* (décrivant un aspect technique)

Les spécifications peuvent être rédigées en utilisant un éditeur texte ou un outil dédié

Facilitent la compréhension, la gestion et la communication

Formulation

(id) le (système) doit <fonction>

Exemple de Formulation

Spécifications fonctionnelles :

- Le système GAB (distributeur) de biller doit vérifier la validité de la carte CIB insérée
- Le système GAB doit valider le code PIN entré par l'utilisateur
- Le système doit allouer une somme maximum de 20000 dinars à l'utilisateur

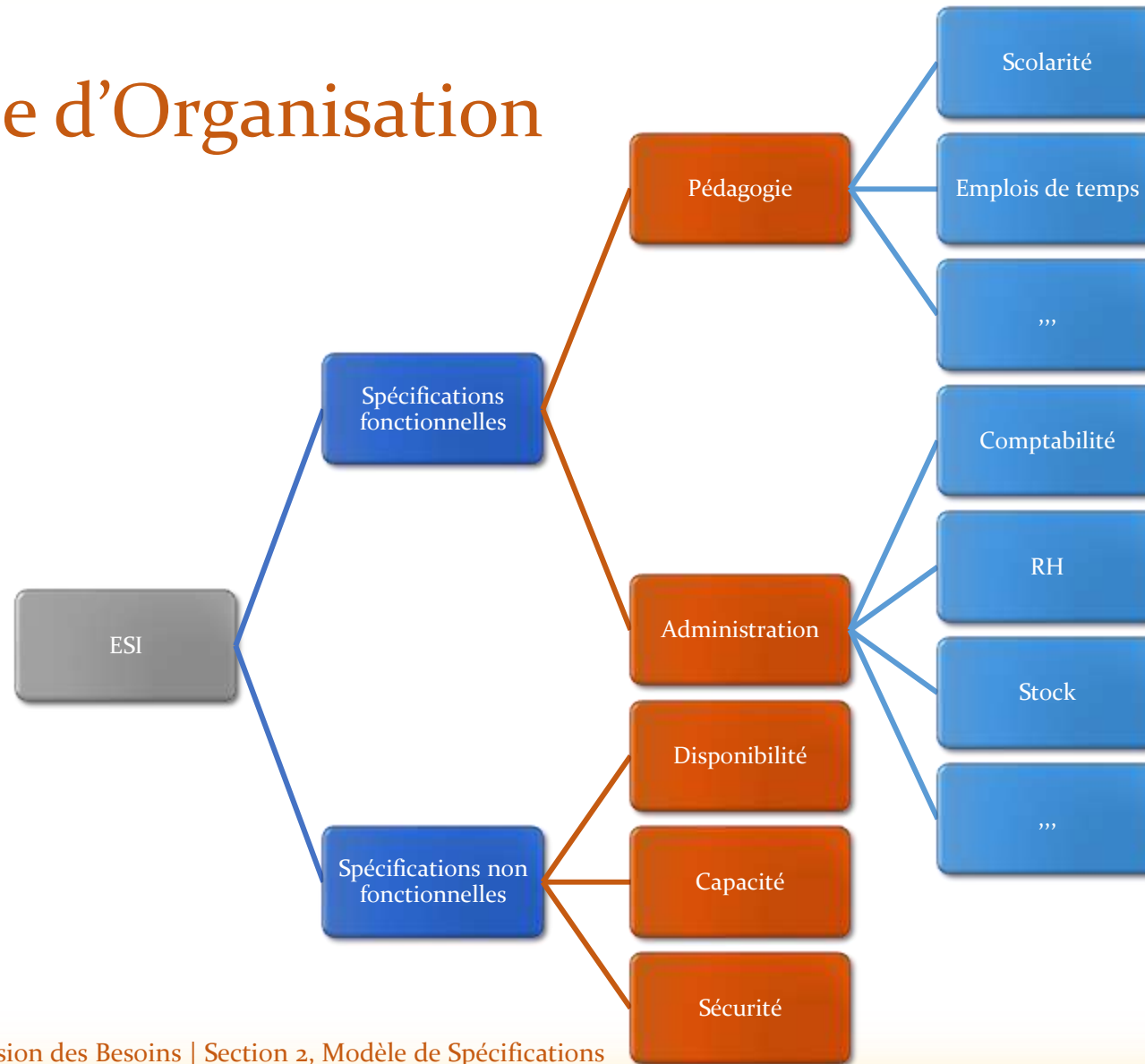
Spécifications non fonctionnelles :

- Le système du GAB doit être écrit en C++
- Le système du GAB doit utiliser un cryptage 256 bits pour les données sensibles
- Le système doit vérifier le PIN en moins de 03 secondes

Organisation des spécifications

- Les systèmes de moyenne et grande taille engendrent un nombre très important de spécifications (de quelques centaines à plusieurs milliers)
- L'organisation des spécifications permet de les structurer et bien gérer les autres activités de développement
- L'organisation permet aussi de scinder les spécifications en sous-domaines

Exemple d'Organisation



Priorités des Spécifications

Chaque spécification doit avoir une priorité

La priorité permet de sélectionner les spécifications les plus importantes

Facilite l'ordonnement

La priorité peut être un nombre (de 1 à 5)

MoSCoW est une méthode permettant d'affecter des priorités aux spécifications

Priorités de spécifications – Méthode MoSCoW

Priorité	Description
M (Must have)	Spécification obligatoire et fondamentale du système
S (Should have)	Spécification importante qui peut être omise sous certaines conditions
C (Could have)	Spécification optionnelle, réaliser si on a le temps
W (Want to have)	Spécifications qui peuvent attendre les dernières livraisons du système

Attributs de spécifications

Attribut	Description
Etat	<ul style="list-style-type: none">• Proposé : en cours de discussion, pas encore validé• Approuvé : validé et attend d'être implémenté• Rejeté : rejeté et n'attendra pas son implémentation• Incorporé : spécification qui a été implémentée durant une livraison
Criticisme	<ul style="list-style-type: none">• Critique : la spécification doit être implémentée sinon le système n'est pas accepté• Important : la spécification peut être omise mais son omission affecterait considérablement l'utilisabilité du système• Utile : la spécification peut être omise et son omission n'a pas un grand impact sur le système

Attributs de spécifications - Suite

Attribut	Description
Effort	Estimation approximative des ressources et du temps nécessaire pour la spécification
Risque	Le risque relatif à cette spécification : élevé, moyen ou bas
Stabilité	La probabilité que la spécification change dans le temps
Cible	La version du produit dans laquelle la spécification est planifiée
Description	Une description plus détaillée de la fonctionnalité

Sources de Recensement

Utilisateurs
directs

Personnes ayant
une relation avec
le système

Autres systèmes
avec lesquels va
interagir le
logiciel

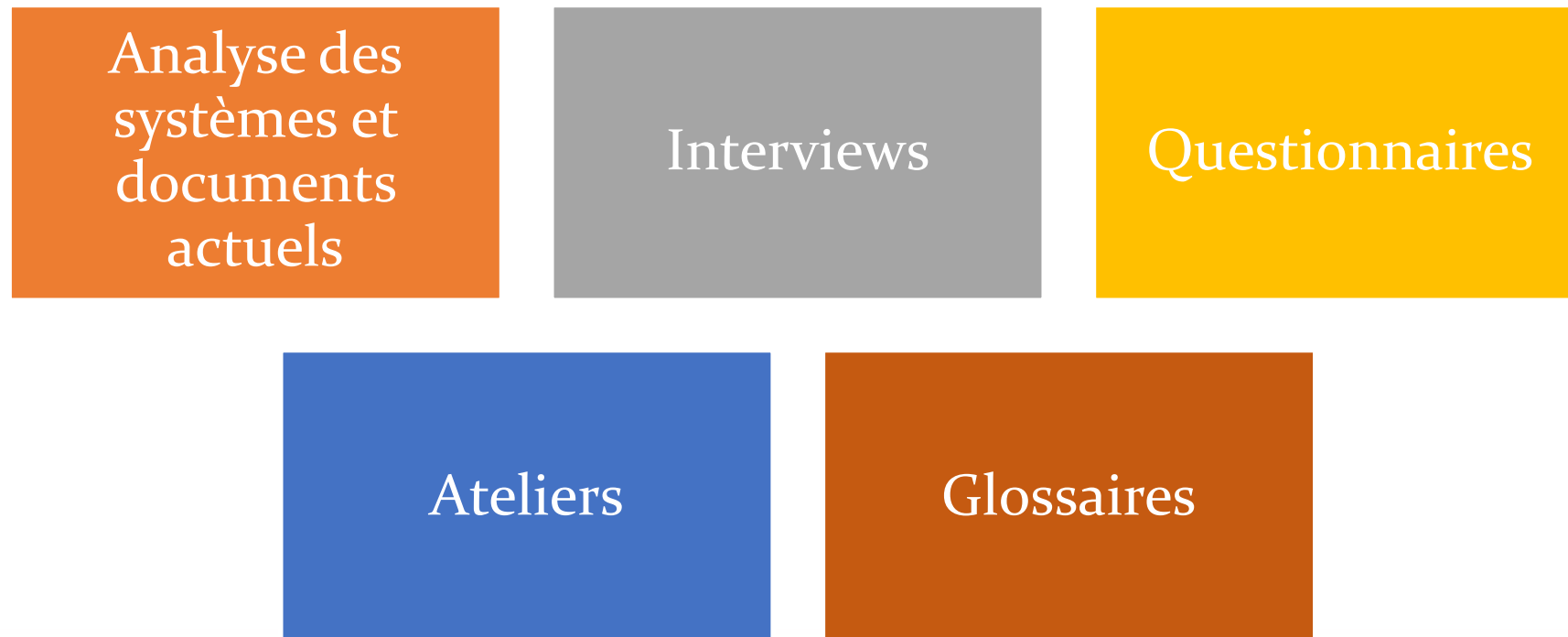
Le matériel sur
lequel va être
déployé le logiciel

Contraintes
juridiques et
administratives

Contraintes
techniques

Objectifs métier

Mécanismes de Recensement



Scénario - Exemple

« Le client parcourt le catalogue de produit et ajoute les produits qui lui plaisent à son panier. Quand le client souhaite finaliser son achat, il fournit ses informations sur sa carte et sur son adresse de livraison. Le système vérifie la carte du client ensuite valide la transaction. »

Scénario – Exemple

Id	Spécification	Etat	Criticisme	Effort	Stabilité	Cible
1	Le système doit permettre à l'utilisateur de parcourir le catalogue de produits	Incorporé	Critique	20j	Stable	1.0
2	Le système doit afficher à l'utilisateur la liste des produits qu'il a cherché par nom	Proposé	Critique	3j	Stable	1.0
3	Le système doit afficher à l'utilisateur la liste des produits qu'il a cherché par référence	Proposé	Important	3j	Stable	1.5
4	Le système doit valider le numéro de la carte de crédit du client	Proposé	Critique	5j	Stable	1.0
5	Le système doit analyser l'historique des achats du client pour proposer les produits qui peuvent l'intéresser le plus	Proposé	Important	8j	Instable	2.0
6	Le système doit permettre à l'utilisateur d'ajouter des produits dans son panier	Incorporé	Critique	5j	Stable	1.0

Modèle de Spécifications



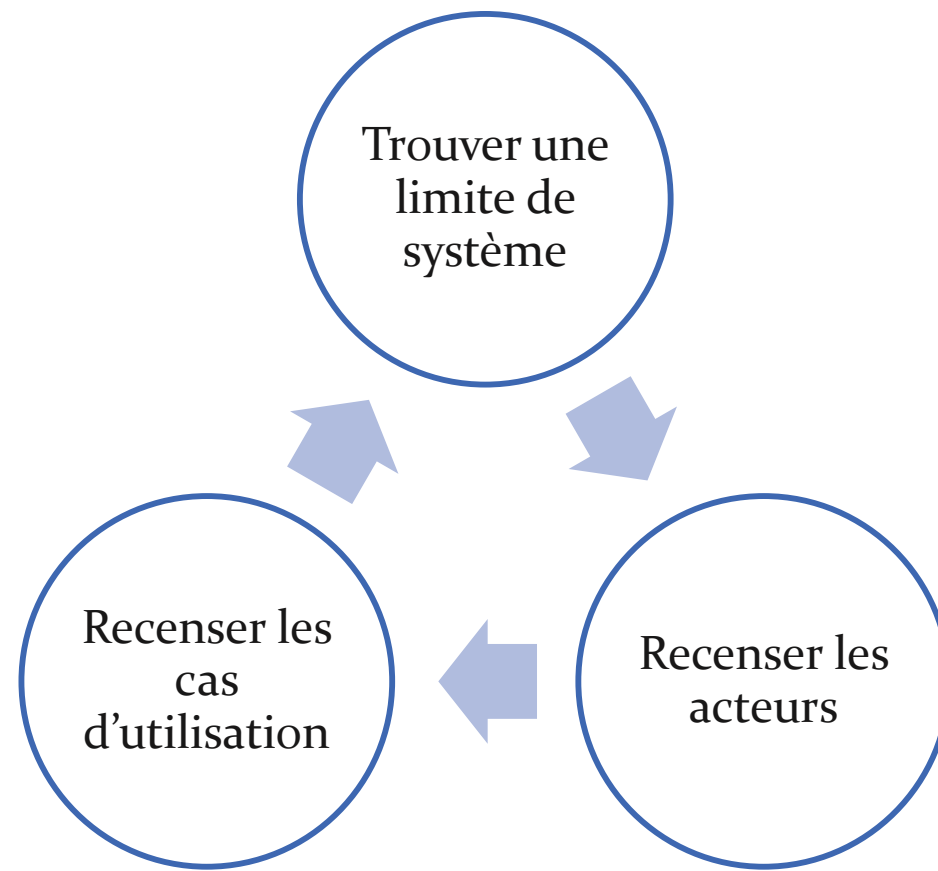
SECTION 2 – DÉBAT (10 MNS)

Modèle des Cas d'Utilisation



SECTION 3

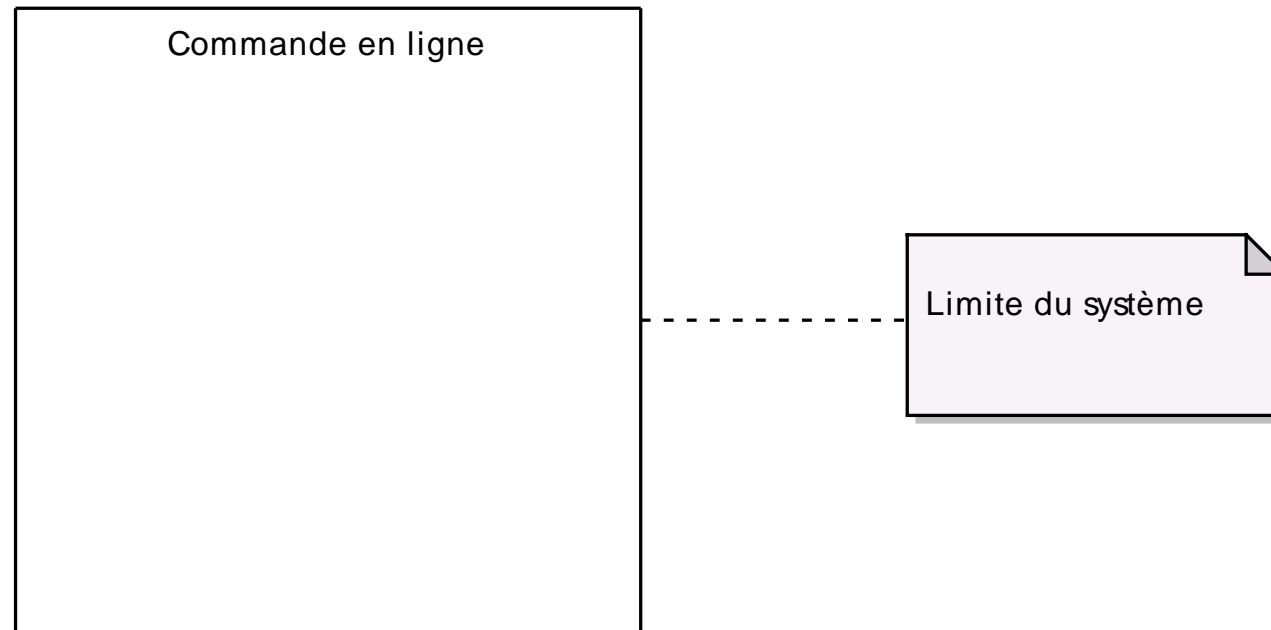
Création des Cas d'Utilisation



Limites du Système

- Quelles sont les limites du système ? Qu'est-ce qui fait partie du système et qu'est-ce qui est externe au système ?
- Décider clairement des limites du système n'est pas aussi évident qu'on le croit
- La limite du système est aussi appelée « sujet »

Limite du système – Notation UML



Acteurs

Les utilisateurs du système
sont des « **acteurs** »

Un acteur identifie un rôle
pas une personne

Un cas d'utilisation peut
avoir plusieurs acteurs

Un acteur peut être
impliqué dans plusieurs cas
d'utilisation

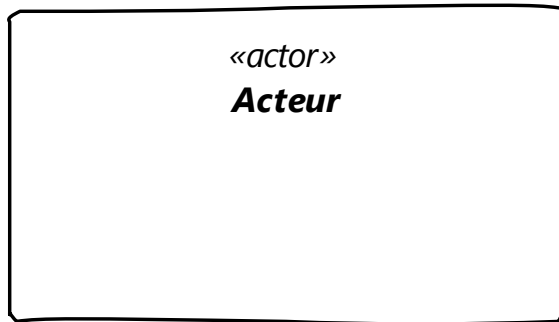
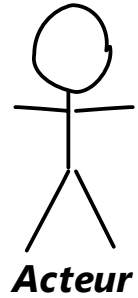
Un acteur interagit
directement avec le système

Un acteur peut identifier
une entité non humaine : un
matériel ou un autre
système interagissant avec
le système, il est aussi utilisé
pour représenter le temps

Les acteurs sont externes au
système

Les acteurs sont nommés en
utilisant des noms courts et
significatifs

Représentation des acteurs



Identification des acteurs

Qui ou qu'est-ce
qui utilise le
système ?

Quel est leur rôle
dans l'interaction
?

Qui installe le
système ?

Qui démarre ou
arrête le système ?

Quels sont les
systèmes qui
interagissent avec
ce système ?

Qui fournit les
informations au
système ?

Quels sont les
événements qui
ont lieu à un
moment donné ?

Scénario

Exemple d'un achat en ligne

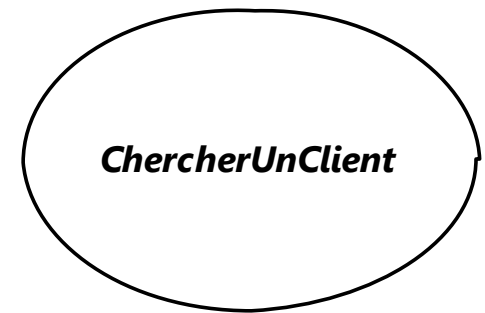
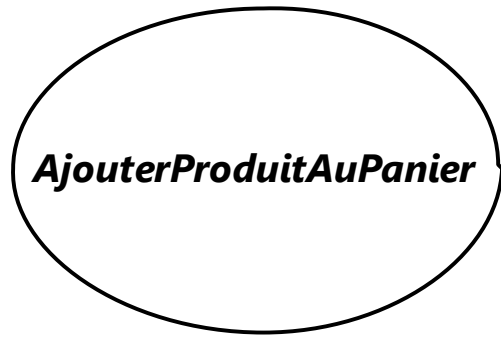
« Le client parcourt le catalogue de produit et ajoute les produits qui lui plaisent à son panier. Quand le client souhaite finaliser son achat, il fournit ses informations sur sa carte et sur son adresse de livraison. Le système vérifie la carte du client ensuite valide la transaction. »

- D'autres scénarios dérivés peuvent avoir lieu : par exemple le client a une remise ou la carte est invalide.
- Malgré leur différences, tous ces scénarios forment un seul cas d'utilisation car l'objectif est unique : acheter un produit.

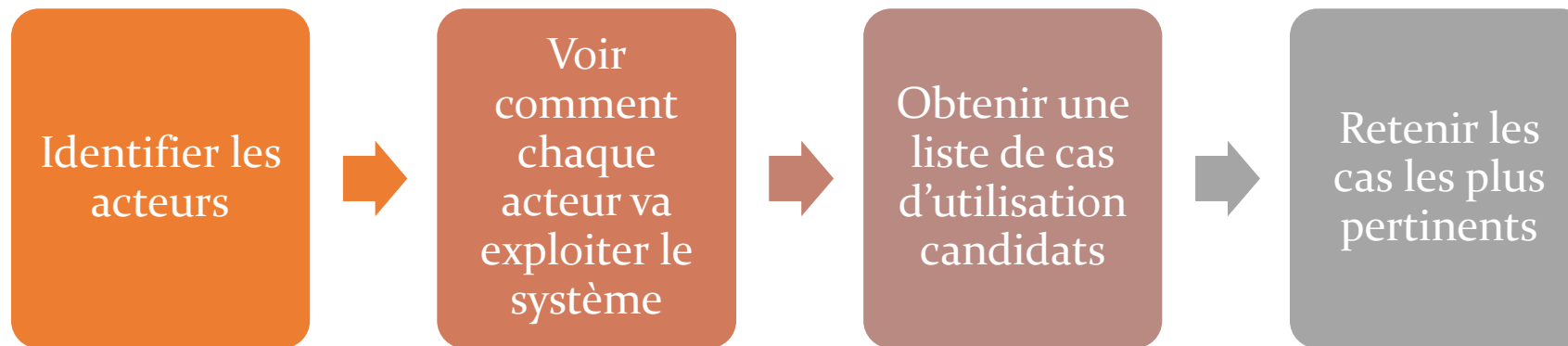
Cas d'utilisation

- Les cas d'utilisation sont une technique de capture des **besoins fonctionnels** du système
- Les cas d'utilisation décrivent les **interactions** entre les utilisateurs du système et le système lui-même
- Les cas d'utilisation est une description “**narrative**” de comment est utilisé le système
- Les cas d'utilisation d'indiquent pas le séquençement des événements
- Les cas d'utilisation sont toujours **déclenchés** par les acteurs
- Les cas d'utilisation représentent le système du **point de vue** de l'acteur

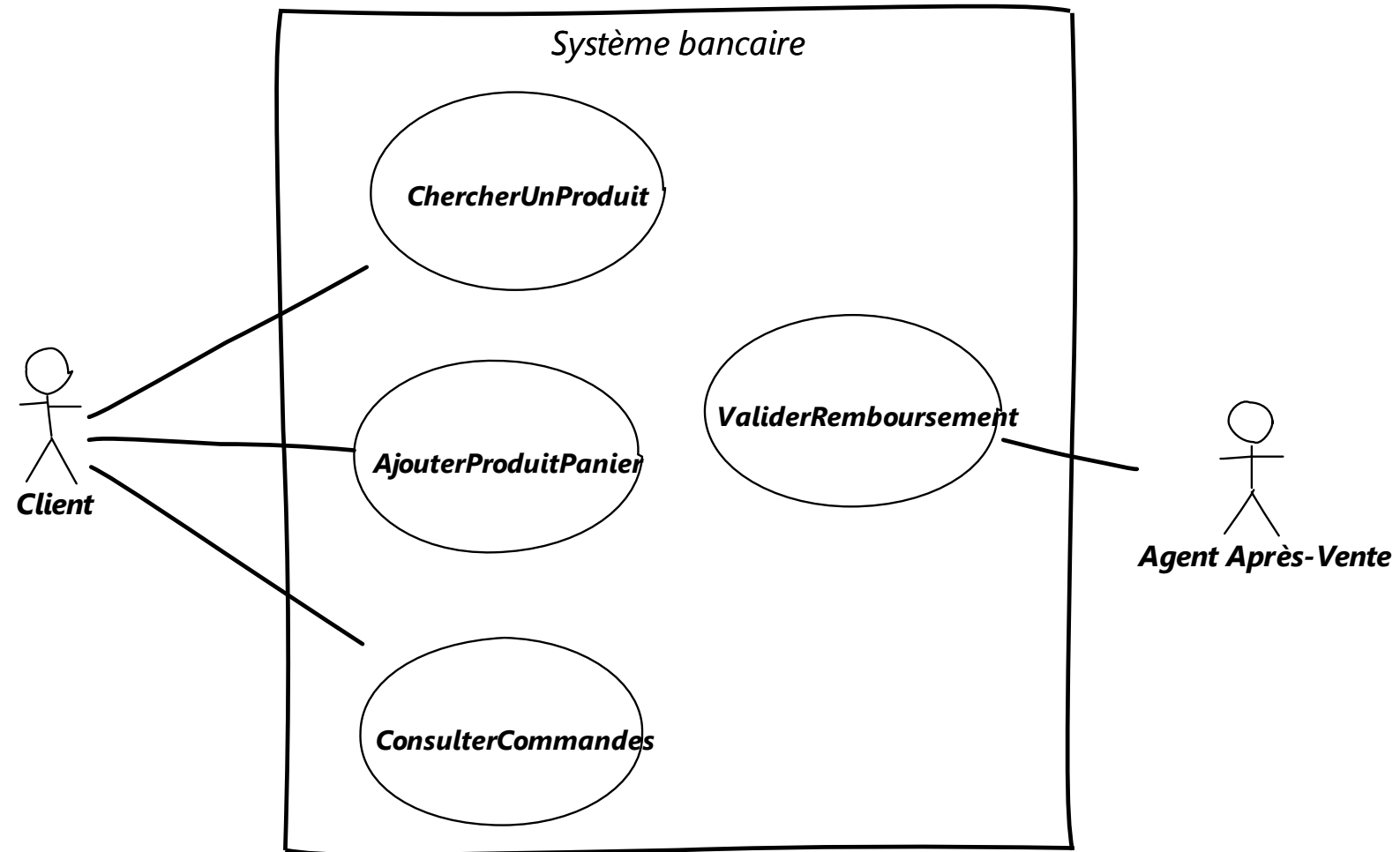
Représentation Cas d'utilisation



Création des Cas d'Utilisation

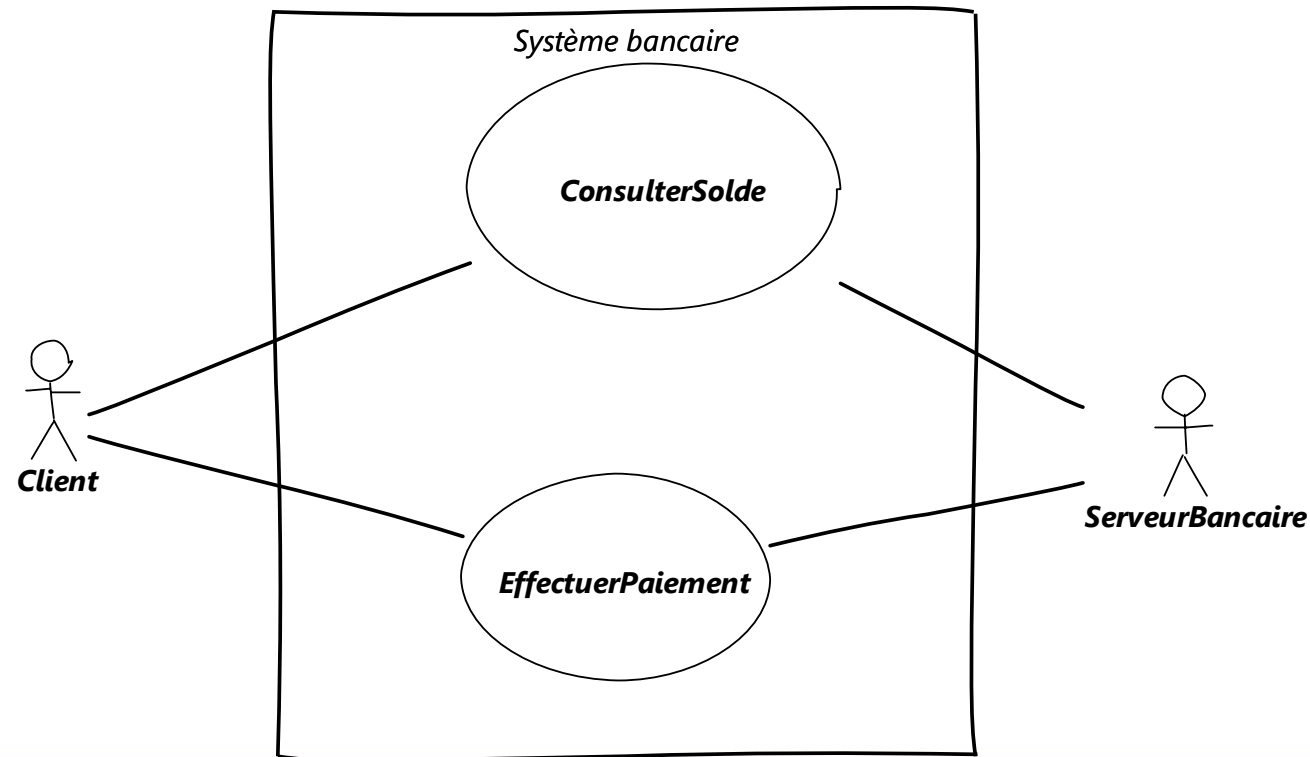


Le diagramme de cas d'utilisation



Acteur principal et secondaire

- L'acteur principal déclenche le cas d'utilisation (CU) tandis que l'acteur secondaire réagit au CU
- Généralement, l'acteur principal est dessiné à gauche et le secondaire à droite



Documentation d'un cas d'utilisation

CU:ChercherUnProduit (Nom du cas d'utilisation)

ID:1 (Numéro d'identification unique)

Description brève : chercher des produits afin d'alimenter éventuellement le panier ou faire des comparatifs entre les produits

Acteurs primaires : Client (liste des acteurs primaires)

Acteurs secondaires : (liste des acteurs secondaires)

Préconditions : le client doit être authentifié (l'état du système avant que le CU ne démarre)

Enchaînement principal

1. Le CU démarre quand le client clique sur le lien « chercher » (flux du CU)
2. Le client entre la référence ou le nom du produit
3. Le système affiche la liste des produits triés par nom

Postconditions : (l'état du système après le CU)

Enchaînement s alternatifs : (Flux se déclenchant sous certaines conditions)

Documentation d'un cas d'utilisation – Attributs

Attribut	Description
Nom du cas d'utilisation	<ul style="list-style-type: none">• Pas d'espace, utilise la convention UpperCamelCase• Décrit un comportement, utiliser des verbes• Le nom doit être parlant et aussi court que possible• Le nom est unique
ID	<ul style="list-style-type: none">• Identifiant numérique unique du CU
Description brève	Un paragraphe qui résume le CU
Acteurs primaires	Acteurs qui déclenchent le CU
Acteurs secondaires	Acteurs qui interagissent avec le CU
Preconditions	<ul style="list-style-type: none">• L'état du système avant le CU• Doit être vrai avant l'exécution du CU
Postconditions	<ul style="list-style-type: none">• L'état du système après le CU• Doit être vrai après l'exécution

Enchaînements des CU

- Un enchaînement est une suite d'actions numérotées
- La première action est formulée comme ceci : Le CU démarre quand
<l'acteur><fonction>
- Chaque action est formulée comme ceci : <quelque chose / quelqu'un>
<fonction>

Enchaînements des CU – Suite

- L'enchaînement principal est une séquence d'évènements décrivant le CU
- Le CU suit idéalement un enchainement mais peut avoir des **déviations**, ces déviations sont appelées **enchaînement alternatifs**
- Les enchaînements sont aussi appelés **scénarios**
- Il existe deux types de déviations : les déviations simples qui sont formalisées dans l'enchaînement et les déviations complexes qui sont formalisées dans un autre enchaînement

Exemples d'enchaînements

1. Le CU démarre quand le client clique sur le bouton « se connecter »
2. Le client entre son nom d'utilisateur dans la zone « login »
3. Le client entre son mot de passe dans la zone « mot de passe »
4. Le système vérifie la validité du nom d'utilisateur et du mot de passe

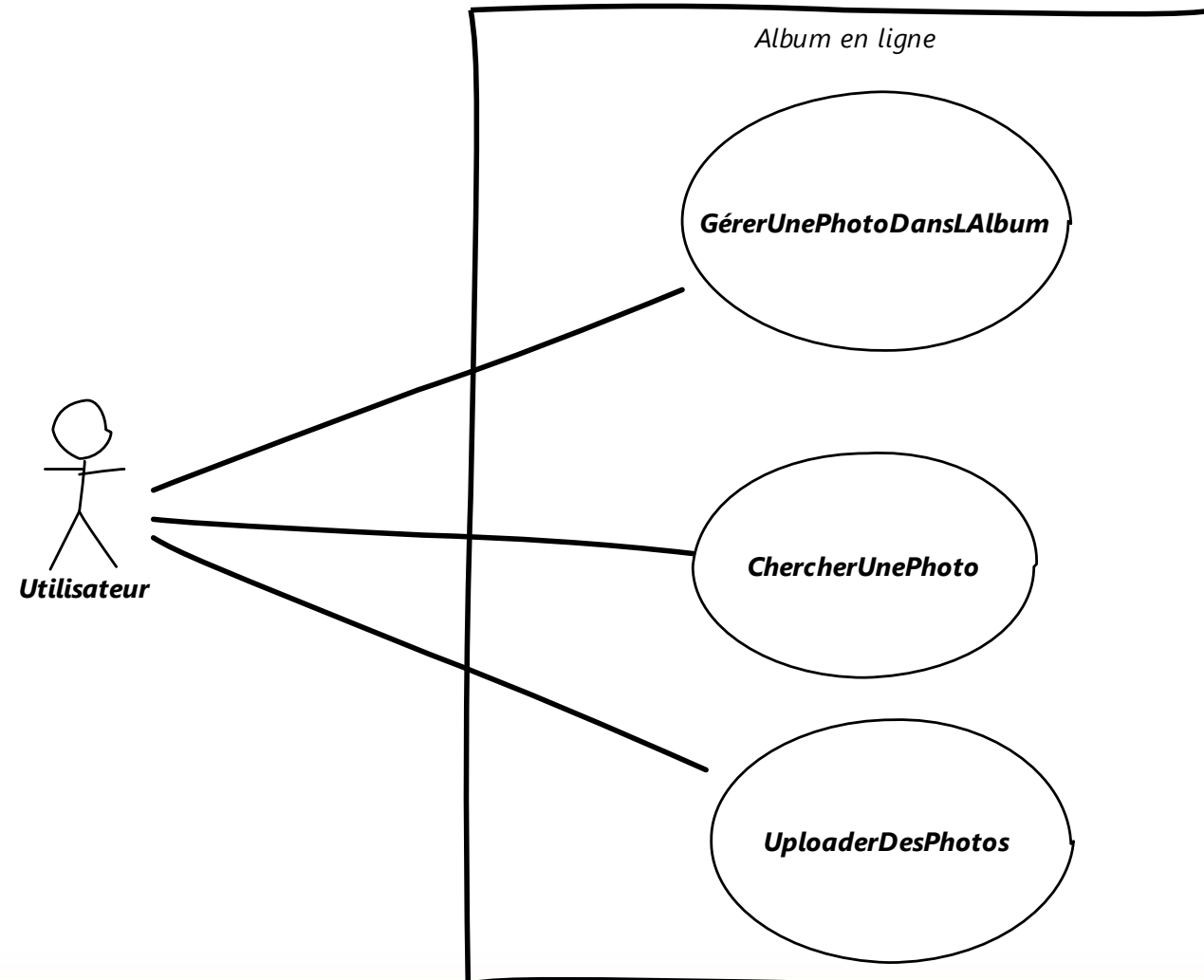
Enchaînements – Mauvaise Formulation

- « Les informations du client sont entrées et vérifiées »
- Qui entre les informations ?
- Dans quoi ?
- Quelles sont ces informations ?
- Ne pas s'exprimer en voix passive
- Se poser les questions : qui, quoi, quand et où ? Répondre avec précision.

Enchaînements – Contrôle de Flux

- Pour éviter de créer des enchaînements alternatifs, on peut faire des instructions de contrôle de flux à l'intérieur d'un enchaînement
- Le mot clé *si* : contrôle conditionnel
- Le mot clé *pour* : répétition pour tous les éléments d'un ensemble
- Le mot clé *tantque* : répétition selon une condition

Documentation d'un cas d'utilisation - Simulation



Simulation – Suite

CU: GérerUnePhotoDansLAlbum
ID: 2
Description brève : Gestion de l'album photo en ligne
Acteurs primaires : Utilisateurs
Acteurs secondaires : Aucun
Préconditions : <ul style="list-style-type: none">• le client doit être authentifié• L'album n'est pas vide
Enchaînement principal <ol style="list-style-type: none">1. Le CU démarre quand l'utilisateur clique sur une photo dans l'album2. Si l'utilisateur clique sur le bouton « supprimer »<ol style="list-style-type: none">2.1 Le système supprimer la photo de l'album3. Si l'utilisateur clique sur le bouton « N&B »<ol style="list-style-type: none">3.1 Le système transforme la photo en noir et blanc
Postconditions : Une photo supprimée ou modifiée
Enchaînement s alternatifs :

Simulation – Suite

CU: ChercherUnePhoto
ID: 3
Description brève : Rechercher une photo dans les albums en ligne
Acteurs primaires : Utilisateurs
Acteurs secondaires : Aucun
Préconditions : le client doit être authentifié
Enchaînement principal <ol style="list-style-type: none">1. Le CU démarre quand l'utilisateur clique sur le bouton « chercher »2. Le client entre les mots clé de recherche3. Le système trouve les photos indexées avec ces mots clés4. Si le résultat de recherche n'est pas vide<ol style="list-style-type: none">4.1. Pour chaque photo trouvée<ol style="list-style-type: none">4.1.1 Le système affiche la miniature de la photo4.1.2 Le système ajoute la photo aux statistiques de recherche5. Sinon<ol style="list-style-type: none">5.1 Le système affiche un message « aucune photo trouvée »
Postconditions :
Enchaînement s alternatifs :

Simulation – Suite

CU: UploaderDesPhotos
ID: 4
Description brève : Uploader des photos dans l'album
Acteurs primaires : Utilisateurs
Acteurs secondaires : Aucun
Préconditions : le client doit être authentifié
Enchaînement principal <ol style="list-style-type: none">1. Le CU démarre quand l'utilisateur clique sur le bouton « upload »2. Le client sélectionne sur son ordinateur les photos à uploader3. Tantque qu'il y a un espace libre dans le quota de l'utilisateur<ol style="list-style-type: none">3.1. Pour chaque photo à uploader<ol style="list-style-type: none">3.1.1 Enregistrer la photo dans l'espace de l'utilisateur3.1.2 Créer une miniature de la photo
Postconditions :
Enchaînement s alternatifs :

Enchaînements Alternatifs

- Les enchaînements décrivent souvent des actions très complexes
- Les enchaînements alternatifs sont des **déviations** ou des **interruptions** des enchaînements principaux
- Les enchaînements alternatifs peuvent être documentés à l'intérieur de la spécification ou séparément
- Un enchaînement alternatif peut ne pas revenir à l'enchaînement principal

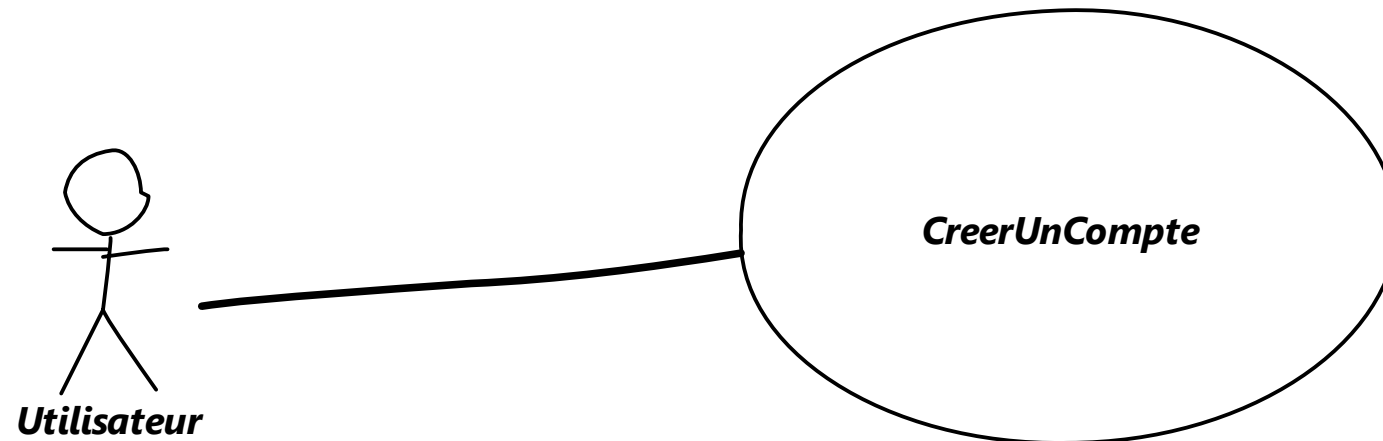
Enchaînements Alternatifs - Suite

- Les noms des enchaînements alternatifs sont formulés de la manière suivante : **Enchaînement Alternatif :**
NomCU:NomEnchaînementAlternatif
- L'id de l'enchaînement alternatif obéit à une numérotation hiérarchique. Par exemple, si l'ID un CU est 5, l'id de son premier enchaînement alternatif serait 5.1

Enchaînements Alternatifs – Démarrage

- L'EA peut démarrer au lieu de l'enchaînement principal. Dans ce cas, c'est l'utilisateur qui déclenche cette déviation.
- L'EA peut démarrer après une étape N de l'enchaînement principal
« *l'EA démarre après l'étape X de l'enchaînement principal* »
- L'EA peut démarrer à n'importe quel moment
- Pour chaque CU il y a un seul enchaînement principal et plusieurs EA
- Eviter de donner trop d'enchaînements alternatifs
- Regrouper les EA similaires

Documentation d'un cas d'utilisation - Simulation



Simulation – Suite

CU: CréerUnCompte
ID: 8
Description brève : Création d'un compte utilisateur
Acteurs primaires : Utilisateurs
Acteurs secondaires : Aucun
Préconditions : Aucune
Enchaînement principal <ol style="list-style-type: none">1. L'utilisateur clique sur le lien « s'inscrire »2. L'utilisateur entre son nom d'utilisateur3. L'utilisateur entre son mot de passe4. Le système valide le nom d'utilisateur et le mot de passe5. Le système crée un compte pour l'utilisateur
Postconditions : Le compte du client est créé
Enchaînements alternatifs : CompteExistant MotDePasseInvalide Annulation

Simulation – Suite

EA: CréerUnCompte:CompteExistant

ID: 8.1

Description brève : Le système informe l'utilisateur que le nom d'utilisateur a déjà été pris

Acteurs primaires : Utilisateurs

Acteurs secondaires : Aucun

Préconditions :

- Le compte entré par l'utilisateur est réservé par un autre utilisateur

Enchaînement principal

1. L'EA démarre à l'étape 4 de l'enchaînement principal
2. Le système affiche que le compte a déjà été donné à un autre utilisateur
3. L'EA revient à l'étape 2 de l'enchaînement principal

Postconditions : Aucune

Enchaînements alternatifs :

Simulation – Suite

EA: CréerUnCompte:Annuler
ID: 8.3
Description brève : L'utilisateur annule la création de son compte
Acteurs primaires : Utilisateurs
Acteurs secondaires : Aucun
Préconditions :
Enchaînement principal <ol style="list-style-type: none">1. L'EA démarre à n'importe quel moment2. L'utilisateur annule la création de son compte
Postconditions : Aucune
Enchaînements alternatifs :

Traçabilité des cas d'utilisation

- Un CU peut décrire une ou plusieurs spécifications
- Une spécification peut être représentée par un ou plusieurs CU
- La matrice de traçabilité définit les relations entre les spécifications et les CU

	CU ₁	CU ₂	CU ₃	CU ₄
S ₁		X		
S ₂	x		X	
S ₃				X
S ₄		x		

Quand utiliser les CU ?

- Le système est dominé par les spécifications fonctionnelles
- Le système est utilisé par plusieurs utilisateurs
- Le système a plusieurs interfaces

Modèles des Cas d'Utilisation



SECTION 3 – DÉBAT (10 MNS)

Modélisation Avancée des Cas d'Utilisation

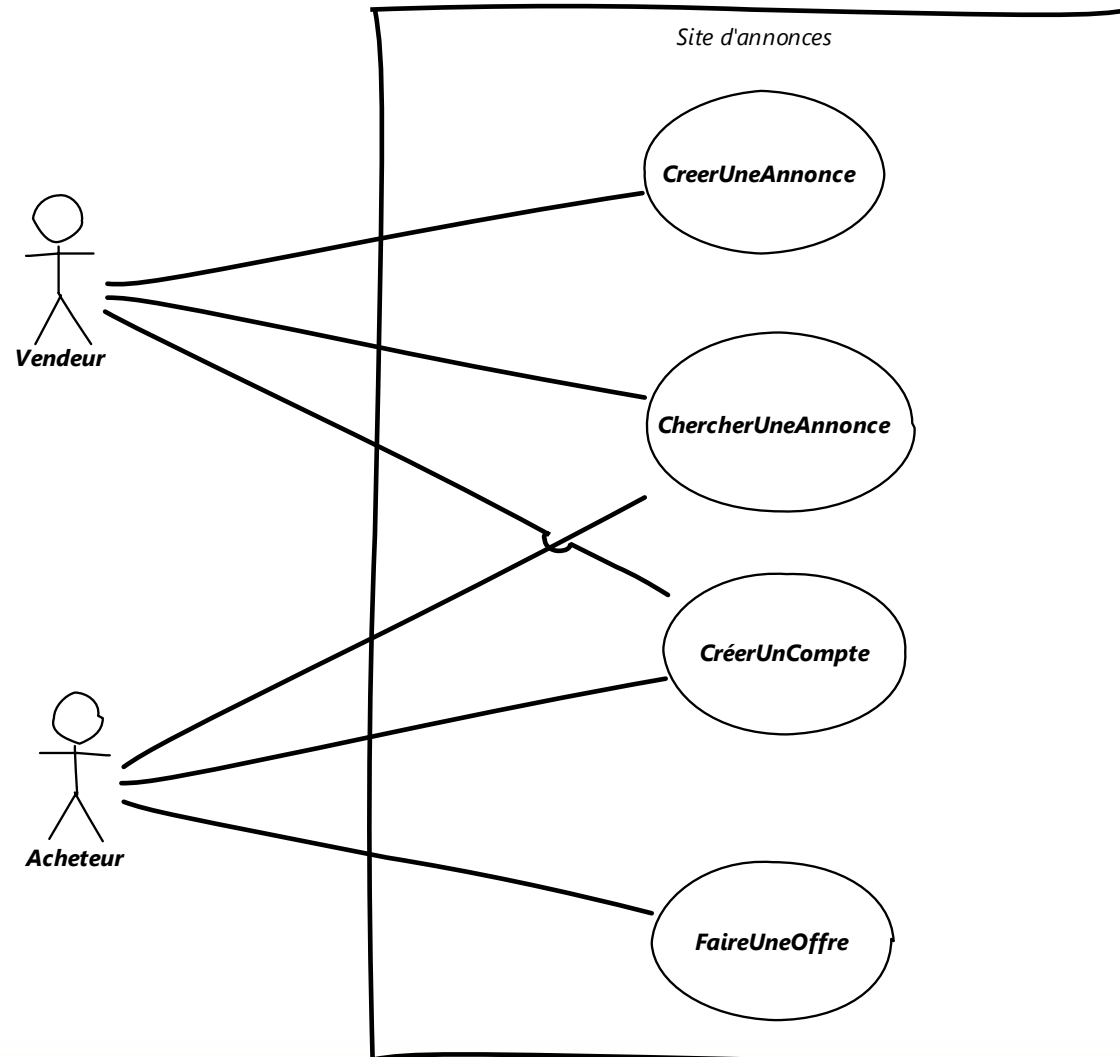


SECTION 4

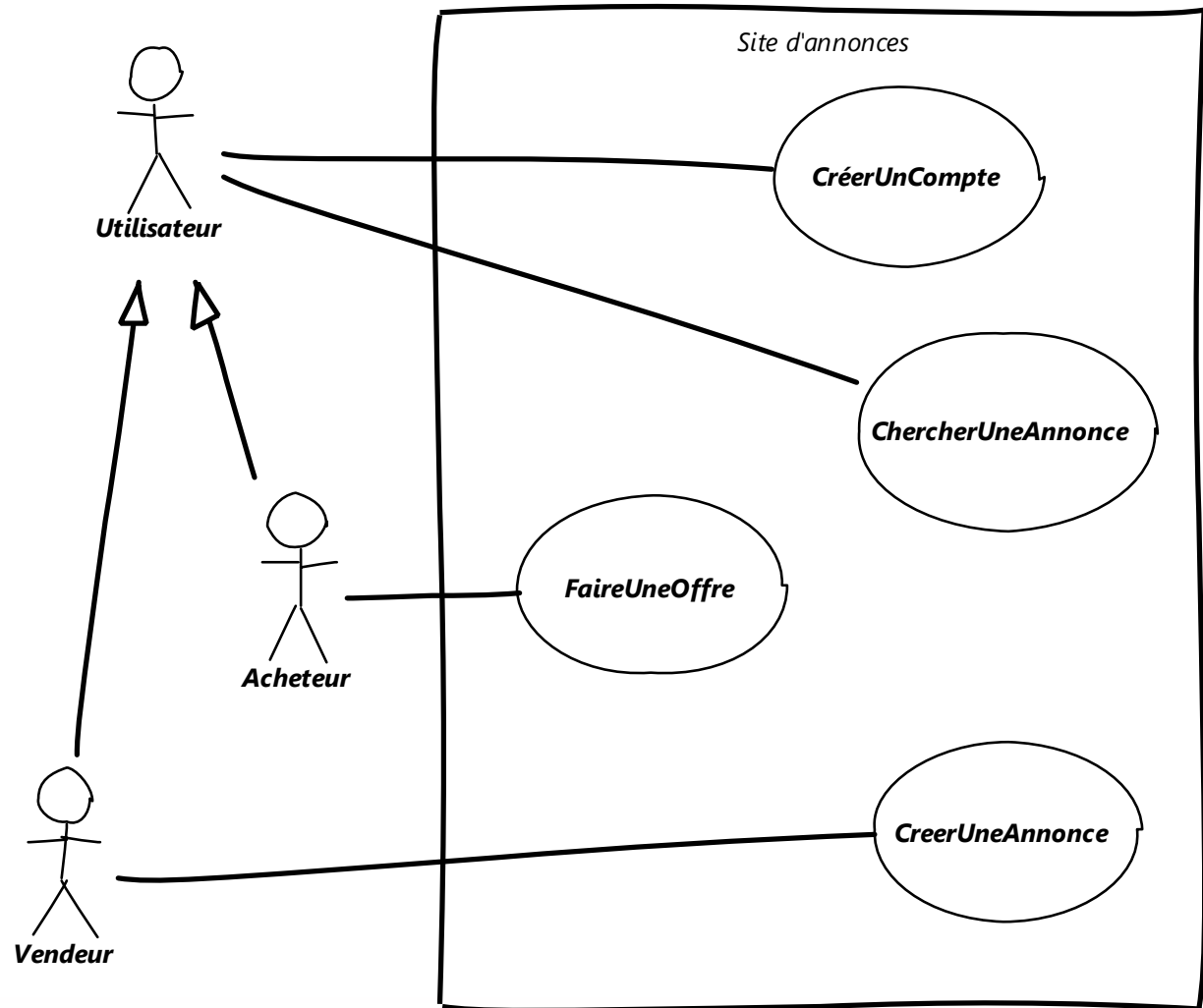
Généralisation des acteurs

- Des acteurs peuvent avoir beaucoup de CU en commun
- Un acteur peut être différent d'un autre acteur par quelques CU supplémentaires
- La généralisation répond au souci d'encombrement des diagrammes de CU
- La généralisation des CU simplifie non seulement la présentation mais aussi la sémantique des CU

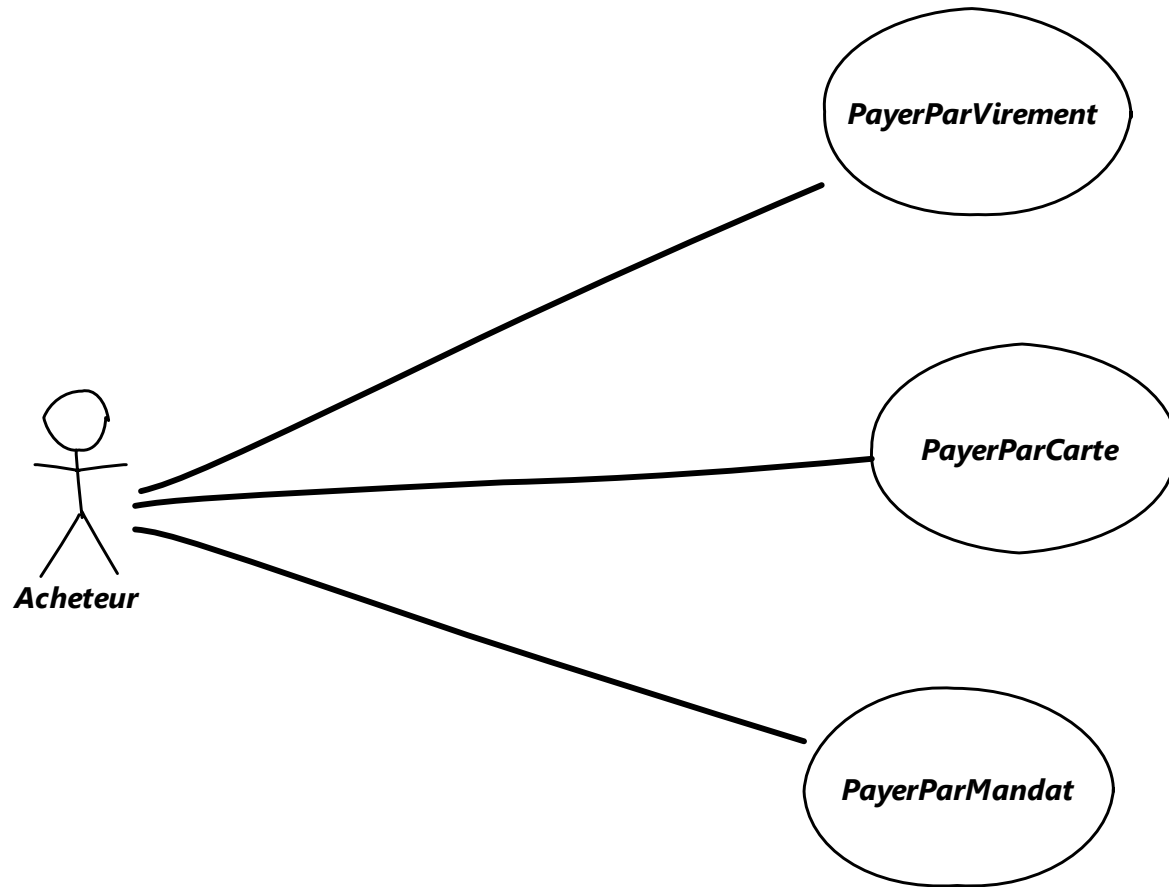
Généralisation des acteurs – Suite



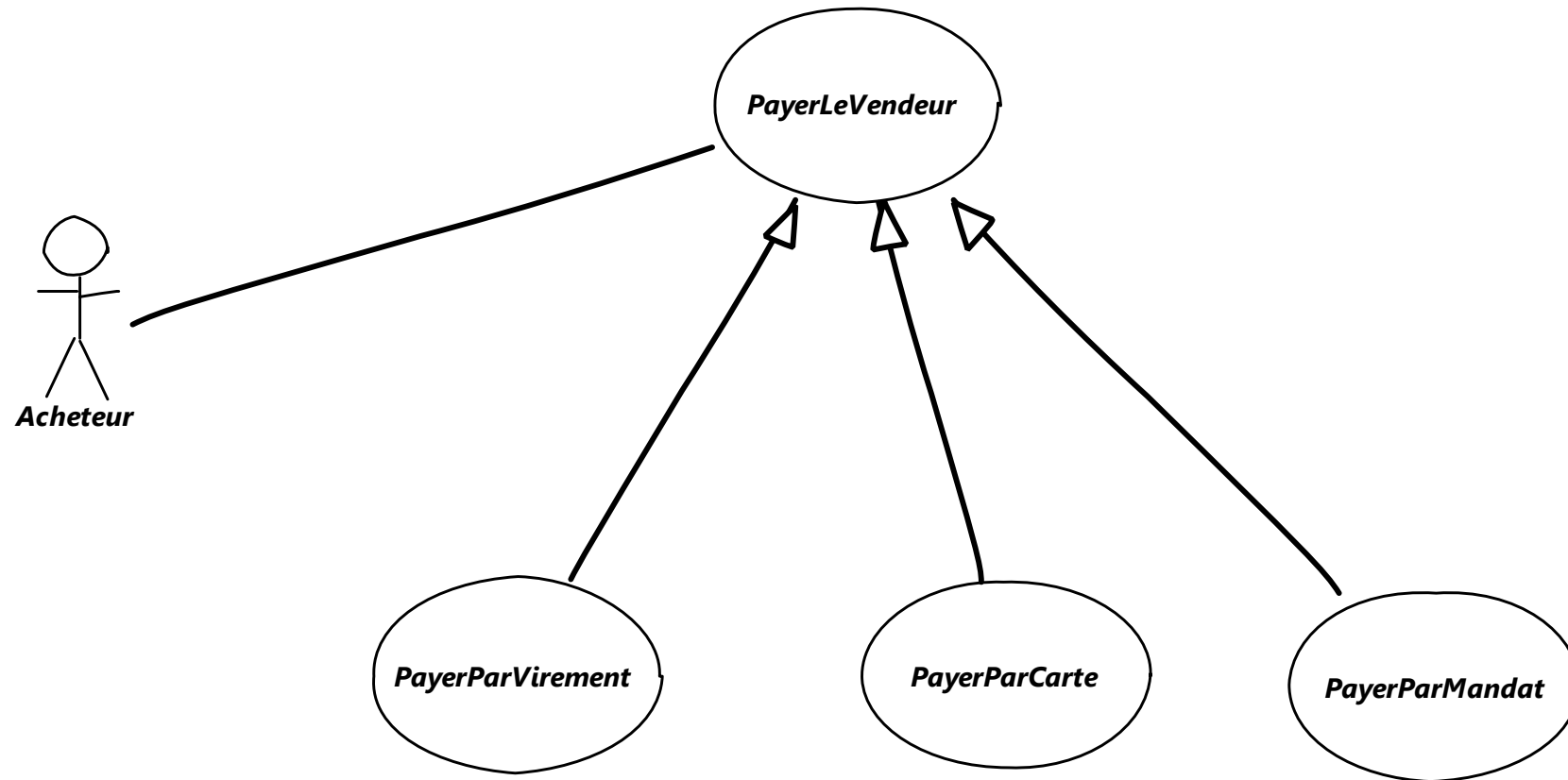
Généralisation des acteurs - Suite



Généralisation des CU

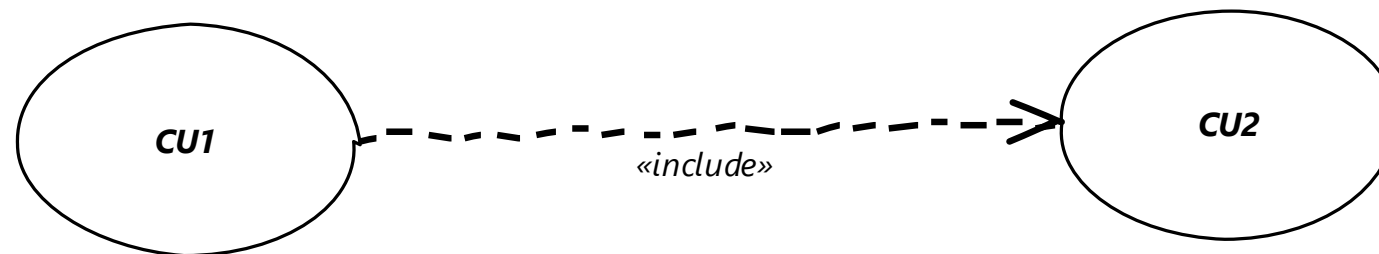


Généralisation des CU



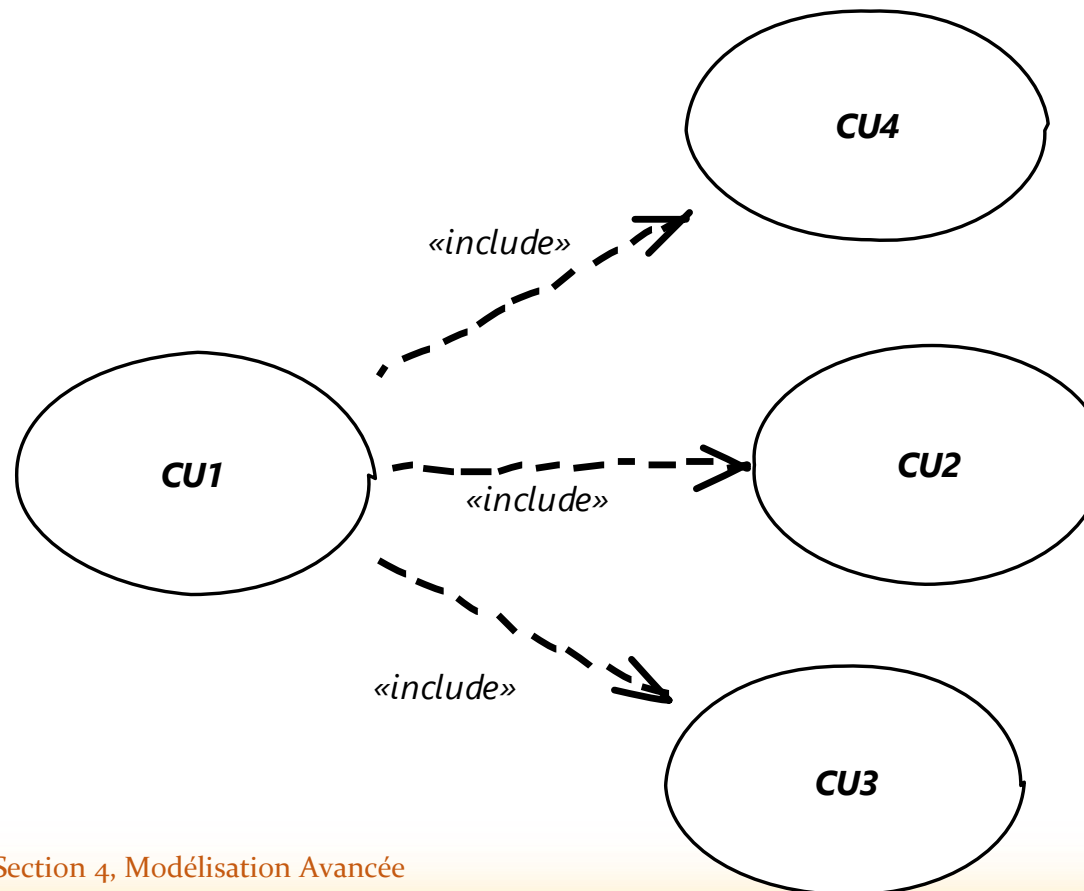
Inclusion des CU

- L'inclusion est une relation entre deux CU (CU₁ et CU₂). CU₁ est appelé **CU de base** et CU₂ est appelé **CU d'inclusion**.
- Quand l'enchaînement arrive au point d'inclusion, CU₁ ne s'exécute que lorsque CU₂ s'exécute.
- L'inclusion évite la répétition

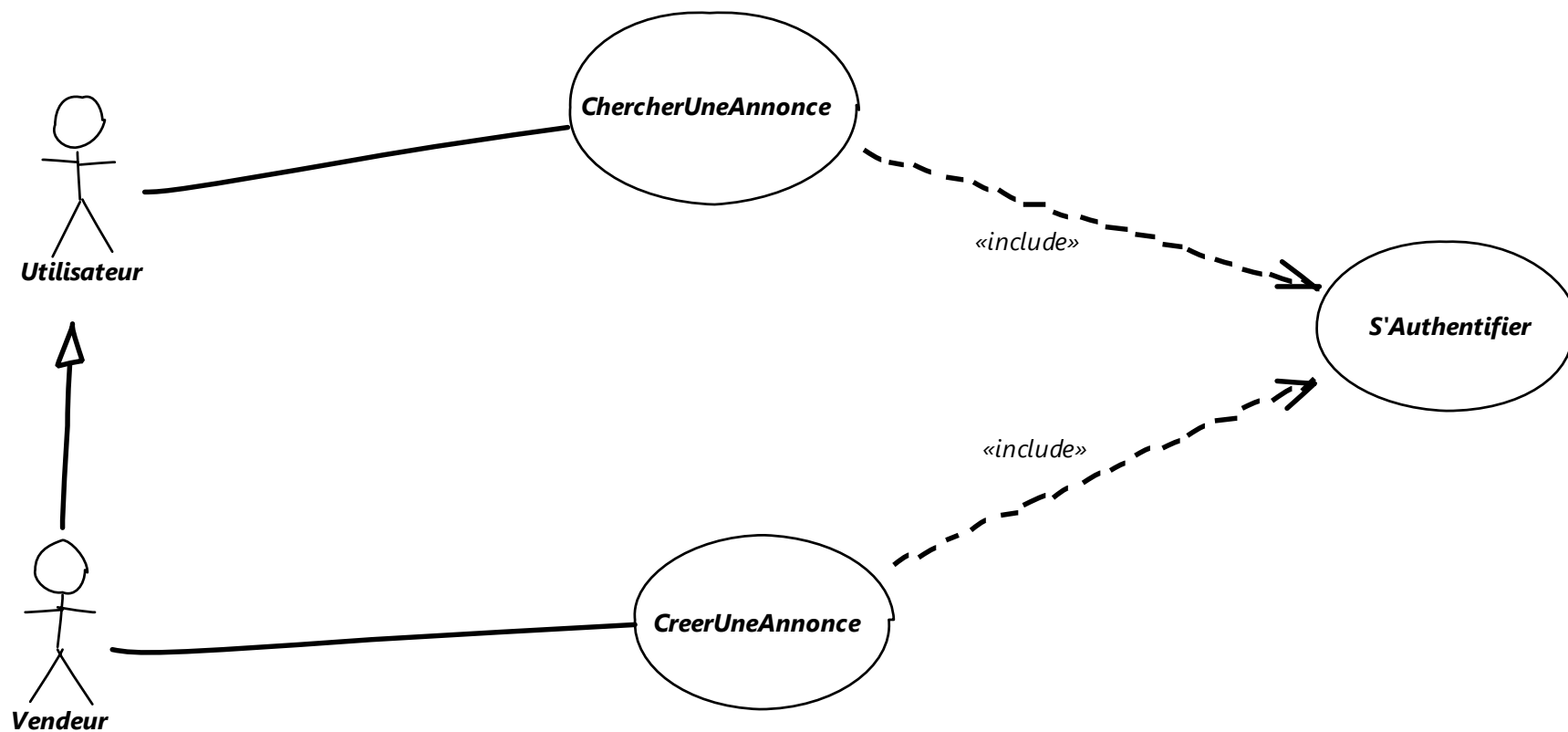


Inclusion des CU

- Un CU peut avoir plusieurs CU d'inclusion
- CU ne s'exécute que lorsque tous les CU d'inclusion ont été exécutés



Inclusion - Exemple



Inclusion des CU - Exemple

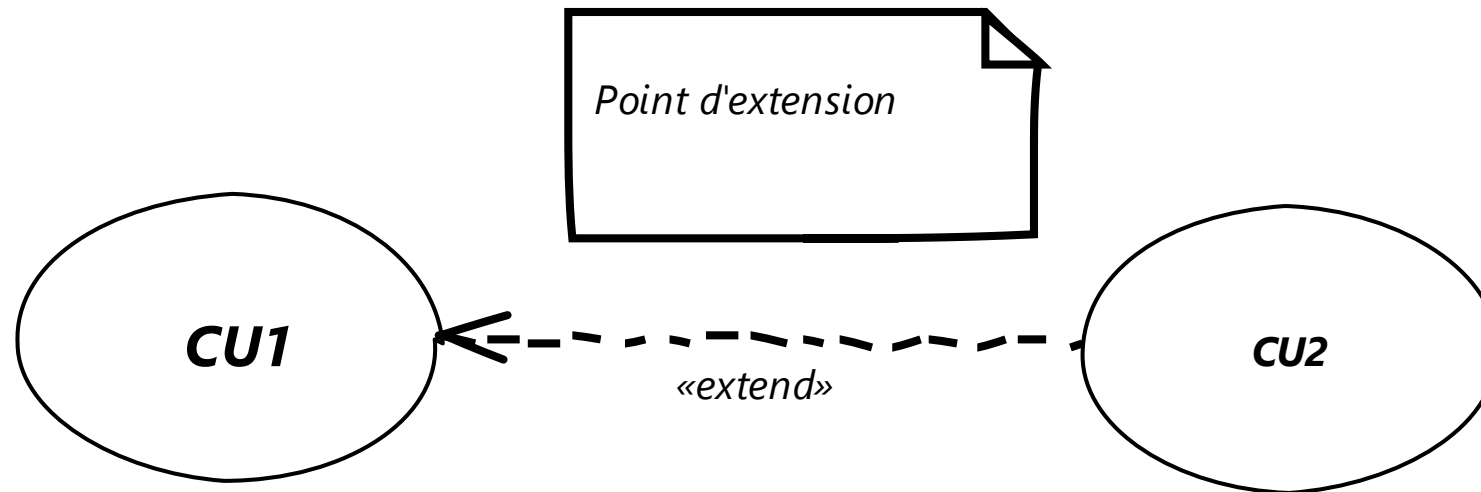
CU: S'Authentifier
ID: 15
Description brève : Authentification de l'utilisateur
Acteurs primaires : Utilisateurs
Acteurs secondaires : Aucun
Préconditions :
Enchaînement principal <ol style="list-style-type: none">1. Le CU commence lorsque l'utilisateur clique sur « se connecter »2. L'utilisateur entre son nom de compte et son mot de passe3. Le système valide le compte et le MDP4. Le système le dirige vers sa page par défaut
Postconditions : Utilisateur connecté »
Enchaînements alternatifs : CompteInvalide MDPInvalide

Inclusion des CU – Suite

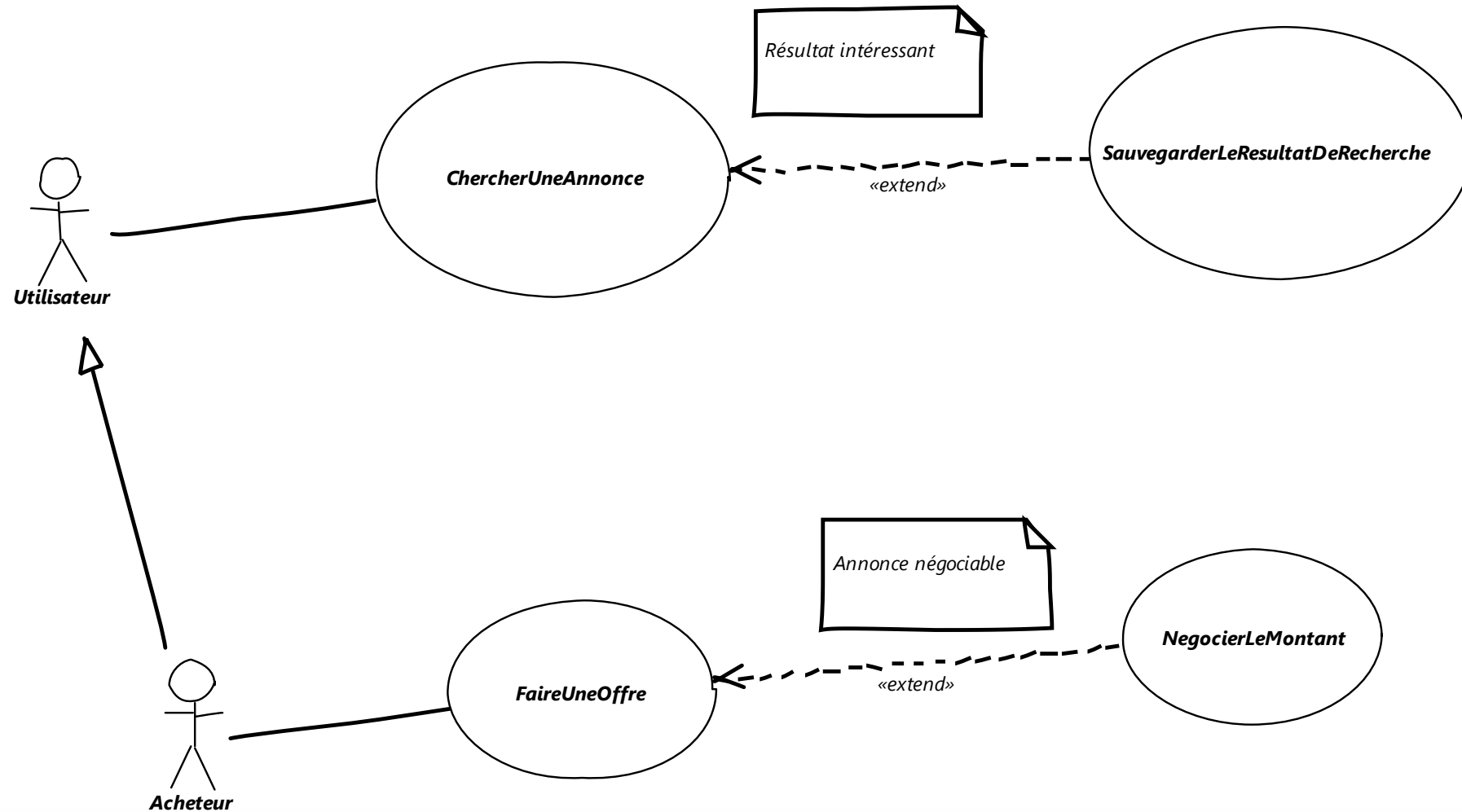
CU: CreerUneAnnonce
ID: 16
Description brève : Création d'une nouvelle annonce dans le site
Acteurs primaires : Vendeur
Acteurs secondaires : Aucun
Préconditions :
Enchaînement principal 1. Inclure (S'authentifier) 2. 3.
Postconditions :
Enchaînements alternatifs :

Extension des CU

- L'extension est une relation entre deux CU : CU₁ et CU₂
- CU₂ étend CU₁ par un comportement optionnel qui a lieu sous une certaine condition
- Cette condition est appelé « point d'extension »



Extension des CU



Extension des CU - Suite

CU: FaireUneOffre
ID: 16
Description brève : L'acheteur désire acquérir le produit de l'annonce et veut faire une offre au vendeur
Acteurs primaires : Acheteur
Acteurs secondaires : Aucun
Préconditions :
Enchaînement principal 1. L'utilisateur consulte les détails de l'offre Point d'extension : NégocierLeMontant (si l'annonce est négociable) 2. Envoyer le montant au vendeur
Postconditions :
Enchaînements alternatifs :

Best practices

Les acteurs doivent être appelés en utilisant leur rôle au singulier

Ne pas montrer le comportement, plutôt montrer la fonctionnalité

Les diagrammes ne doivent pas être trop encombrés. Par exemple, au maximum 15 CU par diagramme

Tous les CU doivent avoir le même niveau d'abstraction

Les spécifications des CU doit être de la même taille (1/2 page à 1 page)

Si les CU sont trop grands, utiliser les inclusions / extensions / généralisations

Sortir les éléments redondants en des CU inclus

Sortir les éléments conditionnels en des CU d'extension

Les CU sont appelé en utilisant des verbes

Ne pas schématiser les communications entre acteurs

Modélisation Avancée des Cas d'Utilisation

SECTION 4, DÉBAT

Bibliographie

- Software Engineering Right Edition, Ian Sommerville, Addison Wesley, 2007
- Software Development and Professional Practice, John Dooley, APress, 2010
- Software Development Life Cycle (SDLC), Togi Berra, course session 2
- Rational Unified Process - Best Practices for Software Development Teams, IBM / Rational, 1998