

Analyse et Conception avec UML

Les diagrammes de cas d'utilisation

Licence Professionnelle IDSE
2015-2016

IUT Nice-Sophia Antipolis

Repris du cours de Mireille Blay-Fornarino
<http://mireilleblayfornarino.i3s.unice.fr/>

Du CdCF à la modélisation



Du CdCF à la modélisation



Du CdCF à la modélisation

des échanges et une collaboration à distance. C'est le succès des technologies de convergence des réseaux, des médias et des modes de formation en réseau.

EMIL : **Environnement d'enseignement d'une matière par l'usage d'une langue étrangère**
 Cet acronyme désigne l'approche consistant à utiliser l'apprentissage des langues à l'usage de l'enseignement d'une matière scolaire ou professionnelle autre que la langue maternelle, par exemple l'histoire ou la géographie de la langue ou le traitement de l'écrit.

Ergonomie
 L'ergonomie peut être définie comme la science de tout ce qui permet d'optimiser la qualité humaine, notamment de l'interaction homme-machine de travail. En informatique, l'ergonomie est à ce que les informations affichées sur l'écran, les interfaces graphiques, soient accessibles pour les utilisateurs.

Functionalité
 C'est une désignation dérivée de celle de service d'une application dans le langage informatique. Exemple : l'ergonomie peut être vue comme une fonctionnalité pour permettre la construction d'un tableau de bord ou de l'édition d'un rapport.

Flux de travail ou workflow
 Il s'agit le plus souvent de schématiser les tâches à accomplir entre les différents acteurs d'un processus, les tâches, les modes de validation, et fournir à chacun des acteurs les informations nécessaires pour la réalisation de sa tâche. Pour un processus de publication en ligne par exemple, il s'agit de la modélisation des tâches de l'embauche de la chaîne éditoriale.

Global
 Un global, souvent nommé un informatique global, désigne un outil pour désigner un modèle de conception de logiciel ou de présentation des données ou un pattern de mise en page de site web pour images et textes. Il permet à des développeurs sans qualification particulière en informatique d'élaborer un site internet en contenu.

Langue cible
 La langue cible est la langue dont l'apprentissage est visé, en plus de la langue de l'enseignement pédagogique ou celle à laquelle on prête cette attention.

Langue source
 La langue source est la langue dans laquelle le support pédagogique ou la ressource sont présentés.

Creative Commons
 Les Creative Commons constituent un ensemble de licences régissant les conditions de réutilisation, ainsi de distribution (diverses licences) d'œuvres multimedias diffusées sur Internet.

Logiciel libre
 Un logiciel libre est fourni avec son code source. Ses utilisations, sa modification et sa diffusion des versions modifiées sont autorisées. Un logiciel libre n'est pas forcément gratuit, même si c'est souvent le cas.

Matériau pédagogique
 Ressource dans la langue cible qui présente des outils pour l'apprentissage et l'enseignement de la langue non que l'apprentissage de la langue, à l'usage dans un TAO, de la grammaire, du lexique, de la syntaxe, de la phonétique, de la prononciation, de la culture, de la littérature et des genres d'art.

Navigation
 La navigation désigne l'ensemble des fonctionnalités qui permettent l'exploration, le repérage de l'interface dans un site ou une application.

Plan
 C'est une désignation des modèles implicites dans la même processus de formation et appartenant à la même communauté professionnelle.

Processus
 Ce terme désigne un schéma d'orchestration d'activités qui forme l'organisation des activités d'un apprenant ou d'un enseignant pédagogique.

Glossaire



Du CdCF à la modélisation

Glossaire

des échanges et une collaboration à distance. C'est le service qui permet de communiquer à distance, de recevoir, de transmettre et de stocker des données, des images et des modèles de formation en temps réel.

EMIL : Accroche d'intégration d'une matière par l'usage d'un langage étranger.

Cet accroche (digne) l'approche consiste à utiliser l'apprentissage de langues à travers l'apprentissage d'une matière via un moyen d'une langue autre que la langue maternelle, par exemple l'anglais ou l'espagnol, la géographie ou le français ou le traitement de l'anglais.

Ergonomie : L'ergonomie peut être définie comme la science de l'homme pour concevoir et améliorer le travail, l'organisation et l'efficacité des systèmes de travail. En informatique, l'ergonomie est ce qui se rapporte à l'interaction humaine avec les ordinateurs.

Functionalité : Comme l'usage ou l'usage de services d'une application dans le langage informatique. Exemple : l'usage d'Internet peut être vu comme une fonctionnalité, tout comme la construction d'un système de distribution ou l'édition d'un rapport.

Flux de travail ou workflow : Il s'agit de l'ensemble de tâches, les tâches à accomplir entre les différents acteurs d'un processus, les tâches, les modes de validation, et l'ordre à l'exécution des tâches et des données qui sont nécessaires pour la réalisation de ce flux. Pour un processus de publication en ligne par exemple, il s'agit de la modélisation des tâches de l'ensemble de la chaîne éditoriale.

Global : Un global, souvent nommé un informatique simple langagier global pour désigner un modèle de conception de langage ou de programmation des données ou un pattern de mise en page ou flux peut être image et texte, il permet à des personnes sans qualification particulière en informatique d'obtenir un site internet en continu.

Langage cible : Un langage cible est le langage dont l'apprentissage est visé, en plus du langage de formation pédagogique ou celle à laquelle on s'adresse cette matière.

Langage source : Le langage source est le langage dans lequel le support pédagogique ou la matière sont présentés.

Creative Commons : Les Creative Commons constituent un ensemble de licences rigoureuses et cohérentes de distribution, ainsi de distribution (d'autres instruments) (d'autres multimedias) (autres ou Internet).

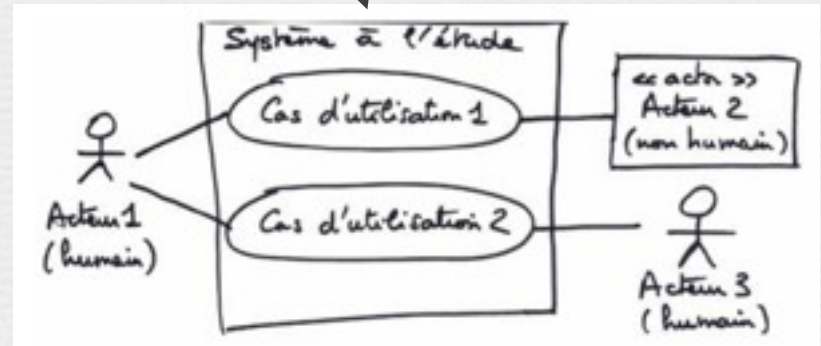
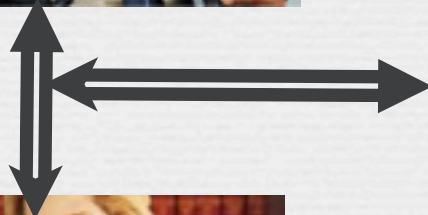
Logiciel libre : Un logiciel libre est fourni avec son code source. Son utilisation, sa modification et sa diffusion des autres logiciels sont autorisées. Un logiciel libre n'est pas forcément gratuit, même si c'est souvent le cas.

Méthodes multilingues : Ressources dans le langage cible qui peuvent être utiles pour l'apprentissage et l'enseignement de langues non qui n'ont pas été adaptées, à l'origine, dans ce but. Ex : grammaire, dictionnaire, notes de cours, logiciels, guides pour les étudiants et les professeurs.

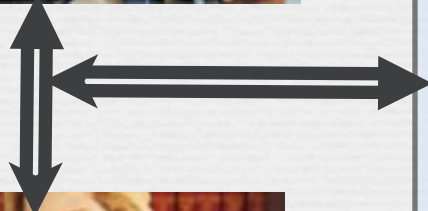
Navigation : La navigation désigne l'ensemble des fonctionnalités qui permettent l'exploration, le repérage et l'accès à des sites ou applications.

Page : Ce terme désigne des unités multiples dans le même processus de formation et appartenant à la même communauté professionnelle.

Parcours : Ce terme désigne un schéma d'enseignement d'activités qui sont dirigées les activités d'un apprenant ou une d'un contenu pédagogique.



Du CdCF à la modélisation



des échanges et une collaboration à distance. C'est le service qui permet de communiquer des données, des images et des modèles de formation en réseau.

EMIS : Accroche d'un événement d'une manière par l'usage d'une langue étrangère.
 C'est une manière d'être qui consiste à utiliser l'apprentissage de langues à travers l'enseignement d'une matière scolaire ou professionnelle autre que la langue maternelle, par exemple l'histoire ou la géographie en se servant de la langue de l'enseignant ou de l'apprenant.

Ergonomie
 L'ergonomie peut être définie comme la science de l'homme pour mieux concevoir et améliorer l'efficacité, le bien-être et la santé des personnes travaillant dans les environnements de travail.
 En informatique, l'ergonomie est ce qui se rapporte à l'interaction humaine-machine, à l'interface graphique, à l'expérience utilisateur, à l'usabilité.

Fonctionnalité
 C'est une manière de concevoir des services d'une application dans le langage informatique. Exemple : l'interface d'un logiciel peut être vue comme une fonctionnalité pour permettre à l'utilisateur d'interagir avec le système de manière intuitive et efficace.

Flux de travail en activité
 Il s'agit de l'ensemble des tâches et des activités qui sont effectuées au cours d'un processus, de la conception à la réalisation, en passant par la validation, le suivi et l'évaluation des données et des résultats. C'est un processus de planification et de gestion de la réalisation des tâches de manière à optimiser les ressources.

Global
 Un global, souvent nommé un langage global ou langage universel pour désigner un modèle de conception de logiciel ou de programmation des données et un langage de haut niveau qui peut être utilisé et adapté à divers environnements informatiques et informatiques.

Langue cible
 La langue cible est la langue dans laquelle l'apprenant est censé apprendre à parler ou à écrire, par opposition à la langue maternelle ou à la langue de la langue d'origine.

Langue source
 La langue source est la langue dans laquelle le support pédagogique ou la ressource sont présentés.

Creative Commons
 Les Creative Commons constituent un ensemble de licences qui permettent de contrôler la réutilisation, la distribution et la diffusion des œuvres créatives (textes, images, vidéos, musique, etc.).

Logiciel libre
 Un logiciel libre est un logiciel dont le code source, les algorithmes, la modification et la diffusion des données sont libres et accessibles. Un logiciel libre peut être révisé, amélioré et redistribué.

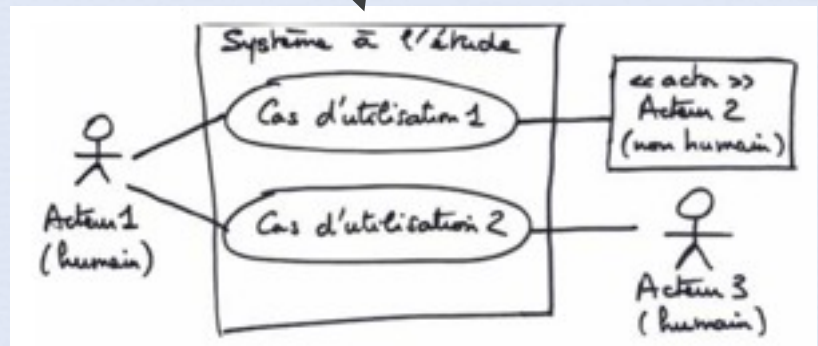
Matériau pédagogique
 Un matériau pédagogique est un support de travail utilisé pour l'apprentissage et l'enseignement de langues, de langues ou de disciplines, à l'usage de la langue ou de la langue d'origine, par exemple, les livres, les vidéos, les logiciels, les jeux, etc.

Navigation
 La navigation désigne l'ensemble des fonctionnalités qui permettent à l'utilisateur de parcourir et d'explorer un site web ou une application.

Page
 Une page est une unité de contenu dans un document électronique, un document ou un document, qui est accessible et utilisable par les utilisateurs.

Processus
 Un processus est une séquence d'activités qui sont effectuées de manière à atteindre un objectif ou à résoudre un problème.

Glossaire



Plan du cours

- Utilisation d'un dictionnaire du domaine (glossaire)

- Des cas d'utilisations (Use-cases/UC)
 - Acteurs, use-cases, system UML

- Processus de construction des Uses-case

- Compléments

UML au travail : Système d'inscriptions

L'université ESU désire automatiser son système d'inscription

- ▶ Le chef du service des inscriptions établit le programme des cours pour un semestre
- ▶ Un cours peut être offert plusieurs fois
- ▶ Les étudiants doivent sélectionner 4 cours primaires et 2 cours secondaires dans le catalogue des cours proposés
- ▶ Dès qu'un étudiant s'est inscrit pour un semestre, le système de facturation est notifié
- ▶ Les étudiants peuvent utiliser le système pour modifier leurs choix pendant une certaine période de temps après leur inscription
- ▶ Les enseignants utilisent le système pour consulter leur emploi du temps (tableau d'activités en fonction des cours qui tournent)
- ▶ Les utilisateurs du système d'inscription reçoivent des mots de passe qui sont nécessaire à la procédure d'identification

UML au travail : Guichet automatique de banque

Le guichet automatique d'une banque (GAB) offre les services suivants :

- Distribution d'argent à partir d'une carte de la banque ou d'une carte Visa.
- Consultation de solde de compte, dépôt en numéraire et dépôt de chèques pour les clients de la banque porteurs d'une carte de la banque.

De plus,

- Toutes les transactions sont sécurisées.
- Il est parfois nécessaire de recharger le distributeur, .



UML au travail : Une ludothèque

- (1) Nous voulons informatiser une ludothèque pour favoriser la consultation des jeux proposés par la ludothèque.
- (2) Les adhérents peuvent emprunter des jeux en s'adressant à un conseiller qui enregistre l'emprunt.
- (3) Les jeux empruntés sont rendus à un conseiller....
- (4) Un adhérent peut réserver des jeux. Une réservation précise l'emprunteur, le jeu et la date de la demande de réservation. L'adhérent est averti quand le jeu revient en rayon.
- (5) Pour organiser un événement le conseiller spécialisé doit alors donner les informations suivantes : les jeux à tester, le nombre maximal et minimal de participants attendus, la date, et l'heure de début de l'événement.
- (6) Un adhérent peut s'inscrire pour participer à un événement à condition qu'il y ait encore de la place.
- (7) Un adhérent peut payer sa cotisation en ligne par un système de paiement externe
- (8) Un internaute peut consulter les jeux et s'inscrire.

Plan du cours

- ➔ Utilisation d'un dictionnaire du domaine (glossaire)

- Des cas d'utilisations (Use-cases/UC)
 - Acteurs, use-cases, system UML

- Processus de construction des Uses-case

- Compléments

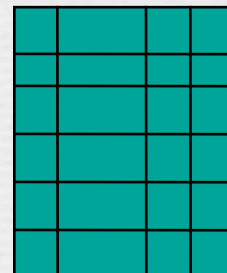
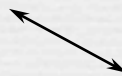
Intérêt du glossaire

- Outil de dialogue
- Informel, évolutif, simple à réaliser
- Etablir et figer la terminologie
 - Permet de figer la terminologie du **domaine d'application**.
 - Constitue le point d'entrée et le référentiel initial de l'application ou du système.

Homonymie

Synonymie

Polysémie



Intérêt du glossaire

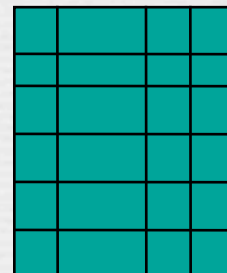
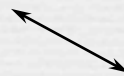
- Outil de dialogue
- Informel, évolutif, simple à réaliser
- Etablir et figer la terminologie
 - Permet de figer la terminologie du **domaine d'application**.
 - Constitue le point d'entrée et le référentiel initial de l'application ou du système.



Homonymie

Synonymie

Polysémie



UML au travail : Système d'inscriptions

L'université ESU désire automatiser son système d'inscription

- ▶ Le chef du service des inscriptions établit le programme des cours pour un semestre
- ▶ Un cours peut être offert plusieurs fois
- ▶ Les étudiants doivent sélectionner 4 cours primaires et 2 cours secondaires dans le catalogue des cours proposés
- ▶ Dès qu'un étudiant s'est inscrit pour un semestre, le système de facturation est notifié
- ▶ Les étudiants peuvent utiliser le système pour modifier leurs choix pendant une certaine période de temps après leur inscription
- ▶ Les enseignants utilisent le système pour consulter leur emploi du temps (tableau d'activités en fonction des cours qui tournent)
- ▶ Les utilisateurs du système d'inscription reçoivent des mots de passe qui sont nécessaire à la procédure d'identification

UML au travail : Système d'inscriptions

L'université ESU désire automatiser son système d'inscription

- ▶ Le chef du service des inscriptions établit le **programme des cours** pour un semestre
 - ▶ Un **cours** peut être offert plusieurs fois
- ▶ Les étudiants doivent sélectionner 4 **cours primaires** et 2 **cours secondaires** dans le **catalogue des cours proposés**
- ▶ Dès qu'un étudiant s'est inscrit pour un semestre, le système de facturation est notifié
- ▶ Les étudiants peuvent utiliser le système pour modifier leurs choix pendant une certaine période de temps après leur inscription
- ▶ Les enseignants utilisent le système pour consulter leur emploi du temps (**tableau d'activités** en fonction des cours qui tournent)
- ▶ Les utilisateurs du système d'inscription reçoivent des mots de passe qui sont nécessaire à la **procédure d'identification**

UML au travail : Une ludothèque

- Les adhérents peuvent emprunter des jeux en s'adressant à un conseiller qui enregistre l'emprunt.
- Les jeux empruntés sont rendus à un conseiller....
- Un adhérent peut réserver des jeux. Une réservation précise l'emprunteur, le jeu et la date de la demande de réservation. L'adhérent est averti quand le jeu revient en rayon.
- Pour organiser un événement le conseiller spécialisé doit alors donner les informations suivantes : les jeux à tester, le nombre maximal et minimal de participants attendus, la date, et l'heure de début de l'événement.
- Un adhérent peut s'inscrire pour participer à un événement à condition qu'il y ait encore de la place.
- Un adhérent peut payer sa cotisation en ligne par un système de paiement externe
- Un internaute peut consulter les jeux et s'inscrire.

UML au travail : Une ludothèque

- Les **adhérents** peuvent emprunter des **jeux** en s'adressant à un **conseiller** qui enregistre l'**emprunt**.
- Les jeux empruntés sont rendus à un conseiller....
- Un adhérent peut réserver des jeux. Une **réservation** précise l'emprunteur, le jeu et la date de la demande de réservation. L'adhérent est averti quand le jeu revient en rayon.
- Pour organiser un **événement** le conseiller spécialisé doit alors donner les informations suivantes : les jeux à tester, le nombre maximal et minimal de participants attendus, la date, et l'heure de début de l'événement.
- Un adhérent peut s'inscrire pour participer à l'événement en en faisant la demande à un conseiller spécialisé, à condition qu'il y ait encore de la place.
- Un adhérent peut payer sa cotisation en ligne par un **système de paiement externe**
- Un internaute peut consulter les jeux et s'inscrire.

UML au travail : Une ludothèque

Adhérents : Personne connue du système *par son nom, prénom, date de naissance, adresse postale, date du dernier paiement de la cotisation et adresse email éventuelle.*

Conseiller : Personne *identifiée* qui a des droits sur le système.

Réservation : Une réservation précise l'emprunteur, le jeu et la date de la demande de réservation. L'adhérent est averti quand le jeu revient en rayon.

Evénement : temps de démonstration et de jeux organisé à l'avance. Il précise les informations suivantes : les jeux à tester, le nombre maximal et minimal de participants attendus, la date, et l'heure de début de l'événement.

Jeux : Un jeu est caractérisé par un identifiant, un nombre de joueurs, ...

vocabulaire....

Base de données

Implémentation

Cahier des charges

.....

vocabulaire....

Base de données

Implémentation

Cahier des charges

.....

Est-ce le vocabulaire
du domaine ?



Plan du cours

- ✓ Utilisation d'un dictionnaire du domaine (glossaire)

- ➔ Des cas d'utilisations (Use-cases/UC)
 - Acteurs, use-cases, system UML

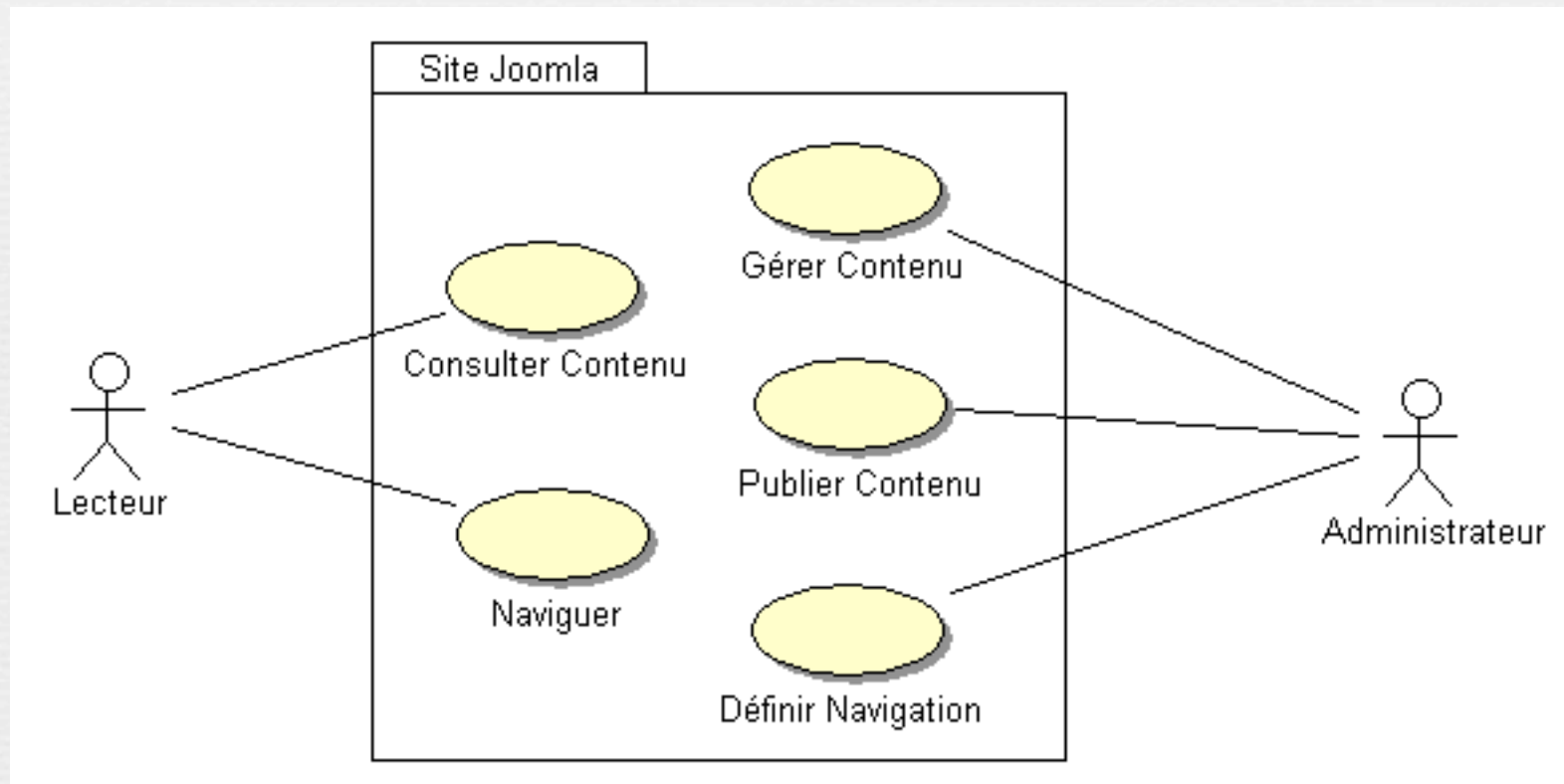
- Processus de construction des Uses-case

- Compléments

Les diagrammes de cas d'utilisation



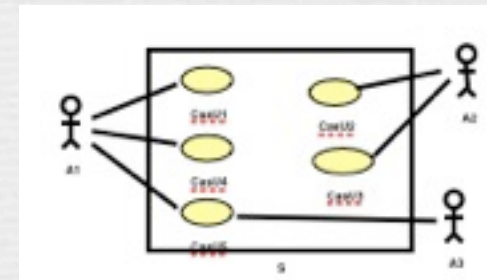
Une notation très simple, compréhensible par tous



cf. [http://linformalibre.f2lt.fr/index.php?title=Comprendre Joomla à l'aide d'UML](http://linformalibre.f2lt.fr/index.php?title=Comprendre_Joomla_%C3%A0_l%27aide_d%27UML)

Les diagrammes de cas d'utilisation

- Une des notations d'UML (use-cases)
- **But :**
 - ▶ définir le système **du point de vue des utilisateurs**
 - ▶ définir les limites précises du système
- Notation très simple, compréhensible par tous
- Permet de structurer :
 - ▶ les besoins (cahier des charges)
 - ▶ le reste du développement
 - ▶ ...
- Les cas d'utilisation sont nommés en utilisant la terminologie décrite dans le dictionnaire



Les diagrammes de cas d'utilisation

Une des notations d'UML (use-cases)

But :

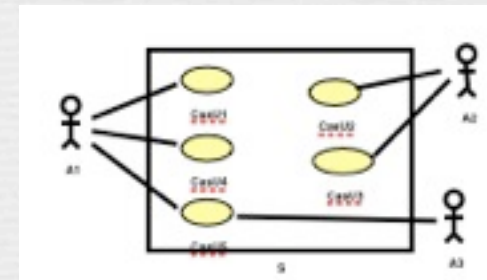
- ▶ définir le système **du point de vue des utilisateurs**
- ▶ définir les limites précises du système

Notation très simple, compréhensible par tous

Permet de structurer :

- ▶ les besoins (cahier des charges)
- ▶ le reste du développement
- ▶ ...

Les cas d'utilisation sont nommés en utilisant la terminologie décrite dans le dictionnaire



Plan du cours

- ✓ Utilisation d'un dictionnaire du domaine (glossaire)

- ✓ Des cas d'utilisations (Use-cases/UC)
 - Acteurs, use-cases, system UML

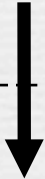
- ➔ Processus de construction des Uses-case
 - Compléments

Processus d'écriture des UC

Trouver les acteurs



Trouver les UC



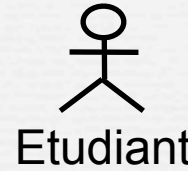
Décrire les UC



Détailer les UC

Processus d'écriture des UC

Trouver les acteurs



Trouver les UC

Décrire les UC

Détailler les UC

Processus d'écriture des UC

Trouver les acteurs



Trouver les UC

Enregistrer des cours

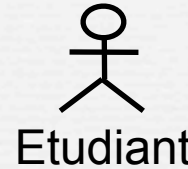
Brève description: Ce UC permet à un étudiant d'enregistrer ses cours... Seuls les formations bien construites sont acceptés. Le catalogue des cours est notifié des inscriptions.

Décrire les UC

Détailler les UC

Processus d'écriture des UC

Trouver les acteurs



Trouver les UC

Enregistrer des cours

Brève description: Ce UC permet à un étudiant d'enregistrer ses cours... Seuls les formations bien construites sont acceptés. Le catalogue des cours est notifié des inscriptions.

Décrire les UC

Description de «Enregistrer des cours»

- Flot d'évènements
- Pas à pas

Détailler les UC

Processus d'écriture des UC

Trouver les acteurs



Etudiant



Catalogue
des cours

Trouver les UC

Enregistrer des cours

Brève description: Ce UC permet à un étudiant d'enregistrer ses cours... Seuls les formations bien construites sont acceptés. Le catalogue des cours est notifié des inscriptions.

Décrire les UC

Description de «Enregistrer des cours»

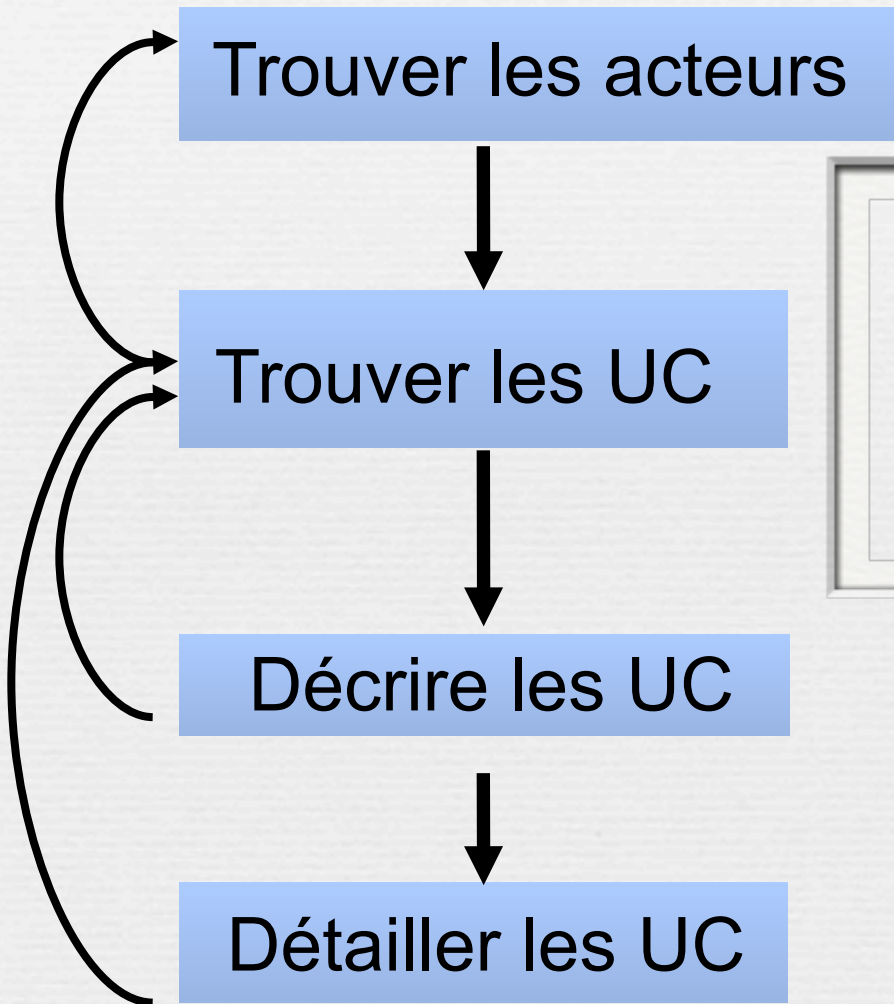
- Flot d'évènements
- Pas à pas

Détailler les UC

Specification de «Enregistrer des cours»

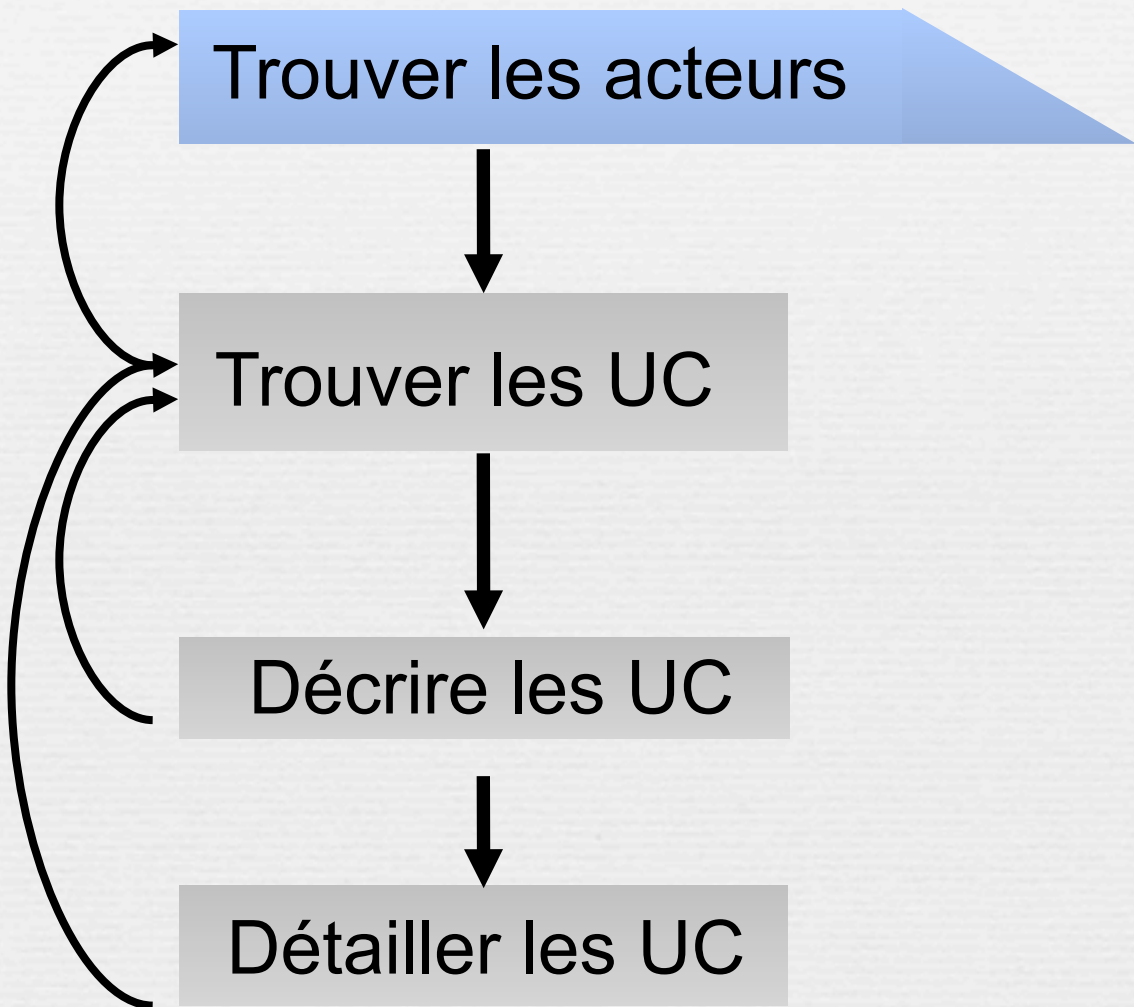
- Flot d'évènements détaillés
- Exigences spéciales
- Pre/Postconditions
- Jeux de données et Tests

Processus d'écriture des UC



Important
C'est un processus
itératif

Processus d'écriture des UC



► Nommer et brièvement décrire les acteurs trouvés

Définir le périmètre du SI : Acteurs

● Définir les acteurs **externes**

- physiques et logiques
- rôle et entité concrète



« Un acteur est une personne ou une chose qui va interagir avec le système »



Client

Acteurs



Un Acteur =

- ▶ élément externe qui interagit avec le système
- ▶ **rôle** qu'un utilisateur joue par rapport au système
ex: un enseignant, un guichetier



Une même personne peut jouer plusieurs rôles

ex: Marie est enseignante et étudiante

Maurice est directeur mais peut faire le guichetier



Plusieurs personnes peuvent jouer un même rôle

ex: Paul et Pierre sont deux clients



Un acteur n'est pas forcément un être humain

ex: un distributeur de billet peut être vu comme un acteur; un gestionnaire de mot de passes

UML au travail : Système d'inscriptions

L'université ESU désire automatiser son système d'inscription

- ▶ Le chef du service des inscriptions établit le programme des cours pour un semestre
 - ▶ Un cours peut être offert plusieurs fois
- ▶ Les étudiants doivent sélectionner 4 cours primaires et 2 cours secondaires dans le catalogue des cours proposés;
- ▶ Dès qu'un étudiant s'est inscrit pour un semestre, le système de facturation est notifié
- ▶ Les étudiants peuvent utiliser le système pour modifier leurs choix pendant une certaine période de temps après leur inscription
- ▶ Les enseignants utilisent le système pour consulter leur emploi du temps (tableau d'activités en fonction des cours qui tournent)
- ▶ Les utilisateurs du système d'inscription reçoivent des mots de passe qui sont nécessaire à la procédure d'identification

UML au travail : Système d'inscriptions

L'université ESU désire automatiser son système d'inscription

- ▶ Le **chef du service des inscriptions** établit le programme des cours pour un semestre
 - ▶ Un cours peut être offert plusieurs fois
- ▶ Les **étudiants** doivent sélectionner 4 cours primaires et 2 cours secondaires dans le **catalogue des cours proposés**

Dès qu'un étudiant s'est inscrit pour un semestre, le **système de facturation** est notifié

- ▶ Les étudiants peuvent utiliser le système pour modifier leurs choix pendant une certaine période de temps après leur inscription
- ▶ Les **enseignants** utilisent le système pour consulter leur emploi du temps (tableau d'activités en fonction des cours qui tournent)
- ▶ Les utilisateurs du système d'inscription reçoivent des mots de passe qui sont nécessaire à la procédure d'identification

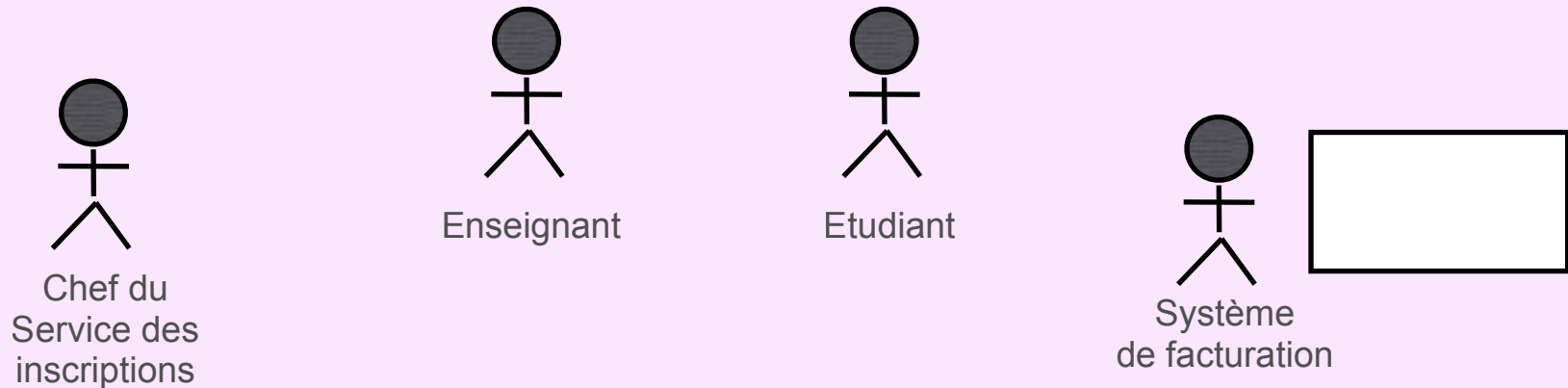
UML au travail : Système d'inscriptions

L'université ESU désire automatiser son système d'inscription

- Le **chef du service des inscriptions** établit le programme des cours pour un semestre
 - Un cours peut être offert plusieurs fois
- Les **étudiants** doivent sélectionner 4 cours primaires et 2 cours secondaires dans le **catalogue des cours proposés**

Dès qu'un étudiant s'est inscrit pour un semestre, le **système de facturation** est notifié

- Les étudiants peuvent utiliser le système pour modifier leurs choix pendant une certaine période de temps après leur inscription
- Les **enseignants** utilisent le système pour consulter leur emploi du temps (tableau d'activités en fonction des cours qui tournent)
- Les utilisateurs du système d'inscription reçoivent des mots de passe qui sont nécessaire à la procédure d'identification



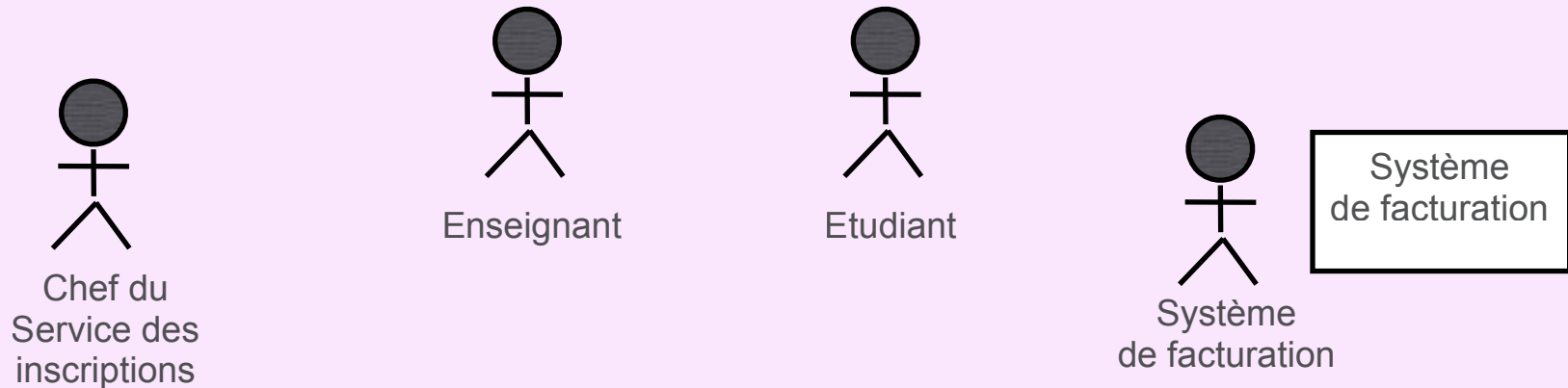
UML au travail : Système d'inscriptions

L'université ESU désire automatiser son système d'inscription

- Le **chef du service des inscriptions** établit le programme des cours pour un semestre
 - Un cours peut être offert plusieurs fois
- Les **étudiants** doivent sélectionner 4 cours primaires et 2 cours secondaires dans le **catalogue des cours proposés**

Dès qu'un étudiant s'est inscrit pour un semestre, le **système de facturation** est notifié

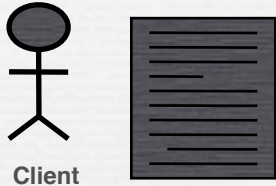
- Les étudiants peuvent utiliser le système pour modifier leurs choix pendant une certaine période de temps après leur inscription
- Les **enseignants** utilisent le système pour consulter leur emploi du temps (tableau d'activités en fonction des cours qui tournent)
- Les utilisateurs du système d'inscription reçoivent des mots de passe qui sont nécessaire à la procédure d'identification



Trouver les acteurs

- Qui ou quoi utilise le système?
- Qui ou quoi obtient de l'information de ce système ?
- Qui ou quoi fournit des informations au système ?
- Où dans la compagnie le système est-il utilisé ?
- Qui ou quoi supporte et maintient le système?
- Quels autres systèmes utilisent ce système?

Description des acteurs



Pour chaque acteur :

- choisir un identificateur représentatif de son rôle
(*un bon nom décrit la responsabilité des acteurs*)
- donner une brève description textuelle

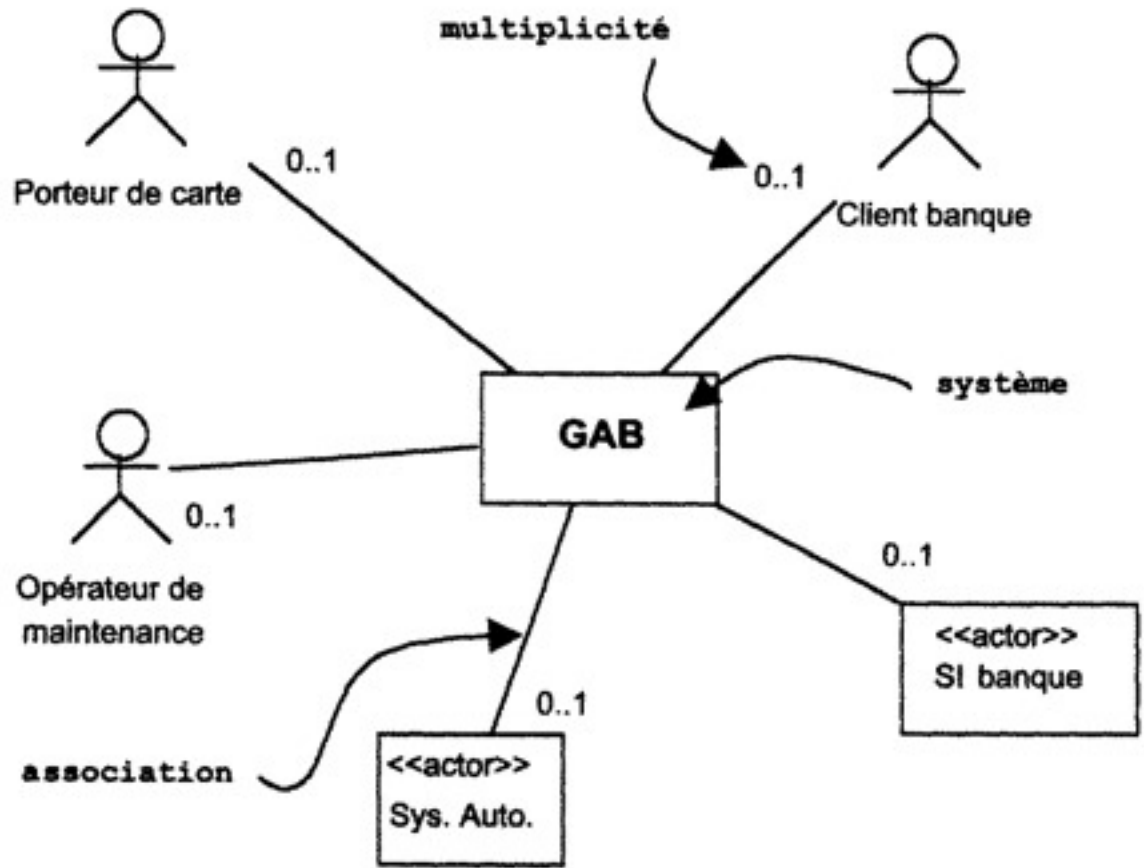


Guichetier

Un guichetier est un employé de la banque chargé de faire l'interface entre le système informatique et les clients qu'il reçoit au comptoir. Le guichetier peut réaliser les opérations courantes : création d'un compte, dépôt et retrait d'argent, etc.

Diagramme de contexte

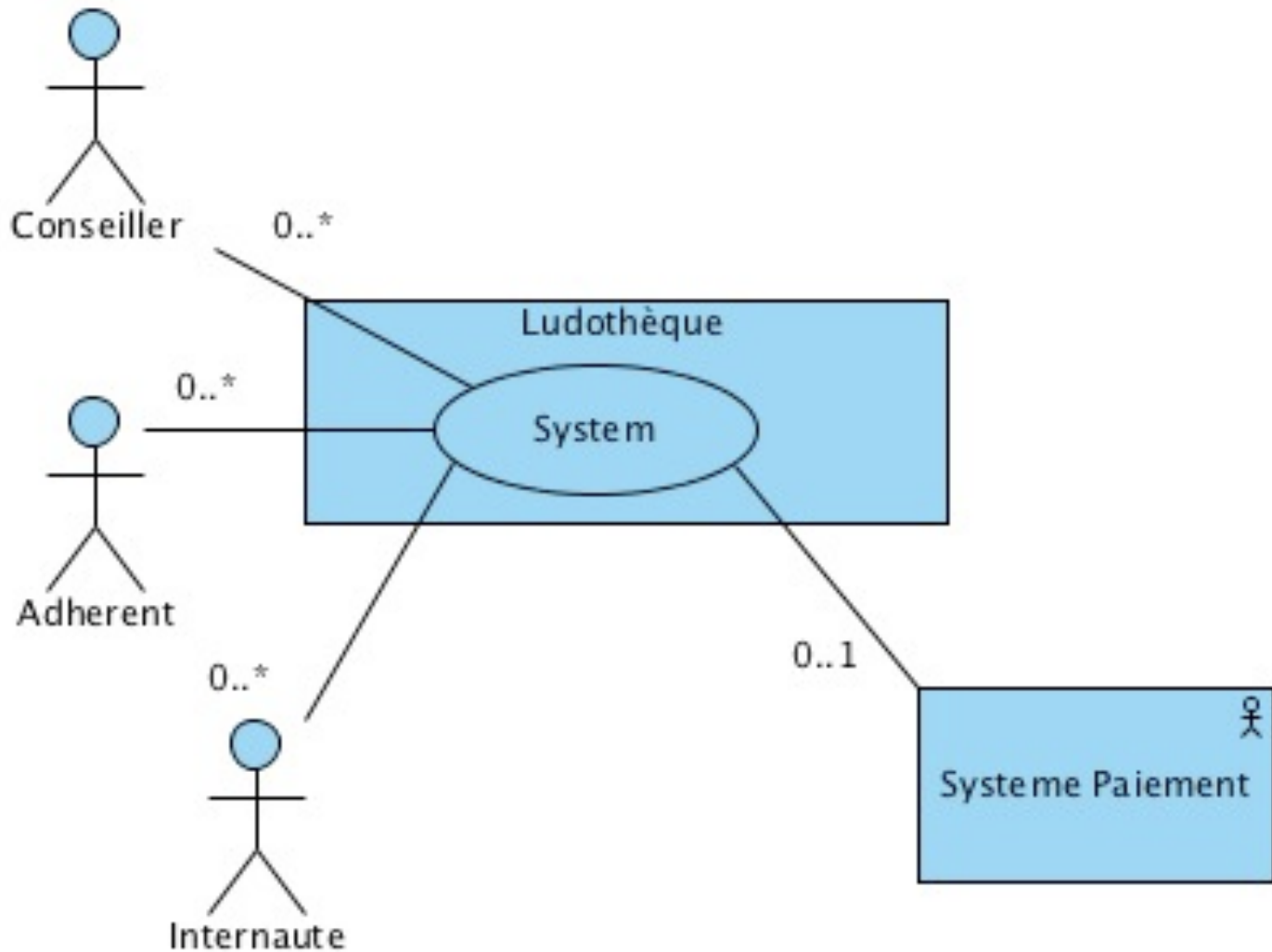
Diagramme de contexte statique



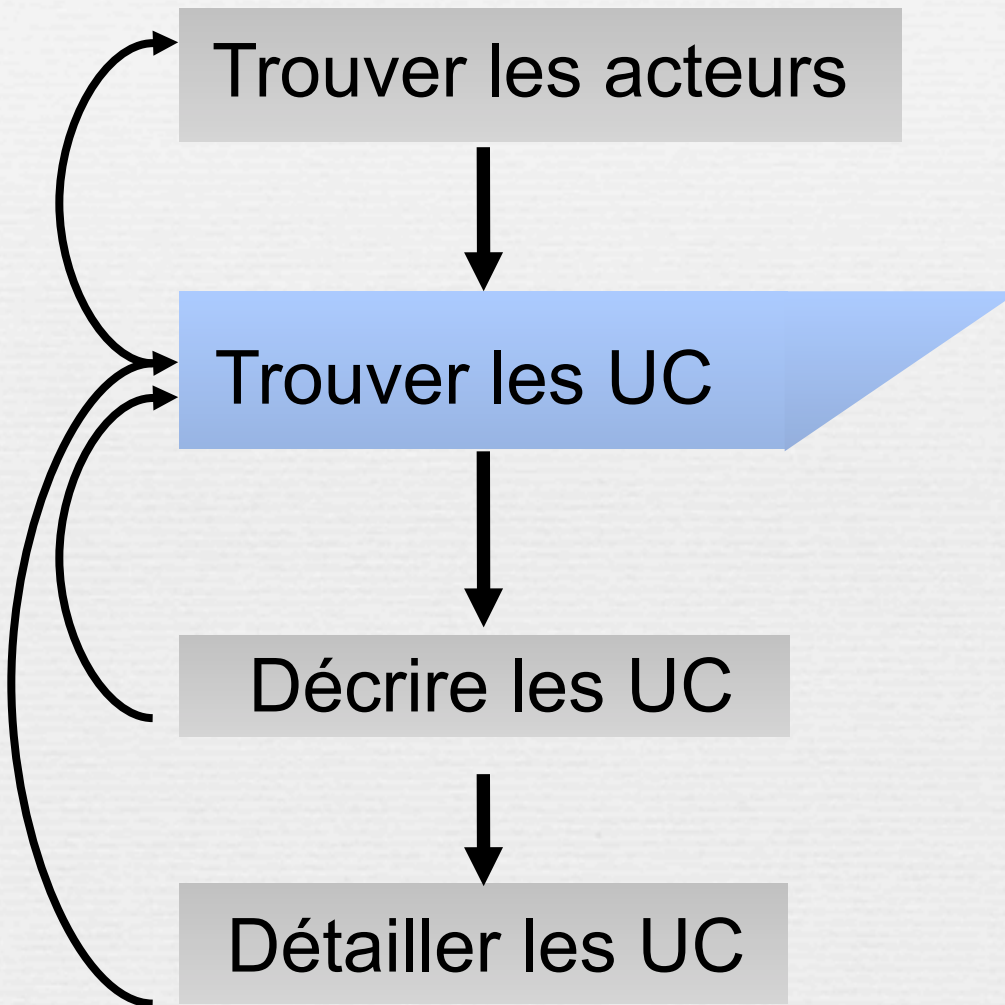
UML au travail : Une ludothèque

- (1) Nous voulons informatiser une ludothèque pour favoriser la consultation des jeux proposés par la ludothèque.
 - (2) Les adhérents peuvent emprunter des jeux en s'adressant à un conseiller qui enregistre l'emprunt.
 - (3) Les jeux empruntés sont rendus à un conseiller...
 - (4) Un adhérent peut réserver des jeux. Une réservation précise l'emprunteur, le jeu et la date de la demande de réservation. L'adhérent est averti quand le jeu revient en rayon.
 - (5) Pour organiser un événement le conseiller spécialisé doit alors donner les informations suivantes : les jeux à tester, le nombre maximal et minimal de participants attendus, la date, et l'heure de début de l'événement.
 - (6) Un adhérent peut s'inscrire pour participer à un événement à condition qu'il y ait encore de la place.
 - (7) Un adhérent peut payer sa cotisation en ligne par un système de paiement externe
- Un internaute peut consulter les jeux et s'inscrire.

UML au travail : Une ludothèque



Processus d'écriture des UC



- ▶ Nommer et brièvement décrire les UC trouvés
- ▶ Créer un diagramme de UC
- ▶ Etablir la plus-value métier et les risques techniques des UC


Cas d'utilisation (UC)

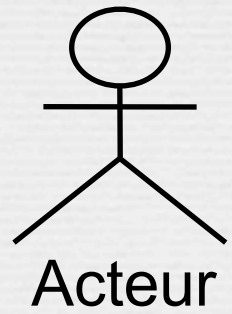
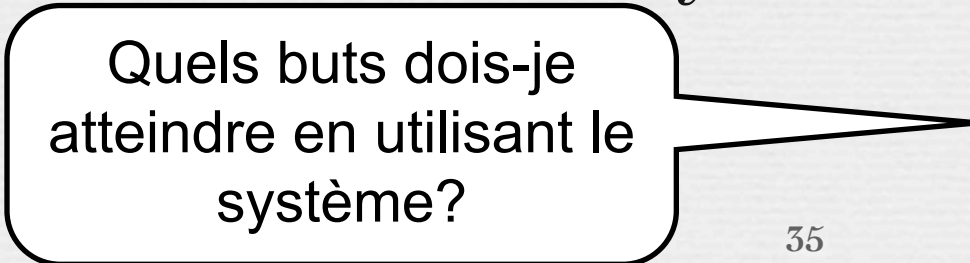


- Un cas d'utilisation est un motif cohérent de comportement
 - réalisé par le système.
- Chaque cas d'utilisation est décrit par une séquence d'actions connectées, effectuées par un dialogue entre des acteurs et le système
 - qui produit un résultat observable
 - d'intérêt pour un ou plusieurs acteurs du système.
 - ne révèle pas la structure interne du système.
- Chaque cas d'utilisation est un flot complet et faisant du sens du point de vue d'un acteur particulier.**

Trouver les use-cases



- 
 Quels sont les objectifs de chaque acteur?
 - ▶ Pourquoi l'acteur utiliserait-il le système?
 - ▶ Est-ce que l'acteur créera, stockera, modifiera, supprimera ou lira des données dans le système? Si oui, pourquoi?
 - ▶ Est-ce que l'acteur nécessite d'informer le système sur des événements externes ou des changements?
 - ▶ Est-ce que l'acteur doit être informé de certains événements dans le système?



UML au travail : Système d'inscriptions

L'université ESU désire automatiser son système d'inscription

- ▶ Le **chef du service des inscriptions** établit le programme des cours pour un semestre
 - ▶ Un cours peut être offert plusieurs fois
- ▶ Les **étudiants** doivent sélectionner 4 cours primaires et 2 cours secondaires dans le **catalogue des cours proposés**

Dès qu'un étudiant s'est inscrit pour un semestre, le **système de facturation** est notifié

- ▶ Les étudiants peuvent utiliser le système pour modifier leurs choix pendant une certaine période de temps après leur inscription
- ▶ Les **enseignants** utilisent le système pour consulter leur emploi du temps (tableau d'activités en fonction des cours qui tournent)
- ▶ Les utilisateurs du système d'inscription reçoivent des mots de passe qui sont nécessaire à la procédure d'identification

UML au travail : Système d'inscriptions

L'université ESU désire automatiser son système d'inscription

▸ Le **chef du service des inscriptions** **établit le programme des cours** pour un semestre

▸ Un cours peut être offert plusieurs fois

Les **étudiants** doivent **sélectionner** 4 **cours** primaires et 2 cours secondaires dans le **catalogue des cours proposés**

Dès qu'un étudiant s'**est inscrit** pour un semestre, le **système de facturation** est notifié

Les étudiants peuvent utiliser le système pour **modifier leurs choix** pendant une certaine période de temps après leur inscription

Les **enseignants** utilisent le système pour **consulter leur emploi du temps** (tableau d'activités en fonction des cours qui tournent)

▸ Les utilisateurs du système d'inscription **reçoivent des mots de passe** qui sont nécessaire à la procédure d'identification

Nommer un use-case

?

- Enregistrer des Cours
- Enregistrement de ses cours
- Cours
- Utiliser le système d'enregistrement
- Accuser réception des cours

Nommer un use-case



Placez vous du point de vue de l'acteur qui déclenche le cas d'utilisation

?

- Enregistrer des Cours
- Enregistrement de ses cours
- Cours
- Utiliser le système d'enregistrement
- Accuser réception des cours

Nommer un use-case

- Placez vous du point de vue de l'acteur qui déclenche le cas d'utilisation
- Il doit commencer par un **verbe** et utiliser une simple combinaison **verbe-nom**

?

- Enregistrer des Cours
- Enregistrement de ses cours
- Cours
- Utiliser le système d'enregistrement
- Accuser réception des cours

Nommer un use-case

- Placez vous du point de vue de l'acteur qui déclenche le cas d'utilisation
- Il doit commencer par un **verbe** et utiliser une simple combinaison **verbe-nom**
- Le nom doit être unique, intuitif et auto-explicatif

?

- Enregistrer des Cours
- Enregistrement de ses cours
- Cours
- Utiliser le système d'enregistrement
- Accuser réception des cours

Nommer un use-case

- Placez vous du point de vue de l'acteur qui déclenche le cas d'utilisation
- Il doit commencer par un **verbe** et utiliser une simple combinaison **verbe-nom**
- Le nom doit être unique, intuitif et auto-explicatif
- Définir clairement et sans ambiguïté le gain des résultats observables

?

- Enregistrer des Cours
- Enregistrement de ses cours
- Cours
- Utiliser le système d'enregistrement
- Accuser réception des cours

Nommer un use-case

- Placez vous du point de vue de l'acteur qui déclenche le cas d'utilisation
- Il doit commencer par un **verbe** et utiliser une simple combinaison **verbe-nom**
- Le nom doit être unique, intuitif et auto-explicatif
- Définir clairement et sans ambiguïté le gain des résultats observables
- Décrire le comportement fournit par le cas d'utilisation

?

- Enregistrer des Cours
- Enregistrement de ses cours
- Cours
- Utiliser le système d'enregistrement
- Accuser réception des cours



Nommer un use-case

- Placez vous du point de vue de l'acteur qui déclenche le cas d'utilisation
- Il doit commencer par un **verbe** et utiliser une simple combinaison **verbe-nom**
- Le nom doit être unique, intuitif et auto-explicatif
- Définir clairement et sans ambiguïté le gain des résultats observables
- Décrire le comportement fournit par le cas d'utilisation

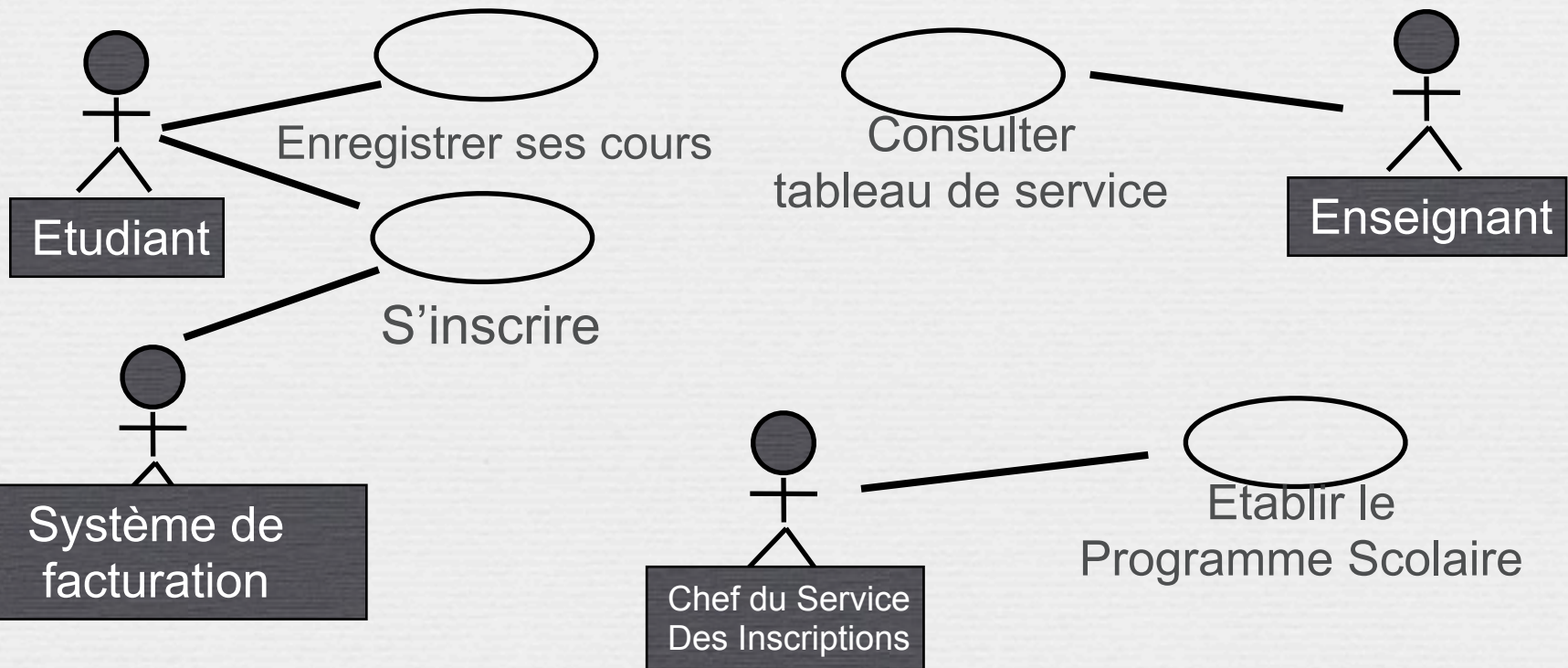
?

- Enregistrer des Cours
- Enregistrement de ses cours
- Cours
- Utiliser le système d'enregistrement
- Accuser réception des cours

Diagramme des UC

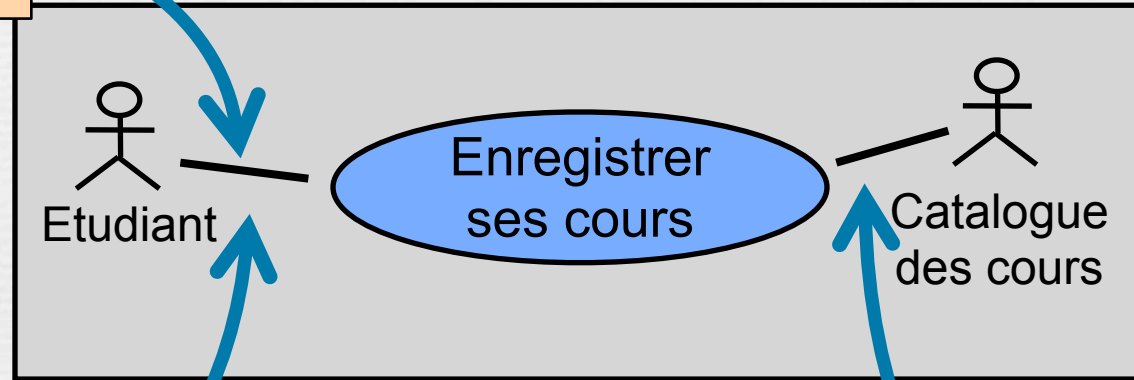


Objectif : visualiser les relations entre acteurs et cas d'utilisation (communication)



Communication : un dialogue

L'étudiant se connecte au système
 Le système approuve la connexion.
 L'étudiant requiert des informations



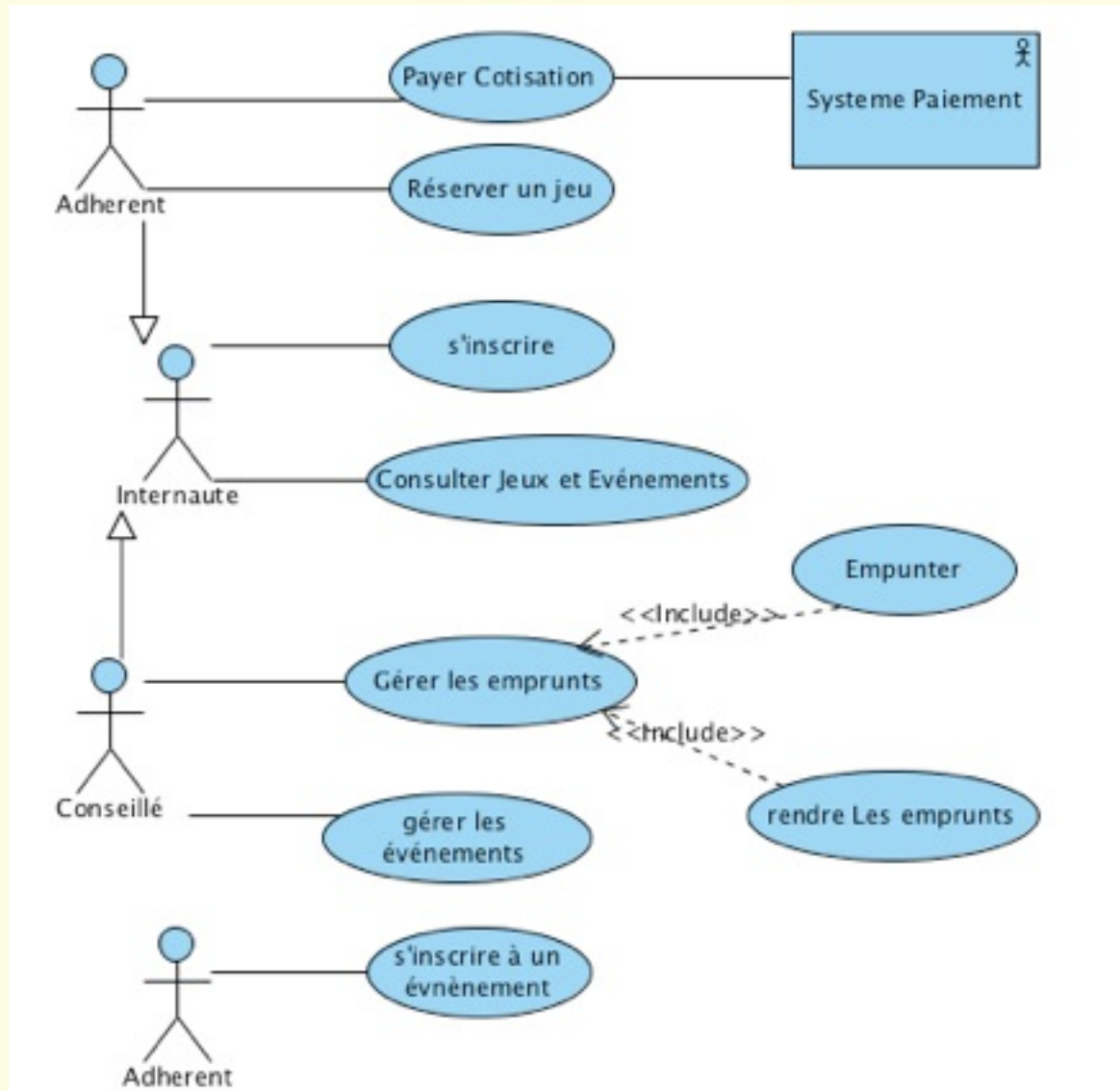
Le système affiche la liste des cours
 L'étudiant sélectionne les cours
 Le système affiche l'edt approuvé

Le système transmet la requête
 Le Catalogue des cours retourne des informations sur les cours.

UML au travail : Une ludothèque

- (1) Nous voulons informatiser une ludothèque pour favoriser la consultation des jeux proposés par la ludothèque.
- (2) Les adhérents peuvent emprunter des jeux en s'adressant à un conseiller qui enregistre l'emprunt.
- (3) Les jeux empruntés sont rendus à un conseiller....
- (4) Un adhérent peut réserver des jeux. Une réservation précise l'emprunteur, le jeu et la date de la demande de réservation. L'adhérent est averti quand le jeu revient en rayon.
- (5) Pour organiser un événement le conseiller spécialisé doit alors donner les informations suivantes : les jeux à tester, le nombre maximal et minimal de participants attendus, la date, et l'heure de début de l'événement.
- (6) Un adhérent peut s'inscrire pour participer à un événement à condition qu'il y ait encore de la place.
- (7) Un adhérent peut payer sa cotisation en ligne par un système de paiement externe
- (8) Un internaute peut consulter les jeux et s'inscrire.

UML au travail : Une ludothèque



Le système

Le système est un ensemble de cas d'utilisation

Le système contient :

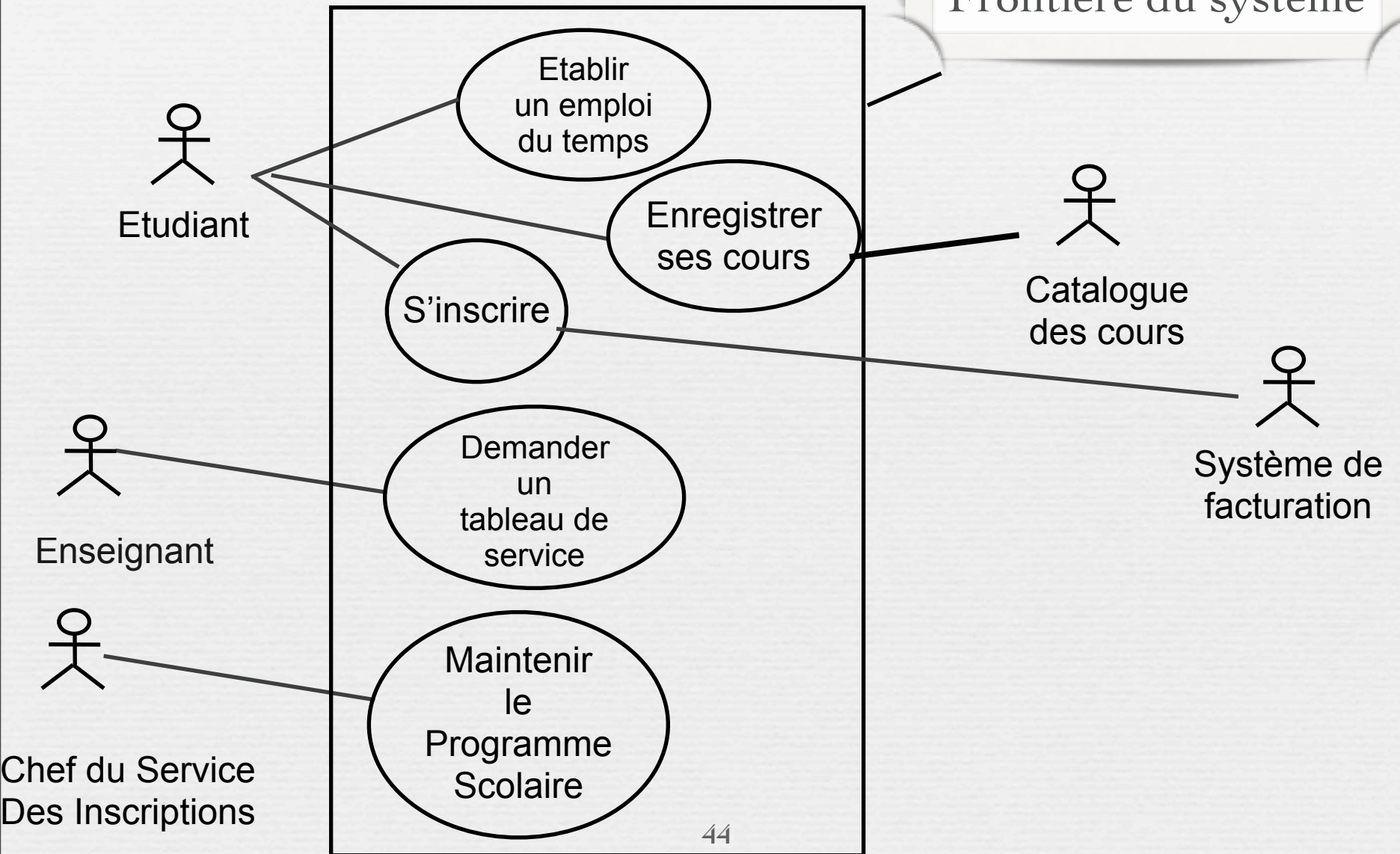
- ▶ les cas d'utilisation,
- ▶ **mais pas les acteurs.**

Un modèle de cas d'utilisation permet de définir :

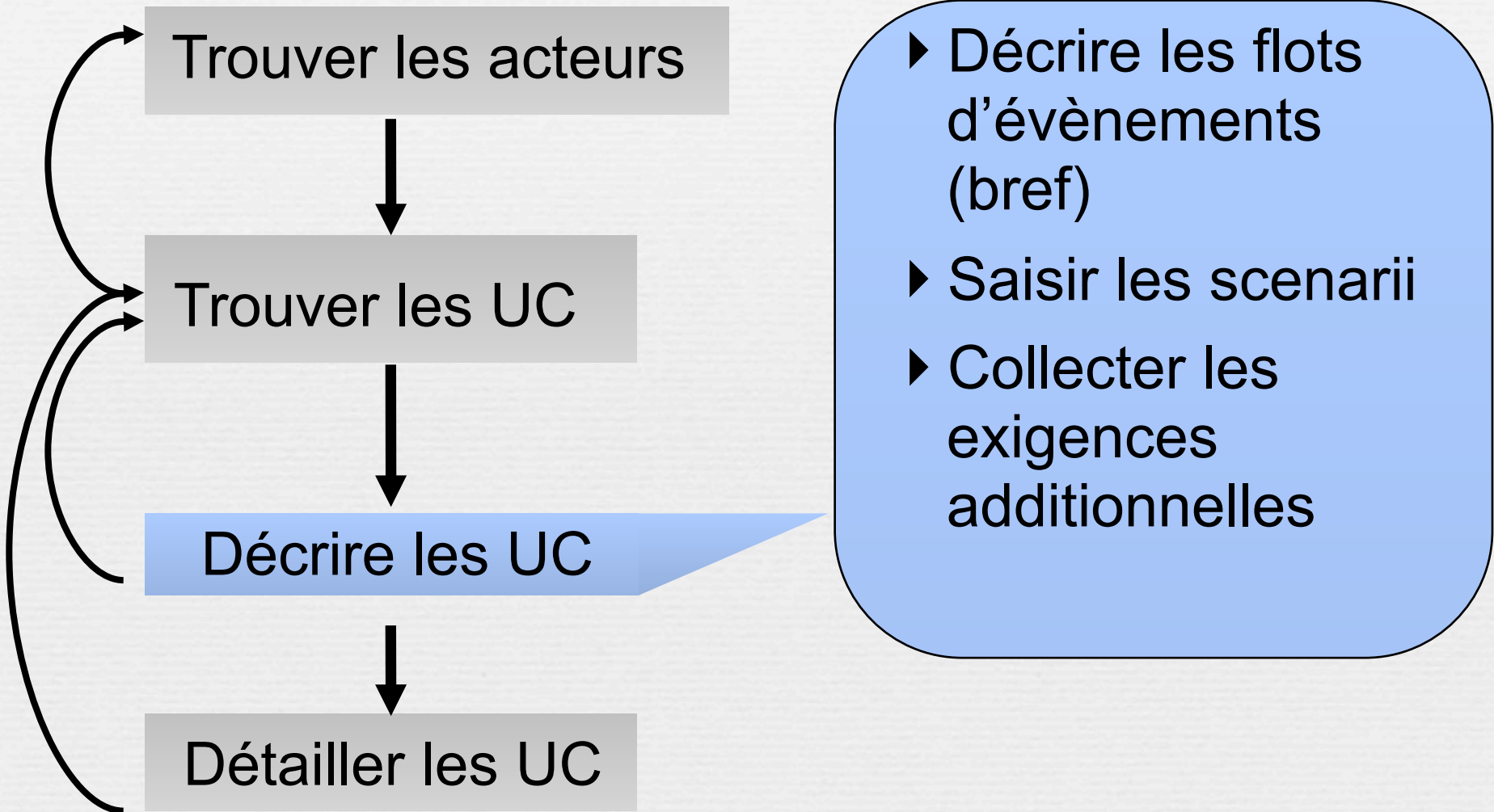
- ▶ les fonctions essentielles du système,
- ▶ les limites du système,
- ▶ le système par rapport à son environnement.

System

Frontière du système



Processus d'écriture des UC



Décrire un UC

description brève

Use Case Name

Brief Description

Basic Flow

1. First step
2. Second step
3. Third step

Alternative Flows

1. Alternative flow 1
2. Alternative flow 2
3. Alternative flow 3

Description succincte d'un UC

Sommaire d'identification :

- Titre
- Résumé
- Acteurs
- Date de création
- Date de mise à jour
- Version
- Responsable

Description textuelle du cas d'utilisation:

« RETIRER DE L'ARGENT »

Sommaire d'identification

Titre : Retirer de l'argent

Résumé : ce cas d'utilisation permet à un porteur de carte, qui n'est pas client de la banque, de retirer de l'argent, si son crédit hebdomadaire le permet.

Acteurs : Porteur de carte non client (principal), *Sys. Auto. (secondaire)*.

Date de création : 03/01/07 **Date de mise à jour** : 09/02/07

Version : 1.0 **Responsable** : Pierre DUMONT

Décrire un UC



Décrire chaque étape du UC par des phrases courtes, organisées séquentiellement.

Use Case Name

Brief Description

Basic Flow

- 1. First step**
- 2. Second step**
- 3. Third step**

Alternative Flows

- 1. Alternative flow 1**
- 2. Alternative flow 2**
- 3. Alternative flow 3**

Décrire un UC

- Décrire chaque étape du UC par des phrases courtes, organisées séquentiellement.

Numéroter et nommer les étapes.

Use Case Name

Brief Description

Basic Flow

1. First step
2. Second step
3. Third step

Alternative Flows

1. Alternative flow 1
2. Alternative flow 2
3. Alternative flow 3

Structurer le flot de base en étapes majeures

Identifier les flots alternatifs.

UML au travail : Une ludothèque

Flot basique:

- 1) **L'adhérent** s'identifie par sa carte
- 2) **L'adhérent** choisit au plus deux jeux
- 2) **Le système** vérifie que les jeux sont disponibles
- 4) **Le système** vérifie l'état de la cotisation, l'éventuel période de pénalité de retard, l'état des emprunts
- 5) **Le conseiller** enregistre l'emprunt (identifiant du jeu + date jour + n° matricule)

Flot alternatif:

- A4a) **L'adhérent** n'a pas payé sa cotisation.
- A4b) **L'adhérent** est sous pénalité de retard

...

Formulation



- Utilisez la voix active

- ▶ Au lieu de : “Quand le Professeur a attribué les notes”

- Dire ce qui déclenche l'étape

- ▶ Au lieu de : “Le UC commence quand le Prof. **décide** de”

- Dire qui fait quoi (utiliser le nom d'acteur)

- ▶ Au lieu de: "L'utilisateur choisit"

<Acteur>
<fait>

- ▶ Au lieu de: "Le choix est validé ..."

<Système>
<fait>

Formulation



- Utilisez la voix active
 - ▶ Dire: “Le Professeur attribue des notes à chaque étudiant”
 - ▶ Au lieu de : “Quand le Professeur a attribué les notes”
- Dire ce qui déclenche l'étape
 - ▶ Au lieu de : “Le UC commence quand le Prof. décide de”
- Dire qui fait quoi (utiliser le nom d'acteur)
 - ▶ Au lieu de: "L'utilisateur choisit"
 - ▶ Au lieu de: "Le choix est validé ..."

<Acteur>
<fait>

<Système>
<fait>

Formulation



- Utilisez la voix active
 - ▶ Dire: “Le Professeur attribue des notes à chaque étudiant”
 - ▶ Au lieu de : “Quand le Professeur a attribué les notes”
- Dire ce qui déclenche l'étape
 - ▶ Dire: “Le UC commence quand le Prof. **choisit** de donner une note.”
 - ▶ Au lieu de : “Le UC commence quand le Prof. **décide** de”
- Dire qui fait quoi (utiliser le nom d'acteur)

▶ Au lieu de: "L'utilisateur choisit"

<Acteur>
<fait>

▶ Au lieu de: "Le choix est validé ..."

<Système>
<fait>

Formulation



- Utilisez la voix active
 - ▶ Dire: “Le Professeur attribue des notes à chaque étudiant”
 - ▶ Au lieu de : “Quand le Professeur a attribué les notes”
- Dire ce qui déclenche l'étape
 - ▶ Dire: “Le UC commence quand le Prof. **choisit** de donner une note.”
 - ▶ Au lieu de : “Le UC commence quand le Prof. **décide** de”
- Dire qui fait quoi (utiliser le nom d'acteur)
 - ▶ Dire: “L'étudiant choisit”
 - ▶ Au lieu de: "L'utilisateur choisit"

 - ▶ Au lieu de: "Le choix est validé ..."

<Acteur>
 <fait>

<Système>
 <fait>

Formulation

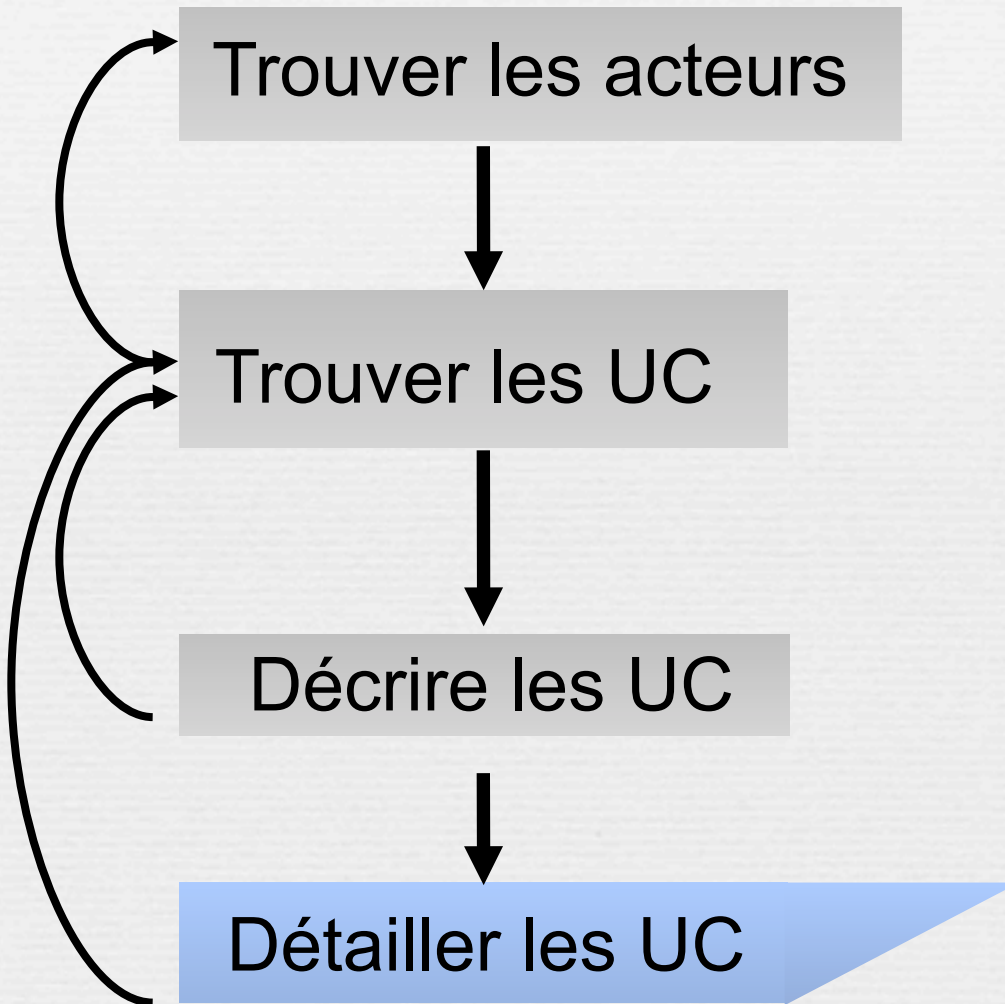


- Utilisez la voix active
 - ▶ Dire: “Le Professeur attribue des notes à chaque étudiant”
 - ▶ Au lieu de : “Quand le Professeur a attribué les notes”
- Dire ce qui déclenche l'étape
 - ▶ Dire: “Le UC commence quand le Prof. choisit de donner une note.”
 - ▶ Au lieu de : “Le UC commence quand le Prof. décide de”
- Dire qui fait quoi (utiliser le nom d'acteur)
 - ▶ Dire: “L'étudiant choisit”
 - ▶ Au lieu de: "L'utilisateur choisit"
 - ▶ Dire: “Le Système valide ...”
 - ▶ Au lieu de: "Le choix est validé ...”

<Acteur>
<fait>

<Système>
<fait>

Processus d'écriture des UC



- ▶ Détailler les flots d'événements
- ▶ Structurer les flots d'événements
- ▶ Spécifier les propriétés additionnelles

Description détaillée d'un UC

- Description des scénarios:
- Préconditions
- Scénario Nominal
- Flots alternatifs
- Flots d'erreur
- Postconditions
- Exigences non fonctionnelles
- Jeux de tests

Préconditions

- Décrire l'état dans lequel doit être le système avant que le UC puisse commencer.
 - ▶ Simples déclarations qui définissent l'état du système, exprimées comme ses conditions qui doivent être remplies
 - ▶ Il ne faut jamais se référer à d'autres UC qui doivent être effectuées avant cet UC
 - ▶ Devraient être énoncées clairement et devraient être facilement vérifiables
- **Facultatif:** Utilisez uniquement si nécessaire pour clarifier

UC «Inscription à des cours»

▶ *Préconditions*

- ▶ La liste des offres de cours pour le semestre a été créée et est disponible au service d'inscription.
- ▶ L'élève a ouvert une session d'inscription dans le système

UML au travail : Une ludothèque

Description:

L'adhérent emprunte un/deux jeu(x).

Pré-condition:

Le client est un adhérent.

Flot basique:

- 1) L'adhérent s'identifie par sa carte
- 2) L'adhérent choisit au plus deux jeux
- 2) Le système vérifie que les jeux sont disponibles
- 4) Le système vérifie l'état de la cotisation, l'éventuel période de pénalité de retard, l'état des emprunts
- 5) Le conseiller enregistre l'emprunt (identifiant du jeu + date jour + n° matricule)

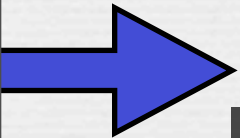
Flot alternatif:

4a) L'adhérent n'a pas payé sa cotisation.

....

Description détaillée d'un UC

- Description des scénarios:
 - Préconditions
 - Scénario Nominal
 - Flots alternatifs
 - Flots d'erreur
 - Postconditions
 - Exigences non fonctionnelles
- Jeux de tests



Postconditions

- Décrire l'état dans lequel doit être le système à la fin du UC
 - ▶ Utiliser lorsque l'état du système est une condition préalable à un autre UC, ou lorsque les résultats possibles du UC ne sont pas évidents pour le lecteur
 - ▶ Il ne faut jamais se référer à d'autres UC qui doivent être effectués avant cet UC
 - ▶ Devraient être énoncées clairement et devraient être facilement vérifiables
- **Facultatif:** Utilisez uniquement si nécessaire pour clarifier

UC «Inscription à des cours»

▶ *Postconditions*

- ▶ À la fin de ce cas d'utilisation, soit l'étudiant a été inscrit à des cours, soit l'inscription a échoué et aucune modification n'a été apportée à aux cours...

UML au travail : Une ludothèque

Flot basique:

- 1) L'adhérent s'identifie par sa carte
- 2) L'adhérent choisit au plus deux jeux
- 3) Le système vérifie que les jeux sont disponibles
- 4) Le système vérifie l'état de la cotisation, l'éventuel période de pénalité de retard, l'état des emprunts
- 5) Le conseiller enregistre l'emprunt (identifiant du jeu + date jour + n° matricule)

UML au travail : Une ludothèque

Flot basique:

- 1) L'adhérent s'identifie par sa carte
- 2) L'adhérent choisit au plus deux jeux
- 3) Le système vérifie que les jeux sont disponibles
- 4) Le système vérifie l'état de la cotisation, l'éventuel période de pénalité de retard, l'état des emprunts
- 5) Le conseiller enregistre l'emprunt (identifiant du jeu + date jour + n° matricule)

Postcondition

L'emprunt est enregistré : l'adhérent a à présent la liste des jeux qu'il avait emprunté précédemment + les emprunts courants ; ...

Description détaillée d'un UC

- Description des scénarios:
 - Préconditions
 - Scénario Nominal
 - Flots alternatifs
 - Flots d'erreur
 - Postconditions
 - Exigences non fonctionnelles
- Jeux de tests

UML au travail : Une ludothèque

Flot basique:

- 1) L'adhérent s'identifie par sa carte
- 2) L'adhérent choisit au plus deux jeux
- 3) Le système vérifie que les jeux sont disponibles
- 4) Le système vérifie l'état de la cotisation, l'éventuel période de pénalité de retard, l'état des emprunts
- 5) Le conseiller enregistre l'emprunt (identifiant du jeu + date jour + n° matricule)

UML au travail : Une ludothèque

Flot basique:

- 1) L'adhérent s'identifie par sa carte
- 2) L'adhérent choisit au plus deux jeux
- 3) Le système vérifie que les jeux sont disponibles
- 4) Le système vérifie l'état de la cotisation, l'éventuel période de pénalité de retard, l'état des emprunts
- 5) Le conseiller enregistre l'emprunt (identifiant du jeu + date jour + n° matricule)

Flot alternatif:

- 4a) L'adhérent n'a pas payé sa cotisation.
 - => 1. Le système invite l'adhérent à payer
 - 2 Paiement de la cotisation
 - 3. puis retour à l'étape 5
- 3a) Un jeu n'est pas disponible
 - => 1. Le système avertit l'adhérent
 - 2. retour à l'étape 2

UML au travail : Une ludothèque

Flot basique:

- 1) L'adhérent s'identifie par sa carte
- 2) L'adhérent choisit au plus deux jeux
- 3) Le système vérifie que les jeux sont disponibles
- 4) Le système vérifie l'état de la cotisation, l'éventuel période de pénalité de retard, l'état des emprunts
- 5) Le conseiller enregistre l'emprunt (identifiant du jeu + date jour + n° matricule)

UML au travail : Une ludothèque

Flot basique:

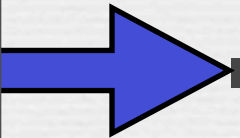
- 1) L'adhérent s'identifie par sa carte
- 2) L'adhérent choisit au plus deux jeux
- 3) Le système vérifie que les jeux sont disponibles
- 4) Le système vérifie l'état de la cotisation, l'éventuel période de pénalité de retard, l'état des emprunts
- 5) Le conseiller enregistre l'emprunt (identifiant du jeu + date jour + n° matricule)

Flot d'erreur:

- 3a) la connexion au système d'emprunt échoue
- =>
1. une alerte vers le système de pilotage est levée
 2. Le système prévient l'adhérent et lui propose de passer plus tard
 3. **le cas d'utilisation se termine en échec.**

Description détaillée d'un UC

- Description des scénarios:
 - Préconditions
 - Scénario Nominal
 - Flots alternatifs
 - Flots d'erreur
 - Postconditions
 - Exigences non fonctionnelles
- Jeux de tests



Description textuelle du cas d'utilisation:

« RETIRER DE L'ARGENT »

informations optionnelles

Exigences non fonctionnelles

Contraintes	Descriptif
Temps de réponse	L'interface du GAB doit réagir en l'espace de 2 secondes au maximum. Une transaction nominale de retrait doit durer moins de 2 minutes
Concurrence	Non applicable (mono-utilisateur)
Disponibilité	Le GAB est accessible 7j/7, 24h/24 . L'absence de papier pour les tickets ne doit pas empêcher les retraits.
Intégrité	Les interface du GAB doivent être très robustes pour prévenir le vandalisme
Confidentialité	La vérification du code saisi doit être fiable à 10^{-6}

UML au travail : Une ludothèque

Flot basique:

- 1) L'adhérent s'identifie par sa carte
- 2) L'adhérent choisit au plus deux jeux
- 3) Le système vérifie que les jeux sont disponibles
- 4) Le système vérifie l'état de la cotisation, l'éventuel période de pénalité de retard, l'état des emprunts
- 5) Le conseiller enregistre l'emprunt (identifiant du jeu + date jour + n° matricule)

UML au travail : Une ludothèque

Flot basique:

- 1) L'adhérent s'identifie par sa carte
- 2) L'adhérent choisit au plus deux jeux
- 3) Le système vérifie que les jeux sont disponibles
- 4) Le système vérifie l'état de la cotisation, l'éventuel période de pénalité de retard, l'état des emprunts
- 5) Le conseiller enregistre l'emprunt (identifiant du jeu + date jour + n° matricule)

Temps de réponse : L'interface du système doit réagir en l'espace de 5 secondes au maximum.

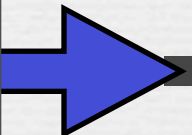
Robustesse : L'enregistrement de l'emprunt est soumis à une transaction.

Sécurité : Authentification,

...

Description détaillée d'un UC

- Description des scénarios:
 - Préconditions
 - Scénario Nominal
 - Flots alternatifs
 - Flots d'erreur
 - Postconditions
 - Exigences non fonctionnelles
- Besoins d'IHM
- Jeux de tests



Détailler un UC



Prévoir les jeux de tests

TABLE 1: Structure of a Test Case

#	Test Case description / Test Steps	Pre-Conditions / Test data	Expected Result	Actual Result
1	a. Create new Order. b. Add Items and choose promotion code created previously. c. Go to review screen to check total amount for the order and then submit the order on the review screen	Customer = TCS Items to add - 3 TV Cables, 5 batteries of 12V DC Use promotion code for Corporate-Offer	The total price calculated should be (3 * price of TV cable) + (5 * price of 12V DC battery) - (5% of the total obtained so far). Also check if this amount is correctly entered in the database record.	As expected
2	a. Go to Order list screen, select the previously created order and click Edit Order. b. Add one more item to the order. c. Go to review screen to check total amount for the order and then submit the order on the review screen	Order created in previous test case Item to add - 2 batteries of 5V DC	Check the total of the order as in the previous test case and also check if the database row is correctly updated.	As expected
3	a. Select Order created in previous step and "Generate Invoice". b. Send invoice to customer		Invoice with new number should be generated in the system. An email with pdf report of the invoice should be sent to the corresponding customer email address	As expected

UI Navigations

Reference to UI Elements

Identical Step as test case 1, step c

Use of a test data descriptor

Test Behavior (the oracle)

Plan du cours





- ✓ Utilisation d'un dictionnaire du domaine (glossaire)

- ✓ Des cas d'utilisations (Use-cases/UC)
 - Acteurs, use-cases, system UML

- ✓ Processus de construction des Uses-case

- ➔ Compléments

Faire le point sur les UC

-  Les interactions entre le système et les acteurs sont claires
-  Les séquences de communication sont conformes aux attentes de l'utilisateur
-  Quand/comment les UC commencent/se terminent est clair
-  Le flot de base de base donne un résultat observable pour un ou plusieurs acteurs

Quel niveau de détail?

✓ Saisir toutes les exigences pour tous les demandeurs

X Pas de détail des interfaces utilisateurs

X Pas de détail des processus internes non liés à une exigence

X Pas les formats de données, ni les contrôles

How much detail in a use case?

Enough to satisfy all stakeholders that their interests (requirements) will be satisfied in the delivered system.

Ecrire des UC : Défis

1. Comment garder les UC précis et concis?
2. Comment traiter les interfaces utilisateur?

1. Comment garder les UC précis et concis? (1)

✓ Capturer le vocabulaire commun dans un glossaire

- ➔ Définir les termes utilisés dans le projet dans le glossaire, pas dans les flux
- ➔ Aide à éviter les malentendus

Use Case

5. Enter Customer Information

The system prompts the Customer to enter their *Customer Details*.

The Customer enters the *Customer Details*.

The Customer creates the account.

Glossary

Customer Details Information that uniquely identifies and provides contact information for a customer located in the U.S.A. The information consists of Name, two address lines, city, state, ZIP code, and daytime phone number.

Implementation

Relations entre UC : Include, Extend, Specialize

Relations entre UC : Include, Extend, Specialize



Au fur et à mesure que les cas d'utilisation sont documentés, des relations peuvent apparaître entre cas d'utilisation.

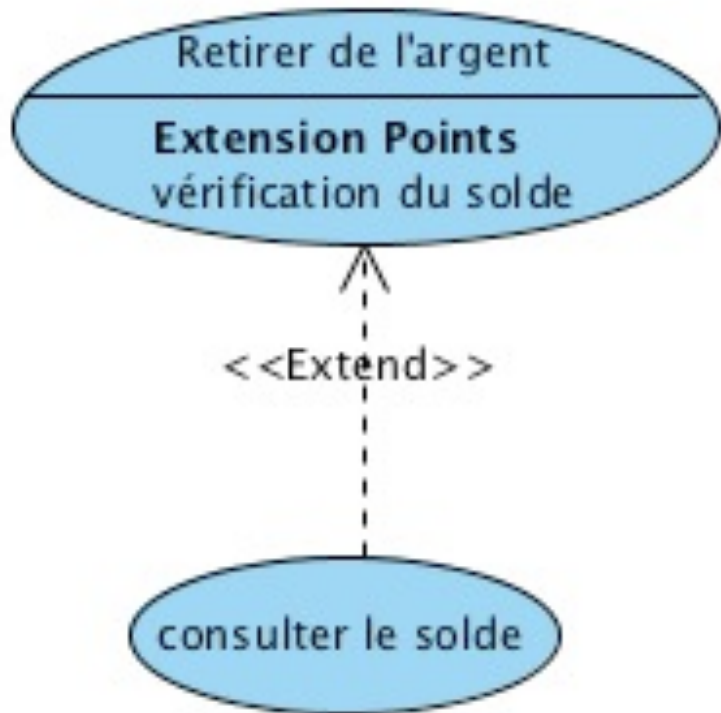
Relations entre UC : Include, Extend, Specialize

- Au fur et à mesure que les cas d'utilisation sont documentés, des relations peuvent apparaître entre cas d'utilisation.
- Mais elles n'expriment **JAMAIS** un ordre entre les cas d'utilisation

Relations entre UC : Include, Extend, Specialize



- Au fur et à mesure que les cas d'utilisation sont documentés, des relations peuvent apparaître entre cas d'utilisation.
- Mais elles n'expriment **JAMAIS** un ordre entre les cas d'utilisation



Extend

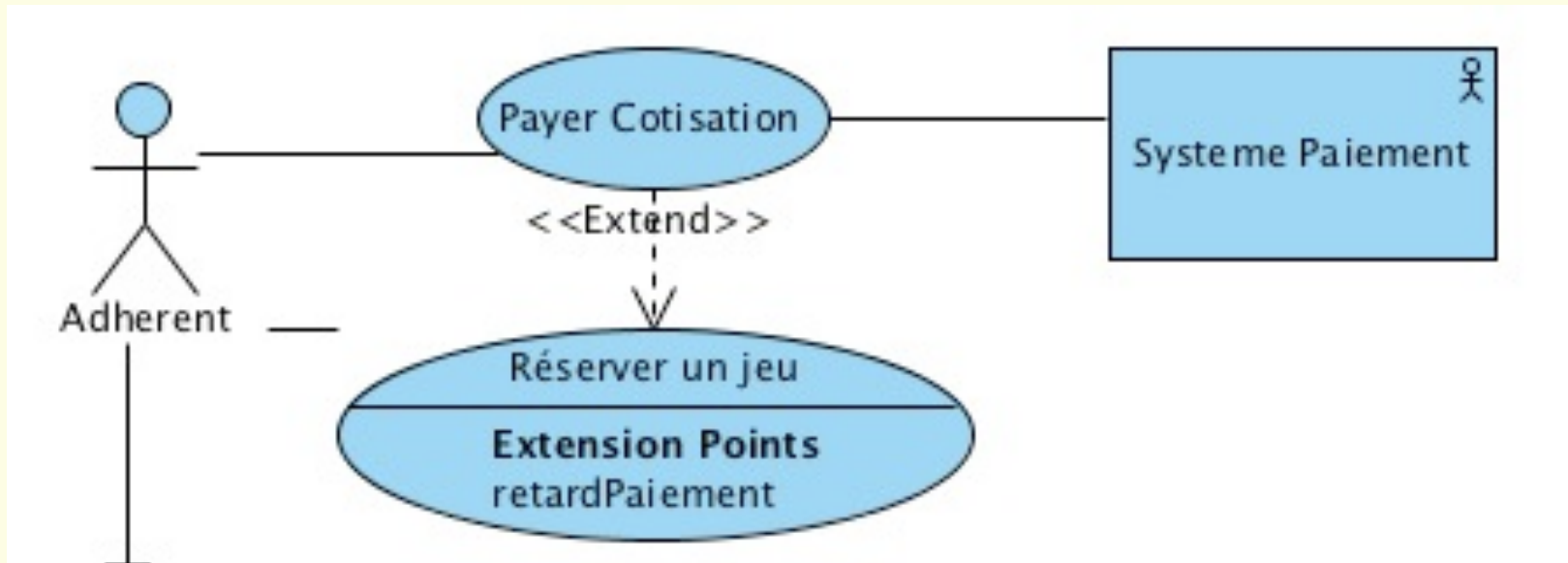
Les deux UC sont indépendants.

Permettre au client de consulter son solde lorsqu'il retire de l'argent : extension optionnelle.

Flot enrichi

8. Le GAB demande au client de saisir le montant
Point d'extension : Vérification du solde
9. Le client saisit le montant désiré de retrait

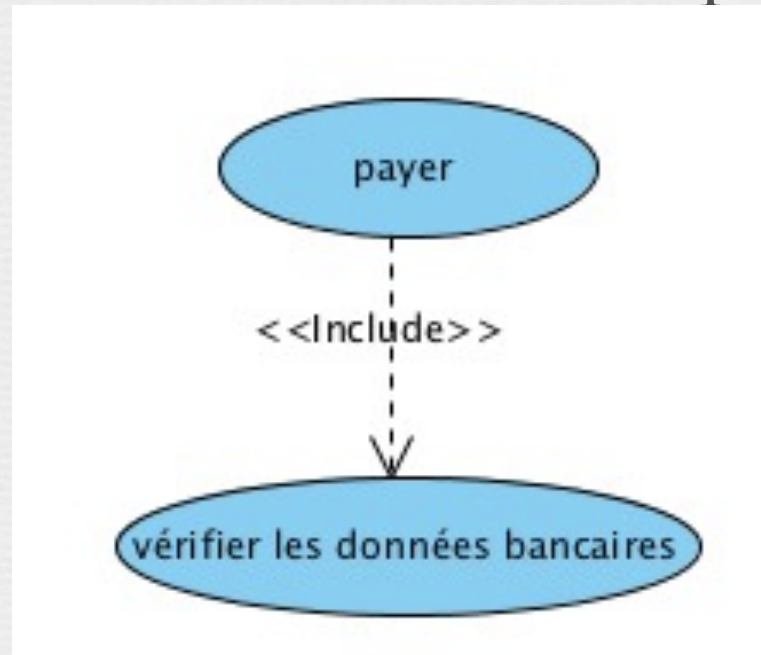
Ludothèque



La relation <<include>>

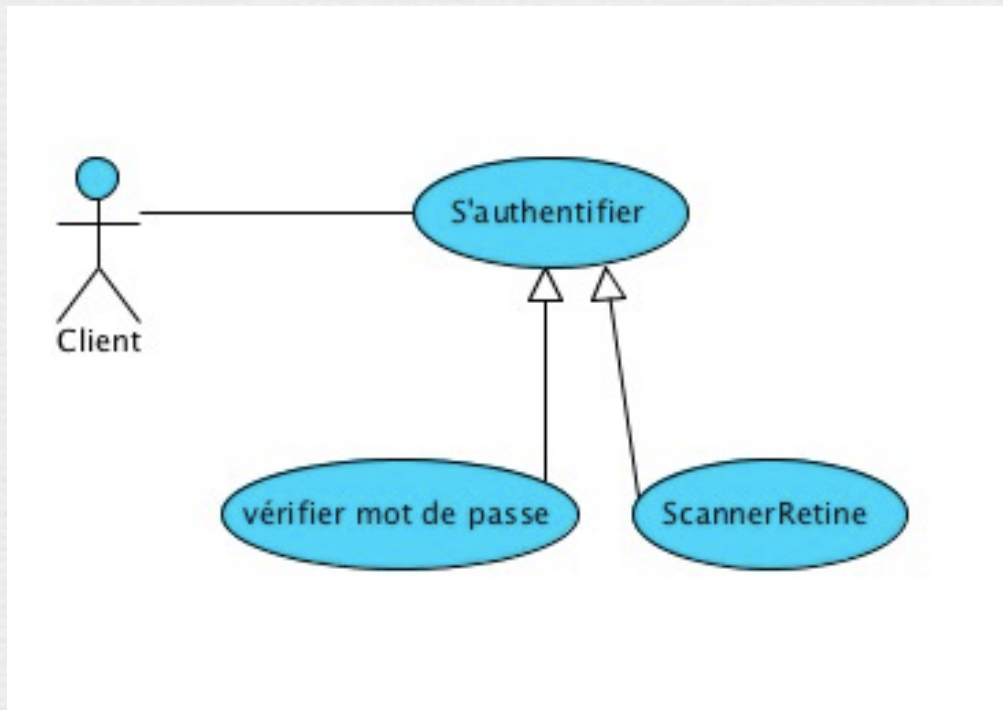
La relation *d'include* permet de **factoriser une partie de la description** d'un cas d'utilisation qui est commune à d'autres cas d'utilisation.

Un cas d'utilisation «inclus» n'est pas «optionnel».

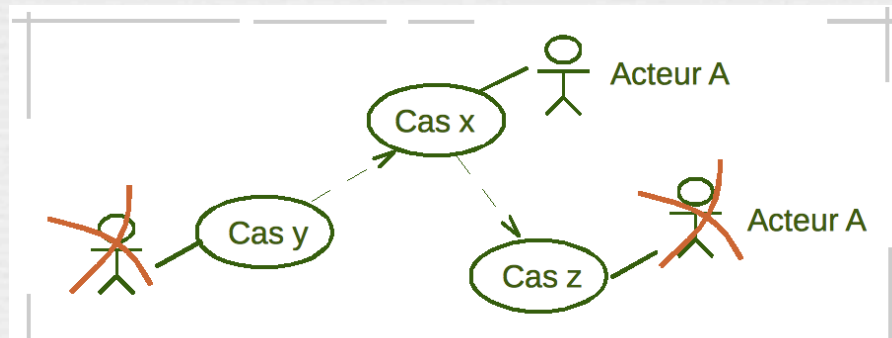


La relation **Spécialisation/** **Généralisation**


La relation *de spécialisation* permet d'expliciter différentes manières de «réaliser» un même cas d'utilisation

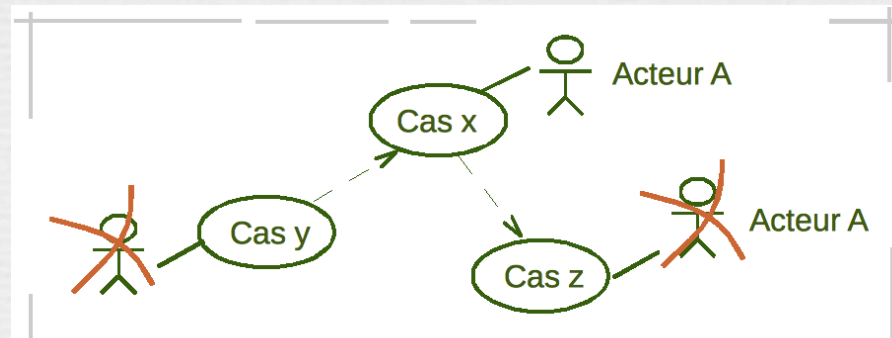


Relations entre UC : Includes et Extends



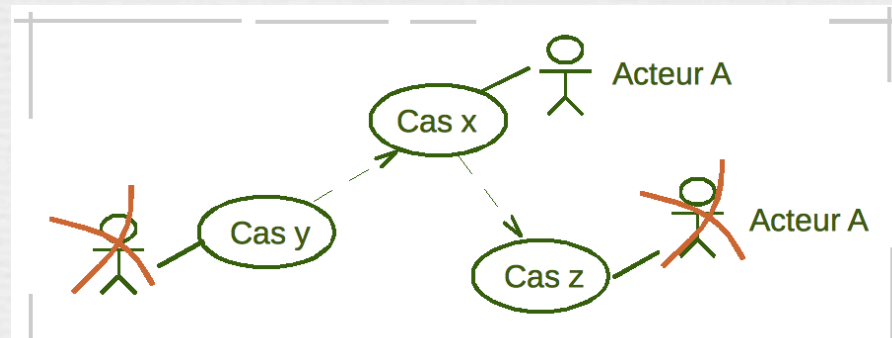
Relations entre UC : Includes et Extends

-  Il ne faut pas répéter les acteurs des cas d'utilisations principaux au niveau des cas d'utilisation étendus ou inclus



Relations entre UC : Includes et Extends




- Il ne faut pas répéter les acteurs des cas d'utilisations principaux au niveau des cas d'utilisation étendus ou inclus
- Répéter les acteurs des cas d'utilisations principaux indique que les cas d'utilisation inclus ou étendus sont aussi accessibles directement (i.e. Ils font partie des cas d'utilisations principaux)



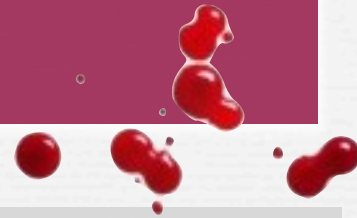
Mauvaises pratiques



Mauvaises pratiques

-  Mauvaise appréciation de la taille d'un cas d'utilisation, amenant à une multiplication de «petits» cas d'utilisation ;
-  Importance exagérée accordée à la forme graphique du diagramme de cas d'utilisation, et aux relations d'utilisation et d'extension entre use cases ;
-  Utilisation erronée des cas d'utilisation comme substitut à une décomposition fonctionnelle.

Confondre processus métier et cas d'utilisation



Symptômes : les événements de début et de fin du cas d'utilisation ne sont pas discernables par le système, on décrit dans le détail les interactions entre les acteurs eux-mêmes.

Titre : Enregistrer l'emprunt de documents

Acteurs : Bibliothécaire (principal), Client

Description : Ce Use Case commence lorsqu'un client arrive au comptoir de prêt pour emprunter des livres et des vidéos. Il les présente au bibliothécaire qui enregistre l'emprunt de chaque document après avoir vérifié l'identité de l'emprunteur. Le système édite un bulletin de prêt et le client part alors avec les documents empruntés.

Focus des use cases «système»

Symptômes : La description d'un use case doit se concentrer sur les interactions entre les acteurs et le système, pas sur les interactions entre les acteurs.

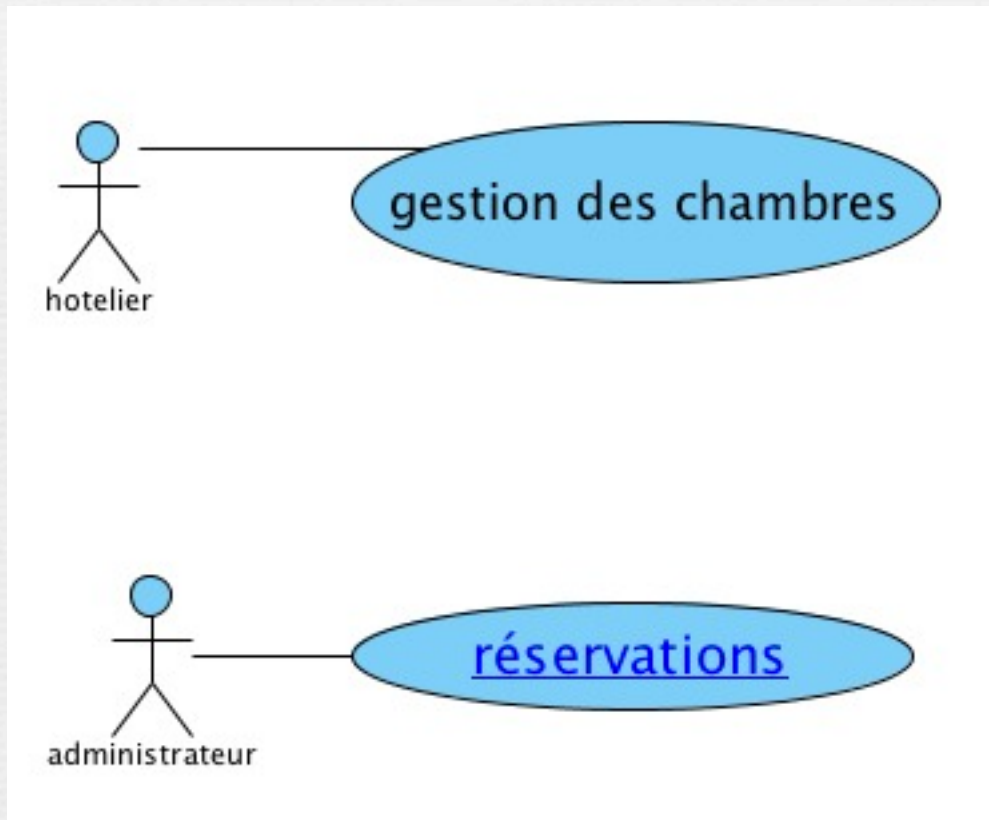
■ **Titre** | : Enregistrer l'emprunt de documents

■ **Acteurs** : Bibliothécaire (principal), Client

■ **Description** :

- *Ce Use Case (niveau métier) commence lorsque le client arrive au comptoir de prêt pour emprunter des livres et des vidéos.*
- *Ce Use Case (niveau système) commence lorsque le Bibliothécaire fournit l'identifiant de l'emprunteur.*
- ...
- *Ce Use Case (niveau système) se termine lorsque le bulletin de prêt (contenant les dates de retour, etc.) est imprimé par le système.*
- *Ce Use Case (niveau métier) se termine lorsque le Bibliothécaire donne le bulletin et les ressources prêtées au Client.*

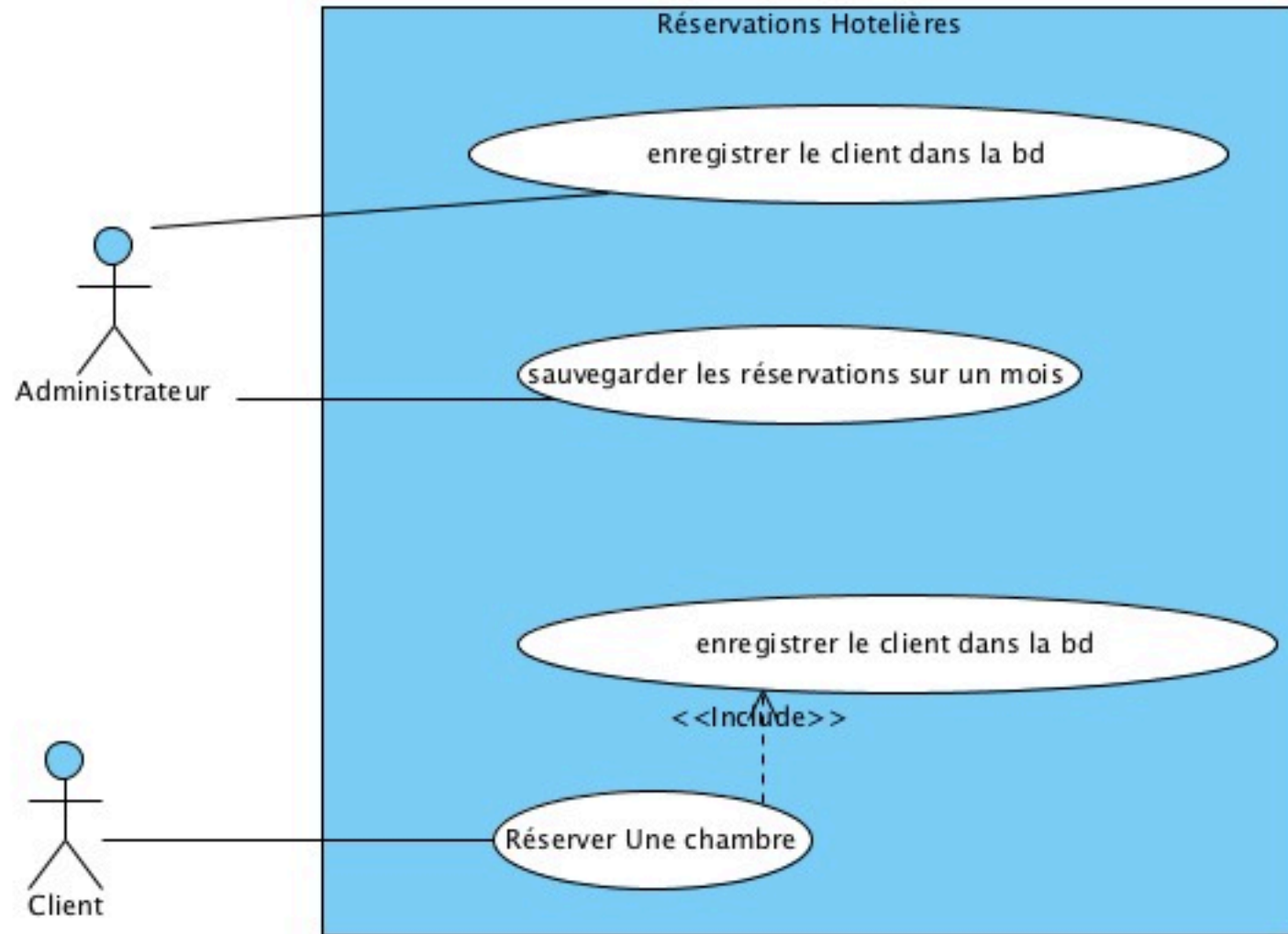
Que signifie ces UC?



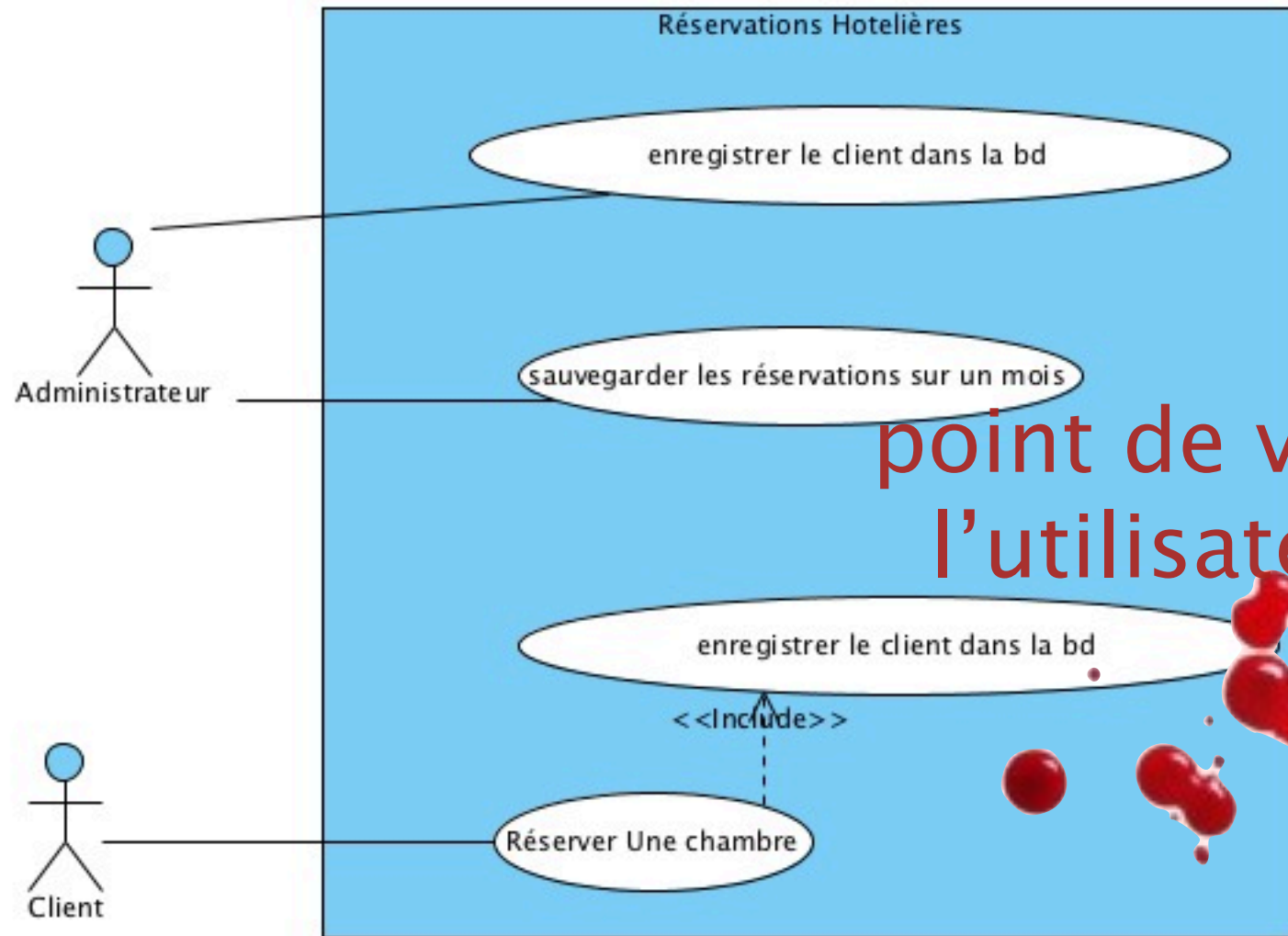
Quelle regle n'est pas respectée ?



Comment interprétez-vous ce diagramme?

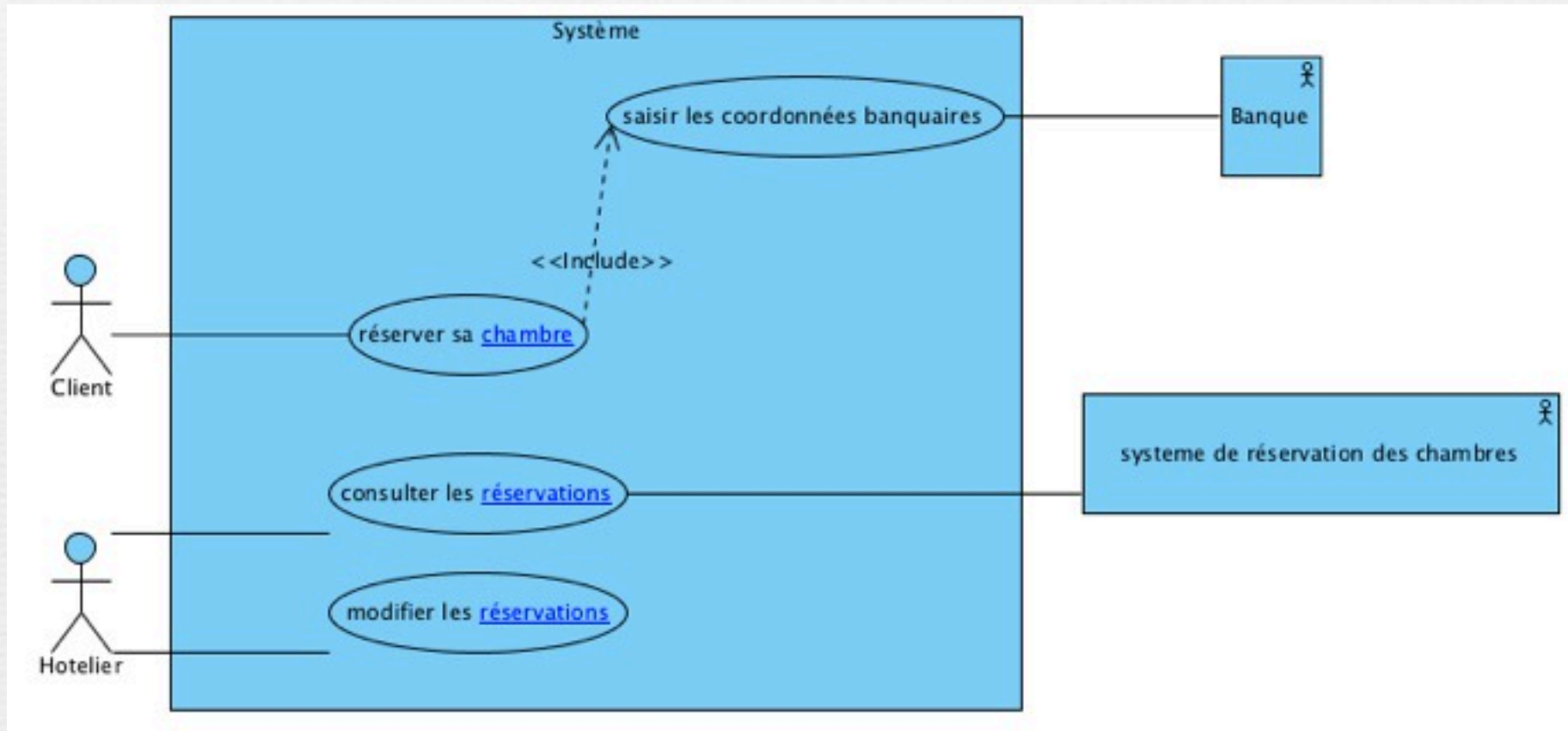


Comment interprétez-vous ce diagramme?

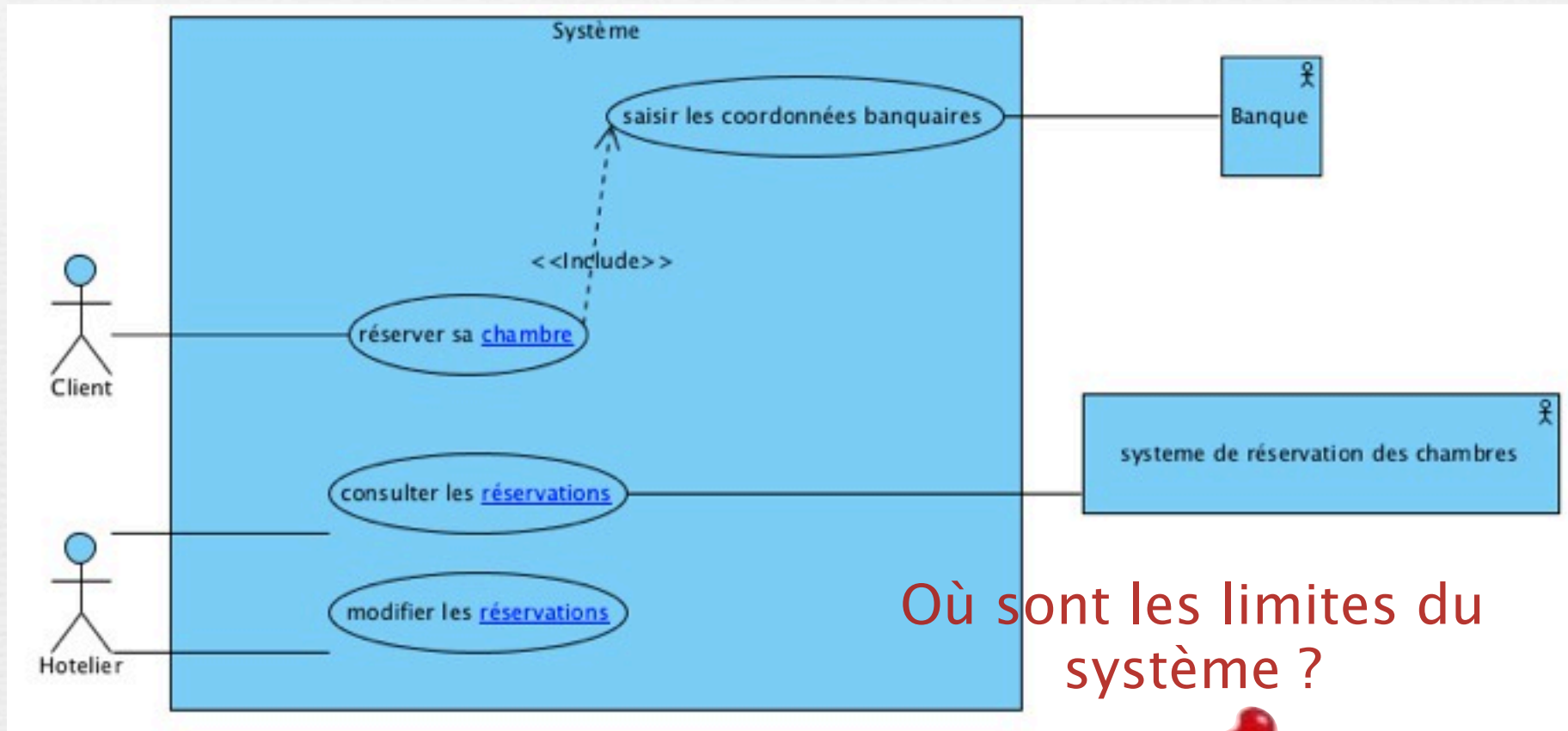


point de vue de l'utilisateur ?

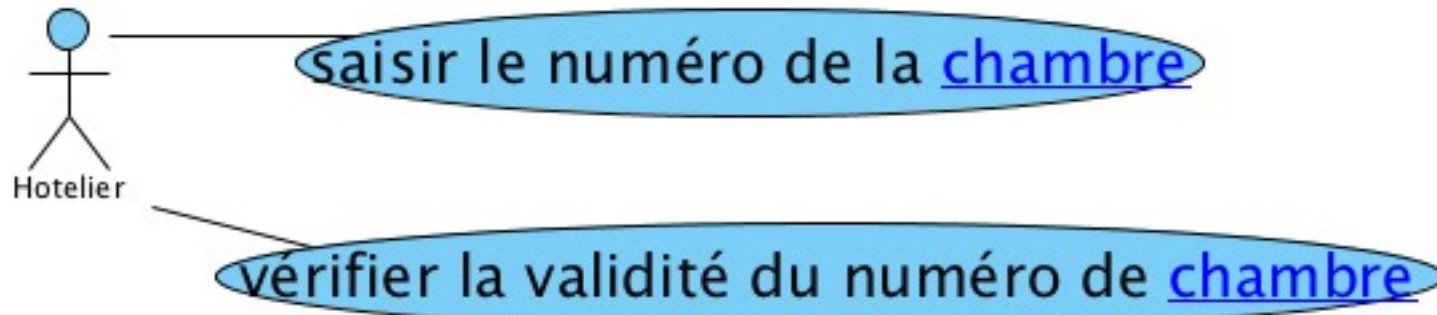
Comment interprétez-vous ce diagramme?



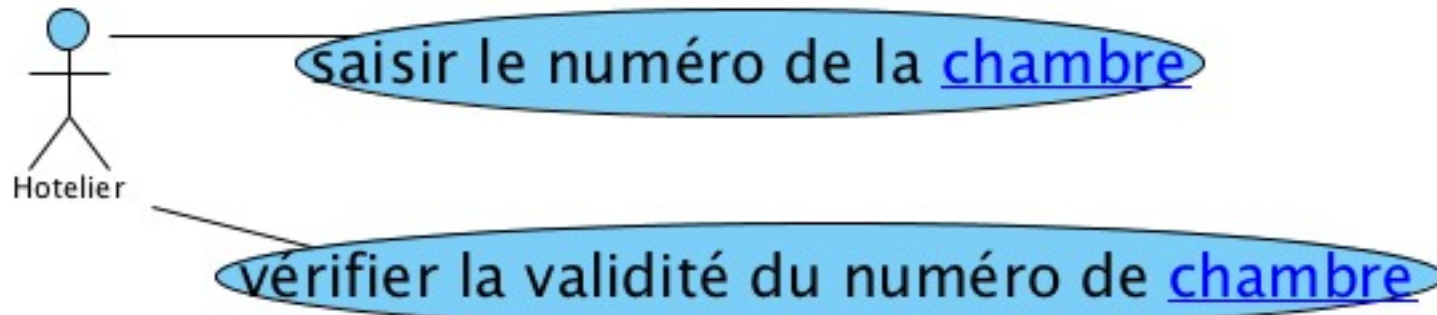
Comment interprétez-vous ce diagramme?



Que pensez-vous de ?



Que pensez-vous de ?



Des scénarios sont-ils
associés aux UC ?



Bénéfices des use-cases

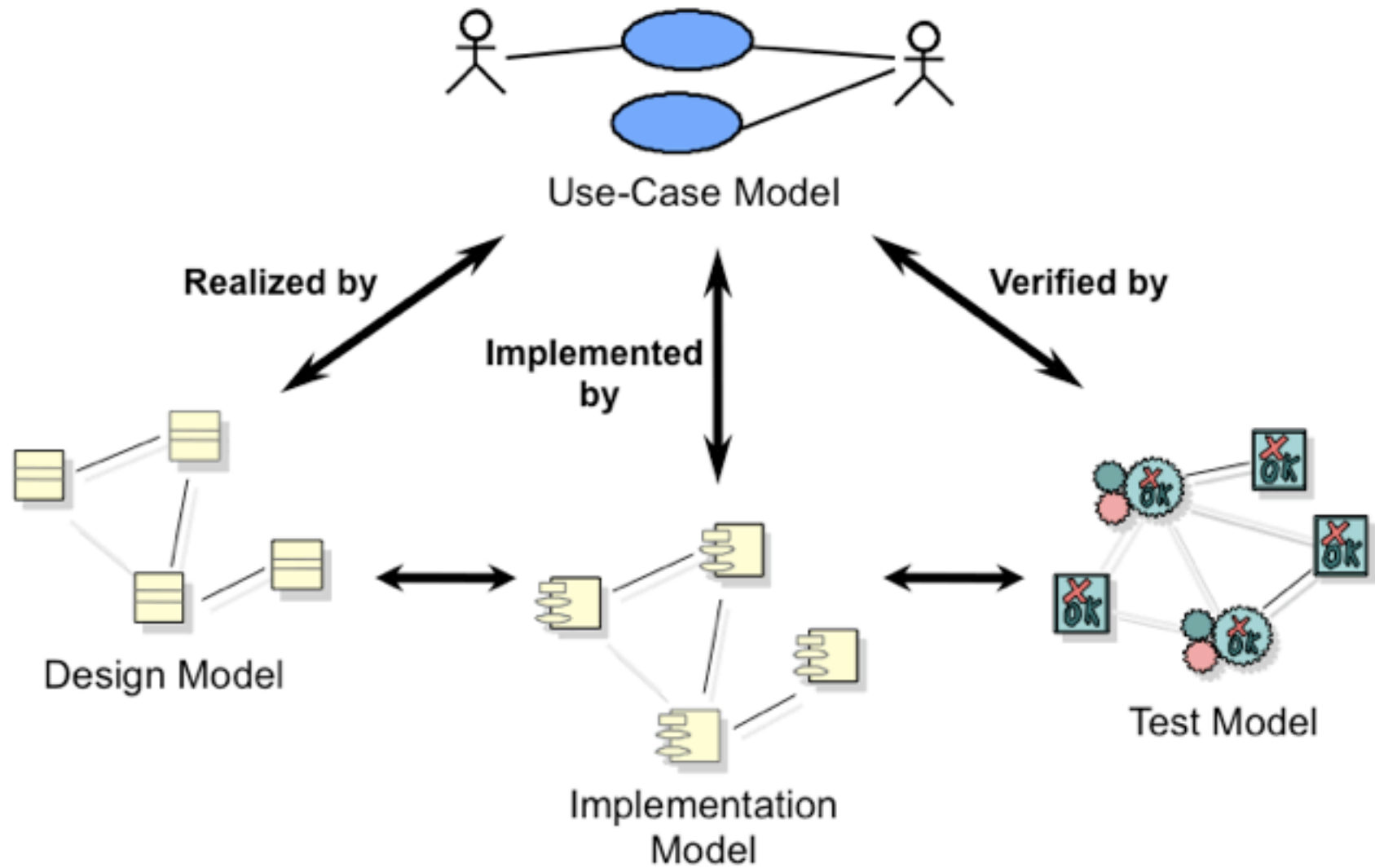
- Organisent les exigences d'un point de vue utilisateur
- Définissent les exigences du système comme des séquences logiques,
- Permettent de vérifier que toutes les exigences sont capturées et qu'elles correspondent à ce qu'attend le demandeur.
- Facilitent l'adéquation des demandeurs
 - mais aussi des cas de tests, la documentation et la réutilisation des exigences.

Des use-cases, pour qui?

- Demandeurs (décrire et approuver)
- Utilisateurs (comprendre)
- Architectes logiciels (identification des fonctions)
- Concepteurs et développeurs
- Testeurs (identifier les tests)
- Managers (Planifier)
- Rédacteurs de documentation (prendre un point de vue utilisateur)






Dev. logiciel dirigé par les use-cases



Résumé

Cas d'utilisation : résumé

- Se servir des Cas d'Utilisation UML pour identifier les exigences fonctionnelles.

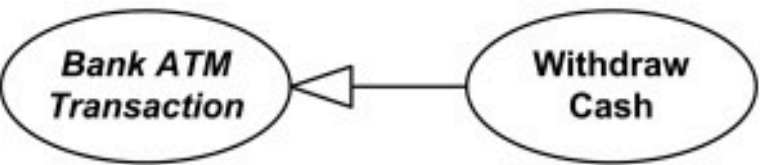

Construct	Description	Syntax
use case	A sequence of actions, including variants, that a system (or other entity) can perform, interacting with actors of the system.	
actor	A coherent set of roles that users of use cases play when interacting with these use cases.	
system boundary	Represents the boundary between the physical system and the actors who interact with the physical system.	



Cas d'utilisation : résumé

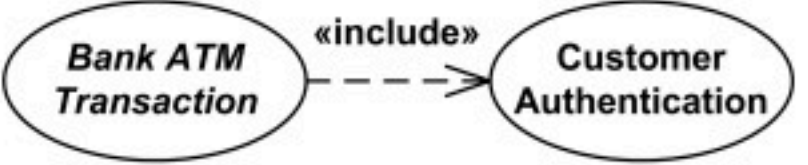
Construct	Description	Syntax
association	The participation of an actor in a use case. i.e., instance of an actor and instances of a use case communicate with each other.	—————
extend	A relationship from an <i>extension</i> use case to a <i>base</i> use case, specifying how the behavior for the extension use case can be inserted into the behavior defined for the base use case.	<<extend>> —————→
generalization	A taxonomic relationship between a more general use case and a more specific use case.	—————→
Construct	Description	Syntax
include	An relationship from a <i>base</i> use case to an <i>inclusion</i> use case, specifying how the behavior for the inclusion use case is inserted into the behavior defined for the base use case.	<<include>> —————→



Generalization	Extend
	
<p>Base use case could be abstract use case (incomplete) or concrete (complete).</p>	<p>Base use case is complete (concrete) by itself, defined independently.</p>
<p>Specialized use case is required, not optional, if base use case is abstract.</p>	<p>Extending use case is optional, supplementary.</p>
<p>No explicit location to use specialization.</p>	<p>Has at least one explicit extension location.</p>
<p>No explicit condition to use specialization.</p>	<p>Could have optional extension condition.</p>



<http://www.uml-diagrams.org/use-case-include.html>

Include

<p>Base use case is incomplete (abstract use case).</p>
<p>Included use case required, not optional.</p>
<p>No explicit inclusion location but is included at some location.</p>
<p>No explicit inclusion condition.</p>

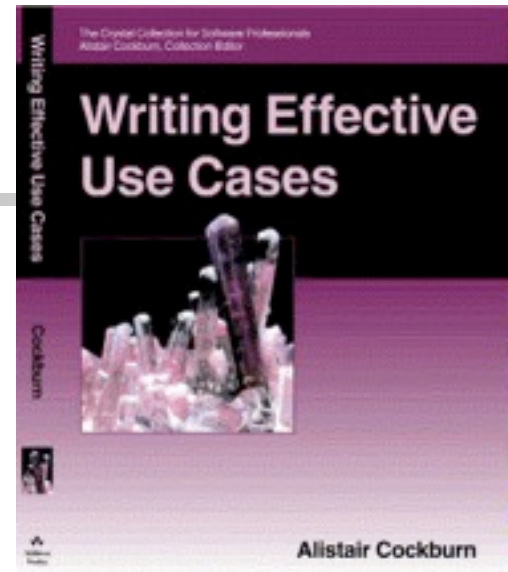
TCSS 360, Spring 2005

Lecture Notes

Use Cases

Relevant Reading:

Writing Effective Use Cases
A. Cockburn





Use cases

- **use cases:** written descriptions of user's interaction with the software product to accomplish a goal
 - (in a business system): "A sequence of transactions in a system whose task is to yield a result of measurable value to an individual actor of the business system"
 - (in an information system): "A behaviorally related sequence of transactions performed by an actor in a dialogue with the system to provide some measurable value to the actor" (Jacobson 1995)
- Use cases are the ways in which a system can be used (the functions which the system provides to its users)
- Use cases help us discover/document requirements



Benefits of doing use cases

- The list of goal names provides executives:
 - Shortest summary of what system will contribute
 - Project planning skeleton (priorities & timing)
- The main success scenario provides all:
 - Agreement as to the system's responsibilities
- The extension conditions provide programmers:
 - List of things programmers have to watch for
 - List of things analysts have to investigate
- The extension handling steps provide dev team:
 - Record of (tricky) business policy decisions



Cockburn's requirements list

Requirements Outline (p13-14) - good template of all functional requirements

1. purpose and scope
2. terms / glossary
- 3. use cases**
4. technology used
5. other
 - 5a. development process - participants, values (fast-good-cheap), visibility, competition, dependencies
 - 5b. business rules / constraints
 - 5c. performance demands
 - 5d. security (now a hot topic), documentation
 - 5e. usability
 - 5f. portability
 - 5g. unresolved / deferred
6. human issues: legal, political, organizational, training



Actors and stakeholders

- What is an actor? A primary actor?
 - **actor**: anything with behavior that acts on the system
 - **primary actor**: initiates interaction to achieve goal
(when system is a software product, primary actor is often the computer user)
 - **supporting actor**: performs sub-goals to help use case
- What is the difference between an actor and a stakeholder?

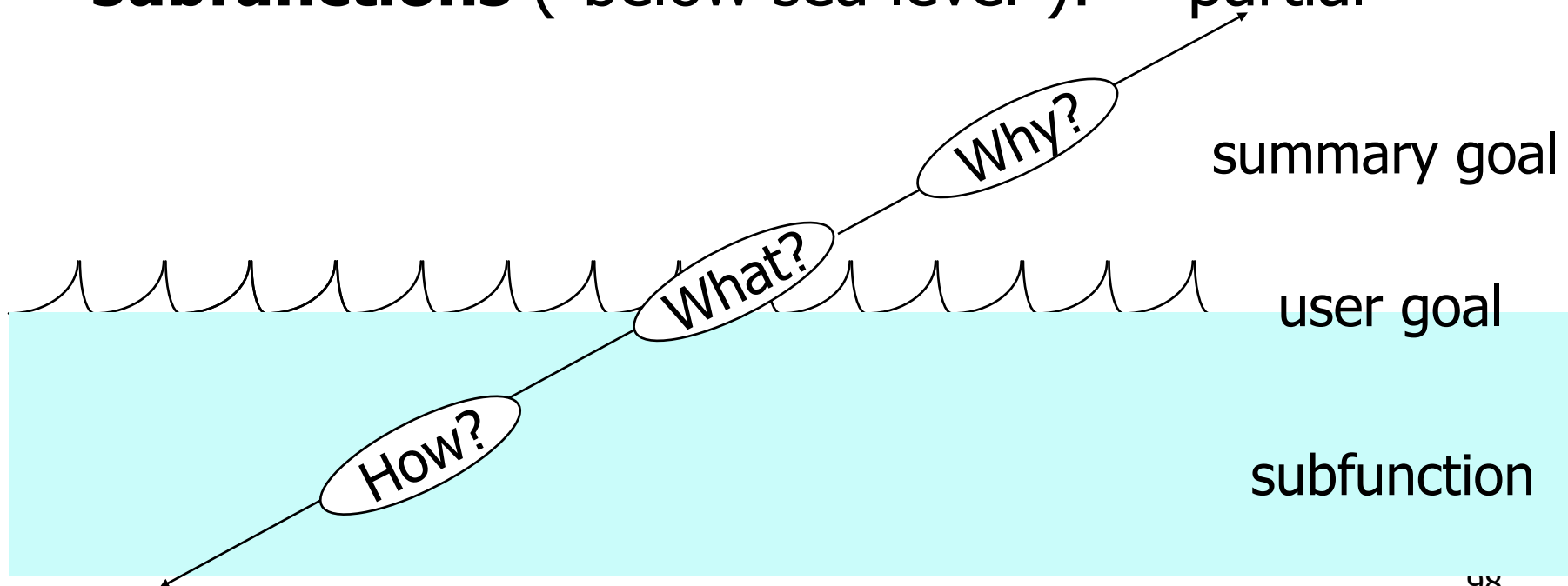


Actors and stakeholders

- What is an actor? A primary actor?
 - **actor**: anything with behavior that acts on the system
 - **primary actor**: initiates interaction to achieve goal
(when system is a software product, primary actor is often the computer user)
 - **supporting actor**: performs sub-goals to help use case
- What is the difference between an actor and a stakeholder?
 - **stakeholder**: anyone interested in the system
 - examples: supplier, stock agency, vendor
 - the difference: stakeholder might not "act" in any scenario

Use case goals and levels

- **goal**: action that actor wants to accomplish
- **level**: type / scope of a goal
 - **summary goals** ("above sea level"): multi-sitting
 - **user goals** ("sea-level"): one sitting
 - **subfunctions** ("below sea level"): partial





Goals and levels, examples

- Withdraw money from an ATM
 - level?
- Purchase a book from the online store, and have it shipped to the user; can be cancelled while in transit
 - level?
- Purchase shares of stock online using a "stock trap."
 - level?
- Update user's balance after a deposit.
 - level?



Goals and levels, examples

- Withdraw money from an ATM
 - level? **user goal**

- Purchase a book from the online store, and have it shipped to the user; can be cancelled while in transit
 - level? **summary goal**

- Purchase shares of stock online using a "stock trap."
 - level? **summary goal**

- Update user's balance after a deposit.
 - level? **subfunction**



Qualities of a good use case

- a good use case:
 - starts with a request from an actor to the system
 - ends with the production of all the answers to the request
 - defines the interactions (between system and actors) related to the function
 - takes into account the actor's point of view, not the system's
 - focuses on interaction, not internal system activities
 - doesn't describe the GUI in detail
 - has 3-9 steps in the main success scenario
 - is easy to read

Use cases vs. internal features

- consider software to run a cell phone:



Use Cases

- call someone
- receive a call
- send a message
- memorize a number

Point of view: user

Internal Functions

- transmit / receive data
- energy (battery)
- user I/O (display, keys, ...)
- phone-book mgmt.

Point of view: developer /
designer



Do use cases capture these?

- Which of these requirements should be represented directly in a use case?
 - Order cost = order item costs * 1.06 tax.
 - Promotions may not run longer than 6 months.
 - Customers only become Preferred after 1 year.
 - A customer has one and only one sales contact.
 - Response time is less than 2 seconds.
 - Uptime requirement is 99.8%.
 - Number of simultaneous users will be 200 max.



Do use cases capture these?

- Which of these requirements should be represented directly in a use case?
 - Order cost = order item costs * 1.06 tax.
 - Promotions may not run longer than 6 months.
 - Customers only become Preferred after 1 year.
 - A customer has one and only one sales contact.
 - Response time is less than 2 seconds.
 - Uptime requirement is 99.8%.
 - Number of simultaneous users will be 200 max.

Answer: **NONE**. Most of these requirements are non-functional, so the use cases wouldn't explicitly mention them. The user doesn't see them directly in the success scenario.



Styles of use cases

1. actor / goal list or UML use case diagram
 - shows all use cases in system
2. informal use case
3. formal use case

Let's examine each of these in detail...



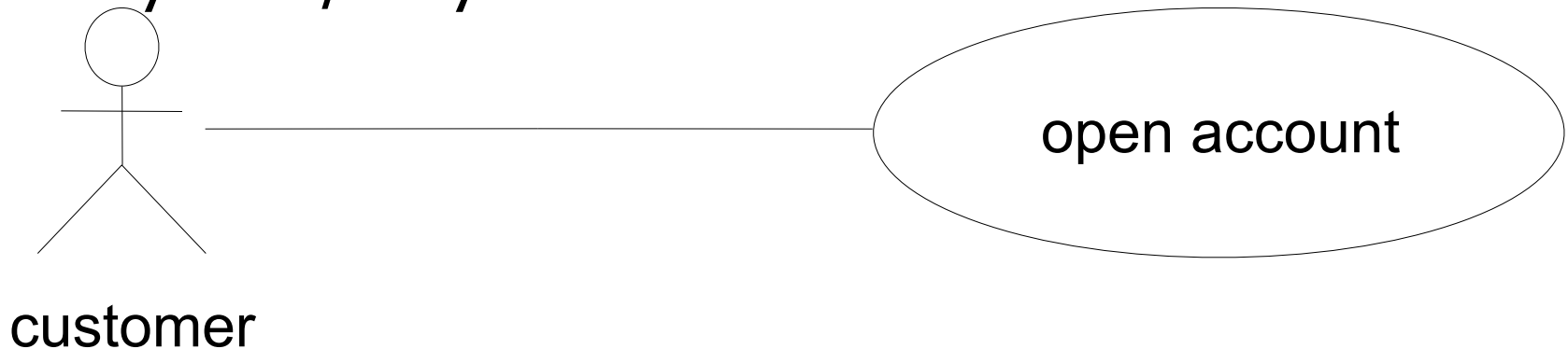
1. Actor / goal list

- it can be useful to create a list or table of actors and their "goals" (use cases they start):

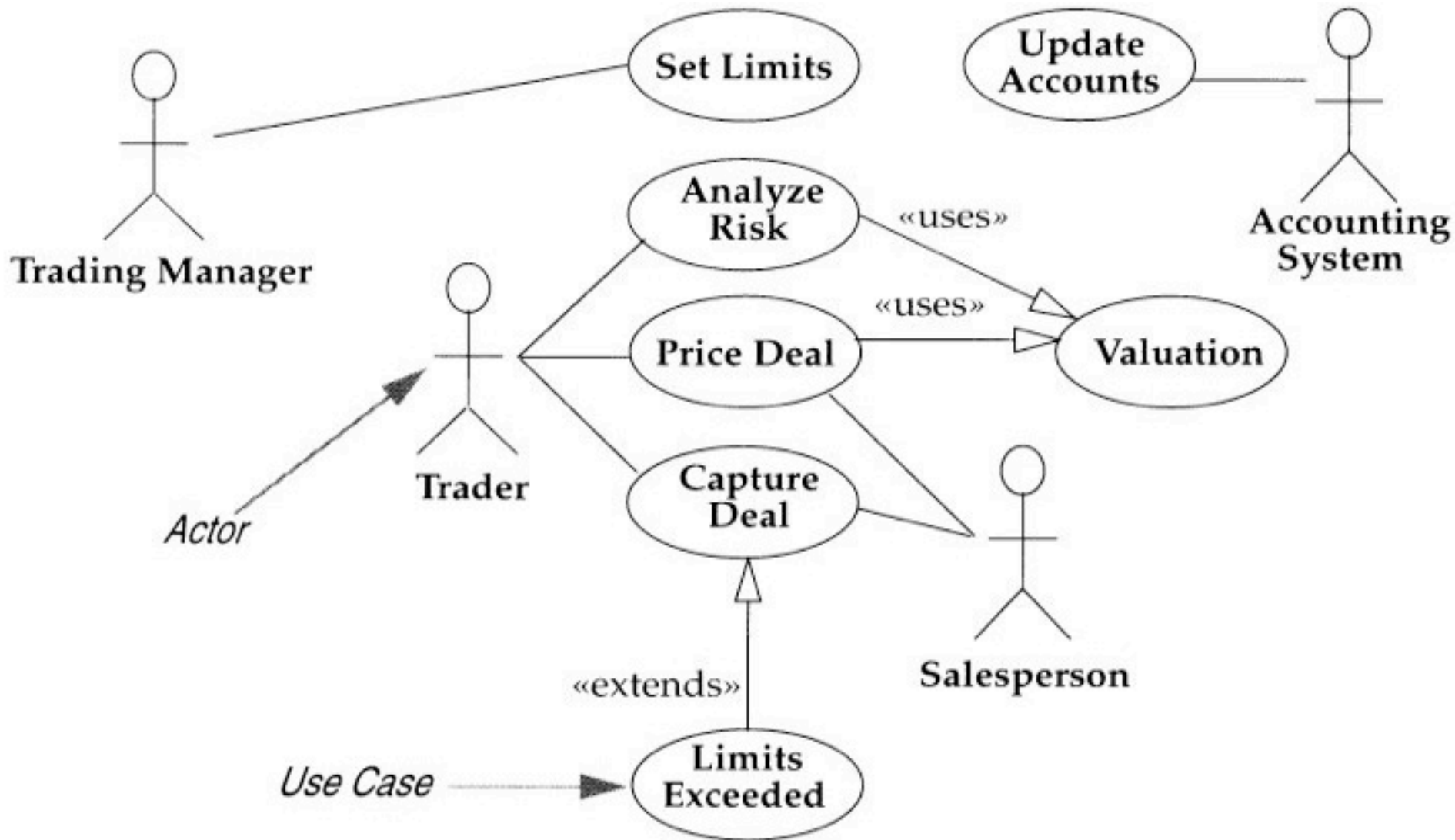
ACTORS		USE CASE
Club Member	Initiates	Submit Promotion Order Submit Regular Order
Potential Member	Initiates	Submit New Subscription
Past Member	Initiates	Submit Subscription Renewal
Membership Services Dept.	Initiates	Request Membership
Marketing Department	Initiates	Create New Monthly Promotion Create New Seasonal Promotion Create New Subscription Program Request Promotion Reports Request Sales Reports
Membership Services System	Initiates	Send New Subscription Offer Send Club Promotion Send Subscription Renewal

UML use case diagrams

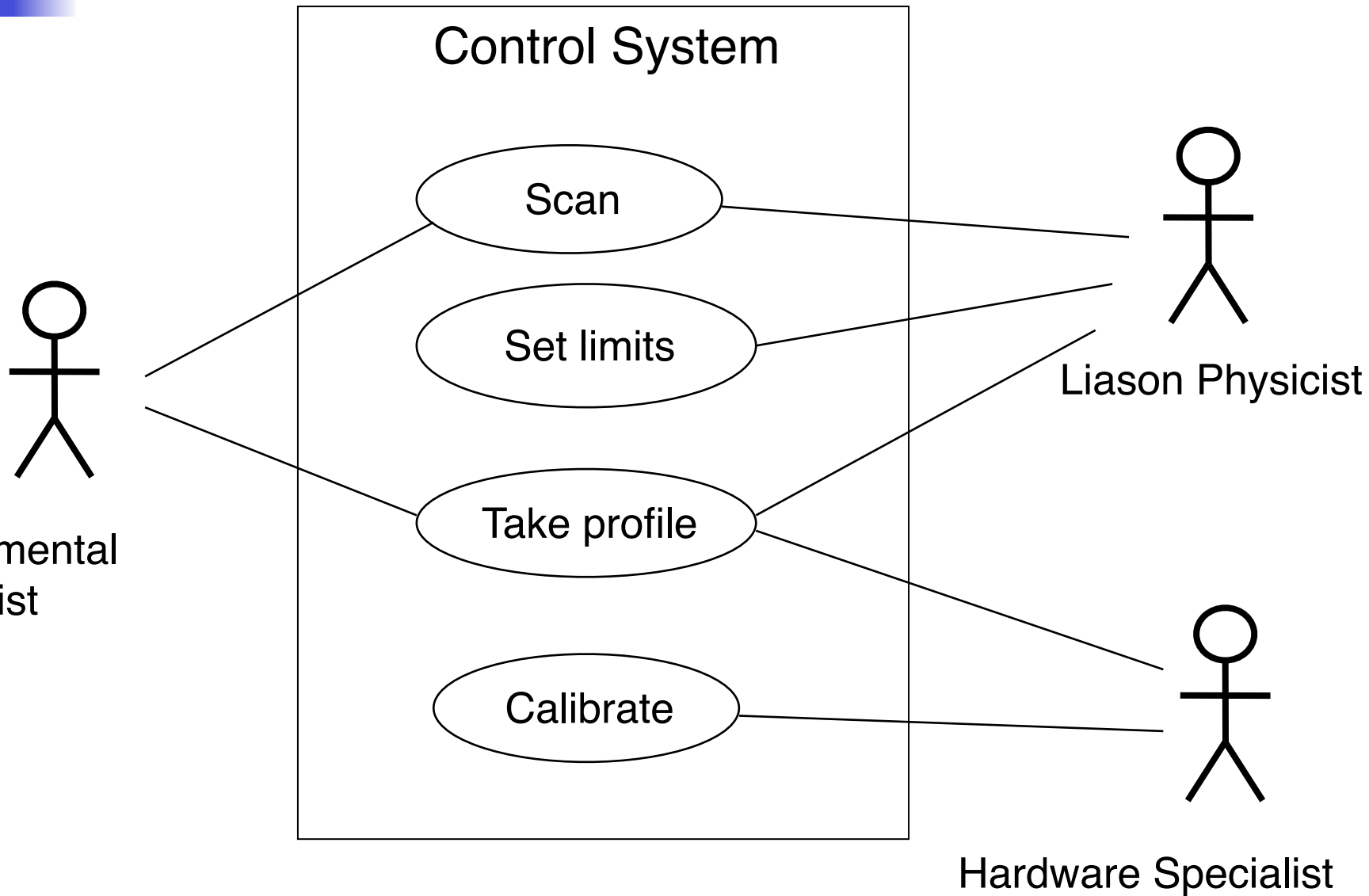
- use cases can be drawn as diagrams, with:
 - actors as stick-men, with their names below
 - use cases as ellipses with their names below or inside
 - association indicated by lines, connecting an actor to a use case in which that actor participates
 - use cases can be connected to other cases that they use / rely on



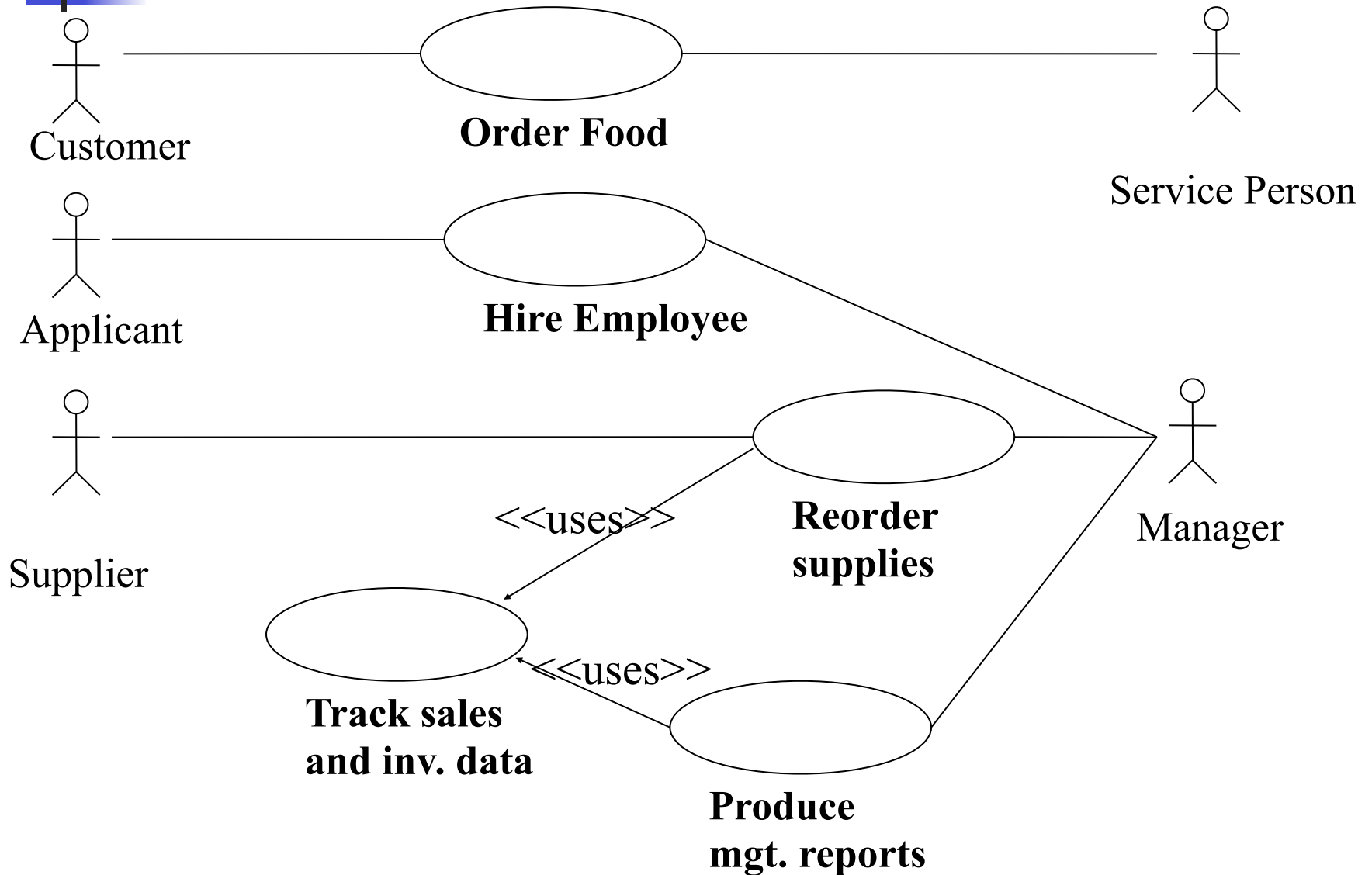
Example use case diagram



Example use case diagram 2



Example use case diagram 3





2. Informal use case

- **informal use case:** written as a paragraph describing the scenario
- Example:
 - Customer Loses a Tape

The customer reports to the clerk that he has lost a tape. The clerk prints out the rental record and asks customer to speak with the manager, who will arrange for the customer to pay a fee. The system will be updated to reflect lost tape, and customer's record is updated as well. The manager may authorize purchase of a replacement tape.



Formal use case example

Use Case 1 Buy Stocks over the Web

Primary Actor: Purchaser

Scope: Personal Advisors / Finance package (PAF)

Level: User goal

Stakeholders and Interests:

Purchaser—wants to buy stocks and get them added to the PAF portfolio automatically.

Stock agency—wants full purchase information.

Precondition: User already has PAF open.

Minimal Guarantee: Sufficient logging information will exist so that PAF can detect that something went wrong and ask the user to provide details.

Success Guarantee: Remote web site has acknowledged the purchase; the logs and the user's portfolio are updated.

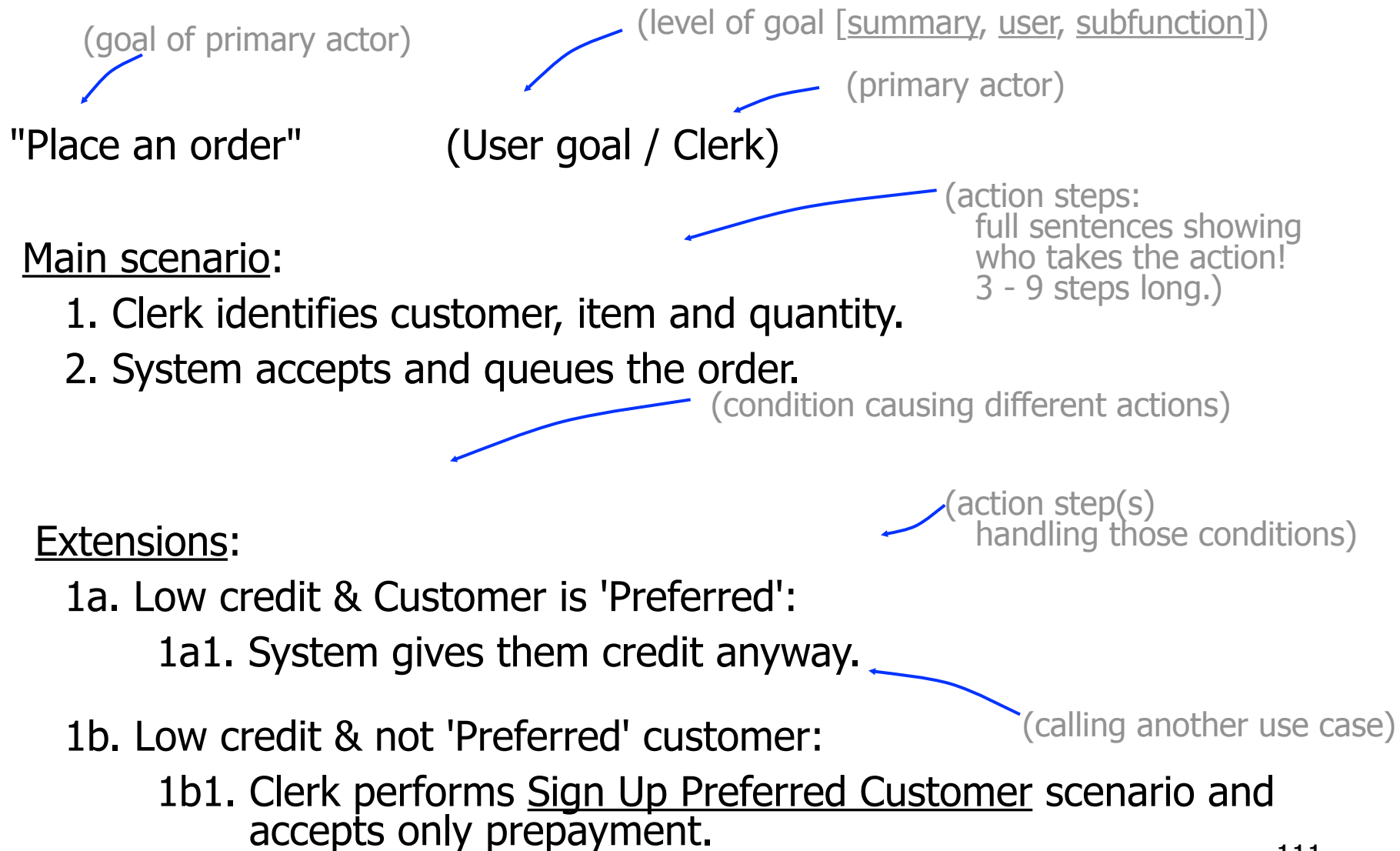
Main Success Scenario:

1. Purchaser selects to buy stocks over the web.
2. PAF gets name of web site to use (E*Trade, Schwab, etc.) from user.
3. PAF opens web connection to the site, retaining control.
4. Purchaser browses and buys stock from the web site.
5. PAF intercepts responses from the web site and updates the purchaser's portfolio.
6. PAF shows the user the new portfolio standing.

Extensions:

- 2a. Purchaser wants a web site PAF does not support:
 - 2a1. System gets new suggestion from purchaser, with option to cancel use case.
- 3a. Web failure of any sort during setup:
 - 3a1. System reports failure to purchaser with advice, backs up to previous step.
 - 3a2. Purchaser either backs out of this use case or tries again.
- 4a. Computer crashes or is switched off during purchase transaction:
 - 4a1. (What do we do here?)
- 4b. Web site does not acknowledge purchase, but puts it on delay:
 - 4b1. PAF logs the delay, sets a timer to ask the purchaser about the outcome.
- 5a. Web site does not return the needed information from the purchase:
 - 5a1. PAF logs the lack of information, has the purchaser update questioned purchase.

Formal use case elements





Example use case

Use Case 12. Buy stocks over the web

Primary Actor: Purchaser (user)

Scope: PAF

Level: user goal

Precondition: User already has PAF open.

Guarantees: sufficient log information exists that PAF can detect what went wrong.

Success Guarantees: remote web site acknowledged purchase, user's portfolio updated.

Main success scenario:

1. User selects to buy stocks over the web.
2. PAF gets name of web site to use (E*Trade, Schwabb, etc.)
3. PAF opens web connection to the site, retaining control.
4. User browses and buys stock from the web site.
5. PAF intercepts responses from the web site, and updates the user's portfolio.
6. PAF shows the user the new portfolio standing.

Extensions:



Use case tables

- formal use cases can also be written as a table:

USE CASE NAME	Submit Promotion Order
ACTOR	Club Member
DESCRIPTION	Describes the process when a club member submits a club promotion order to either indicate the products they are interested in ordering or declining to order during this promotion
Normal Course	1. This use is initiated when the club member submits the promotion order to be proceeded
	2. The club member's personal information such as address is validated against what is currently recorded in member services
	3. The promotion order is verified to see if product is being ordered
	4. The club member's credit status is checked with Accounts Receivable to make sure no payments are outstanding
	5. For each product being ordered, validate the product number
	6. For each product being ordered, check the availability in inventory and record the ordered information which includes "quantity being ordered" and give each ordered product a status of "open"
	7. Create a Picking Ticket for the promotion order containing all ordered products which have a status "open"
	8. Route the picking ticket to the warehouse
PRECONDITION	Use case <i>send club promotion</i> has been processed
POST CONDITION	Promotion order has been recorded and the picking ticket has been routed to the warehouse
ASSUMPTIONS	



One method to do use cases

Now that we know the syntax for doing use cases, what 4 steps does Cockburn recommend when actually brainstorming and writing our use cases?

Let's look at each step in detail...



One method to do use cases

Now that we know the syntax for doing use cases, what 4 steps does Cockburn recommend when actually brainstorming and writing our use cases?

1. identify actors and their goals
2. write the main success scenario
3. identify and list possible failure extensions
4. describe how the system handles each failure

Let's look at each step in detail...



1. Identify actors and goals

Ask oneself the following questions:

- what computers, subsystems and people will drive our system? (actors)
 - examples: Customer, Clerk, Corporate Mainframe
- what does each actor need our system to do?
 - each need may show up as a trigger to a use case

result: a list of use cases, a sketch of the system

- short, fairly complete list of usable system function
- can now draw UML use case diagram for reference



Identify actors/goals example

- exercise:

Together, let's identify some major actors and their goals for software for a video store kiosk system. The software can be used for looking up movies and actors by keywords, as well as usable to check out movies from the kiosk to known customers, without a cashier present. A customer can check out up to 3 movies at a time, for up to 5 days each. If a movie is returned late, late fees can be paid at the time of return or time of next checkout. The data is stored internally in a database system, which communicates with the kiosk. The manager of the store can log in to update employee data.



2. Write the success scenario

- main success scenario is the preferred "happy" case
 - example: customer=good credit and item=in stock
 - easiest to read and understand
 - everything else is a complication on this
- capture each actor's intent and responsibility, from trigger to goal delivery
 - say what information passes between them
 - number each line

exercise: Let's do this for the Customer Returns a Movie scenario.



3. List the failure extensions

- usually, almost every step can fail
 - example: customer has bad credit
 - example: item is not in stock in desired quantity
- note the failure condition separately, after the main success scenario

exercise: Let's do this for the Customer Returns a Movie scenario.



4. Describe failure-handling

- recoverable extensions rejoin main course
 - example: low credit + valued customer -> accept
 - example: low stock + reduce quantity -> accept
- non-recoverable extensions fail directly
 - not a valued customer -> decline order
 - out of stock -> decline order
- each scenario goes from trigger to completion
 - "extensions" are merely a writing shorthand
 - can write "if" statements
 - can write each scenario from beginning to end

exercise: Let's do this for the Customer Returns a Movie scenario.



Pros and cons of use cases

pro:

- they hold functional requirements in an easy-to-read text format
- they make a good framework for non-functional requirements & project scheduling

con:

- they show only the functional reqs



User stories (usage narratives)

- **user story:** narrative told from user's perspective, describing his/her usage of the system
- Example:

Bill is a marine biologist who wants to see an article about fish. He selects "Article or journal" from the menu. He chooses topic "fish" from the subsequent list shown. The system returns articles to Bill about his chosen topic. The annotated list designates the physical location of articles. Bill clicks articles of interest to him. Abstracts of each flagged article are displayed. Bill makes a final selection of articles based on abstracts. The abstracts are printed, and Bill retrieves them from the printer.
- How is this different from an informal use case?



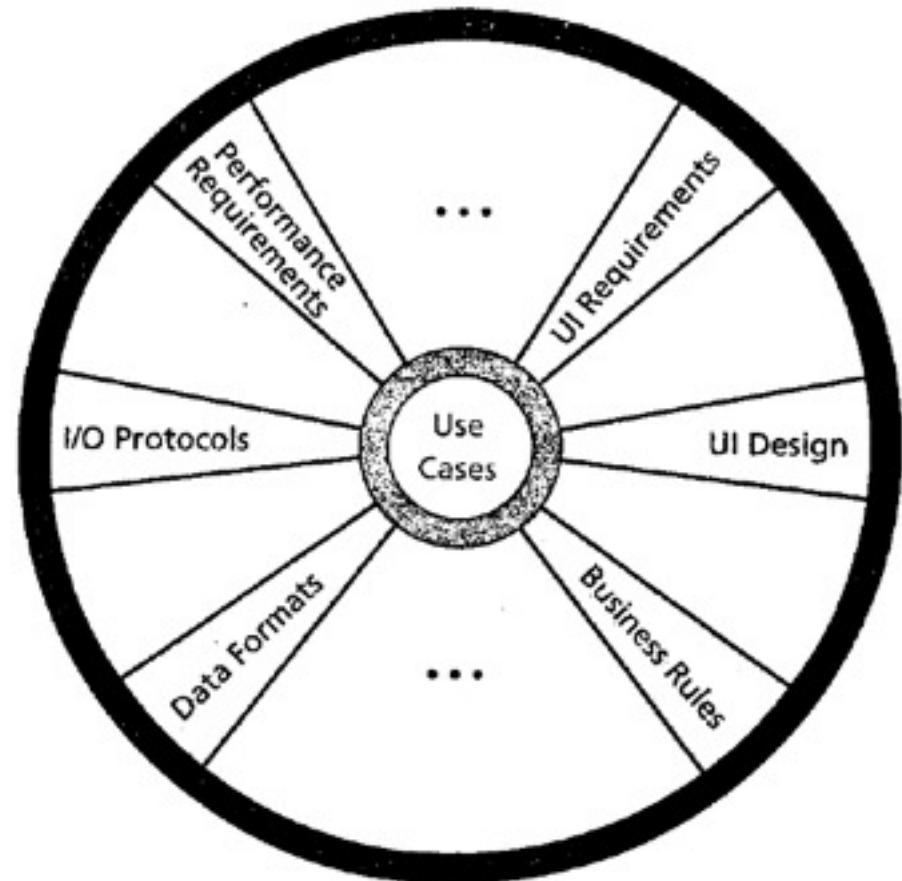
User stories (usage narratives)

- **user story:** narrative told from user's perspective, describing his/her usage of the system
- Example:

Bill is a marine biologist who wants to see an article about fish. He selects "Article or journal" from the menu. He chooses topic "fish" from the subsequent list shown. The system returns articles to Bill about his chosen topic. The annotated list designates the physical location of articles. Bill clicks articles of interest to him. Abstracts of each flagged article are displayed. Bill makes a final selection of articles based on abstracts. The abstracts are printed, and Bill retrieves them from the printer.
- How is this different from an informal use case?
 - too personal; too focused on UI; contains non-software details (printing)

How do use cases fit in?

- "Hub and spokes" model puts use cases as central to all requirements
 - Adolph's "Discovering" Requirements in New Territory
 - What do you think?
- use cases help us discover functional requirements in our system and document them
 - Do use cases affect UI design decisions?





Use case exercises

- Consider the case of a video store that wants a kiosk with intelligent software that can replace human checkout workers. A customer with an account can simply use their membership and credit card with a reader at the kiosk to check out a video.
 - Come up with 5 use case names for such software, and draw a UML use case diagram of these cases and their actors.
 - Write a formal (complete) use case for the Customer Checks Out a Movie scenario.