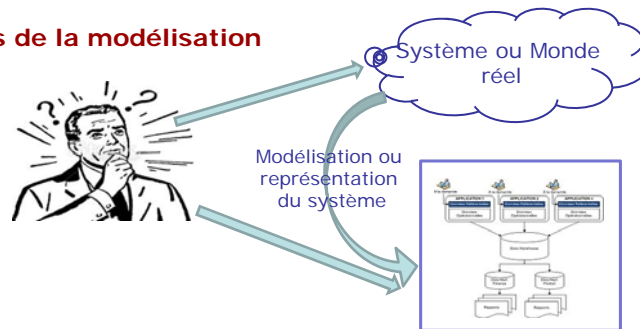


Introduction

▪ Rappel: les axes de la modélisation



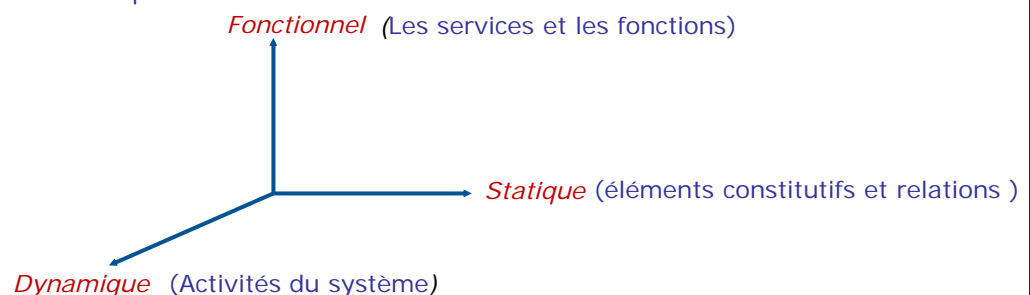
- ✓ La modélisation d'un système est une tâche assez complexe, en raison notamment de ses multiples caractéristiques.
- ✓ UML permet de réduire la complexité en produisant plusieurs modèles (diagrammes) et en abordant les systèmes selon trois axes.
- ✓ Chaque axe correspond à une perspective, et ne prend compte dans la modélisation que certains aspects du système et ignore les autres.

Introduction

▪ Rappel: les axes de la modélisation

UML permet de représenter un système selon trois points de vue distincts et complémentaires:

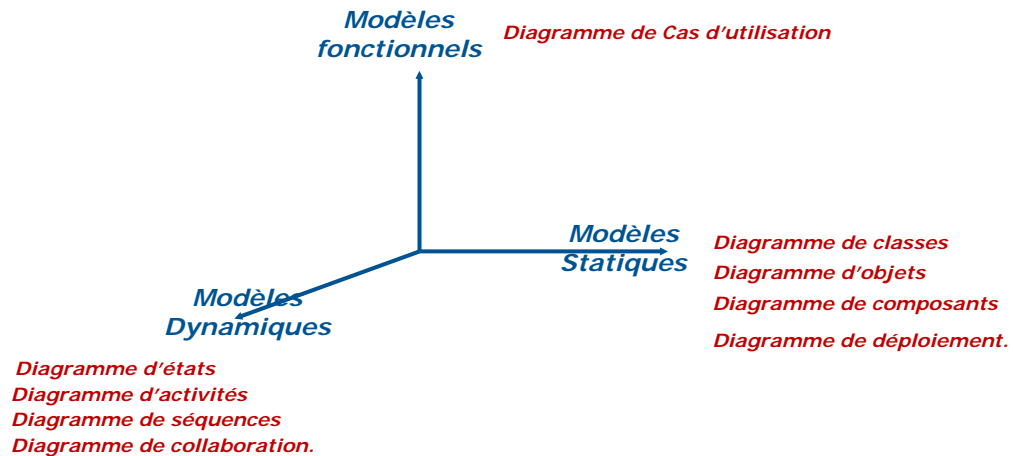
- ✓ La *structure du système*: Ce que le système est.
- ✓ La *fonction du système* : Ce que le système fait.
- ✓ La *dynamique du système* : Quelles sont les activités qui se déroulent au sein du système et comment est-ce qu'elles évoluent?



Introduction

▪ Rappel: les axes de la modélisation

UML rend compte des trois points de vue à travers trois types de modèles:



Introduction

▪ Le modèle Fonctionnel :

✓ Apporte une réponse à la question fondamentale suivante :

▪ Que fait le système ?

✓ Fait abstraction (ignore) des questions relatives à l'organisation et aux moyens :

▪ Comment, qui , quand et avec quoi ?

✓ Traduit les attentes et la vision du système de la part de ses utilisateurs.

- Les **Utilisateurs** sont les personnes auxquelles est destiné le système à construire; ils auront à exploiter le futur système pour exercer leur métier.
- Ces personnes sont aussi appelés « **Acteurs Métier** ».

✓ Permet une **bonne compréhension** du système à développer, et aussi de **s'assurer** en phase de conception, que les **besoins des utilisateurs** sont pris en charge par le système .

Introduction

▪ Le modèle Fonctionnel :

- ✓ Le système à construire devra donc répondre aux attentes de ses **Utilisateurs**.
- ✓ Pour cela les **Utilisateurs** sont amenés à exprimer leurs attentes, on dit aussi leurs **Besoins** ou encore leurs **Exigences**, vis à vis du système à construire.
- ✓ Les **Besoins** sont les conditions auxquelles le système à construire, doit satisfaire.
- ✓ Il est donc fondamental dans tout projet SI, de:
 - Rechercher,
 - Communiquer, et
 - Mémoriser les **Besoins des Utilisateurs**.
- ✓ L'expression des besoins doit se faire, sous une forme clairement compréhensible par les **Utilisateurs** et l'**Equipe** chargé de la réalisation du Système.

Introduction

▪ Le modèle Fonctionnel:

Il y'a lieu de distinguer deux types de besoins :

- ✓ Les besoins fonctionnels : Ils précisent les fonctions et les services rendus par le système.
- ✓ Les besoins non fonctionnels : Ils expriment les conditions d'exécution des fonctions du système: **Performance**, **Sécurité**, **confidentialité**, **disponibilité** etc.

Le modèle fonctionnel se limite à la description des Besoins Fonctionnels

La modélisation fonctionnelle du système nécessite l'identification :

- ✓ Des utilisateurs potentiels du système à construire.
- ✓ Des fonctions et des services que le système doit assurer pour supporter les activités de ses utilisateurs.
- ✓ Des relations entre utilisateurs et fonctions.

Concepts de Base

Le modèle fonctionnel fait appel aux Concepts de Base suivants:

- ✓ Acteur.
- ✓ Cas d'utilisation.
- ✓ Relation.

Ces concepts sont combinés et regroupés dans un diagramme appelé Diagramme des Cas d'Utilisation.

Concepts de Base

▪ Le concept d'Acteur:

Définition: Un **acteur** représente un **rôle**, ou un ensemble cohérent de rôles, joué par des entités: une personne ou une chose, externes au système à construire, et qui interagissent avec ce système.

Un acteur peut être :

- ✓ Un utilisateur humain.
- ✓ Un dispositif matériel.
- ✓ Un autre système.

Remarques:

- ✓ Le même rôle peut être joué par des personnes différentes. Les différentes personnes sont représentées par un même acteur.
- ✓ Le même individu peut jouer par des rôles différents. Dans ce cas Il y'aura autant d'acteurs que de rôles.

Concepts de Base

▪ Le concept d'Acteur:

Exemple : Système « gestion de la scolarité».

Messieurs Kzaz et Boutabia, sont deux individus différents, ils jouent le même rôle vis-à-vis du système. Ils auront tous les deux la même vision du système et de ses fonctionnalités; et ils **interagiront de la même manière** avec le système.

Le système « Gestion de la scolarité » devra assurer pour les deux professeurs les fonctionnalités suivantes :

- ✓ Consulter le planning de leurs enseignements.
- ✓ Enregistrer les absents d'une séance.
- ✓ Enregistrer les notes des étudiants.
- ✓ Etc.

Ceci nous conduit à considérer que les deux professeurs jouent le même rôle; ils seront par conséquent représentées par un seul acteur : Professeur.

Concepts de Base

▪ Le concept d'Acteur:

Exemple : Système « gestion de la scolarité».

Mr Boutabia est un utilisateur du système; il a deux casquettes, il est professeur et responsable de filière. Il va avoir deux visions différentes du même système et jouer deux rôles différents.

En plus de son rôle de Professeur, il jouera un second rôle, celui Responsable de Filière.

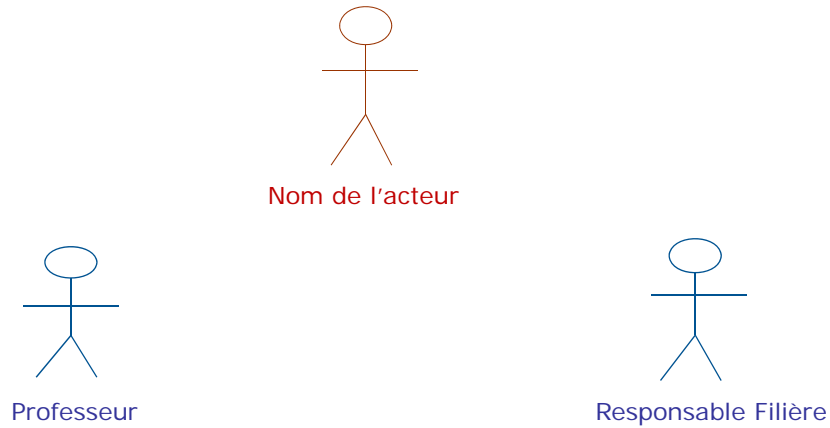
Le système devra lui assurer des fonctionnalités qui correspondent à ce second rôle:

- ✓ Consulter les notes de cours de l'ensemble des professeurs de sa filière.
- ✓ Affecter un enseignement à un professeur.
- ✓ Consulter l'état de la saisi des notes par les professeurs de sa filière.
- ✓ Etc.

Concepts de Base

Le concept d'Acteur: Représentation UML

Représentation des acteurs Humains:



Concepts de Base

Le concept d'Acteur: Représentation UML

Représentation des acteurs non Humains:



Deux représentations possibles des acteurs de type Système ou Machine

Concepts de Base

▪ Le concept d'Acteur: Représentation UML

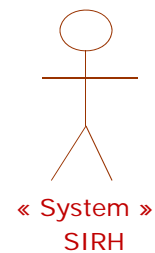
Exemple :

Le Système d'Information sur les Ressources Humaines est en interaction avec celui de la « Gestion de la Scolarité » de l'établissement:

- Partage des données sur les professeurs.
- Volume des cours assurés pour la paie des vacataires.



Ou bien



Concepts de Base

▪ Le concept d'Acteur:

Les acteurs se déterminent en observant les utilisateurs directs du système: ceux qui sont chargés de son exploitation, de son administration et de sa maintenance.

Application : Déterminer les acteurs des systèmes suivants:

- Gestion de la scolarité: Konosys
- Plateforme E-learning: Black Board
- Bibliothèque en ligne; ScholarVox
- Guichet Automatique de Banque

Concepts de Base

Le concept de Cas d'Utilisation: (Use Case)

Définitions:

- ✓ Un UC est une manière spécifique d'utiliser un système.
- ✓ Un UC est l'image d'une fonctionnalité du système déclenchée en réponse à la stimulation d'un acteur externe.
- ✓ Un UC exprime un besoin fonctionnel d'un utilisateur-acteur.

Exemples: ✓ Consulter le planning de enseignements de la semaine.

- ✓ Enregistrer les absents d'une séance.
- ✓ Enregistrer les notes des étudiants.
- ✓ Affecter un enseignement à un professeur.
- ✓ Consulter l'état de la saisi des notes par les professeurs de sa filière.

Cours UML Modèle Fonctionnel

Pr. L.Kzaz

Concepts de Base

Le concept de Cas d'Utilisation: Représentation UML

Nom du Cas d'Utilisation

Exemples :

Consulter le planning des enseignements

Enregistrer les absences d'une séance

Affecter un enseignement à une filière

Cours UML Modèle Fonctionnel

Pr. L.Kzaz

Concepts de Base

▪ Description des Cas d'Utilisation

La description des UC consiste à décrire la chronologie des interactions entre l'utilisateur et le système lors de la mise en œuvre d'un UC.

- ✓ Les UC sont décrits sous formes d'actions et de réactions, le comportement d'un système du point de vue de son utilisateur.
- ✓ La description d'un UC se fait en langage naturel, selon différents niveaux de précision.
- ✓ Le niveau de précision de la description dépend des objectifs de l'étape en cours.

En phase d'analyse, on se contentera d'une description permettant de :

- Comprendre globalement La fonction ou le service décrit par l'UC.
- Connaître ce que doit faire (le rôle de) le système.
- Connaître ce que doit faire (le rôle de) l'acteur.
- Évaluer la complexité du cas.

Concepts de Base

▪ Description des Cas d'Utilisation

Exemple: Description de l'UC: « Consulter le planning des enseignements », concernant l'acteur « Professeur »

1. Le Professeur s'identifie.
2. Le Système authentifie le professeur et ouvre une session.
3. Le Système affiche le planning des cours du Professeur.
4. Le Professeur se déconnecte.
5. Le Système ferme la session.

Concepts de Base

▪ Scénarios des Cas d'Utilisation

- ✓ Un UC représente une fonctionnalité ou un service rendu par le système pour un utilisateur.
- ✓ La description d'un UC met en évidence les interactions entre acteur et système.
- ✓ La description complète d'un UC doit inclure tous les **comportements significatifs** pour une fonctionnalité donnée.
- ✓ Chaque **comportement significatif** correspond à un déroulement réel « **une instanciation** » de l'UC: il est appelé **scénario**.
- ✓ La description complète d'un UC doit inclure tous les **scénarios significatifs**.

Concepts de Base

▪ Scénarios des Cas d'Utilisation

On distingue trois types de scénarios :

- ✓ Le **scénario nominal**: il correspond au comportement normal. Ce comportement est le plus fréquent du cas.
- ✓ Les **scénarios alternatifs**: ils correspondent à des situations alternatives au scénario nominal. Ils correspondent à des comportements secondaires.
- ✓ Les **scénarios d'exception**: ils correspondent à des situations exceptionnelles, d'erreurs ou d'interruption du déroulement du cas.

Concepts de Base

▪ Scénarios des Cas d'Utilisation

La séquence suivante est un exemple de scénario possible de l'UC « Enregistrer les Absences » .

1. Le Professeur s'identifie.
2. Le Système authentifie le professeur et ouvre une session.
3. Le Système affiche la liste des étudiants de la séance.
4. Le Professeur fait l'appel et note les absents.
5. Le Système enregistre les absences.
6. Le Professeur termine la session.
7. Le Système ferme la session.

Concepts de Base

▪ Scénarios des Cas d'Utilisation

La séquence suivante est un autre Scénarios possible de l'UC « Enregistrer les Absences » .

1. Le Professeur s'identifie.
2. Le Système authentifie le professeur et ouvre une session.
3. Le Système affiche la liste des étudiants absents à la séance.
4. Le Professeur modifie la liste des absents.
5. Le Système modifie et enregistre les absences.
6. Le Professeur termine la session.
7. Le Système ferme la session.

Concepts de Base

▪ Scénarios des Cas d'Utilisation

La séquence suivante est un troisième Scénarios possible de l'UC « Enregistrer les Absences » .

1. Le Professeur s'identifie.
2. Le Système authentifie le professeur et ouvre une session.
3. Le Système affiche la liste des étudiants absents à la séance.
4. Le Professeur annule l'appel et supprime les absences
5. Le Système supprime les absences de la séance.
6. Le Professeur termine la session.
7. Le Système ferme la session.

Concepts de Base

▪ Description des Cas d'Utilisation

Modèle de fiche de description d'un UC. (non normalisé par UML)

Cas d'utilisation : Titre du Cas.	Auteur :	Date:
Version :		
Acteurs :		
Pré conditions :		
Scénario nominal : Description du scénario nominal :		
Scénarios alternatifs : Description des scénarios alternatifs		
Scénarios Exceptionnels : Description des scénarios Exceptionnels		
Postconditions :		

Concepts de Base

▪ Concept de Relation

On distingue les relations entre :

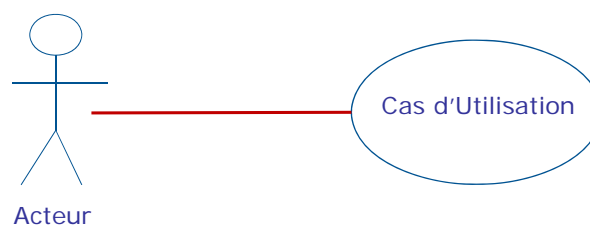
- ✓ Acteur et Cas d'utilisation.
- ✓ Acteurs.
- ✓ Cas d'Utilisation.

▪ Relation entre Acteur et Cas d'utilisation:

- ✓ Elle exprime l'existence d'une ou de plusieurs interactions entre l'acteur et le UC.
- ✓ L'acteur déclenche et initialise l'exécution du UC.
- ✓ Il n'y a pas d'intermédiaire entre l'acteur et le système qui exécute le UC .
- ✓ L'acteur est le bénéficiaire du ou des résultats de l'exécution du UC.

Concepts de Base

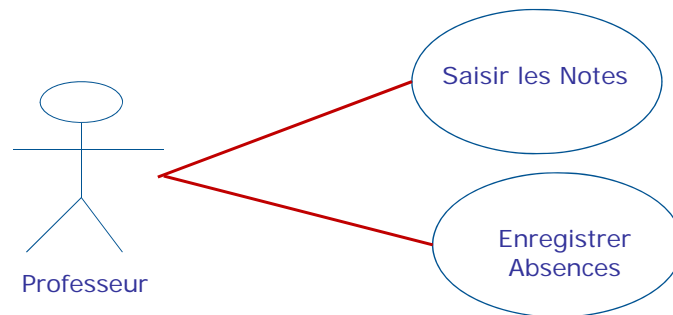
▪ Relation entre Acteur et Cas d'utilisation:



Représentation UML de la relation « Acteur », « Cas d'Utilisation »

Concepts de Base

Relation entre Acteur et Cas d'utilisation:

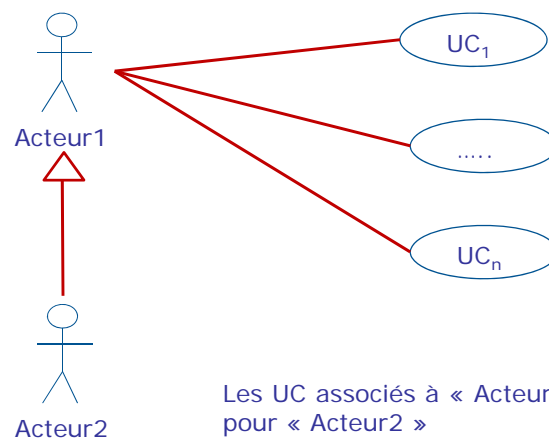


Représentation UML de la relation « Acteur », « Cas d'Utilisation »

Concepts de Base

Relation entre Acteurs:

La seule relation qui existe entre acteurs est la « **Généralisation** »

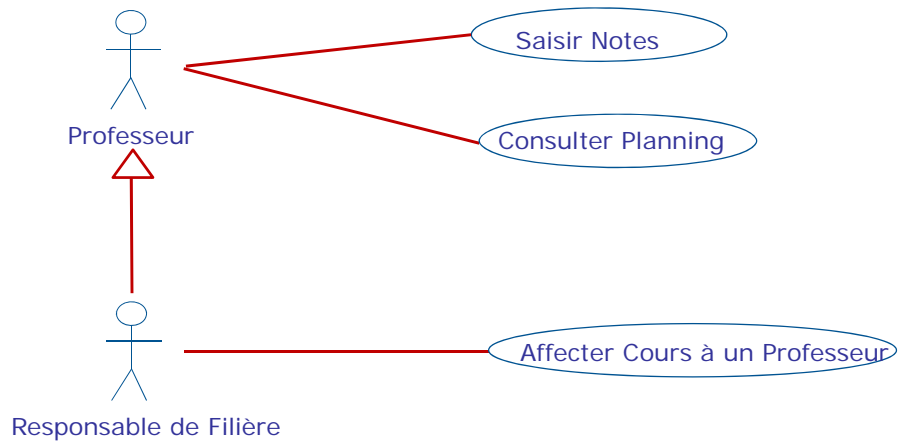


Les UC associés à « Acteur1 » sont également disponibles pour « Acteur2 »

Concepts de Base

Relation entre Acteurs:

Exemple:



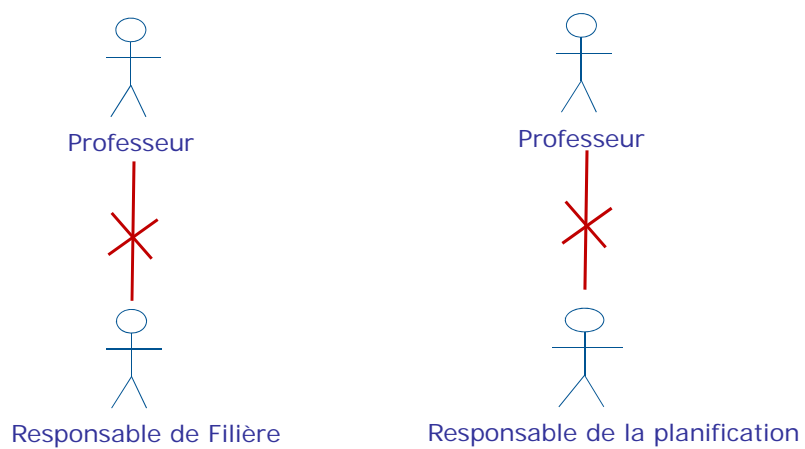
Cours UML Modèle Fonctionnel

Pr. L.Kzaz

Concepts de Base

Relation entre Acteurs:

La communication entre acteurs n'est pas modélisée



Cours UML Modèle Fonctionnel

Pr. L.Kzaz