



Méthodologie de conception



Introduction à la Modélisation Orientée Objet

Dr Ismahène Hadj Khalifa

1

Critères de qualité d'un logiciel



- Utilité (Adéquation entre le logiciel et les besoins des utilisateurs)
- Utilisabilité
- Fiabilité
- Interopérabilité (Interactions avec d'autres logiciels)
- Performance
- Portabilité
- Réutilisabilité
- Facilité de maintenance

2

Etapes du développement



1. Étude de faisabilité
2. Spécification (Déterminer les fonctionnalités du logiciel)
3. Conception (Déterminer la façon dont le logiciel fournit les différentes fonctionnalités recherchées)
4. Codage
5. Tests
6. Maintenance

3

Pourquoi modéliser?

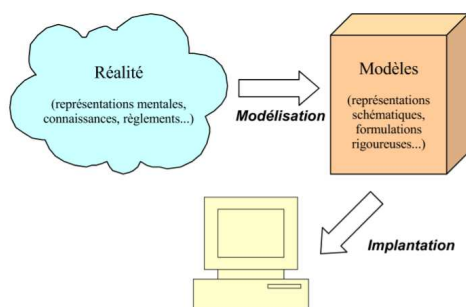


Modéliser un système avant sa réalisation permet de :

- mieux comprendre le fonctionnement du système.
- maîtriser sa complexité
- assurer sa cohérence.

4

Modélisation



5

Modèle



- Un modèle est une représentation abstraite de la réalité.
- Il permet de réduire la complexité d'un phénomène en éliminant les détails qui n'influencent pas son comportement de manière significative.
- Il reflète ce que le concepteur croit important pour la compréhension et la prédiction du phénomène modélisé.

Est-ce possible de construire un bâtiment sans plan?

6

Modèle



Qu'est-ce qu'un « bon » modèle ?

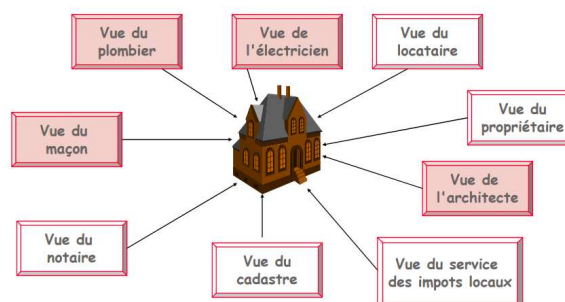
A est un bon modèle de B si A permet de répondre de façon satisfaisante à des questions prédéfinies sur B (d'après D.T. Ross)

Un bon modèle doit donc être construit :

- au bon niveau de détail
- selon le bon point de vue.

7

Multiples vues



8

Langage de modélisation



Un langage de modélisation doit définir :

- La sémantique des concepts
- Une notation pour la représentation de concepts
- Des règles de construction et d'utilisation des concepts.

9

UML



UML est un langage conçu pour:

- visualiser
- spécifier
- construire
- documenter

les artefacts d'un système à forte composante logicielle

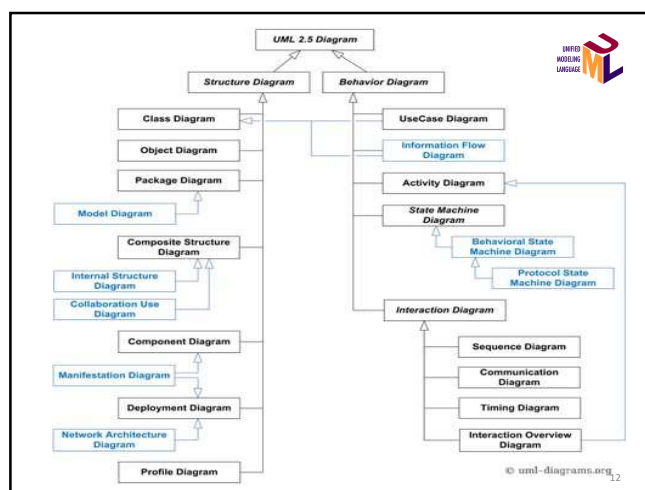
10

UML



- **UML est un Langage de modélisation graphique et textuel destiné à :**
 - Comprendre et décrire des besoins
 - Spécifier et documenter des systèmes
 - Tracer des solutions logicielles
 - Concevoir des solutions et communiquer des points de vue
- **UML unifie les notations et les concepts orientés objets**
 - Il ne s'agit pas d'une simple notation
 - Les concepts transmis par un simple diagramme
 - ont une sémantique précise
 - et sont porteurs de sens au même titre que les mots d'un langage

11





Chapitre 1

Diagramme de cas d'utilisation

13

Etapes du développement



1. Étude de faisabilité
2. **Spécification des besoins**
3. Conception
4. Codage
5. Tests
6. Maintenance

14



Diagramme des cas d'utilisation



Modélisation des besoins

Qu'est ce que le logiciel doit faire?

15

Diagramme de cas d'utilisation



Un diagramme de cas d'utilisation est modélisé par :

- des acteurs qui utilisent le système
- les « services » offerts par le système.

16

Intérêts des cas d'utilisation



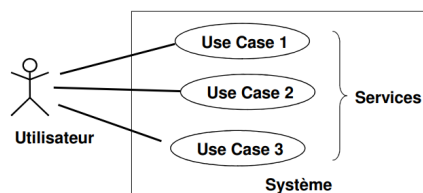
- Recenser l'expression des besoins sur les utilisateurs
- Faciliter la structuration des besoins des utilisateurs
- Permettre une représentation simple et expressive
- Eviter de dériver vers des spécifications inadaptées et inutiles
- Obliger les utilisateurs à définir la manière dont ils voudraient interagir avec le système
- Exprimer les limites et les objectifs du système (fonctionnalités + champ de l'étude)

17

Diagramme de cas d'utilisation



L'utilisateur et le système



18

Diagramme de cas d'utilisation



Diagramme de cas d'utilisation répond aux questions suivantes :

- Quelles sont les tâches principales réalisées par les acteurs (entités matériels ou logiciels externe au logiciel qui entrent en interaction) ?
- Les informations manipulées par les acteurs ?
- Les informations manipulées par le logiciel ?
- Le diagramme contient les acteurs, les cas d'utilisation (services) et les applications.

19

Principaux concepts des diagrammes de cas d'utilisation



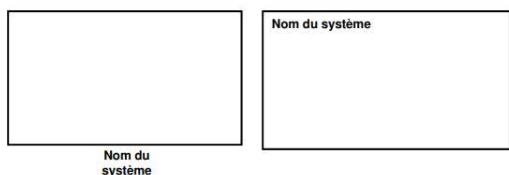
- Acteurs
- Système
- Cas d'utilisation
- Relations (entre cas d'utilisation, entre acteurs, entre acteurs et cas d'utilisation)

20

Système



- Le système est un ensemble de cas d'utilisation
- Le système ne comprend pas les acteurs.



21

Acteur



- Un acteur représente un rôle joué par une entité externe (utilisateur humain, dispositif matériel, ou autre système) qui interagit avec le système étudié.
- Un acteur est une entité externe agissant sur le système qui peut:
 - Echanger de l'information avec le système ou
 - Consulter ou modifier l'état du système

Notation:



22

Acteurs vs. utilisateurs



- Ne pas confondre la notion d'Acteur et de personne utilisant le système:
 - Une même personne physique peut jouer le rôle de plusieurs acteurs
Ex. : Maurice est un Chef d'agence et est aussi un client de la banque.
 - 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

23

Recensement des acteurs



Comment ?

- Par un dialogue avec le client et les utilisateurs ;
- en repérant les frontières du système.

Qui sont-ils ?

- Des utilisateurs humains : utilisateurs du logiciel à travers son interface graphique, par exemple;
- des périphériques manipulés par le système (imprimantes, capteurs, ...) ;
- des logiciels déjà disponibles à intégrer dans le projet : disponibles qui communiquent avec le système grâce à une interface logicielle (API, ODBC,...)
- attention à ne pas oublier les acteurs qui administrent le système

24

Recensement des acteurs



Un acteur peut être:

- **Principal**: utilise les fonctions principales du système
- **Secondaire**: effectue des tâches administratives ou de maintenance
- **Matériel externe**:
 - Les dispositifs matériels:
 - Autres que les machines sur lesquelles s'exécute l'application
 - Faisant partie du domaine de l'application et nécessaires au fonctionnement du système

25

Exemple



26

Cas d'utilisation



Un cas d'utilisation décrit un ensemble de séquences d'actions réalisées par le système et produit un résultat observable par un acteur.

- Un CU représente **une exigence fonctionnelle** envers le système dans son ensemble
- Un CU correspond à **une manière spécifique** d'utiliser le système
- C'est la représentation d'une **fonctionnalité** déclenchée en réponse à une simulation du système

27

Cas d'utilisation



- Notation des cas d'utilisation:



- Dans la pratique: les noms des cas d'utilisation sont des petites phrases verbales actives qui décrivent le comportement existant dans le vocabulaire du système en cours de modélisation
- Les cas d'utilisation sont activés par des acteurs

28

Acteur et Cas d'utilisation



Un acteur peut participer à des relations de **généralisation** avec d'autres acteurs.



29

Acteur et Cas d'utilisation



Les acteurs sont connectés aux CU par **une association** qui indique que l'acteur et le CU communiquent entre eux, chacun pouvant envoyer et recevoir des messages.



30

Relations entre CU

Relation d'inclusion:

- La relation d'inclusion a un caractère **obligatoire**
- La source doit indiquer à quel endroit le CU cible doit être inclus.
- Formalisée par le mot-clé <<includes>>: le cas d'utilisation de base en incorpore explicitement un autre, de façon obligatoire.



31

Relations entre CU

Relation d'extension

- Le CU source ajoute son comportement au CU destination, de façon **optionnelle**.
- La relation peut être soumise à une condition :
 - le comportement ajouté est inséré au niveau d'un point d'extension
 - Ce point d'extension est défini dans le CU destination

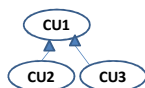


32

Relations entre CU

Relation de généralisation

- Le CU fils est une **spécialisation** du CU parent.
- Les CU descendants héritent de la description de leur parent commun. Chacun d'entre eux peut néanmoins comprendre des interactions spécifiques supplémentaires.



33

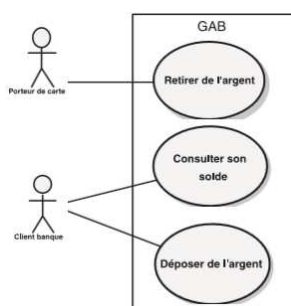
La démarche de construction d'un diagramme de CU

- Délimiter le champs de l'étude
- Identifier les acteurs
- Identifier les cas d'utilisation
- Organiser les cas d'utilisation par acteur
- Ajouter les relations entre cas d'utilisation
- Finaliser le diagramme de cas d'utilisation

34

Exemple de Diagramme de cas d'utilisation (Gestion d'un DAB)

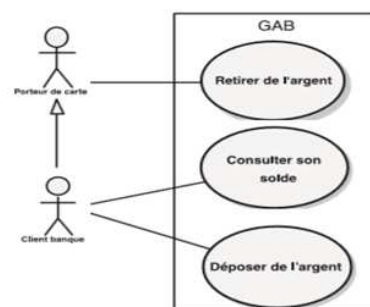
Acteurs primaires



35

Exemple de Diagramme de cas d'utilisation (Gestion d'un DAB)

Relations entre les acteurs primaires

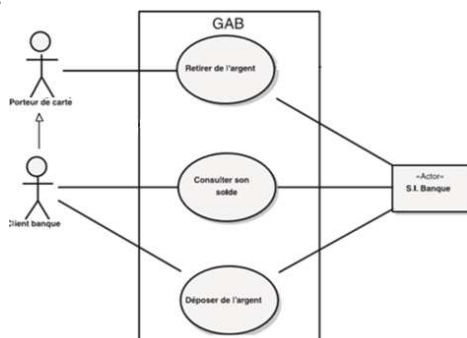


36

Exemple de Diagramme de cas d'utilisation (Gestion d'un DAB)



Ajout des acteurs secondaires



37

Description textuelle des CU



La représentation graphique des cas d'utilisation:

■ Donne une vue d'ensemble (sans détails) sur les différentes fonctionnalités du système.

■ Peut être affinée par les scénarios

Un CU est une collection de scénarios de succès et d'échec qui décrit la façon dont un acteur particulier utilise un système pour atteindre un objectif.

■ **Définition:** un scénario est une suite spécifique d'interaction entre les acteurs et le système. C'est une instance du CU, un chemin particulier dans sa combinatoire.

38

Description textuelle des CU



Dans la description d'un CU on trouve:

- **Un scénario nominal:** celui qui permet de satisfaire les objectifs des auteurs; le chemin le plus direct du succès. Le CU se réalise comme prévu par l'utilisateur.
- **Des extensions** qui regroupent tous les autres scénarios du succès (alternatifs) ou d'échecs (exceptions).
 - **Un scénario alternatif:** le CU se réalise mais pas comme prévu par l'utilisateur.
 - **Un scénario d'échec:** le Cas d'utilisation ne se réalise pas.

Il n'existe pas de norme UML établie pour la description textuelle de cas d'utilisation.

39

Spécification textuelle d'un CU (1/2)



Sommaire d'identification

<Nom du cas d'utilisation>

- **Acteurs principaux/acteurs secondaires**
- **Objectif:** Une description résumée permettant de comprendre l'intention principale du cas d'utilisation
- **Pré-conditions:** définissent les conditions qui doivent être satisfaites pour que le CU puisse démarrer.
- **Post-conditions:** définissent l'état du système lorsque le CU se termine avec succès.
 - **Date de création :** Date de création du cas d'utilisation.
 - **Date de mise à jour :** Date de mise à jour du cas d'utilisation.
 - **Version :** Le numéro de la version.
 - **Responsable :** Le nom du responsable.

40

Spécification textuelle d'un CU (2/2)



Description des enchaînements

- **Scénario nominal**
- **Extensions** (scénarios alternatifs et scénarios d'exception)
- **Exigences supplémentaires :** définissent les exigences non fonctionnelles et les contraintes de conception se rapportant à la spécification du CU (sécurité, ergonomie,...)

41