

Département Informatique
Master1 SIGL
Cours: Ingénierie des Besoins

Chapitre 6
Spécification par la
modélisation

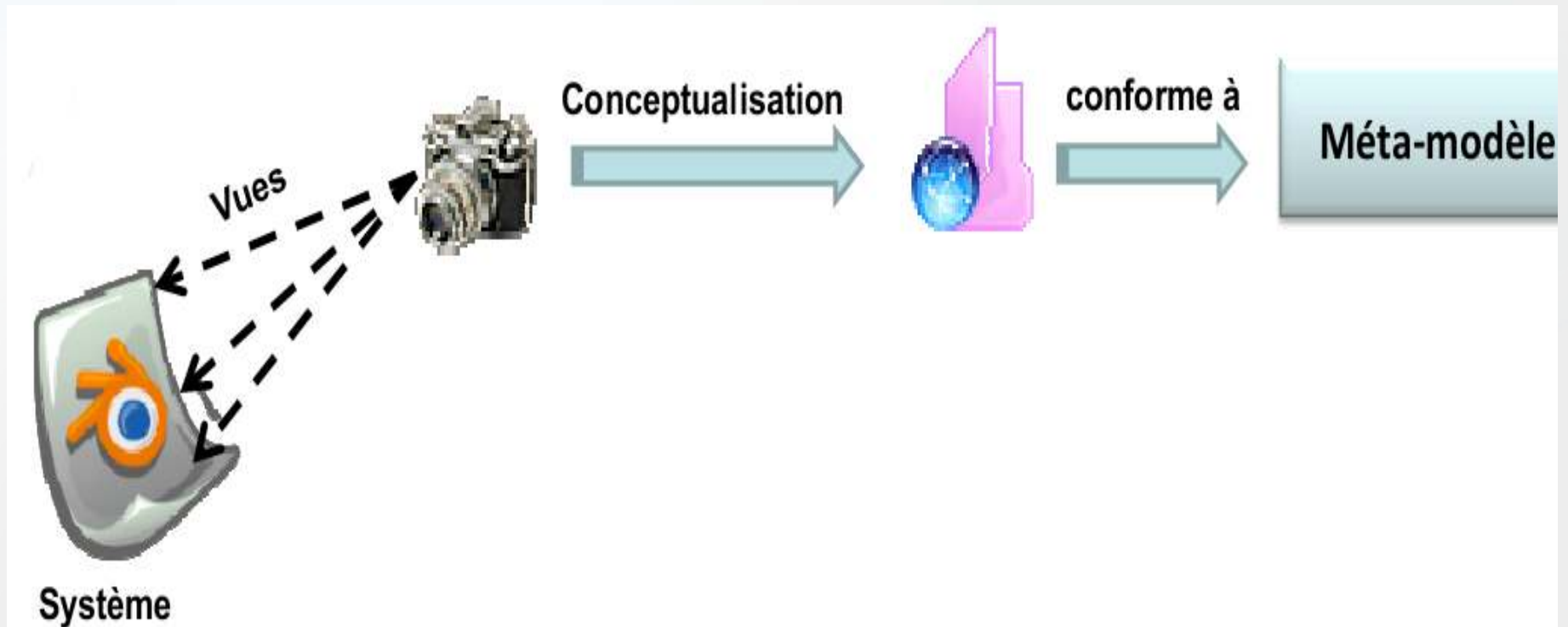
Problème

- Les spécifications textuelles sont ambiguës.
- Trop de texte complique la gestion de l'information.
- Peut-on faire appel aux modèles pour simplifier la réalité?
- Utilisez des figures et des diagrammes pour illustrer l'intention.

Modélisation: modèle

- Un modèle est une abstraction de la réalité;
- Chaque modèle est conforme à un méta-modèle;
- Les modèles servent à exposer et à partager une vue conceptuelle qui réside dans le cerveau humain, pour être jugée et reformulée.

Modélisation: vues



- Les vues représentent les différentes facettes du système réel.
- Les vues sont conceptualisées dans un ou plusieurs modèles.

Modélisation: vues (UML)

- **Vue utilisateur:**

- Cas d'utilisation

- **Vue Logique:**

- Diagramme de classes
- Diagramme d'objets
- Diagramme de paquetage

- **Vue d'implantation:**

- Diagramme de composants

- **Vue comportementale:**

- Diagramme d'états
- Diagramme d'activités
- Diagramme de séquence
- Diagramme de collaboration

- **Vue de déploiement:**

- Diagramme de déploiement

Avantages des cas d'utilisation

- Capturer le comportement désiré du système;
- Spécifier ce que le système **fait (fonctions)**, mais pas comment il le fait;
- Servir d'entente entre les développeurs, les utilisateurs et les experts sur les fonctions disponibles et sur la façon d'interagir avec le système;
- Relativement facile à écrire et à lire par rapport à d'autres méthodes de spécifications;
- Force les développeurs à penser dans la perspective des utilisateurs;
- Aide à augmenter l'engagement des utilisateurs (facile à lire);
- Utile pour la conception et les tests.

Démarche pour l'identification des cas d'utilisation (Rappel)

1. Choisir les frontières du système à étudier;
2. Identifier les acteurs primaires;
3. Trouver les buts de chaque acteur;
4. Définir un cas d'utilisation pour chaque but;
5. Identifier les variations possibles et les conditions d'erreur;
6. Définir les relations entre les acteurs;
7. Décomposer les cas d'utilisation complexes en sous cas;
8. Organiser les solutions alternatives en tant qu'extension des cas d'utilisation.

Identification des cas d'utilisation

- **Méthode basée sur les acteurs:**
 1. Identifier les acteurs liés au système;
 2. Identifier, les processus initiés par chacun des acteurs.
- **Méthode basée sur les évènements:**
 1. Identifier les évènements externes auxquels un système doit répondre;
 2. Relier ces évènements aux acteurs et aux cas d'utilisation.

Cas d'utilisation et leurs formats

- Les cas d'utilisation peuvent être spécifiés avec différents niveau de détail:
- **Format de haut niveau:**
 - Abrégé, peu détaillé,
 - Décrit le scénario principal seulement.
- **Format étendu:**
 - Détaillé;
 - Description des scénarios alternatifs.

Cas d'utilisation de haut niveau

- Un exemple de use case pour le guichet automatique:

Cas d'utilisation:	Effectuer un retrait
Acteur(s):	Client
Type:	Primaire
Description:	Un client se présente au guichet automatique, s'identifie, retire un montant d'argent et quitte avec son argent.

Cas d'utilisation étendu

Cas d'utilisation:	Effectuer un retrait
Système:	Guichet automatique
Acteurs:	Client
Parties prenantes et intérêts:	Client: Il désire repartir avec son argent. Gestionnaire de la banque: Il veut que le système enregistre la transaction.
Préconditions:	Le client a un compte bancaire actif.
Garanties en cas de succès:	Le client a reçu son argent et son compte a été débité.
Scénario principal:	1. Un client se présente au guichet automatique 2. Il s'identifie. 3. ...
Scénarios alternatifs:	...

UML: diagramme d'activités

- Permet de représenter graphiquement le comportement ou le déroulement d'un cas d'utilisation;
- Une activité représente une exécution d'un déroulement d'étapes séquentielles;
- Le passage d'une activité vers une autre est matérialisé par une transition;
- Les transitions sont déclenchées par la fin d'une activité et provoquent le début immédiat d'une autre;