

Master mention Informatique
Spécialité ISI

Génie Logiciel et modélisation
M1 / 177EN003

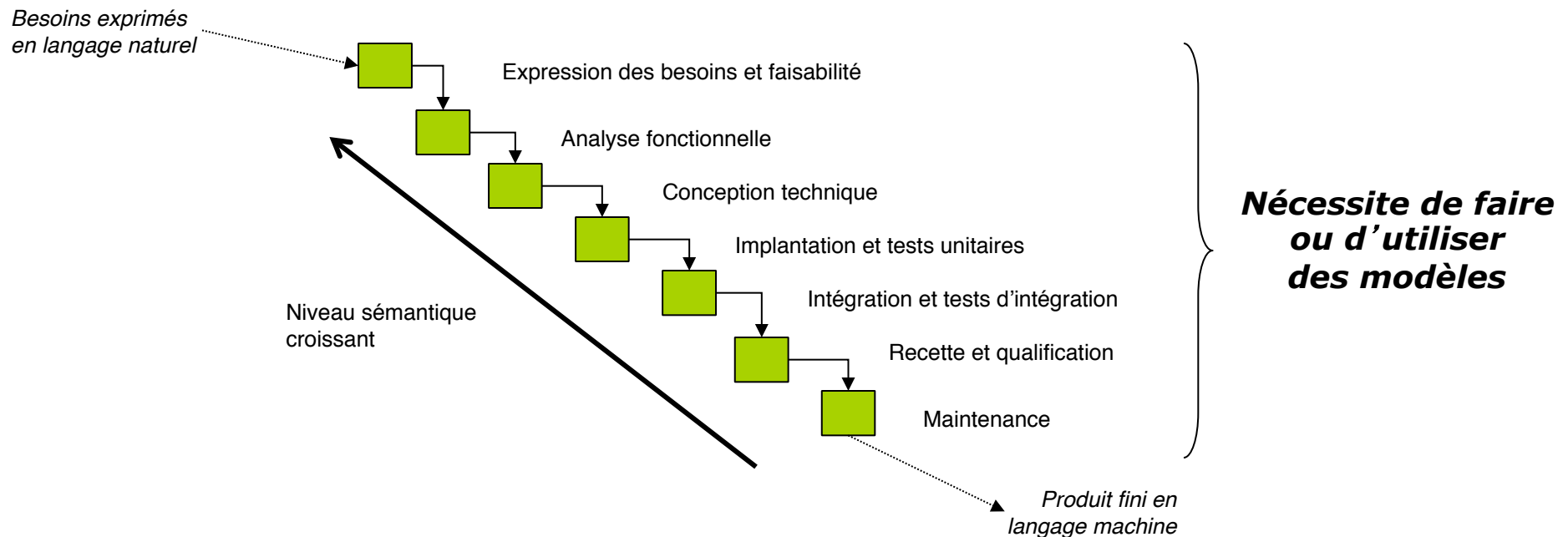
C1 – Introduction à UML
Diagramme de cas d'utilisation

Claudine Piau-Toffolon

- ❑ L'activité de modélisation et le génie logiciel
- ❑ UML : c'est quoi ?
- ❑ Points de vue et diagrammes UML
 - Liste Outils UML: http://www.objectsbydesign.com/tools/modeling_tools_fr.html
- ❑ Diagramme de cas d'utilisation
 - Théorie :
 - ❑ Intérêts des cas d'utilisation
 - ❑ Les diagrammes de cas d'utilisation
 - ❑ Description textuelle des cas d'utilisation
 - ❑ Construction des cas d'utilisation
 - Pratique :
 - ❑ Exemple du GAB

□ Rappels :

- Génie logiciel : « ensemble de méthodes, techniques et outils pour la production et la maintenance de composants logiciels de qualité »
- Cycles de vie :
 - Organisation des activités de développement d'un produit logiciel
 - Cycle en cascade, cycle en V, cycle en spirale, cycle par incrément...
 - En substance, le processus de développement consiste à traduire en **langage machine** des idées exprimées en **langage naturel**.

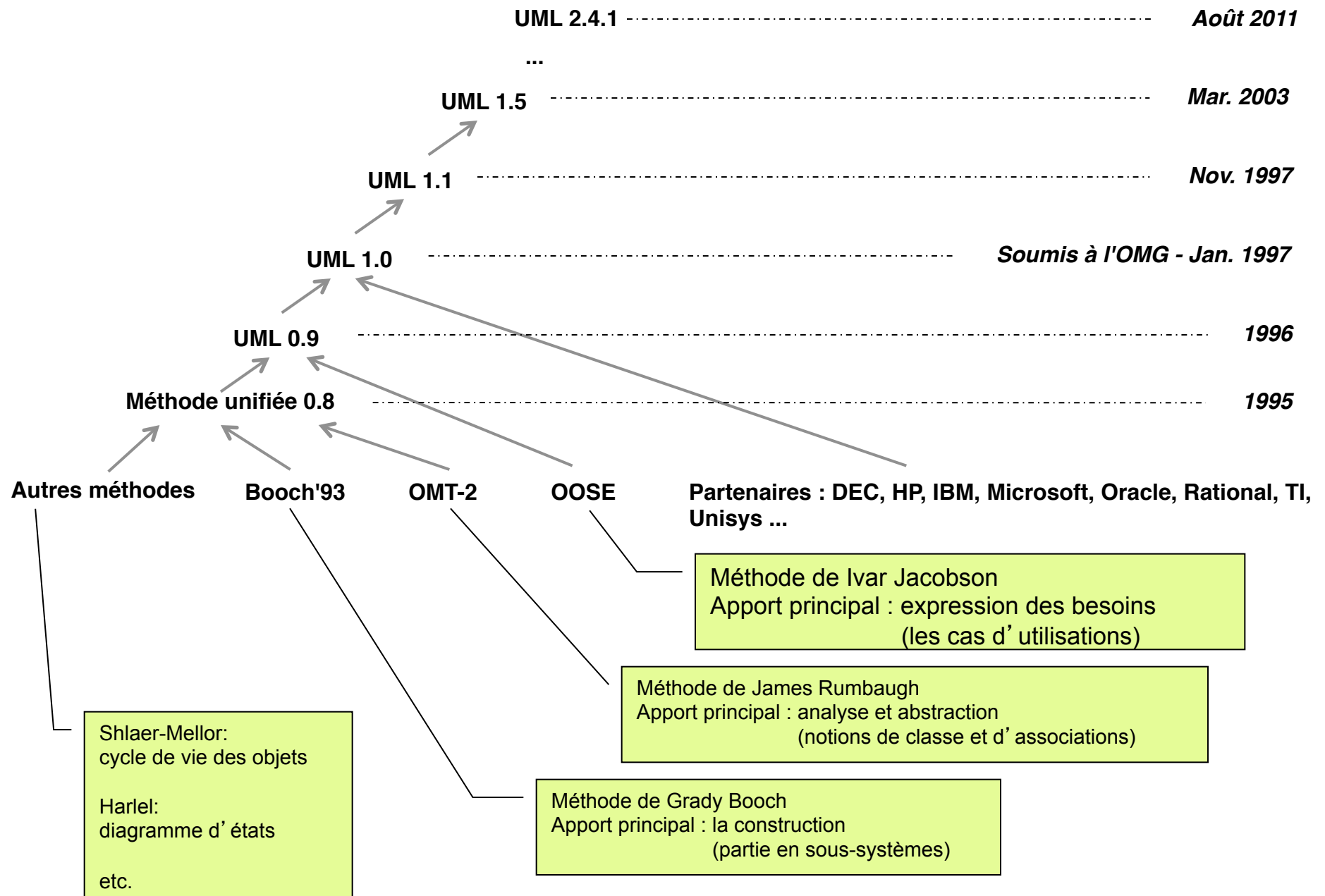


- ❑ L'activité de modélisation selon les activités du processus :
 - En analyse
 - ❑ Comprendre le domaine et le métier de l'utilisateur / client
 - ❑ Représenter sa compréhension du domaine dans des modèles d'analyse
 - En conception
 - ❑ Créer des modèles de conception
 - ❑ Intégrer / reprendre les éléments des modèles d'analyse
 - ❑ Reprendre / réutiliser des modèles existants (*framework, design pattern...*)
 - En développement
 - ❑ Consulter / comprendre les modèles de conception
 - ❑ Compléter les modèles de conception
 - En maintenance
 - ❑ Consulter les modèles
 - ❑ Maintenir à jour / faire évoluer les modèles (analyse et conception)

- ✓ L'activité de modélisation et le génie logiciel
- UML : c'est quoi ?
- Points de vue et diagrammes UML
- Diagramme de cas d'utilisation
 - Théorie :
 - Intérêts des cas d'utilisation
 - Les diagrammes de cas d'utilisation
 - Description textuelle des cas d'utilisation
 - Construction des cas d'utilisation
 - Pratique :
 - Exemple du GAB

- UML (*Unified Modeling Language*)
 - **UML n'est pas une méthode mais un langage !**
 - Langage de modélisation orienté objet :
 - Langage indépendant de toute méthode et de tout langage de programmation
 - Issu de l'unification d'un grand nombre de notations orientés objets du début des années 1990
 - Permet de couvrir toutes les phases du processus logiciel
 - Langage universel :
 - Langage normalisé par l'OMG depuis 1997
 - OMG (*Object Management Group*) : consortium qui regroupe la plupart des industriels de l'informatique (www.omg.org)
 - Supportés par de nombreux ateliers ou d'outils de génie logiciel : Rational Rose Modeler, Modelio, Artisan Studio, Jude, etc.

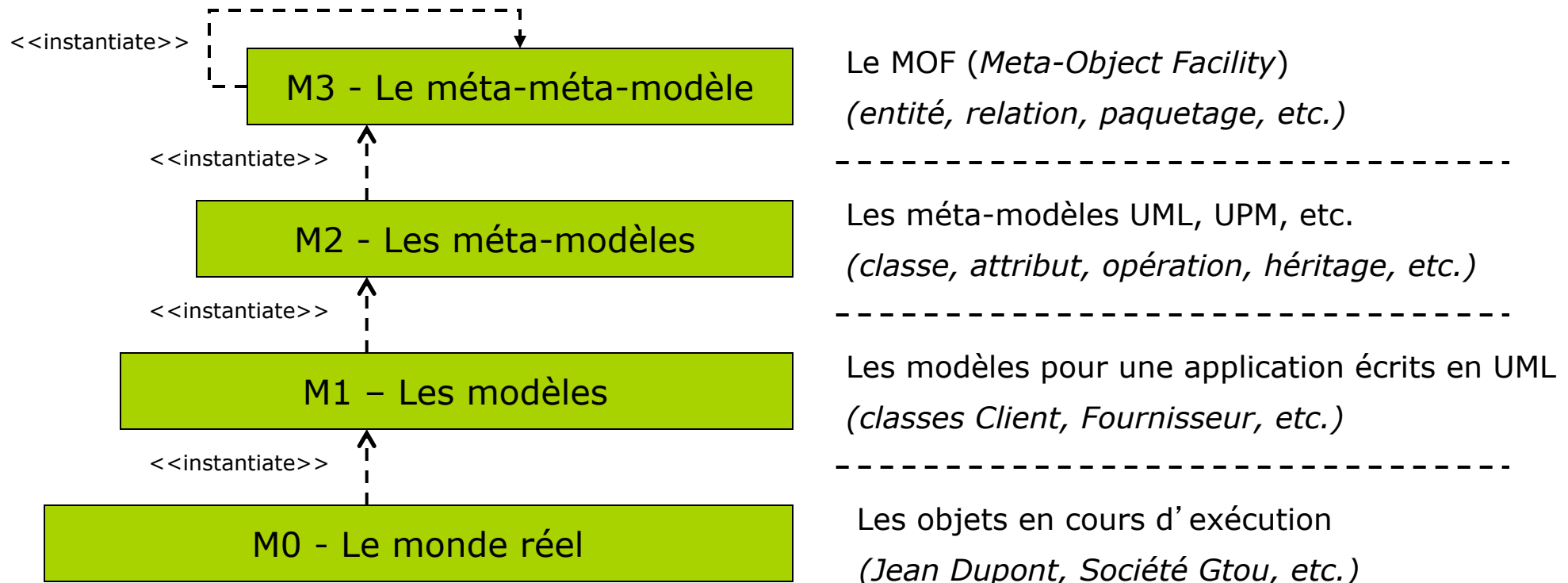
UML : c'est quoi ? (2/4)



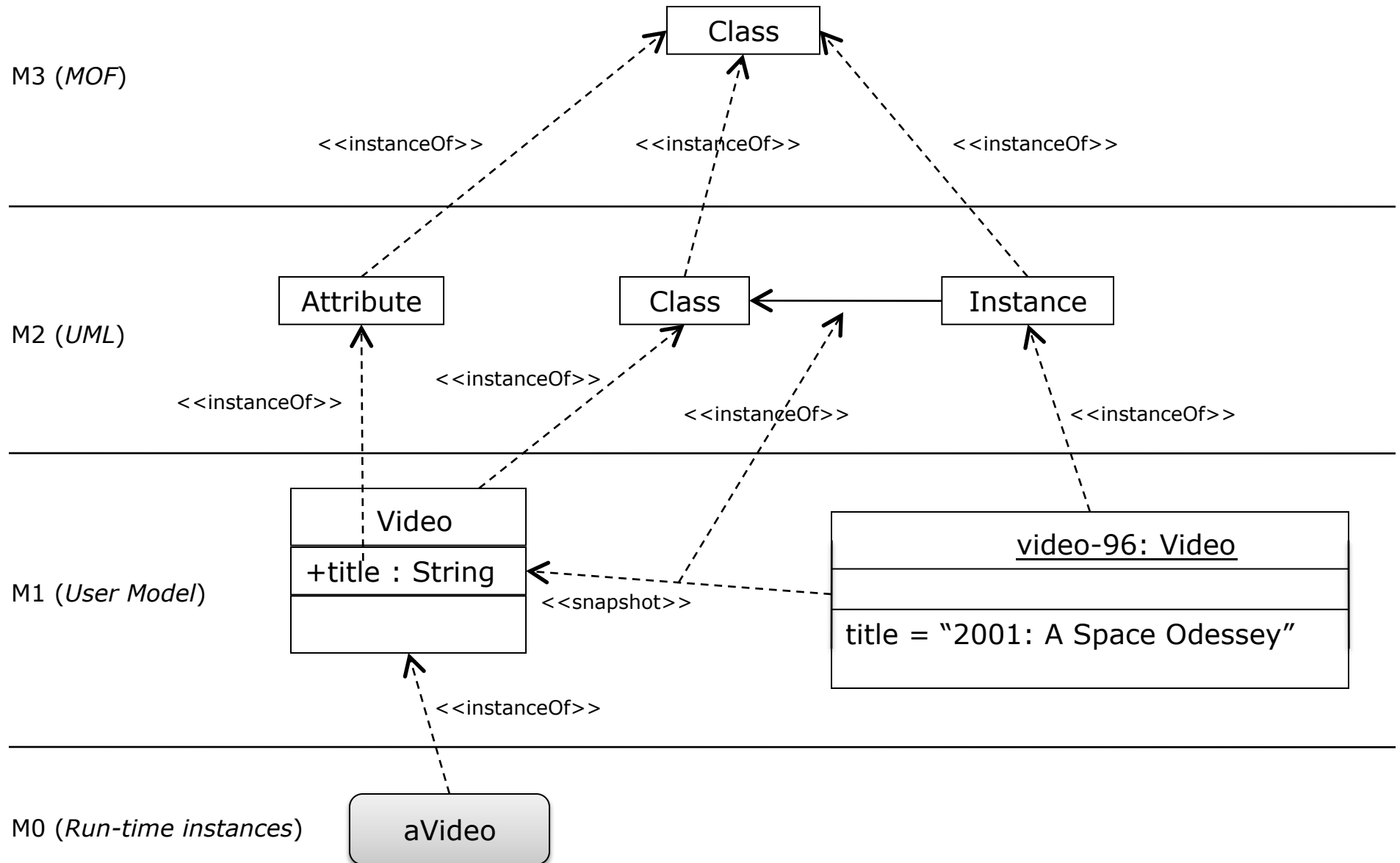
UML : c'est quoi ? (3/4)

□ UML et MOF :

- Le langage UML sert à écrire des modèles qui décrivent des applications informatiques : *c'est un méta-modèle*.
- UML est lui-même une instance du *méta-méta-modèle* MOF.
- MOF est réflexif (il se décrit lui-même).
- Organisation des standards de l'OMG : architecture à 4 niveaux



UML : c'est quoi ? (4/4)



- ✓ L'activité de modélisation et le génie logiciel
- ✓ UML : c'est quoi ?
- Points de vue et diagrammes UML
- Diagramme de cas d'utilisation
 - Théorie :
 - Intérêts des cas d'utilisation
 - Les diagrammes de cas d'utilisation
 - Description textuelle des cas d'utilisation
 - Construction des cas d'utilisation
 - Pratique :
 - Exemple du GAB

- Un produit logiciel peut être considéré sous différents points de vue complémentaires.
- L'ensemble des points de vue permet d'avoir une vision globale du produit modélisé.
- Dans UML 2.x, il y a 9 points de vue répartis en 4 domaines et 11 types de diagrammes (13 dans la version UML version 2.3)
 - Domaine structurel : description statique des éléments
 - Vue statique
 - Modélise les concepts du domaine sans tenir compte du facteur temporel (niveau analyse)
 - Modélise les classes logicielles (niveau conception)
 - **Diagramme de classe**
 - Vue de conception
 - Modélise la structure interne de l'application (en particulier les structures permettant les collaborations entre classes pour fournir les services).
 - **Diagramme de composants** : allocations des classes au sein de composants logiciels et dépendances entre ces composants.
 - *Diagramme de collaboration* : classes provenant de différents composants structurées pour fournir un service.
 - Vue du cas d'utilisation
 - Modélise les transactions entre les acteurs et le système via les fonctionnalités fournies.
 - **Diagramme de cas d'utilisation** : acteurs, unités fonctionnelles et leurs relations.

- **Domaine dynamique : description du comportement du système**
 - **Vue de machine d'états**
 - Modélise la vie d'un élément (objet...). Chaque période de cette vie est modélisée par un état.
 - **Diagramme de machine d'états** : états d'un objet et transitions.
 - **Vue d'activité**
 - Modélise un calcul ou un *workflow* (enchaînement d'activités).
 - **Diagramme d'activités** : flots de contrôle séquentiels et/ou simultanés.
 - **Vue des interactions**
 - Modélise les séquences d'échanges de message entre l'extérieur et le système ou les parties internes du système au sein d'une collaboration.
 - **Diagramme de séquence** : aspect temporel de la collaboration.
 - **Diagramme de communication** : aspect structurel de la collaboration.
- **Domaine physique : description des ressources et du déploiement**
 - **Vue de déploiement**
 - Représente les artefacts physiques à l'exécution (fichiers de données, fichiers binaires, fichiers sources, fichiers de configurations, etc.) sur un noeud (processeur + mémoires).
 - **Diagramme de déploiement** : noeuds + artefacts + communication.
- **Domaine de gestion du modèle : organisation des modèles**
 - **Vue de gestion du modèle**
 - Sert à la fois à l'organisation du modèle (hiérarchisation des modèles)
 - **Diagramme de packages** : packages et leurs dépendances.
 - **Profil** :
 - Extension d'UML avec des stéréotypes, des valeurs étiquetées et des contraintes.

- ✓ L'activité de modélisation et le génie logiciel
- ✓ UML : c'est quoi ?
- ✓ Points de vue et diagrammes UML
- Diagramme de cas d'utilisation
 - Théorie :
 - Intérêts des cas d'utilisation
 - ▣ Les diagrammes de cas d'utilisation
 - ▣ Description textuelle des cas d'utilisation
 - ▣ Construction des cas d'utilisation
 - Pratique :
 - ▣ Exemple du GAB

□ Principes :

- Les cas d'utilisation décrivent le comportement apparent du système selon un point de vue externe au système : on ne se pas comment il fonctionne mais on sait « ce qui rentre et ce qui sort »
- L'ensemble des cas permet de définir les limites du système
- Découverte des cas : lecture **active** du cahier des charges + interviews d'experts du domaine + évaluations avec le client

□ Intérêts :

- Aide à déterminer et à comprendre des besoins :
 - Détermination des catégories d'utilisateurs
 - Centrer l'expression des besoins par rapport aux catégories d'utilisateurs
- Formalisme simple et donc « accessible » aux non-informaticiens
- Aide les utilisateurs à structurer et préciser leurs souhaits
 - Oblige à définir les interactions, les informations échangées et les connaissances du métier impliquées
- Aide les informaticiens à se focaliser sur les « vrais » besoins

- ✓ L'activité de modélisation et le génie logiciel
- ✓ UML : c'est quoi ?
- ✓ Points de vue et diagrammes UML
- Diagramme de cas d'utilisation
 - Théorie :
 - ✓ Intérêts des cas d'utilisation
 - Les diagrammes de cas d'utilisation
 - ▣ Description textuelle des cas d'utilisation
 - ▣ Construction des cas d'utilisation
 - Pratique :
 - ▣ Exemple du GAB

Diagramme de cas d'utilisation - Les acteurs (1/2)

■ Les acteurs :

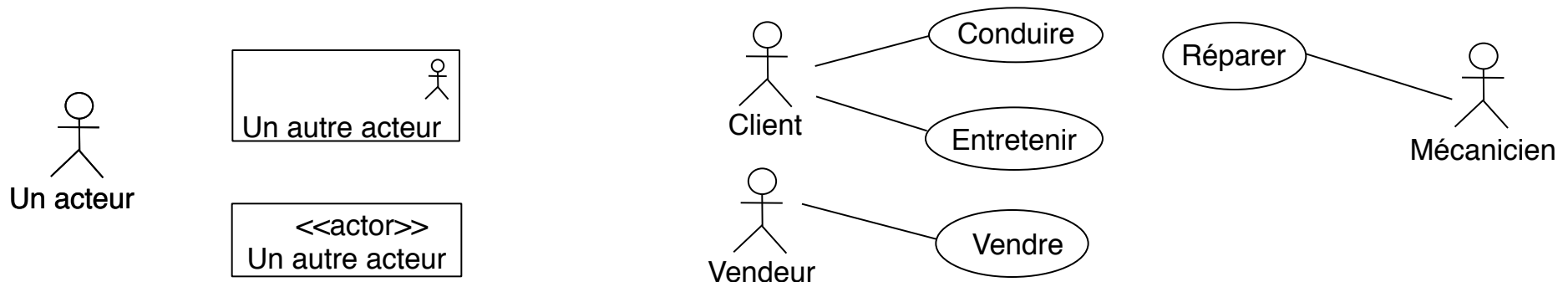
■ Principe :

- Un acteur (*une personne* ou *une chose*) joue un rôle et interagit avec le système.
- Une personne ou une chose peut jouer les rôles de plusieurs acteurs et plusieurs personnes ou choses peuvent jouer le même rôle.
- Le nom de l'acteur doit donner une indication sur son rôle.

■ Cas particulier des acteurs non humain (les « choses ») :

- Les autres systèmes : les systèmes extérieurs qui interagissent avec le système modélisé sont représentés par des acteurs.
- Le matériel (dispositifs matériels périphériques) *faisant partie du domaine de l'application* n'est pas modélisé comme acteur.

■ Représentation : l'homme bâton (*stickman*)

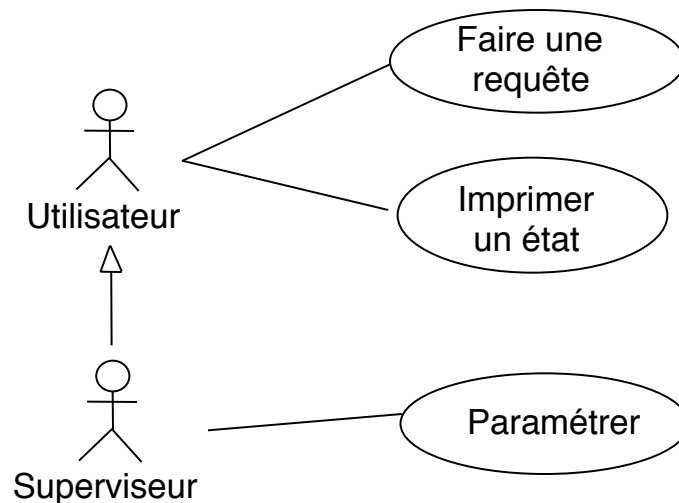


■ Les acteurs (suite) :

- Description : chaque acteur est décrit textuellement (quelques lignes) pour indiquer les attentes métiers vis-à-vis du système.

■ La seule relation possible entre acteurs :

- Relation de généralisation : l'acteur enfant peut communiquer avec le système comme le fait l'acteur parent.
- Représentation :



L'Utilisateur peut :

- Faire une requête
- Imprimer une requête

Le Superviseur peut :

- Faire le paramétrage
- Imprimer un état
- Faire une requête

Diagramme de cas d'utilisation - Les cas d'utilisation (1/2)

■ Les cas d'utilisation :

■ Principe :

- Un cas d'utilisation modélise un dialogue entre les acteurs et le système menant à la réalisation d'un service métier par le système pour les acteurs.
- L'échange de messages entre le système et ses acteurs est appelé un scénario d'interaction (description textuelle, *diagrammes de séquences*, *diagrammes d'activités*).
- Un cas d'utilisation représente l'utilisation d'une ou plusieurs fonctionnalités du système et les interactions liées à cette (ces) fonctionnalité(s).

- Les cas d'utilisation servent de fil conducteur tout au long du projet.
- La relation entre un acteur et un cas d'utilisation est une association.
- Représentation :

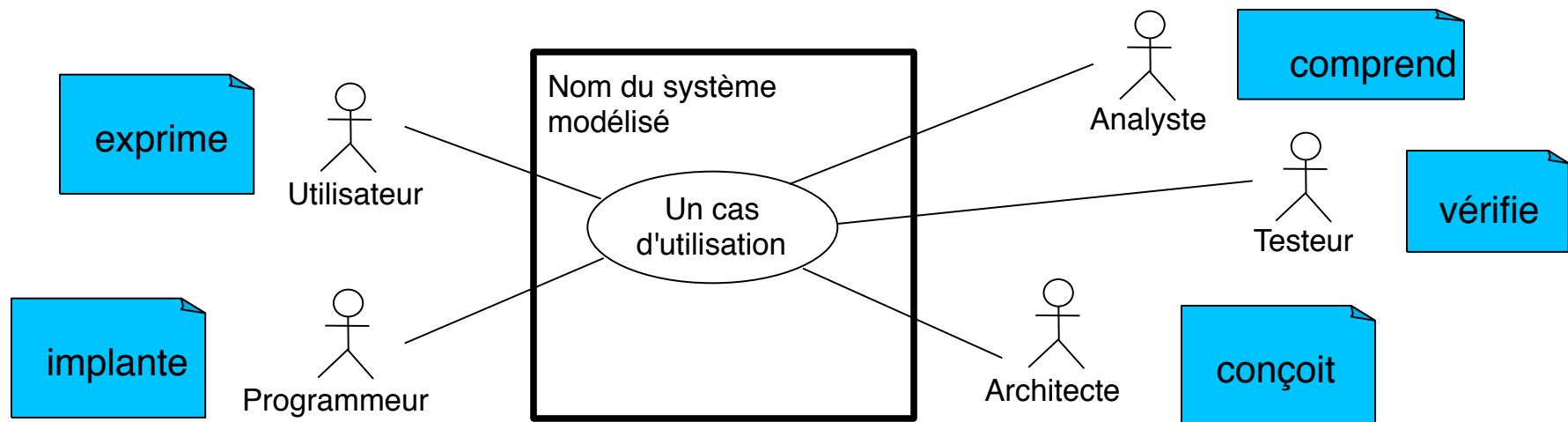
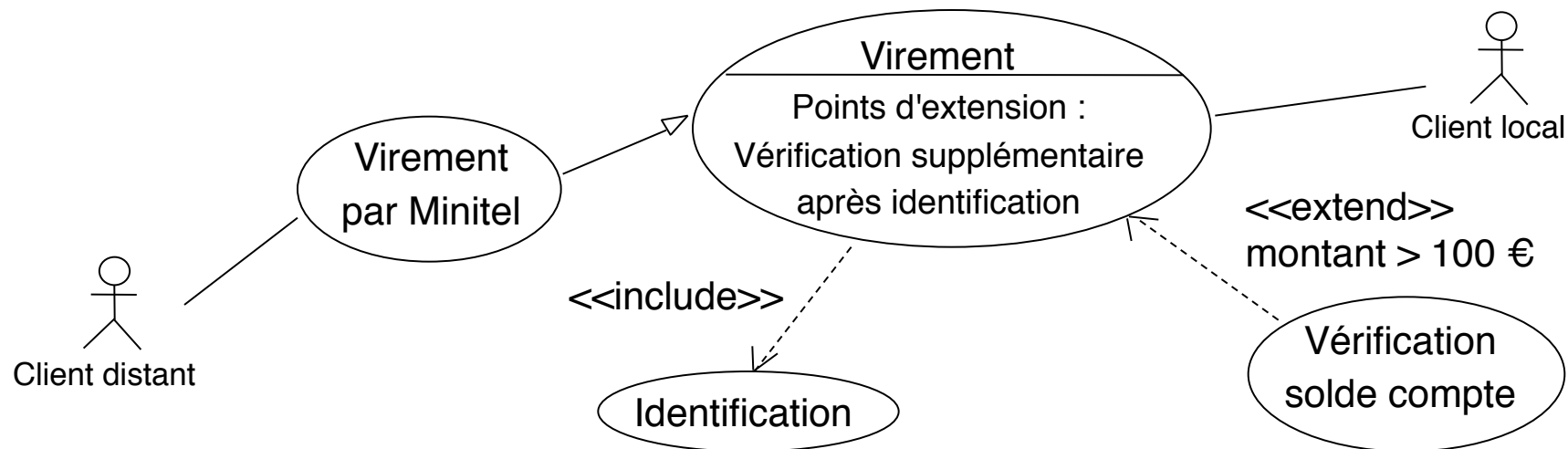


Diagramme de cas d'utilisation - Les cas d'utilisation (2/2)

■ Les relations entre cas d'utilisation :

- **La relation de généralisation** : le cas d'utilisation enfant est une spécialisation du cas d'utilisation parent.
- **La relation d'inclusion** : le cas d'utilisation source comprend également le comportement décrit par le cas d'utilisation destination. Cette relation permet de définir des comportements obligatoires et partageables en modélisant les cas d'utilisation communs.
- **La relation d'extension** : le cas d'utilisation source ajoute son comportement au cas d'utilisation destination. L'extension peut être soumise à une condition. Cette relation permet de définir des cas d'utilisation exceptionnels ou optionnels.
- **Les relations n'expriment pas de relations temporelles mais uniquement des relations fonctionnelles.**



- ✓ L'activité de modélisation et le génie logiciel
- ✓ UML : c'est quoi ?
- ✓ Points de vue et diagrammes UML
- Diagramme de cas d'utilisation
 - Théorie :
 - ✓ Intérêts des cas d'utilisation
 - ✓ Les diagrammes de cas d'utilisation
 - Description textuelle des cas d'utilisation
 - ▣ Construction des cas d'utilisation
 - Pratique :
 - ▣ Exemple du GAB

- Description textuelle des cas d'utilisation :
 - Nécessaire pour une communication précise avec les clients et les membres de l'équipe.
 - Sert à recenser les interactions et les contraintes qu'elles induisent en terme d'IHM et de contraintes non-fonctionnelles.
- Fiche de description :
 - Sommaire d'identification (obligatoire) : fiche d'identité du cas d'utilisation (titre, résumé, dates, version, responsable, acteurs)
 - Description des *enchaînements* (obligatoire) : décrit les enchaînements nominaux (début et fin clairement identifiés), les enchaînements alternatifs, les exceptions, les pré-conditions et les post-conditions.
 - Besoins en IHM (optionnel) : contraintes d'interface homme-machine (ce qu'il faut montrer à l'utilisateur et ce qu'il peut faire).
 - Contraintes non-fonctionnelles (optionnel) : indiquer les contraintes de fiabilité, de confidentialité, de performances, etc.

- ✓ L'activité de modélisation et le génie logiciel
- ✓ UML : c'est quoi ?
- ✓ Points de vue et diagrammes UML
- Diagramme de cas d'utilisation
 - Théorie :
 - ✓ Intérêts des cas d'utilisation
 - ✓ Les diagrammes de cas d'utilisation
 - ✓ Description textuelle des cas d'utilisation
 - Construction des cas d'utilisation
 - Pratique :
 - Exemple du GAB

Diagramme de cas d'utilisation - Construction

- 1) Définir un guide de style pour la description textuelle
- 2) Identification grossière des cas d'utilisation
 - 1) Identification des acteurs : personnes et choses extérieures au système et qui interagissent avec lui en échangeant des informations (en entrée et en sortie) :
 - 1) Qui est intéressé par un certain besoin ?
 - 2) Où dans l'organisation le système est-il utilisé ?
 - 3) Qui bénéficiera de l'utilisation du système ?
 - 4) Qui fournira l'information, qui l'utilisera et la maintiendra ?
 - 5) Qui va supporter et maintenir le système ?
 - 2) Identification des cas d'utilisation : observation et précision acteur par acteur des séquences d'interactions :
 - 1) Quelles sont les tâches de chaque acteur ?
 - 2) Est-ce qu'un acteur crée, enregistre, modifie, enlève ou consulte une information dans le système ?
 - 3) Quel cas d'utilisation sera responsable de ces tâches ?
 - 4) Quels cas d'utilisation supportent ou maintiennent le système ?
- 3) Chaque cas d'utilisation est détaillée en scénarios par un groupe d'analystes grâce à des descriptions textuelles et des diagrammes d'interactions
- 4) S'assurer de la cohérence globale (réunions de mise en commun régulières)

- ✓ L'activité de modélisation et le génie logiciel
- ✓ UML : c'est quoi ?
- ✓ Points de vue et diagrammes UML
- **Diagramme de cas d'utilisation**
 - ✓ **Théorie :**
 - ✓ Intérêts des cas d'utilisation
 - ✓ Les diagrammes de cas d'utilisation
 - ✓ Description textuelle des cas d'utilisation
 - ✓ Construction des cas d'utilisation
 - **Pratique :**
 - Exemple du GAB

- ❑ Cette étude de cas concerne un système simplifié de Guichet Automatique Bancaire (GAB).
- ❑ On cherche à *modéliser le GAB dans son ensemble* (logiciel et matériel).
- ❑ Les caractéristiques du GAB sont :
 - 1 Distribution d'argent à tout porteur de carte de crédit (carte Visa ou carte de la banque), via un lecteur et un distributeur de billet.
 - 2 Consultation de solde de compte, dépôt en numéraire et dépôt de chèques pour les clients de la banque porteurs d'une carte de crédit de la banque.
 - 3 Transactions sécurisées
 - 4 Recharges du distributeur

- Première étape : identification des acteurs du GAB
 - Principe : analyse de l'énoncé + interviews de l'expert métier et du client
 - Phrase 1 :
 - Le porteur de carte de crédit
 - Le lecteur de carte et le distributeur de billets font partie du GAB : ce ne sont pas des acteurs dans ce cas (*mais si on veut modéliser le logiciel uniquement alors ils deviennent des acteurs*)
 - Idem pour la carte de crédit : l'interaction de la carte avec le système n'est pas modéliser à ce niveau puisqu'elle est représentée par l'interaction de l'utilisateur avec le GAB
 - Phrase 2 :
 - Le porteur de carte de la banque : le client de la banque
 - Phrase 3 :
 - Qui sécurise les transactions ? Une interview de l'expert métier donne :
 - le système d'autorisation Visa pour les retraits effectués avec une carte Visa
 - le système d'information de la banque pour les clients de la banque
 - Phrase 4 :
 - Opérateur de maintenance

- ❑ Première étape (suite) : identification des acteurs du GAB
 - Diagramme de contexte statique :
 - ❑ Diagramme de classe ! (cf. cours suivant)
 - ❑ Le système modélisé est représenté comme une « boîte noire » et tous les acteurs sont représentés autour

- ❑ Seconde étape : identification des cas d'utilisation
 - Un cas d'utilisation modélise un service rendu par le système
 - Principe : étudier les interactions de chacun des acteurs avec le système afin de déterminer les différentes intentions *métiers*

- ❑ Troisième étape : description textuelle des cas d'utilisation
 - Exemple du cas d'utilisation « Retirer de l'argent avec une carte Visa »

Sommaire d'identification

Titre : Retirer de l'argent avec une carte Visa

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

Acteurs : Porteur de CB Visa (principal), SA Visa (secondaire)

Date de création : 02/03/00

Date de mise à jour : 09/11/00

Version : 2.2

Responsable : Pascal Roques

Exemple du GAB (5/9)

Description des enchaînements

Préconditions :

La caisse du GAB est alimentée

Aucune carte bancaire ne se trouve dans le lecteur

Scénario nominal :

- 1 Le porteur de CB Visa introduit sa carte Visa dans le lecteur de cartes du GAB
- 2 Le GAB vérifie que la carte introduite est bien une carte Visa
- 3 Le GAB demande au porteur de CB Visa de saisir son code d'identification
- 4 Le porteur de CB Visa saisit son code d'identification
- 5 Le GAB compare le code d'identification avec celui de la puce de la carte
- 6 Le GAB demande une autorisation au système d'autorisation Visa
- 7 Le système d'autorisation Visa donne son accord et indique le solde hebdomadaire
- 8 Le GAB demande au porteur de CB Visa de saisir le montant désiré du retrait
- 9 Le porteur de CB Visa saisit le montant désiré du retrait
- 10 Le GAB contrôle le montant demandé par rapport au solde hebdomadaire
- 11 Le GAB demande au porteur de CB Visa s'il veut un ticket
- 12 Le porteur de CB Visa demande un ticket
- 13 Le GAB rend sa carte au porteur de CB Visa
- 14 Le porteur de CB Visa reprend sa carte
- 15 Le GAB délivre les billets et un ticket
- 16 Le porteur de CB Visa prend les billets et le ticket

Exemple du GAB (6/9)

Enchaînements alternatifs :

A1 : code d'identification provisoirement erroné

L' enchaînement A1 démarre au point 5 du scénario nominal

6 Le GAB indique au client que le code est erroné, pour la première ou la seconde fois

7 Le GAB enregistre l' échec sur la carte

Le scénario nominal reprend au point 3

A2 : montant demandé supérieur au solde hebdomadaire

L' enchaînement A2 démarre au point 10 du scénario nominal

...

A3 : ticket refusé

...

Enchaînements d'exception :

E1 : carte non-valide

L'enchaînement E1 démarre au point 2 du scénario nominal

3 Le GAB indique au porteur que la carte n'est pas valide (illisible, périmée, etc.), la confisque ; le cas d'utilisation est terminé

E2 : code d'identification définitivement erroné

L'enchaînement E2 démarre au point 5 du scénario nominal

6 Le GAB indique au client que le code est erroné pour la troisième fois

7 Le GAB confisque la carte

8 Le système d'autorisation Visa est informé ; le cas d'utilisation est terminé

E3 : retrait non autorisé

L'enchaînement E3 démarre au point 6 du scénario nominal

...

E4 : carte non reprise

...

E5 : billets non pris

...

Postconditions :

La caisse du GAB contient moins de billets qu'au début du cas d'utilisation (le nombre de billets manquants est fonction du montant du retrait)

Besoins d'IHM

Les dispositifs d'entrée/sortie disponible au porteur de CB Visa

- Un lecteur de carte à puce
- Un clavier numérique + touches de validation, de correction et d'annulation
- Un écran pour l'affichage des messages du GAB
- Des touches autour de l'écran pour sélectionner un montant
- Un distributeur de billets
- Un distributeur de tickets

Contraintes non-fonctionnelles

Temps de réponse : L'interface du GAB doit réagir en moins de 2 secondes

Disponibilité :

- Le GAB est accessible 7 jours sur 7 et 24 heures sur 24
- La maintenance ne doit pas durer plus d'une heure par semaine
- L'absence de papier d'impression des tickets ne doit pas arrêter les retraits

Intégrité : Robustesse des interfaces pour empêcher tout vandalisme

Confidentialité : La comparaison du code de la puce et celui saisi au clavier doit être fiable à 10^{-6}

- ▣ Quatrième étape : Le diagramme de cas d'utilisation complet