CHAPITRE 5

Construction d'un modèle conceptuel

Sommaire

- Identification des concepts,
- Ajout des associations,
- Ajout des attributs,
- Enregistrement des termes dans un glossaire.

POLYTECHNIQUE MONTRÉAL
LE GÉNIE EN PREMIÈRE CLASSE

Modéliser la vue logique: diagrammes de classes et d'objets

Vue logique

Diagrammes de classes
Diagrammes d'objets

Vue implémentation

Diagrammes de composantes

Vue utilisateur

Diagrammes de cas

Vue comportement

Diagrammes d'état
Diagrammes d'activités
Diagrammes de séquence
Diagrammes de collaboration

Vue déploiement

Diagrammes de déploiement

Identification des concepts

L'étape centrale d'une approche d'analyse orientée objet est la décomposition du problème en concepts individuels ou objets.

Le modèle conceptuel est une représentation des éléments du monde réel, pas de composants logiciels.

Le modèle conceptuel est illustré en UML en utilisant un ensemble de diagrammes structurels sans opérations.

Identification des concepts

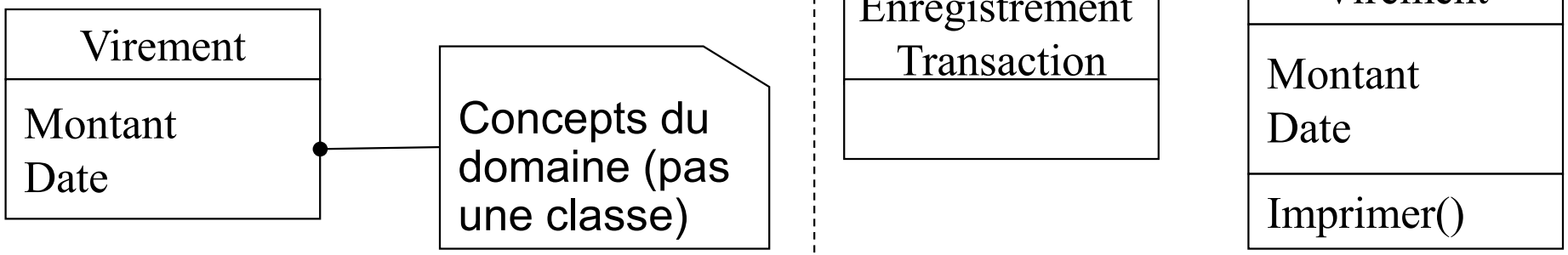
En plus de fournir une décomposition du problème en éléments compréhensibles, la construction du modèle conceptuel permet de clarifier la terminologie et le vocabulaire du domaine.

POLYTECHNIQUE MONTRÉAL
LE GÉNIE EN PREMIÈRE CLASSE

Identification des concepts

Le modèle conceptuel est une description d'éléments réels dans le domaine du problème, pas d'éléments logiciels:

➔ Le modèle conceptuel ne présente pas d'artefacts logiciels comme des fenêtres ou une base de données.



OUI !

NON!

Identification des concepts: un processus de subdivision

Pour aborder l'analyse de systèmes logiciels complexes, une stratégie courante consiste à subdiviser l'espace du problème pour mieux appréhender la complexité:

- ➔ Approche de « diviser pour régner »,
- ➔ Une distinction fondamentale entre l'approche d'analyse orientée objet et l'approche structurée est le fait que l'approche orientée objet utilise les concepts (objets) comme base de division alors que l'approche structurée utilise les fonctions.

Identification des concepts

Concepts tirés du domaine des guichets automatiques (liste partielle):

- CompteÉpargne
- CompteChèque
- Retrait
- Virement
- Dépot
- Client

Stratégies d'identification des concepts

Règle: il vaut mieux sur spécifier le modèle conceptuel avec beaucoup de concepts ayant une granularité très fine, que de le sous spécifier.

Il est fréquent de rater des concepts lors de l'analyse et de les découvrir plus tard, par exemple durant la conception.

Il ne faut pas exclure des concepts simplement parce qu'il n'y a pas d'information à sauvegarder qui leur est associée. Certains concepts peuvent très bien n'avoir aucun attribut et uniquement un rôle comportemental dans le domaine.

À lire hors-classe

Stratégies d'identification des concepts: Les listes de catégories

Catégorie de concept	Exemple
Objets physiques ou tangibles	CarteDeGuichet
Spécification ou description d'objet	SpécificationDeCompte
Endroit	Banque
Transaction	Virement, dépot
Items de transaction	Ligne d'état de compte
Rôles de personnes	Client, caissier
Contenant d'autres objets	Enveloppe
Objets dans un contenant	Chèque, devise
Autres systèmes informatiques ou électro-mécaniques	Tiroir à enveloppes, distributeur de devises

À lire hors-classe

Stratégies d'identification des concepts: Les listes de catégories

Catégorie de concept	Exemple
Concepts abstraits	Insécurité
Organisations	DépartementDeCrédit
Événements	Alarme, vol, insertionDEnveloppe
Processus (pas nécessairement des concepts, mais peut-être)	EffectuerUneTransaction
Règles et politiques	PolitiqueDeCrédit
Catalogues	CatalogueDesFondsCommuns
Document financiers, légaux	CertificatDHypothèque
Services et outils financiers	LigneDeCrédit, actions
Manuels, livres	ManuelDeRéparation

Stratégies d'identification des concepts: les noms et les phrases nominatives

On peut identifier un grand nombre de concepts en parcourant les documents de requis et de descriptions textuelles du domaine et en considérant les noms comme des concepts candidats ou des attributs:

➔ Il faut procéder avec prudence: une conversion mécanique des noms en concepts n'est pas possible, les langages naturels sont très ambigus.

Identification des concepts: règles

Pour construire un modèle conceptuel:

1. Faire la liste des concepts candidats en utilisant la liste de catégories et l'analyse des phrases nominales tirés des requis,
2. Les rassembler dans un diagramme conceptuel,
3. Ajouter les associations nécessaires pour conserver les relations qui méritent d'être mémorisées,
4. Ajouter les attributs nécessaires pour remplir les requis informationnels.

Nommer et modéliser: le cartographe

Construire un modèle conceptuel ressemble à un processus de cartographie, on doit :

- Utiliser les noms existants,
- Exclure les détails inutiles,
- Ne pas ajouter d'éléments inexistants.

Identification des concepts: une erreur courante

L'erreur la plus courante lors de la construction d'un modèle conceptuel est probablement le fait de représenter sous forme d'attribut quelque chose qui aurait dû être représenté comme un concept:

➔ Si on ne pense pas à X en terme de texte ou de nombre dans le monde réel, X est probablement un concept et non un attribut.

Dépot
destination

ou… ?

Dépot

Compte
numéro

➔ En cas de doute, en faire un concept séparé.

Séparer les concepts similaires

Dans bien des cas, le vocabulaire du domaine utilise des synonymes qu'ils faut réconcilier:

- ➔ Par définition, un modèle conceptuel n'est pas « bon » ou « mauvais », mais bien plus ou moins utile en terme de capacité à communiquer de l'information.
- ➔ Au niveau des concepts, on choisira de préférence les termes plus usuels, ou ceux qui représentent le mieux une abstraction dans son ensemble.

Concepts de spécification ou de description

Supposons que:

- Un Compte représente un type de compte dans une banque, associé à un client spécifique, possédant un numéro,
- Un Compte a une description contenant, entre autres, un taux d'intérêt, qui n'est pas mémorisé ailleurs,
- Tout le monde dans la banque est amnésique,
- Chaque fois qu'un Compte est fermé, un item logiciel correspondant à ce compte est effacé de la mémoire de l'ordinateur (une instance de la classe Compte est détruite).

Concepts de spécification ou de description

Que se passe-t-il, dans le cas suivant:

- Les Comptes « Petits Trésors » pour les 0-3 ans sont automatiquement convertis en Comptes « Épargne Jeunesse » le jour du 3ième anniversaire du propriétaire du Compte,
- À cause de la dénatalité, il n'y a plus de client de moins de 3 ans à la banque,
- Quelqu'un demande: « Combien d'intérêt rapporte le Compte « Petits Trésors » ? »

Concepts de spécification ou de description

Plus personne ne peut répondre, le dernier Compte « Petits Trésors » a été effacé automatiquement et le taux d'intérêt était attaché aux Comptes.

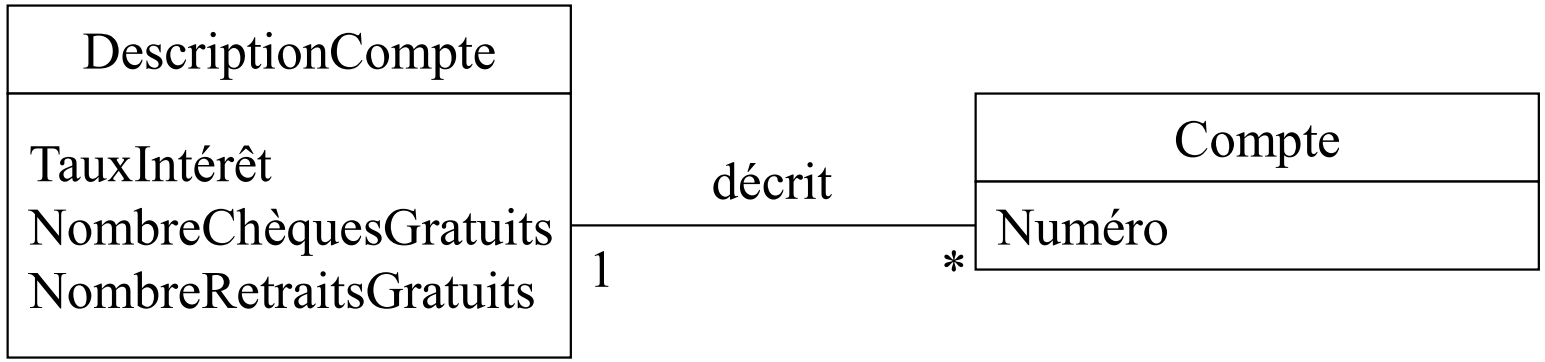
- ➔Il faut introduire un concept de spécification ou de description de compte pour conserver l'information à propos des items,
- ➔La DescriptionDuCompte ne représente pas un compte comme tel, mais bien sa description,
- ➔Les objets de spécification ou de description sont fortement liés aux objets qu'ils décrivent.

Concepts de spécification ou de description

Compte
Numéro TauxIntérêt NombreChèquesGratuits NombreRetraitsGratuits

Moins bon

Meilleur



Des concepts, des classes et des types

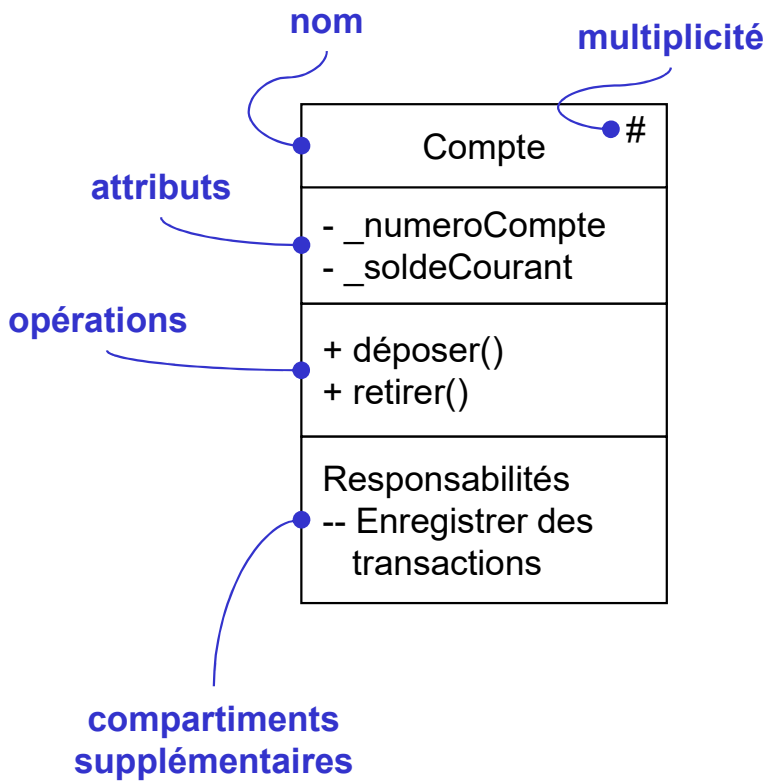
On distingue:

- **Un concept**: la référence à un élément du monde réel, issu du domaine du problème. Le terme concept n'est pas utilisé ni défini en UML,
- **Une classe**: l'implémentation logicielle d'un concept, avec des attributs et des méthodes. Le terme classe est défini en UML,
- **Un type**: similaire à une classe, mais sans les méthodes. Un type, en UML, est indépendant du langage.

Une *classe* est la description d'une collection d'objets qui partagent les mêmes attributs, les mêmes opérations, les mêmes relations et la même sémantique.

Représentation d'une classe

- Compartiment Classe
  - nom
  - stéréotype
  - multiplicité
- Compartiment Attribut
- Compartiment Opération
- Compartiment Supplémentaire
  - Les responsabilités de la classe



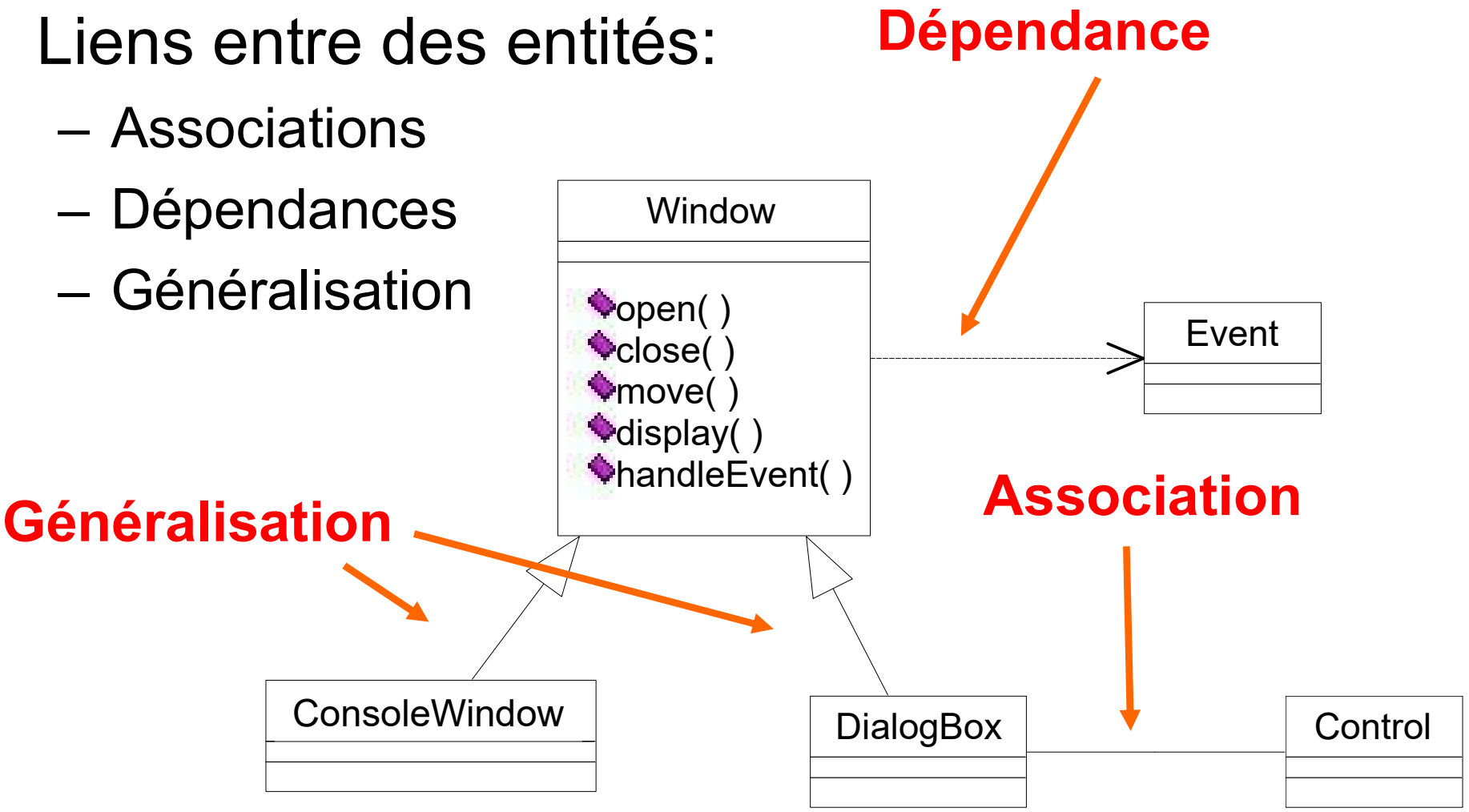
Nommer une classe

- Chaque classe doit avoir un nom distinct des autres classes.
- Constitué d'une chaîne de caractères.
- Peut être sur plusieurs lignes.
- Doit être précis.
- Toutes les premières lettres des mots doivent débuter par une majuscule incluant le premier mot.

ClientBanque	Type	java::awt::Rectangle

Ajout des relations entre les concepts

- Liens entre des entités:
  - Associations
  - Dépendances
  - Généralisation

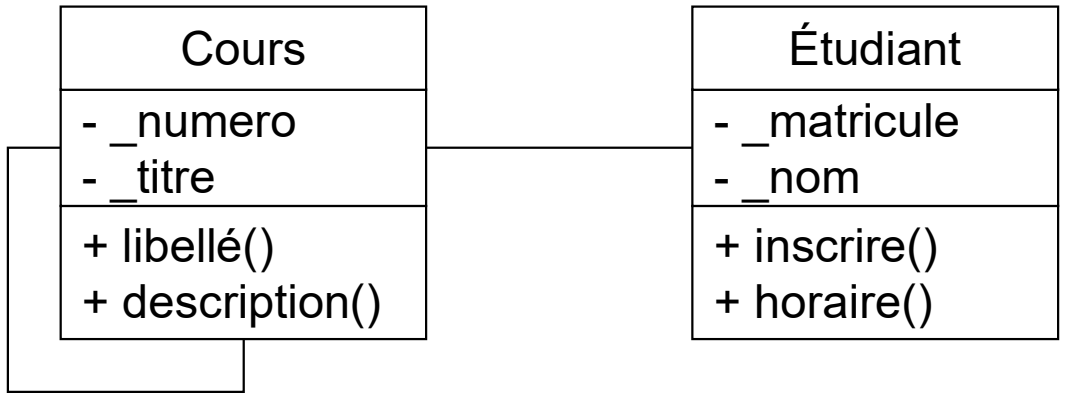


POLYTECHNIQUE MONTRÉAL
LE GÉNIE EN PREMIÈRE CLASSE

Les associations

Une association est une relation entre des concepts qui indique une connexion sémantique intéressante entre eux:

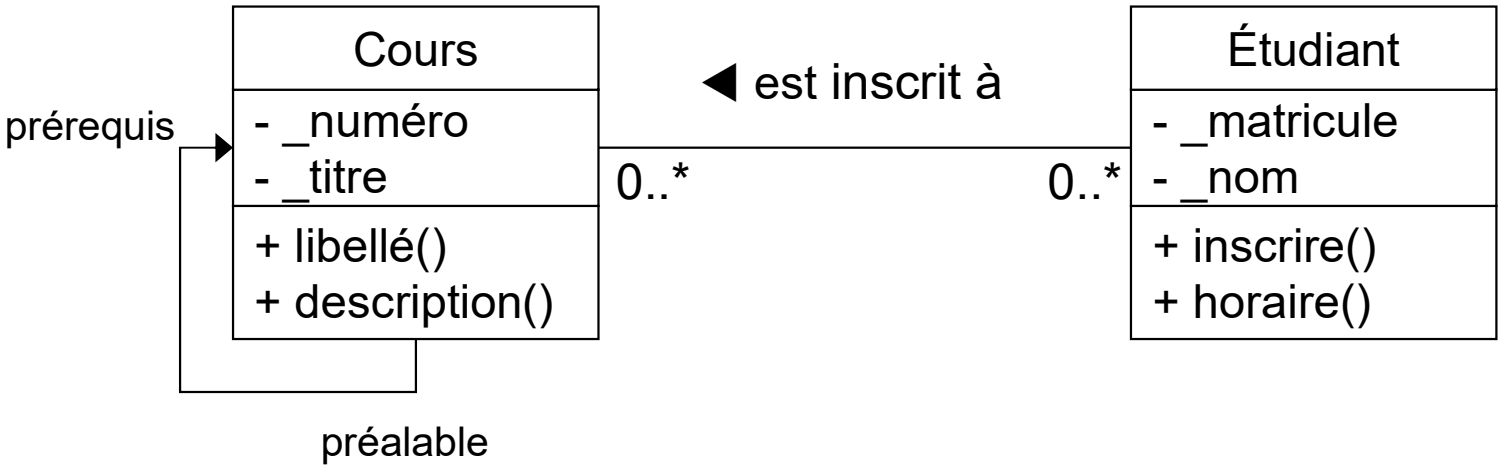
- Une association qui mérite d'être modélisée implique généralement la présence d'une relation qui doit être connue et conservée pour un certain temps,
- L'association permet de voyager d'un objet à l'autre et vice versa,
- Types d'association
  - Unaire
  - Binaire
  - N-aire



Les associations

Une association peut être décorée :

- Nom : décrit la nature de la relation.
- Rôle : nomme la façon dont une classe participe à la relation.
- Multiplicité : indique le nombre d'objets participants.
- Direction : indique le sens de parcours de l'association.



Contraintes sur les associations

- UML définit cinq contraintes pour les associations :
  - `Implicite (« implicit »)`
    spécifie que la relation est conceptuelle (ex: relation héritée)
  - `Ordonné (« ordered »)`
    spécifie que l'ensemble des objets est trié
  - `Changeable (« changeable »)`
    spécifie que l'on peut ajouter, ôter, modifier les liens
  - `ajoutSeulement (« addOnly »)`
    spécifie que l'on peut uniquement ajouter des éléments
  - `Gelé (« frozen »)`
    spécifie que le lien ne peut être détruit ou modifié

Identification des associations

Une façon d'identifier les associations pertinentes consiste à passer au travers de la liste de catégories d'associations courantes suivante:

Catégorie	Exemple
A fait physiquement partie de B	Tiroir-Guichet
A fait logiquement partie de B	LigneTransaction-Relevé
A est physiquement contenu dans B	Chèque-Enveloppe
A est logiquement contenu dans B	FondCommun-CatalogueFonds
A est une description de B	DescriptionDeCompte-Compte
A est un item du rapport B	LigneTransaction-ÉtatDeCompte

Identification des associations

Catégorie	Exemple
A est connu,enregistré,capturé par B	Transaction-Guichet
A est une unité dans l'organisation B	DépartementCrédit-Banque
A utilise ou gère B	Caissier-Guichet
A communique avec B	Client-Caissier
A est lié à la transaction B	Client-Virement
A est une transaction liée à une autre transaction B	Débit-Virement
A est proche de B	Guichet-Guichet
A appartient à B	Guichet-Banque

Des associations prioritaires

Les associations suivantes devraient être incluses de façon prioritaire dans le modèle conceptuel:

- A fait physiquement ou logiquement partie de B,
- A est physiquement ou logiquement contenu dans B
- A est enregistré par B

Détails dans les associations

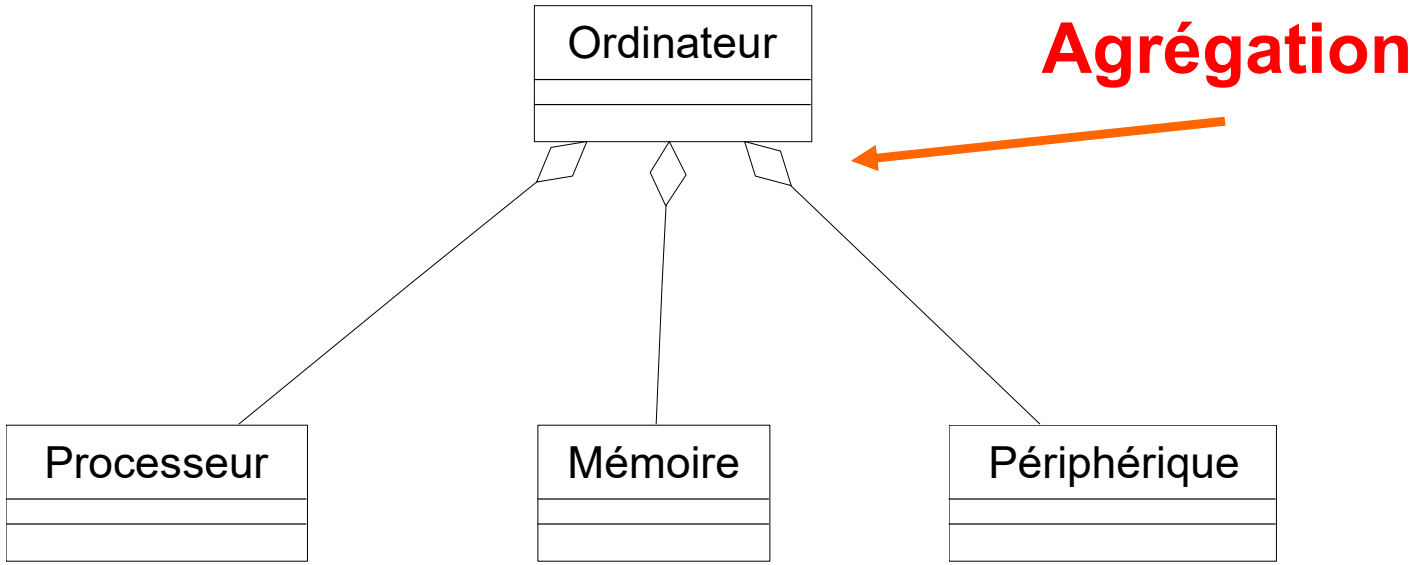
Jusqu'à quel point les associations devraient-elles être détaillées ?

- ➔Il faut éviter de passer trop de temps à essayer de découvrir toutes les associations,
- ➔Il est beaucoup plus important de trouver les concepts que les associations,
- ➔Trop d'associations ont tendance à rendre le modèle conceptuel confus,
- ➔Il faut éviter de montrer les associations redondantes ou qui peuvent être retrouvées.

Association spécialisée : l'agrégation

Dans le diagramme de classes seulement, *pas dans le modèle conceptuel!*

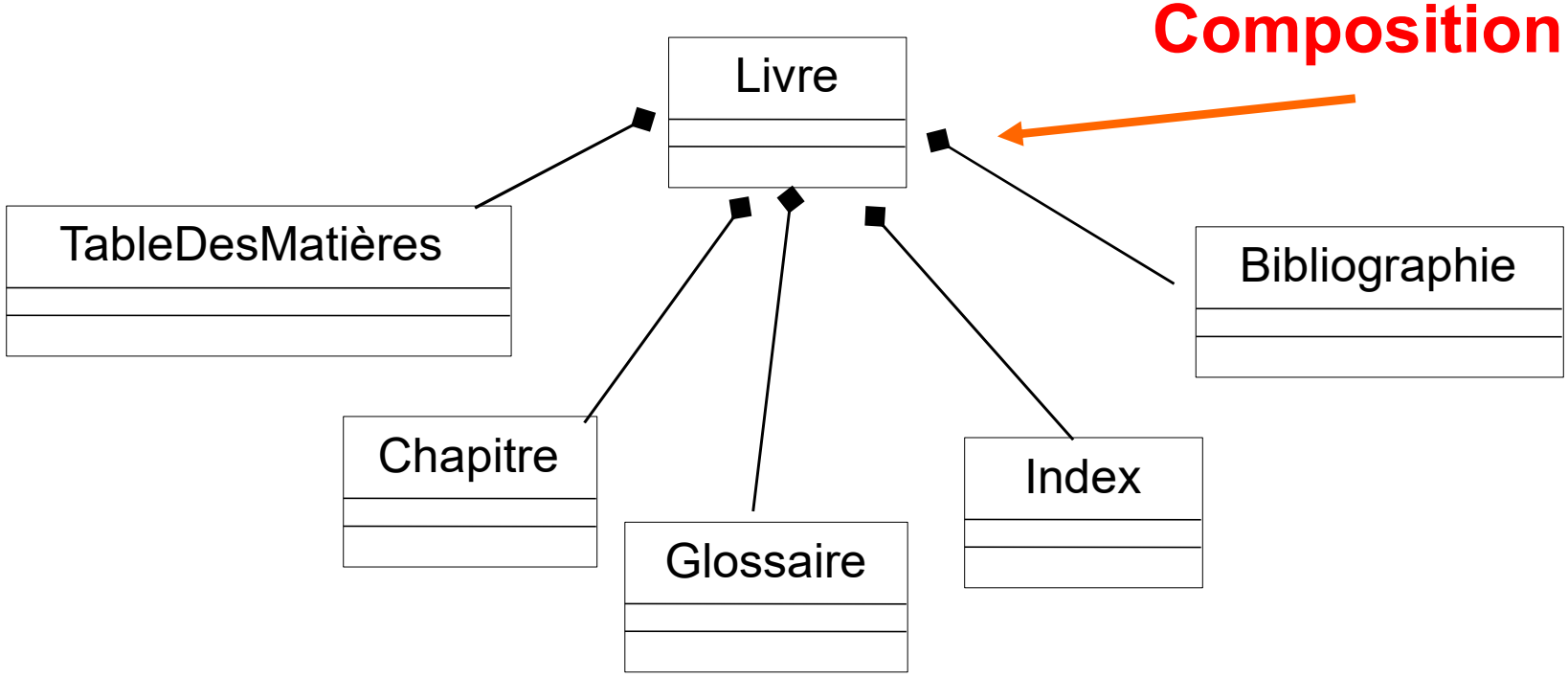
L'agrégation est une association binaire entre un tout et ses parties.



Association spécialisée : la composition

Dans le diagramme de classes seulement, *pas dans le modèle conceptuel!*

La composition est une agrégation dans laquelle la partie ne peut survivre sans son tout.

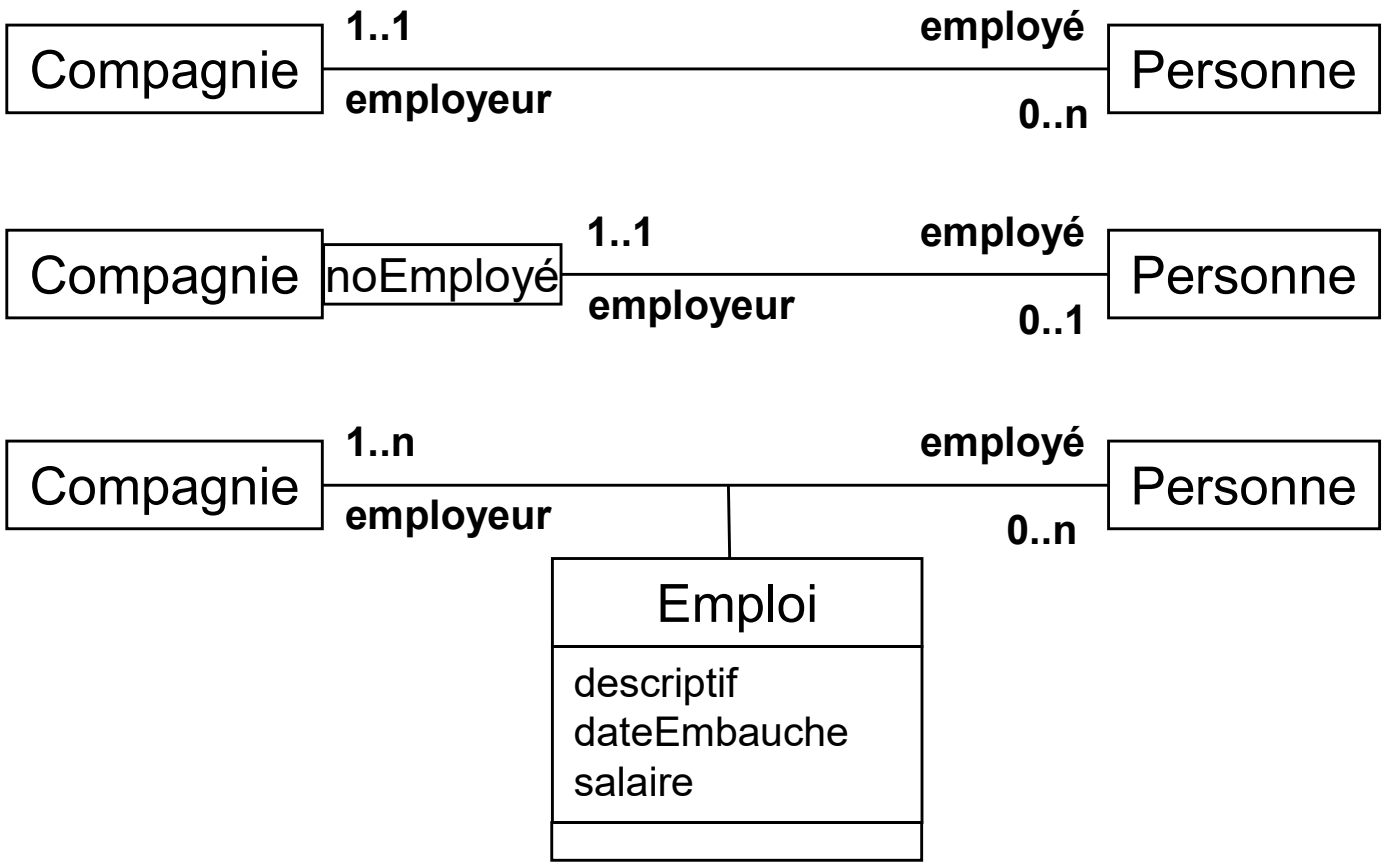


Référence seulement

POLYTECHNIQUE MONTRÉAL
LE GÉNIE EN PREMIÈRE CLASSE

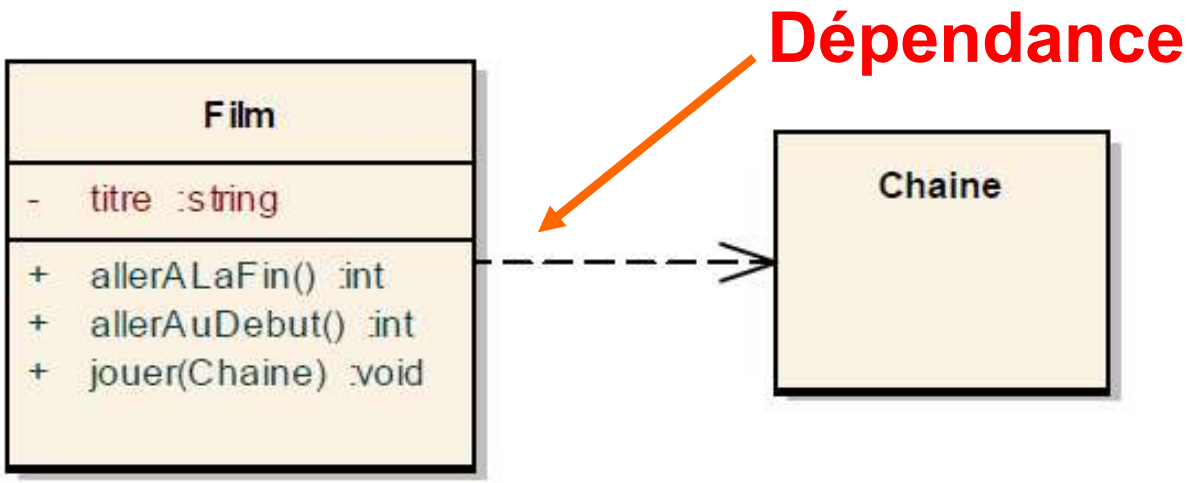
Classe d'association et associations qualifiées

- UML permet de modéliser les propriétés d'une association entre deux classes par un attribut ou une classe d'association.



Dépendance

- Le changement des spécifications d'une entité peut affecter une autre entité, mais pas nécessairement l'inverse.
- Souvent les dépendances seront utilisées pour démontrer qu'une classe en utilise une autre comme paramètre dans la signature d'une opération.



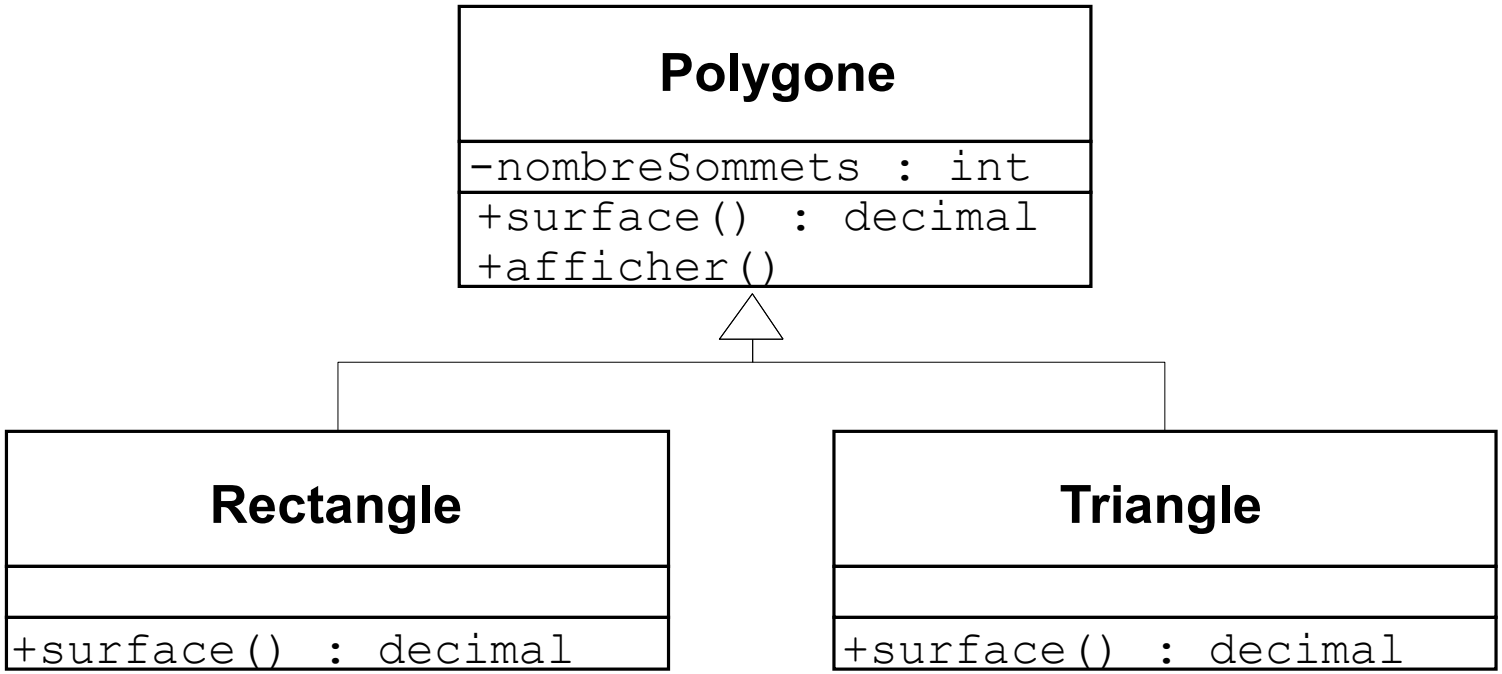
Généralisation

- Relation entre une entité plus générale (super classe ou parent) et une entité plus spécifique (sous classe ou descendant).
- Les objets du descendant peuvent être utilisés partout ou le parent apparaît, mais pas l'inverse.
- Un descendant hérite des propriétés de son parent, spécifiquement des attributs et opérations.
- Un descendant peut avoir des attributs que le parent ne possède pas.
- Une opération du descendant ayant la même signature qu'une opération du parent aura prépondérance sur cette dernière.

  → Polymorphisme

Généralisation et héritage

- L'*héritage* est le mécanisme qui, basé sur la généralisation, permet aux sous-classes d'hériter, c'est à dire d'avoir les mêmes attributs, opérations et associations que la super classe.

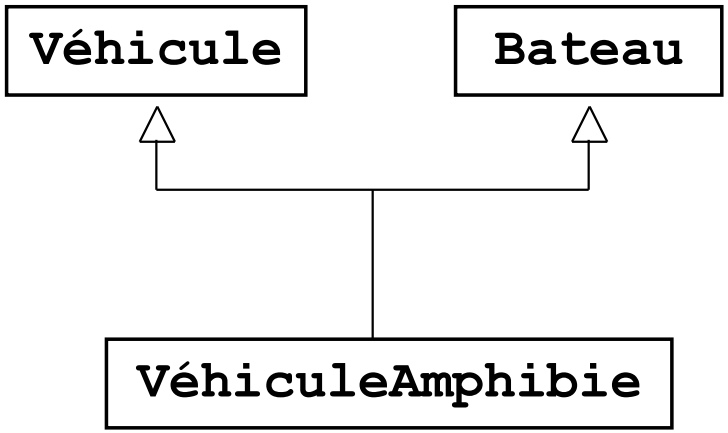


Généralisation: concret et abstrait, racine et feuille

- Une *classe concrète* est une classe qui peut avoir des instances.
- Une *classe abstraite* est une classe qui n'a pas d'instance.
- Une *classe racine* est une classe qui n'a pas de super classe.
- Une *classe feuille* est une classe qui n'a pas de sous-classe.

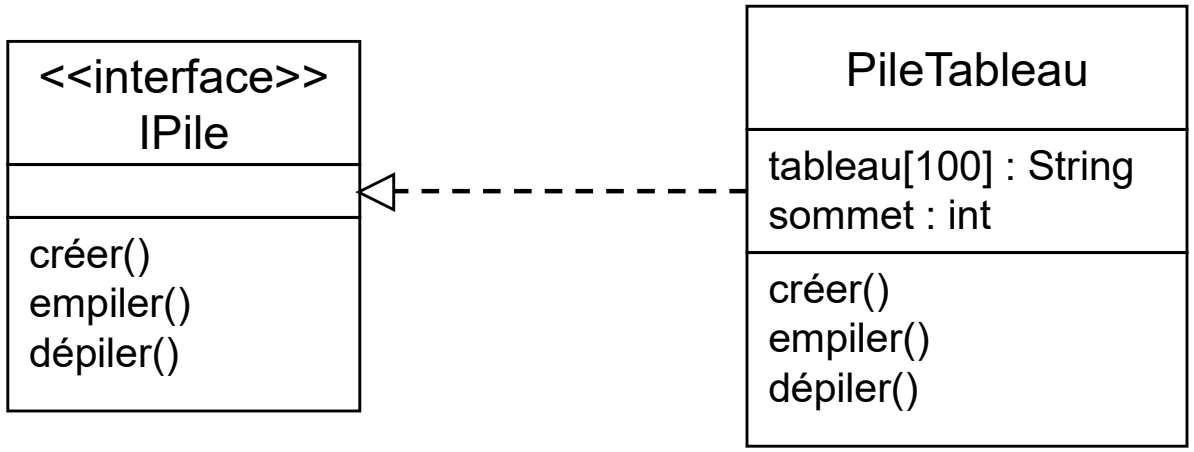
Héritage multiple

- L'*héritage multiple* permet à une classe d'hériter de plusieurs classes.



Réalisation

- Une *réalisation* est une association entre deux classificateurs dans laquelle un classificateur s'engage à exécuter le contrat défini par l'autre classificateur.



Les attributs

Les attributs forment la description de la partie structurelle ou statique:

- Représente une propriété d'une classe partagée par tous les objets de la classe qui, une fois cette dernière instanciée, contiendra un ensemble possible de valeurs.
- Une classe peut avoir plusieurs attributs ou aucun.
- Constitué d'une chaîne de caractère.
- Doit être bref et précis.
- Toutes les premières lettres des mots doivent débuter par une majuscule sauf pour le premier mot.
  - ➔ noDeTéléphone
  - ➔ adresse

Identification et spécification des attributs

Syntaxe d'un attribut en UML:

[visibilité] nom [multiplicité] [: type] [= valeur-initiale] [{propriétés}]

La visibilité peut être:

- *public* (publique) (+)
- *protected* (protégée) (#)
- *private* (privée) (-)

Exemple:

+ numéroTéléphone [1..3] : String = "(514) 555-1212" {readOnly}

Spécification des attributs : exemples

- Type:
  - adresse : String
- Valeur par défaut:
  - adresse : String = “pas d’adresse”
- Multiplicité:
  - prénom : String [3]

Propriétés d'un attribut

- UML définit trois propriétés qui peuvent être utilisées avec les attributs :
  - changeable

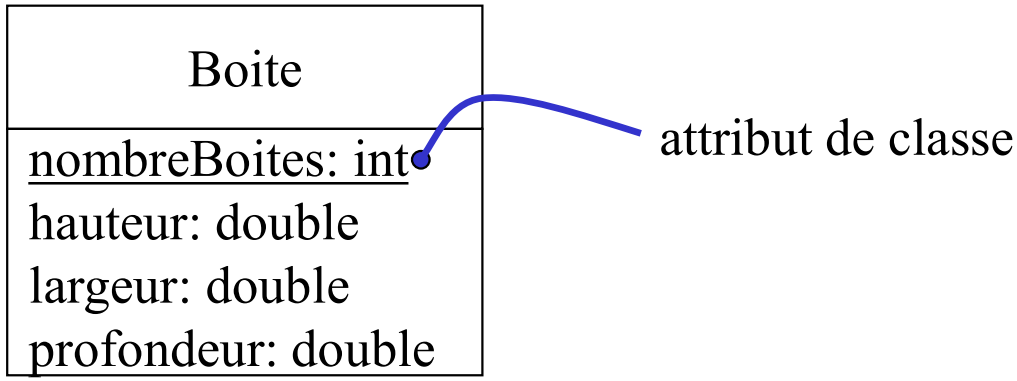
    Aucune restriction sur les modifications de la valeur de l'attribut.
  - addOnly

    Pour les attributs dont la multiplicité est supérieure à 1, des valeurs additionnelles peuvent être ajoutées, mais ne peuvent ensuite être supprimées ou altérées.
  - frozen

    La valeur de l'attribut ne peut pas être changée après l'initialisation de l'objet (pensez à un attribut « const » en C++).

Portée d'un attribut

- Attribut d'instance
  - Attribut dont la valeur peut varier pour chaque instance de la classe. Chaque instance a sa propre copie de l'attribut.
- Attribut de classe
  - Attribut propre à la classe dont la valeur est fixe pour toutes les instances de la classe.



Identification et spécification des attributs

Dans la majorité des cas, au niveau du modèle conceptuel, la grande majorité des attributs devraient être des types de données primitifs, ou des valeurs pures:

- ➔Au niveau conceptuel, les autres attributs devraient être représentés comme des associations,
- ➔Des types d'attributs courants sont: booléen, date, nombre, chaîne de caractères, temps, adresse, couleur, numéro de téléphone, NAS, UPC, code postal, types énumératifs.

Identification et spécification des attributs

Les attributs devraient être des valeurs pures, pour lesquels le fait de parler d'identité n'a pas de sens, par exemple, un logiciel ne veut généralement pas distinguer entre:

- Différentes instances du chiffre 5,
- Différentes instances de la chaîne 'chat',
- Différentes instances du nombre π.

Identification et spécification des attributs

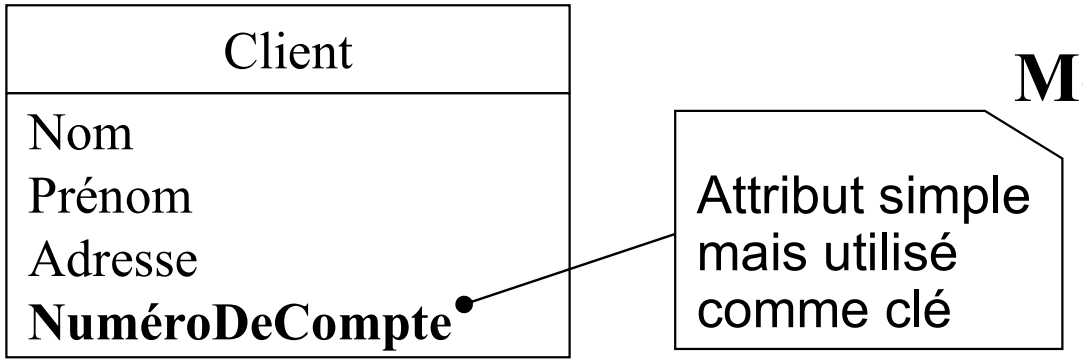
Par contre, il est généralement utile de distinguer deux instances de Personne qui contiennent toutes les deux le nom ‘Michel Tremblay’:

- ➔Les deux instances peuvent représenter deux personnes qui portent le même nom.
- ➔En cas de doute, il vaut mieux modéliser avec un concept plutôt qu’avec un attribut.

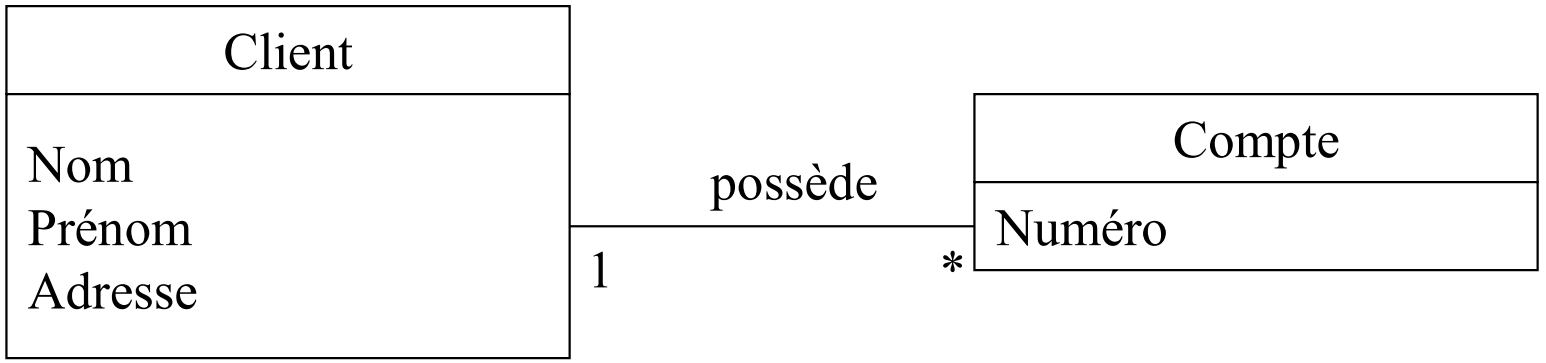
Identification et spécification des attributs

Dans le modèle conceptuel, les attributs ne devraient pas être utilisés pour relier des concepts:

➔Il faut éviter d'insérer des clés externes comme attributs:



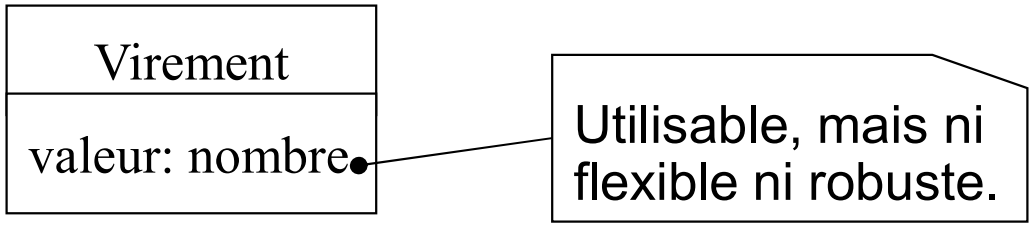
Moins bon



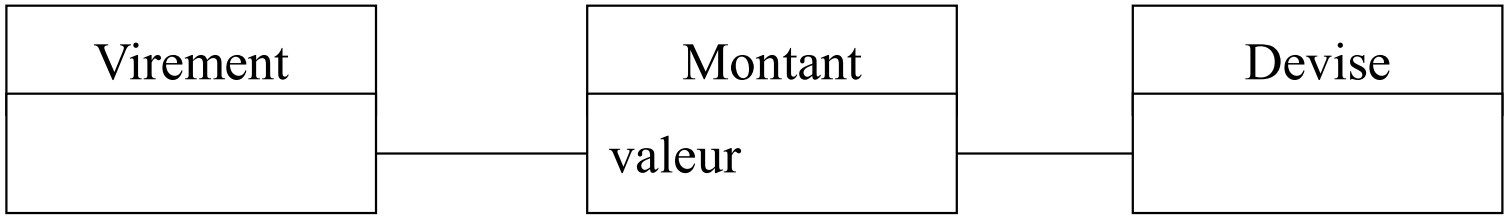
Meilleur

Modélisation de quantités et d'unités

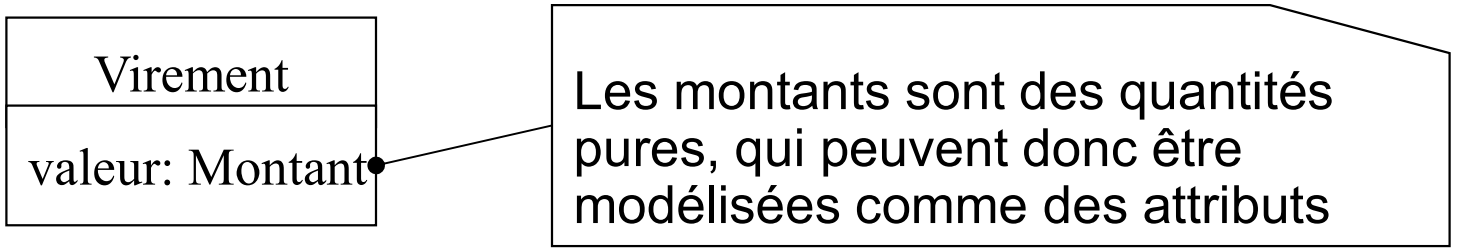
Bien que le montant d'une transaction, par exemple, puisse être représenté avec un nombre, il est plus robuste et flexible de le modéliser à l'aide d'un concept séparé pour pouvoir tenir compte des unités:



Moins bon



Meilleur



Visibilité des attributs

Règle générale, les attributs ne devraient pas être déclarés publics:

- Favorise la consistance de l'interface,
- Fournit un contrôle plus précis sur l'accessibilité aux attributs membres,
- Augmente le niveau d'abstraction d'une classe.

Visibilité des attributs: Consistance de l'interface

En déclarant tous les attributs protégés ou privés, les clients n'ont accès qu'aux opérations:

➔Les clients n'ont plus à se demander s'il faut mettre ou non des parenthèses lors de l'utilisation de membres des objets de la classe.

Visibilité des attributs: Contrôle des droits d'accès

En déclarant tous les attributs protégés ou privés, il devient facile de limiter l'accès en lecture ou en écriture de chaque attribut:

➔Des règles d'accès R/O, R/W et même W/O peuvent être choisies pour chaque attribut individuellement.

Visibilité des attributs: Augmentation du niveau d'abstraction

En déclarant tous les attributs protégés ou privés, il devient facile de modifier l'implantation d'une fonctionnalité sans affecter les clients:

- ➔Une valeur conservée dans chaque objet de la classe peut par exemple être remplacée par une méthode qui recalcule la valeur à chaque appel sans affecter les clients.

Le glossaire

Le glossaire est un document simple qui définit les termes:

- Au minimum, le glossaire, ou dictionnaire du modèle, contient la liste de tous les termes qui doivent être clarifiés afin d'améliorer la communication et éviter les risques de mauvaise compréhension,
- Le glossaire devrait être constitué et continuellement mis à jour dès le début de la phase de définition des requis.

Un exemple de glossaire

Terme	Catégorie	Commentaire
EffectuerDépot	Cas d'utilisation	Description du processus par lequel un client dépose un montant dans son compte.
Montant	Type	Quantité d'argent échangée lors d'une transaction.
Compte.solde	Attribut	Montant total d'argent conservé dans un compte.
Virement.date	Attribut	Date à laquelle la transaction a été effectuée.
DescriptionCompte .TauxIntéret	Attribut	Pourcentage annuel d'intérêt comptabiliser sur le solde du compte