

MINI PROJET – GENIE LOGICIEL

APPLICATION DE COMMERCE ELECTRONIQUE POUR UNE SOCIETE DE VENTE EN LIGNE DE MATERIEL INFORMATIQUE (INFORMATICA STORE)

Université des sciences et de la technologie Houari Boumediene (USTHB)
3^{ème} année Licence académique
Année universitaire 2010/2011

1 Contenu du dossier à rendre

Ce mini-projet est à réaliser par équipe de 2 ou 3. Le mini-projet donnera lieu à la remise d'un dossier, dans un format papier. La date de remise du mini-projet vous sera communiquée plus tard. Elle coïncidera avec la fin des cours, avant les examens semestriels.

Le dossier comprendra l'analyse et la conception du système informatique de l'entreprise fictive INFORMATICA.

Il sera composé des éléments suivants :

- L'analyse et la conception en UML, avec les diagrammes de cas d'utilisation, un diagramme détaillé de classes (signature typée des attributs et des méthodes) et quelques diagrammes d'activité et de séquence.

2 Présentation

Ce projet vise à analyser le système informatique de l'entreprise fictive INFORMATICA. Cette société vend du matériel informatique. Elle continue d'exercer son métier telle qu'elle le faisait à ses débuts, c'est-à-dire qu'elle répertorie ses clients et ses articles sur des fiches de papier bristol, reçoit les commandes par fax, les chèques par courrier puis envoie le bon de commande au client. Une fois le chèque encaissé par la banque AlgeriaBank, elle utilise la société de transport TranspoEx pour acheminer la marchandise aux clients. INFORMATICA est depuis toujours implantée dans l'ouest de l'Algérie où sont domiciliés ses principaux clients.

Récemment elle a ouvert son marché à d'autres villes de l'Algérie, ainsi qu'à l'étranger. INFORMATICA n'arrive plus à gérer manuellement cette expansion et souhaite créer un système informatique pour lui permettre de faire face à sa récente croissance. Elle attend de celui-ci qu'il lui permette de vendre son matériel en ligne, de gérer son catalogue d'articles et sa base de données de clients. De plus, ses partenaires (la banque AlgeriaBank et la société de transport TranspoEx) souhaitent avoir la possibilité d'échanger des données aux formats électroniques via Internet.

L'objectif fondamental du futur site web de INFORMATICA est de permettre aux clients de rechercher des articles de se constituer un panier virtuel, puis de pouvoir les commander et les payer directement sur le Web.

3 Expression des besoins

Afin de décrire les besoins de la société INFORMATICA, nous utiliserons des diagrammes de cas d'utilisation, de séquence et d'activité UML.

3.1 Exigences fonctionnelles

Le site web de la société INFORMATICA devra regrouper toutes les fonctionnalités nécessaires de Gestion des clients, de recherche d'article, de commande et d'achat.

Gérer les clients

INFORMATICA veut pouvoir créer ses clients dans le système à partir des données existantes. Elle souhaite également pouvoir les modifier, les supprimer et les rechercher. Un employé de la société est chargé de la gestion des clients. Les éléments caractérisant un client sont les suivants :

- identifiant unique du client;
- login et mot de passe utilisés par le client pour se connecter à l'application ;
- prénom et nom de famille;
- numéro de téléphone où l'on peut joindre le client et son adresse mail ;
- adresse postale;
- pays de résidence, ville et code postal;
- date de naissance : INFORMATICA veut pouvoir envoyer des cartes de vœux à la date d'anniversaire du client ;
- âge du client.

Une fois les données saisies, l'employé souhaite pouvoir les exploiter. Ainsi, à partir d'un identifiant, le système doit donner la possibilité d'afficher les coordonnées du client et proposer à l'employé de les mettre à jour ou de les supprimer. Dans le cas de la suppression, le système doit attendre une confirmation de l'employé avant de supprimer définitivement le client du système.

Le système doit aussi pouvoir afficher la totalité des clients présents dans le système.

Se créer un compte

Un internaute peut créer un compte dans le système et devenir ainsi un client. Dans ce cas, l'internaute ne peut renseigner que ses propres données. Pour se créer un compte, il doit saisir un login, un mot de passe et ressaisir une seconde fois son mot de passe. Le système lui demande alors de saisir ses coordonnées et informations personnelles (identiques à celles des clients).

Se connecter et se déconnecter

Un client peut se connecter et se déconnecter du système. Il doit saisir son login et son mot de passe. Il est reconnu par le système, qui affiche alors son nom et prénom. Lorsque le client se déconnecte, il redevient internaute jusqu'à sa prochaine connexion.

Consulter et modifier son compte

Le client peut également consulter et mettre à jour ses informations personnelles dans le système, mais il doit être connecté au système.

Gérer le catalogue

Le catalogue d'articles de la société INFORMATICA est divisé en catégories (ordinateurs de bureau, imprimantes, scanners, Laptops, ...). Une catégorie est définie par les données suivantes :

- identifiant unique de la catégorie ;
- nom;
- description.

Chacune de ces catégories est divisée en produits. Par exemple pour les ordinateurs, on peut avoir les produits suivants : Dell, Macintosh, HP, ... Chaque produit est défini comme suit :

- identifiant unique du produit;
- nom;
- description.

Enfin, chaque produit est, à son tour, divisé en articles. Ce sont ces articles qui sont proposés et vendus aux clients. Par exemple, le produit Macintosh regroupe les articles suivants : MacBook, iMac, MacPro, MacBook Air. Chaque article est défini comme suit :

- identifiant unique de l'article ;
- nom;
- prix unitaire de l'article ;
- image : elle représente l'article en question

Visualiser les articles du catalogue

Les internautes et les clients peuvent visualiser la totalité du catalogue. L'organisation de l'affichage doit être intuitive, c'est-à-dire que le système doit afficher la liste des catégories, à partir desquelles le client choisit un produit puis un article.

Pour chaque article, une image devra être affichée. À tout moment, il doit être possible d'afficher les produits d'une catégorie différente.

Rechercher un article

En plus de visualiser le catalogue de manière linéaire, les internautes et les clients peuvent rechercher les articles contenus dans le système à partir d'une chaîne de caractères. Si aucun article ne correspond aux critères demandés, une information est affichée à l'internaute pour lui indiquer que sa recherche n'a pas abouti et qu'il doit modifier le critère de recherche.

Acheter des articles

Pour acheter des articles, le client doit être connecté au système. Il visualise le catalogue ou recherche un article. Lorsqu'il est intéressé par un article, il lui suffit de l'ajouter dans son panier électronique. Cette opération peut être exécutée plusieurs fois sur des articles différents. Le client a ensuite la possibilité de modifier la quantité désirée pour chaque article ou supprimer un ou plusieurs de ces articles du panier. Lorsque la quantité d'un article est inférieure ou égale à zéro, l'article est automatiquement supprimé du panier.

Pendant toute la durée de sa session, le client peut visualiser le contenu de son panier quand bon lui semble. Lorsque le Caddie est vide, un message avertit le client. Sinon, le système affiche la liste des articles avec le nom, la description du produit, la quantité désirée, le prix unitaire et le sous-total (prix unitaire \times quantité). Le montant total du panier est également renseigné. Ce Caddie est illimité en taille, un client peut donc acheter autant d'articles qu'il le souhaite.

Lorsque le client est satisfait, il valide son panier électronique. Il doit alors saisir les informations de sa carte bancaire ainsi que l'adresse de livraison. Par défaut, l'adresse de livraison est la même que celle du client mais elle peut être modifiée. Les données de la carte bancaire sont les suivantes :

- Numéro de carte bancaire.
- Type de carte bancaire (Visa, Master Card et American Express).
- Date d'expiration de la carte bancaire.

Créer un bon de commande

Une fois le panier électronique validé par le client, le système crée automatiquement un bon de commande. Ce dernier contient toutes les informations nécessaires pour être traité :

- un numéro de bon de commande ;
- la date de création de ce bon de commande ;
- les références du client qui a acheté les articles ;
- **les lignes de commande** : une ligne de commande référence l'article acheté et sa quantité. Il y a autant de lignes de commande que d'articles contenus dans le panier électronique ;
- les informations de la carte bancaire ;
- l'adresse de livraison des articles.

Cette création du bon de commande entraîne plusieurs traitements :

1. Le bon commande est imprimé puis stocké dans les archives de la société INFORMATICA.
2. Toutes les informations nécessaires à l'acheminement des articles sont envoyées au transporteur TranspoEx de manière électronique au format XML. TranspoEx livre ensuite les articles aux clients.
3. Un e-mail est envoyé au client pour l'informer du bon déroulement de sa transaction. Cet e-mail contient le numéro du bon de commande ainsi qu'un récapitulatif de son contenu.

NB. Les données de la carte bancaire sont validées par BarkBank. Si la banque rejette la carte bancaire, le client en est averti et peut ressaisir ses données.

Visualiser et supprimer les commandes

L'employé peut visualiser la liste des commandes présentes dans le système. Pour chacune de ces commandes, il peut en consulter les informations et les supprimer.

4 Travail à réaliser

4.1 Analyse

Pour exprimer les besoins de la société INFORMATICA, nous allons utiliser le formalisme UML des cas d'utilisation.

A. IDENTIFIER LES ACTEURS

Les acteurs candidats sont systématiquement :

- les utilisateurs humains directs;
- les autres systèmes connexes qui interagissent aussi directement avec le système étudié.

Distinguer acteur principal et acteur secondaire :

- l'acteur principal est celui pour qui le cas d'utilisation produit un résultat observable; généralement, c'est lui qui déclenche, le cas d'utilisation.
- Les acteurs secondaires sont les autres participants du cas d'utilisation
 - sollicités pour des informations complémentaires
 - consultent ou surveillent le système

B. IDENTIFIER LES CAS D'UTILISATION

L'ensemble des cas d'utilisation doit décrire exhaustivement les exigences fonctionnelles du système. Chaque cas d'utilisation correspond donc à une fonction du système, selon le point de vue d'un de ses acteurs.

Pour chaque acteur, il convient de :

- rechercher les différentes intentions avec lesquelles il utilise le système,
- déterminer dans le cahier des charges les services fonctionnels attendus du système.

Nommez les cas d'utilisation par un verbe à l'infinitif suivi d'un complément, du point de vue de l'acteur (et non pas du point de vue du système).

C. ORGANISER LES CAS D'UTILISATION

1) Ajouter des relations entre cas d'utilisation:

a) Relation «*include*» entre cas d'utilisation:

- i) Le cas de base incorpore explicitement un autre cas.
- ii) Le cas inclus n'est jamais exécuté seul, mais uniquement en tant que partie
- iii) Cette relation est utilisée pour factoriser un comportement commun présent dans plusieurs cas d'utilisation

b) Relation «*extend*» entre cas d'utilisation

- i) Le cas de base incorpore un autre cas, à un endroit spécifié.
- ii) Le cas de base peut fonctionner tout seul, ou être aussi complété par un autre.
- iii) Utilisé pour séparer le comportement optionnel.

c) de généralisation/spécialisation

- i) Les cas d'utilisation peuvent être hiérarchisés par généralisation/spécialisation.
- ii) Les cas d'utilisation enfant héritent des comportements de leur parent.

2) Les exceptions dans les cas d'utilisation

Un cas d'utilisation décrit le comportement normal de l'application. Si des exceptions apparaissent, elles peuvent être référencées dans la description à l'aide de numéros. Par exemple : « le client à un identifiant unique, si cette valeur n'est pas unique, une exception est levée ; « L'adresse du client » est une donnée obligatoire. Si cette donnée est manquante, une exception doit être levée.

D. DESCRIPTION TEXTUELLE DES CAS D'UTILISATION

On se limitera dans cette étude à la description textuelle des cas d'utilisation suivants :

- Créer un compte;
- Acheter des articles;
- Créer un bon de commande.

La fiche de description textuelle d'un cas d'utilisation n'est pas normalisée par UML.

Décrire les cas d'utilisation susmentionnés sous la forme du tableau ci-dessous :

Nom	Nom du cas d'utilisation : le plus souvent le nom contient un verbe à l'infinitif puisqu'un cas d'utilisation décrit une interaction entre un acteur et le système.
Résumé	Une brève description du cas d'utilisation.
Auteur	Nom de l'auteur de cette description
Dates de création	Dates de création de la description
Date de mise à jour	Date de la dernière mise à jour
Acteurs	Cette rubrique décrit la liste des acteurs interagissant avec le cas d'utilisation.
Pré conditions	Ce sont les conditions nécessaires pour déclencher le cas d'utilisation. Contient aussi les contraintes sur l'état du système avant l'exécution
Post conditions	Ce sont les conditions remplies après l'exécution du cas d'utilisation (état du système après réalisation du cas d'utilisation).
Description des scénarios	Décrit le scénario nominal, les scénarios (ou enchaînements) alternatifs, les scénarios (ou enchaînements) d'erreur.
Exigences non fonctionnelles (optionnel)	Ajoute, si c'est pertinent, les informations suivantes : fréquence d'utilisation dans le système, volumétrie, disponibilité, fiabilité, intégrité, confidentialité, Performances (Contraintes temporelles), concurrence, etc. Précise également les contraintes d'interface homme-machine comme des règles d'ergonomie, une charte graphique, etc.

Un cas d'utilisation n'est ni une transaction, ni une fonction: ne pas descendre trop bas en terme de granularité (difficulté de savoir à quel niveau de détail s'arrêter.)

Un cas d'utilisation ne doit pas se réduire à une seule séquence ou à une simple action.

E. IDENTIFICATION DES CLASSES CANDIDATES DU MODELE STATIQUE D'ANALYSE

Une des étapes dans l'analyse peut aussi être la réalisation d'une liste d'objets candidats. Cette liste peut être établie en soulignant les noms dans la définition du problème.

Réaliser un premier diagramme de classes qui contiendra les abstractions clés retenues et qui mettra en évidence des associations entre les classes. Les associations seront de simples traits banalisés : pas de nom, pas de multiplicité (cardinalité).

F. DIAGRAMME D'ACTIVITES DE LA VISUALISATION DES ARTICLES DU CATALOGUE

Donnez le diagramme d'activités des actions effectuées pour visualiser le contenu du catalogue.

Il doit être lu de la manière suivante : « Le système affiche les catégories du catalogue. Lorsque l'internaute en sélectionne une, le système affiche les produits de la catégorie... Notez qu'à tout moment on peut revenir à l'action - Afficher les produits de la catégorie ».

G. DIAGRAMME DE SEQUENCE

Complétez la description textuelle des cas d'utilisation : Créer un compte; Acheter des articles; Créer un bon de commande par une représentation graphique UML très utile : le diagramme de séquence.

4.2 Conception

A. DIAGRAMME DE CLASSE DE CONCEPTION

Réaliser un diagramme de classe sur le système étudié en partant du diagramme réalisé dans l'étape précédente.

Le nouveau diagramme contiendra des informations complémentaires en termes de caractéristiques des associations (nom, cardinalités¹, **classes d'associations**², ...).

- Définir les rôles des classes dans les associations.
- Compléter les classes en termes d'attributs et opérations. Si nécessaire, créer des classes pour définir de nouveaux types.
- Définir les niveaux de visibilités (public, private, protected), les types, les paramètres, les valeurs d'initialisation.

¹ Vous pouvez réaliser un digramme objet pour valider les cardinalités.

² Voir cours UML

- Reprendre les associations existantes et définir si ce sont de simples associations, des agrégations ou des compositions. Faire éventuellement des restrictions sur le sens de navigation.
- Identifier d'éventuelles classes d'associations permettant de caractériser les associations.
- Identifier les liens d'héritage possibles. Cette tâche fait ajouter des classes au diagramme initial qui permette une généralisation ou une spécialisation de classe.
- Vérifier que le diagramme permet de remplir les différents cas d'utilisation et leurs scénarii associés.

Pour faire cette vérification, il faut prendre un scénario et suivre les associations sur le diagramme de classes en même temps que les messages sont envoyés. Les associations présentes doivent permettre de « porter » la communication entre les objets.

B. DU DIAGRAMME DE CLASSES UML VERS UN SCHEMA RELATIONNEL

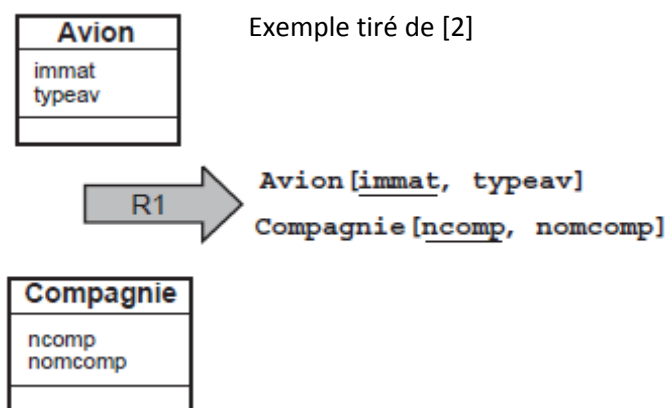
Proposez un schéma relationnel qui correspond au diagramme de classes élaborer lors de l'étape précédente.

Nous donnons ci-après quatre règles (de R1 à R4) pour traduire un schéma conceptuel

UML en un schéma relationnel équivalent [2]. Il existe d'autres solutions de transformation, mais ces règles sont les plus simples et les plus opérationnelles.

Transformation des classes

R1 Chaque classe du diagramme UML devient une relation. Chaque attribut de la classe devient attribut de la relation. Il faut choisir un attribut de la classe pouvant jouer le rôle d'identifiant.
Si aucun attribut ne convient en tant qu'identifiant, il faut en ajouter un de telle sorte que la relation dispose d'une clé primaire (les outils proposent l'ajout de tels attributs).



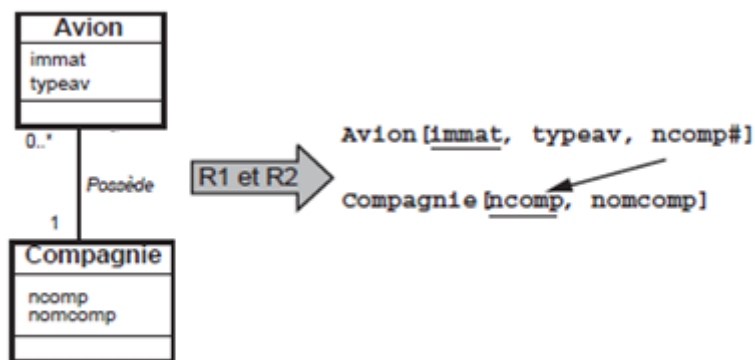
Transformation des associations

Les règles de transformation des associations dépendent des cardinalités/multiplicités maximales des associations. Nous distinguons trois familles d'associations :

- un-à-plusieurs ;
- plusieurs-à-plusieurs ou classes-associations, et n-aires ;
- un-à-un.

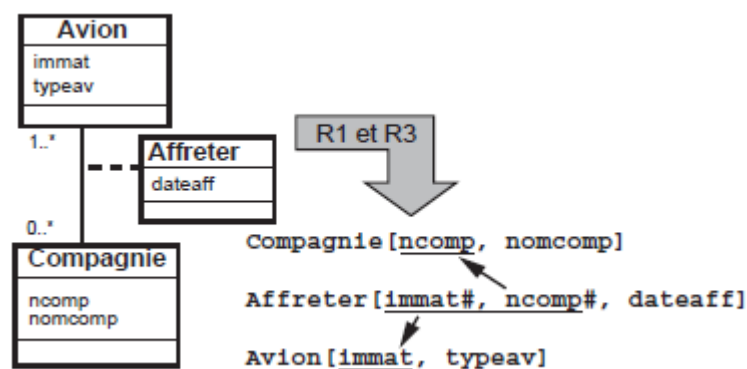
Associations un-à-plusieurs

R2 Il faut ajouter un attribut de type clé étrangère dans la relation Avion. L'attribut porte le nom de la clé primaire de la relation Compagnie.

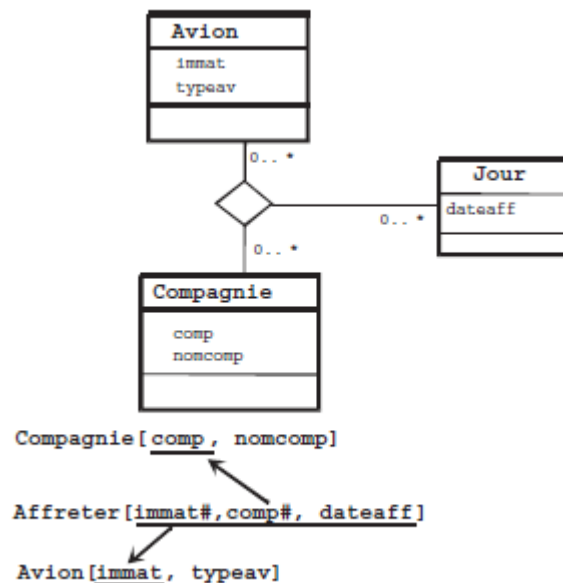


Associations plusieurs-à-plusieurs et n-aires

R3 La classe-association devient une relation dont la clé primaire est composée par la concaténation des identifiants des classes connectés à l'association. Les attributs de la classe-association doivent être ajoutés à la nouvelle relation.



Transformation d'une association plusieurs-à-plusieurs



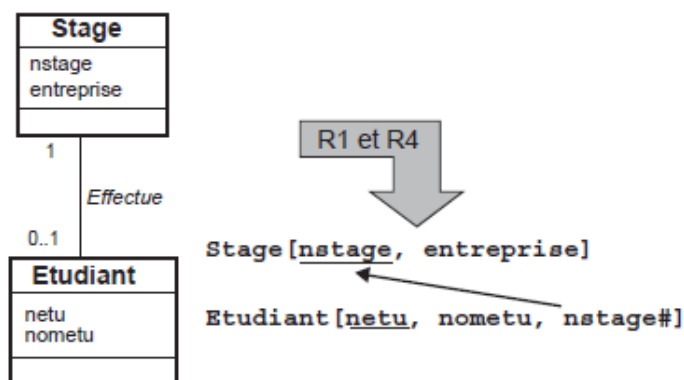
Transformation d'une association n-aire

Associations un-à-un

R4 La règle est la suivante, elle permet d'éviter les valeurs NULL dans la base de données.

Il faut ajouter un attribut clé étrangère dans la relation dérivée de la classe ayant la multiplicité minimale égale à un. L'attribut porte le nom de la clé primaire de la relation dérivée de la classe connectée à l'association.

Si les deux cardinalités (multiplicités) minimales sont à zéro, le choix est donné entre les deux relations dérivées de la règle R1. Si les deux cardinalités minimales sont à un, il est sans doute préférable de fusionner les deux classes en une seule.



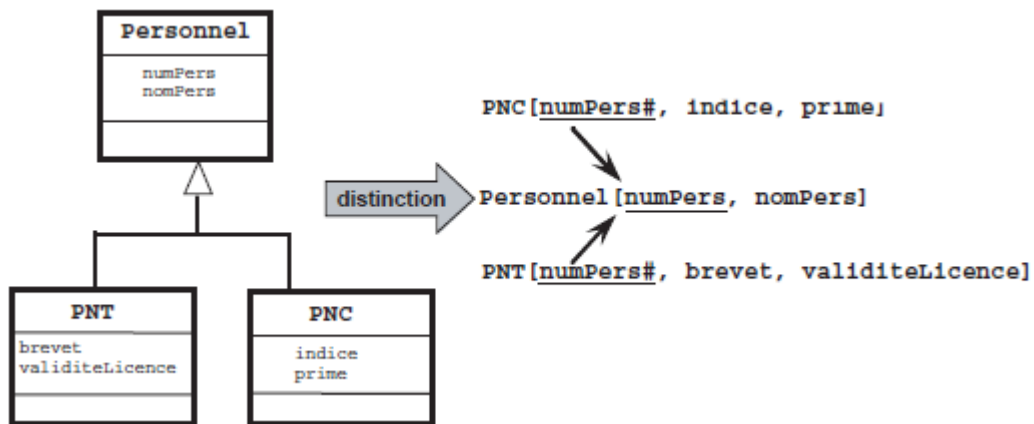
Transformation de l'héritage

Trois décompositions sont possibles pour traduire une relation d'héritage en fonction des contraintes existantes :

- décomposition par distinction ;
- décomposition descendante (push-down). S'il existe une contrainte de totalité ou de partition sur l'association d'héritage, il y aura deux cas possibles de décomposition ;
- décomposition ascendante (push-up).

Décomposition par distinction

Il faut transformer chaque sous-classe en une relation. La clé primaire de la sur-classe migre dans la (les) relation(s) issue(s) de la (des) sous-classe(s) et devient à la fois clé primaire et clé étrangère.

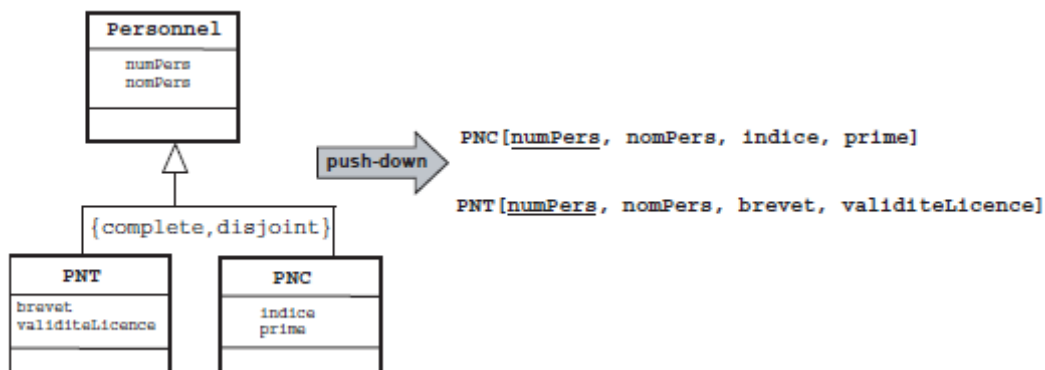


Décomposition par distinction d'une association d'héritage

Décomposition descendante (push-down)

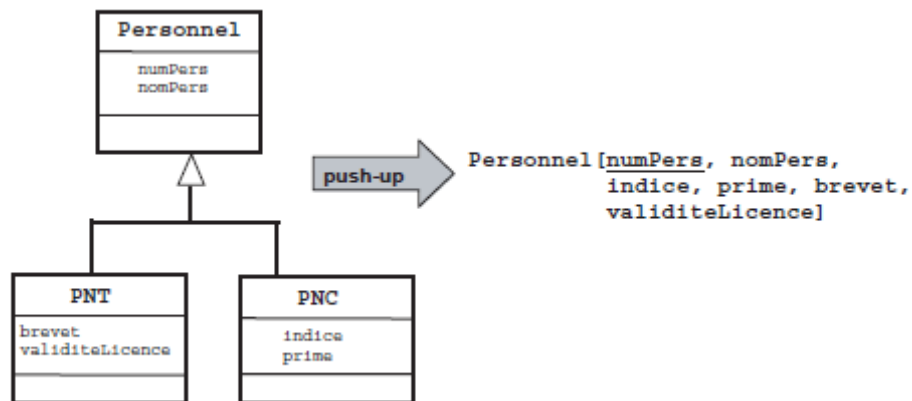
Deux cas sont possibles :

- S'il existe une contrainte de totalité ou de partition sur l'association, il est possible de ne pas traduire la relation issue de la sur-classe. Il faut alors faire migrer tous ses attributs dans la (les) relation(s) issue(s) de la (des) sous-classe(s).
- Dans le cas contraire, il faut faire migrer tous ses attributs dans la ou les relation(s) issue(s) de la (des) sous-classe(s) dans la (les) relation(s) issue(s) de la (des) sous-classe(s).



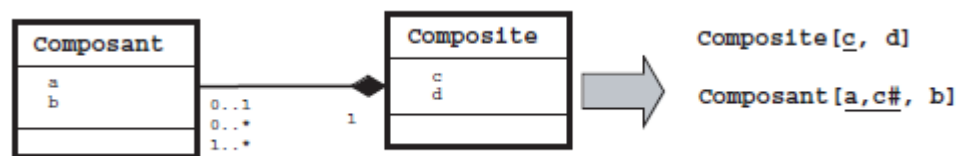
Décomposition ascendante (push-up)

Il faut supprimer la (les) relation(s) issue(s) de la (des) sous-classe(s) et faire migrer les attributs dans la relation issue de la sur-classe.



Transformation d'une composition

Alors que l'agrégation partagée de UML se traduit au niveau logique comme une simple association, il n'en est pas de même pour la composition.



Transformation d'une composition

5 Outils de modélisation

- Les diagrammes UML pourront être réalisés avec un simple outil de dessin, un outil UML gratuit ou en version d'évaluation. Pour produire les diagrammes UML avec des outils spécialisés, vous pouvez par exemple utiliser :
 - EclipseUML
http://www.eclipsedownload.com/download_studio_eclipse_3x.html
 - Objectteering
http://www.objectteering.com/downloads_uml_free_edition.php
 - Poseidon (version « community edition ») <http://www.gentleware.com/>.
 - Umbrella.
 - ArgoUML
 - Etc³.

6 Références

[1] Pascal ROQUES, Franck VALLÉE « UML en Action De l'analyse des besoins à la conception en Java ». Edition Eyrolles (2^{ème} édition)

[2] Christian SOUTOU « UML2 pour les bases de données » Edition Eyrolles

[3] Antonio Goncalves, les Cahiers du Programmeur. JEE5

³ Voir <http://uml.developpez.com/outils/>