

Imbrication en UML et en relationnel

Table des matières

Objectifs	3
I - Cours	4
1. Imbrication en UML : composition, attribut multivalué, datatype	4
1.1. Exercice : Entreprise	4
1.2. Association de composition	4
1.3. Notion de clé locale	6
1.4. Clé locale dans les compositions	6
1.5. Attribut multivalué	7
1.6. Attribut composé	8
1.7. Stéréotype datatype	8
2. Transformation de l'imbrication en relationnel	9
2.1. Transformation des compositions	9
2.2. Transformation des attributs multivalués	10
2.3. Transformation des datatypes	11
2.4. Exercice : Pays	12
2.5. Trousseau de clés	13
II - Exercices	14
1. Exercice : Objets Numériques Libres	14
2. Exercice : Lab III	15
Contenus annexes	17
Glossaire	20
Abréviations	21
Index	22

Objectifs



Prérequis :

- Savoir faire un MCD UML avec des classes, des associations simples, de l'héritage.
- Savoir faire un MLD relationnel à parti d'un MCD UML avec des classes, des associations simples, de l'héritage.



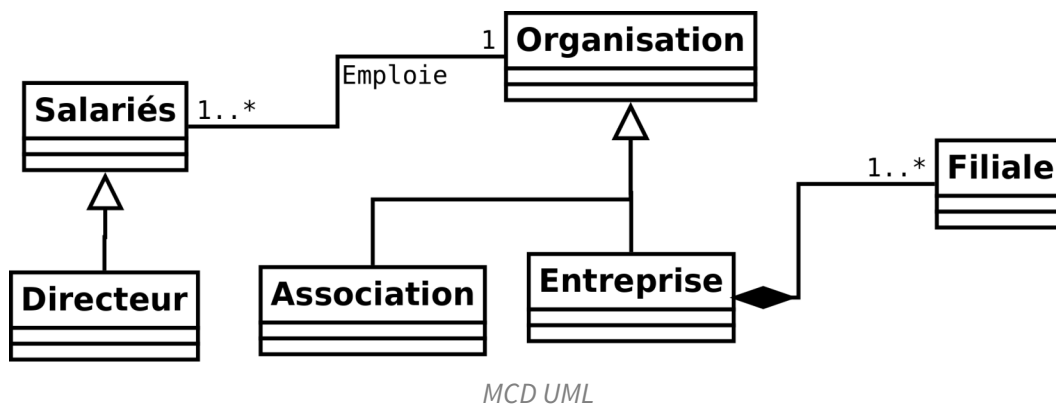
1. Imbrication en UML : composition, attribut multivalué, datatype

Objectifs

Maîtriser le diagramme de classe UML dans le cas de la conception de BD.

1.1. Exercice : Entreprise

En analysant le schéma UML ci-après, sélectionner toutes les assertions vraies.



- ☐ Une association peut employer un directeur.
- ☐ Une association peut employer plusieurs directeurs.
- ☐ Une association peut ne pas employer de directeur.
- ☐ Il existe des organisations qui ne sont ni des entreprises ni des associations.
- ☐ Une filiale peut appartenir à plusieurs entreprises.
- ☐ Une entreprise peut racheter la filiale d'une autre entreprise

1.2. Association de composition

Association de composition



Définition

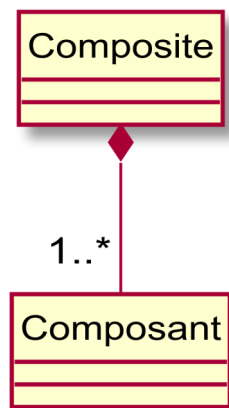
On appelle composition une association particulière qui possède les propriétés suivantes :

- La composition associe une classe composite et des classes parties, tel que tout objet partie appartient à un et un seul objet composite. C'est donc une association 1:N (voire 1:1).
- La composition n'est pas partageable, donc un objet partie ne peut appartenir qu'à un seul objet composite à la fois.
- Le cycle de vie des objets parties est lié à celui de l'objet composite, donc un objet partie disparaît quand l'objet composite auquel il est associé disparaît.



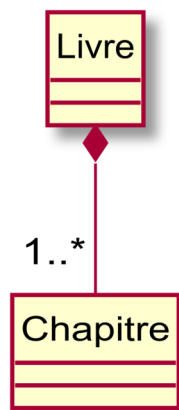
- La composition est une association particulière (binaire de cardinalité contrainte).
- La composition n'est pas symétrique, une classe joue le rôle de conteneur pour les classes liées, elle prend donc un rôle particulier a priori.
- La composition est une agrégation avec des contraintes supplémentaires (non partageabilité et cycle de vie lié).

Notation d'une composition en UML



Notation de la composition en UML

Exemple de composition



Livre et chapitres

On voit bien ici qu'un chapitre n'a de sens que faisant partie d'un livre, qu'il ne peut exister dans deux livres différents et que si le livre n'existe plus, les chapitres le composant non plus.

Composition et cardinalité



La cardinalité côté composite est toujours de exactement 1.

Côté partie la cardinalité est libre, elle peut être 0..1, 1, * ou bien 1..*.

Composition et agrégation



L'agrégation (noté avec un losange blanc) est différent de la composition, puisqu'elle ne pose aucune dépendance au cycle de vie (c'est une association classique)

Composition et entités faibles

**Complément**

La composition permet d'exprimer une association analogue à celle qui relie une entité faible à une entité identifiante en modélisation $E-A^*$. L'entité de type faible correspond à un objet partie et l'entité identifiante à un objet composite.

Voir aussi

**Complément**

- Agrégation (cf. p.17)

1.3. Notion de clé locale

Le concept de clé locale appartient au niveau conceptuel, il est hérité de l'entité faible du modèle conceptuel Entité-Association (équivalent de la composition en UML). Dans une entité faible ou une composition, une clé de la classe composant est dite locale, car elle ne permet d'identifier l'objet que si l'on connaît la classe composite.

**Définition**

Dans certaines constructions en UML (association N:M et composition) la clé peut être locale, c'est à dire qu'au lieu d'identifier pleinement un objet (comme une clé classique), elle identifie un objet étant donné un contexte (les membres de l'association N:M ou l'objet composite).

**Attention**

Une clé locale n'est donc pas une clé au sens relationnel, elle ne permet pas d'identifier un enregistrement (mais elle deviendra une partie d'une clé lors du passage au relationnel).

1.4. Clé locale dans les compositions

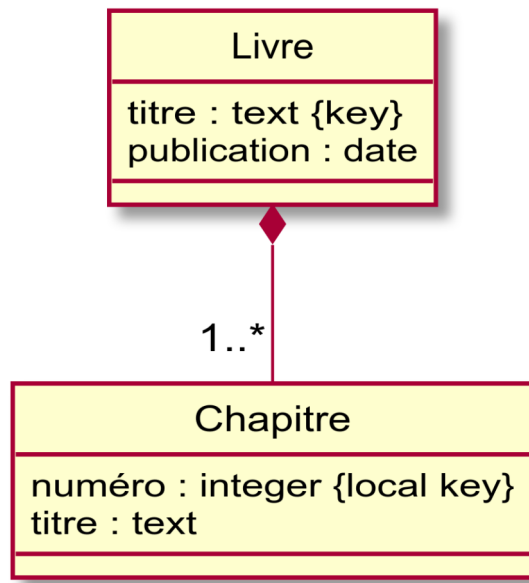
**Rappel**

Notion de clé locale (cf. p.6)

**Attention**

Dans une composition on a en général uniquement des clés locales : en effet si le composant est identifiable indépendamment de son composite, c'est en général qu'il a une vie propre et donc que l'on est pas en présence d'une composition.

? Exemple



Livre et chapitres (avec clé locale)

Il existe deux plusieurs livres avec un chapitre numéro 1, mais il n'existe pas deux chapitre numéro 1 au sein d'un même livre.

1.5. Attribut multivalué

🔑 Définition

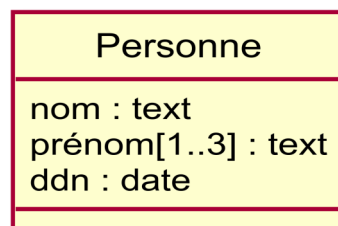
Un attribut multivalué est un attribut qui peut prendre plusieurs valeurs distinctes dans son domaine.

§ Syntaxe

```
1 attribut_mv[nbMinValeurs..nbMaxValeurs]:type
```

La classe Personne

? Exemple



Attribut multivalué

Dans cet exemple, le prénom est un attribut multivalué, ici une personne peut avoir de 1 à 3 prénoms.

Composition et attribut multivalué

📋 Conseil

Une composition avec une classe partie dotée d'un seul attribut peut s'écrire avec un attribut multivalué.

Voir aussi



Composition (cf. p.4)

1.6. Attribut composé



Définition

Un **attribut peut être composé** (ou composite) joue le rôle d'un groupe d'attributs (par exemple une adresse peut être un attribut composé des attributs numéro, type de voie, nom de la voie). Cette notion renvoie à la notion de variable de type `Record` dans les langages de programmation classiques.



Remarque

Ce concept est hérité de la modélisation E-A.

On utilise pas les attributs dérivés en UML



Attention

En UML on préfère l'usage de compositions (ou de *dataType*) aux attributs composés.

- On utilisera des composition ou des *dataTypes* pour les attributs composés.
- On utilisera des compositions pour les attributs composés et multivalués.

Voir aussi



Composition (cf. p.4)

DataType (cf. p.8)

1.7. Stéréotype datatype

Datatype

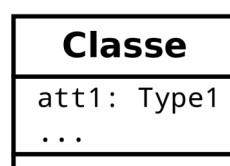
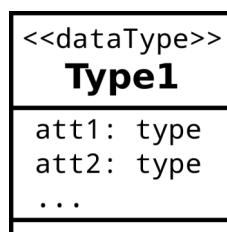


Définition

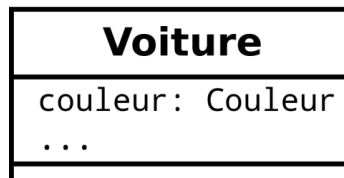
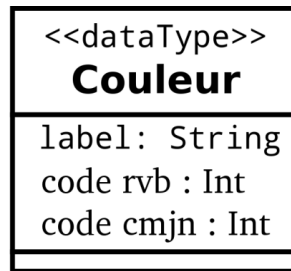
Les *datatypes* permettent de définir des types complexes propres en extension des types primaires (entier, chaîne, date...).



Syntaxe



Stéréotype datatype



Stéréotype datatype (exemple)

Attributs composés



Les *datatypes* sont utilisés en base de données pour exprimer des attributs composés.
 Cette modélisation est équivalente à une composition 1:1.



Stéréotype (cf. p.18)

2. Transformation de l'imbrication en relationnel

Objectifs

Savoir faire le passage d'un schéma conceptuel UML à un schéma relationnel dans tous les cas.

Reconnaître les cas de transformation qui se traitent toujours de la même façon et ceux qui nécessitent une modélisation complémentaire.

2.1. Transformation des compositions

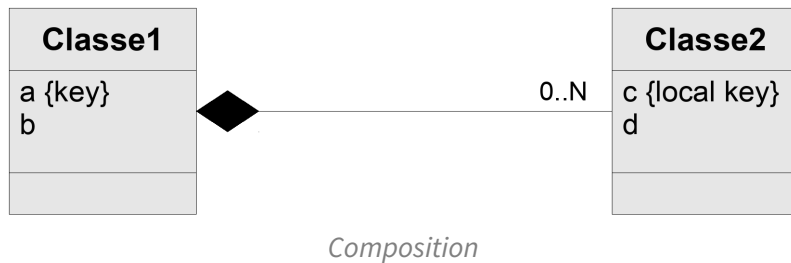
Clé locale

Pour identifier une classe partie dans une composition, on utilise une clé locale concaténée à la clé étrangère vers la classe composite, afin d'exprimer la dépendance entre les deux classes.



Une composition

- est transformée comme une association 1:N,
- puis on combine une clé locale de la classe partie et la clé étrangère vers la classe composite pour construire une clé primaire composée.



Classe1 (#a, b)

Classe2 (#c, #a=>Classe1, d)

Clé candidate



Si une clé candidate (globale) permet d'identifier de façon unique une partie indépendamment du tout, on préférera la conserver comme clé candidate plutôt que de la prendre pour clé primaire.

Si on la choisit comme clé primaire cela revient à avoir transformé la composition en agrégation, en redonnant une vie propre aux objets composants.

Composition et entités faibles en E-A



Une composition est transformée selon les mêmes principes qu'une entité faible en E-A.

Voir aussi



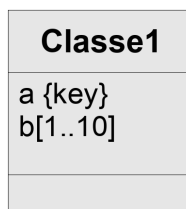
Transformation des attributs (cf. p.19)

2.2. Transformation des attributs multivalués



Pour chaque attribut multivalué b d'une classe C,

- on crée une nouvelle relation RB,
- qui comprend un attribut monovalué correspondant à b,
- plus la clé de la relation représentant C ;
- la clé de RB est la concaténation des deux attributs.



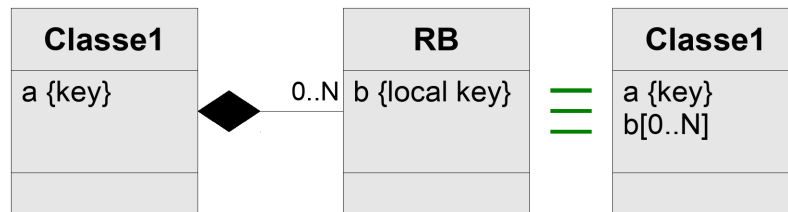
Attribut multivalué

R1 (#a)

RB (#b, #a=>Classe1)



La transformation d'une composition avec un seul attribut pour la classe composante donne un résultat équivalent à la transformation d'un attribut multivalué.



Composition et attribut multivalué

R1 (#a)

RB (#b, #a=>Classe1)

Méthode alternative



Dans le cas où le nombre maximum de *b* est fini, et petit, on peut également adopter la transformation suivante : R1 (#a, b1, b2, b3, b4, b5, b6, b7, b8, b9, b10) .

Si le nombre d'attributs est infini (*b*[1..*]) c'est impossible, s'il est trop grand ce n'est pas souhaitable.

Voir aussi



Transformation des attributs (cf. p.19)

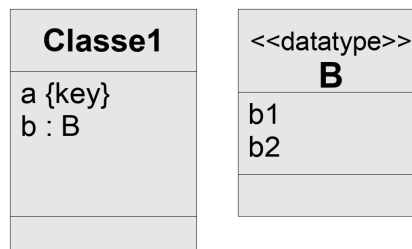
2.3. Transformation des datatypes

Datatype



Pour chaque datatype comprenant N sous-attributs,

- on crée N attributs correspondants,
- dont les noms sont la concaténation du nom de l'attribut composite avec celui du sous-attribut.



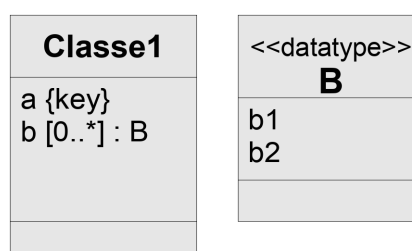
Attribut composé

R1 (#a, b_b1, b_b2)

Datatype multivalué



On combine les règles énoncées pour les attributs composés et pour les attributs multivalués.



Attribut composé multivalué

R1 (#a)

RB (#b_b1, #b_b2, #a=>Classe1)



La transformation des datatype multivalué est similaire à la transformation des compositions.

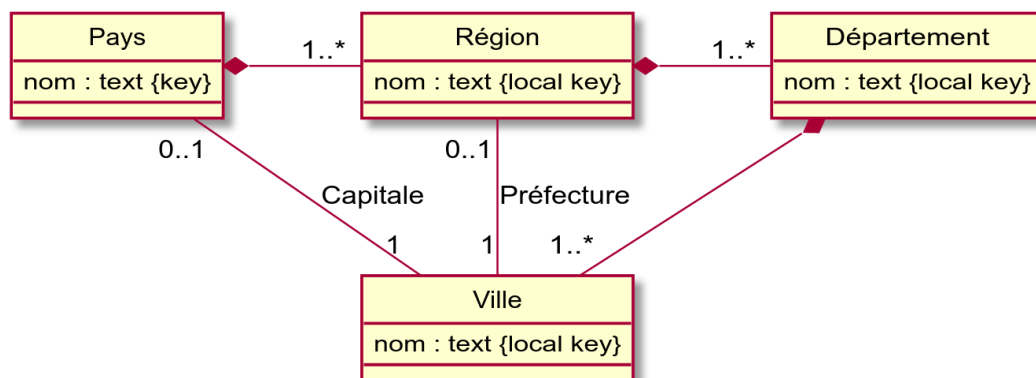
Voir aussi



Transformation des compositions (cf. p.9)

2.4. Exercice : Pays

Soit le schéma UML suivant (on ignore les domaines à des fins de simplification) :



Quels sont les modèles relationnels qui correspondent à ce modèle conceptuel ?

- ☐ Pays(#Nom, Capitale=>Ville)
 Region(#Nom, Prefecture=>Ville, Pays=>Pays)
 Departement(#Nom, Region=>Region)
 Ville(#Nom, Departement=>Departement)
- ☐ Pays(#Nom, Capitale=>Ville)
 Region(#Nom, Prefecture=>Ville, Pays=>Pays)
 Departement(#Nom, Region=>Region, Pays=>Pays)
 Ville(#Nom, Departement=>Departement, Region=>Region, Pays=>Pays)
- ☐ Pays(#Nom)
 Region(#Nom, Pays=>Pays)
 Departement(#Nom, Region=>Region)
 Ville(#Nom, Departement=>Departement, Capitale=>Pays, Prefecture=>Region)
- ☐ Pays(#Nom, Capitale=>Ville)
 Region(#Nom, #Pays=>Pays, Prefecture=>Ville)
 Departement(#Nom, #Region=>Region)
 Ville(#Nom, #Departement=>Departement)
- ☐ Pays(#Nom, CapitaleVille=>Ville, CapitaleDepartement=>Ville, CapitaleRegion=>Ville, CapitalePays=>Ville)
 Region(#Nom, #Pays=>Pays, PrefectureVille=>Ville, PrefectureDepartement=>Ville, PrefectureRegion=>Ville, PrefecturePays=>Ville)
 Departement(#Nom, #Region=>Region, #Pays=>Region)

Ville(#Nom, #Departement=>Departement, #Region=>Departement, #Pays=>Departement)

2.5. Trousseau de clés



Attention

En UML et en relationnel, il existe plusieurs termes mobilisant le mot « clé », le seul concept qui est commun est le concept de **clé**. Tous les autres sont spécifiques au niveau conceptuel ou relationnel.

Concept commun au niveau UML et relationnel



Rappel

Clé (key) *

Concept spécifique au niveau UML



Rappel

Clé locale (local key) *

Concepts spécifiques au niveau relationnel



Rappel

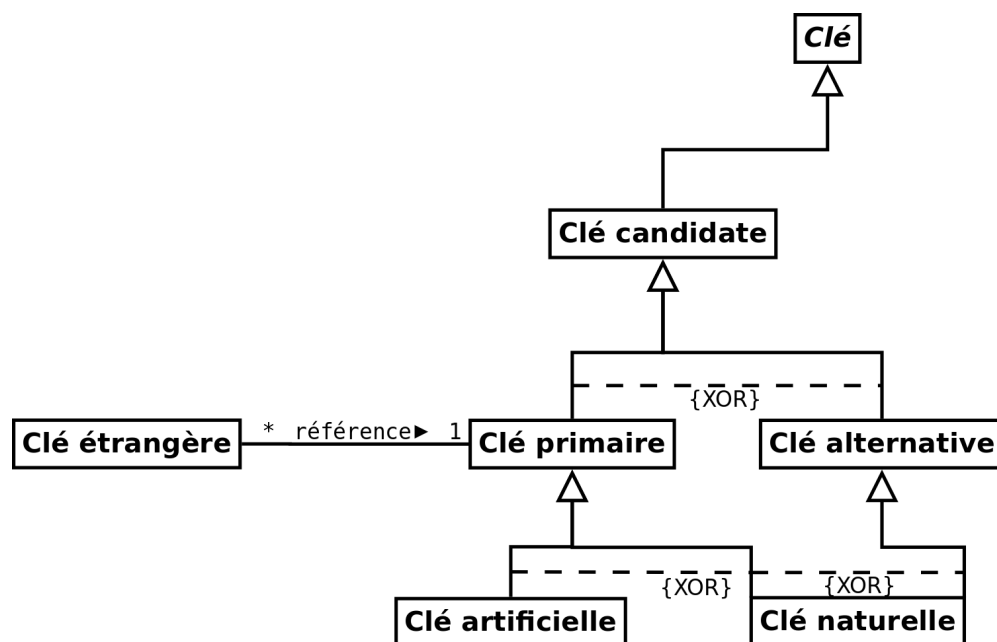
Clé (key) *

Clé candidate (candidate key) *

Clé primaire (primary key) * et Clé alternative (alternate key) *

Clé artificielle (surrogate key) * et Clé naturelle (natural key, business key) *

Clé étrangère (foreign key) *



Définition des clés en relationnel

Exercices



1. Exercice : Objets Numériques Libres

[30 min]

L'association ONL (Objets Numériques Libres) est une association de promotion des logiciels libres. Elle souhaite exposer sur un site Internet une liste de logiciels libres. Ce site sera adossé à une base de données relationnelle ou relationnel-objet. La première étape de sa démarche est de réaliser un modèle conceptuel représentant ce qu'elle souhaite faire.

- La base de données permet de gérer des applications.

Les applications sont identifiées par leur nom (LibreOffice, Gimp...) et leur version (1.0, 2.1), et comportent une description courte et une URL. Tous les attributs sont obligatoires. Chaque application a une URL unique.

- La base de données permet de gérer des librairies.

Les librairies sont des logiciels mais pas des applications. Elles ont les mêmes attributs que les applications (nom, version, description courte, URL), mais les URL ne sont pas nécessairement uniques. Les applications peuvent dépendre de librairies ou d'autres applications, et les librairies peuvent dépendre d'autres librairies (mais pas d'une application).

- La base de données permet de gérer des composants.

Les composants sont intégrés à une application ou librairie. Les composants ont un code interne à l'application ou la librairie qu'il servent, une version et une description courte, et une URL. Le code et le numéro de version permettent d'identifier localement le composant au sein de la librairie ou de l'application, la description courte et l'URL sont optionnelles.

- La base de données permet de gérer des licences.

Les applications, librairies et composants sont attachés à une ou plusieurs licences identifiées par leur nom (GPL, MPL...), leur version et leur langue, et comportant le texte intégral de la licence. Les versions des logiciels et licences sont de type "numéro de licence majeur point numéro de licence mineur", comme "1.0" ou "2.2".

- La base de données permet de gérer des catégories.

Chaque logiciel est rangé dans une catégorie principale, et plusieurs catégories secondaires. Exemple de catégories : bureautique, dessin, multimédia, physique...

Exemple (factice) de données

- Applications :
 - Scenari 4.1, une chaîne éditoriale XML, <http://scenari.org>, dépend de Libreoffice 4.3 et de ImageMagick 6.8
 - Libreoffice 4.3, une suite bureautique WYSIWYG, <http://libreoffice.org>
- Librairie :
 - ImageMagick 6.8, permet de créer, de convertir, de modifier et d'afficher des images, <http://imagemagick.org>

- Composant :
 - impng 0.2 est un composant de ImageMagick 6.8, permet de compresser une image au format PNG.
- Toutes ces applications, librairies et composants sont disponibles sous une licence LGPL 3.0 et GPL 3.0 françaises.
- Toutes ces applications et librairies sont rangées dans la catégorie principale "document". Scenari est rangé dans la catégorie secondaire "Édition WYSIWYM", Libreoffice dans la catégorie secondaire "Bureautique", ImageMagick dans la catégorie secondaire "Multimédia".

Question

Réaliser un MCD en UML.

2. Exercice : Lab III

[30 min]

Description du problème

Un laboratoire souhaite gérer les médicaments qu'il conçoit.

- Un médicament est décrit par un nom, qui permet de l'identifier. En effet il n'existe pas deux médicaments avec le même nom. Un médicament comporte une description courte en français, ainsi qu'une description longue en latin. On gère aussi le conditionnement du médicament, c'est à dire le nombre de pilules par boîte (qui est un nombre entier).
- À chaque médicament on associe une liste dédiée de contre-indications, généralement plusieurs, parfois aucune. Les contre-indications sont triées par ordre d'importance. L'ordre est total et strict pour un médicament, donc chaque contre-indication possède une importance et il n'existe pas deux contre-indications associées au même médicament avec la même importance.
- Tout médicament possède au moins un composant, souvent plusieurs. Un composant est identifié par un code unique et possède un intitulé. Tout composant peut intervenir dans la fabrication de plusieurs médicaments. Il existe des composants qui ne sont pas utilisés pour fabriquer des médicaments et que l'on veut quand même gérer.

Données de test

Afin de matérialiser notre base de données, nous obtenons les descriptions suivantes :

- Le **Chourix** a pour description courte « *Médicament contre la chute des choux* » et pour description longue « *Vivamus fermentum semper porta. Nunc diam velit, adipiscing ut tristique vitae, sagittis vel odio. Maecenas convallis ullamcorper ultricies. Curabitur ornare.* ». Il est conditionné en boîte de 13.

Ses contre-indications sont :

1. Le Chourix ne doit jamais être pris après minuit.
2. Le Chourix ne doit jamais être mis au contact avec de l'eau.

Ses composants sont le **HG79** et le **SN50**.

- Le **Tropas** a pour description courte « *Médicament contre les dysfonctionnements intellectuels* » et pour description longue « *Suspendisse lectus leo, consectetur in tempor sit amet, placerat quis neque. Etiam luctus porttitor lorem, sed suscipit est rutrum non.* ». Il est conditionné en boîte de 42.

Ses contre-indications sont :

1. Le Tropas doit être gardé à l'abri de la lumière du soleil

Son unique composant est le **HG79**.

- Les composants existants sont :
 - **HG79** : "Vif-argent allégé"
 - **HG81** : "Vif-argent alourdi"
 - **SN50** : "Pur étain"

Question 1

Effectuez le modèle conceptuel en UML de ce problème.

Indice :

On note dans l'énoncé que les contre-indications sont **dédiée** aux médicaments, et par ailleurs on note dans les données exemples que les contre-indications sont énoncées spécifiquement par rapport aux médicaments.

Indice :

Ce n'est pas parce que les composants s'appellent ainsi dans l'énoncé que l'on est en présence d'une composition.

Question 2

En mobilisant les règles adéquates, proposer un modèle logique de données correspondant en relationnel. Le repérage des domaines et des clés est obligatoire.

Question 3

Dessiner des tableaux remplis avec les données fournies en exemple, afin de montrer que le modèle fonctionne selon le besoin exprimé initialement. On pourra mettre le premier mot seulement des descriptions pour gagner du temps.

Contenus annexes



1. Agrégation

Association d'agrégation

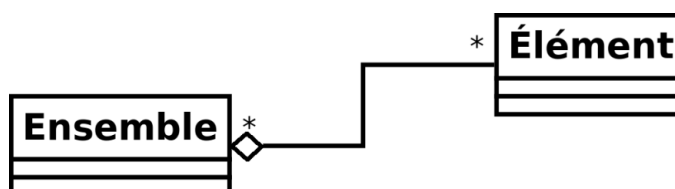


Définition

L'agrégation est une association particulière utilisée pour préciser une relation tout/partie (ou ensemble/élément), on parle d'association **méréologique**.



Syntaxe



Notation de l'agrégation en UML

La cardinalité, peut être exprimée librement, en particulier les instances de la classe **Élément** peuvent être associées à plusieurs instances de la classe **Ensemble**, et même de plusieurs classes.



Attention

L'agrégation garde toutes les propriétés d'une association classique (cardinalité, cycle de vie, etc.), elle ajoute simplement une terminologie un peu plus précise via la notion de tout/partie.

Transformation des agrégations en relationnel



Rappel

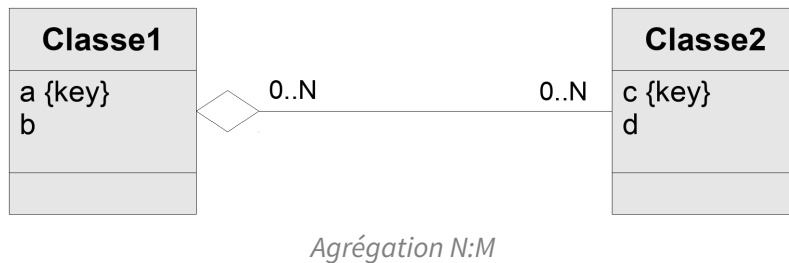
Les associations de type agrégation se traitent de la même façon que les associations classiques.



Agrégation 1:N

Classe1 (#a, b)

Classe2 (#c, d, a=>Classe1)



```
Classe1 (#a, b)
```

```
Classe2 (#c, d)
```

```
Assoc (#a=>Classe1, #c=>Classe2)
```

2. Stéréotype

Stéréotype UML



Définition

Un stéréotype UML est une syntaxe permettant d'ajouter de la sémantique à la modélisation des classes. Il permet de définir des **types de classe**, afin de regrouper conceptuellement un ensemble de classes (à l'instar d'une classe qui permet de regrouper conceptuellement un ensemble d'objets).

C'est une mécanique de méta-modélisation : elle permet d'étendre le méta-modèle UML, c'est à dire le modèle conceptuel du modèle conceptuel.

Méta-modèle



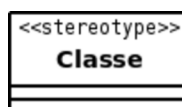
Définition

Un méta-modèle est le modèle d'un modèle. Par exemple le méta-modèle UML comprend les concepts de classe, attribut, association, cardinalité, composition, agrégation, contraintes, annotations, ... On mobilise ces concepts (on les instancie) pour exprimer un modèle particulier suivant le formalisme UML.

Les stéréotypes permettent donc d'ajouter au méta-modèle UML standard, celui que tout le monde utilise, des concepts locaux pour enrichir le langage de modélisation que l'on utilise pour réaliser des modèles.



Syntaxe



Notation d'un stéréotype en UML

Stéréotypes spécifiques et stéréotypes standard



Conseil

Un stéréotype spécifique enrichit le méta-modèle UML, mais selon une sémantique qui est propre à celui qui l'a posé, non standard donc. La conséquence est que pour un tiers, l'interprétation du stéréotype n'est plus normalisée, et sera potentiellement plus facilement erronée. Il convient donc de ne pas abuser de cette mécanique.

Deux ou trois stéréotypes spécifiques, correctement définis, sont faciles à transmettre, plusieurs dizaines représenteraient un nouveau langage complet à apprendre pour le lecteur du modèle.

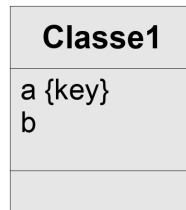
Il existe des stéréotypes fournis en standard par UML, ou communément utilisés par les modélisateurs. L'avantage est qu'il seront compris plus largement, au même titre que le reste du méta-modèle (ils ont une valeur de standard).

3. Transformation des attributs



Pour chaque attribut élémentaire et monovalué d'une classe,

- on crée un attribut correspondant dans la relation correspondant à la classe.



Attribut

Classe1 (#a,b)

Glossaire



Clé (key)

En UML et relationnel on parle de clé pour désigner un groupe d'attributs qui permet d'identifier un objet.

Clé alternative (alternate key)

En relationnel les clés alternatives sont les clés candidates non retenues comme clés primaires.

Clé artificielle (surrogate key)

En relationnel une clé artificielle est un attribut artificiel (qui ne représente aucune donnée) ajouté à la relation pour servir de clé primaire. On mobilise cette technique lorsque l'on n'a pas pu ou voulu choisir une clé naturelle pour clé primaire.

Clé candidate (candidate key)

Clé candidate est un synonyme pour clé, toutes les clés sont candidates. En fait on parle juste de clé au niveau conceptuel. En relationnel on parle de clés candidates pour différencier les clés alternatives (qui n'ont pas été retenues comme clé primaire) de la clé primaire (la clé candidate qui a été élue pour effectuer les références par clé étrangère).

Clé étrangère (foreign key)

Une clé étrangère est en relationnel un ensemble d'attributs qui référence les attributs de la clé primaire de la même relation ou d'une autre relation. La clé étrangère est plutôt mal nommée car ce n'est pas une clé (elle n'identifie pas un objet), mais une référence à une clé.

Clé locale (local key)

Dans certaines constructions au niveau conceptuel (association N:M et composition) la clé peut être locale, c'est à dire qu'au lieu d'identifier pleinement un objet (comme une clé classique), elle identifie un objet étant donné un contexte (les membres de l'association N:M ou l'objet composite).

Une clé locale n'est donc pas une clé au sens relationnel, elle ne permet pas d'identifier un enregistrement, mais elle deviendra une partie d'une clé lors du passage au relationnel.

Clé naturelle (natural key, business key)

Clé naturelle est synonyme de clé au niveau conceptuel (toutes les clés sont naturelles à ce stade). En relationnel, les clés naturelles sont toutes les clés candidates non artificielles.

Clé primaire (primary key)

Une clé primaire est une clé candidate qui a été choisie pour être référencée par les autres relations par le mécanisme des clés étrangères.

Abréviations



E-A : Entité-Association

Index



Association.....	4, 9, 10
Attribut	7, 8, 19, , 11
Composition	4, 9, 10
Conceptuel.....	4, 9
Diagramme	4
Logique.....	9
multivalué.....	7
Passage	9
Propriété.....	7, 8
Relationnel	9, 9, 19, 10, , 11
UML	4, 4, 7, 8, 9, 9, 19, 10, , 11