



**CENTRE NATIONAL D'ORIENTATION ET DE PRÉPARATION AUX
CONCOURS D'ENTRÉE DANS LES GRANDES ÉCOLES ET
FACULTÉS DU CAMEROUN**

Préparation au Concours d'Entrée en Troisième Année de l'ENSP et FGI

Support
de Cours

Uml 2

© COPYRIGHT 2018

Avec Intelligentsia Corporation, Il suffit d'y croire !!!

☐ 698 222 277 / 671 839 797
fb : Intelligentsia Corporation
email : contact@intelligentsia-corporation.com

*" Vous n'êtes pas un
passager sur le train de la
vie, vous êtes l'ingénieur. "*

-- Elly Roselle --

Instructions :

Il est recommandé à chaque étudiant de traiter les exercices de ce recueil (du moins ceux concernés par la séance) avant chaque séance car **le temps ne joue pas en notre faveur.**





I. INTRODUCTION

Au début des années 90, il existait près d'une cinquantaine de méthodes objet. La recherche d'un langage commun unique utilisable par toute méthode objet dans toutes les phases du cycle de vie et compatible avec les techniques de réalisation actuelles a donnée naissance à la méthode UML ((Unified Modeling Language) développée en réponse à l'appel à propositions lancé par l'OMG (Object Management Group) dans le but de définir la notation standard pour la modélisation des applications construites à l'aide d'objets. Elle est héritée de plusieurs autres méthodes telles que OMT (Object Modeling Technique) et OOSE (Object Oriented Software Engineering) et Booch. Les principaux auteurs de la notation UML sont Grady Booch, Ivar Jacobson et Jim Rumbaugh..

II. LA MODÉLISATION AVEC UML

Modéliser un problème avec MERISE revient à mettre sur pied une base de données répondant aux spécifications posées à l'aide d'un MCD et du MLD correspondant. En revanche, UML utilise des schémas et diagrammes pour modéliser les systèmes informatiques. Contrairement à MERISE, il décrit le système, les utilisateurs et les interactions entre ces derniers et le système. La modélisation d'un problème consiste à décrire à l'aide de symboles, de schéma... une solution de ce problème ; c'est trouver un modèle de solution au problème posé. Un modèle doit permettre de :

- ***Mieux comprendre le système à développer***
- ***avoir un "patron" (document et non le boss) pour guider la construction du système***
- ***documenter les décisions qui ont été prises***

© COPYRIGHT
2018

III. LES DIAGRAMMES UML

Il existe en UML 2.0 13 diagrammes regroupés en 2 grandes catégories : Les diagrammes comportementaux qui représentent la partie dynamique du système et qui réagissent aux événements et permettent de produire les résultats attendus par les utilisateurs (diagramme des cas d'utilisation, de séquence, d'état transition, de collaboration ou de communication, d'activité, de temps et le diagramme global d'interaction) et les diagrammes structurels qui représentent l'aspect statique des objets (diagramme de classe, d'objet, de composant, de paquetage, de déploiement et de structure composite).

1. Le diagramme de classes

Le diagramme de classes identifie la structure des classes d'un système, y compris les propriétés et les méthodes de chaque classe. Les diverses relations, telles que les relations d'héritage, de composition... qui peuvent exister entre les classes y sont également représentées.

Le diagramme de classes est le diagramme le plus largement répandu dans les spécifications d'UML. Les éléments d'un diagramme des Classes sont les classes et les relations qui les lient. - Classes

Les classes sont les modules de base de la programmation orientée objet. Une classe regroupe des caractéristiques communes à plusieurs objets. Un objet est une instance (une initialisation) d'une classe. Une classe est représentée en utilisant un rectangle divisé en trois compartiments. Le premier compartiment contient le nom de la classe, qui doit utiliser le vocabulaire du domaine. Les deux autres compartiments contiennent les attributs et les opérations.

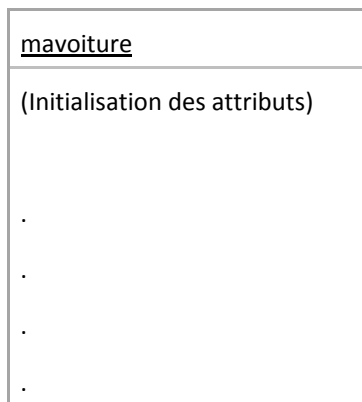


Nom de la classe
Attributs
Opérations

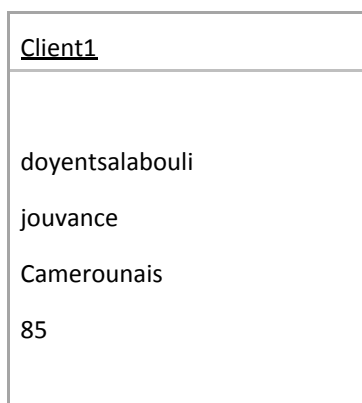
EX :

client	Voiture
Nom : String Aderesse : String Nationalité : String Age: int	Type : String Marque : String Couleur : String
selectionner() Payer-une-commande()	+ démarrer() - rouler() + freiner() # arrêter () +Repeindre(couleur c)

Les objets quant à eux sont représentés ainsi :



Exemple :



Il est utile de préciser que la représentation des objets sera utilisée dans plusieurs autres diagrammes importants d'UML. C'est le cas notamment du diagramme de séquence ou encore du diagramme d'état-transition. *

- attributs

Un attribut est une propriété commune à tous les objets d'une classe. « Nom », « Adresse » et « Nationalité » sont des attributs de la classe « client ». Chaque instance de cette classe possède Un attribut est une propriété commune à tous les objets d'une classe. « Nom », « Adresse » et « Nationalité » sont des attributs de la classe « client ». Chaque instance de cette classe possède ces attributs. Un attribut doit contenir une valeur pure et pas un autre objet. Sinon on préfère utiliser la notion de relation.

- Opérations et méthodes

Une opération est associée à une classe ; c'est une fonction ou une transformation qui peut être appliquée aux objets d'une classe. Par exemple, « Embaucher », « licencier », « Payer » sont des opérations de la classe « Société ». Chaque opération possède un objet cible ou instance sur



lequel elle s'applique. La même opération peut s'appliquer sur plusieurs classes d'objets : on dit qu'elle est polymorphe. Une méthode est l'implémentation d'une opération sur une classe. En programmation orientée objet (Java, C++) on parle de méthodes et en conception orientées objet (OMT, UML), on parle d'opérations.

- Multiplicité

Elle indique un domaine de valeurs pour préciser le nombre d'instance d'une classe vis-à-vis d'une autre classe pour une association donnée. La multiplicité peut aussi être utilisée pour d'autres usages comme par exemple un attribut multivalué.

Le domaine de valeurs est décrit selon plusieurs formes :

"1" signifie un exactement

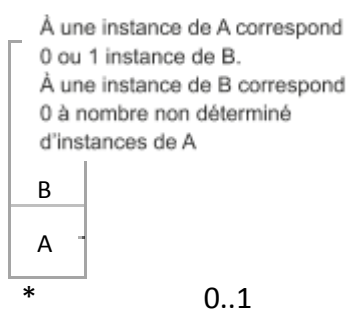
"1..*" signifie de un à plusieurs

"*" et "0..*" signifient de zéro à plusieurs

"3..5" signifie l'intervalle 3 à 5 inclus

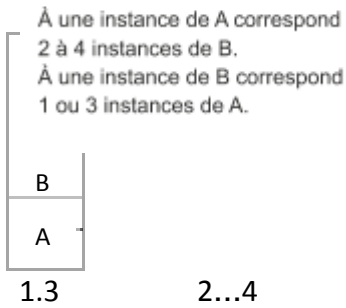
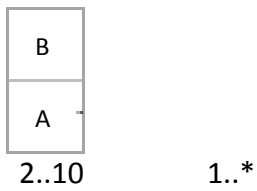
"2, 4, 18" signifie explicitement les valeurs 2, 4, 18

Exemple Explicatif :



© COPYRIGHT 2018

À une instance de A correspond
1 à un nombre non déterminé
d'instances de B.
À une instance de B correspond
2 à 10 instances de A.

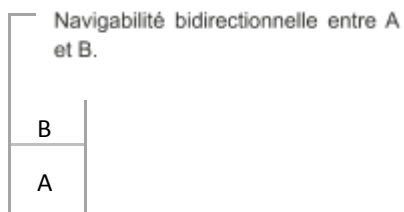
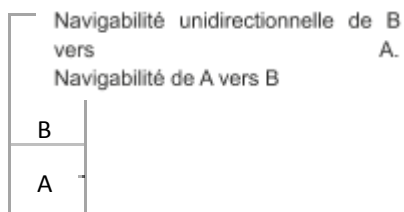
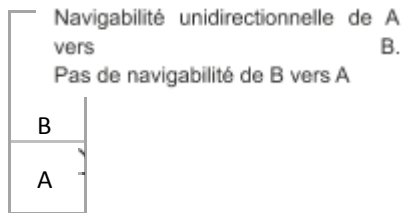


- Navigabilité

Elle indique si l'association fonctionne de manière unidirectionnelle ou bidirectionnelle, elle est matérialisée par une ou deux extrémités fléchées. La non navigabilité se représente par un « X »

Exemple Explicatif :

© COPYRIGHT 2018



© COPYRIGHT 2018

- Visibilité

Public (+) Attribut ou opération visible par tous

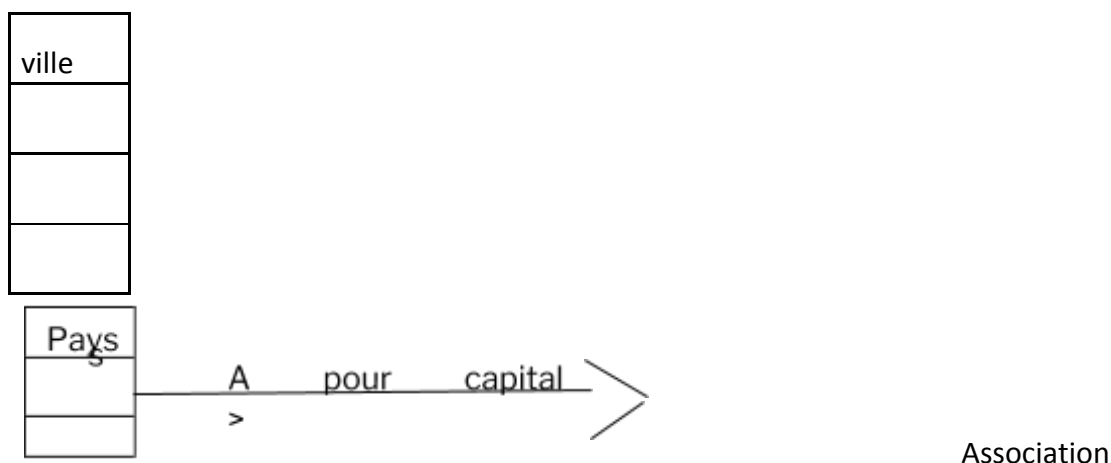
Protégé (#) Attribut ou opération visible seulement à l'intérieur de la classe et par les sous-classes

Privé (-) Attribut ou opération visible seulement à l'intérieur de la classe

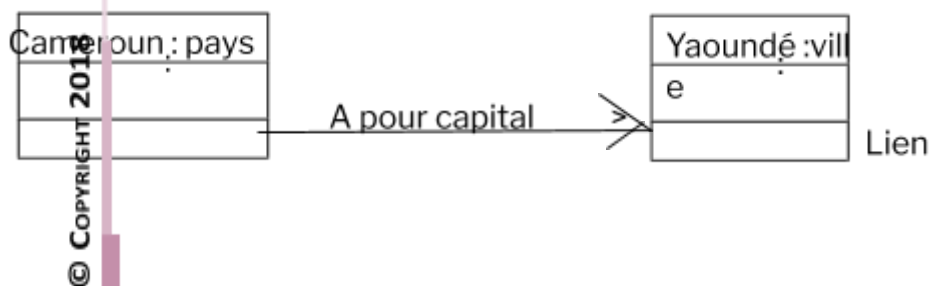
Paquetage (⌞) Attribut ou opération ou classe seulement visible à l'intérieur du paquetage où se trouve la classe

- Liens et associations

Les liens permettent d'établir des relations entre objets (ou instances). Les associations permettent d'établir des relations entre classes.



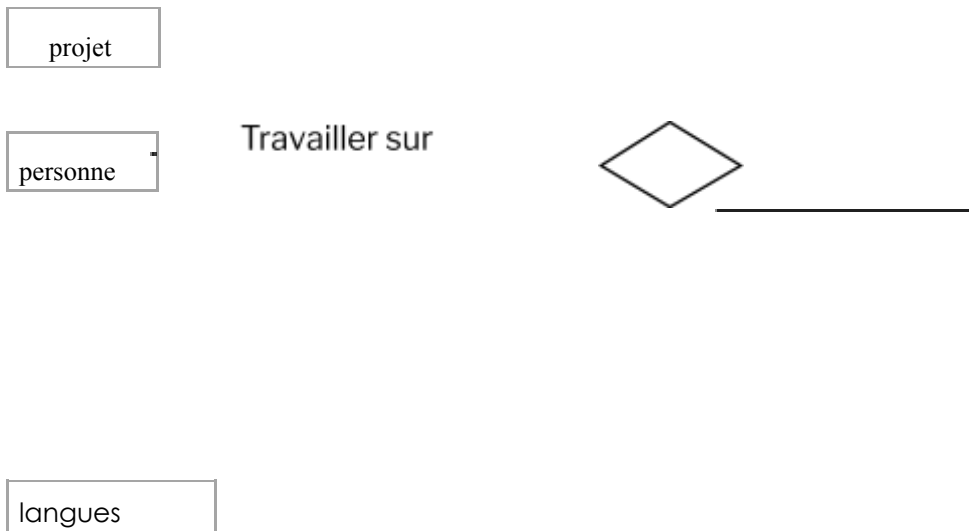
« a-pour-capital » est une association de la classe « Pays » avec la classe « Ville ».





Ex : Un étudiant est identifié par son matricule, son nom, son prénom et son année de naissance. Il peut s'inscrire à plusieurs cours. Un cours est dispensé par un seul enseignant qui peut en dispenser plusieurs. Proposer un diagramme de classe.

Une association peut relier une, deux ou plusieurs classes. Dans le cas où elle relie n classes elle est dite n-aire et dans le cas où elle relie deux classes, elle est dite binaire. Le symbole représentant une association n-aire est le losange. Il est placé au centre de l'association.

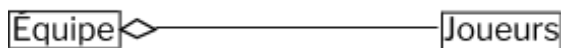


- L'agrégation et composition

L'agrégation est une relation « composé-composant » ou « partie de » dans laquelle les objets représentant les composants d'une chose sont associés à un objet représentant l'assemblage (ou l'agrégation) entier. Il est représenté par un losange vide. La composition est une sorte d'agrégation dans laquelle les composants sont vus comme des attributs pour le composé c'est-à-dire que la destruction du composé entraîne dans ce cas celle des composants ; ce qui n'est pas le cas dans une relation d'agrégation. La composition est symbolisée par un losange plein.

Ex : Une équipe est constituée de plusieurs joueurs

Il est clair que la destruction de l'équipe n'entraîne pas celle des joueurs. Il s'agit d'une relation d'agrégation



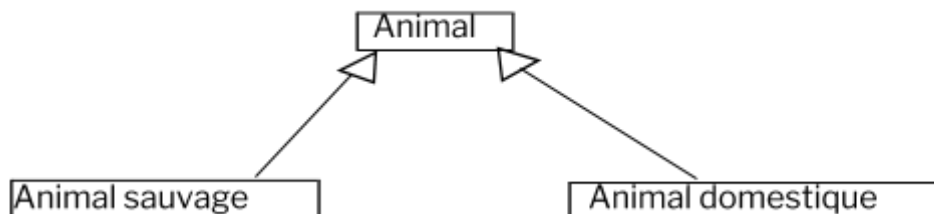
Ex : Un disque dur contient plusieurs plateaux.

Il s'agit ici d'une composition car la destruction du disque entraine évidemment celle des plateaux.



- Généralisation et spécialisation

La généralisation de classes consiste à factoriser dans une classe, appelée super-classe, les attributs et/ou opérations des classes considérées. Elle se représente par un triangle vide. Ex: Il existe 2 catégories d'animaux : les animaux sauvages et les animaux domestiques



La relation inverse de la généralisation est la spécialisation. Dans l'exemple, les classes «Animal sauvage» et «Animal domestique» spécialisent la classe « Animal ». La notation utilisée pour représenter la spécialisation en UML est aussi le triangle. La différence entre généralisation et spécialisation est faible ; ce n'est qu'une question de sens de lecture.

Ex 1 : Modéliser les périphériques clavier et souris

Ex 2 : Un ordinateur est composé d'une unité centrale, de mémoires (RAM, Cache L1, Cache L2), d'un disque dur et des périphériques (clavier, souris, écran, imprimante, scanner, hauts-parleurs). Un ordinateur après sa fabrication est acheté par une seule personne. Donner le diagramme de classe.



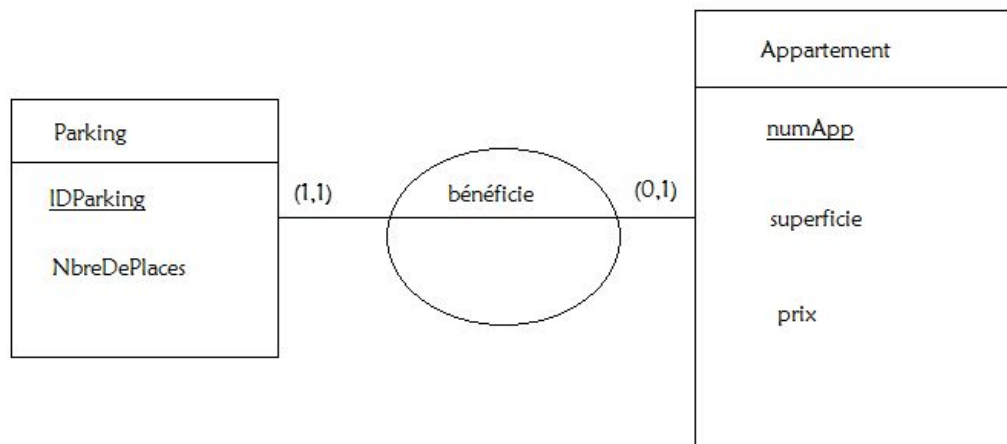
- Passage du MCD au diagramme de classe UML

Règles de passage du MCD au MLD

- 1) Dans une association du type $(*,1)-(*,1)$ avec $*$ =0 ou $*$ =1, on fait migrer de manière indifférente la clé primaire de l'un vers l'autre.

Exercice : Donner une explication concrète et implémenter avec MYSQL.

MCD



MLD

Appartement (numApp,superficie,prix)

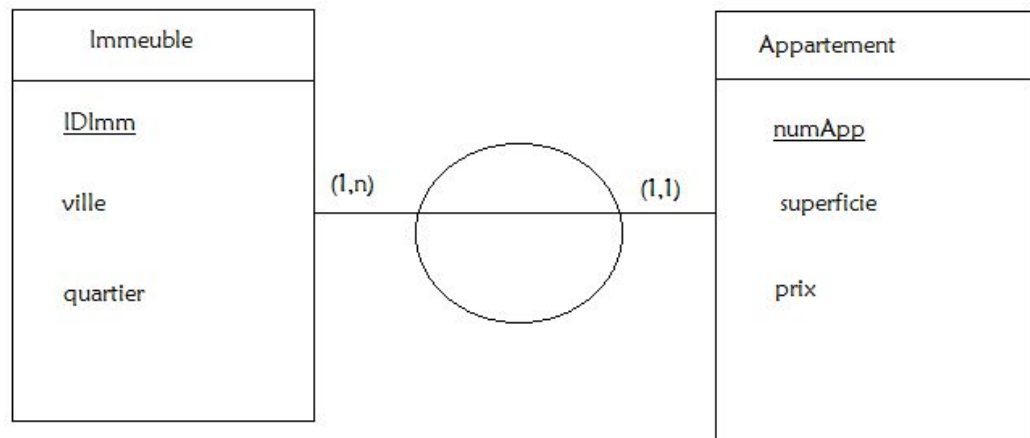
Parking (IDParking,NbreDePlaces,#numApp)

- 2) Dans les associations de type $(*,1)-(*,n)$ avec $*$ =0 ou $*$ =1, on fait migrer la clé de A1 vers A2.

Exercice : Donner une explication concrète et implémenter avec MYSQL.

MCD

© COPYRIGHT 2018

**MLD**

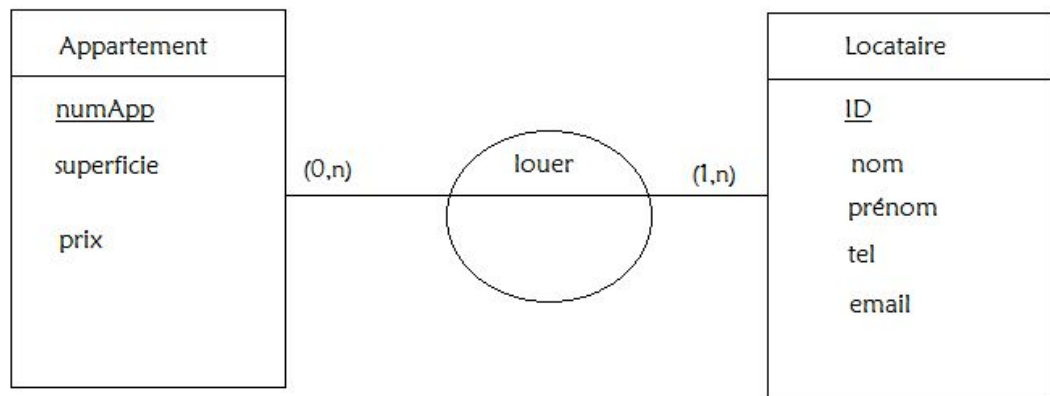
Immeuble (IDImm, ville, quartier)

Appartement (numApp,superficie,prix,#IDImm)

- 3) Dans une association du type $(*,n)-(*,n)$ où $*$ =0 ou $*$ =1, on crée une nouvelle table qui aura pour clé primaire la clé composée de $Id(A1)$ et $Id(A2)$.

Exercice: Donner une explication concrète et implémenter avec MySQL.

MCD

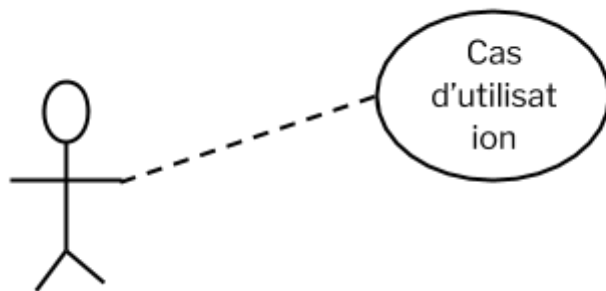


MLD

Appartement (numApp, superficie, prix)

Locataire (ID, nom, prénom, tel, email)

2. Le diagramme des cas d'utilisation



Le diagramme des cas d'utilisation identifie les fonctionnalités fournies par le système (cas d'utilisation), les utilisateurs qui interagissent avec le système (acteurs), et les interactions entre ces derniers. Les cas d'utilisation sont utilisés dans la phase d'analyse pour définir les besoins de "haut niveau" du système. Les objectifs principaux des diagrammes des cas d'utilisation sont:

- fournir une vue de haut-niveau de ce que fait le système
- Identifier les utilisateurs ("acteurs") du système
- Déterminer des secteurs nécessitant des interfaces homme-machine. (IHM)

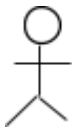


-Représentation Graphique

Les composants de base des diagrammes des cas d'utilisation sont l'acteur, le cas d'utilisation, et l'association.

a) Acteur

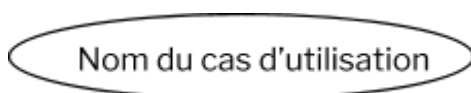
Un acteur est un utilisateur du système, et est représenté par un bonhomme. Le rôle de l'utilisateur est écrit sous l'icône. Les acteurs ne sont pas limités aux humains. Si le système communique avec une autre application, et effectue des entrées/sorties avec elle, alors cette application peut également être considérée comme un acteur



Rôle de l'acteur

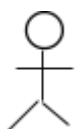
b) Cas d'utilisation

Un cas d'utilisation représente une fonctionnalité fournie par le système, typiquement décrite sous la forme Verbe+objet (par exemple immatriculer voiture, supprimer utilisateur). Les cas d'utilisation sont représentés par une ellipse contenant leur nom.

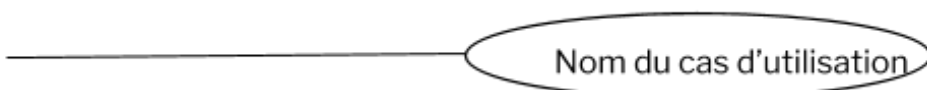


c) Association

Les associations sont utilisées pour lier des acteurs avec des cas d'utilisation. Elles indiquent qu'un acteur participe au cas d'utilisation sous une forme quelconque. Les associations sont représentées par une ligne reliant l'acteur et le cas d'utilisation.



rôle de l'acteur

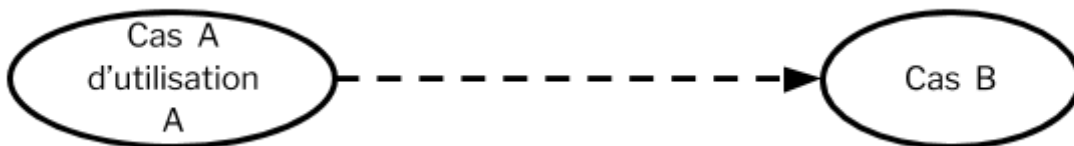


© COPYRIGHT 2018

d) Les relations entre cas d'utilisation

UML définit 3 grands types de relations entre les cas d'utilisation : généralisation/spécialisation, include, extends. Il est important de noter que l'utilisation de ces relations n'est pas primordiale dans la rédaction des cas d'utilisation.

-Uses

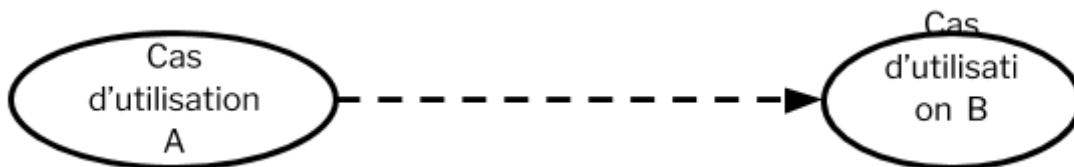


Le **cas d'utilisation A** utilise le **cas d'utilisation B** c'est-à-dire la réalisation de **A** passera toujours par **B**.

« Uses »

NB : une relation d'utilisation entre cas d'utilisation signifie qu'une instance du cas d'utilisation « **source** » comprend également le comportement décrit par le cas d'utilisation « **destination** ».

-extend



« Extends »

Le **cas d'utilisation A** étend le **cas d'utilisation B**.

NB : une relation d'extension entre cas d'utilisation, signifie que le cas d'utilisation source étend le comportement du cas d'utilisation destination.

-Héritage.

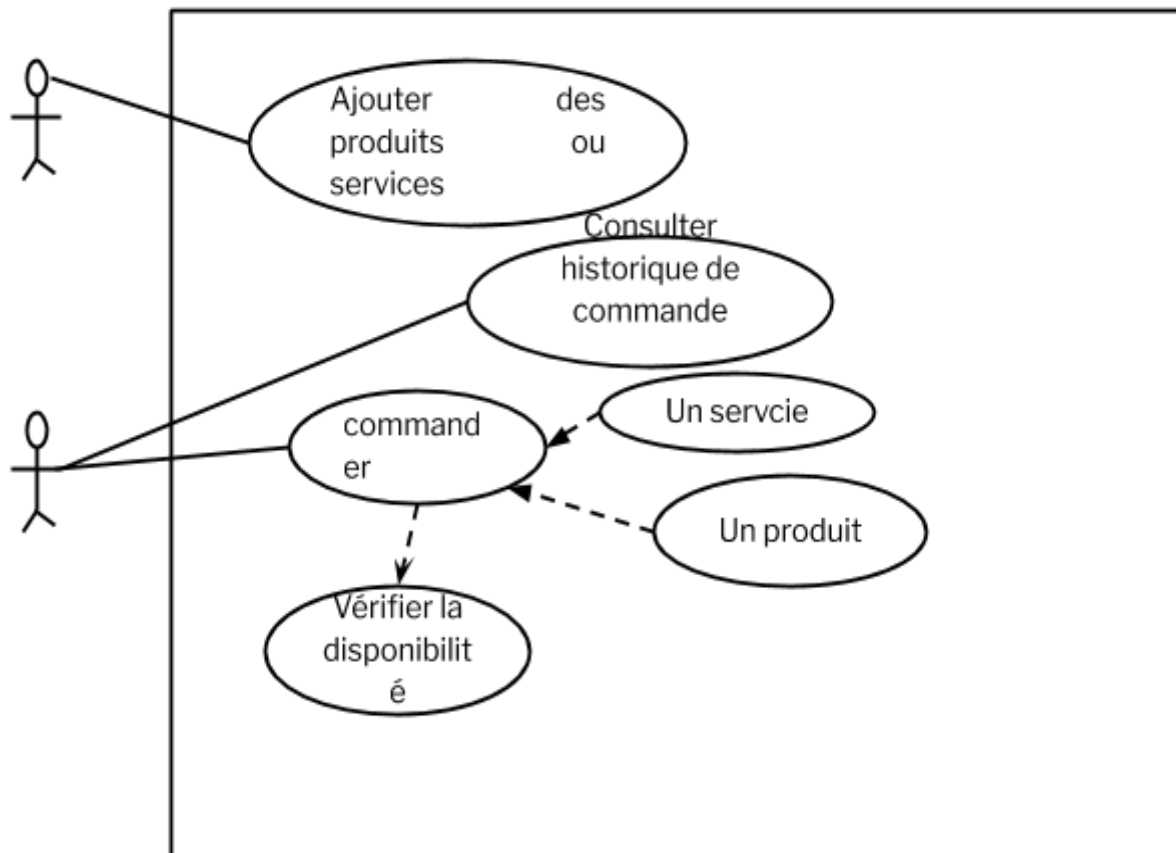
Ça revient tout le sens de la relation « **est un** »



« Est un »

A est une spécification de **B** et **B** est une généralisation de **A**.

Exemple Explicatif :



admin

client

« Uses »

© COPYRIGHT 2018

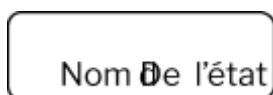
3. Le diagramme d'état transition

Les diagrammes d'état sont utilisés pour documenter les divers modes ("état") qu'un objet peut prendre, et les événements qui causent une transition d'état. Par exemple, un téléviseur peut être éteint (dans l'état "éteint"), et l'action sur le bouton d'alimentation le fait s'allumer (état "allumé"). Une nouvelle action sur le bouton d'alimentation provoque une transition d'état "allumé" vers l'état "éteint". En comparaison avec les autres diagrammes comportementaux qui modélisent les interactions entre des classes multiples, les diagrammes d'état modélisent eux typiquement les transitions d'une seule classe.

-Représentation

a) État

La notation de l'état décrit le mode de l'entité. Elle est représentée par rectangle avec les coins arrondis, contenant le nom de l'état.



b) Transition

Une transition décrit le changement de l'état d'un objet, provoqué par un événement. La notation utilisée pour représenter une transition est une flèche, avec le nom d'événement écrit au-dessus, au-dessous, ou à côté.



c) Etat initial

L'état initial est l'état d'un objet avant toute transition. Pour des objets, ceci pourrait être l'état lors de leur instanciation. L'état initial est représenté par un cercle plein. Un seul état initial est autorisé sur un diagramme.



d) Etat final

L'état final représente la destruction de l'objet que nous modélisons. Ces états sont représentés par des cercles pleins entourés d'un autre cercle.

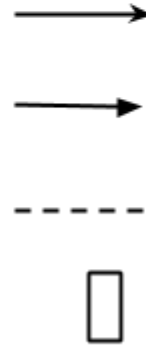
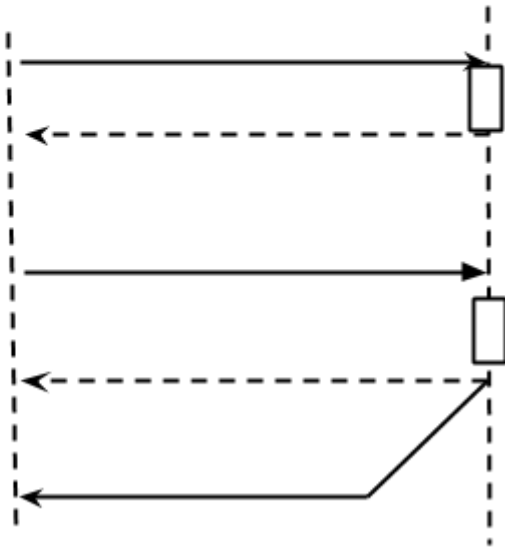
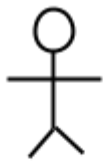


Exercice : Donner le diagramme d'état transition d'un réveil matin.

Indication : Le réveil peut avoir les états (désactivé, programmé, alarmé).

4. Le diagramme de séquences

© COPYRIGHT 2018



Information

Temps

Asynchrone

Synchrone

: User

: Système

NomMessage1()

NomMessageInfo2

NomMessage3()

NomMessage5()

NomMessageInfo4()

Durée de vérification t=1ns

© COPYRIGHT 2018



le diagramme de sequences permet de représenter les interactions entre les objets du système avec un accent sur la chronologie des échanges. Au cours de la modélisation on doit représenter le diagramme de sequence de chaque cas d'utilisation. Ceci induit la recherche des objets qui vont entrer en interaction dans le but de réaliser les cas d'utilisation. Cette interaction doit être représentée soit par un diagramme de sequence soit par un diagramme de communication. Un diagramme de sequence est composé :

- ☐ des lignes de vie généralement en interrompu ;
- ☐ des messages ;
- ☐ Les objets ;
- ☐ La durée du déroulement des opérations

Il serait possible de préciser les contraintes temporelles dans le cas où l'émission d'un message requiert une certaine durée

Exercice : Une bibliothèque gère 3 types de documents (livre, journal, CD-Rom). Chaque document identifié par un numéro, un titre et le nom de l'auteur est disponible dans la bibliothèque en plusieurs exemplaires. Chaque exemplaire est identifié par un numéro unique qui permet de le distinguer des autres. Un abonné de la bibliothèque identifié par un numéro, son nom, son adresse peut être un élève, un étudiant ou un enseignant (Le prix d'abonnement diffère selon le type d'abonné). Un abonné a les avantages suivants :

- Il peut réserver un exemplaire d'un document. Au bout de 7 jours s'il ne passe pas emprunter cet exemplaire ou si aucun exemplaire n'est disponible, sa réservation sera annulée.
- Il peut emprunter un exemplaire d'un document (réservé ou pas). - Il a droit à 3 emprunts au maximum.

Lors d'un emprunt ou d'une réservation on doit enregistrer la date. Lorsque des nouveaux documents arrivent à la bibliothèque, le bibliothécaire se charge de les enregistrer dans la base de données. C'est aussi lui qui enregistre les prêts et les réservations. Il est enfin chargé d'archiver les documents ayant passé plus de 10 ans sans être consultés.

- 1- Donner le diagramme de classe
- 2- Etablir le diagramme des cas d'utilisation
- 3- Donner le diagramme d'état transition de l'exemplaire d'un document

