UML

|  |  |
| --- | --- |
| Diagramme de classe | DLC |
| Diagramme d’objet | DOB |
| Diagramme d’etat transition | DET |
| Diagramme d’activité | DAC |
| Diagramme de collaboration | DCO |
| Diagramme de cas d’utilisation | DCU |
| Diagramme de sequences | DES |
| Diagramme de composent | DCP |
| Diagramme de déploiement | DDP |

Schéma synoptique :

IMPLEMENTATION

Mise en place des

Connection Interface/ serceur

Test et installation

Niveau Fonctionnel

DCU

DSE

Interaction

Les composants de l’application DCP

DDP Mise en production

Niveau Physique

Niveau Dynamique

DET

DAC

Niveau Logique

DCL

DOB

DCO

Diagramme des cas d’utilisation DCU

Définition :

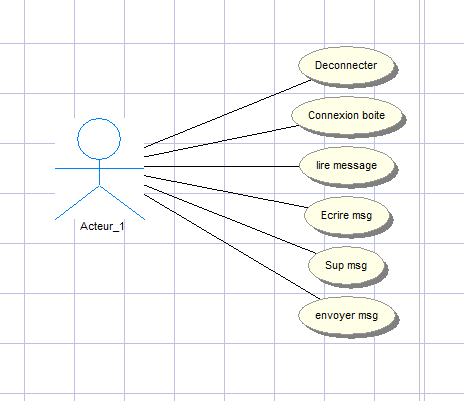
Un DCU permet de représenter le niveau fonctionnel d’une application. Son rôle est de représenter les différentes interfaces et le rôle des utilisateurs vis-à-vis de ses interfaces.

Les notions essentielles d’un DCU

Cas d’utilisation formalisme

C’est un évènement réaliser soit par une procédure, une fonction, un trigger, une méthode ou un script et il peut être la combinaison de procédure et de fonction ou un ensemble de script etc...

Un cas d’utilisation peut être composer de plusieurs sous cas avec dépendance ou sans dépendance



Acteur formalisme :

Un acteur peut être interne ou externe au SI qui a une ou plusieurs interactions avec un ou plusieurs cas d’utilisation selon le rôle qu’on lui a attribuer en tant qu’utilisateur. Quand plusieurs utilisateurs sont en association ou en interaction avec un cas d’utilisation ont les appelles rôles.

Rôle : Il représente un ensemble de privilège dédiés a un groupe d’utilisateurs

Il existe parfois des cas ou deux acteurs sont en interaction avec le même cas d’utilisation mais qui n’ont pas le même rôle.

On peut appliquer la notion de généralisation, spécialisation (héritage) sur les acteurs. Une association représente le lien entre l’acteur et le cas d’utilisation.

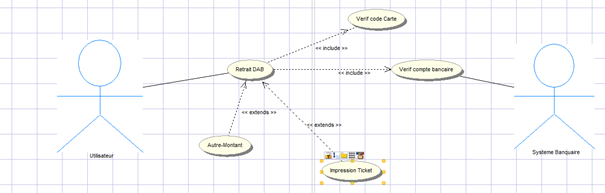
Une multiplicité exprime le nombre de participation d’un instance acteur vis-à-vis des instances des cas d’utilisation. Combien de fois l’utilisateur x peut se connecter sur hyperplanning.

Remarque : Il existe des associations qu’on appelle dépendance.

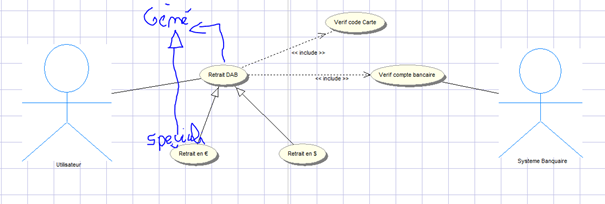
Les différentes contraintes de la DCU

Une contrainte d’inclusion exprime le fait qu’un cas d’utilisation ne peut être valider que si et seulement s’il a été vérifier par un ou plusieurs sous cas au cas quel il dépends. Mr X ne peut se connecter que si et seulement s’il a rentré les bons identifiants.

La contrainte d’extension <extends> exprime le fait qu’un cas d’utilisation peut être étendu a plusieurs sous cas qui reste optionnel.



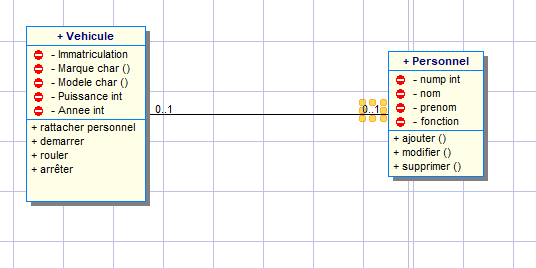
La contrainte de spécialisation généralisation sur les cas d’utilisation exprime le concept de l’héritage.



**Diagramme de classe**

1. Présentation :

Un diagramme de classe est composé d’un ensemble de classe ayant un intérêt relationnel et structurel. Une classe est compose D’un ensemble d attribut et de méthode qui la caractérise chaque classe possède un identifiant physique qu’on appelle le IOD.



1. Les notions essentielles d’un DCL
2. Attributs

**Attribut atomique** : est un attribut élémentaire qu’on ne peut pas décomposer

**Attribut statique** est un attribut qui prend une et une seule valeur et qui peut être utiliser en dehors de sa classe.

**Attribut dérivé** est un attribut calculé par une fonction arithmétique ou une fonction mathématique

**Attribut concaténé** est un attribut composé d’au moins deux ou plusieurs attributs atomiques

1. Visibilité

Elle permet de représenter le type d’accès aux classes, attributs et méthodes

(+) Public : visible par tous

(-) Private : visible que par les membre de la classe

(#) Protected : visible par la classe fille et mère

1. Associations
2. Elle permet de représenter les relations qui existent entre un ou plusieurs classes. Il existe plusieurs types d’association.
3. Association unidirectionnelle : Exemple d’application

Dans un centre d’examen un candidat doit produire 5 copies d’examens

Un association unidirectionnel uni une dépendance forte entre deux classes cad qu’une classe ne peut existe que ssi ne peut être générer que par une autre classe.

1. Association bidirectionnelle
2. Agrégation exprime la notion de classe composer et de classe composant, il existe deux types d’agrégation :

* Une agrégation de composition exprime le fait qu’en supprimant une instance de la classe composer ont supprimer automatiquement toutes les instances de la classe composant relative à la classe composée.
* Une agrégation simple exprime le fait qu’en supprimant une instance de la classe composée ont ne supprime pas forcement les instances de la classe composant.