**Cours du 8 janvier**

**UML .**

Professeur : B. Chauvet

Géré par consortium OMG (Object Management Group)

UML (Unified Modeling Language)  
UML1 : 1995  
UML 2 : 2004

Format xmi : import / export au format xml (notamment pour pouvoir ouvrir sur StarUML un doc écrit sur un autre modeliseur)

Modélisation :

Voir [ici](http://lipn.univ-paris13.fr/~gerard/uml-s2/uml-cours04.html) cours a l’air pô mal  
flèches en pointillé « include » (comprendre « obligatoire »)  
autre flèche pointillé « extends » vient étendre la fonctionnalité principlale,  
Flèche dans l’autre sens, on est dans du facultatif.

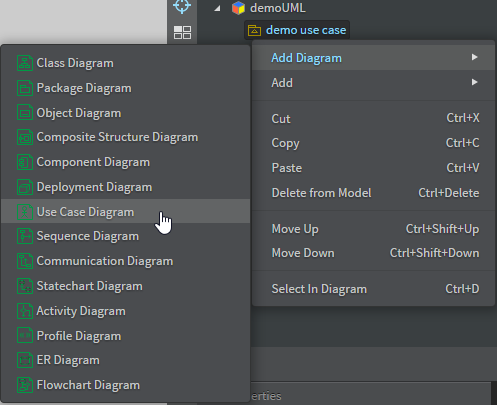
Il y a des conditions d’extension notées [ guarde ].

Extension points : à quel **moment** je dois exécuter le extends.

Héritage multiple est possible pour les acteurs.

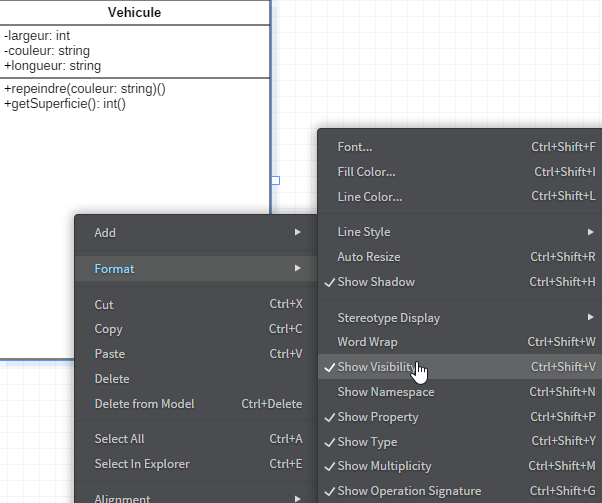
On représente l’héritage par des flèches pleines, attention au sens : elles partent de celui qui « récupère » les droits

Création d’un useCase Diagram



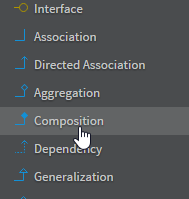
Attention : supprimer avec SUPPR ne supprime que du diagramme : l’objet/la connexion reste dans le modèle !  
On utilise donc **CTRL + SUPP** pour supprimer (raccourcis de delete from model)

2TUP : two tracks unified process



Pour les associations :





Lien du côté du composé, permet de lier par exemple roues a voiture.

Scénario nominal : qui se déroule quand tout va bien.  
Enchainement alternatif : scénario prévu qui dévie du nominal, et retour au principal au bout d’un moment.

Dans les enchainements alternatifs, on fait un **cadre**

Diagramme de séquence, cours Sophia Antopolis [clic](http://mbf-iut.i3s.unice.fr/lib/exe/fetch.php?media=2012_2013:s2:umlsequences.pdf)

Pour ajout d’un opérande double clic sur le frame

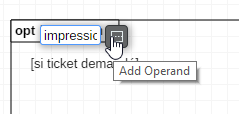
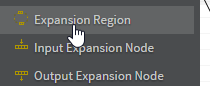
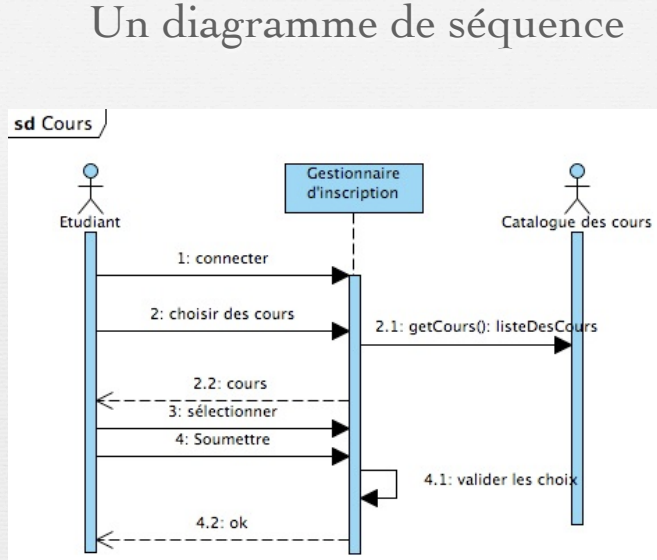


Diagramme d’activité : Process inscription  
Recherche d’un stage et inscription

Interruptible activity region : sortie du chemin. Assortie d’un signal, un événement.

Send signal / acccept signal.





**Diagramme de package (namespace) :**   
  
Il peut contenir d’autres packages, et des relations entre les packages (dépendances).

L’idée est de pouvoir changer si besoin facilement une donnée qui changerait, par exemple un fournisseur de service (en ne touchant qu’au package adéquat)

Diagramme de composants & diagramme de

Couche présentation (IHM), importante dans le final.

L’application doit savoir lire et écrire dans la bdd : accès aux données.

La couche présentation s’adresse à la couche business (métier), qui s’adresse à la couche DataAccess : découpage, chacun son rôle.

« Une commande ne doit pas dépasser les 5.000 € » : pas le rôle du DAO, mais le rôle de la couche métier.

La couche d’accès aux données a les opérations simples (CRUD create read update delete)

.ear : enterprise archive

On utilise les interfaces pour permettre une modularité du code, une adaptation au changement potentiel.  
Ex : Media Center qui permet de lire toute une playlist,  
On utilise l’interface Media avec ses propres méthodes, redéfinies pour les classes MP3, WMV, JPEG… Qui ont leur comportement propre.

Il faut éviter les couplages forts !  
On utilise ainsi les interfaces, rapides à écrire ; si l’on veut évoluer dans le futur ça facilite les choses (+gain de temps).

On cherche à diminuer les impacts des changements.

**Cours 12 mars 2018**

Professeur : Didier Defrance

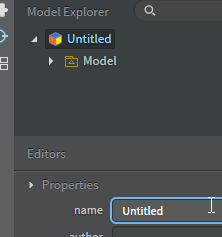
Modéliser les principales interactions, les acteurs, les relations n-n ou quoi :

réalisables facilement, et très rentable.

Include : à faire en priorité .

Extend : non prioritaire (si le client a demandé à imprimer un ticket, facultatif)

Renommer le diagramme à l’ouverture de SUML.

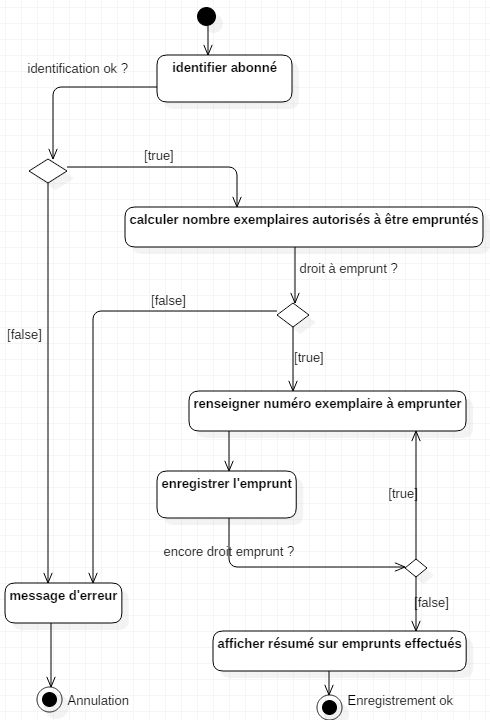


Use case :

1. Préconditions
2. Scénario : scénario nominal (et éventuellement d’exception)
3. Règles de gestion explicitées
4. Description : titre, résumé, préciser les acteurs, conditions (1).

Use case : enregistrer début emprunt

**Préconditions** : la personne souhaitant emprunter est déjà abonnée et a le droit d’emprunter (nombre de livres empruntés < nombre max possible + abonnement valide)  
**scénario** : renseigner le numéro d’abonné et vérifier la validité de l’abonnement, retourner le numéro d’abonné identifié ou exception.  
**règles** **de** **gestion** **explicitées** :   
**description** :



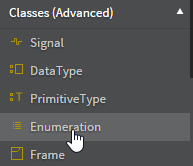
Création d’un glossaire :

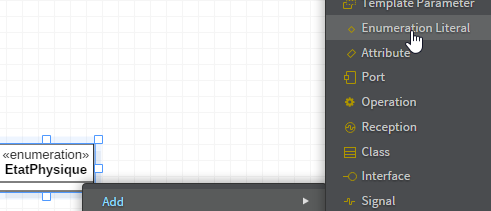
|  |  |  |
| --- | --- | --- |
| **Classes (entités)** | **Définitions** | **Caractéristiques** |
| Exemplaire (livre) | Exemplaire physique d’une œuvre littéraire | N° (id), lien avec œuvre littéraire, etatPhysique (Bon, Abime, HS), disponible (bool) |
| Œuvre littéraire | Œuvre intellectuelle (description abstraite) | Tite, id (ex :isbn), auteur, éditeur, date de premiere sortie, … |
| Emprunt | Information sur un emprunt (en cours ou passé) effectué par un abonné | Date de début, date de fin, n° d’emprunt, lien avec abonné emprunteur, lien avec exemplaire emprunté |
| Abonné | Personne inscrite à la bibliothèque qui peut emprunter des livres sur une période donnée | N°, nom, prénom, adresse, DoB, email, tel |
| Contentieux | Contentieux entre Bibliothèque & abonné (ex : retard important, dégradation) dans le cadre d’un emprunt | N°, date, motif, détails |
| Réservation | V2 (plus tard) |  |

Package fonctionnels & dépendances :

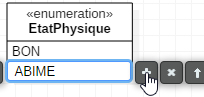
livres (Œuvre, Exemplaire)  
personnes (Abonné, Personne, …)  
emprunts (Emprunt, Réservation, Contentieux …)  
emprunts dépend de livres et de personnes.

En UML :  
- contraintes sont rares, et mises entre accolades  
- en italique les abstractions  
- les dépendances flèches pointillées  
- les attributs avec un + avant sont publics, ceux avec un – sont privés  
- l’héritage est représenté par des flèches pleines, et flèches « entieres »  
- fleche auto (reflexive) 🡪 par exemple de employé à employé (supérieur hiérarchique) ou terrain à terrain (un terrain contenant des terrains)

Enumération en StarUML :  
 On add ensuite des valeurs :

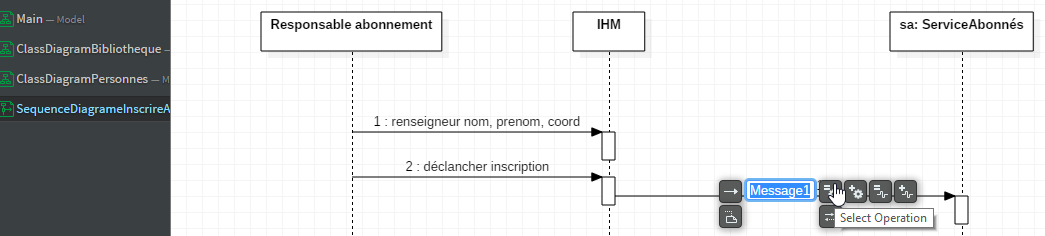


Puis bouton + :



On fait des glisser-déposer pour créer les diagrammes adéquats, si l’entité a déjà été décrite.

Quant au diagramme de séquence :



Si la fonction existe déjà, on va la chercher avec le bouton +.