

UML

Table des matières

Définition.....	2
Historique.....	2
Dans quel contexte utiliser UML ?.....	2
Que peut-on représenter avec UML ?.....	2
Méthode UML ?.....	3
Les besoins utilisateurs.....	3
Vocabulaire.....	3
Les symboles d'UML.....	4
Les flèches d'UML.....	4
Relations.....	4
Messages.....	4
Les principaux diagrammes d'UML.....	5
Le diagramme des cas d'utilisation.....	6
Le diagramme de packages.....	7
Le diagramme d'activité.....	8
Le diagramme de séquence.....	9
Le diagramme de classes.....	10
Le diagramme d'état-transition.....	11
Le diagramme de communication.....	12
Outils.....	13
StarUml.....	13
BoUml.....	13
PlantUML.....	13
Autres.....	14
Approche MDA.....	14
Références.....	14
Mises en application.....	14

Définition

Acronyme de « Unified Modeling language », langage de modélisation unifié, UML est donc un langage de modélisation : un langage visuel permettant de modéliser un système logiciel. Et non pas une méthode.

Bien qu'un modèle ne représente pas une réalité absolue, un modèle reflète les aspects importants de la réalité, il en donne donc une vue juste et pertinente.

UML a pour fonction d'être un langage commun, visuel facilitant la communication entre les participants d'un projet.

Historique

Née au milieu des années 1990, UML est le fruit de la fusion des trois principales méthodes de modélisation objet (OMT, Booch et OOSE).

On en comptait des dizaines !!

C'est en 1997 qu'UML devient le standard de l'OMG (Object Management Group)

Dans quel contexte utiliser UML ?

L'on mettra en œuvre UML dans l'étape de conception de projets adoptant une approche objet.

UML permettra de représenter graphiquement des concepts abstraits,
limiter les ambiguïtés (parler un langage commun, au vocabulaire précis, indépendant des langages orientés objet),
faciliter l'analyse (simplifier la comparaison et l'évaluation de solutions).

Il est donc important de « penser objet » pour modéliser avec UML, sans toutefois aller jusqu'à penser en Java ou en d'autres langages.

L'on pourra utiliser UML à la fois dans des projets agiles et dans des projets « cycle en V ».

Que peut-on représenter avec UML ?

Avec UML, il est possible de représenter un système via

- des acteurs
- des activités
- des processus
- des composants logiciels

Il existe un diagramme spécifique pour représenter des bases de données

Méthode UML ?

Parmi les différentes méthodes utilisant UML, nous allons voir puis mettre en œuvre « Unified Process », une méthode itérative et incrémentale basée sur les cas d'utilisations.

Les besoins utilisateurs

Au commencement (du projet) étaient les besoins utilisateurs.

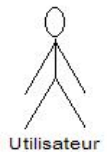
Source de toute activité informatique, les besoins des utilisateurs sont à la racine du déroulement de notre méthode UML.

La première étape est de les recenser, ils donneront lieu à des « cas d'utilisation » (use case).

Vocabulaire

Acteurs

Ce sont des entités externes qui agissent sur le système.
On nommera acteur les différents rôles à prendre en compte dans le projet.



En 2001, Miller identifie quatre catégories d'acteurs :

- Initiateur : acteur qui active le système et déclenche le cas.
- Serveur : acteur aidant le système à assumer ses responsabilités.
- Receveur : acteur recevant les informations du système (système de backup)
- Facilitateur : acteur dont les actions sont effectuées au bénéfice d'un autre acteur

Use-Case (UC)

Cas d'utilisation. Basé sur la notion d'objectif d'un acteur car un acteur utilisera le système pour un/des objectifs précis.

Cet objectif est atteint à la suite d'interactions de l'acteur avec le système.

Un UC décrit ces interactions : un service que rend le projet/le système

Packages

Ou paquets. Servent à regrouper les cas d'utilisation, les classes

Itérations

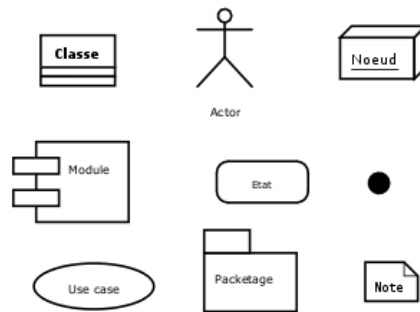
Ou phases.

Aucun diagramme ne sera parfait du premier coup.

Tout ne sera pas modélisé au premier jet non plus

De ce fait, la vie du projet sera divisée en itérations.

Les symboles d'UML



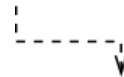
Un nœud pourra représenter un serveur, il contiendra un/des modules tel qu'une JVM par exemple

Les flèches d'UML

Les flèches servent à indiquer des relations, des messages ou des créations d'objet

Relations

Dépendance : à des bibliothèques, à des étapes



Association d'aggrégation : les objets agrégés ne sont pas liés au cycle de vie de l'objet qui les contient



Association de composition : relation forte au cycle de vie
Exemple : Si l'on détruit une voiture, le moteur l'est aussi



Messages

Asynchrone : la source envoie le message puis continue sa propre exécution



Synchrone : la source attend le message de retour

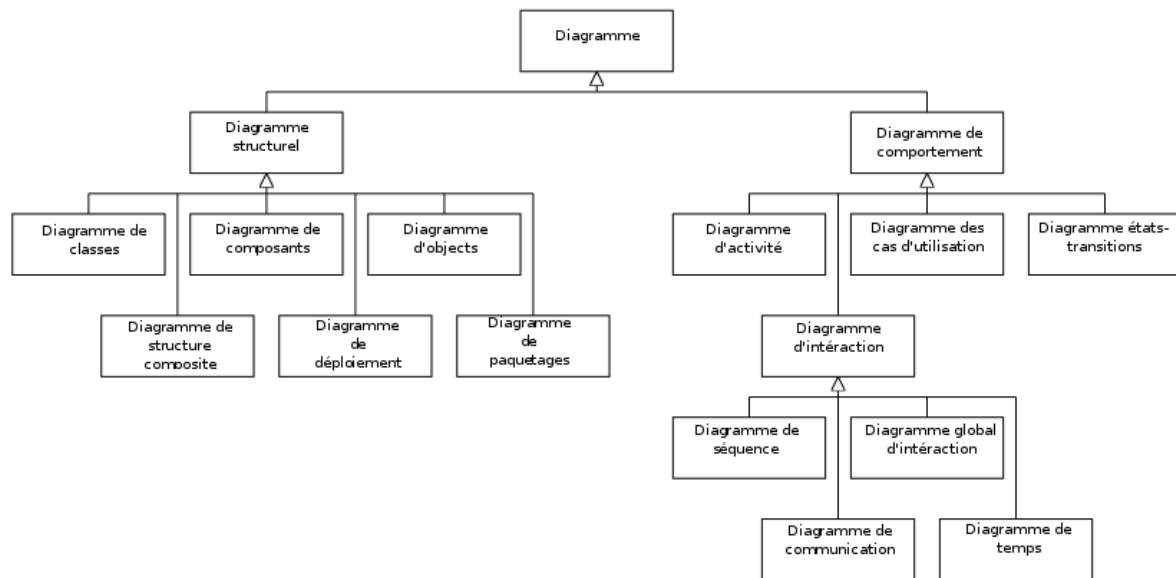


La création d'objet est une flèche en pointillés, à tête ouverte (comme le message asynchrone)

Les principaux diagrammes d'UML

Les diagrammes se complètent, de façon à permettre la modélisation d'un projet tout au long de son cycle de vie. Il en existe quatorze depuis UML 2.3. Auparavant, il y en avait 25.

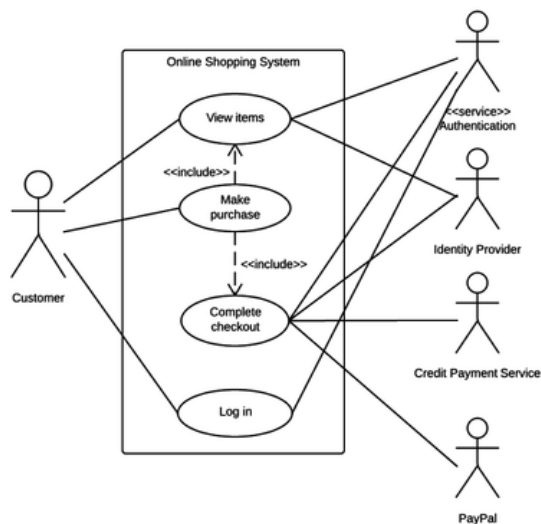
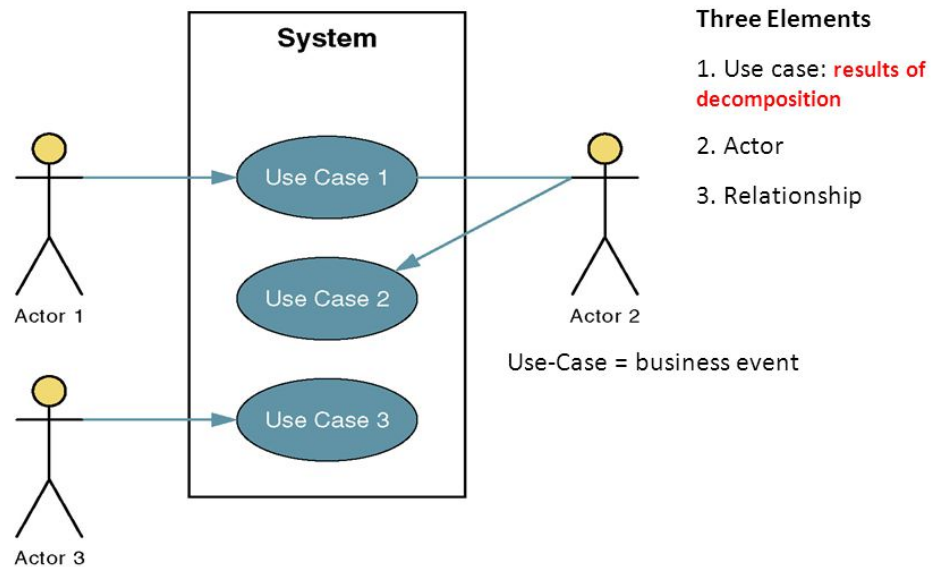
Voici comment ils se repartissent. Selon qu'ils sont structurels ou de comportement.



Le diagramme des cas d'utilisation

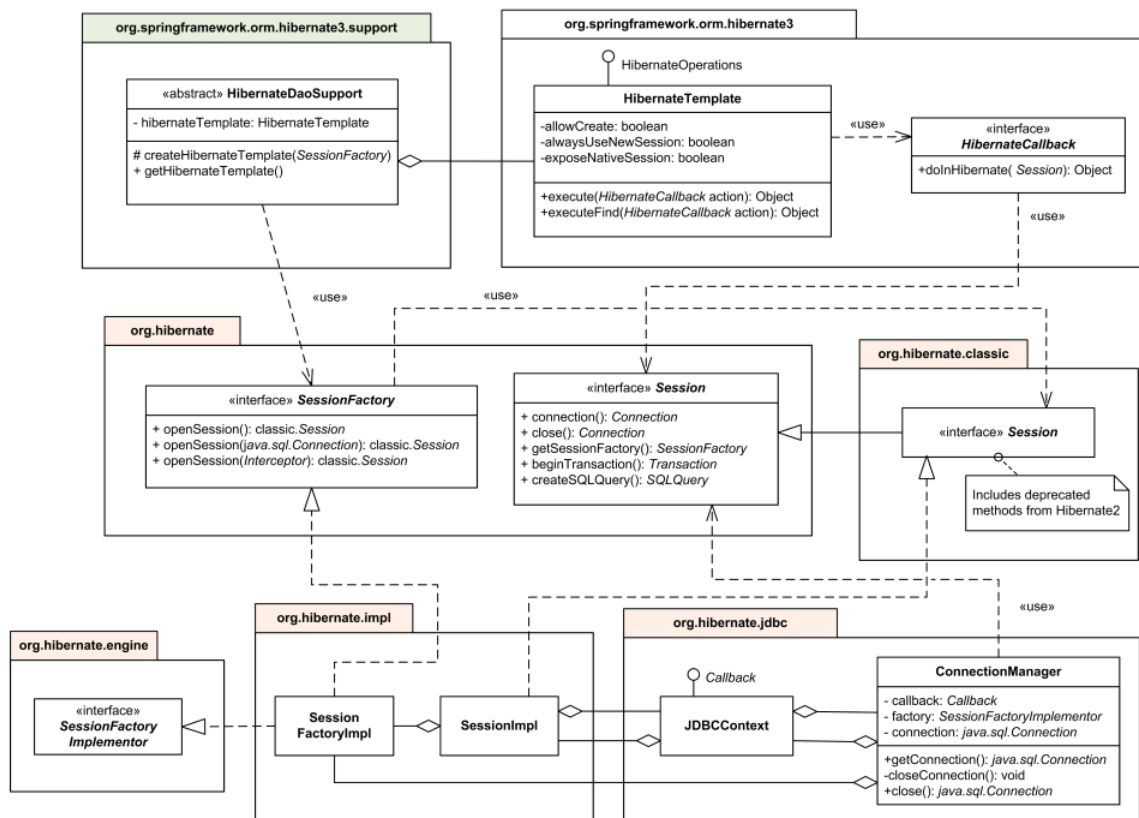
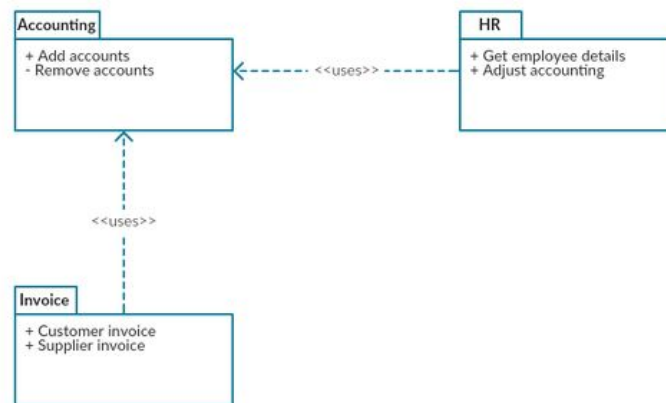
Permet d'indiquer les acteurs impliqués dans les CU

Sample Use-Case Diagram



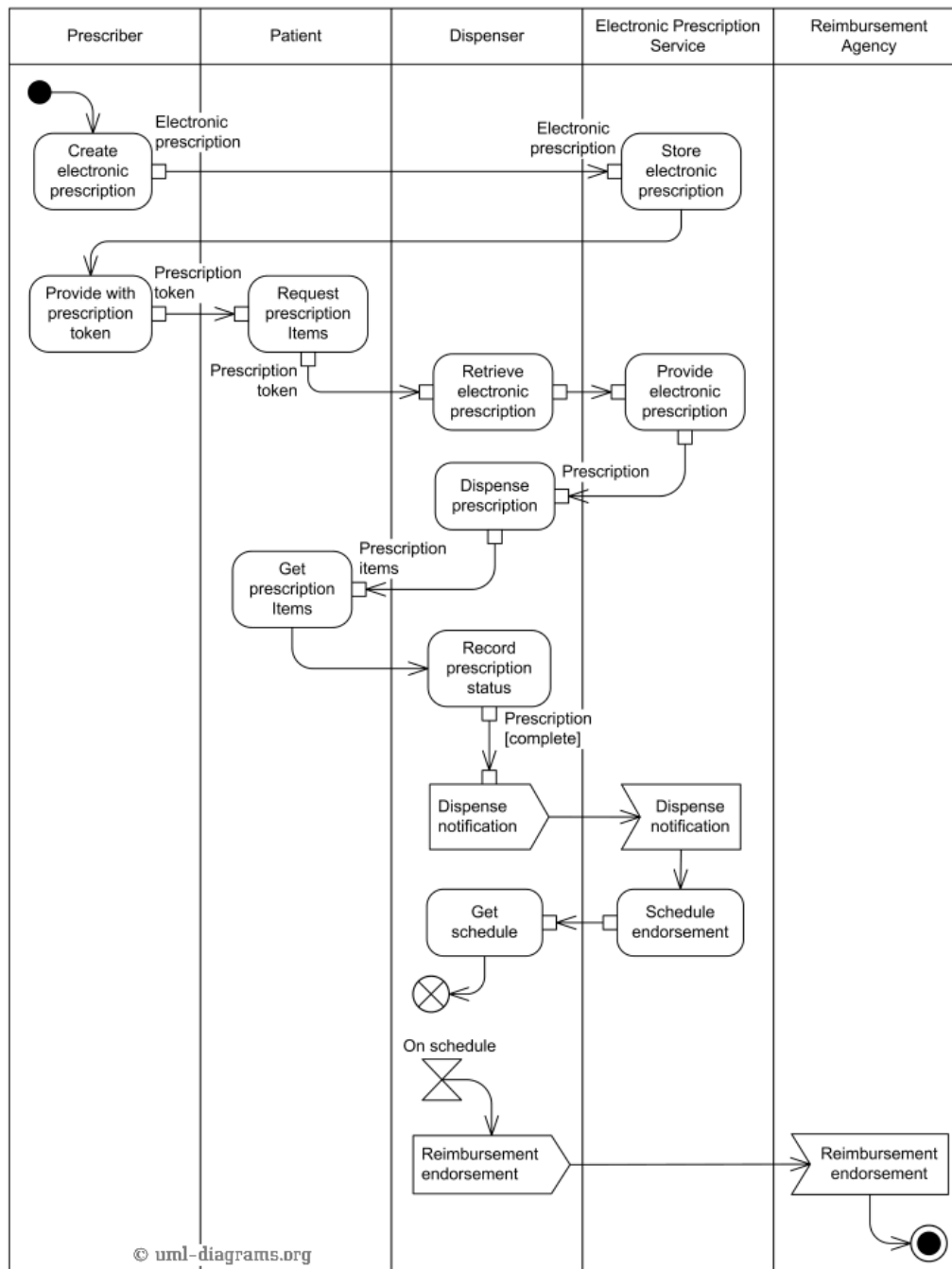
Le diagramme de packages

Un paquet contient des éléments. L'on pourra faire des diagrammes de packages pour les cas d'utilisation ET des diagrammes de packages pour les classes.



Le diagramme d'activité

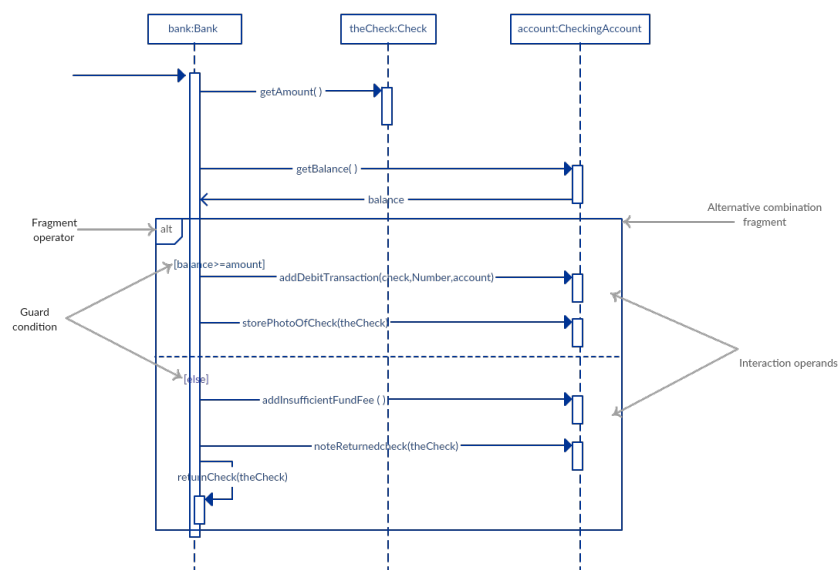
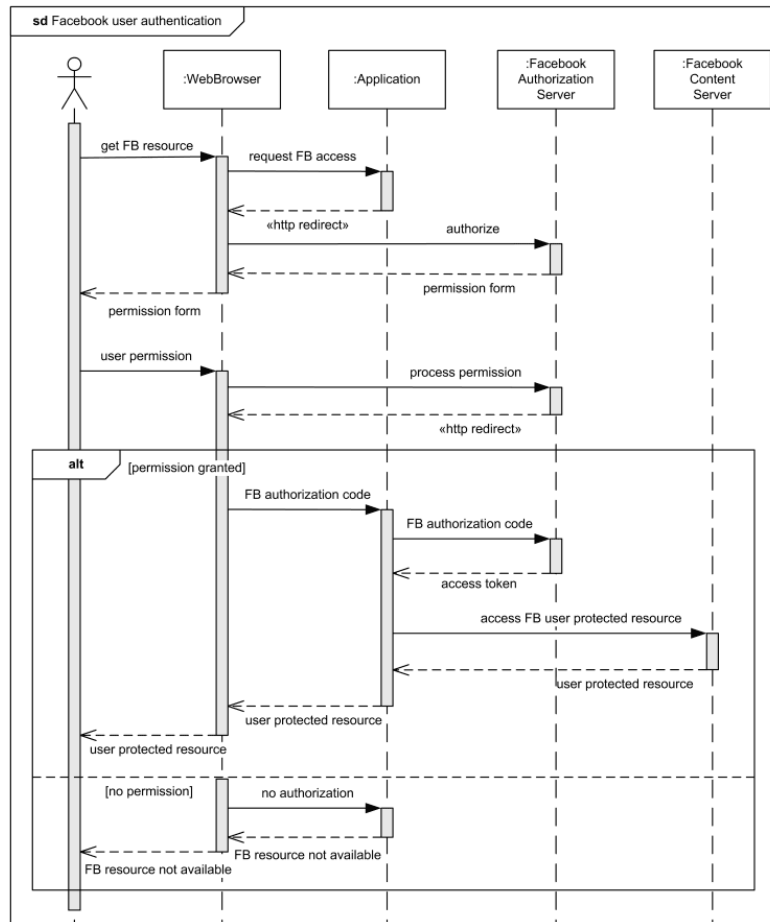
Représente les déclenchements d'événements en fonction des état du système.
Décrit le workflow d'un CU, ordonne les actions.



Le diagramme de séquence

Il représente la succession chronologique des opérations réalisées par un acteur pour un CU particulier.

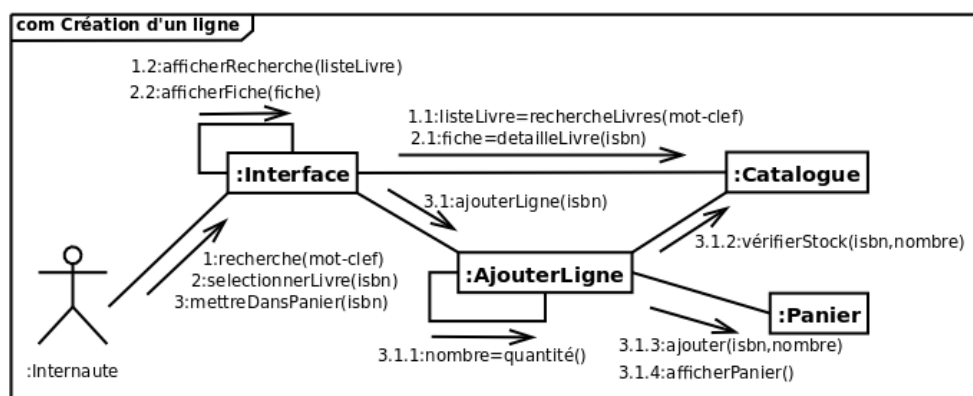
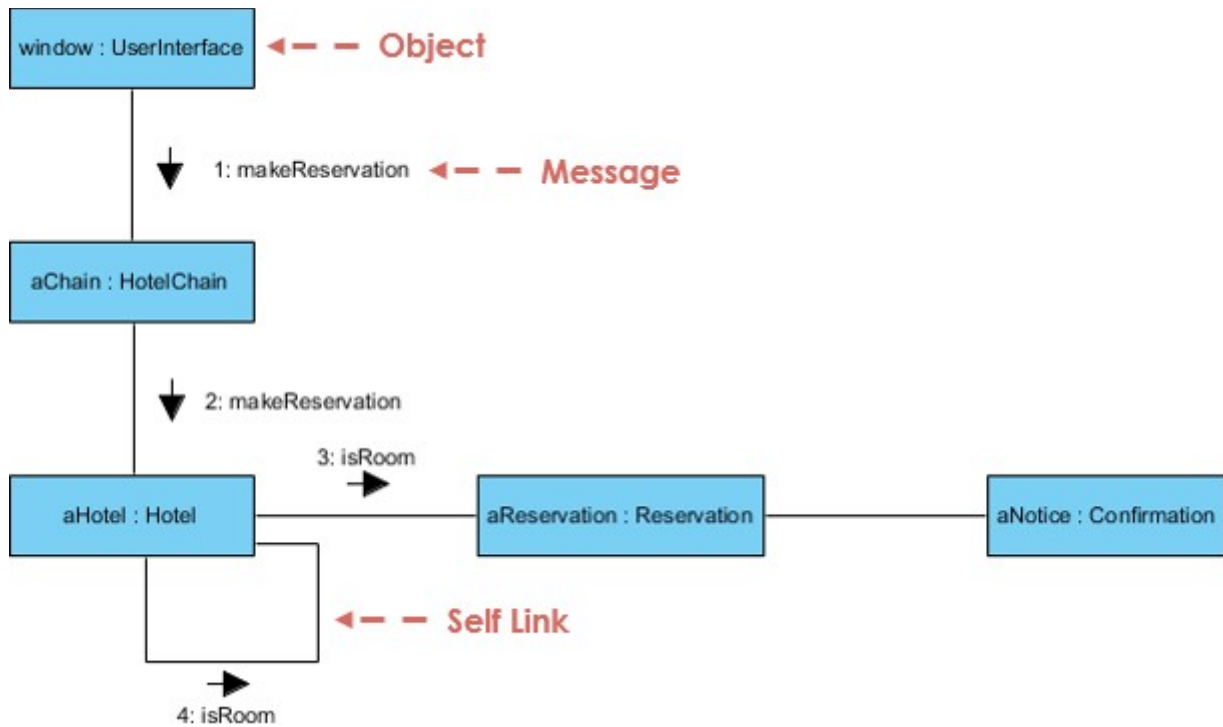
A lire du haut vers le bas.



Le diagramme de collaboration

Il montre les interactions entre les objets tout en donnant également la répartition spatiale de ces objets. Le diagramme de collaboration et celui de séquence sont très proches. Le fond est le même mais la forme change.

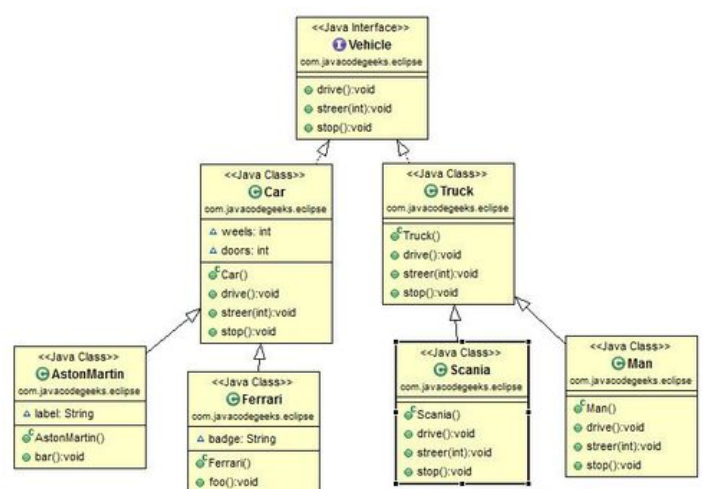
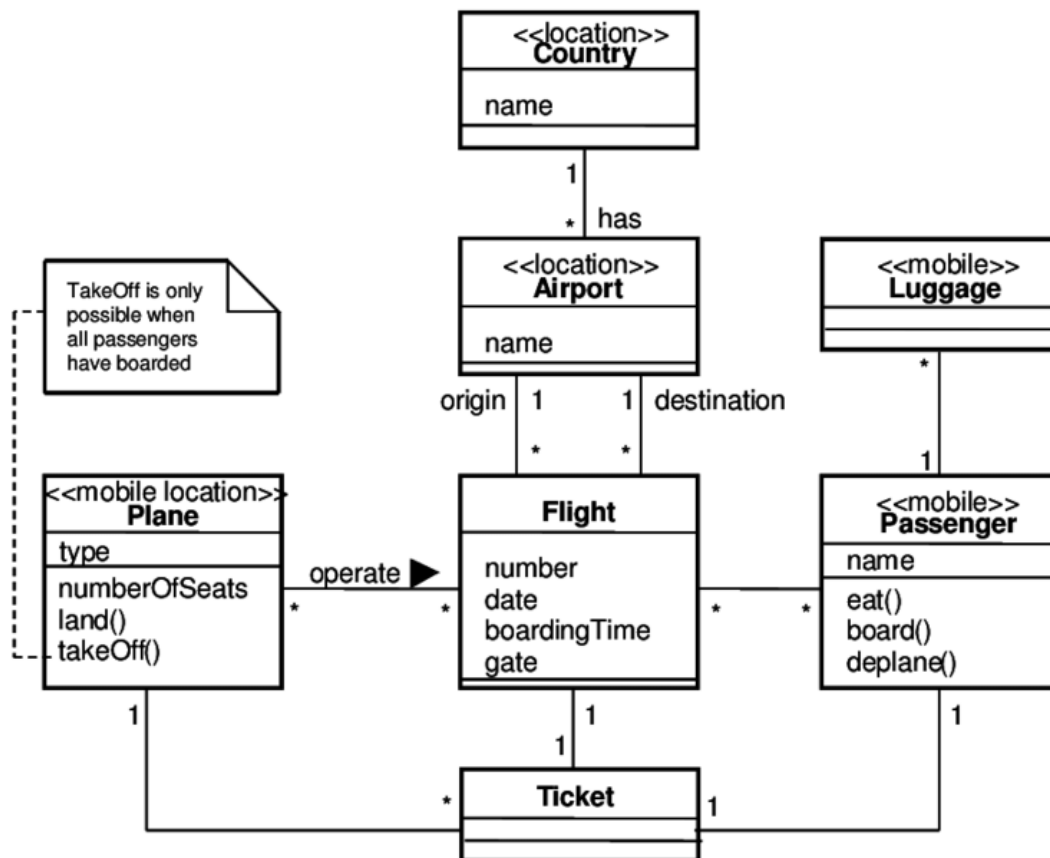
Les messages sont ordonnés



Le diagramme de classes

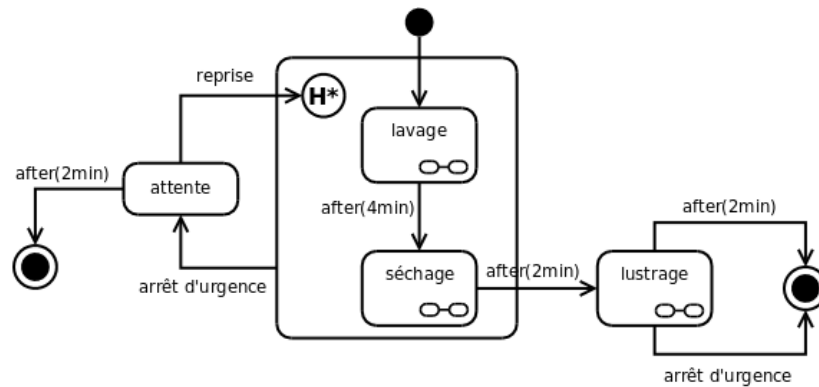
Le diagramme central, il représente les classes d'un système.

Pour chaque une classe, on aura son état (attributs), son comportement (méthodes), son héritage, ses interfaces.



Le diagramme d'état-transition

Comme son nom l'indique, il représente les états d'un objet et les transitions qui permettent de passer d'un état à l'autre



Le rond noir représente l'état initial.

Le rond noir entouré représente l'état final.

Le H représente la notion d'historique, de mémoire

Outils

Parmi les nombreux outils disponibles, intéressons-nous à :

StarUml

<http://staruml.io/>

BoUml

<https://www.bouml.fr/>

PlantUML

<http://plantuml.com/fr/>

<https://sourceforge.net/projects/plantuml/>

A partir d'un langage simple, les différents schémas peuvent être générés sous forme d'image.

Il existe de nombreux plugins PlantUML (NetBeans, Eclipse, VS Code, IntelliJ, Word).

Attention, ils peuvent contacter un serveur distant.

Heureusement, si vous êtes attaché à la confidentialité, il est possible d'avoir son propre serveur PlantUML via une image Docker (configurer son éditeur pour l'utiliser).

```
docker run -d -p 8080:8080 plantuml/plantuml-server:jetty
```

A noter d'ailleurs l'intégration de PlantUML dans AsciiDoc
(format de document orienté texte, comparable à Markdown)

Exemple :

Soit un fichier sequenceDiagram.txt contenant :

@startuml

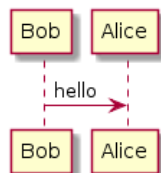
Bob->Alice : hello

@enduml

Pour produire le schéma correspondant, il faudra exécuter :

```
java -jar plantuml.jar sequenceDiagram.txt
```

Ce qui générera l'image suivante :



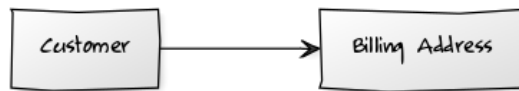
Autres

Très intéressant aussi, **yUml** <https://yuml.me/>

un site web qui générera vos diagrammes sous forme d'image par utilisation d'URL personnalisées

Exemple

``
donnera



Approche MDA

Acronyme anglais de « model driven architecture ».

A connaître, MDA est approche d'ingénierie dirigée par les modèles (IDM).

Un outil MDA prendra en entrée un ensemble de modèles UML pour fournir en sortie tout ou partie d'un projet/d'une application.

Références

Pour compléter et détailler les nombreuses notions entrevues ici, je vous conseille :

[https://fr.wikipedia.org/wiki/UML_\(informatique\)](https://fr.wikipedia.org/wiki/UML_(informatique))

<http://uml.free.fr/>

<https://laurent-audibert.developpez.com/Cours-UML>

<https://cian.developpez.com/uml2/tutoriel/sequence/>

Livre « Modélisation objet avec UML » par Pierre-Alain Muller

Mises en application

Par groupe, il faudra produire un dossier permettant l'implémentation d'un système.

Ce dossier reprendra les différents diagrammes listés précédemment, tout ou partie (absences à justifier) ainsi que les autres diagrammes vus pendant vos recherches si vous les estimez utiles.

Il est conseillé de faire les diagrammes dans l'ordre indiqué.

Le sujet sera au choix par groupe (3-5 personnes), à valider avec le professeur, de manière à pouvoir amener les connaissances métier des uns aux autres.

Pour ceux qui travaillent dans le secret défense, quelques sujets possibles :

Etude d'une application de gestion d'école (inscriptions, présence, ...)

Gestion d'un point de magasin (gestion du stock, vente, indicateurs)

Etude d'une application de gestion de publications multi-canal

Etude d'une application d'information multi canal (et son back office) (news choisies et possible aggrégation de sources diverses comme fb, twttr, etc ...)

Etude d'un système de gestion d'accès