

Génie logiciel

Claude Godart

Sources

Software engineering

Definitions

Un processus fournit des principes (règles flexibles) pour le bon enchaînement des activités de dev. log.

Un processus éncle, c'est un guide qui définit des garde fous au sein des quels ils existe beaucoup de libertés (planification imprécise)..

TD1 UML : Unified Modelling Language

On va étudier 8 types de Diagram UML

Esperanto : langage très simplifié, accessible à tout le monde

1. Diagrammes de cas d'utilisation (Use case diagram)

Se fier aux diapos, il est plus approfondi

Spécification du besoin, généralement le premier à utiliser

Obj : Capturer les besoins

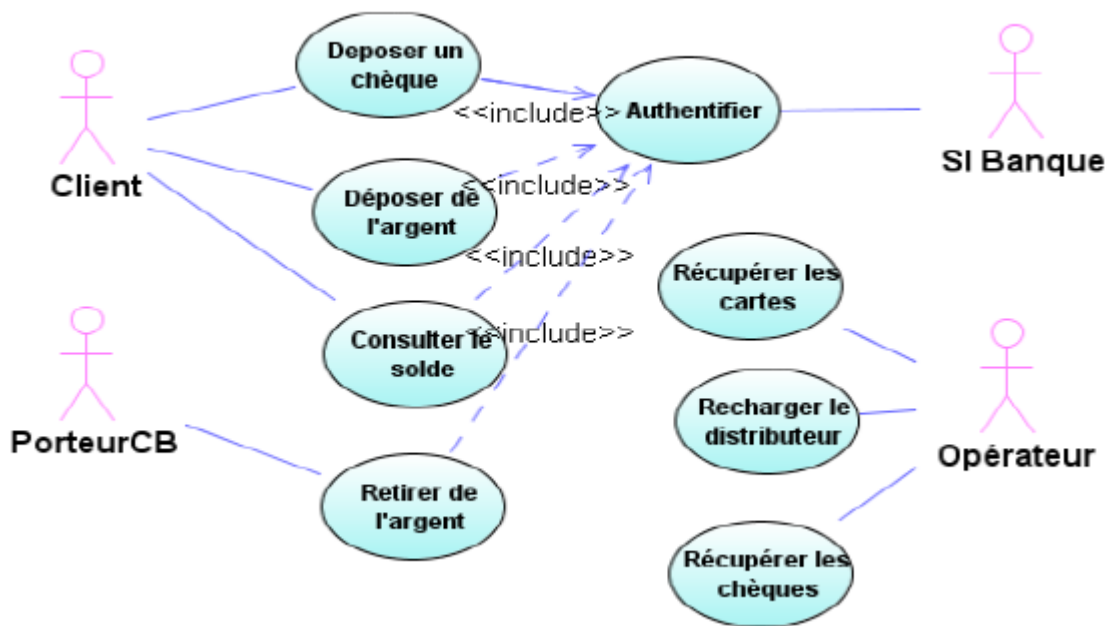
Toute chose identifiable que le client veut faire avec le système

Description sommaire et nom pour chaque chose. On approfondira plus tard.

Quoi ? Comment utiliser ? ET PAS LE comment faire ?

Modélise service, comportement, boutons

Exemple : Gestionnaire automatique de banque



Remarque, on pourrait avoir un Porteur CB qui « implémente » Client il se spécialisent.

Intérêt : éviter les oublis

Représente l'expression des besoins fonctionnels.

Utile au dialogue

Servent de base au processus de dév. y compris test

Cas d'utilisation modélise un service rendu par le système. (on n'en fait pas pour chaque scénario)

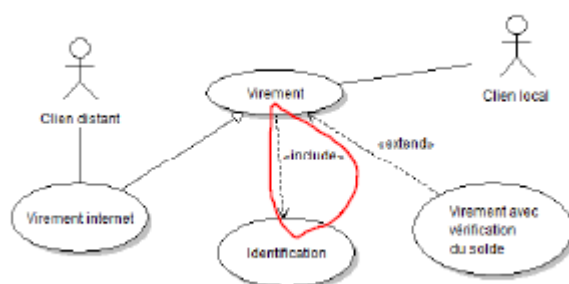
Deux types d'acteurs : principal (utilise, modifie le modèle) et secondaire (est sollicité pour obtenir une info complémentaire).

Un cas peut en inclure un autre (NE PAS EN ABUSER, DEVIENT VITE LOURD ET INUTILEMENT COMPLIQUÉ)

KISS : Keep It Small and Simple

Il est possible (et souhaitable !) d'organiser les cas : inclusions, spécialisation, packages :

- Inclusion



- Generalisation
- Extension

Le cas d'utilisation ne décrit pas de scénarios. On fait une description textuelle adaptée au client.

Scénario nominal

- 1. Le porteur de CB introduit sa carte dans le lecteur du GAB.
- 2. Le GAB vérifie que la carte introduite est bien une CB.
- 3. Le GAB demande au porteur de saisir son code.
- 4. Le porteur de CB saisit son code.
- 5. Le GAB compare le code saisi avec celui inscrit sur la puce.
- 6. Le GAB demande une autorisation au SA CB.
- 7. Le SA CB donne son accord et indique le solde hebdomadaire.
- 8.....

Le scénario nominal est le scénario type, déroulement parfait. Ensuite on fait les scénarios alternatifs. (Scénarios alternatifs / exceptionnels). Il y a énormément de scénarios alternatifs, il faut déjà maîtriser les scénarios représentatifs !

Besoins en IHM

Formalisation avec diagrammes de séquence ou d'activité

Diagramme d'activité avec des portes AND et OR.

Scénarios

On ne fait que les scénarios pour les cas centraux. Diff :

- Scénario(s) nominal
- Scénario(s) exceptionnel
- Scénario(s) alternatifs

Exemple cas bibliothèque, gérer les emprunts :

Correction

Scénario nominal (le livre est disponible mais le client ne l'a pas récupérer)

1. Vérification de l'existence du client
2. Vérifier la validité du client (pas trop de livres en retard)
3. Vérification du référencement du titre
4. Vérification de la disponibilité
5. Aller chercher le livre
6. Enregistrer l'emprunt

Scénario exceptionnel (rare)

1. Vérification de l'existence du client
2. Le client n'existe pas, on lui propose l'inscription (**Comme ça on découple les cas d'utilisation et on évite la prog. Spaghetti -celle avec des morceaux de code redondant dans tous les coins-**)

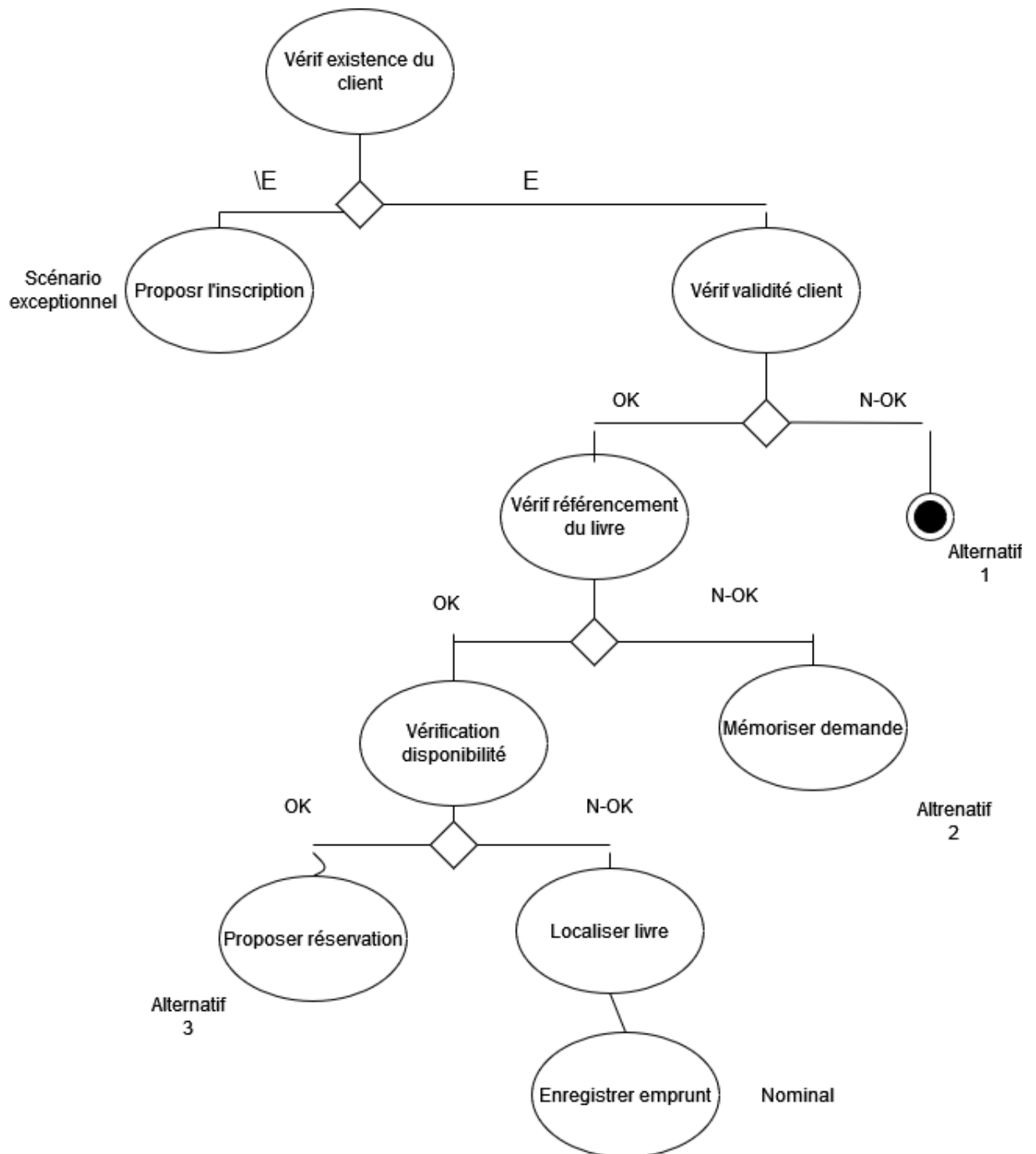
Scénario alternatif (non disponible)

1. Vérification de l'existence du client
2. Vérifier la validité du client (pas trop de livres en retard)
3. Vérification du référencement du titre
4. Vérification de la disponibilité
5. Le titre n'est pas disponible, on propose au client de le réserver (à condition que la fonction réservation comprenne la vérif de la date)

Scénario alternatif (non référencé)

1. Vérification de l'existence du client
2. Vérifier la validité du client (pas trop de livres en retard)
3. Vérification du référencement du titre
4. Le titre n'est pas référencé, mémoriser la demande (pour que le gestionnaire décide de l'achat)

Le scénario exceptionnel est celui qu'on ne veut vraiment pas qui se produise (automate de banque avec une carte volée), dans l'exemple de la bibliothèque il n'y en a pas vraiment.



Contraintes

- Performance
- Sécurité :
 - Login/mdp
 - **Carte numérique d'adhérent**
- **Maquetter des interfaces (visuels de l'interface user) :**
 - Définir les champs utiles dans le visuel

Emprunt	N° tel
Vérif client	<input type="text"/>
Vérif livre	Adresse
	<input type="text"/>
	Titre
	<input type="text"/>

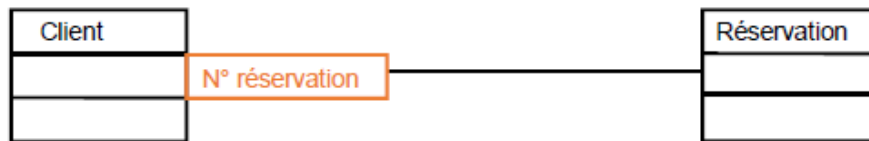
○

2. Diagramme de classe UML

Propriétés d'un objet :

- Attributs
- Opérations
- Ses liens (associations) avec les autres objets

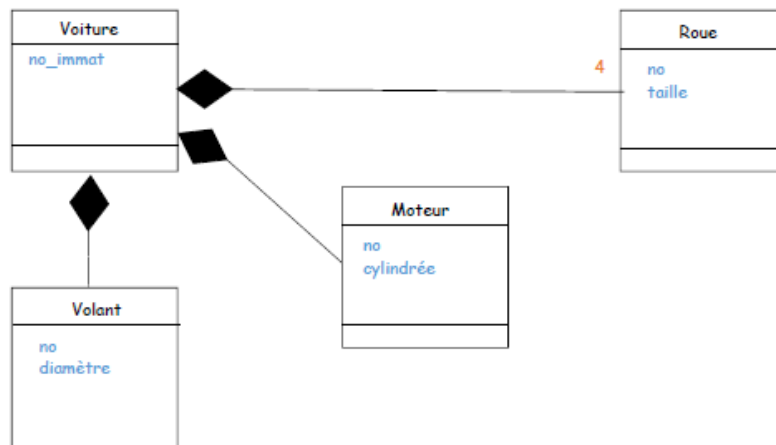
Qualification (a éviter mais ça existe):



« Un client et un numéro de réservation détermine une réservation »

Composition

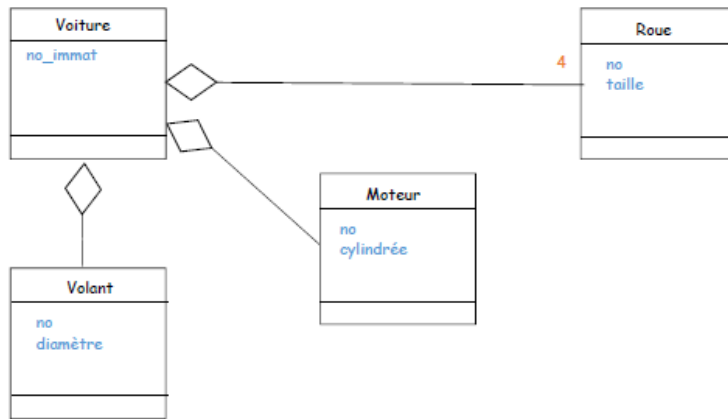
Composition



◆ : Composition : la vie du composant est liée à celle du composé (si on détruit l'objet composé, on détruit ses composants).

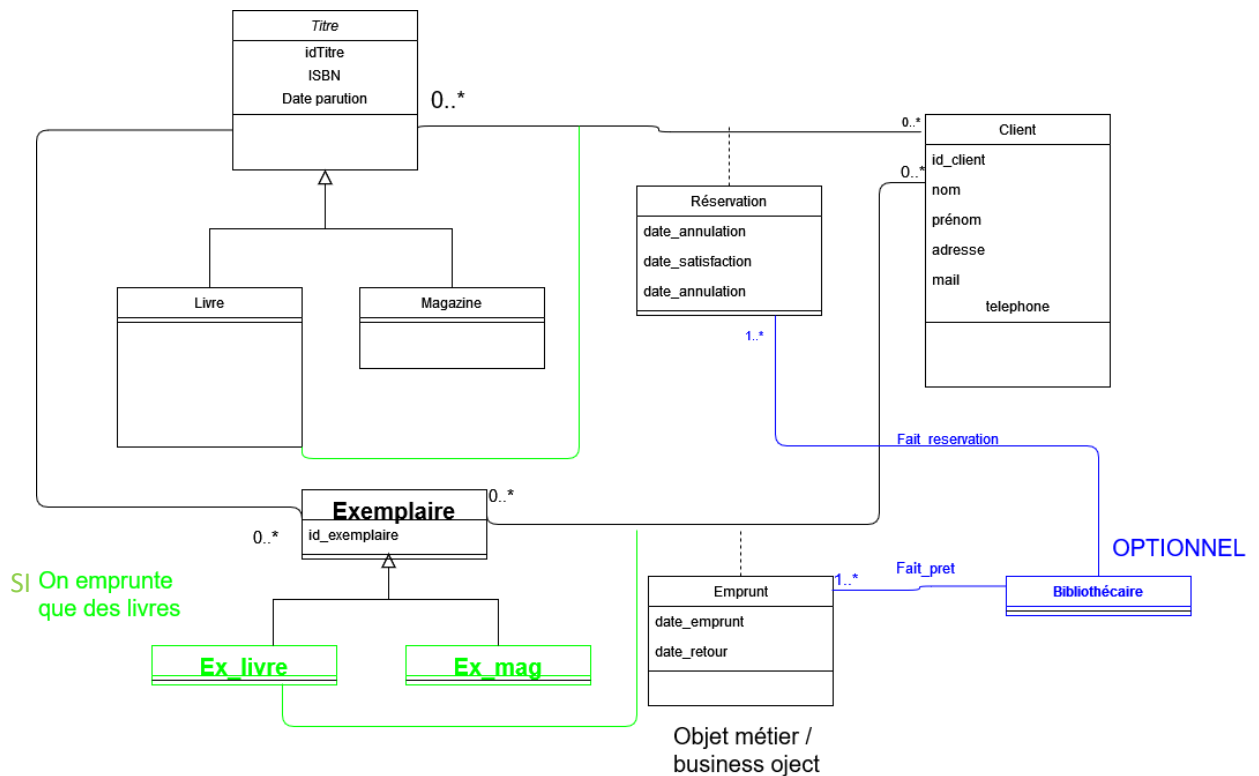
Composition propage la suppression : suppression voiture supprime roue

Agrégation



◊ : Agrégation : la vie du composant n'est pas liée à celle du composé (si on détruit l'objet composé, ses composants survivent individuellement).

Agrégation ne propage pas la suppression

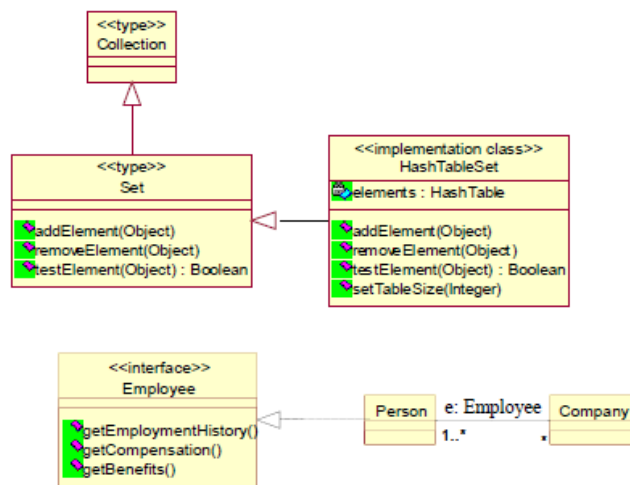


On décrit d'abord un diagramme à base de business objects / objets métiers.

Le modèle évoluera suivant si on s'oriente vers de la POO ou de la BDD ou autre

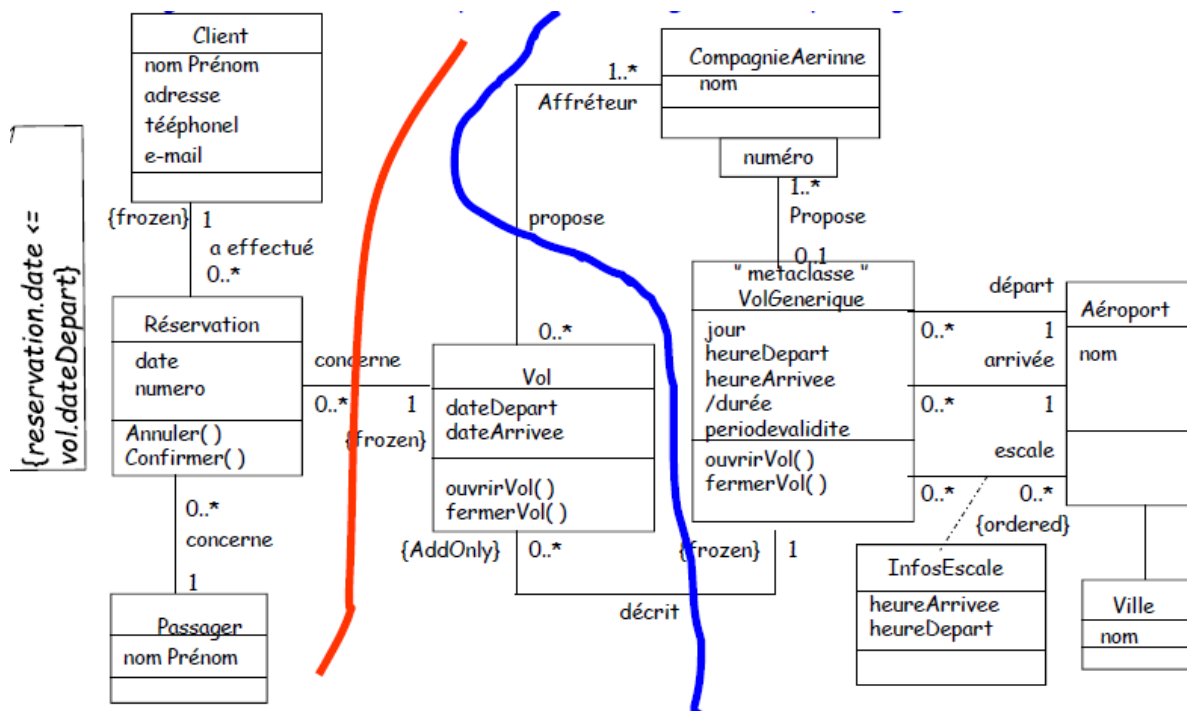
Interface = classe abstraite sans attributs

Séréotypes orientés développement, associations d'implémentation, de dépendance



Package

Cas compagnie aérienne : une partie concerne les vols et une partie business (vente) peuvent être scindées en deux. D'autres découpages auraient pu être effectués.



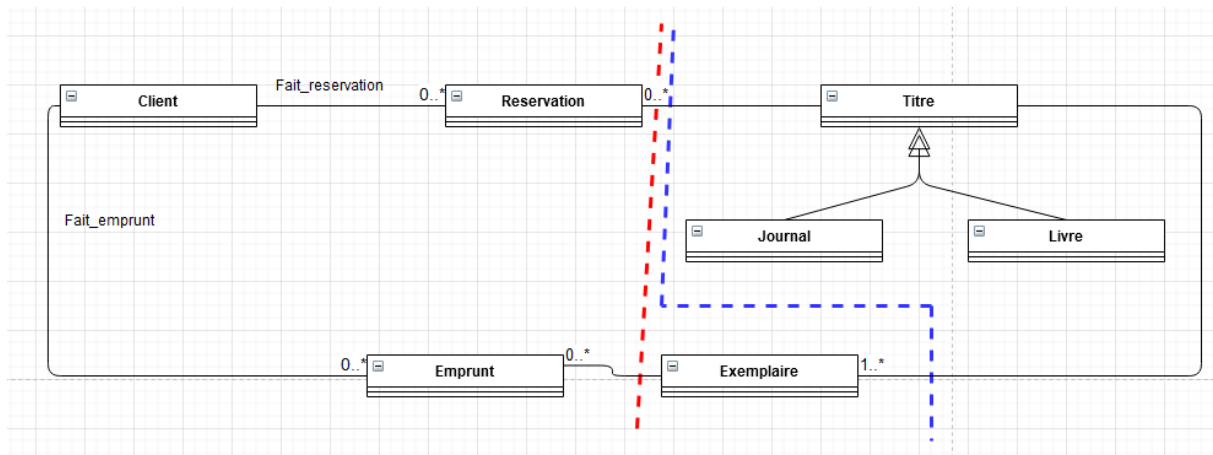
Cependant si on met vol dans le package de droite on ne conserve une seule association entre les packages (moins de couplage, ce qui est bien mieux).

Les packages permettent de :

- Simplifier la complexité, de traiter le prob morceau par morceau
- Réutiliser pour d'autres applis

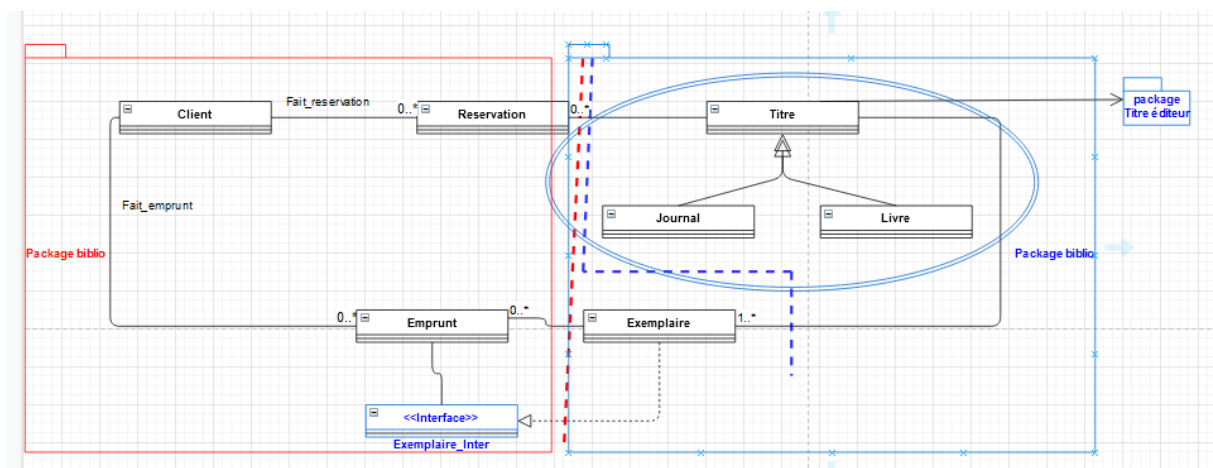
- Réutiliser pour d'autres modèles

Application au cas bibliothèque :



Dans ce cas il est difficile de décider car dans les deux cas il reste deux associations entre les packages.

On pourrait imaginer un modèle quelque peu différent :



Généralisation et réutilisation

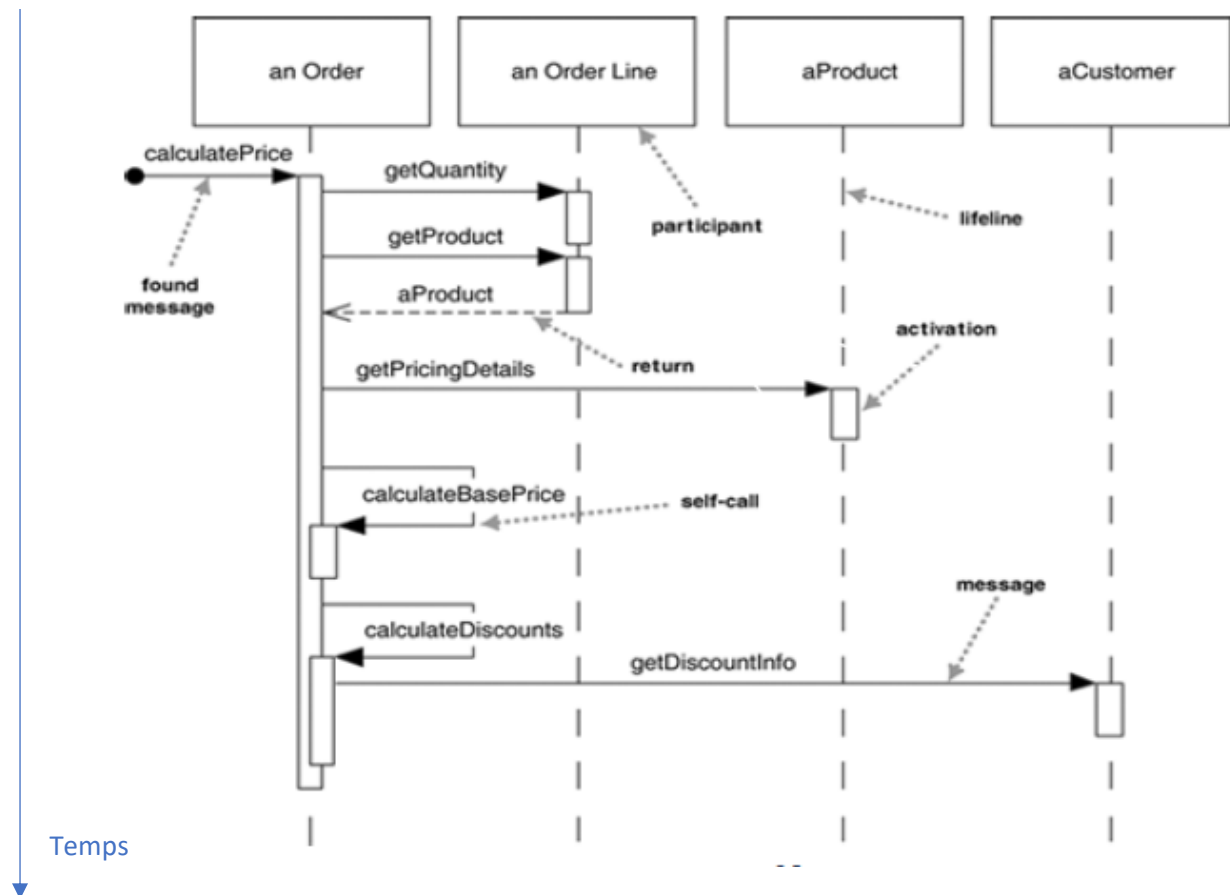
La société qui gère ces applis pour Air France veut aussi mettre en place un système de gestion de bus.

Fusion des deux modèles

3. Diagrammes d'interaction

Se base sur le diagramme de classes. Les classes sont sous-entendues par les objets (les rectangles en haut du diagramme).

Il est aussi lié au scénario.



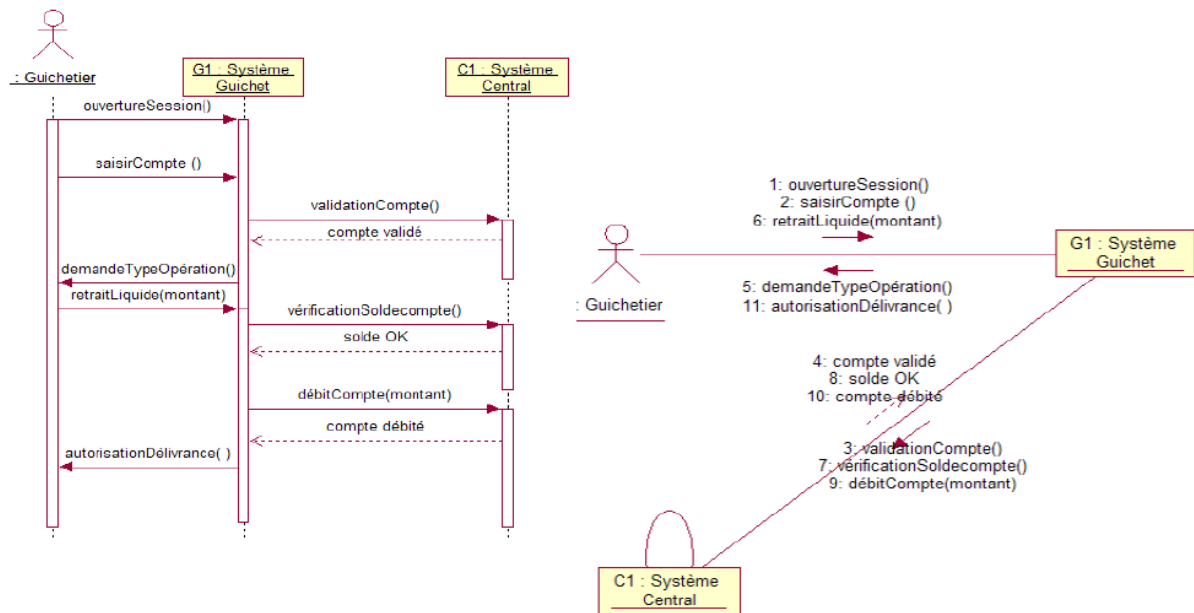
4. Diagramme de collaboration

Se base sur le diagramme de séquences, on garde l'ordre en numérotant les actions.

Diagramme de séquence est important dans les premiers temps du projet et plus simple à mettre en place pour définir le séquençement.

Diagramme de collaboration permet de montrer des choses supplémentaires (couches,...)

Chacun à ses avantages mais il est conseillé de faire les deux à des phases différentes du projet.



Exemple : Diagramme de séquence cas nominal de emprunt

On a un package de fonctions de base :

- Existe ?
- Créer
- Supprimer

Le diagramme de séquence doit être mis en relation avec le diagramme de classe.

Le diagramme de séquence permet d'implémenter le diagramme de classe :

- On ajoute un nombre d'emprunts en cours à la classe client
- Une fonction autorisation d'emprunt à client
- Un attribut est disponible : booleen à Titre
- Une fonction exemplaire disponible à Titre
- Une fonction rendre_indisponible à Exempleire (cette fonction sera lancée lors de la création d'un nouvel emprunt)
- Attribut nombre exemplaire disponible dans titre
- Fonction decrementer nb disponible dans Titre

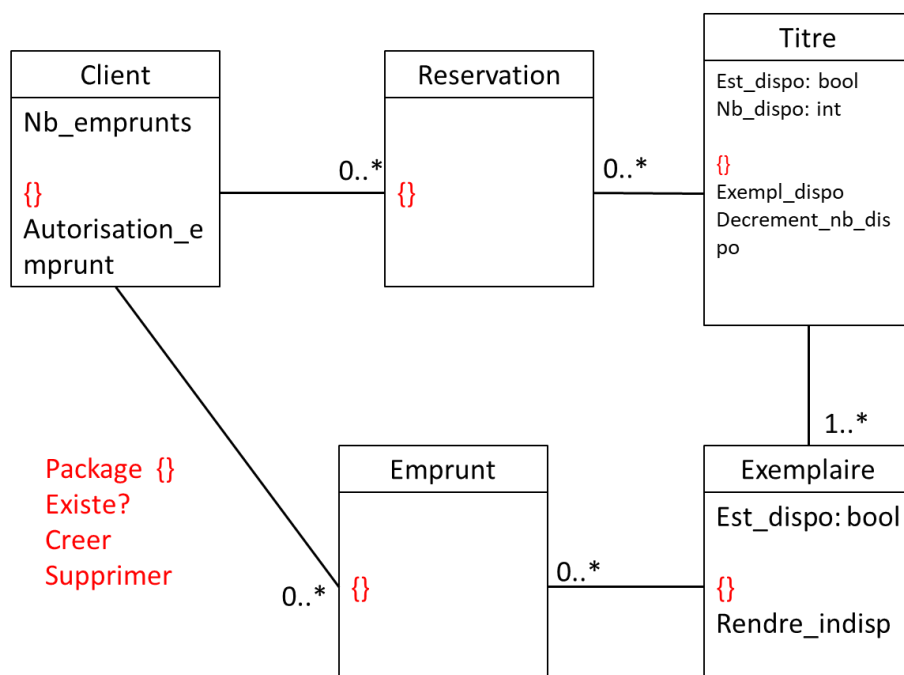
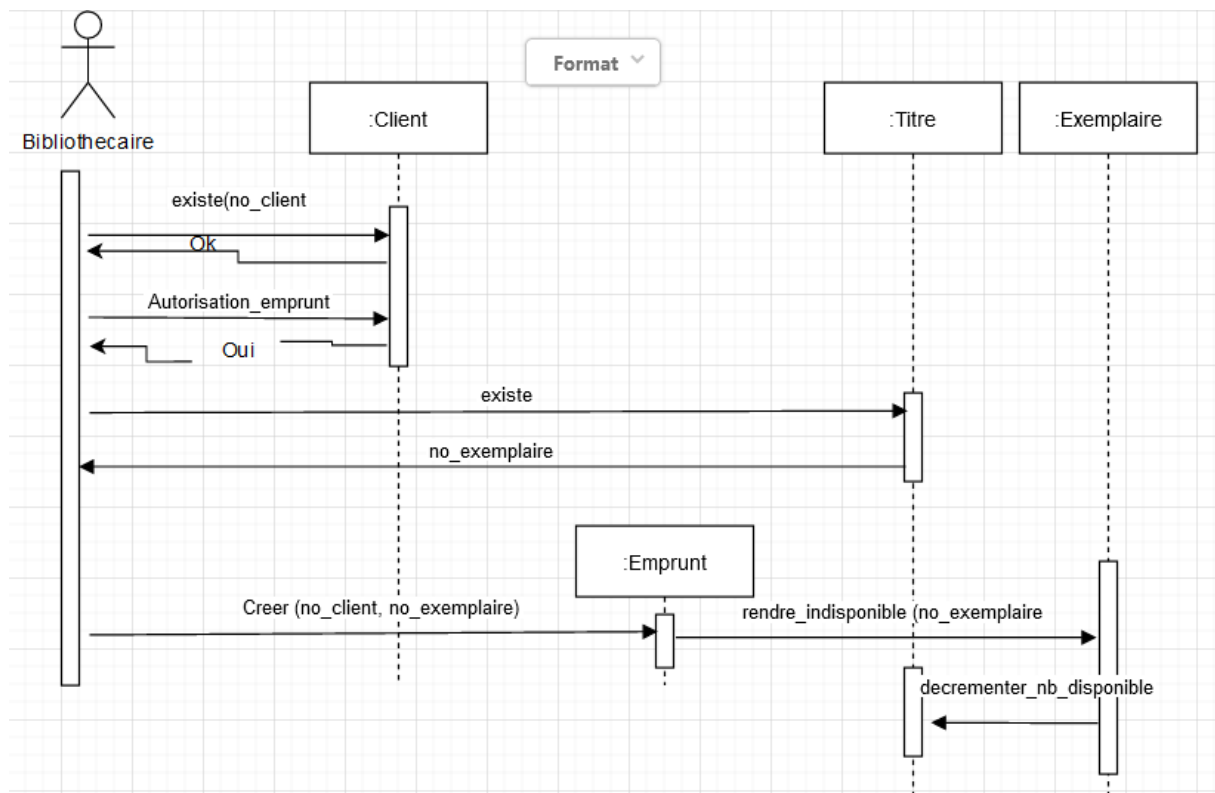


Diagramme à changements d'états / statechart

Donne la dynamique des objets dans **UNE CLASSE**

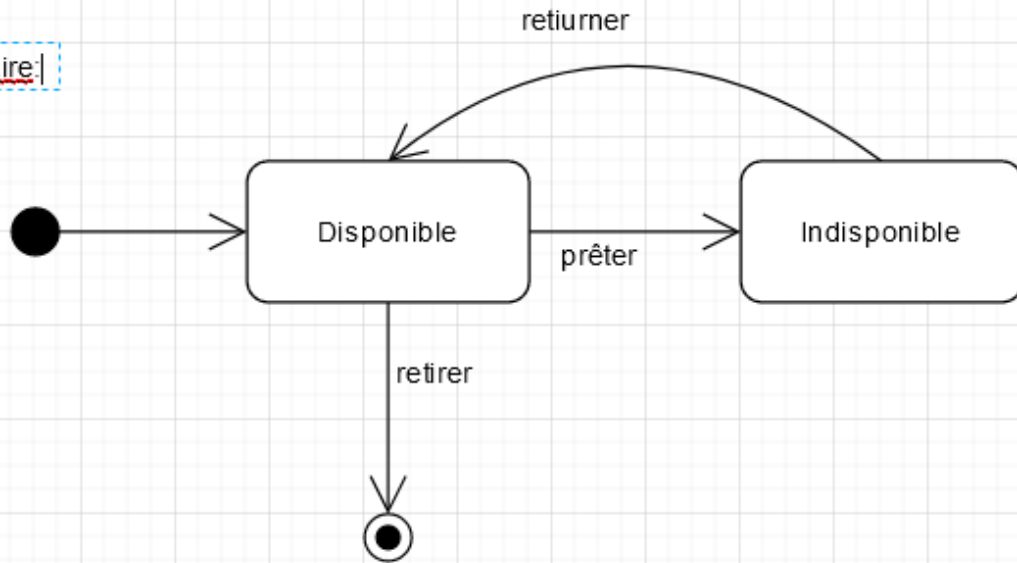
On en fait **un qui réponds à tous les scénarios**.

Voir réseaux de Pétri

[Exemple sur le cas bibliothèque](#)

On se limite à exemple

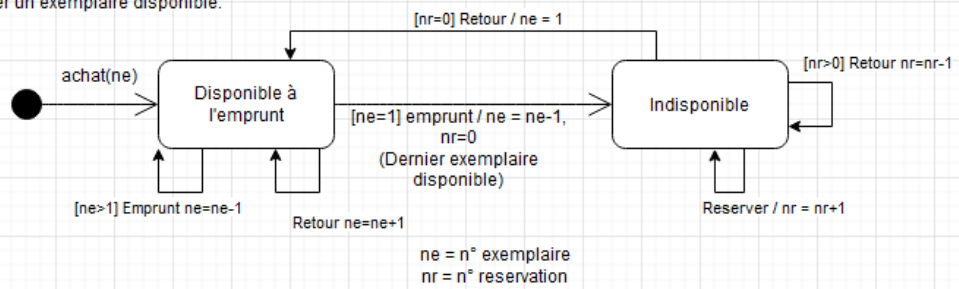
Exemplaire:



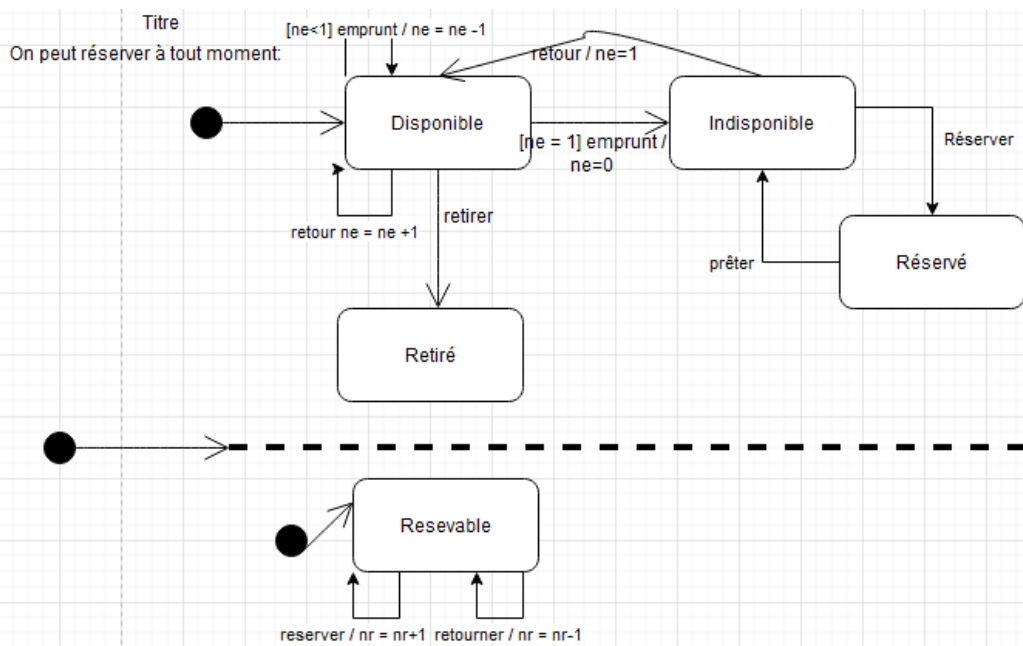
Ajout règle :

On ne peut pas réserver un titre dont un exemplaire est disponible.

On ne peut pas réserver un exemplaire disponible:



Changement de règle : on peut réserver n'importe quand :



6. Diagrammes d'activité

Tout ça pour le DS, deux pages (recto verso) de documentation

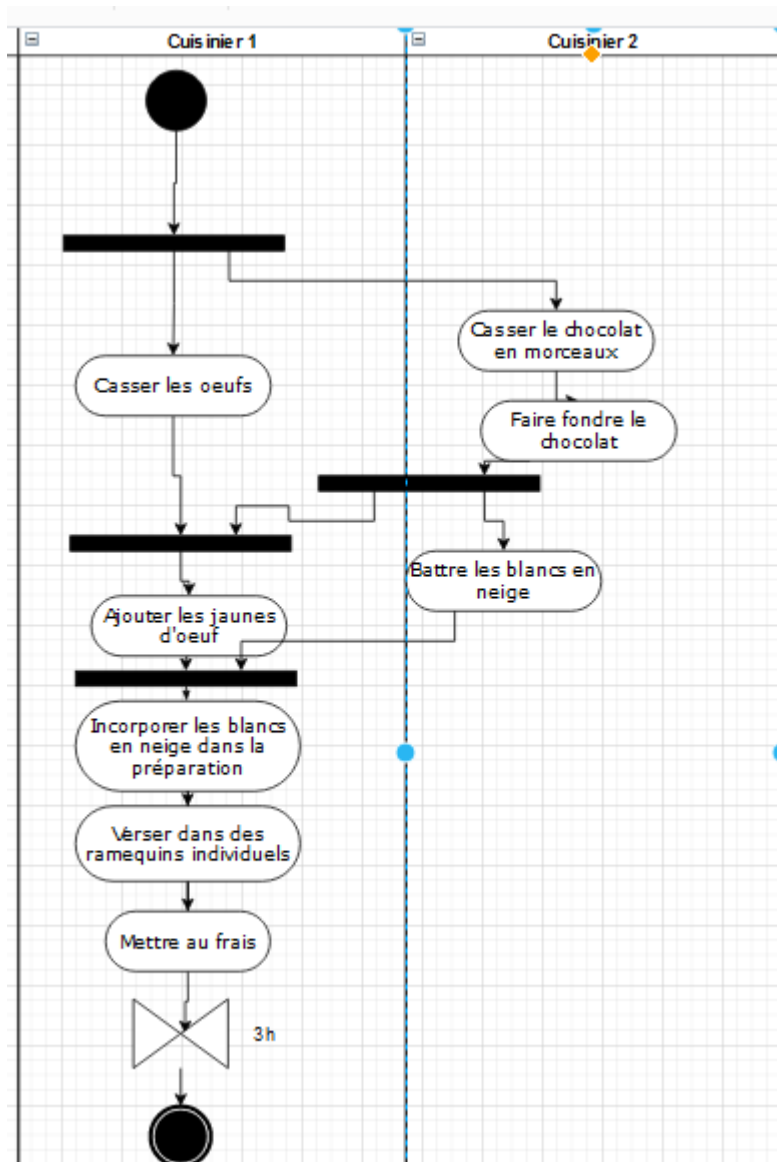
Orienté objet : diagramme de séquence

BDD, workflow : Diagramme d'activité

Barre = porte ET

On peut décomposer des activités avec des diagrammes plus précis dédiés

Exemple : gâteau au chocolat



DS 2 feuilles A4 recto/verso

7. Diagrammes de déploiement

Méthode = processus + notation

Projet

Sujet 2

Subject

Warehouse organization

- Two aisles, each aisle organized in bins
- Identification: 12-10 with 12 standing for the aisle and 10 for the bin

Receiving dock: where goods arrive

- o Verify that there is a purchase order existing and corresponding to the shipment
- o Mark the purchase order off (no more on the list)
- o Charge items on pallet
- o Go through the warehouse and unload at the right place
- o **Many different types of items on one pallet**
- o For each item, a corresponding bin which is not full is searched. If no one corresponds, items are put in an empty bin.
- o One kind of item per bin
- o **Think of how a particular bin is chosen**

Aisle and bin location

- **Stock should be as low as possible but still enabling to fill order (fill rate)**
 - o Count items on stock
 - Couple of team members manually count.
 - One aisle per day
 - Hard to find the bins
 -
 - o Order what you need

Ordering

- Reorder levels
- Sales rate
- Replenishment time

Picking an order

- Get from the order desk
- Walk around to find the items (seems there is no location list)
- IF not found -> order can't be filled
- Put into tote (cart)
- Ring back to order desk

Order desk

- Packaging
- Charging truck

1. Use case diagram

What we want

a. Paperwork

User -> verify shipment number = purchase number

retire purchase number from list

(? Verify items and quantities ?)

b. Storing goods

For each item -> search for location and search if there is still place

> Charge on pallet

If no location registered or full bin -> attribute new location near

(? Attribute with an order like product n° or type of product ?)

1. Phase 1: création

Objectifs

Approfondir 10% des cas d'utilisation

Définir les limites du système

Identifier les usages

- Cas d'utilisation
- Définir un ordre de priorités
- Interfaces utilisateurs

Esquisser une architecture initiale

Identifier les risques principaux

Conception

Implémentation

Préparation des tests

Planification de la phase d'élaboration

Analyse de l'existant (diagramme de classe des objets métiers initiaux)

Identification des cas d'utilisation

Use case diagrams **for a better warehouse organization, not for the actual one**

Scénario nominal arrivage produits :

- Item shipment arrives
- Search in system for corresponding shipment number
 - o Shipment number exists
- Count the items

- Number of items is right
- Add items to the virtual stock on the system
- Search for location in the system
 - Location exists
 - System returns a location giving clue to aisle, shelf and bin
- Bring items to locations and put them into the bins
 - There was enough space left in the bin

Scénario nominal préparer une commande :

- System checks if there is the right number of items left in stock
 - Inventory is ok
- Open order on the system
 - Acknowledge that you take the order
- System shows a table with items, quantity, location and ordered by location
- Go to the first location
- Take the right number of items
 - There were enough items
- Put them in the tote
- Acknowledge to the system
 - System removes items from inventory
- Do the same for every item
 - Went ok
- Go back to the order desk
- Acknowledge to system

Scénario nominal expedition:

- Systems sends order.
- Acknowledge to system.
- Package the order
 - Acknowledge
- Do the same.
- All went ok, no remaining item or no lost item
- Put on truck
- Acknowledge to system

Diagramme des cas d'utilisation (pour quelques cas)

Ordonné (ordre de priorités) des diagrammes d'utilisation

Diagramme de classe des objets métiers recensés dans l'étude

Liste des contraintes fonctionnelles et risques

Organisation des premières itérations (quel cas d'utilisation dans quelle itération).