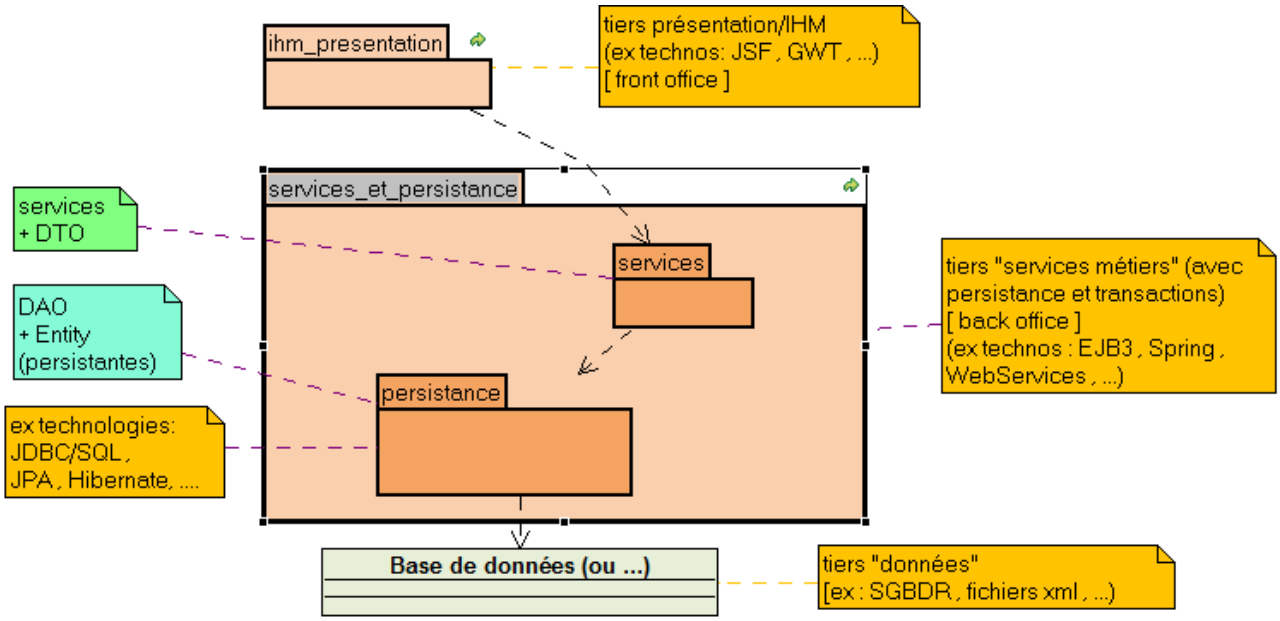
I - Exemple de diagrammes UML / architecture

1. Conception générique (récurrente)

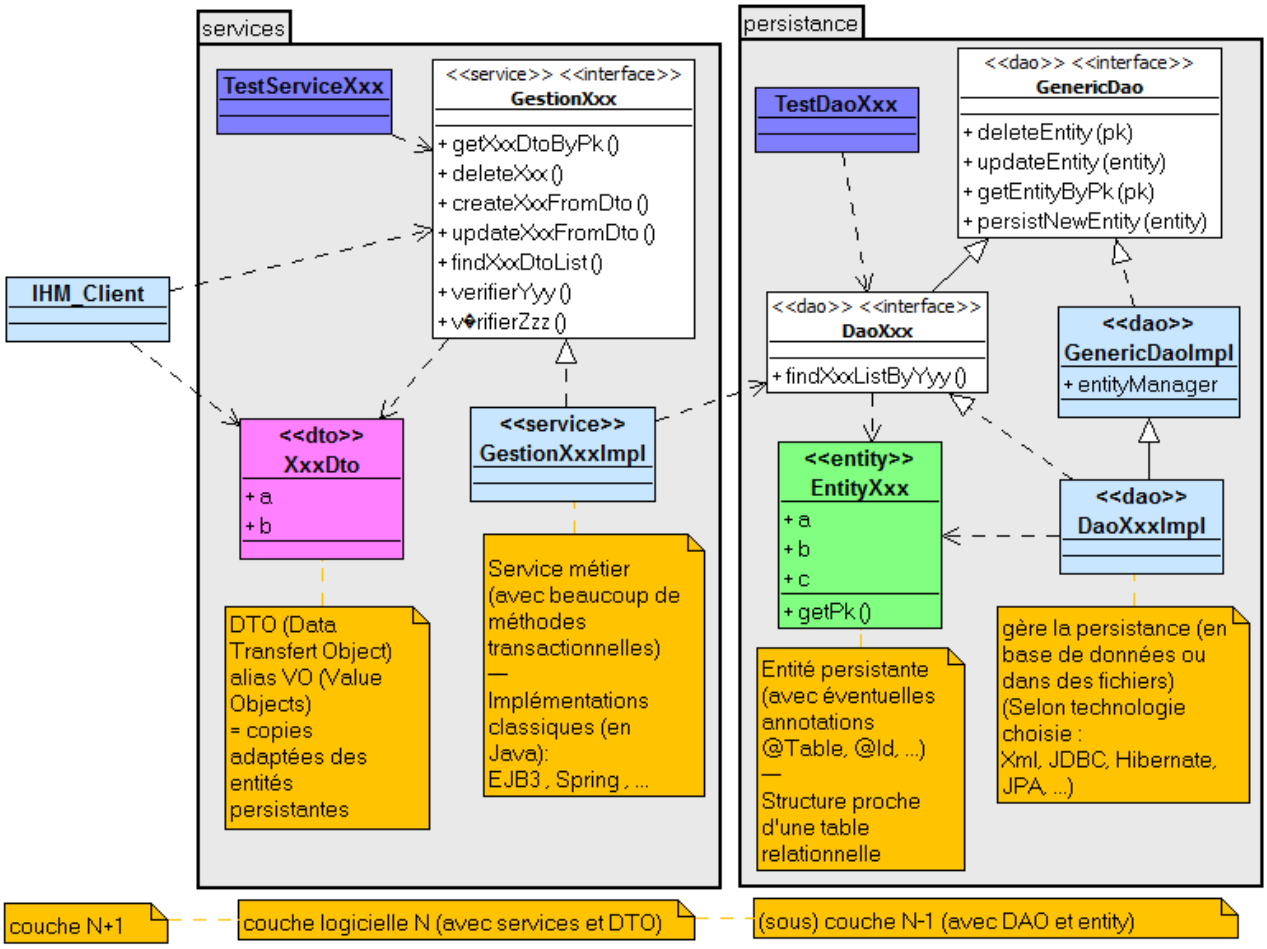
1.1. Vue d'ensemble générale en tiers/parties :



Ici modélisé avec diagramme de classes UML (packages , dépendances et commentaires)

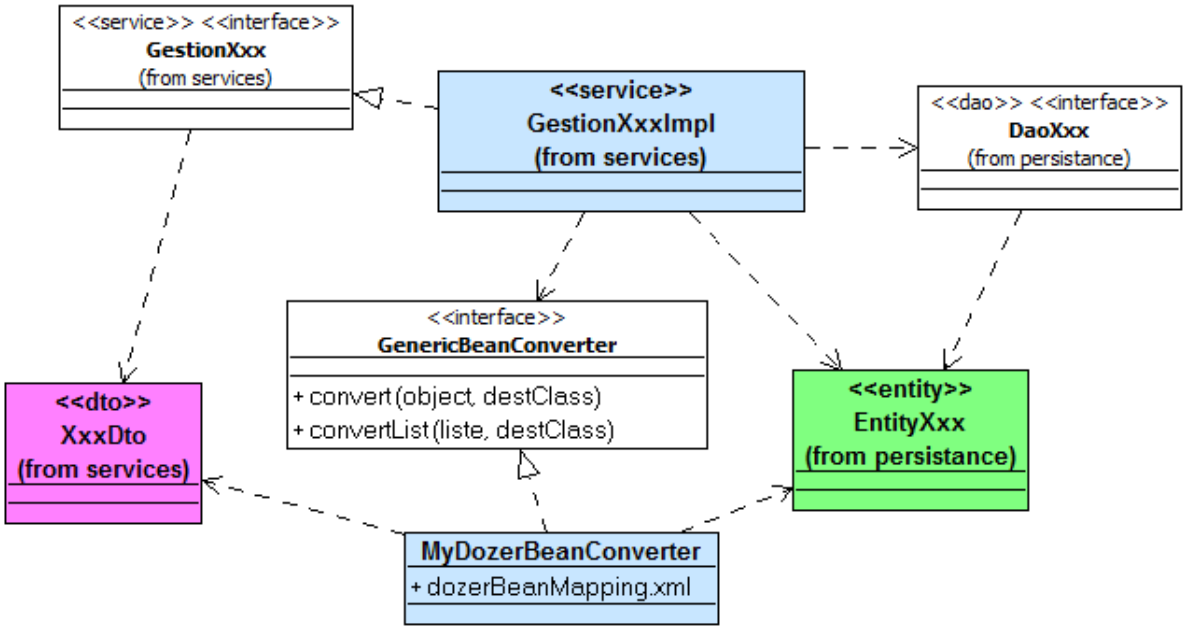
1.2. Couches logicielles et structures génériques (ici avec DTO) :

Zoom sur la partie "services et persistance"



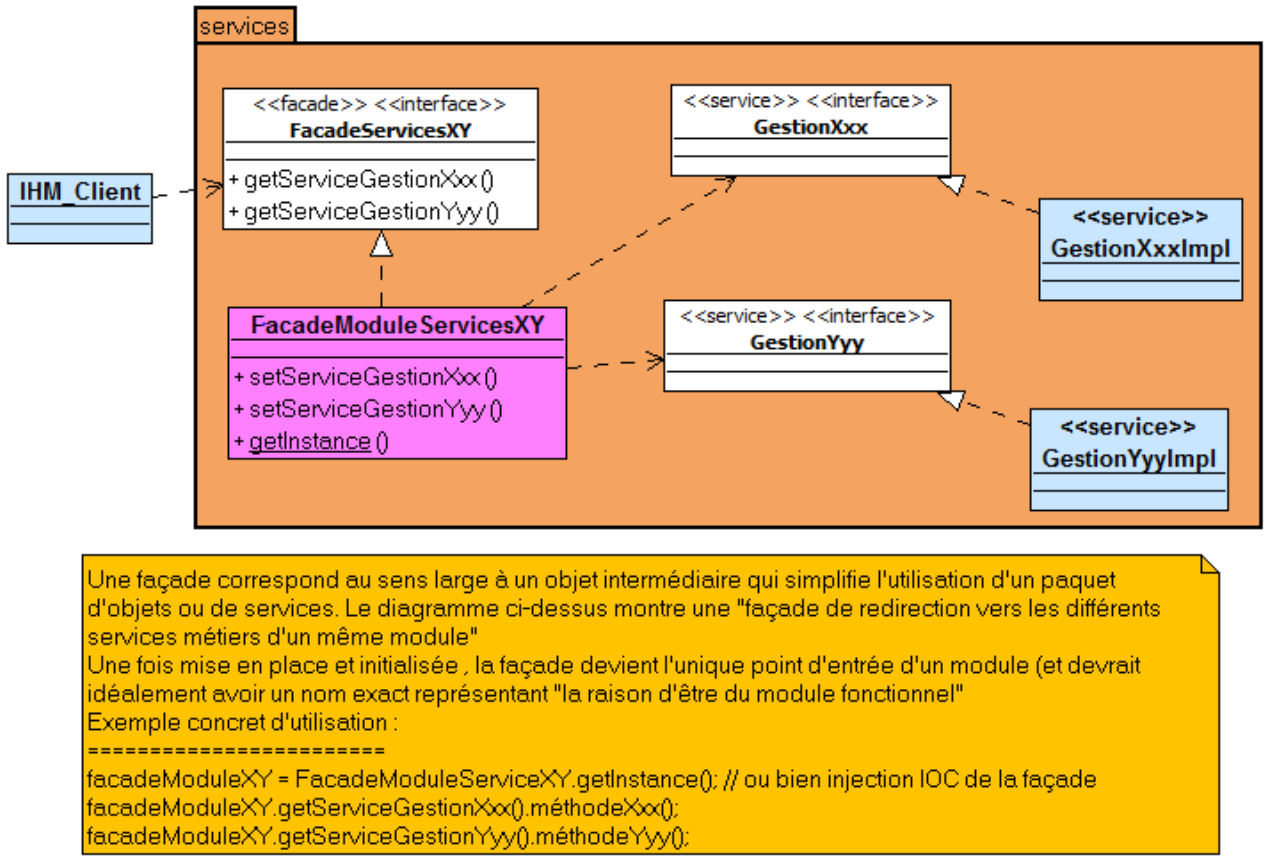
NB : Le diagramme ci-dessus montre une architecture possible avec des DTO/VO.
On peut éventuellement envisager des architectures simplifiées (sans DTO/VO) où les services métiers remontent directement les entités persistantes et des architectures ultra-simplifiées sans DTO (en utilisant directement "entityManager" et des "NamedQuery" dans le service métier).

Zoom sur conversion "dto / entity" :



Les méthodes internes du service métier "GestionXxxImpl" utiliseront un objet utilitaire
(de la classe MyDozerBeanConverter implémentant l'interface GenericBeanConverter).
NB: La thechnologie open source "Dozer" sert à recopier automatiquement les valeurs d'un objet java vers un autre en s'appuyant sur un fichier de configuration Xml si les propriétés à recopier n'ont pas les mêmes noms.
Exemple concret:
==========
La méthode updateXxxFromDto() recoit un paramètre d'entrée de type XxxDto.
celui-ci est converti via :
    entityXxx = genericBeanConverter.convert(xxxDto, EntityXxx.Class);
l'entité convertie est ensuite repassée au DAO:
    daoXxx.updateEntity(entityXxx)
==========
Le convertisseur est quelquefois utilisé dans le sens Entity —> DTO pour transformer les valeurs de retour

Zoom sur une façade :

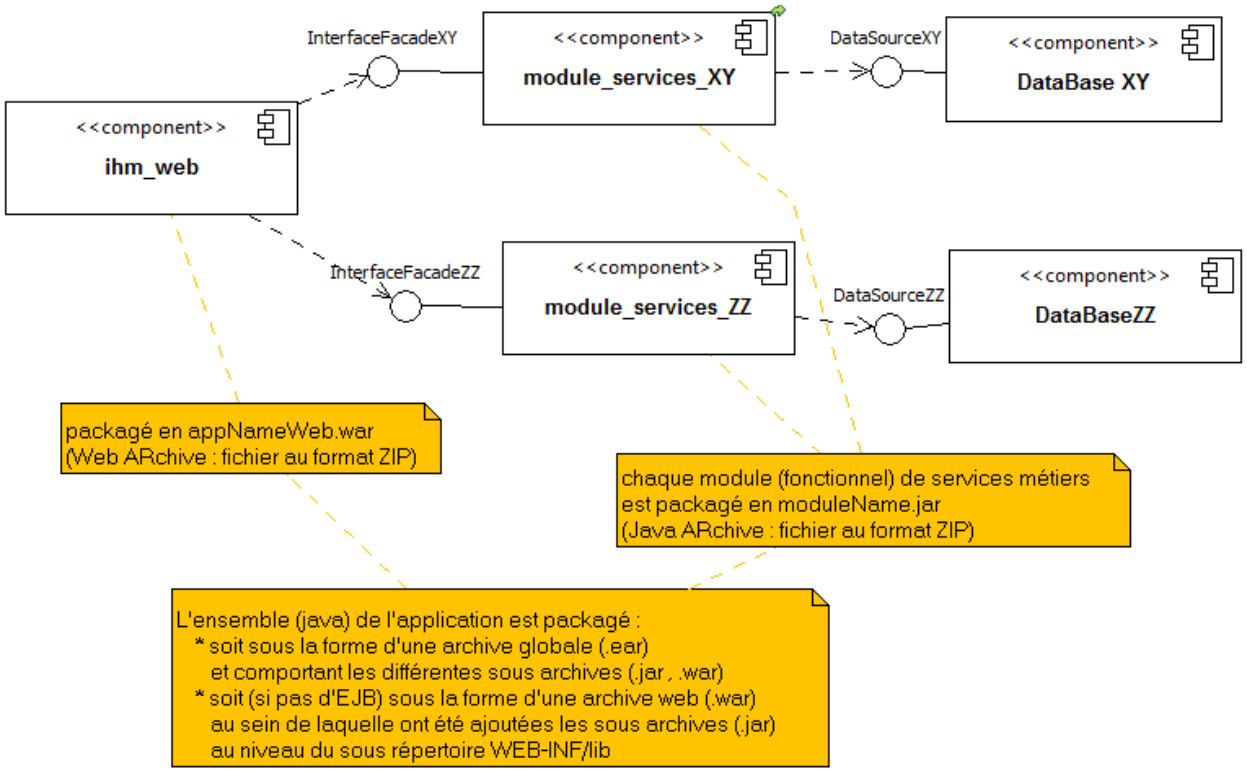


NB : La façade modélisée ci dessus est de type "point d'entrée / façade de redirection".

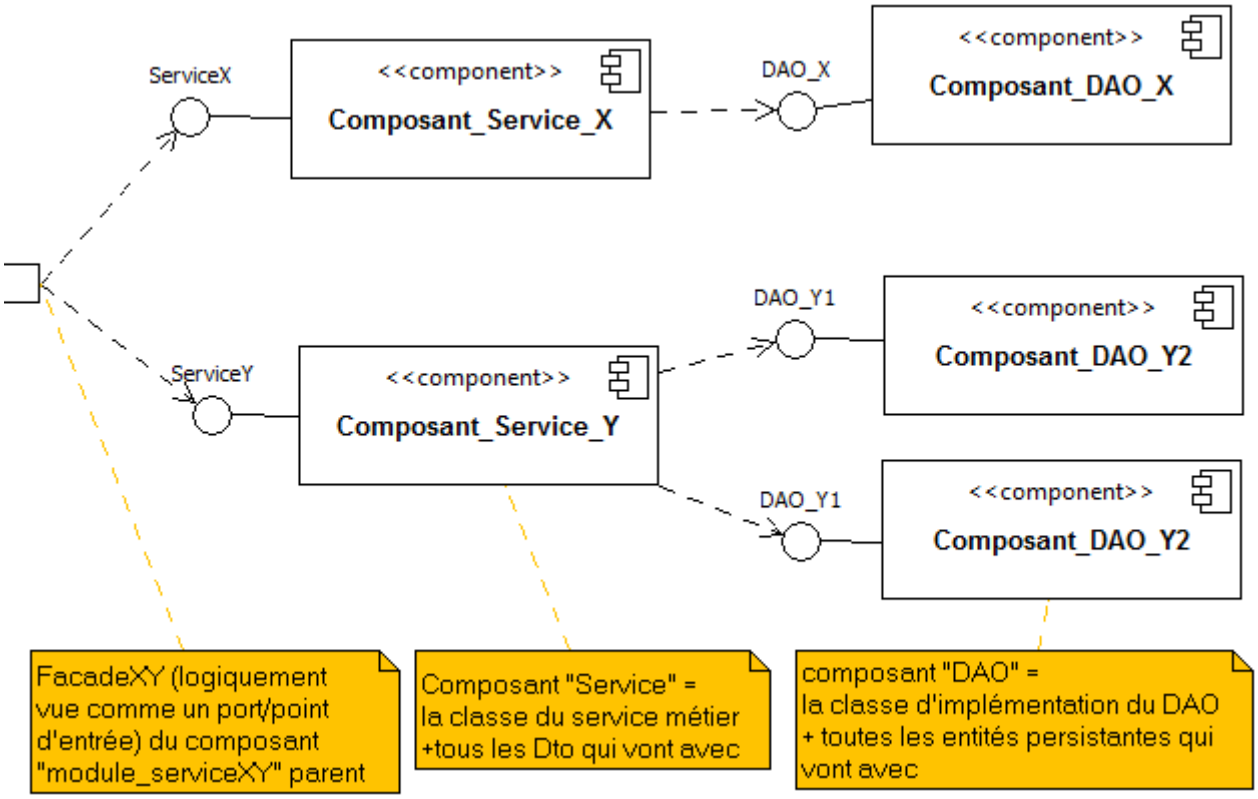
On peut envisager plein d'autres types de façades :

- agnostiques (masquant volontairement les technologies utilisées en arrière plan)
- ….
- par rôles "utilisateurs" (regroupement des méthodes accessibles à un certain type d'utilisateur)
- …

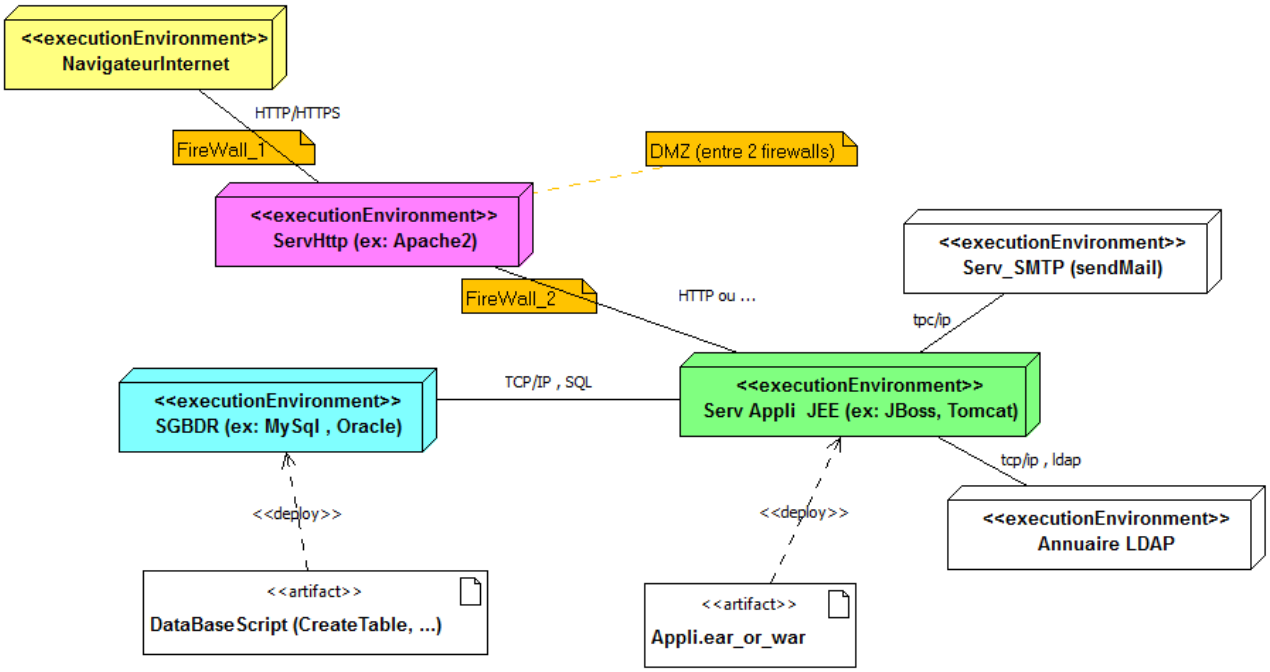
1.3. Diagrammes de composants



Sous diagramme (composants dans un des modules):

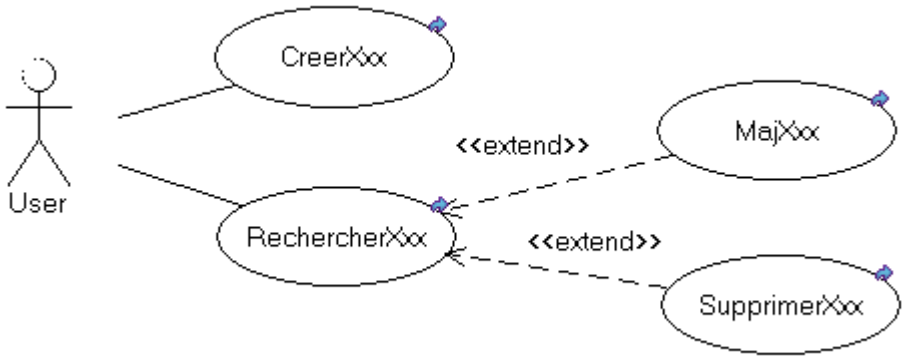


1.4. Diagramme de déploiement (topologie) :



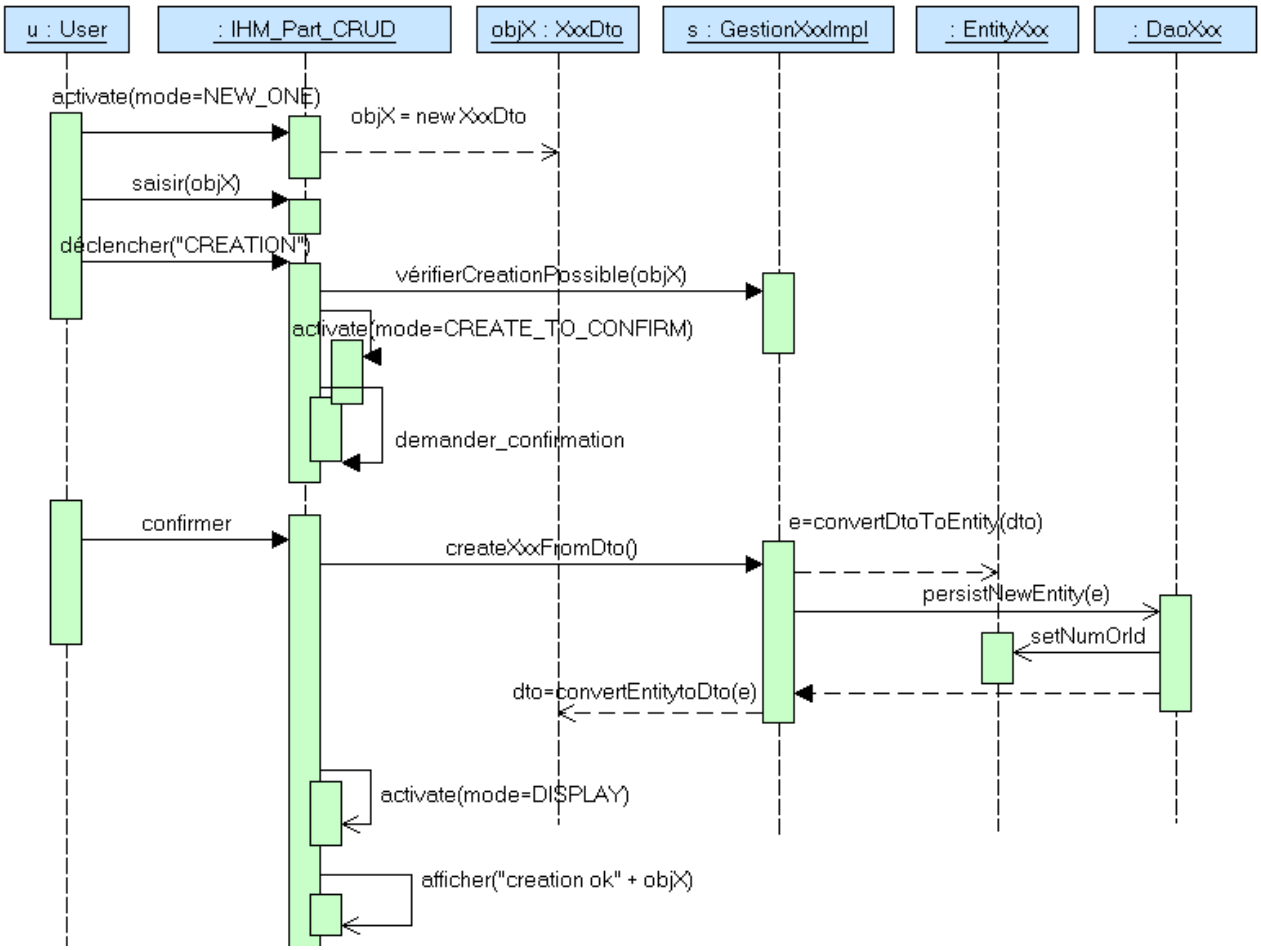
1.5. Uses Cases et séquences techniques/génériques :

Pour la syntaxe et l'idée : un exemple de diagramme de "Uses Cases" génériques :

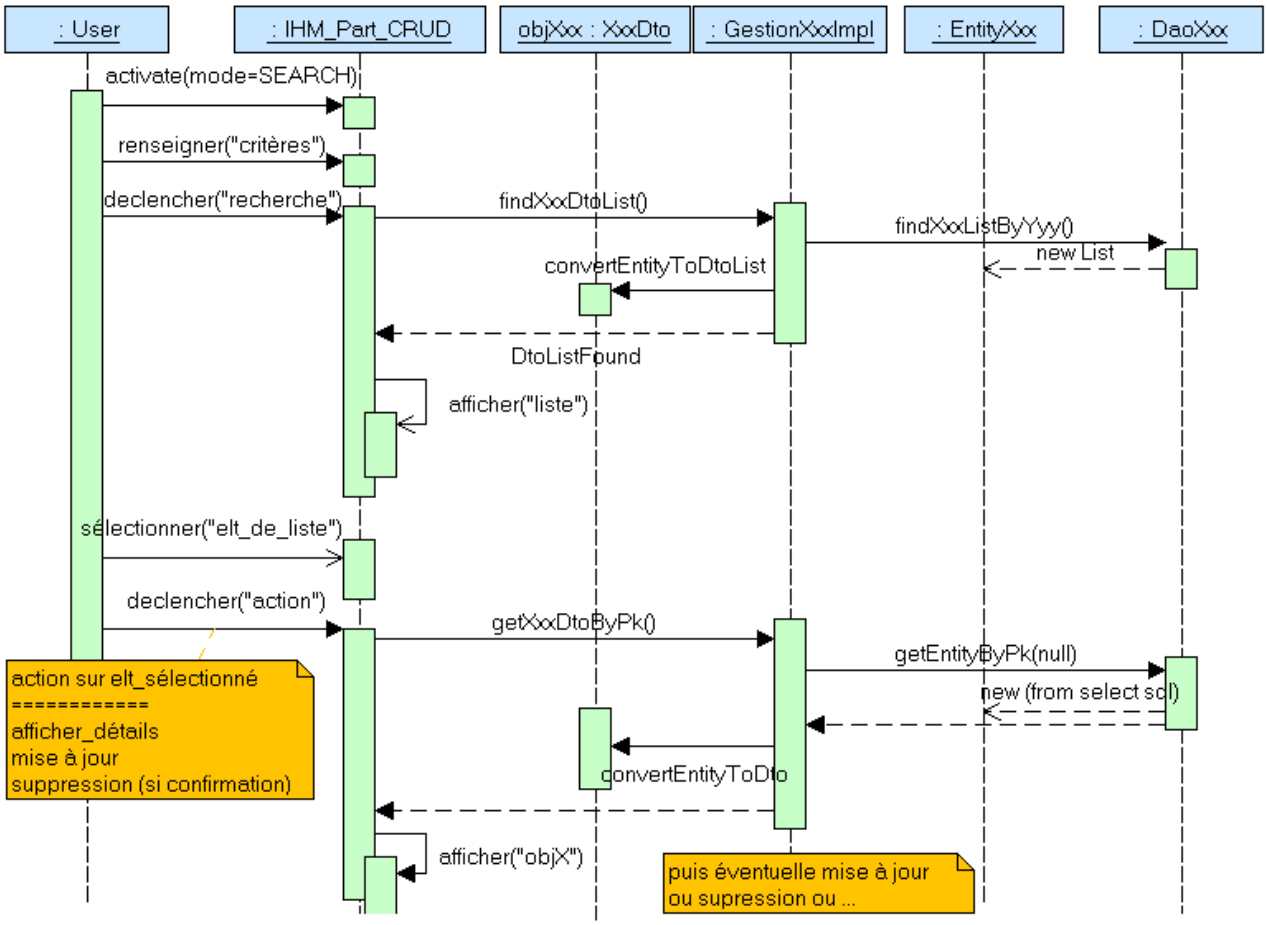


Ceci permet (au sein d'un logiciel UML élaboré) de naviguer intuitivement vers des diagrammes de séquences techniques montrant une façon de réaliser une des opérations CRUD classiques.

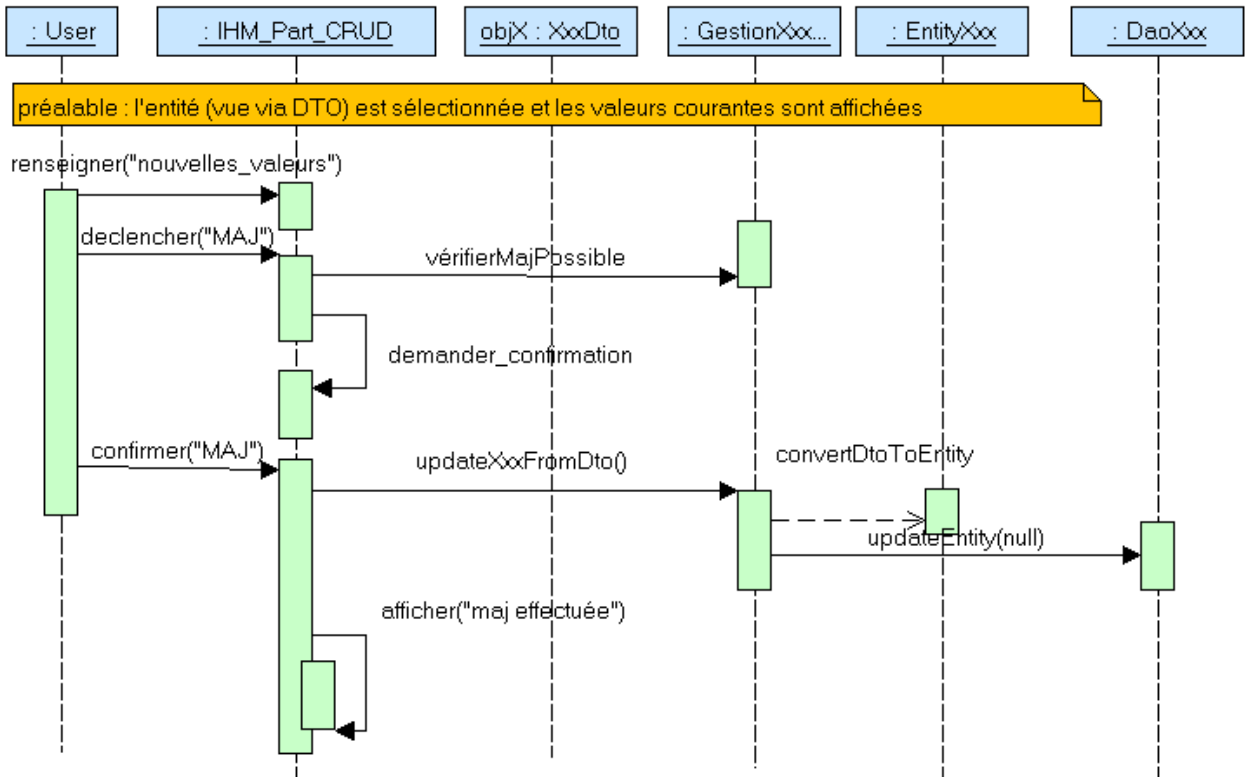
Exemple "seq_create_xxx_avec_dto_et_confirmation" :



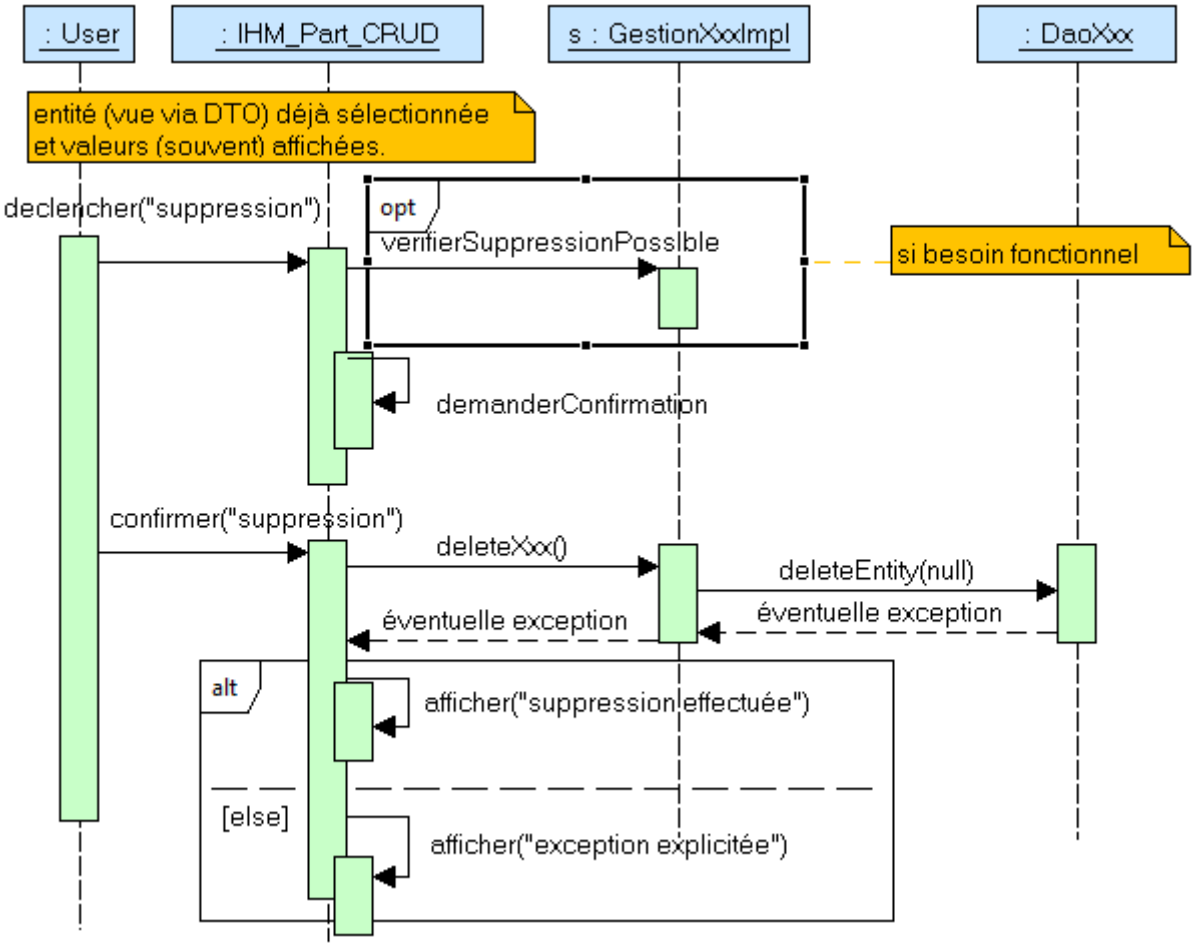
diag_seq_rechercher_xxx_with_dto (liste + sélection + details) :



diag_séquence_Mise_a_jour_Xxx_with_dto_et_confirmation :



diag_sequence_supprimer_Xxx_avec_dto_confirmation_et_notification :



2. Conception spécifique à un système fonctionnel

---> Au cas par cas .
---> séparer si possible les parties à transactions courtes , des parties "non transactionnelles"
---> Eventuelle projection (en partie automatique) via MDA et paramétrée par des annotations

…/...

Exemple d'urbanisation élémentaire (ventes en ligne) :