



UFR Sciences et Techniques

Master 2^{ème} année STIC spécialité professionnelle

« Bases de Données – Intelligence Artificielle »

Systeme d'Information Orientés Objets

Préparé par Elie MATTA

Copyright © 2010-2011, eliematta.com. All rights reserved

Table des matières

Chapitre 1: UML: Based web engineering	3
Introduction:	3
1) Caractéristique du processus.....	3
2) Meta-modélisation	5
Chapitre 2: Modélisation des exigences: cas d'utilisation	5
Chapitre 3 : Modelisation des flux: Diagramme d'activités.....	9
1) Base des diagrammes d'activités.....	9
2) Activités et actions.....	10
3) Décisions et confluences:	10
4) Effectuer plusieurs tâches à la fois	11
5) Evénements temporels.....	11
6) Appels à d'autres activités	11
7) Les objets	12
8) Envoi et réception de signaux.....	12
9) Fin de flux.....	13
10) Partitions et couloirs.....	13
Chapitre 4: Modélisation de la structure logique: Diagramme de classes.....	14
1) Les classes	14
2) Visibilités	14
3) Multiplicité.....	14
4) Comportement d'une classe: Operations.....	14
5) Relation entre classes	15
6) Les contraintes.....	16
7) Classes abstraites.....	16
8) Interfaces	16
Chapitre 5: Adaptation d'UML: Les profils.....	17
1) Qu'est-ce qu'un Profile?	17
2) Les stéréotypes	17
3) Les valeurs marquées	18
4) Créer un profil.....	18

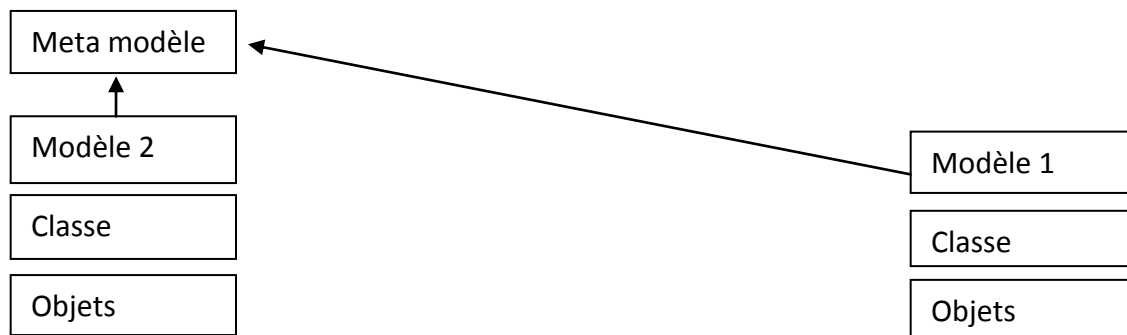
5) Utiliser les profils	19
Chapitre 6: Le méta-modèle UWE	19
1) Le paquetage "Besoins"	20
2) Le paquetage "contenu"	20
3) Le paquetage "Navigation"	20
4) Le paquetage "présentation"	22
5) Le paquetage "Processus"	23
a) Intégration des processus.....	23
b) Définitions de l'interface utilisateur	24
c) Définition du comportement	24
Chapitre 7 : La méthode par l'étude d'un cas.....	29
Introduction	29
Description	29
Objectif.....	29
1) Commencer avec la spécification des besoins	29
2) Définir le contenu	31
3) La structure de navigation	32
a) Esquissé le modèle de navigation.....	32
b) Modèle de navigation.....	33
c) Ajoute le processus métiers.....	34
4) Raffinement du modèle de processus.....	35
5) Esquisse la présentation	36
Exemple 1:.....	38
Exemple 2 - TP1.....	41

Chapitre 1: UML: Based web engineering

Introduction:

UWE: UML Web Engineering en 1999

- Les objectifs:
 - Trouver une méthode d'analyse et conception de systèmes web sur les méthodes (OOHDM, RMM, WSDM)
- Entre temps, l'UML devient la "langue franca", c'est l'ingénierie logiciel (Object management group)
- Domains-specific modeling languages (DSL), mécanisme d'extension avec des spécifications web
- UWE utilise:
 - UML (Mécanisme d'extension)
 - XMLI (format d'échange de modèle)
 - MOF (méta modélisation)
 - MDA (Model driven architecture)
 - QVT (langage de transformation)



On fait le méta modèle pour traduire les sous-modèles (comme modèle 2) pour faire communiqué modèle 1 avec modèle 2. Dans ce cas alors, on doit traduire les objets et classes aussi.

1) Caractéristique du processus

- Le développement des systèmes web change constamment (à cause de l'utilisateur ou besoin technologiques,...)
- Les modèles doivent être adaptables à ces changements
- Pour répondre efficacement à ce besoin de flexibilité UWE sépare les phases de développement et implémente un processus de développement guidé par les modèles
 - Construire 1 modèle
 - Triangle d'un modèle

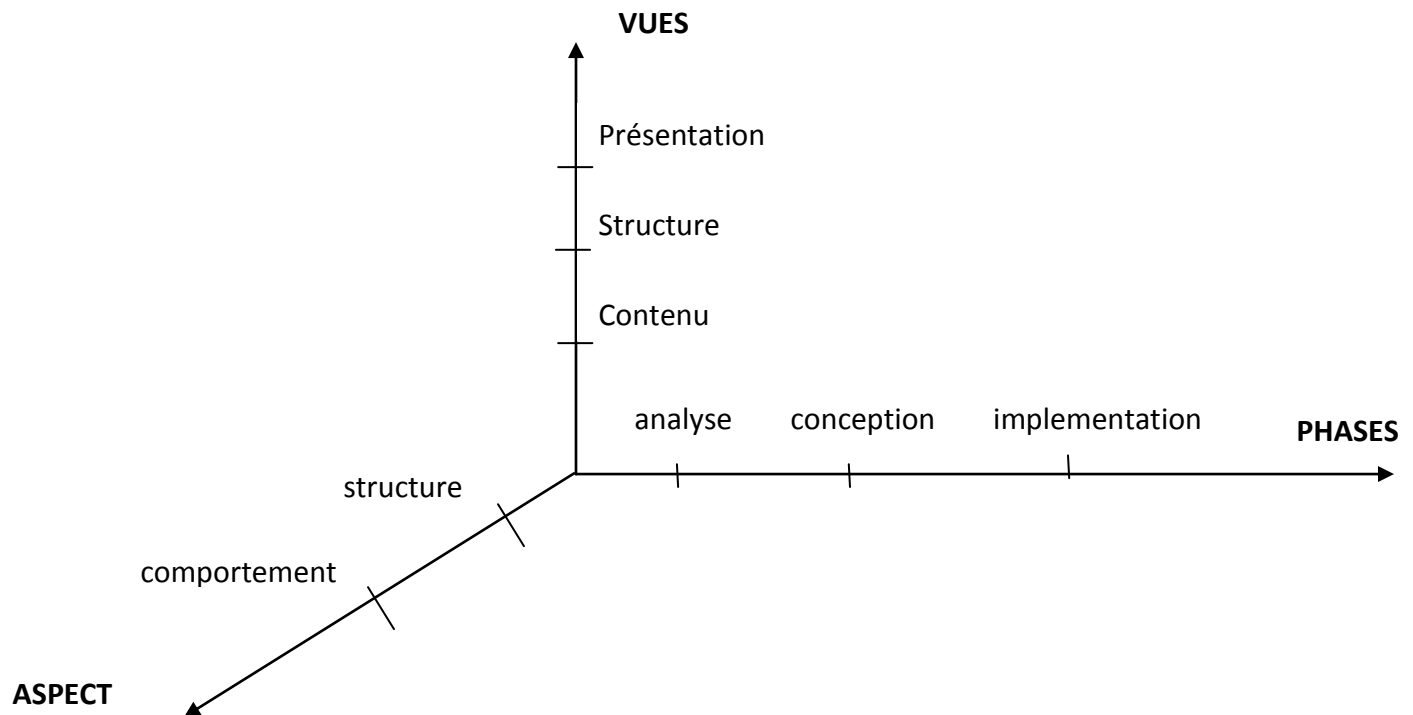
=> **Objectif final: génération automatique de code**

a) Séparation des besoins

i. Les modèles sont construits à travers différentes étapes

1. Analyse
2. Conception
3. Implémentation

ii. Les modèles sont des **VUES** sur la même application web pour les différents besoins (contenu, structure de navigation, présentation)



b) Caractéristique de la notion UWE

Modélisation: vues statiques et dynamiques

- Diagramme
 - d'objet
 - de classes
 - de composantes
 - de déplacement
 - cas d'utilisation
 - d'activité
- Le mécanisme d'extension d'UML est utilisé pour des stéréotypes (IHM pour une fenêtre par exemple) sur nœuds et liens.

- Des étiquettes et des contraintes OCL
- Les 3 points précédents permettent de définir un profil UML => Notation UWE est un profil UML

2) Meta-modélisation

- Un méta-modèle est une définition précise des éléments d'un langage de modélisation, de leurs relations et les règles d'écriture
- Meta-modèle UWE est un méta modèle conservatif

Chapitre 2: Modélisation des exigences: cas d'utilisation

- Définition: On crée pour chaque exigence un cas d'utilisation, c'est à dire un élément dans le diagramme.
- Les cas d'utilisations sont des besoins fonctionnels
- Nom fonctionnel (ex: performance, langage de programmation, etc)
- Il n'existe aucune "manière recommandée"

1) A l'extérieur du système: **les acteurs**

- **Notation UML**

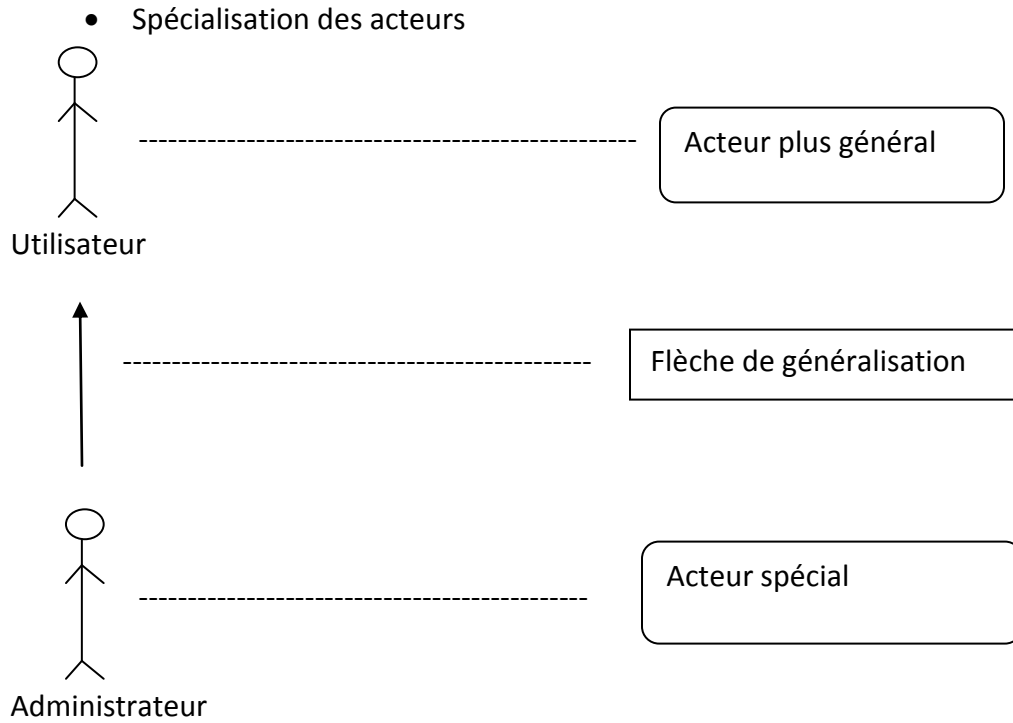


"Acteur" (Les personnes sont actives mais pas passives)
Administrateur



Administrateur

- Pas nécessairement des personnes (ex: System B2B (business to business))
- Un acteur ne peut pas influencé (héritage)



2) Cas d'utilisation

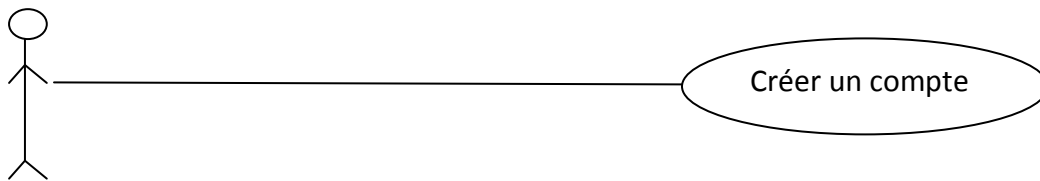
- A partir des exigences des utilisateurs
- Un cas est une utilisation du system

Exemple:

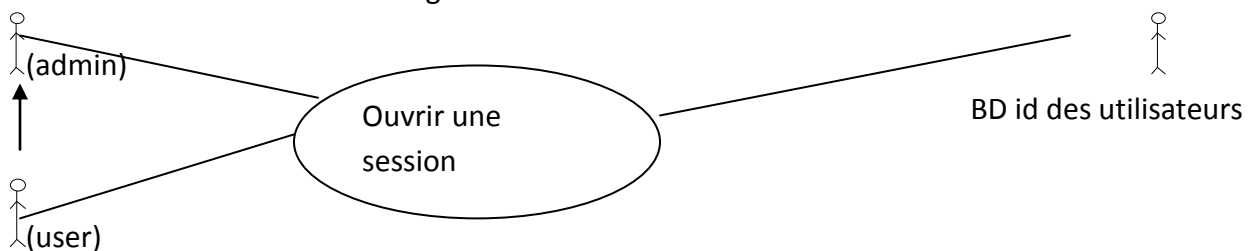
Créer un compte

3) Lignes de communication

- Une ligne connecte un acteur et un cas d'utilisation

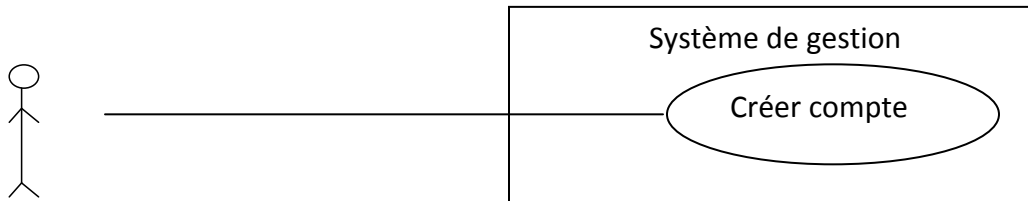


- Pas de limite de lignes de communication



4) Frontière du système:

- c'est une boîte autour des cas d'utilisations
- Acteur en dehors
- boîte est nommée en fonction du system développé



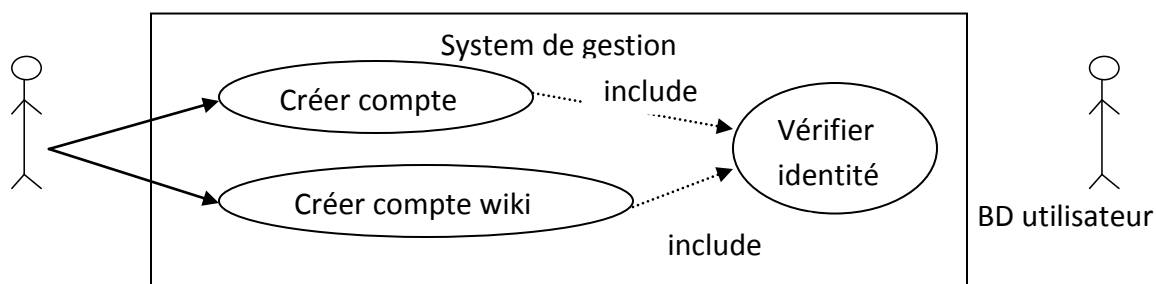
5) Description de cas d'utilisation

- UML ne fixe pas les règles, mais il est bon d'écrire un maximum
 1. Exigence associées
 2. Objectif dans le contexte
 3. Préconditions
 4. Condition de succès
 5. Condition d'échec
 6. Acteur principaux
 7. Acteur secondaires
 8. Déclencheur
 9. Flux principal
 10. Extensions

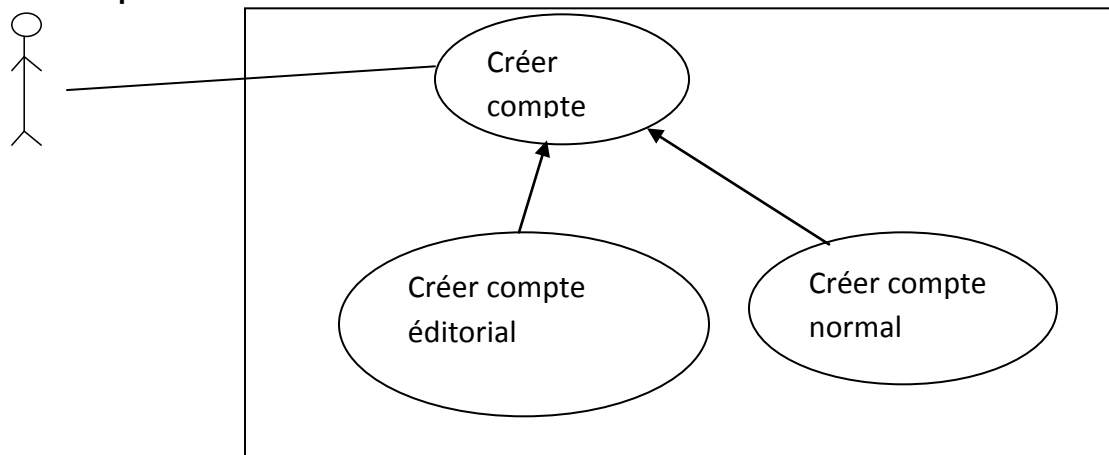
6) Relations de cas d'utilisation

- Il est possible de montrer des comportements
 - Réutilisabilité "includes"
 - Facultatif "extend"
 - Spécialisés ↗

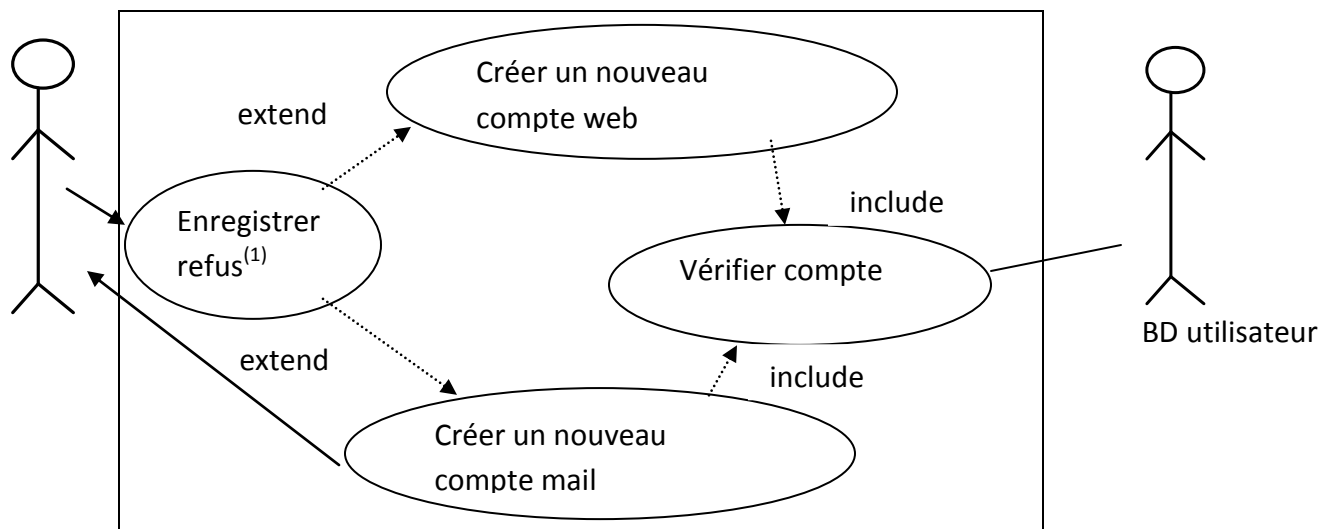
➤ La relation "include"



➤ Spécialisation: Généralisation de cas d'utilisation



➤ La relation d'extension

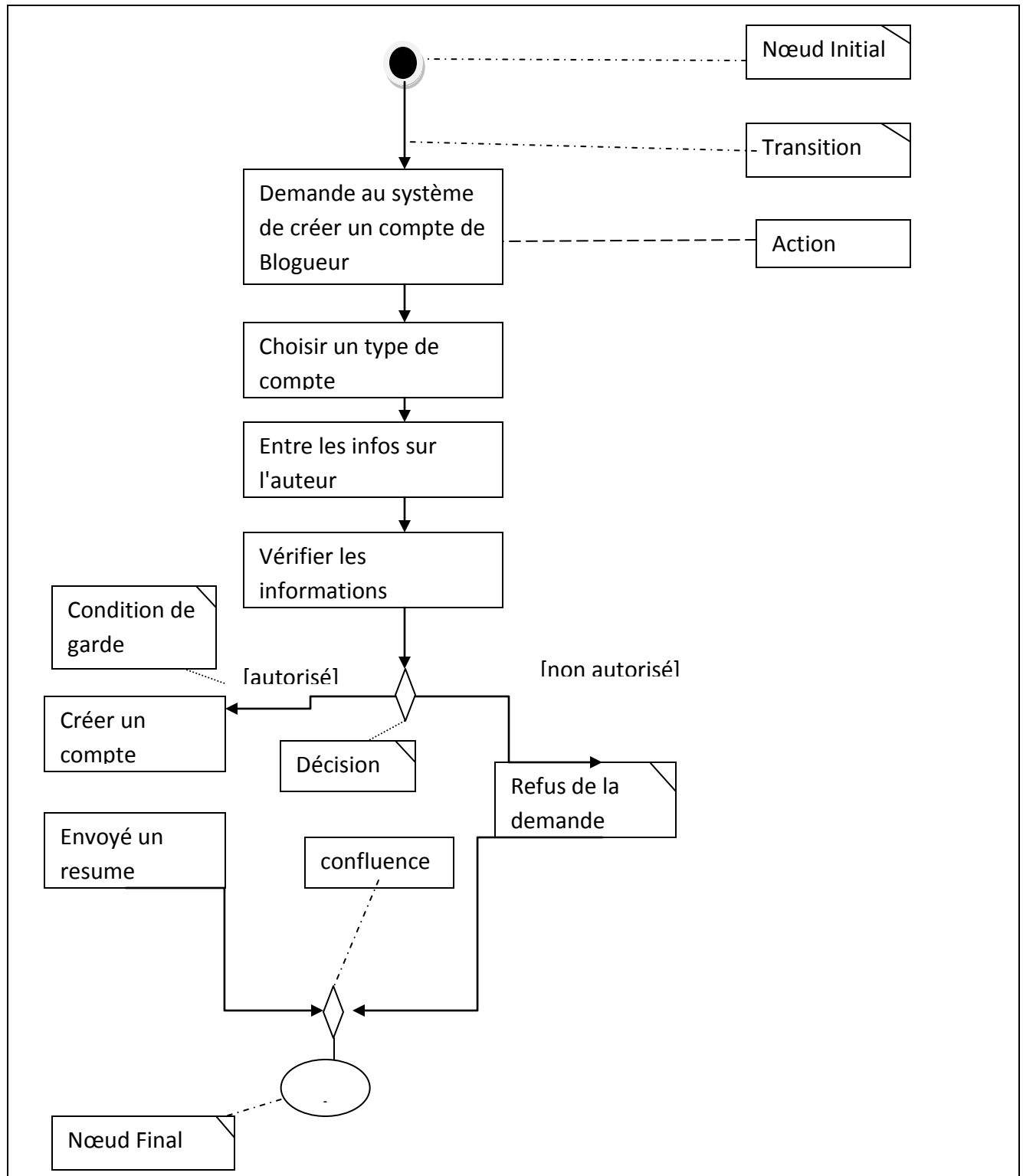


- Extend: On met extend quand nous ne sommes pas sûrs du cas (Le refus de l'enregistrement = Enregistrer refus⁽¹⁾ par exemple)
- Include: c'est sûr qu'on doit passer par le cas (vérifier compte par exemple)

Chapitre 3 : Modelisation des flux: Diagramme d'activités

Précise comment le système accomplit les tâches

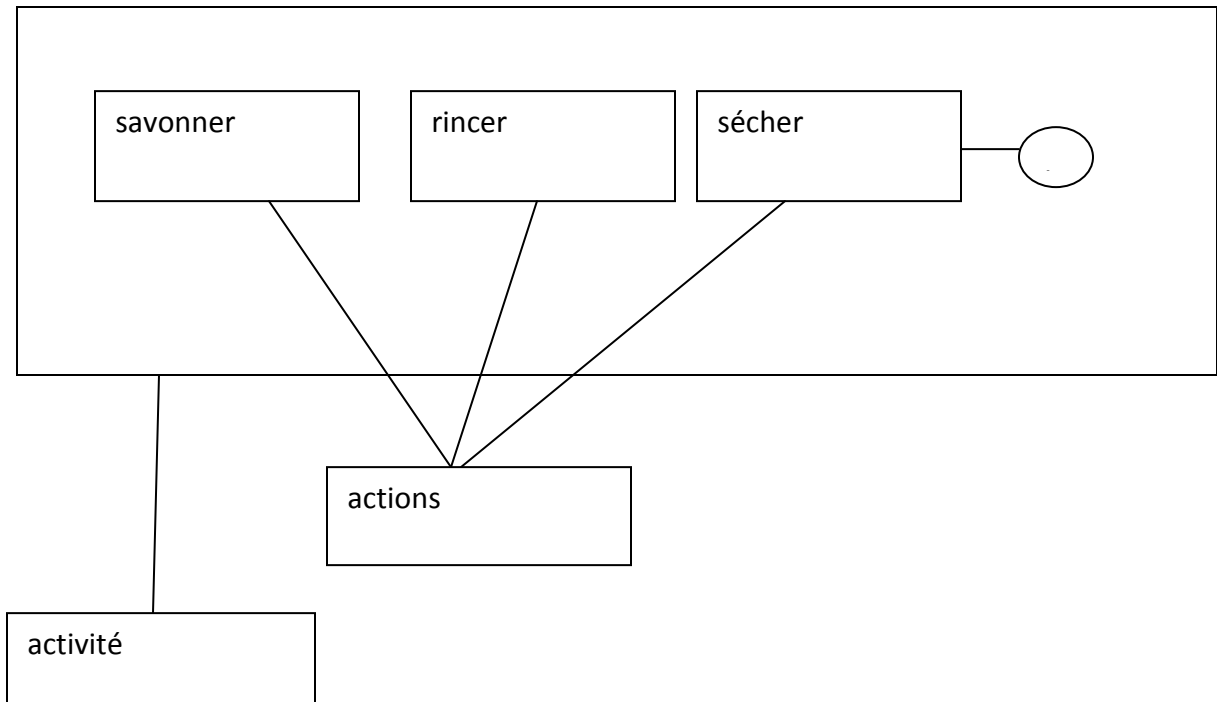
1) Base des diagrammes d'activités



2) Activités et actions

- Le terme "activités" est souvent mal employé à la place du terme "action"
- L'activité représente le processeur à modéliser (ex: laver une voiture)
- première action est une étape de l'activité globale

Exemple



3) Décisions et conflues:

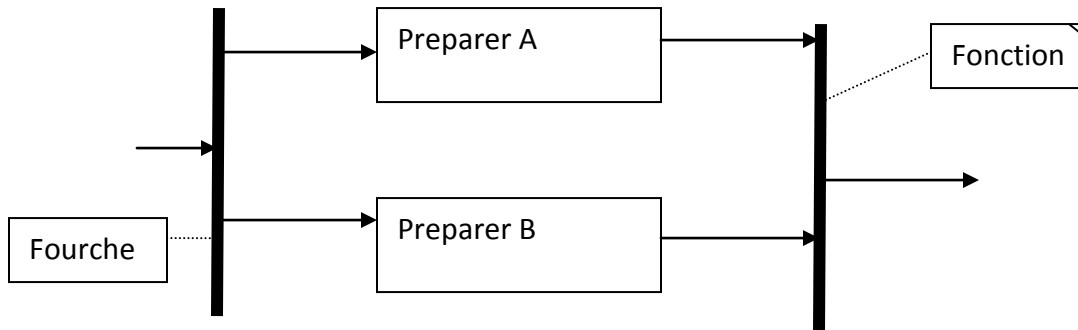
[autorise]: Si l'évaluation est vraie, alors suivre cette transition

[nb >= 10]: Si la valeur de la variable est >= 10 alors suivre cette transition



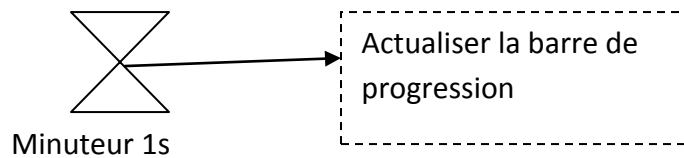
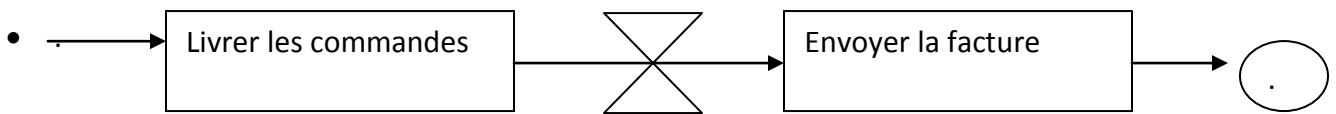
4) Effectuer plusieurs tâches à la fois

- Les actions concurrentes se présentent sous la forme de fourches



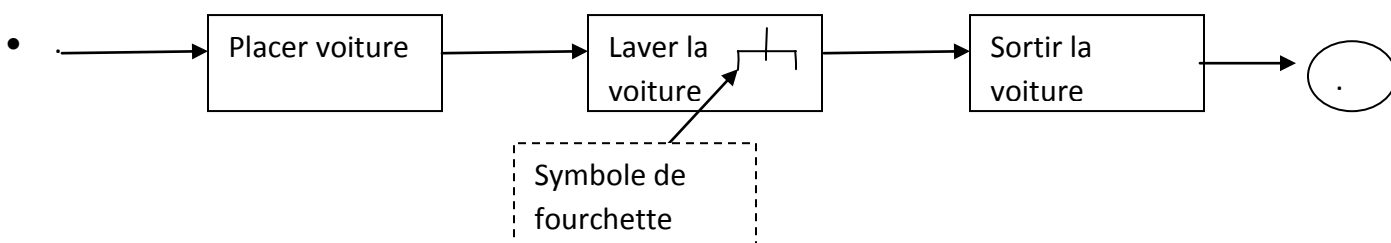
5) Evénements temporels

- Représentés par un sablier



6) Appels à d'autres activités

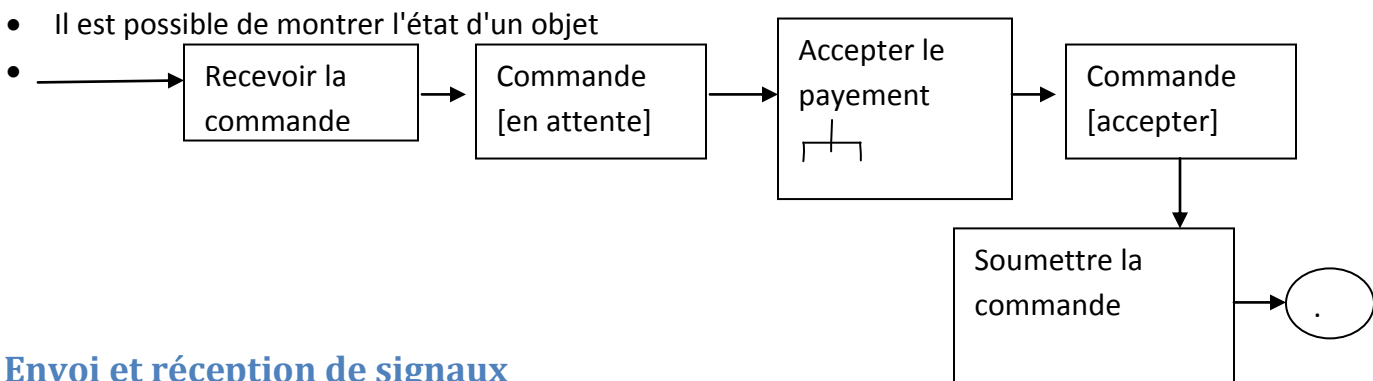
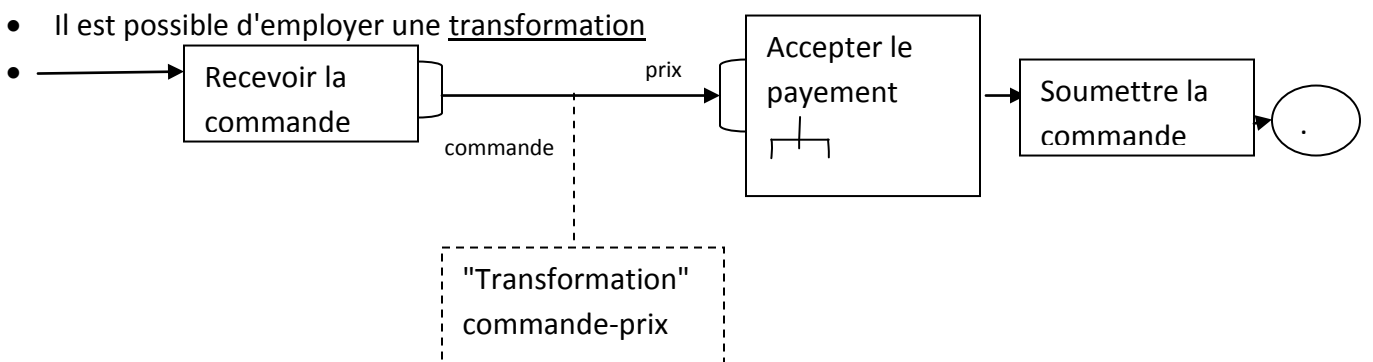
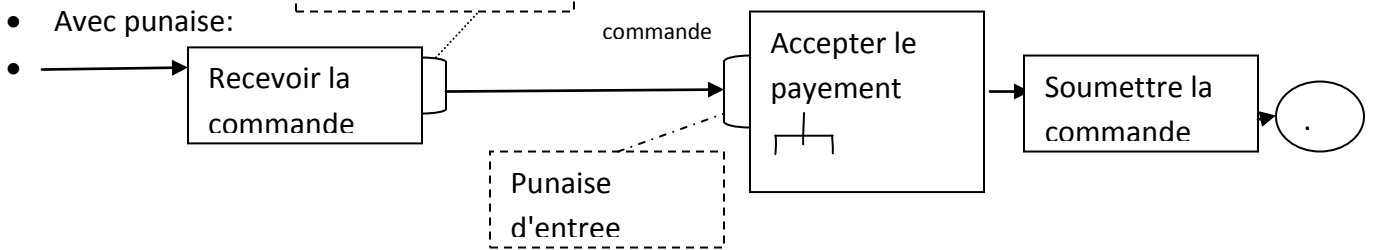
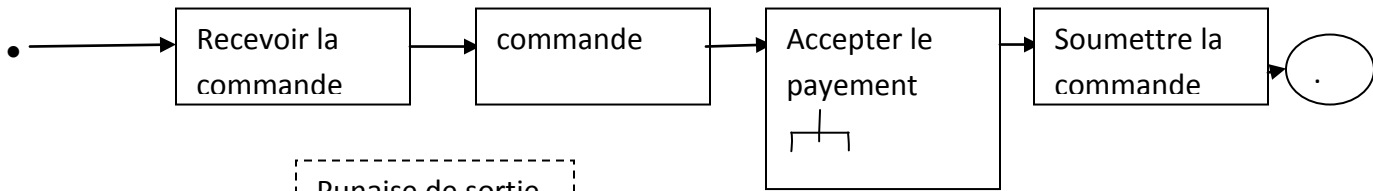
- Trop de détails
 - même séquence
- améliore la lisibilité



! Cadre d'activité obligatoire pour l'appel

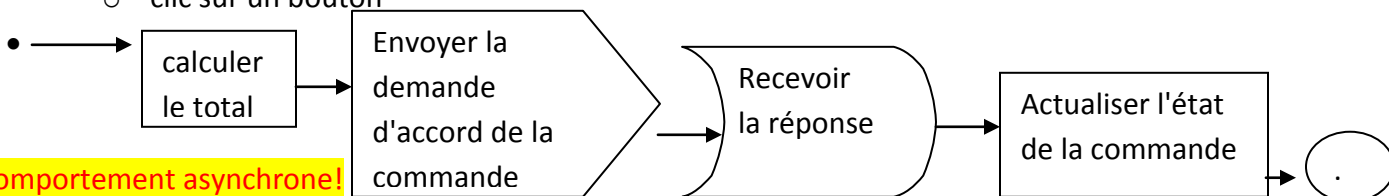
7) Les objets

- Actions avec un besoin d'information
- Utilisation d'un objet



8) Envoi et réception de signaux

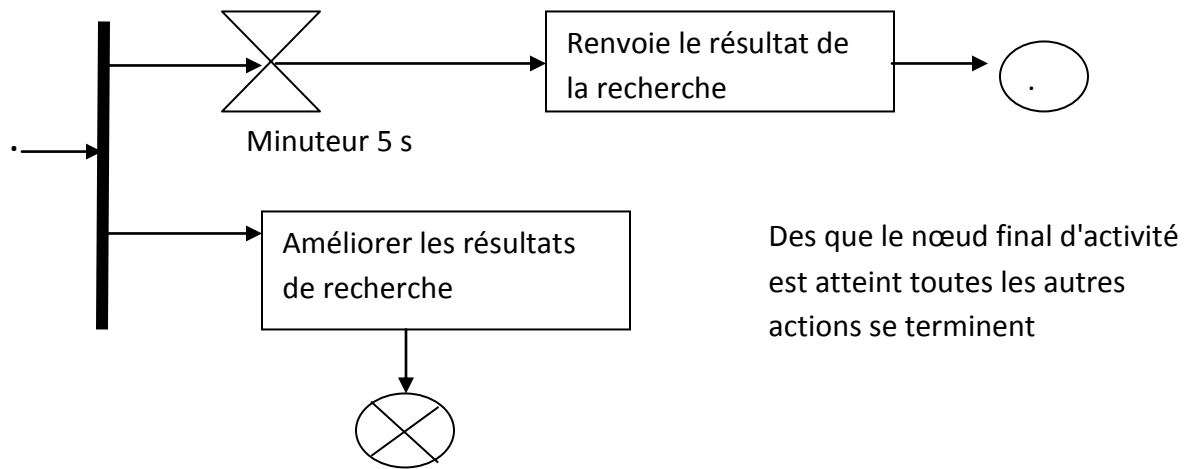
- les signaux représentent des interactions avec des parties externes
 - réception d'une commande déclenche son processus
 - clic sur un bouton



! Comportement asynchrone!

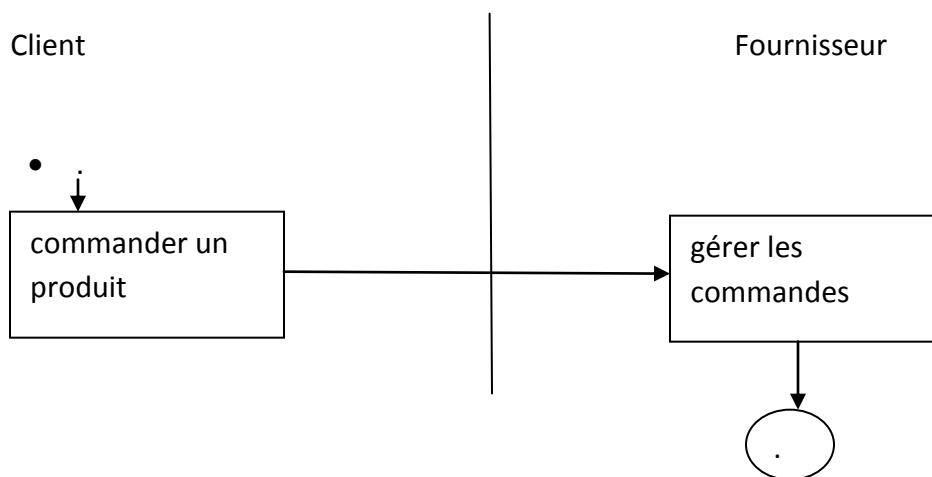
9) Fin de flux

- Un nœud final de flux clôt sa propre transition et non l'action globale



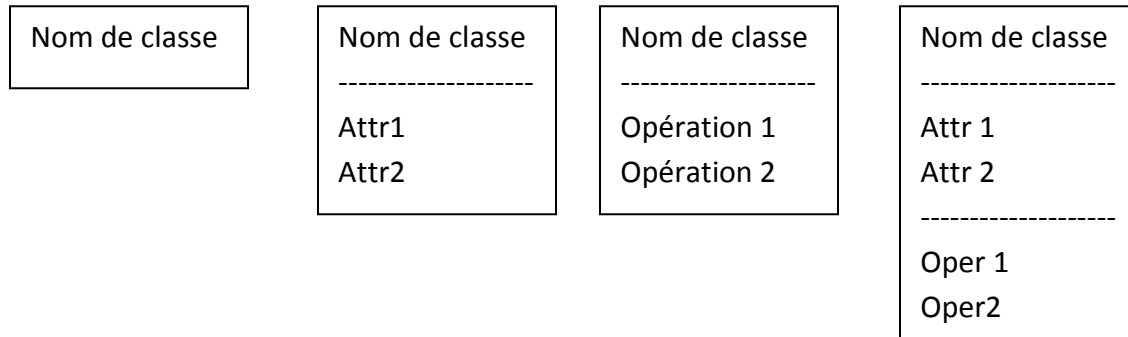
10) Partitions et couloirs

- Certaines activités impliquent plusieurs participants
 - groupes
 - rôles

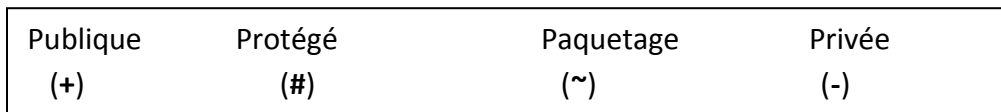


Chapitre 4: Modélisation de la structure logique: Diagramme de classes

1) Les classes



2) Visibilités



(+): Attribut et opération accessibles par **toutes les classes**

(#): Attribut et opération accessibles par **la classe et les classes filles**

(~): Attribut et opération accessibles par **les mêmes paquetages**

(-): Attribut et opération accessibles par **la classe**

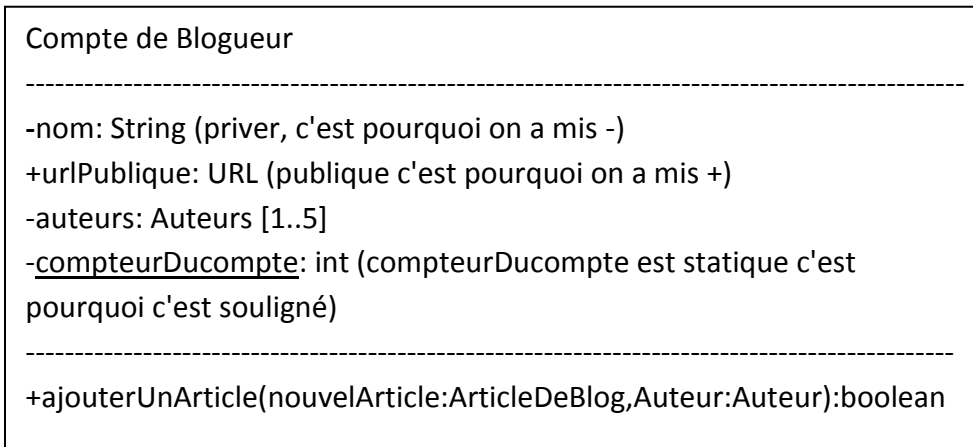
3) Multiplicité

- un attribut qui représente un ensemble d'objets

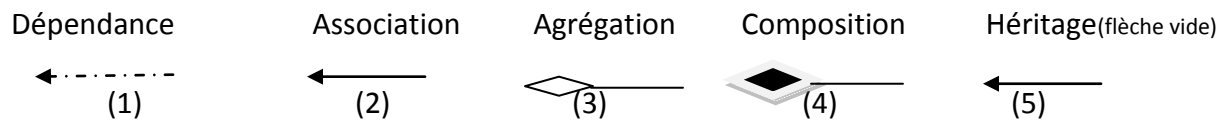


4) Comportement d'une classe: Operations

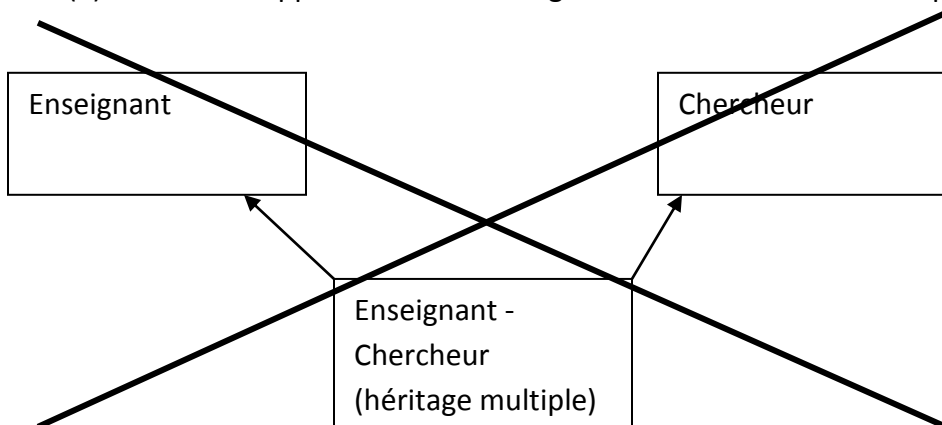
- Paramètres
- Type de retour
- Visibilité



5) Relation entre classes



Si on a (3) alors un compte de blogueur possède plusieurs articles de blog
Si on a (4) alors si on supprime article de blog alors on doit enlever le compte de blogue



6) Les contraintes

- Permet de restreindre la manière dont une classe opère
- Pour cela langage OCL
- Permet 3 types de contraintes
 1. Invariants: Toujours vrai
 2. Précondition: Avant
 3. Postcondition: Après

Exemple:

```
Article De Blog
-----
-URL: url
-classement: int {classement >= 0}
(exemple sur 1. Invariants)
+actualiserClassement(nouveacuClass
ement:int):void
```

```
self.url --> nonVide()
(exemple sur 1. Invariants, car cette condition doit
être toujours vrai)
```

```
Contexte ArticleDeBlog: actualiser(...):void
(exemple sur 2. Précondition: pre.classement >= 0)
(exemple sur 3. Postcondition: post-classement
<=5)
```

7) Classes abstraites

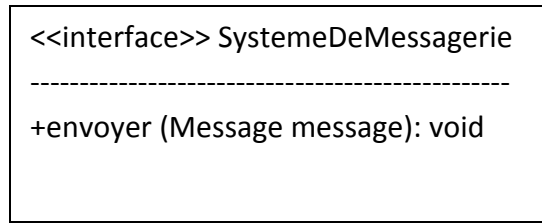
```
Stockage
-----
+stock (Article:Article[]): void
+retourner(): Article[]
```

```
public abstract void stocker (Article[] article)
```

8) Interfaces

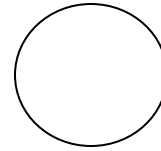
- Si vous souhaitez déclarer les méthodes que les classe concrètes, doivent implémenter, mais sans utiliser l'abstraction
- Une interface est une classe abstraite dont aucune méthode n'est implémentée

! JAVA



notion stéréotype

ou



SystemeDeMessagerie
+envoyer (Message message): void

Chapitre 5: Adaptation d'UML: Les profils

- Les systèmes partagent des caractéristiques
 - structurelles
 - comportementales

Mais quand on arrive aux plates-formes et aux domaines ils présentent souvent de nombreuses différentes terminologiques

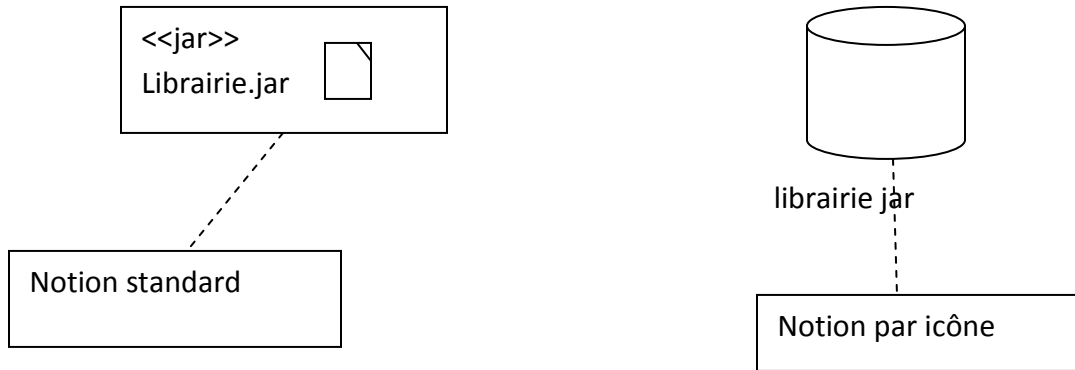
J2EE	.NET
EJB, JAR, JSP	MOO, Assembly, ASP

1) Qu'est-ce qu'un Profile?

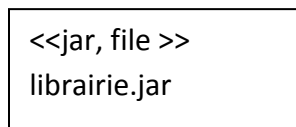
- Une manière simple pour adapter l'UML
- Ou un domaine (médecine, finances, web, etc...)
- Composés
 - de stéréotypes
 - de valeur marquée
 - ensemble de contraintes
- Les outils de génération de code peuvent exploiter les profiles
- Dans UML 1 --> Définition à la volée
 - Confusion sur les conditions d'emploi
- Dans UML 2 : Les stéréotypes et valeurs marquées sont regroupées dans un profile

2) Les stéréotypes

- **Stéréotype signifie qu'un élément a une utilisation spécifique**

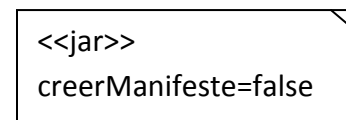
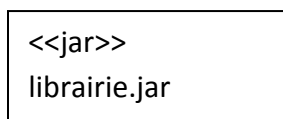


- Pas de limite au nombre de stéréotypes

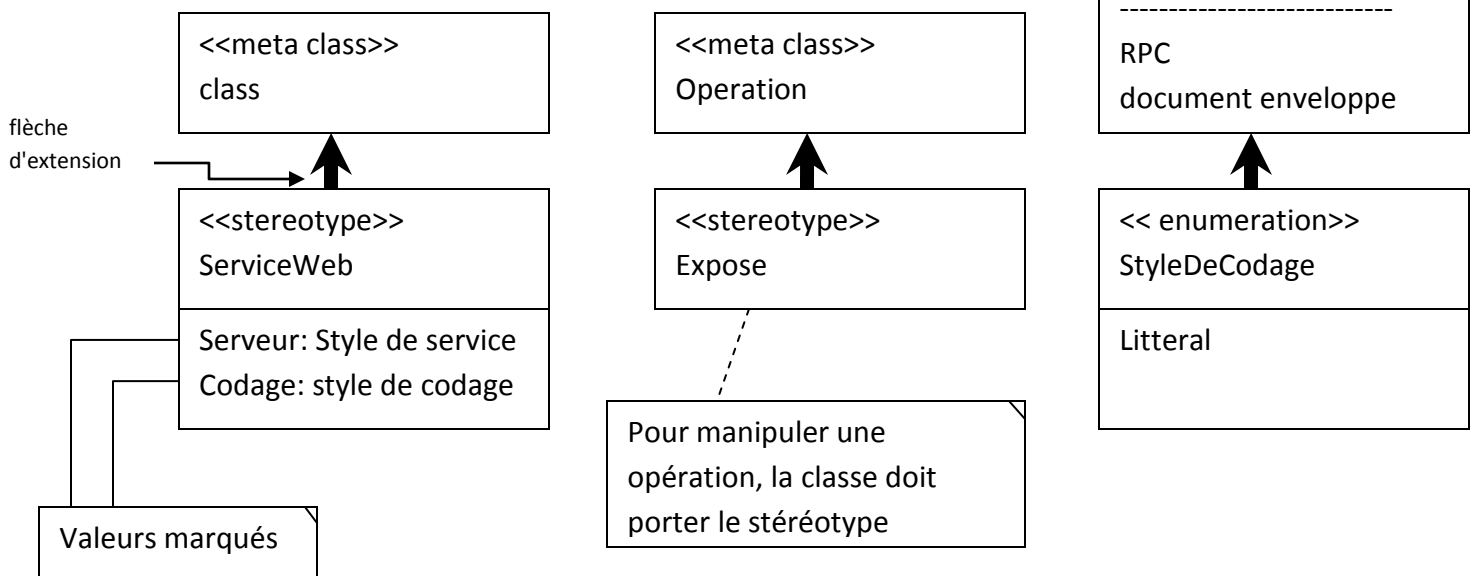


3) Les valeurs marquées

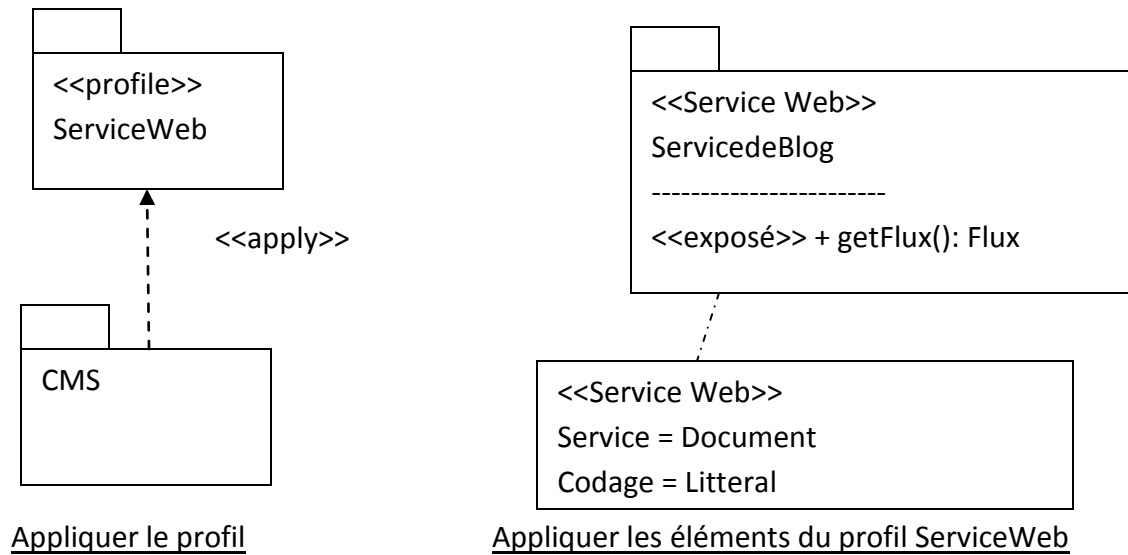
- Plusieurs valeurs marquées peuvent être associées à des stéréotypes
 - Informations supplémentaires
- Elle se présente sous la forme d'une note attachée



4) Créer un profil

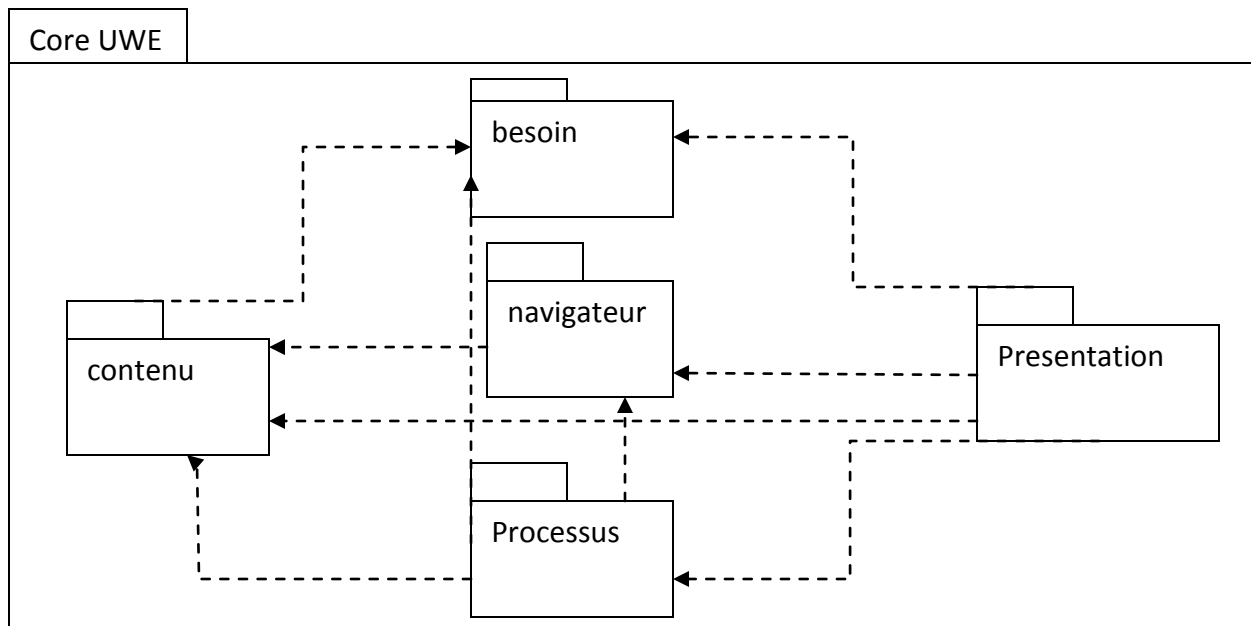


5) Utiliser les profils



Chapitre 6: Le méta-modèle UWE

- Le méta-modèle UWE est une extension conservatif du méta-modèle UWE
 - Les éléments du méta-modèle ne sont pas modifiés
- Tous les éléments du méta-modèle d'UWE sont liés par héritage au méta-modèle UML 2.0.
- Le méta-modèle est profilable car il est possible de faire la correspondance entre ce méta-modèle et un profil UML.



1) Le paquetage "Besoins"

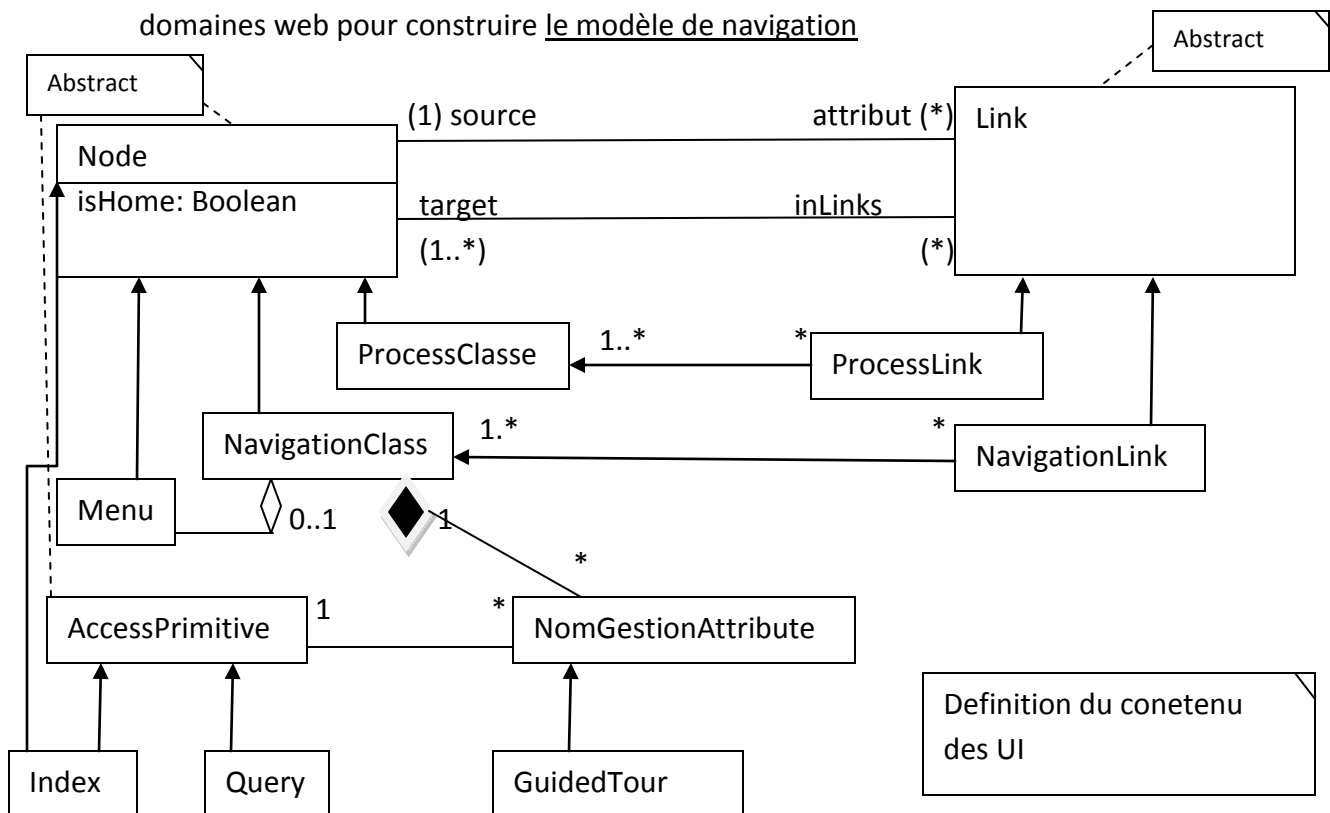
- Il fait référence aux cas d'utilisation
- Il fait la différence entre :
 - Processus métier
 - Personnalisation
 - navigation

2) Le paquetage "contenu"

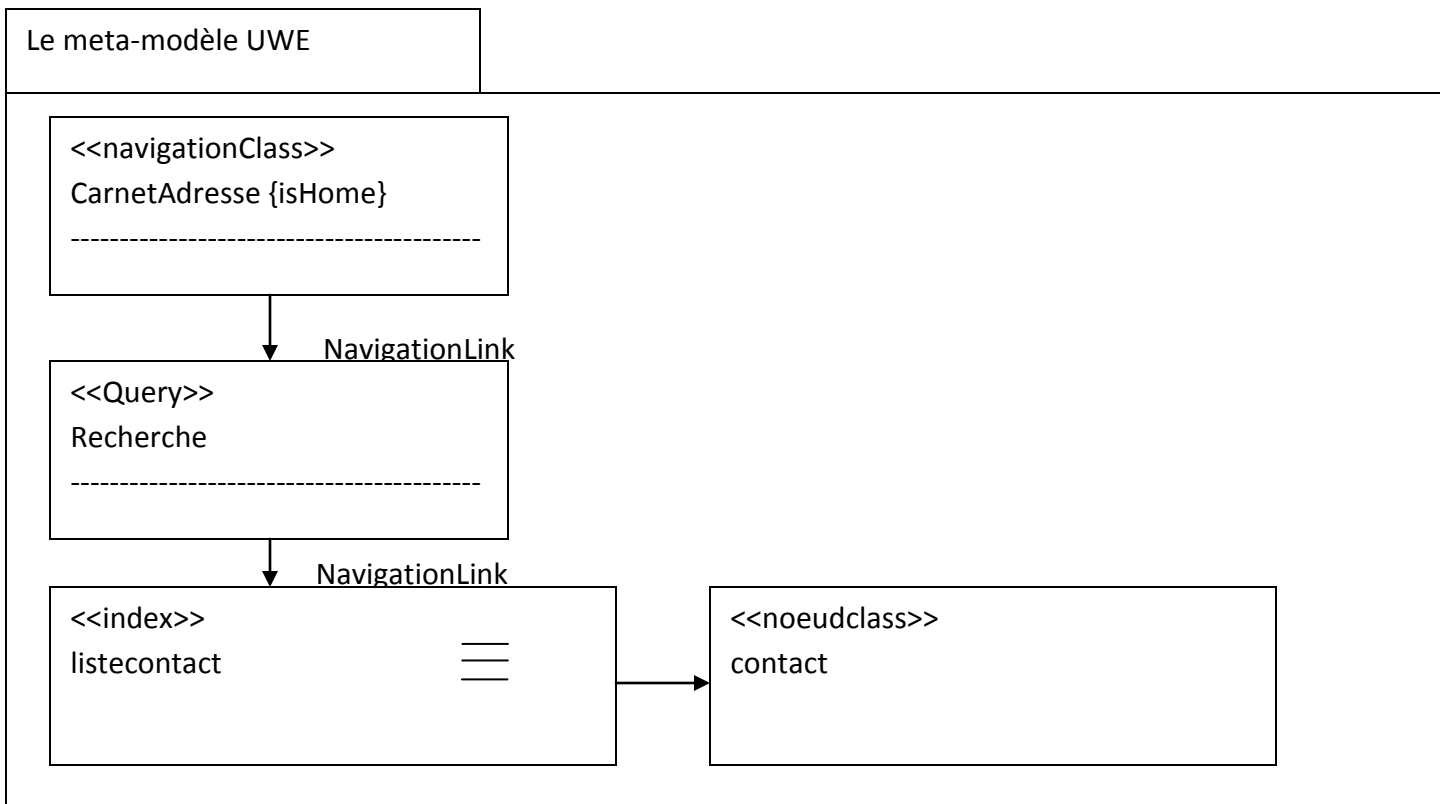
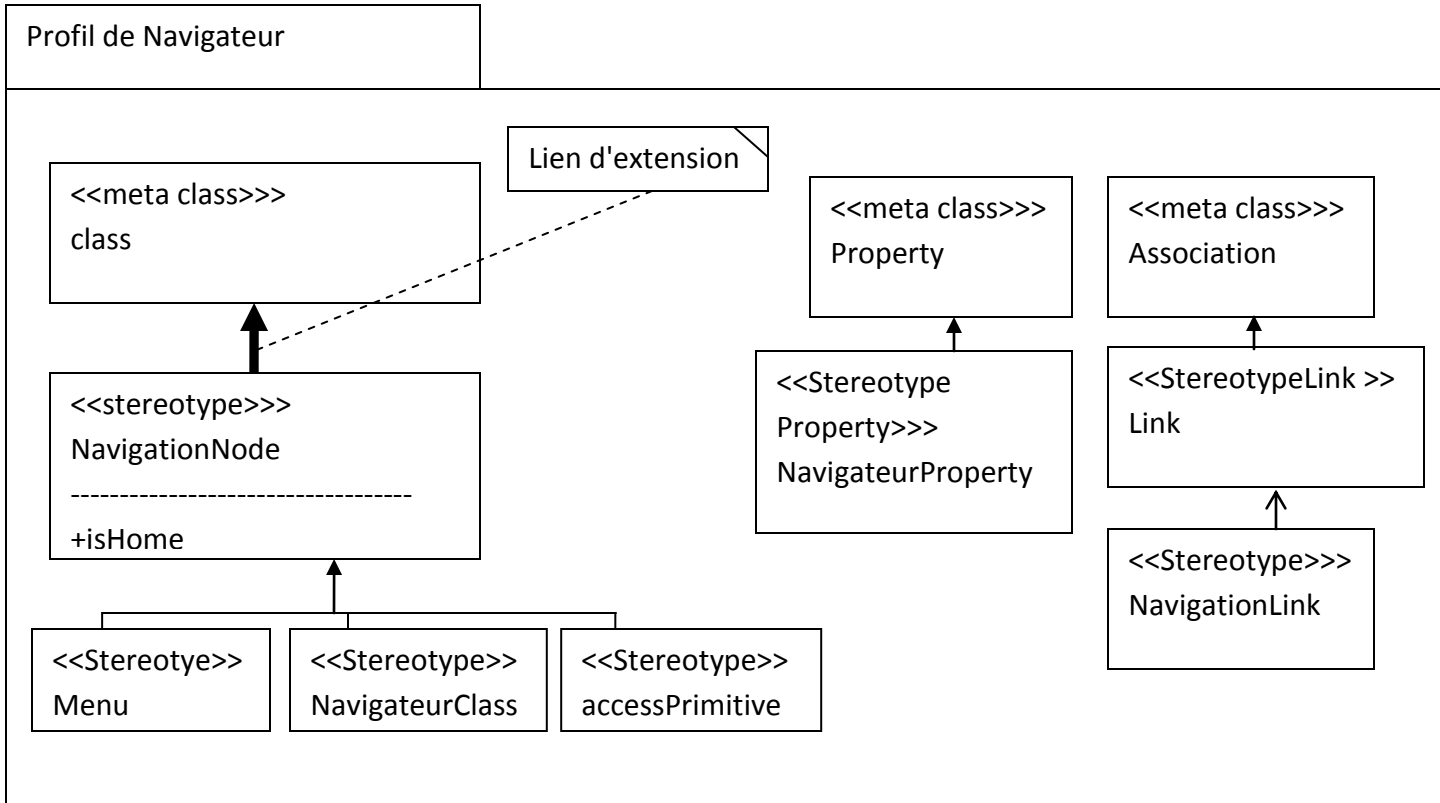
La modélisation du contenu des applications web avec UWE sont équivalente aux applications non-web

3) Le paquetage "Navigation"

- Le méta-modèle se base sur une paire de méta-classes abstraites et des associations entre ces deux classes
 - Node
 - Link
- L'ensemble de sous-classes de Nœud et lien fournit des meta-classes spécifiques, du domaines web pour construire le modèle de navigation

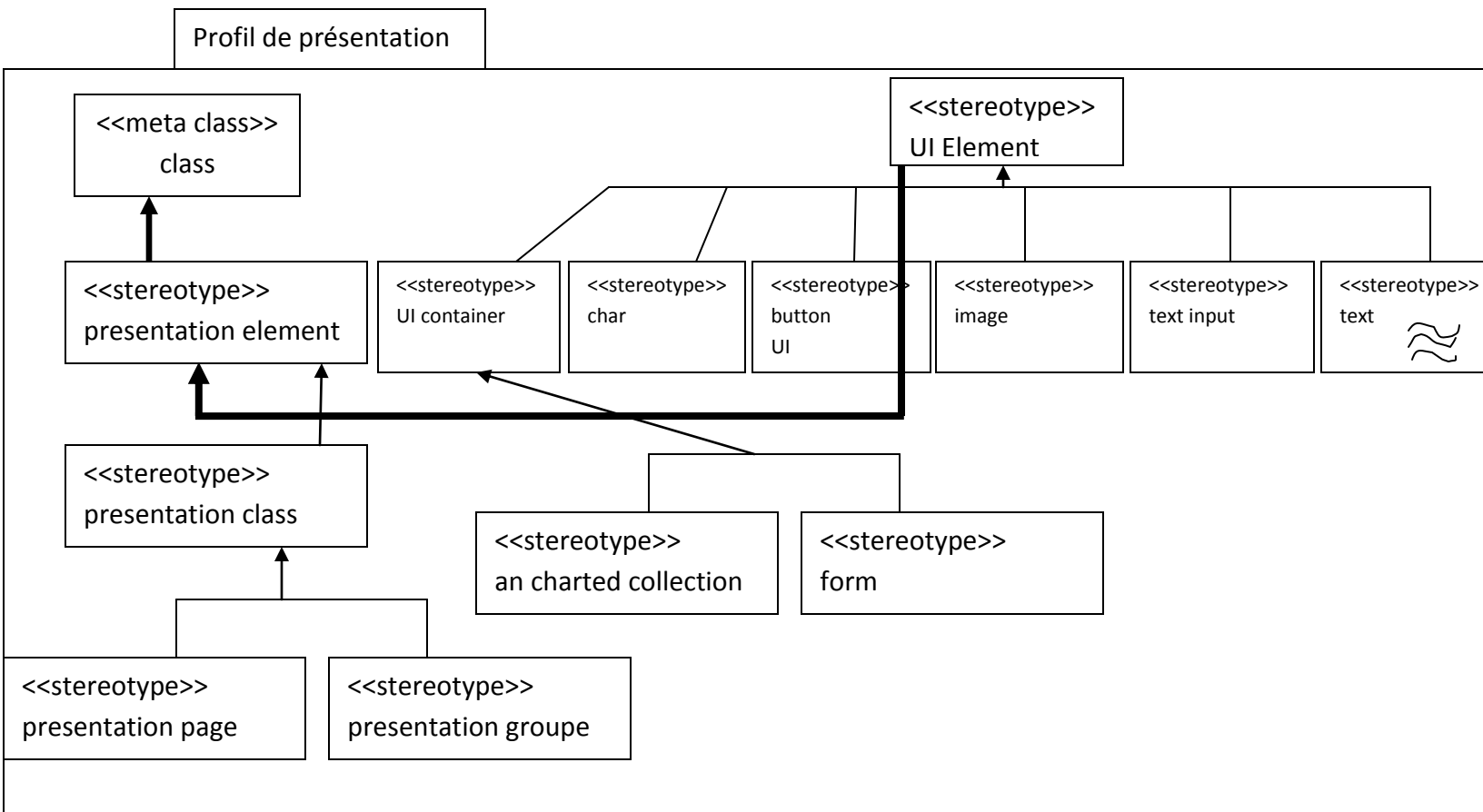


Le paquetage de navigation

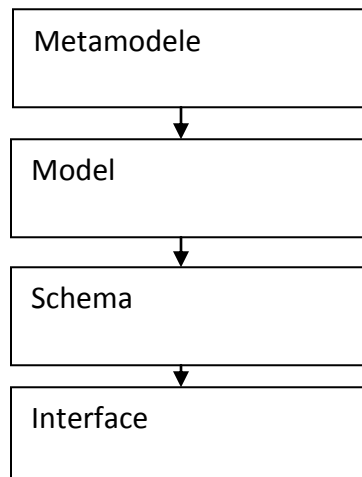


4) Le paquetage "présentation"

- Fournit une UWE abstraite de l'interface utilisation (UI) de l'application web.
- Fait abstractions des éléments plus concrets comme la couleur ou le placement
- Aucune définition du rendu des éléments utilisateurs.

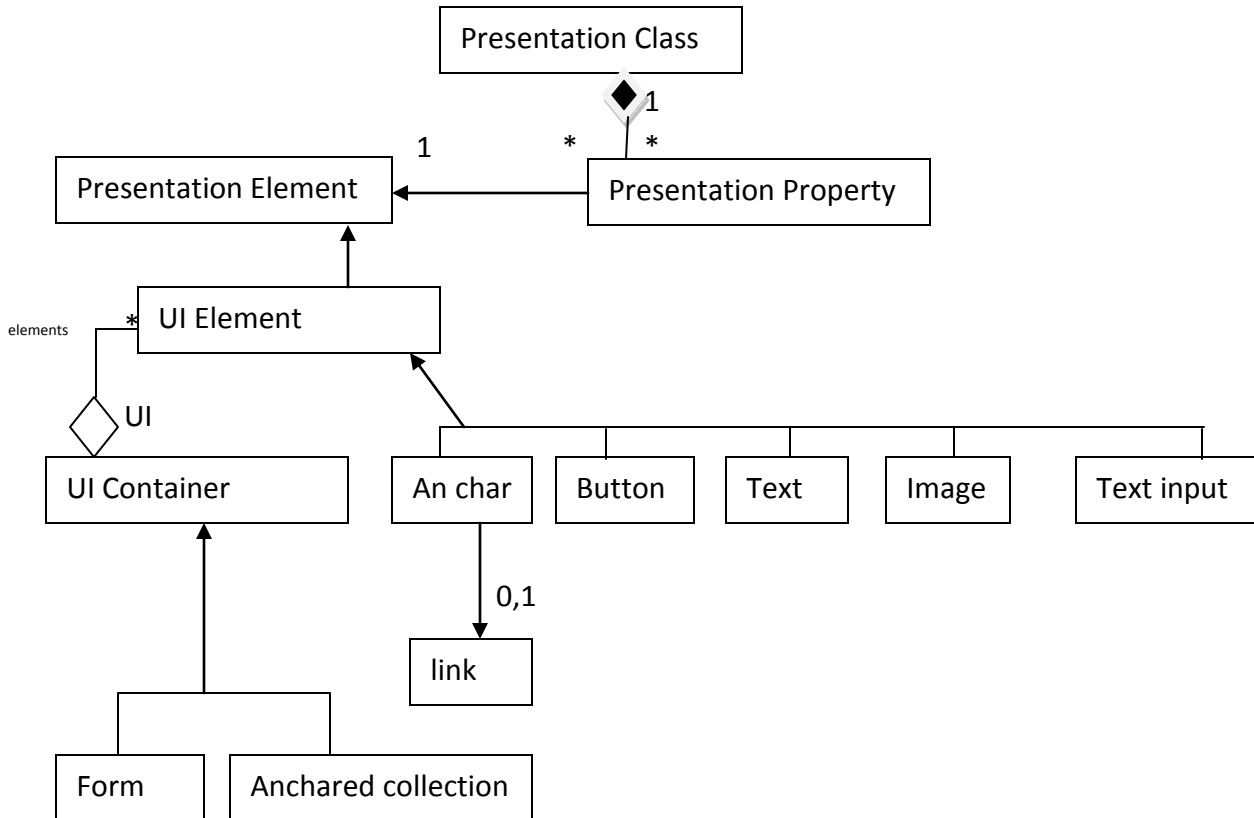


Explication:



Pour plus de détail sur les 4 couches, cherchez "MOF-UML".

Paquetage de présentations

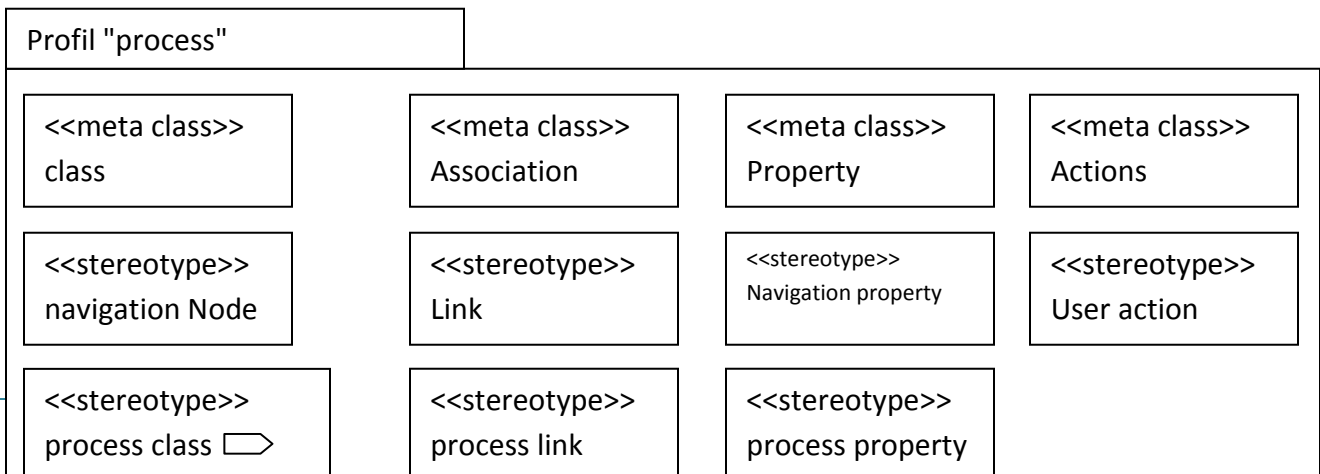


5) Le paquetage "Processus"

- Ce paquetage fournit les éléments de modèle pour l'intégration du processus métier
- Cela peut être séparé en trois tâches.

a) Intégration des processus

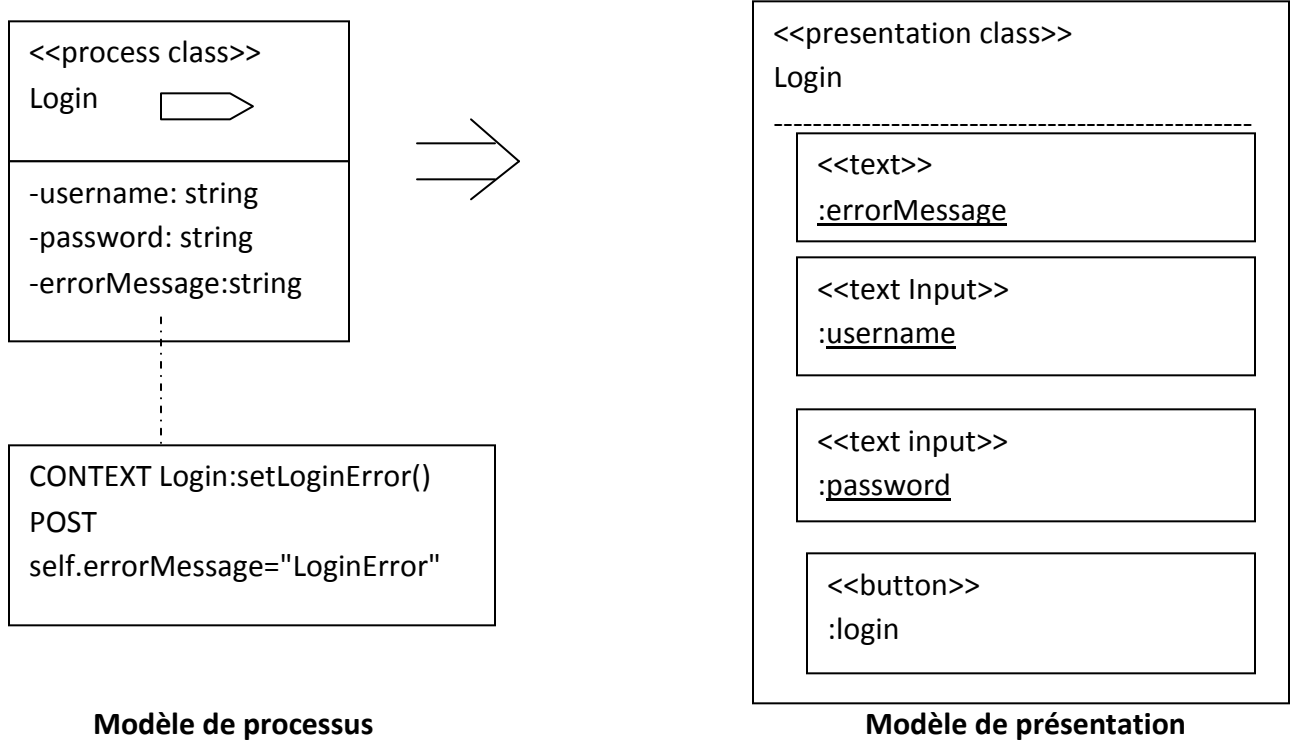
C'est l'intégration des processus métiers dans le modèle de navigation, cela est possible grâce aux méta classes processClass et Process montrant comment un processus peut être atteint à travers la navigation et comment la navigation continue.



b) Définitions de l'interface utilisateur

On définit l'IU pour prendre en compte les processus

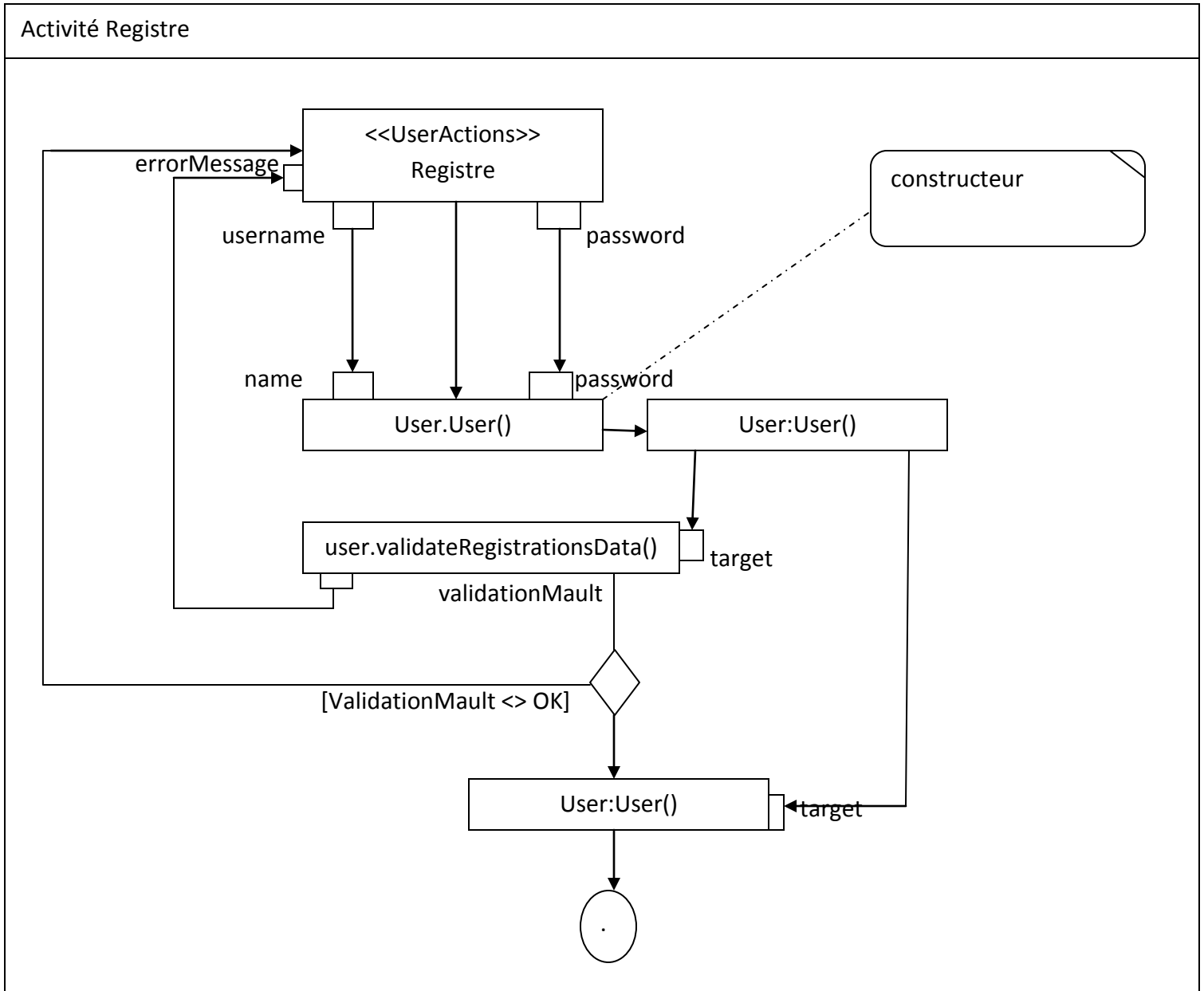
- Les processus requièrent une IU pour l'acquisition des données.
- Cette IU peut être définie à l'aide du modèle de présentation pour chaque classe de processus.
- Pour chaque classe des processus, une classe de présentation est créée définissant l'interface utilisateur.
- Les éléments de IU sont connectés aux propriétés du processus.

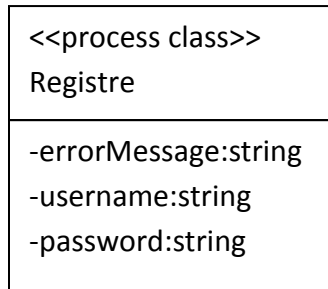


c) Définition du comportement

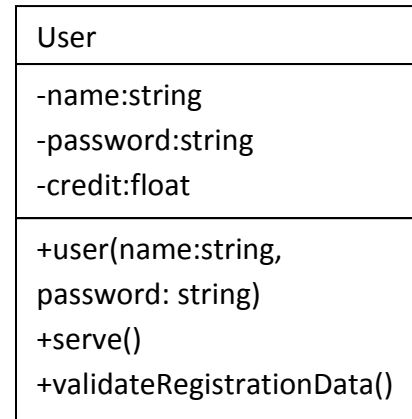
- le comportement est défini à l'aide du diagramme d'activité
- Les restrictions et sémantiques spécifiques sont appliquées
 - Un stéréotype spécial " UserActions " est utilisé pour marquer un point dans le flot de contrôle quand l'utilisateur doit entrer des données.
 - Le flot de contrôle contenu après que l'utilisateur a renseigné les données.
 - Le "user action" est associé à une classe de processus pour identifier quelle domaine est éditer et quelle classe de présentation est montrée.
 - Un "process property" de la "process class" fournit les données d'entrée depuis les éléments IU à travers les punaises de sortie du " UserActions " qui possède le même nom que "process property"

- De même les propriétés du processus peuvent être mise à jour avec des punaises d'entrées de "UserActions" correspondant.
- Les valeurs sont utilisés pour initialiser les éléments IU connectés.



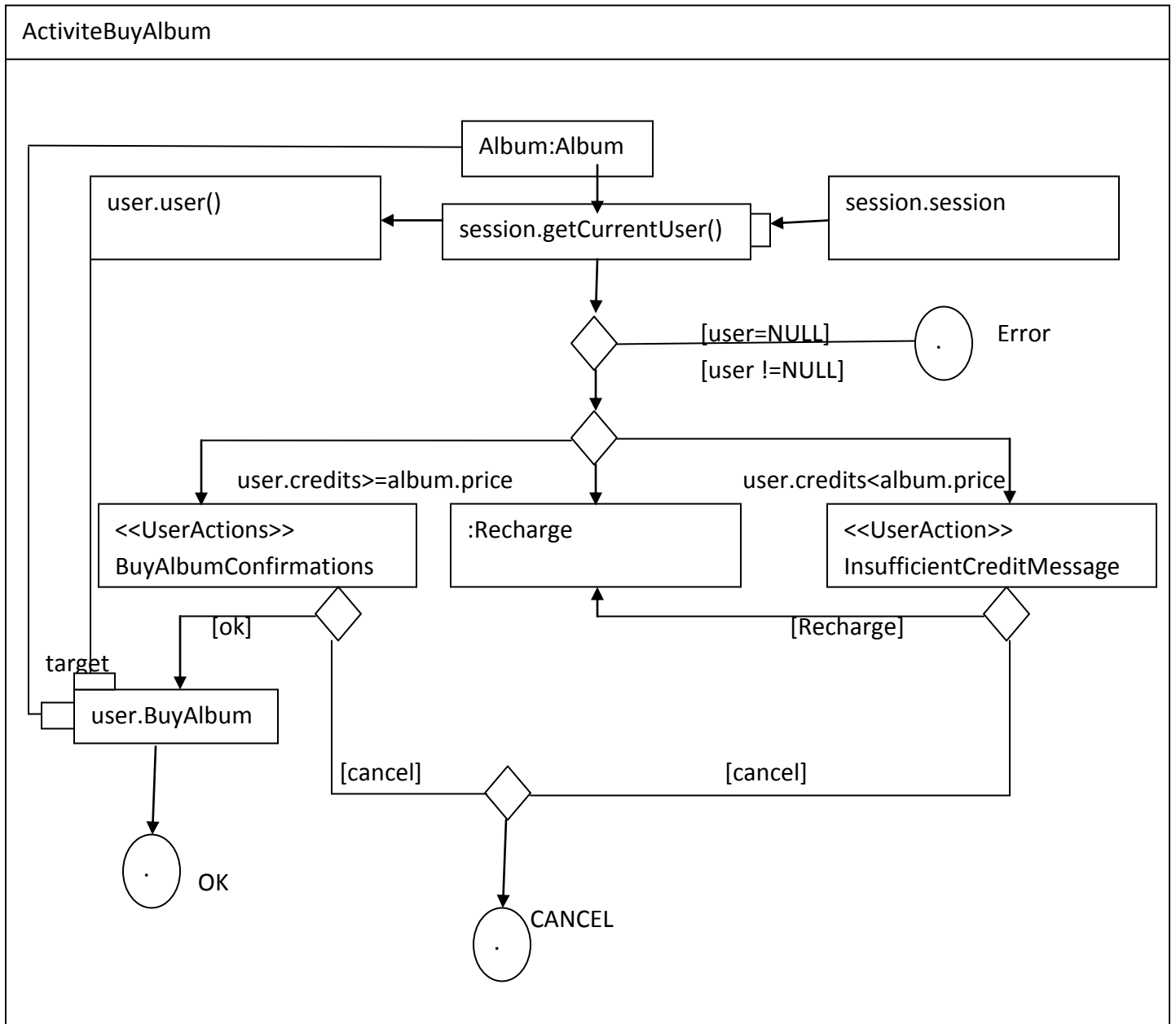


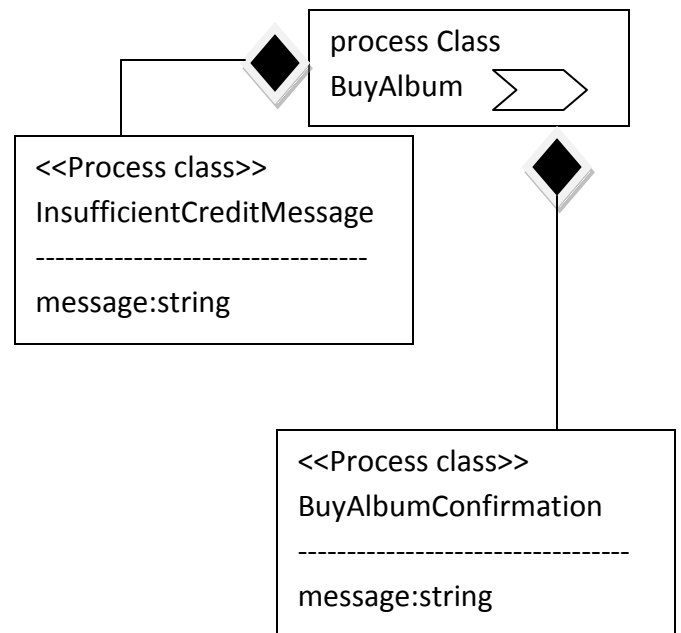
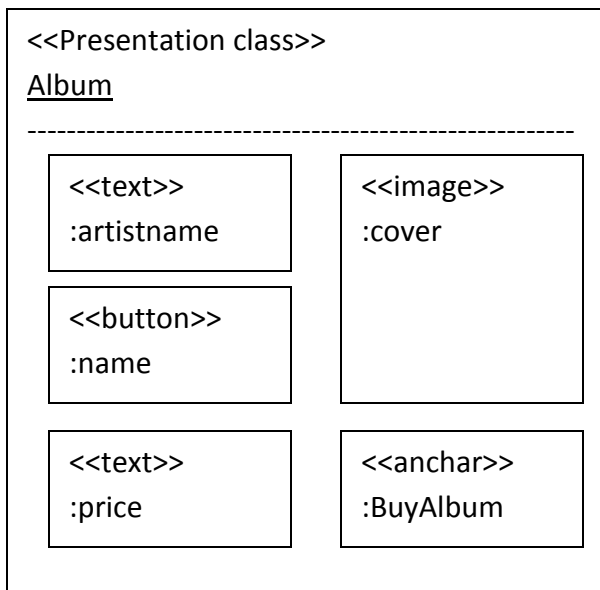
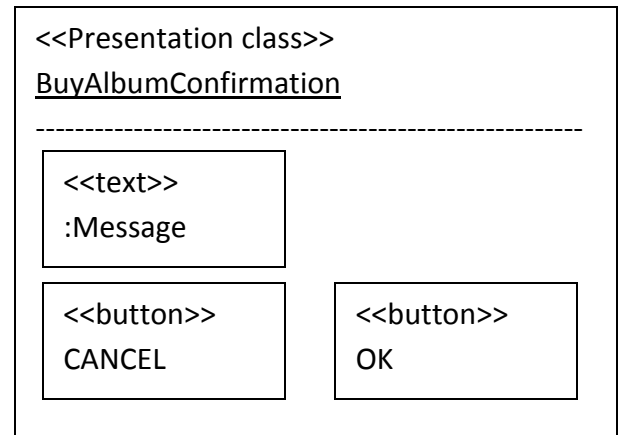
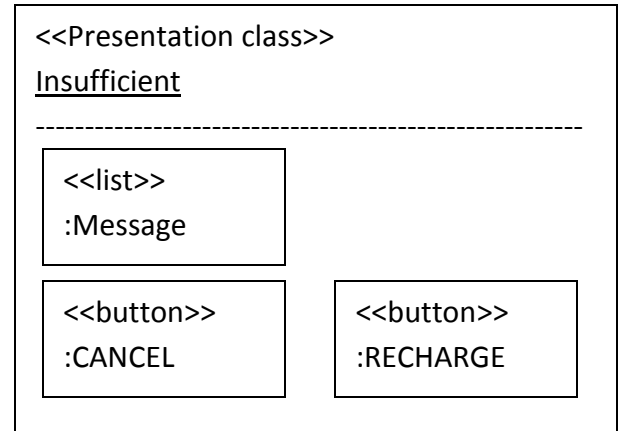
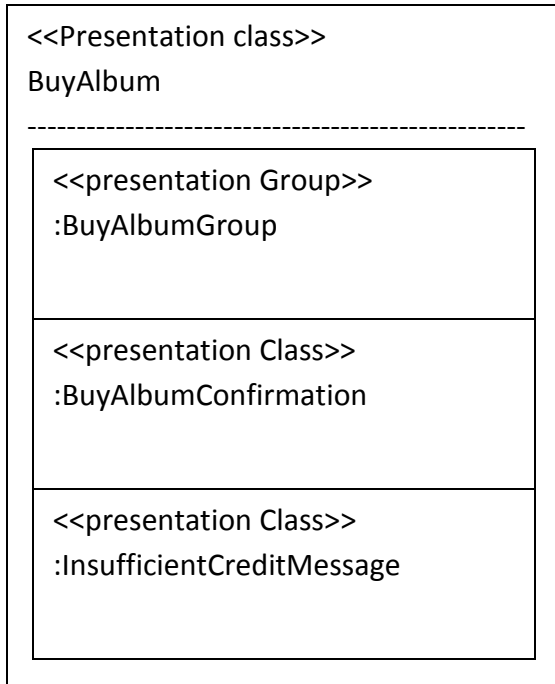
Modèle Processus



Modèle utilisation

- Dans beaucoup de cas, le processus a besoin de données d'entrées provenant du nœud précédant dans le graphe de navigation
 - Cette situation est modélisée par un nœud de paramètre d'activité utilisé à la place du nœud d'actions initiale
 - Le nœud de paramètre doit être de même type que la classe de contenu et que la classe de navigation.





Modèle de navigation

Modèle de processus

Chapitre 7 : La méthode par l'étude d'un cas

(Exemple sur le site web www.imdb.org : Base de données sur les films)

Introduction

Description

- Ces films sont caractérisés par:
 - Genres
 - castings
 - bandes annonces
- Les personnes liées aux films incluent
 - directeurs
 - producteurs
 - acteurs
- Les utilisateurs doivent être enregistrés (identifier log)
 - Commentaires
 - Evaluations
 - Des votes
- Le système personnalise
 - Application
 - recommandation
 - news

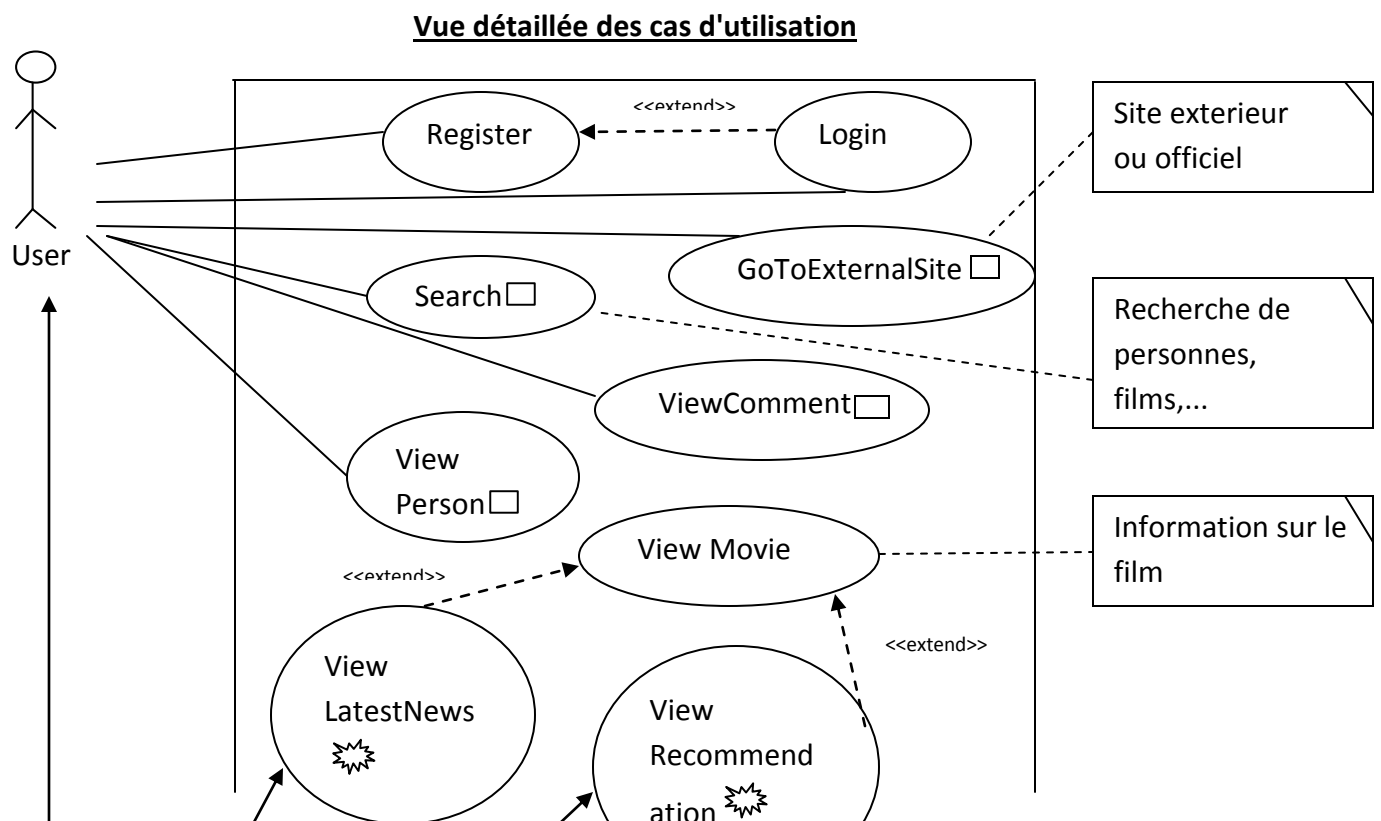
Objectif

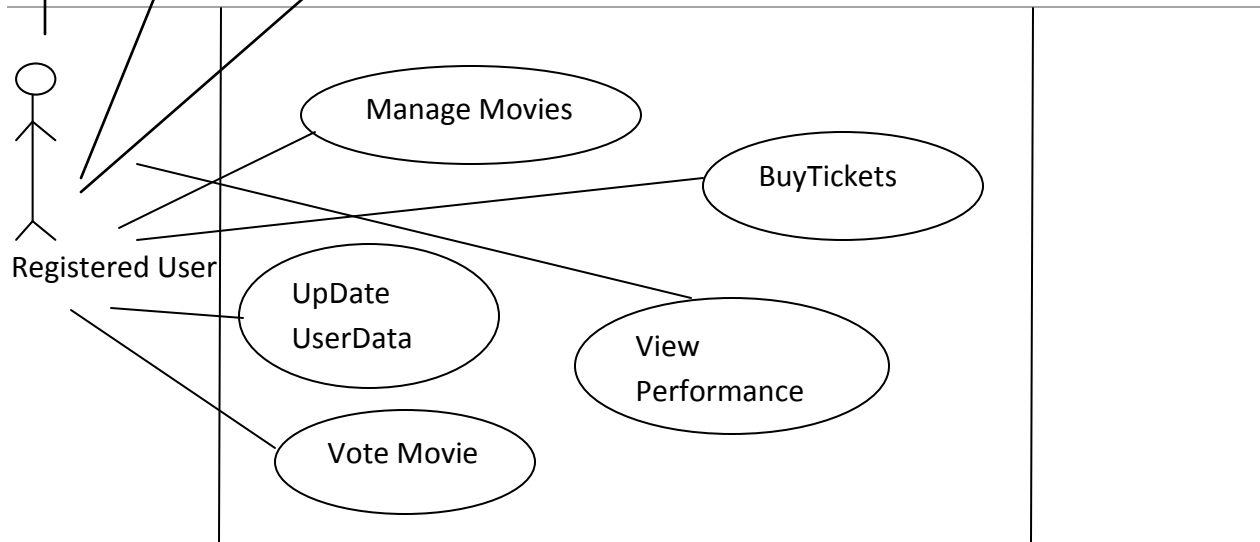
Construire des modèles des différents vues sur les phases d'analyse et conception.

1) Commencer avec la spécification des besoins

- A l'aide du modèle des besoins
 - Deux niveaux de détails
 - (1) Description grossière des fonctionnalités à l'aide d'un diagramme des cas d'utilisation
 - (2) Description plus détaillé à l'aide des diagrammes d'activités
- Diagramme de cas d'utilisations
 - Les acteurs
 - Utilisateur anonyme (USER)
 - Utilisateur enregistré (Registered User)
 - Administrateur (Admin)

- UWE fait la distinction entre 3 types de cas d'utilisations
 - A. Navigation - <<navigation>> □
 - B. Processus - default
 - C. Personnalisation - <<personalized>> ✨
- A. Navigation : Modélisé la navigation web
 - Exemple:
 - (1) ViewMore
 - (2) Search
 - (3) GoToExternalSite
- B. Processus: Modélise les tâches métiers que l'utilisateur veut réaliser
 - Exemple:
 - (1) Register
 - (2) CommentMovie
 - (3) BuyTicket
- Personnalisation: Guidée par le comportement de l'utilisateur
 - Exemple:
 - (1) ViewRecommendation
 - (2) ViewLatestNews

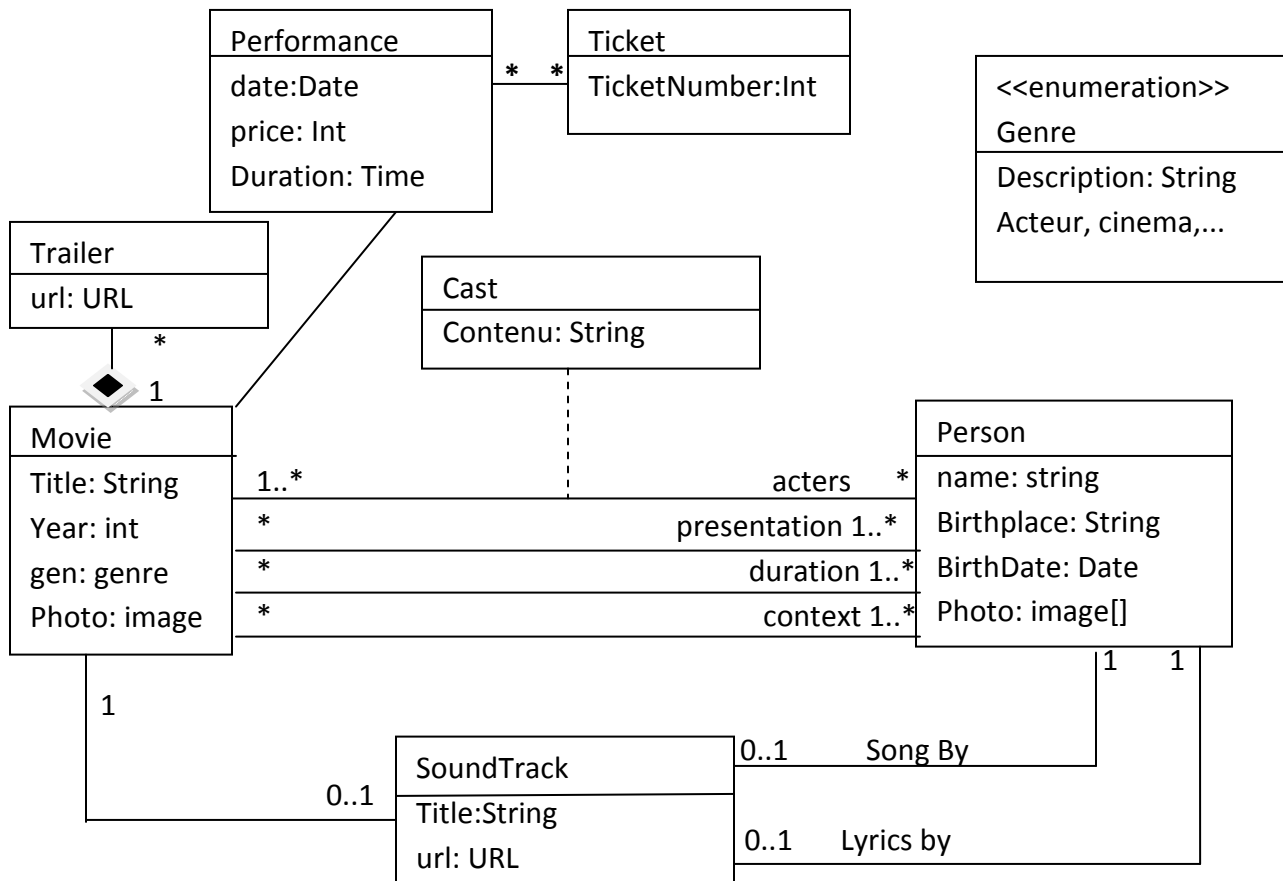




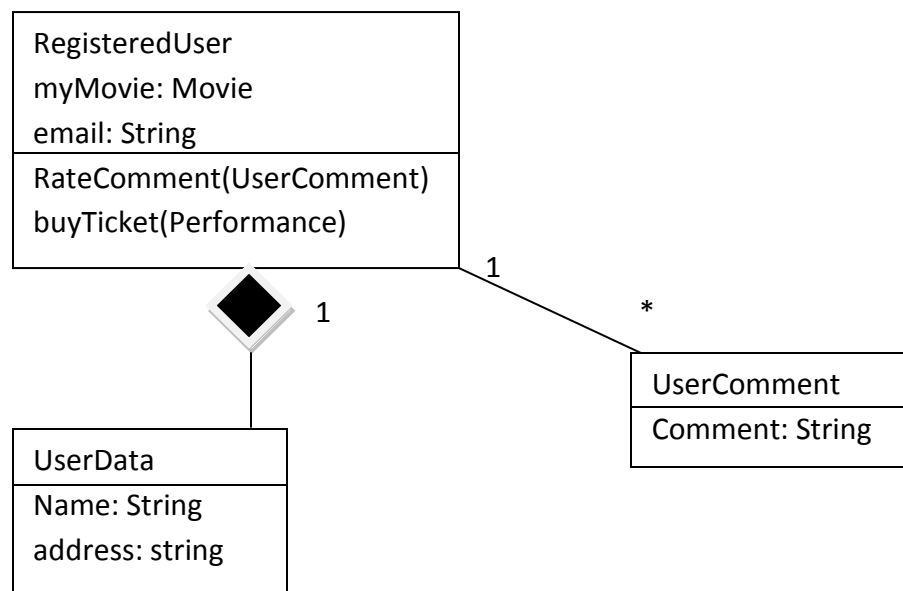
2) Définir le contenu

- Les modèle de contenu définie les informations du domaine
 - Contient le modèle utilisateur au contexte






Modèle de contenu



Modèle Utilisateur



3) La structure de navigation

- Elle se base sur l'analyse et la modélisation du contenu
 - Classe navigation <<navigationClass (□)>> (nœud navigable de la structure hypertexte)
 - Liens de navigations montrent les liens directs entre classes de navigation <<navigationLink>>
 - NavigationInteractive <<Menu>>()
 - Accès aux instances multiples <<index>> () <<guidedtour>> ()
 - La recherche utilise la classe <<query>> 
- Les processus métiers sont intégrés à la structure de navigation
 - <<processClass>> ()
 - <<permLink>>

a) Esquissé le modèle de navigation

Règle 1: Les classes du modèle de contenu sont pertinente pour la navigation.

Règle 2: Les liens entre les classes du modèle de contenu peuvent servir aussi bien de classe de navigation que de lien de navigation. (Si on a 3 classes C1, C2 et C3 alors C2 vas servir à relier C1 à C2).

Règle 3: Un lien de navigation entre deux classes de navigation dans les deux sens doivent être séparés.

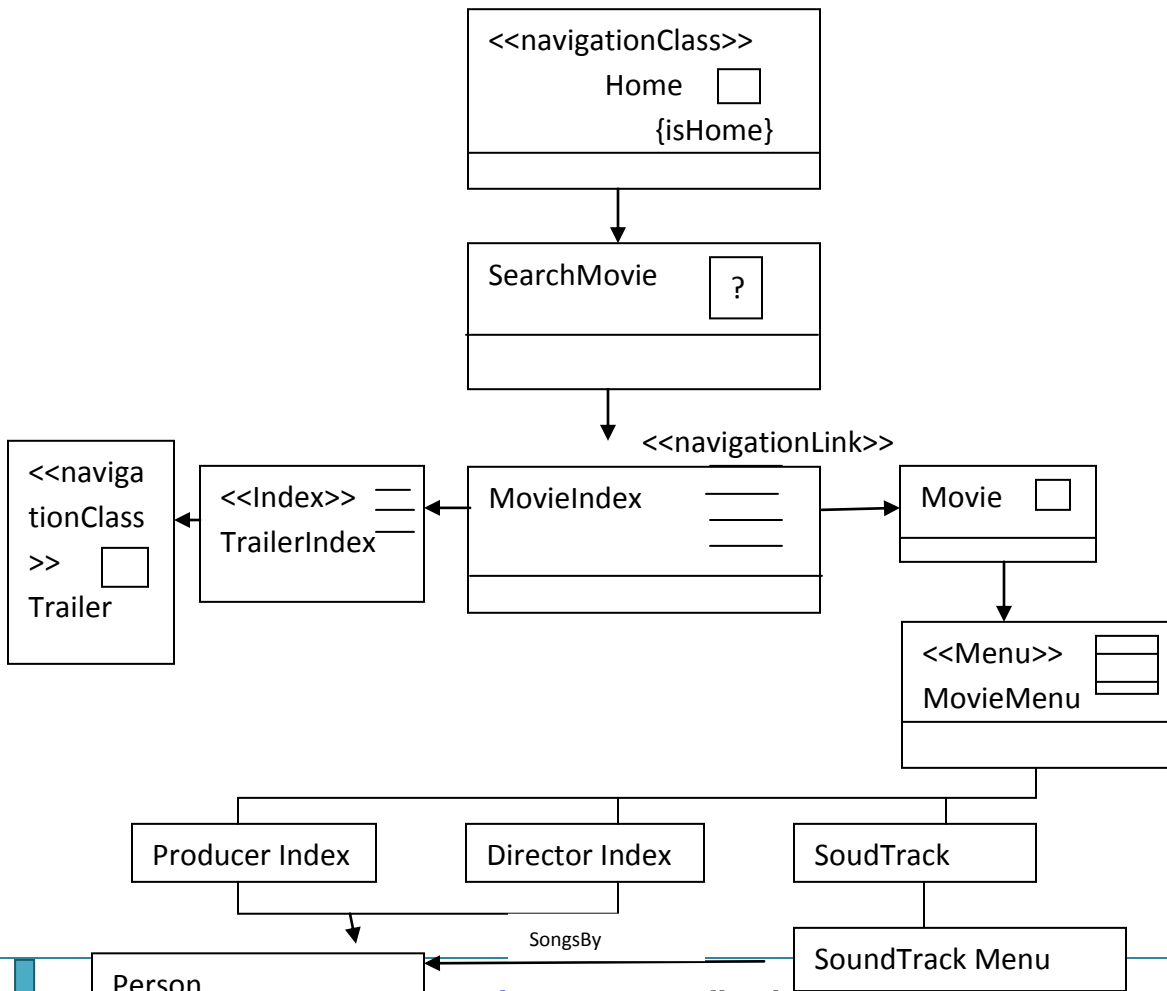
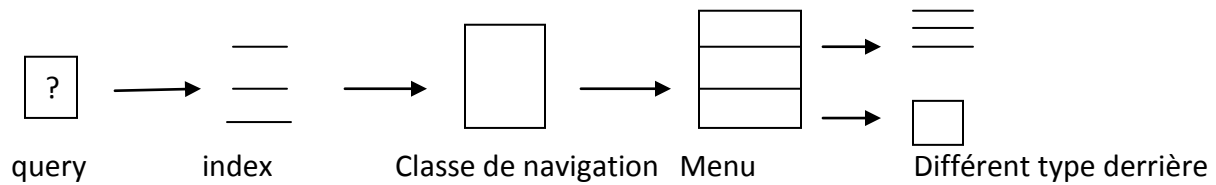
Règle 4: Les menus sont ajoutés pour chaque classe de navigation qui possède plusieurs types d'association de sortie. (Exemple: L'utilisateur choisit une option du menu dans page principale ce qui va déclencher une requête (query) avec un index, pour finalement avoir une nouvelle page avec un nouveau menu donc nouvelle classe de navigation)

Règle 5: Un index, un tour guidée ou une recherche doit être placé entre deux classes de navigation quand la cardinalité > 1.

b) Modèle de navigation

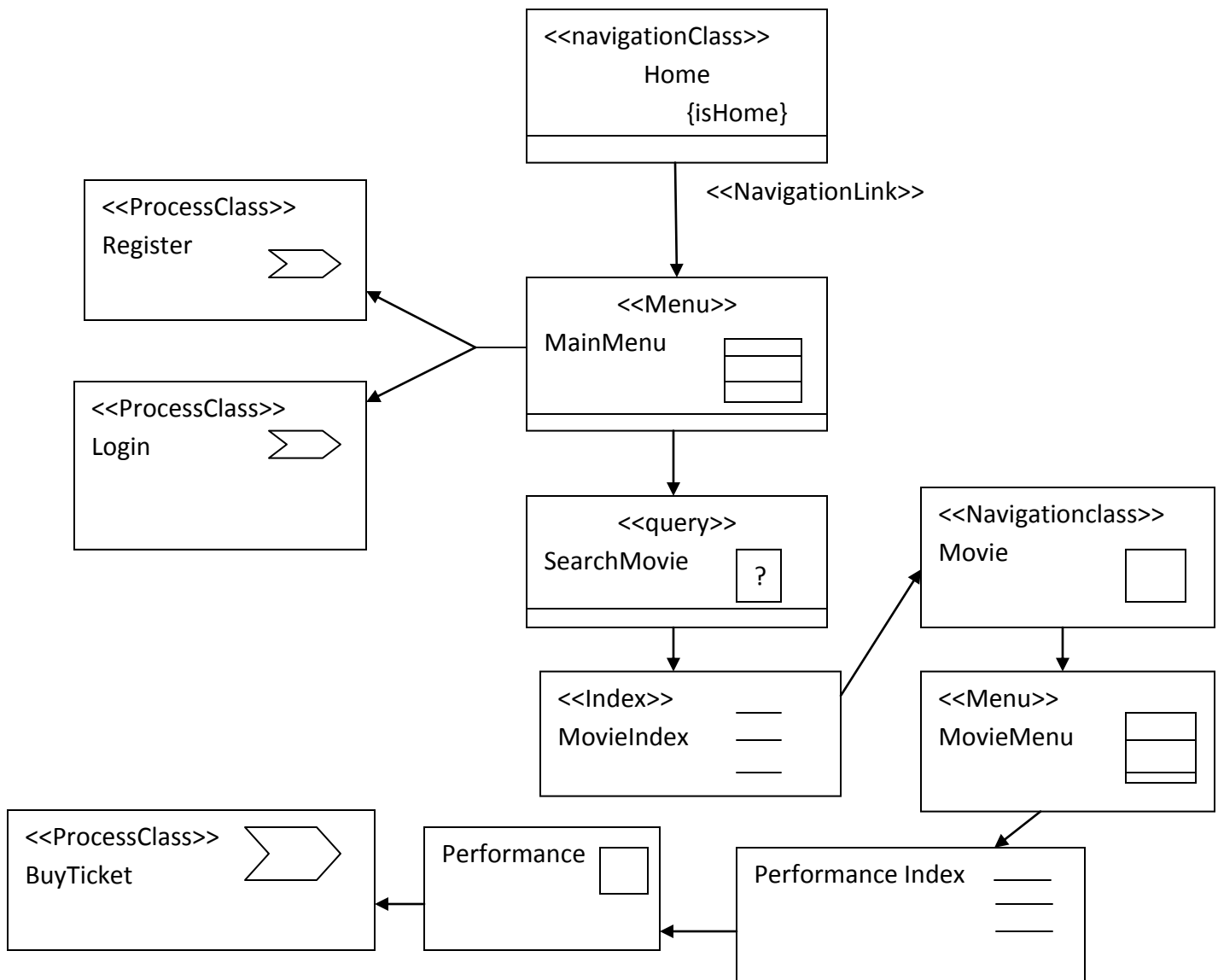
! Si on a une query alors on a forcément un index avec.

! Derrière d'un index, on aura une classe de navigation.



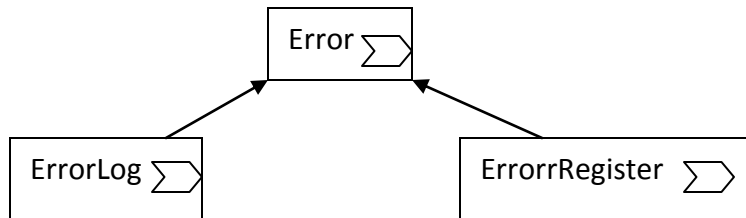
c) Ajoute le processus métiers

- L'étape suivante nécessite à étendre la structure de navigation à l'aide de la classe de processus représentant les entrées/sorties des processus métiers.
- Pour cela, il faut utiliser les cas d'utilisations navigationnels
Exemple: Register , Login



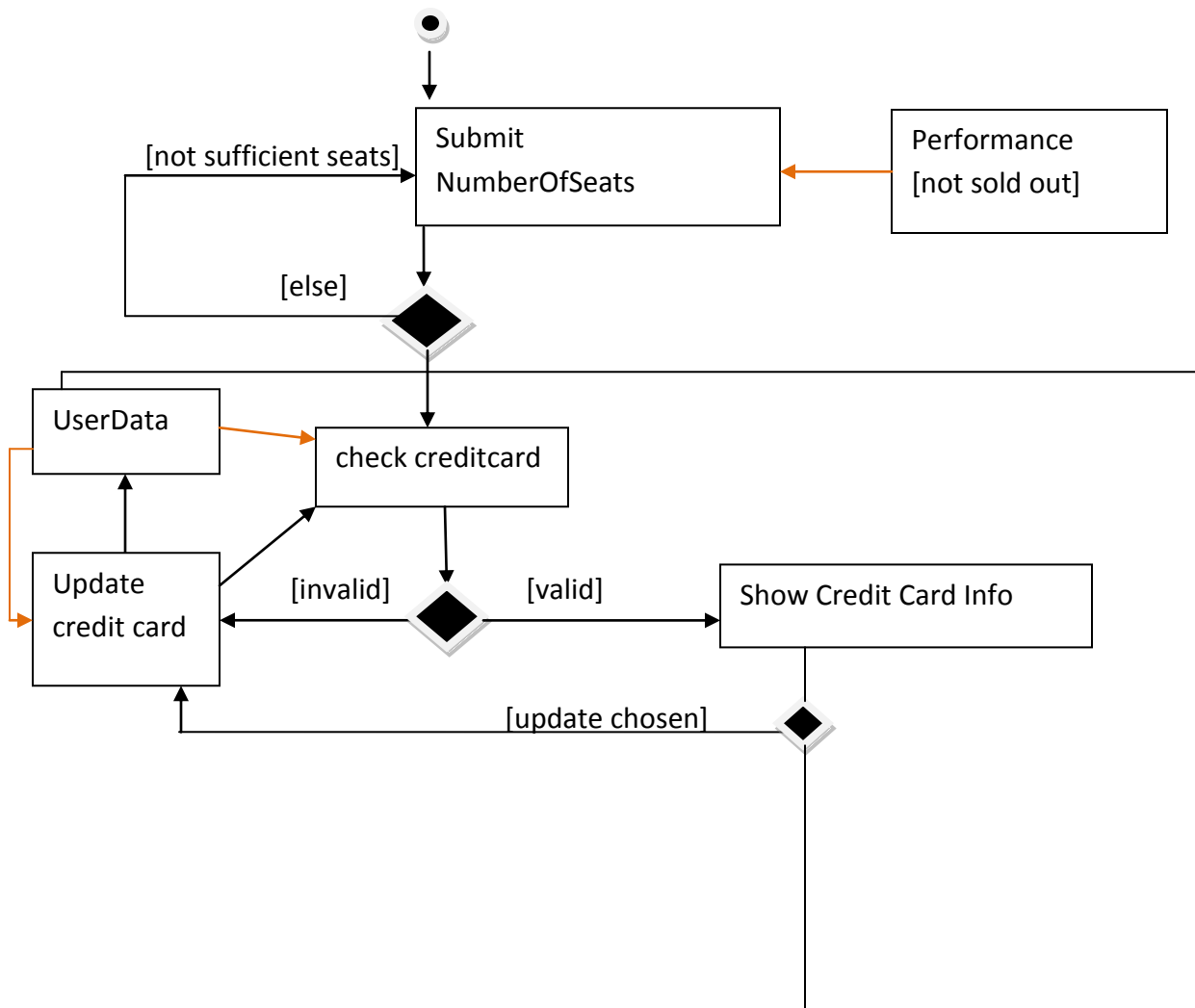
4) Raffinement du modèle de processus

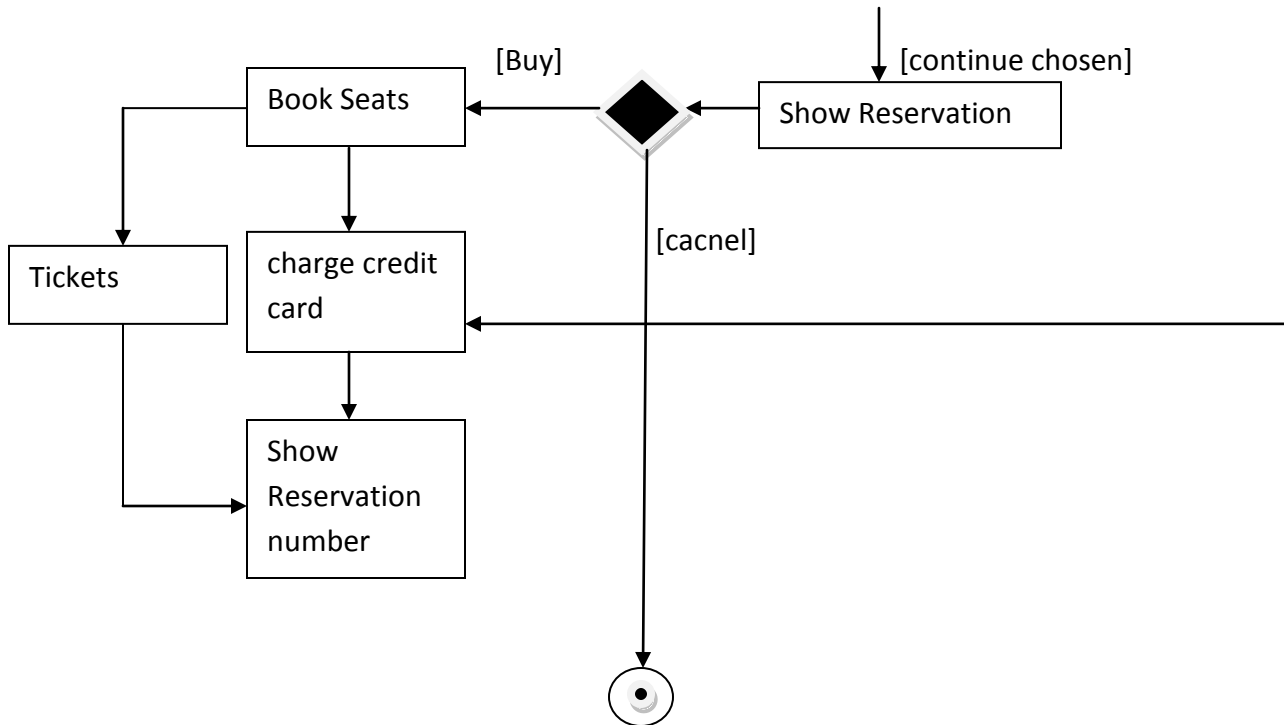
- Chaque classe de processus est ajoutée dans le modèle de navigation.
- Le modèle de processus est constitué de:
 - Modèle de structure de processus (Drag class)



- Modèle de flot de processus
 - Contrôle } Modèle à l'aide de diagramme d'activité
 - Flot de données }

Diagramme d'activité BuyTicket











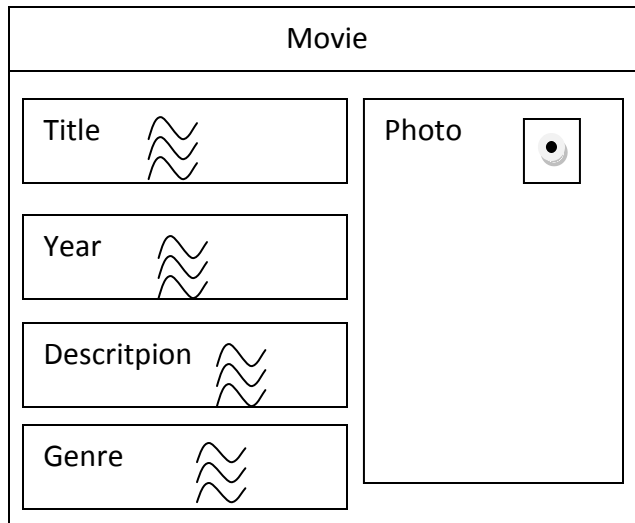
Les flèches colorées en **orange** indiquent les objets qui sont consommés par les tâches. Les autres flèches montre le flot des données.


5) Esquisse la présentation

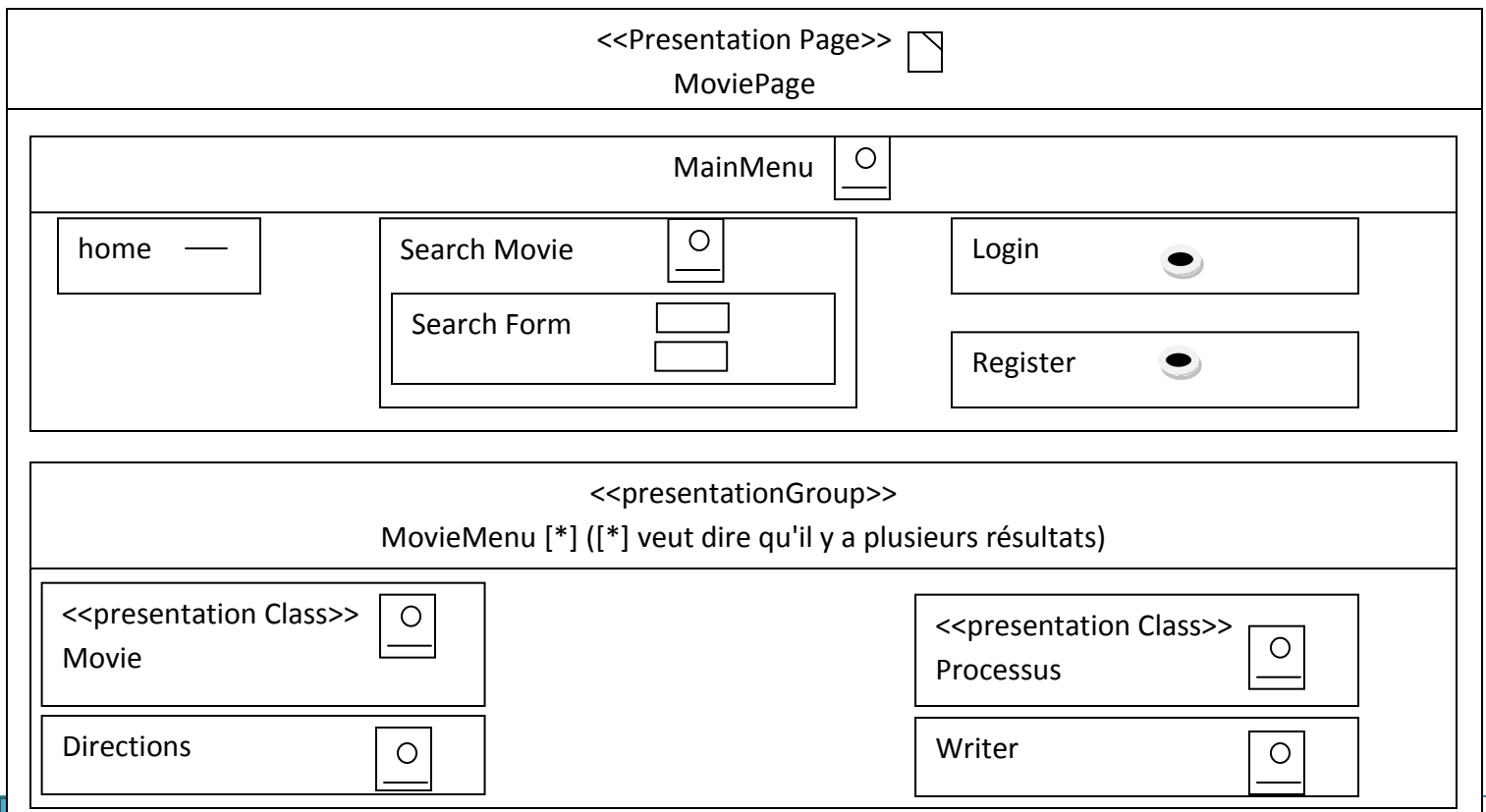
- Le modèle de présentation fournit une vue abstraite de l'interface utilisateur de l'application web
- Il est basé sur le modèle de navigation
- Il fait abstraction des aspects concrets de l'interface utilisateur comme la couleur, layout, etc...
- Le modèle de présentation dessert les éléments de base (texte, images, formulaire, etc.)
a fin de représenter les nœuds de navigation

- Les éléments de base du modèle
 - <<présentation>> ()
 - <<text>> ()
 - <<anchor>> (—)
 - <<button>> ()
 - <<image>> ()
 - <<formulaire>> ()

- o <<anchoredCollection>> ()



- Plusieurs nœuds de navigation sont représentés sur une seule page modélisé par <<presentation Page>> ()
- Une page peut contenir des classes de présentation et des groupes de présentation.
- Les groupes de présentation peuvent contenir des groupes de présentation et des classes de présentation.



Exemple 1:

Site web statique (statique c'est à dire il n'y a pas de base de données)

Objectif: Expression des besoins de l'application web avec navigation.

Description: Homepage composé de l'introduction (texte), index articles (nom, description) et lien vers remerciements (texte).

Chaque article est composé de texte d'introduction et liste de sections (considération et besoin, solution, discussions et comparaison, contribution, biblio), chaque section est composé de titre, texte, éventuellement page externe.

Navigation: Page accueil (autre page), page index sections (section)

(Explication:

On commence par:

a) *Besoin*: il n'ya pas de besoin car il n'y a pas de requête sur le système, pas d'interrogation à la base de données.

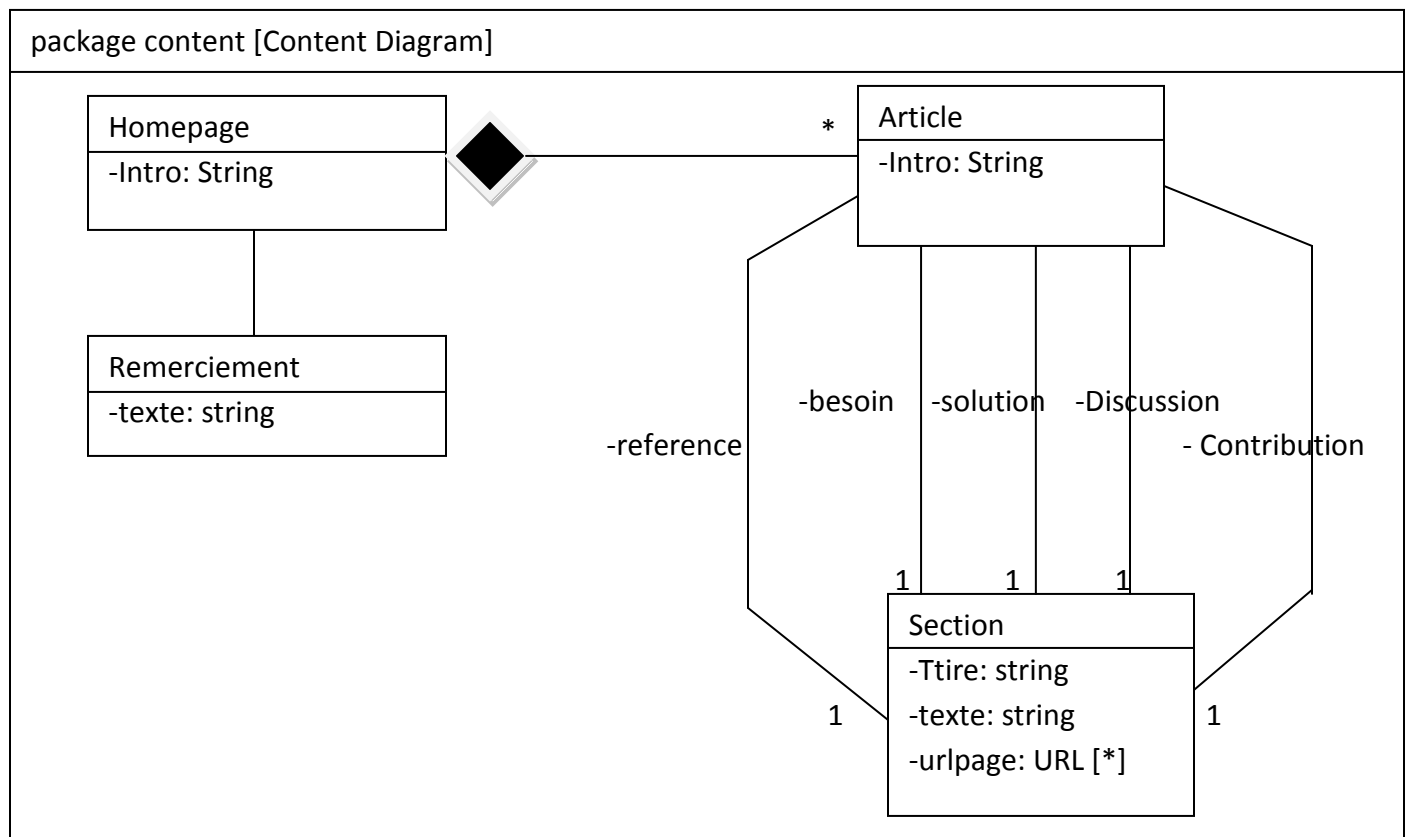
b) *Contenu*: on fait inclure les infos à modélisé.

c) *Navigation*: on inclus les règles, modèle,...)

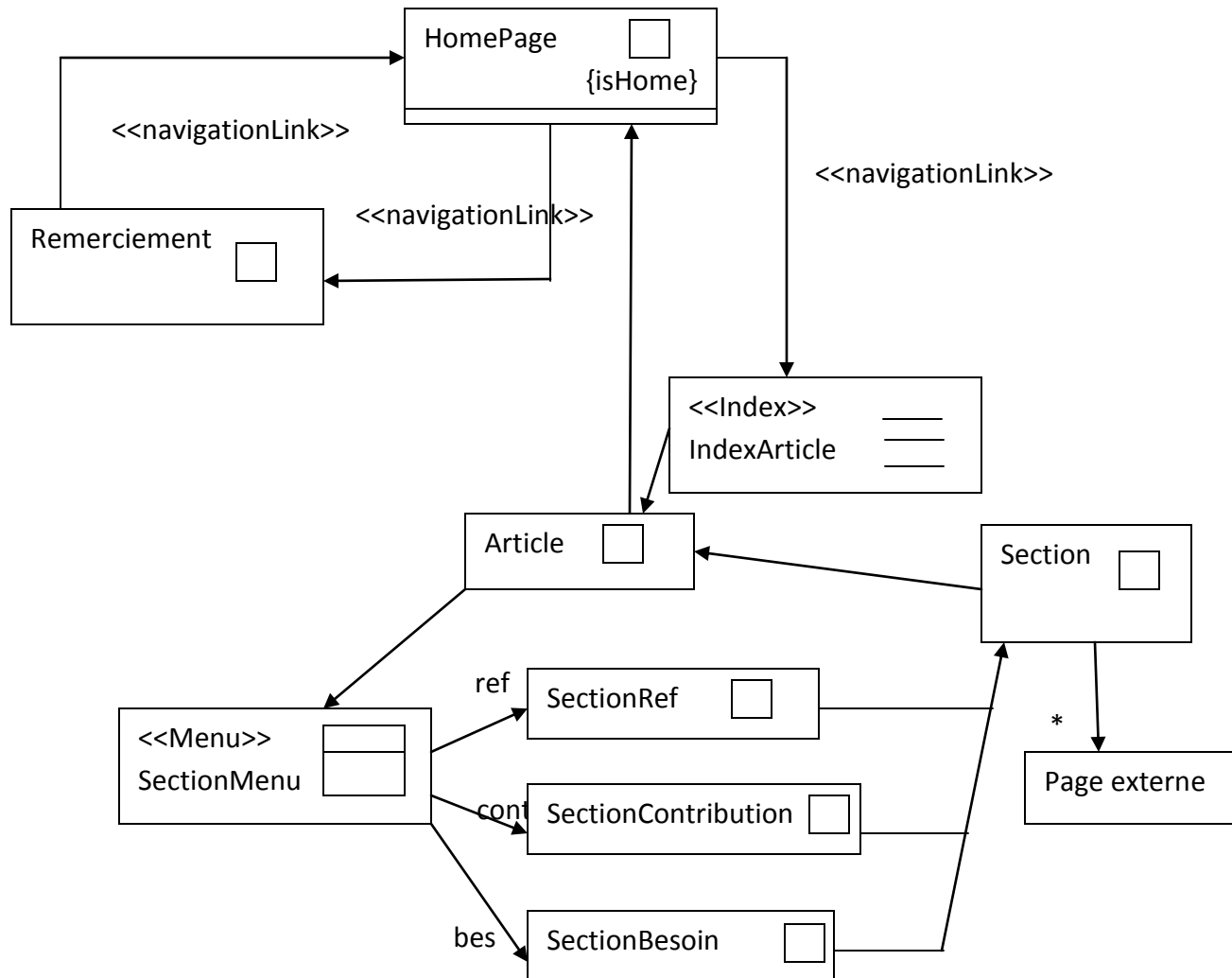
Solution

a) *Besoin*: Il n'y a pas de besoin

b) *Contenu* - Content diagram

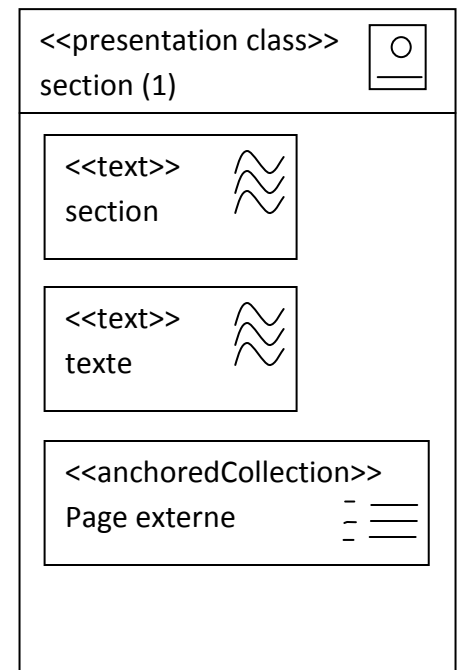
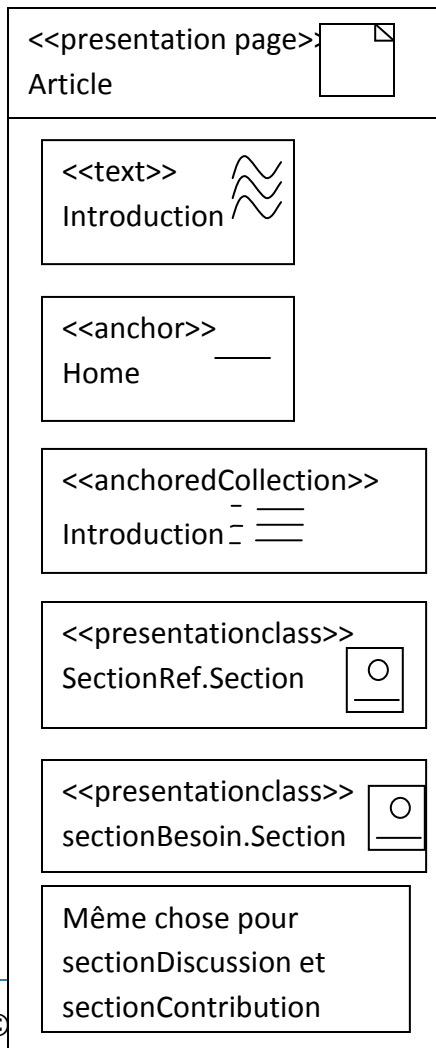
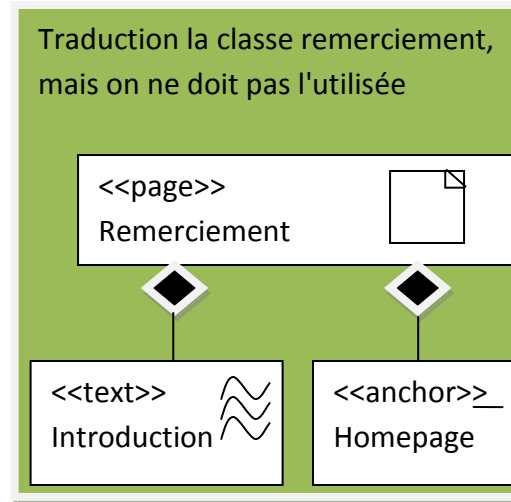
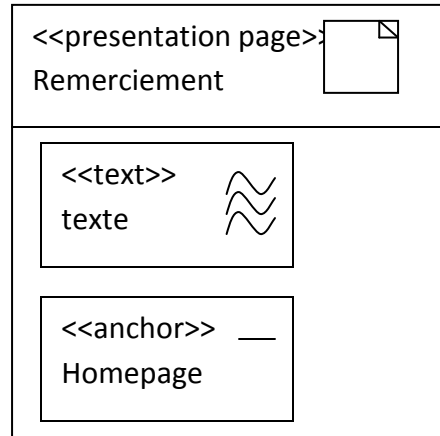
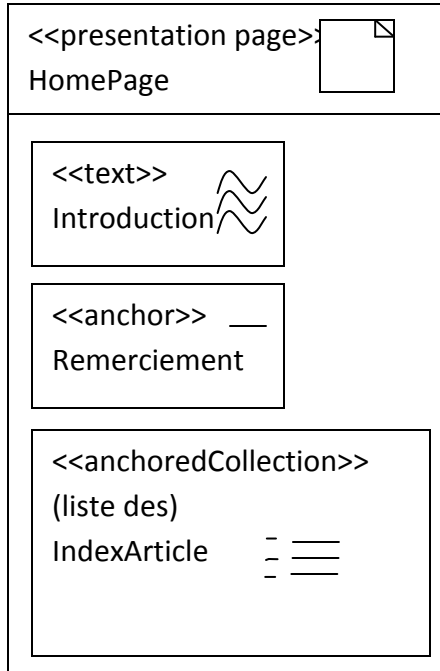


c) Navigation



Quand on a une navigation en deux sens, on met deux flèches (comme pour le remerciement)

d) Présentation



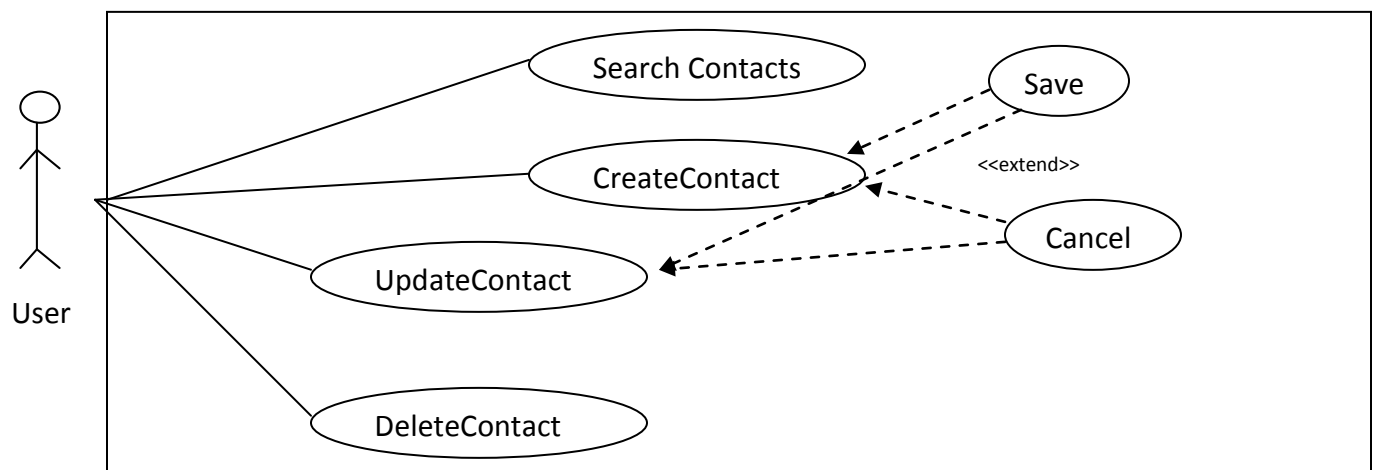
(1) (c'est une présentation class car ca va être utilisé par l'article, dans cet exemple on a plusieurs sections dans l'article, (comme par exemple: ref, besoin, discussion, ..., chacune d'eux est une section)

e) Processus, il n'y a pas car il n'y a pas de besoin.

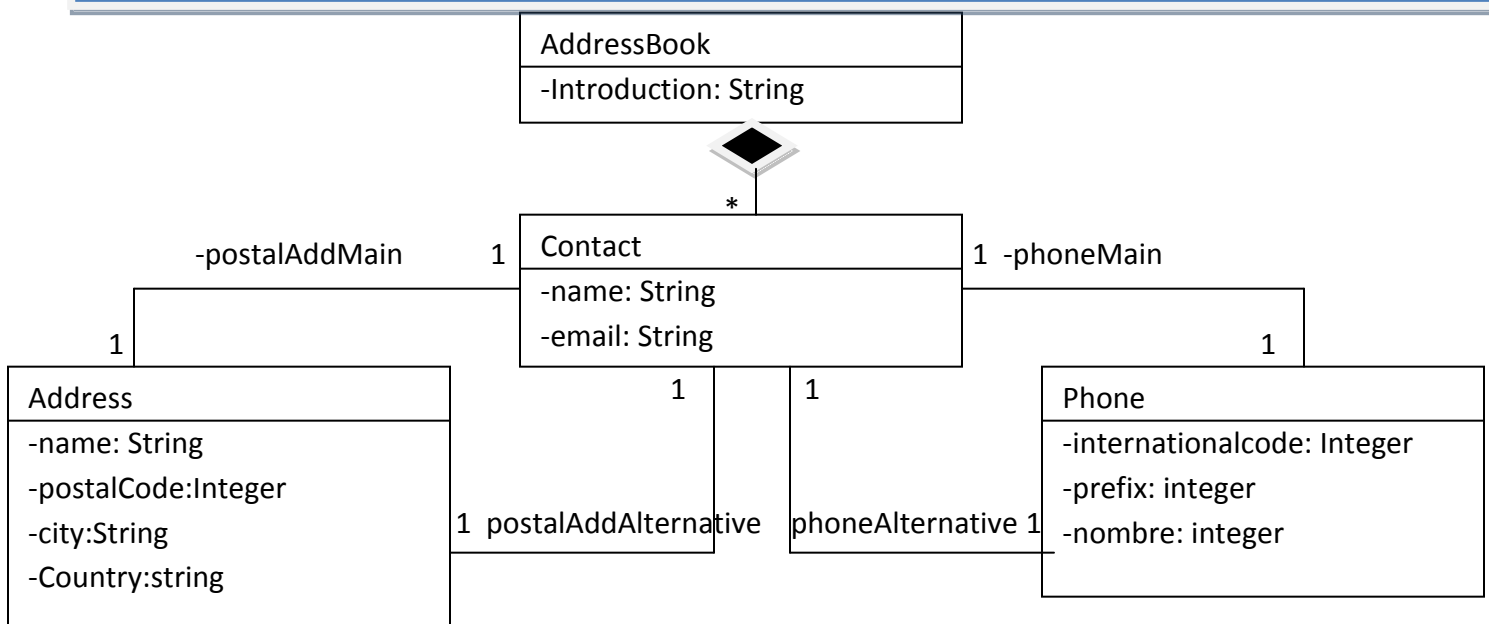
Exemple 2 - TP1

Address Book

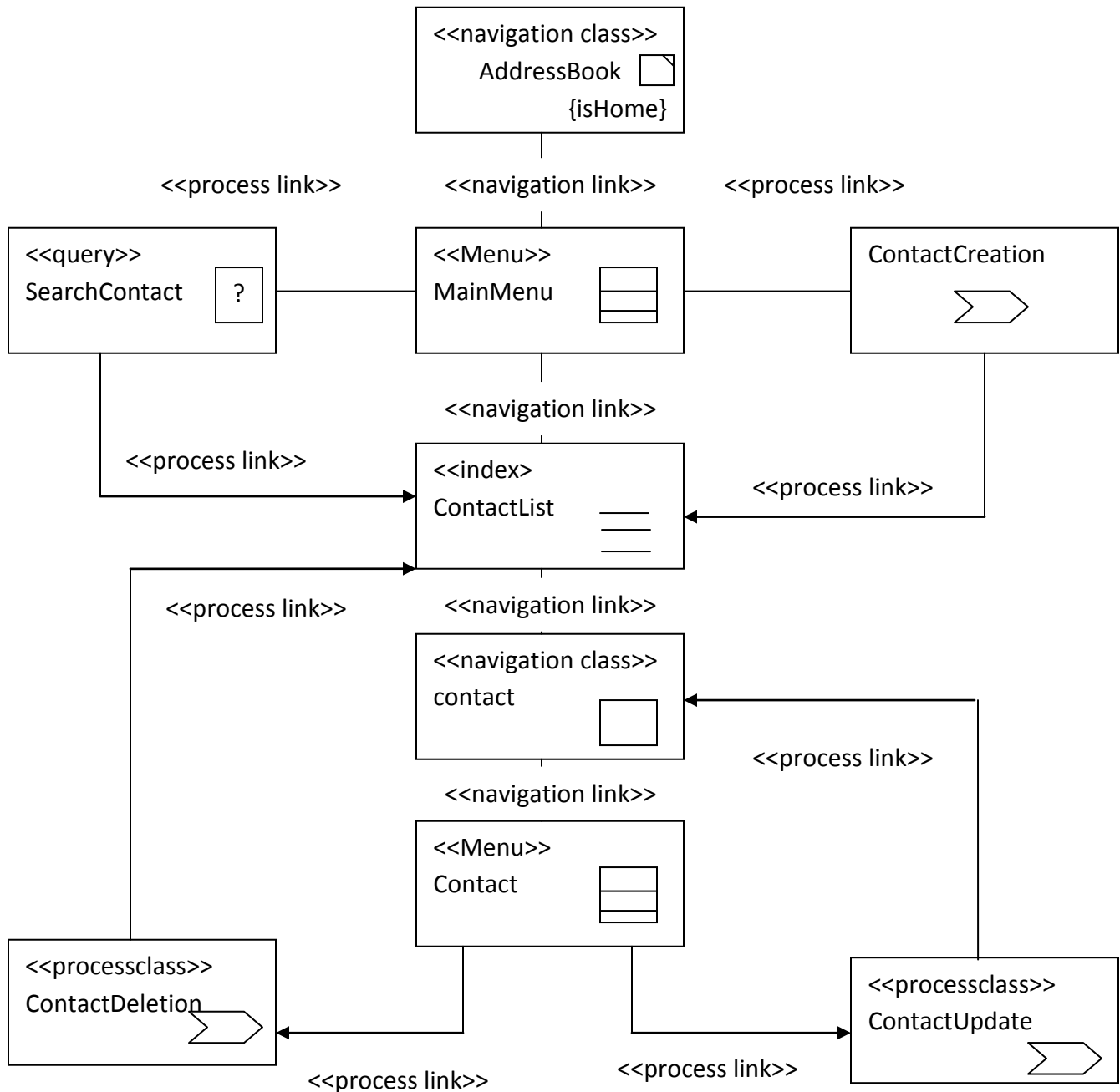
1) Besoin



2) Contenu



3) Navigation



NB: On commence par une page, puis toujours on passe par le menu, puis on a différent type de page.

Quand il y a plusieurs processus métiers, il faut y avoir un menu avant.

Le plus important c'est de faire le lien entre les modèles.

On doit normalement garder les même nom (même avec les boutons) durant la transition d'un modèle à un autre.

Le search est un cas spécial de processus métier c'est pourquoi on ne le présente pas dans le diagramme d'activité.

Système d'Information Orientés Objets

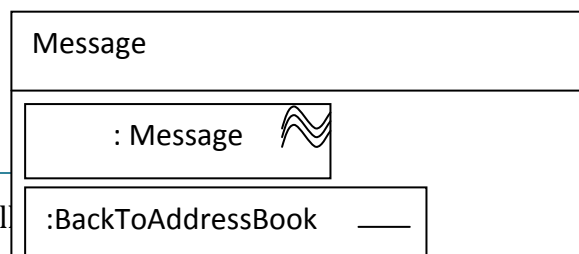
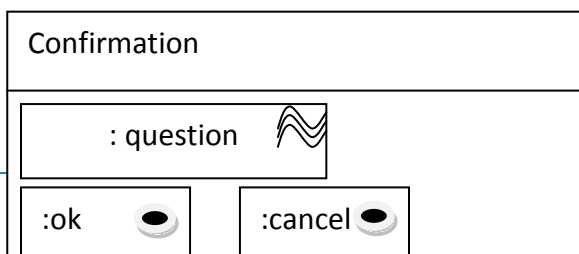
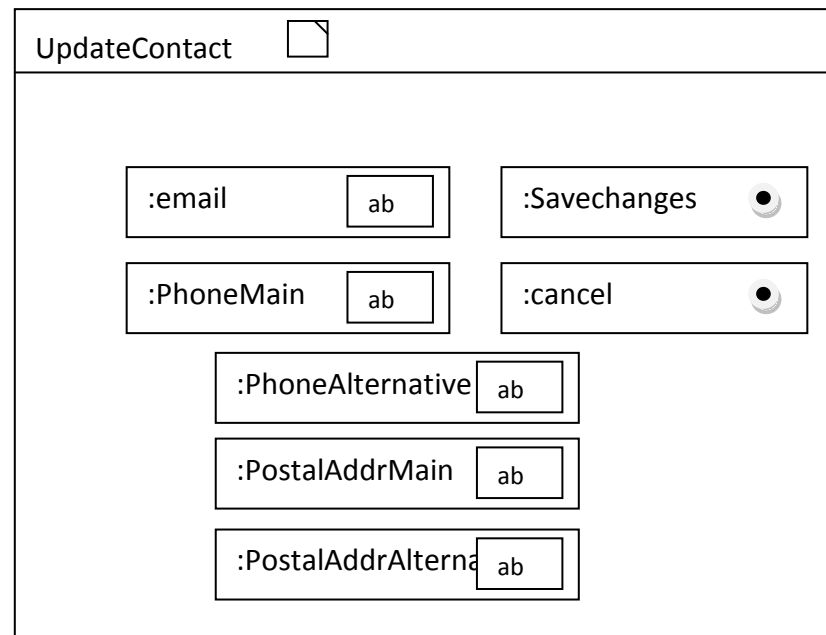
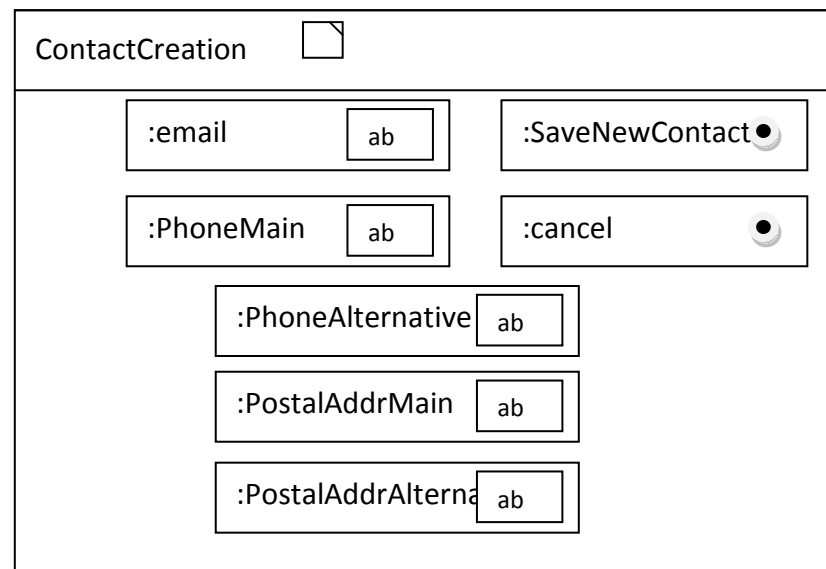
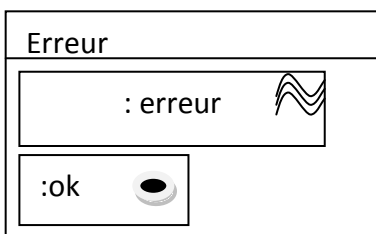
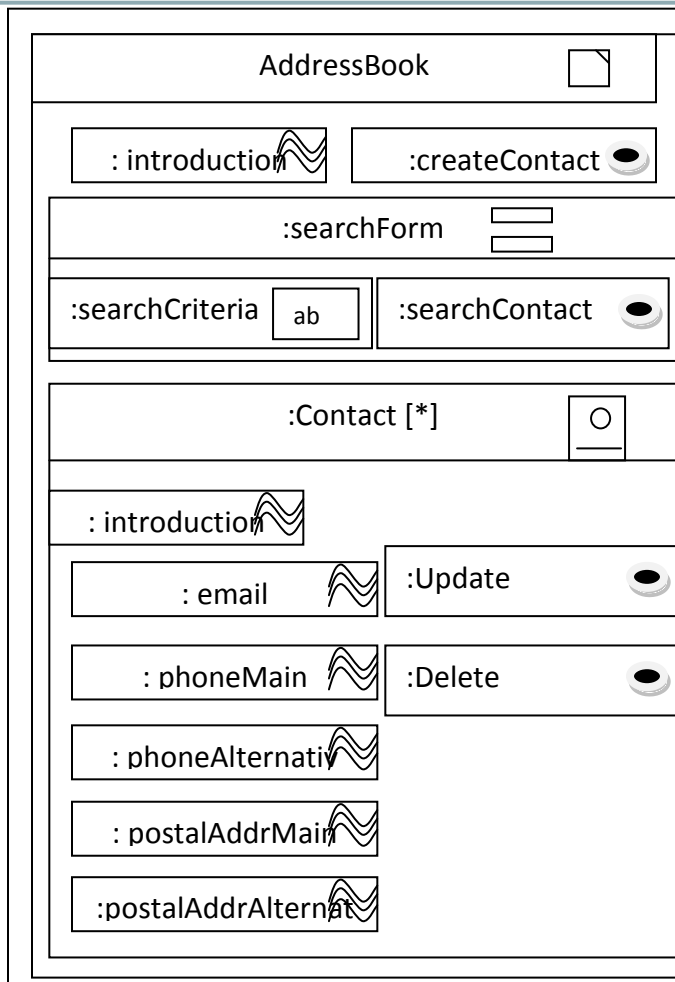
Préparé par Elie MATTA

[1] besoin (DCU), 2) Contenu (Diagramme de Classe (DC)), 3) Navigation (DC), 4) Présentation (DC), 5) Processus (DC+Diagramme d'Activité)]

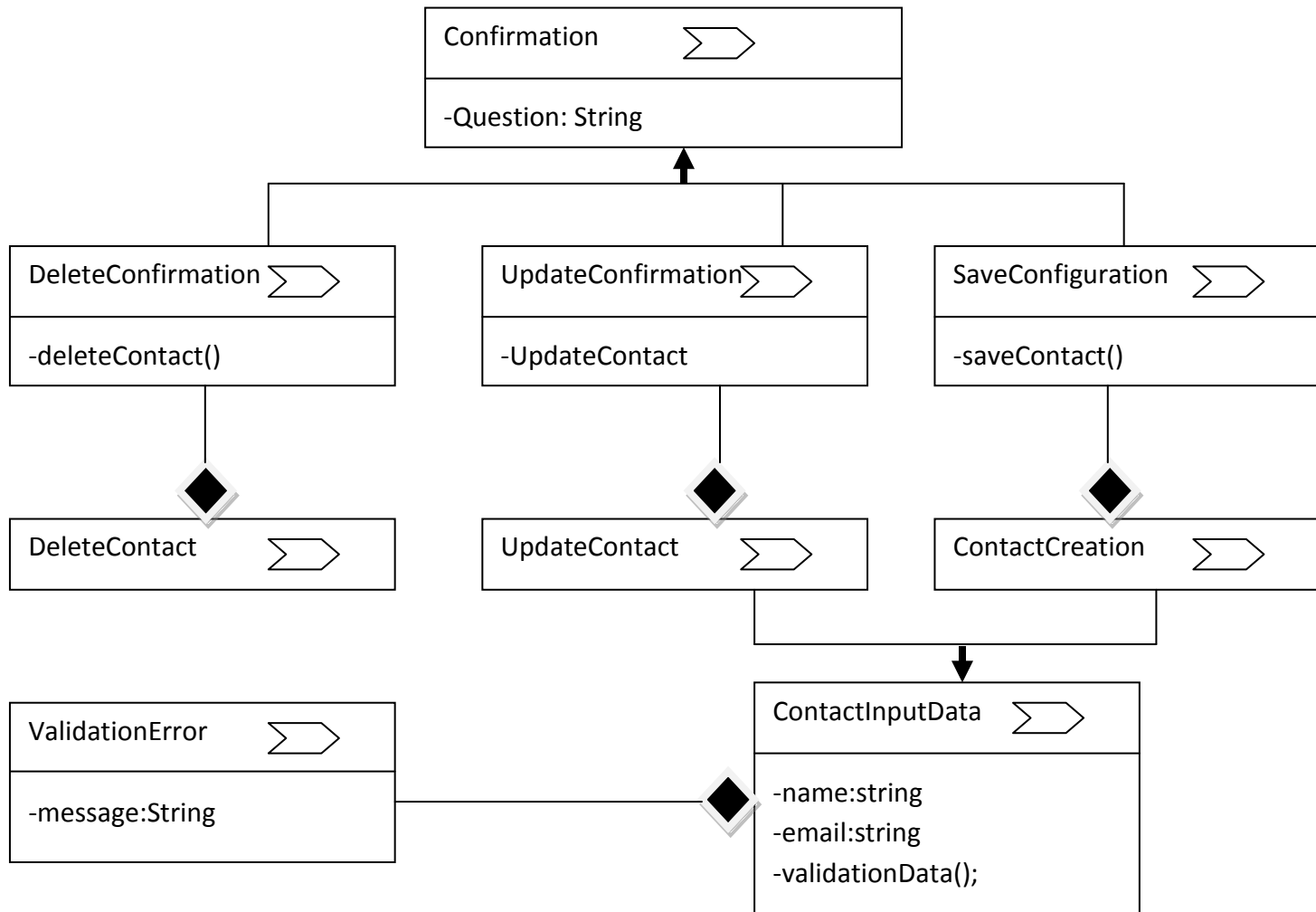
Dans l'examen:

- Il nous donne les 2 premiers diagramme de besoin et du contenu
- Il n'y aura pas de problème compliqué comme cet exemple.
- On doit expliquer la façon qu'on a conçu notre diagramme.

4) Package presentation [Presentation Diagram 1]



5) Package processus [Process Diagram]



5) Package processus [activityContactCreation (Contact Creation Workflow - Digram d'activité)]

