

Quelle démarche pour passer des besoins au code ?

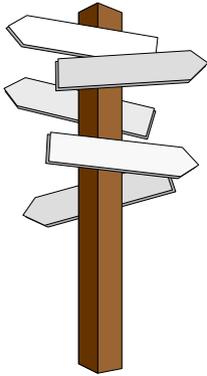
Chantal Reynaud

Université Paris X – Nanterre - UFR SEGMI

Maîtrise MIAGE



Code



Plan

- I. Processus de développement d'une application et UML
- II. Un processus simplifié de développement d'une application
- III. Autres processus de développement

Partie I. Processus de développement d'une application et UML

I. Notion de processus

Un ensemble d'étapes, partiellement ordonnées, qui concourent à l'obtention d'un système logiciel ou à l'évolution d'un système existant.

Son objectif est de produire des logiciels de qualité qui répondent aux besoins de leurs utilisateurs dans des temps et des coûts prévisibles.

II. Démarches de modélisation et UML

UML est largement indépendant des processus.

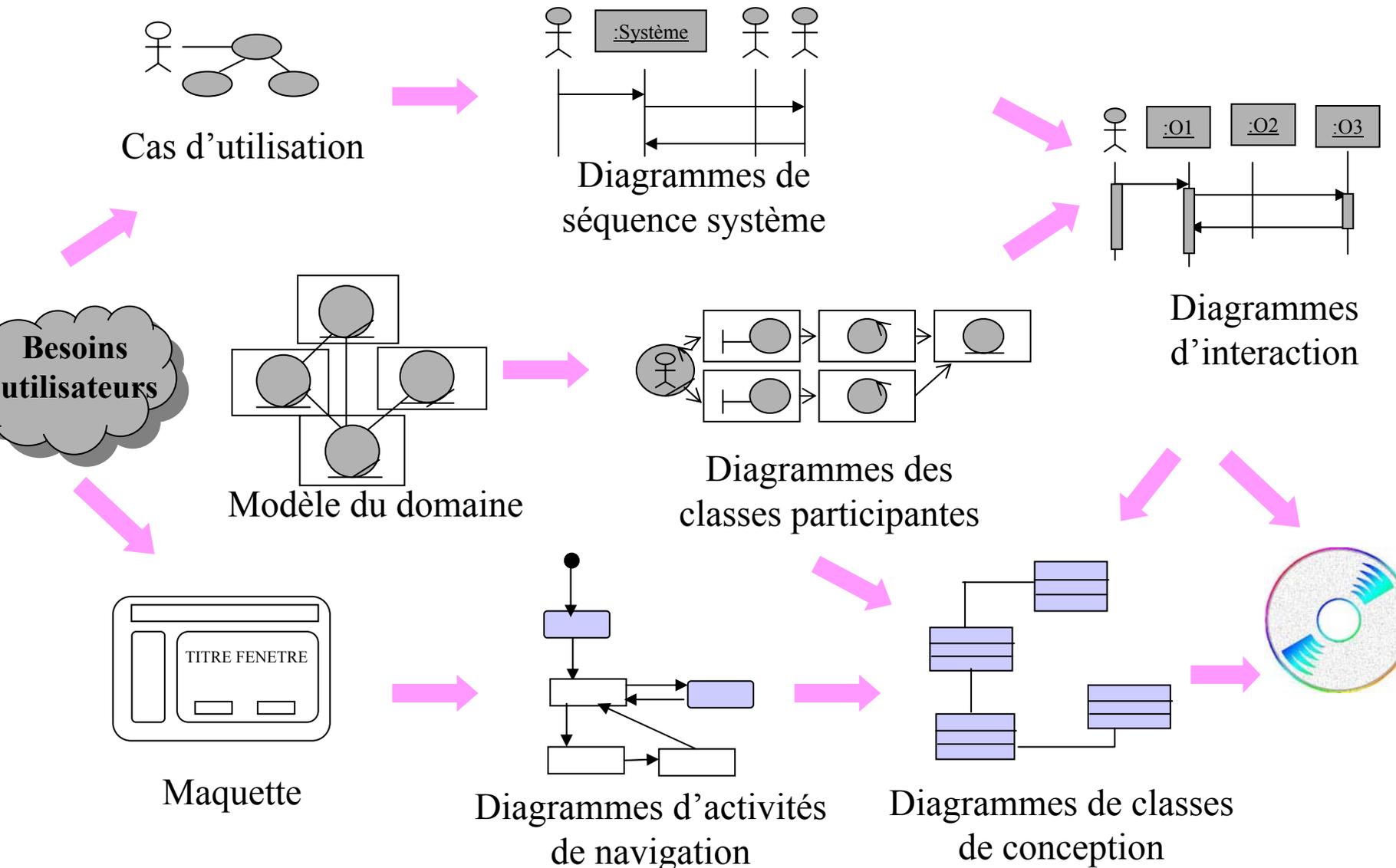
Propositions :

- UP : Rational Unified Process
- XP : eXtreme Programming
- Proposition d'une démarche à mi-chemin émanant de divers auteurs dont P. Roques fondée sur l'idée qu'il est possible de modéliser 80 % des problèmes avec 20 % d'UML.

Partie II. Processus simplifié de développement d'une application

- I. Aperçu de la démarche
- II. Description détaillée

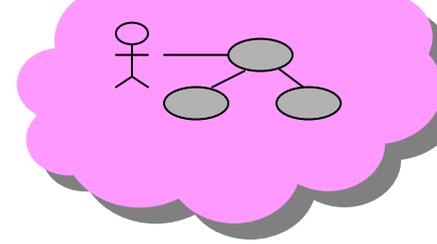
I. Aperçu de la démarche



II. Description détaillée

- I. Spécification des besoins d'après les cas d'utilisation
- II. L'analyse du domaine – les objets métiers
- III. Spécification détaillée des besoins
- IV. La réalisation des cas d'utilisation – Classes d'analyse
- V. La modélisation de la navigation
- VI. La conception objet préliminaire
- VII. La conception objet détaillée

II.1 La spécification des besoins d'après les cas d'utilisation



Cas d'utilisation

Démarche pour aboutir au diagramme de cas d'utilisation

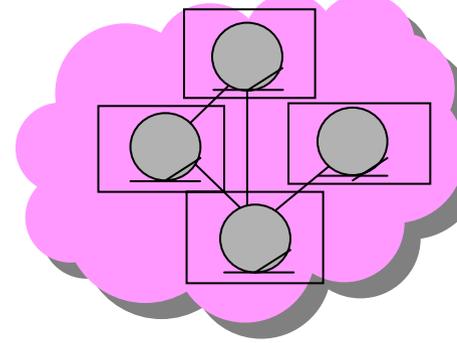
- Identifier les acteurs
- Identifier les cas d'utilisation
- Structurer les cas d'utilisation en packages
- Ajouter les relations entre cas d'utilisation
- Classer les cas d'utilisation

Un exemple de classement de cas d'utilisation et de planification d'un projet en itérations

Cas d'utilisation	Priorité	Risque	Itération
Rechercher des ouvrages	Haute	Moyenne	2
Gérer son panier	Haute	Bas	3
Effectuer une commande	Moyenne	Haut	4
Consulter ses commandes en cours	Basse	Moyen	6
Consulter l'aide en ligne	Basse	Bas	7
Maintenir le catalogue	Haute	Haut	1
Maintenir les informations éditoriales	Moyenne	Bas	5
Maintenir le site	Moyenne	Bas	5

II.2 L'analyse du domaine

Les objets métier

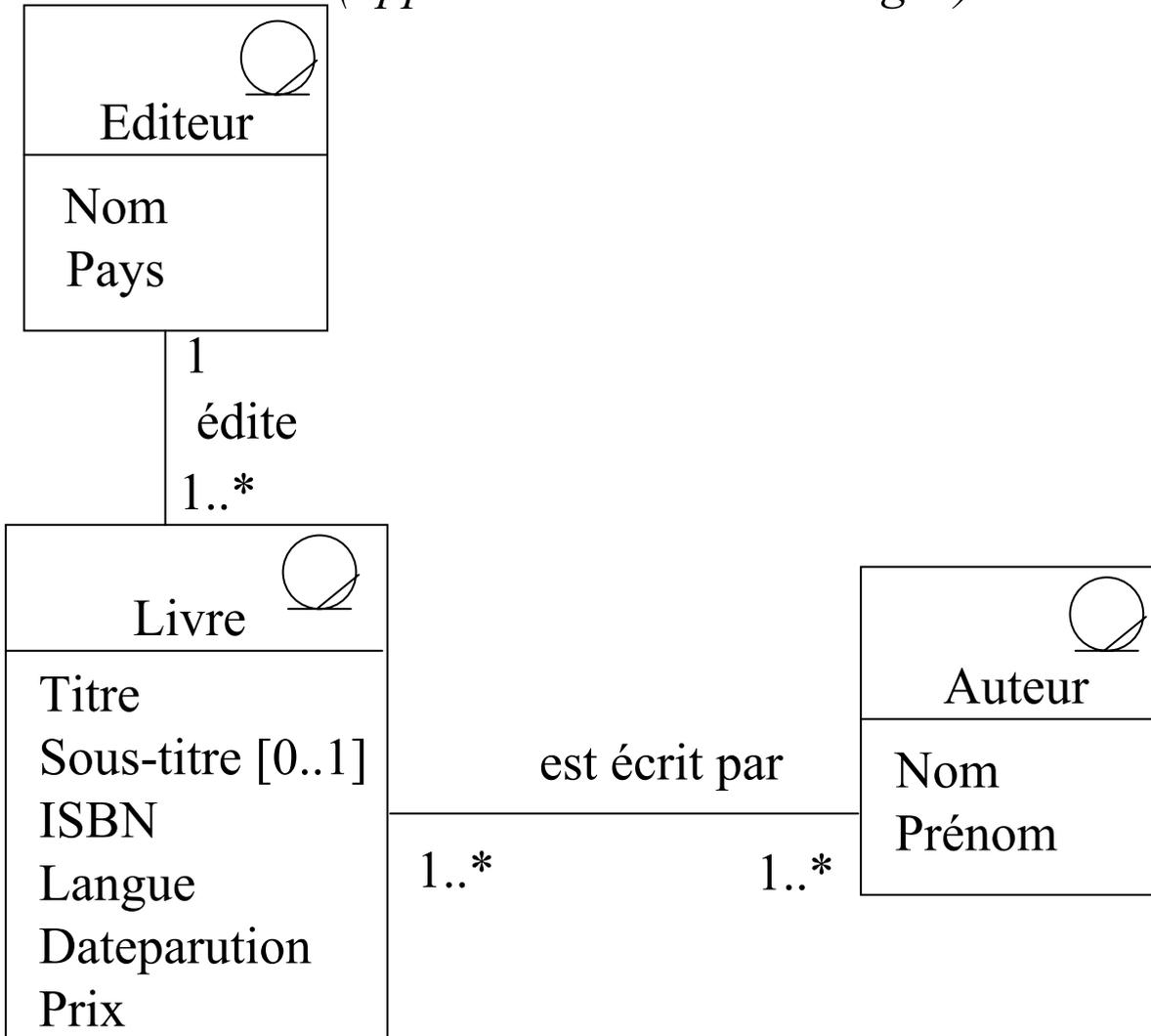


Modèle du domaine

- Les concepts du domaine peuvent être identifiés directement à partir de la connaissance du domaine ou par interview des experts métier.
- Le modèle du domaine, selon la syntaxe UML, est constitué d'un ensemble de classes dans lesquelles aucune opération n'est définie (elles sont ajoutées en phase de conception).
- Etapes de la démarche :
 - * Identifier les concepts du domaine
 - * Ajouter les associations et les attributs
 - * Généraliser les concepts
 - * Structurer en packages : structuration selon les principes de *cohérence* et d'*indépendance*.

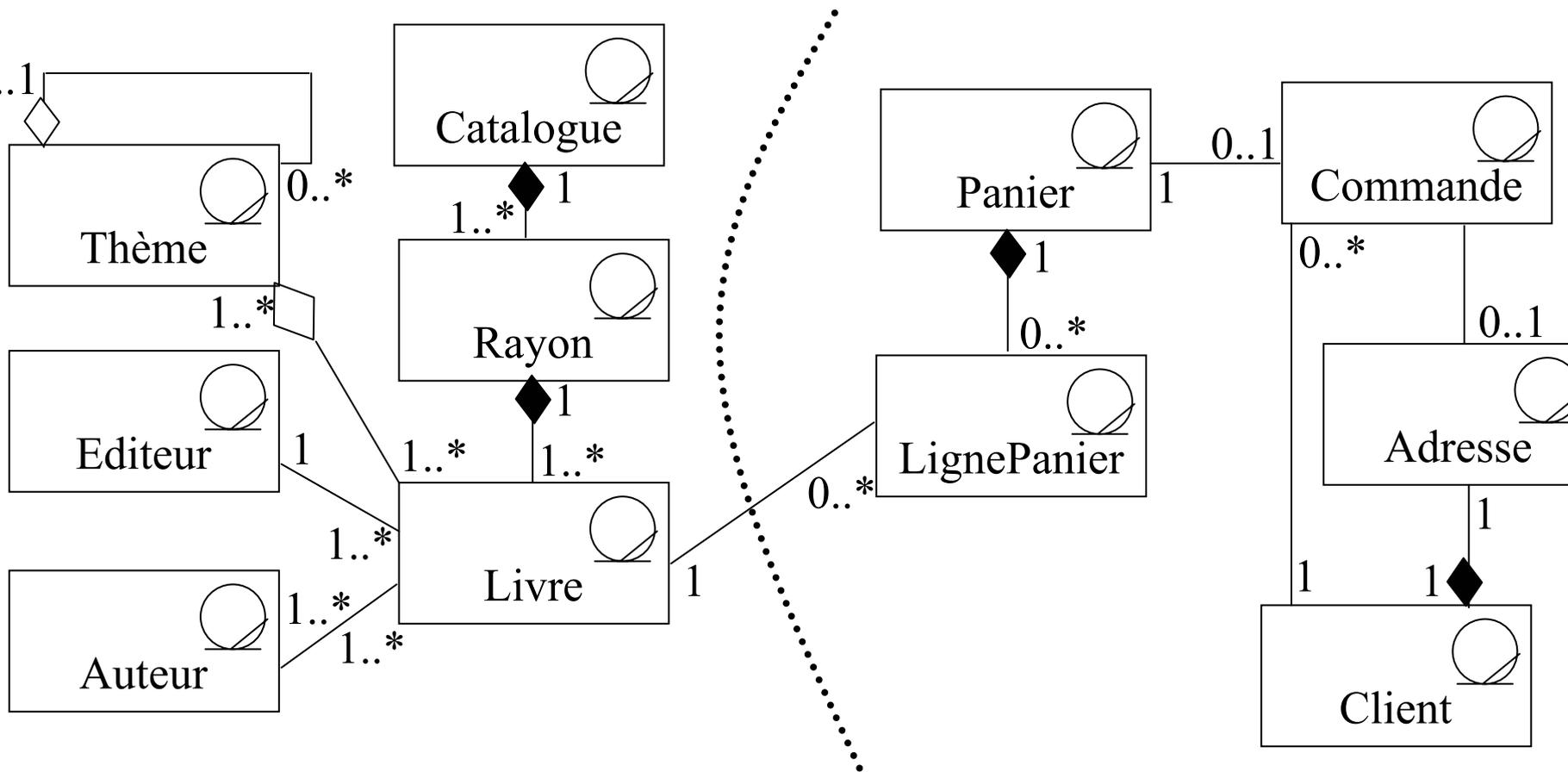
Exemple de modèle du domaine

*Concepts et attributs liés à la recherche d'ouvrages
(application librairie en ligne)*



Proposition de découpage en packages

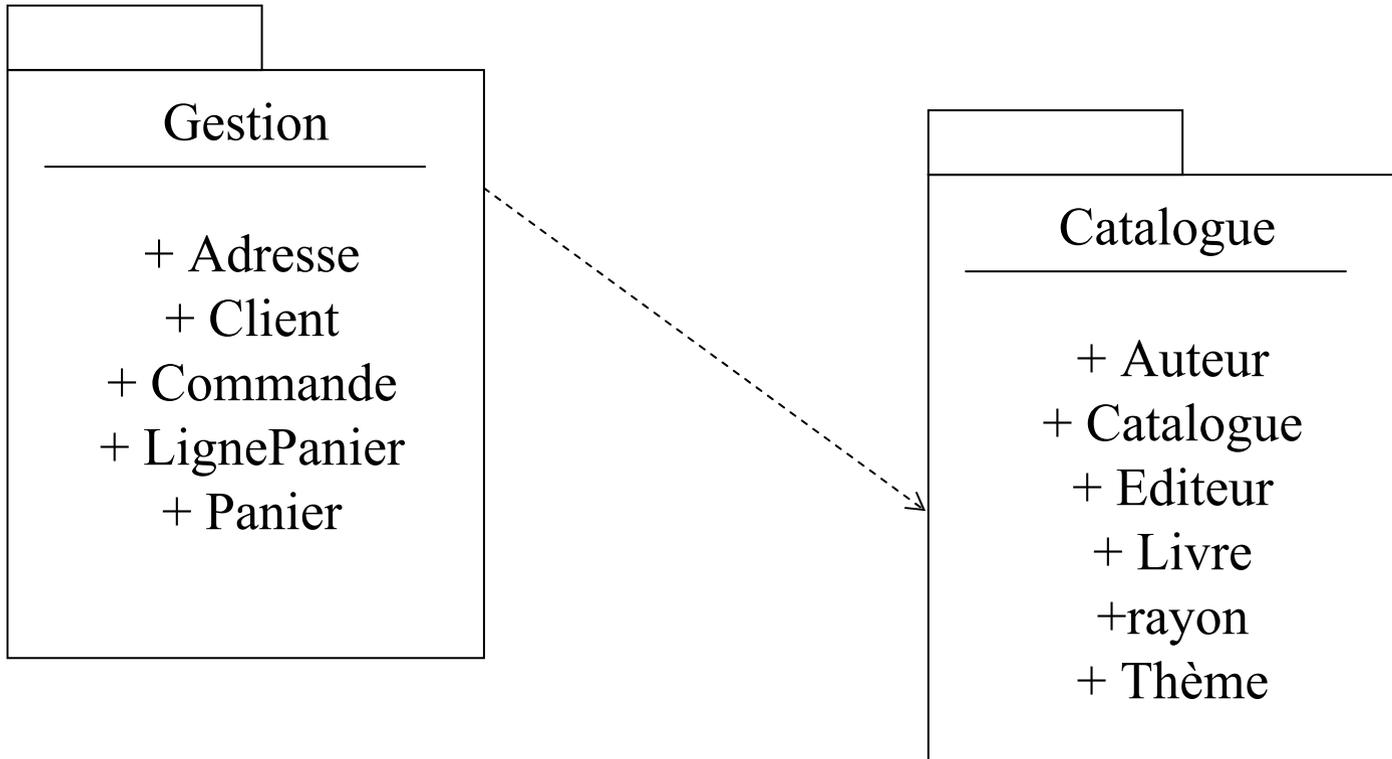
(application librairie en ligne)



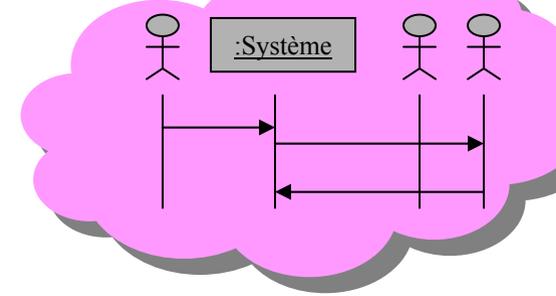
Rq : le package est un espace de nommage. Deux éléments ne peuvent porter le même nom au sein d'un même package.

Représentation plus synthétique des packages et de leur contenu

(application librairie en ligne)



II.3 Spécification détaillée des besoins



Diagrammes de
séquence système

- Description textuelle des cas d'utilisation (non normalisée)
- Mise à jour des diagrammes de cas d'utilisation
- Construction des diagrammes de séquence système
- Spécification des opérations *système*

Exemple de description d'un cas d'utilisation

Cas Rechercher des ouvrages :

Acteur principal : l'internaute

Objectifs : L'internaute veut trouver le plus rapidement possible un ouvrage recherché dans l'ensemble du catalogue. Il veut ainsi flâner comme il le ferait dans une vraie librairie et chercher des livres avec des critères variés.

Préconditions : le catalogue est disponible

Postconditions : L'internaute a trouvé l'ouvrage qu'il cherchait ou un ouvrage qui l'intéresse. Il l'a enregistré dans son panier virtuel (la gestion du panier est détaillée dans « Gérer son Panier »)

Scénario nominal :

1. L'internaute lance une recherche rapide à partir d'un ou deux mots-clés : un thème, un titre, le nom d'un auteur. Il peut aussi saisir directement un numéro ISBN.
2. Le système affiche une page de résultat (cf. ...). Les ouvrages sont classés par défaut par date de parution, le plus récent en premier.
3. L'internaute sélectionne un ouvrage
4. Le système lui présente une fiche détaillée (cf. ...) pour l'ouvrage sélectionné. On trouvera : une image (pour la majorité des ouvrages), ses titre, auteur(s), éditeur, date de parution, nombre de pages, langue, son prix et sa disponibilité, d'éventuels commentaires de lecteurs, la table des matières détaillée, des extraits de chapitres, ...
5. L'internaute met l'ouvrage de côté dans son panier virtuel.

Exemple de description d'un cas d'utilisation (suite)

Scénarios alternatifs :

1a. L'internaute n'a pas d'idée préconçue et préfère flâner dans les rayons de la librairie. Le système lui propose un ensemble de pages : meilleures ventes, nouveautés, sélection du libraire (par thème).

- L'internaute navigue dans ces pages et peut enchaîner sur l'étape 3 du scénario nominal.

1.b L'internaute choisit d'effectuer une recherche avancée

- L'internaute accède à un formulaire spécialisé lui permettant de combiner plusieurs types de recherche : par sujet, titre, auteur, éditeur, langue, etc. Il peut ne saisir que le début significatif d'un mot en terminant par *. Les suffixes notamment les pluriel sont supprimés automatiquement pendant la recherche. L'internaute peut utiliser des opérateurs logiques entre les mots de sa recherche (ET, OU, NON)

2.a Le système n'a pas trouvé d'ouvrage correspondant à la recherche

- Le système signale l'échec à l'internaute et lui propose d'effectuer une nouvelle recherche. Le cas d'utilisation redémarre à l'étape 1 du scénario nominal.

2.b Le système a trouvé de très nombreux ouvrages.

- Le système signale le nombre d'ouvrages à l'Internaute et lui affiche une première page de résultats. Les autres pages sont accessibles directement ou par des symboles « suivante » et « précédente ».
- L'internaute navigue dans ces pages et peut enchaîner sur l'étape 3 du scénario nominal. Il peut aussi reclasser les ouvrages obtenus sur différents critères : titre, auteur, langue, disponibilité, etc.

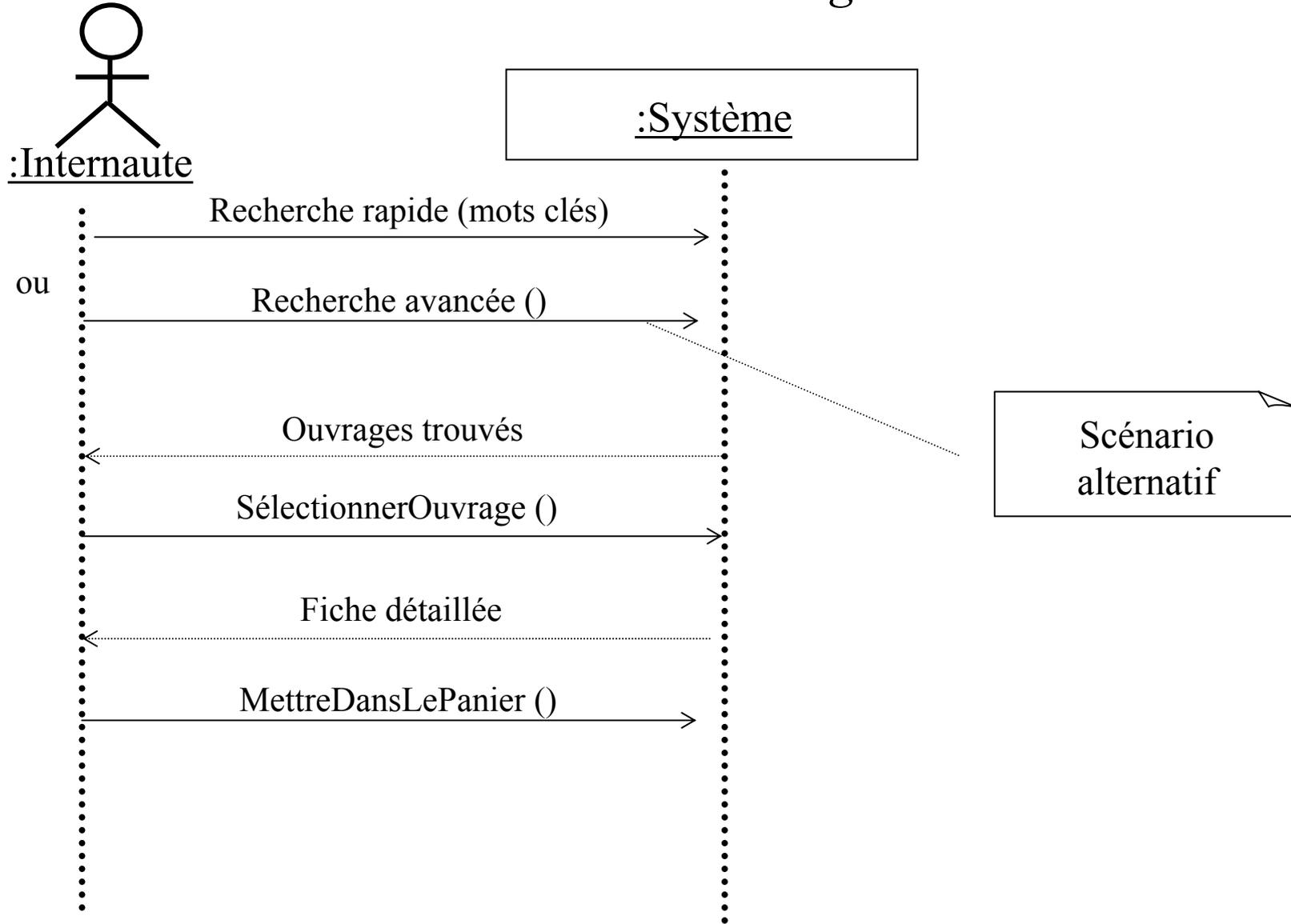
3-5a L'internaute n'est pas intéressé par les résultats

- L'internaute revient à l'étape 1 du scénario nominal pour lancer une nouvelle recherche.

1. L'internaute abandonne la recherche. Le cas d'utilisation se termine (échec).

Diagramme de séquence système

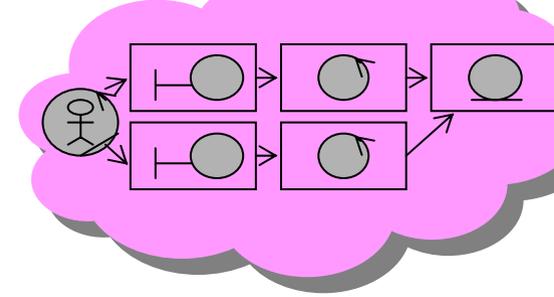
rechercher des ouvrages



Opérations système

<code><<system>></code> NomApplication
AccéderAuPanier() ClasserRésultatsRecherche() commanderPanier() MettreDansLePanier() RechercheRapide() RechercheAvancée() SélectionnerOuvrage()

II.4 La réalisation des cas d'utilisation : classes d'analyse

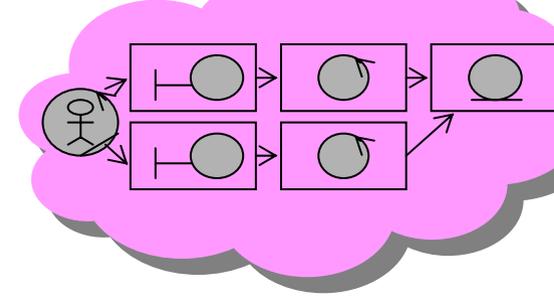


Diagrammes des classes participantes

Typologie des classes d'analyse

- Les classes dialogue : elles permettent les interactions entre les utilisateurs et l'application
- Les classes contrôle : elles contiennent la cinématique de l'application et font la transition entre les classes dialogues et les classes métiers.
- Les classes métier ou entités : elles représentent les objets métier. Elles proviennent directement du modèle du domaine (mais peuvent être complétées en fonction des cas d'utilisation).

II.4 La réalisation des cas d'utilisation : classes d'analyse

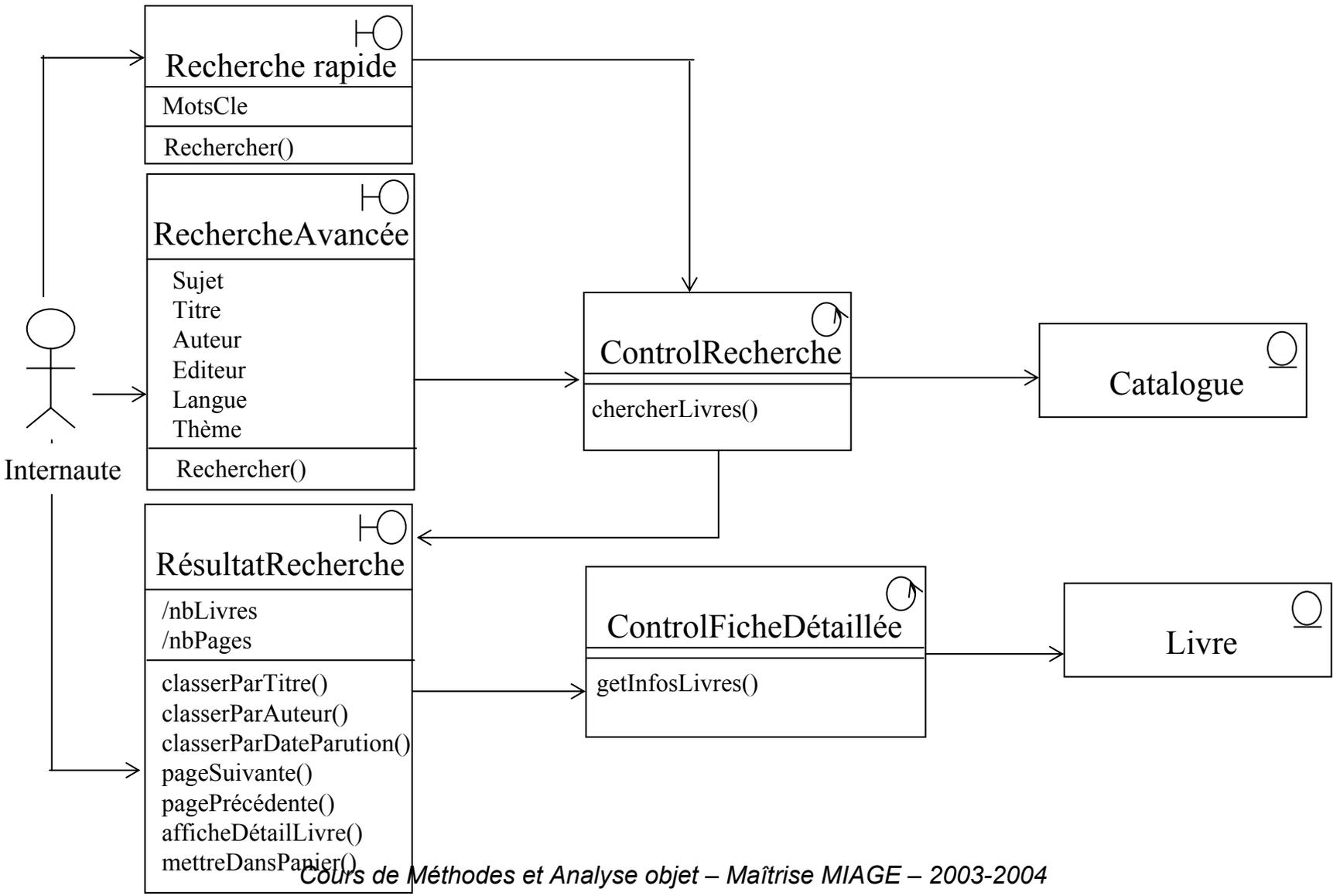


Diagrammes des classes participantes

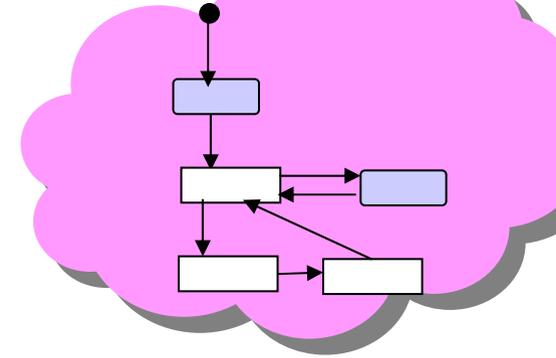
Le diagramme des classes participantes

- Un diagramme de classes UML décrivant les principales classes d'analyse dans lequel on ajoute les acteurs
- Les attributs et les opérations :
 - Les classes métier n'ont que des attributs (cf. modèle du domaine).
 - Pas d'attribut pour les classes contrôle, uniquement des opérations.
 - Les dialogues ont des attributs et des opérations (attributs : champs de saisie ou de résultats, opérations : actions de l'utilisateur sur l'IHM)
- Les associations :
 - Les dialogues ne peuvent être reliés qu'aux contrôles ou à d'autres dialogues (en général, associations unidirectionnelles)
 - Les classes métier ne peuvent être reliées qu'aux contrôles ou à d'autres classes métier.
 - Les contrôles ont accès à tous les types de classes.

Diagramme de classes participantes de « Effectuer une recherche »



II.5 La modélisation de la navigation



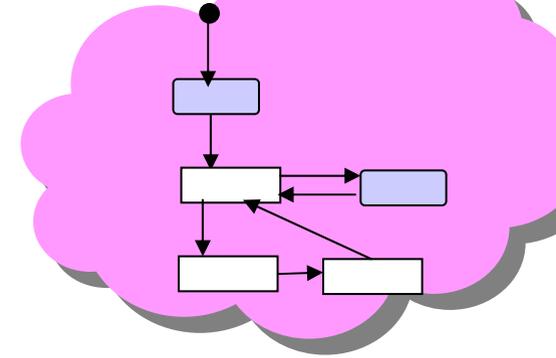
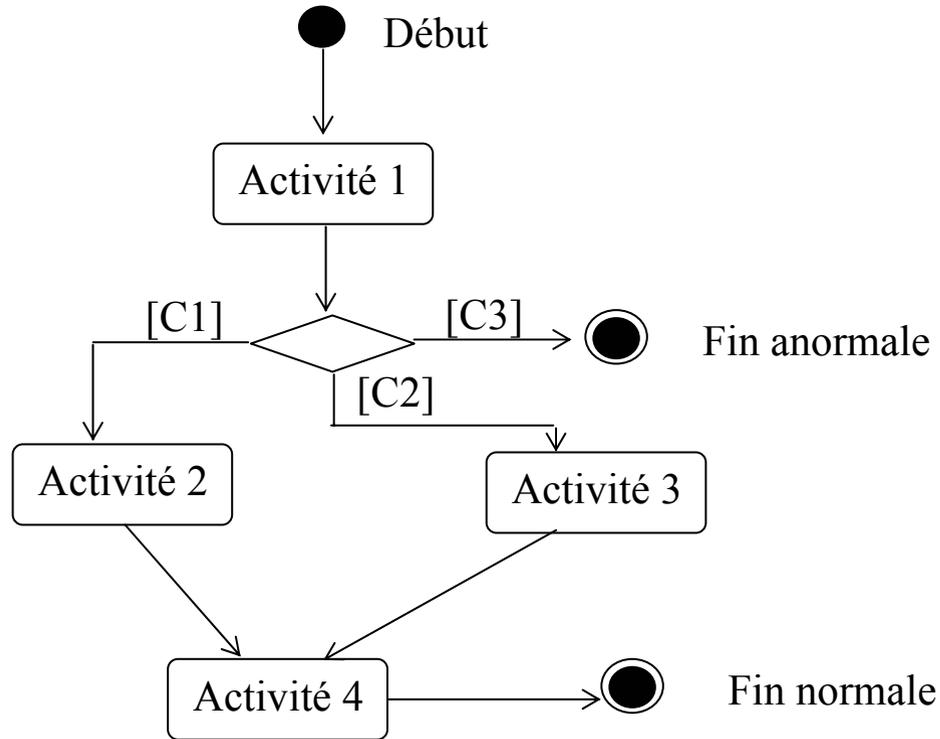
Diagrammes d'activités de navigation

- Il s'agit d'exploiter les maquettes de façon à représenter l'ensemble des chemins possibles entre les principaux écrans proposés à l'utilisateur.

- La notation utilisée est celle du *diagramme d'activités*, un diagramme UML, ressemblant à un organigramme, dont les éléments de base sont : les activités, les transitions entre activités pouvant comporter des conditions, des branchements conditionnels, un début et une (ou plusieurs) terminaison(s).

Une activité est une opération non atomique exécutée dans un état. Elle peut être interrompue et a une certaine durée à la différence d'une action qui est une opération atomique ininterrompue.

II.5 La modélisation de la navigation



Diagrammes d'activités de navigation

Notation graphique du diagramme d'activités

II.5 La modélisation de la navigation

Conventions spécifiques

<<page>>

Une page complète du site

<<frame>>

Un frame particulier

<<action>>

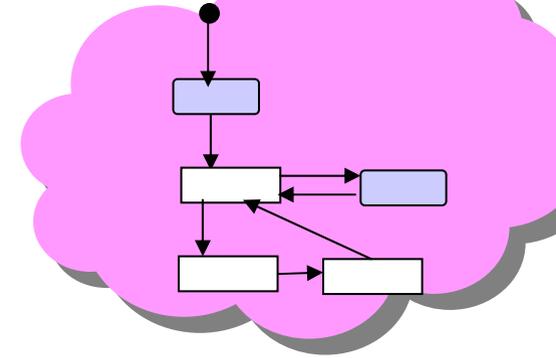
Une action simple

<<exception>>

Une erreur ou une action inattendue du système

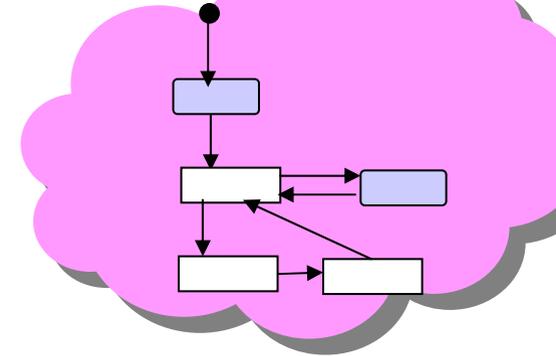
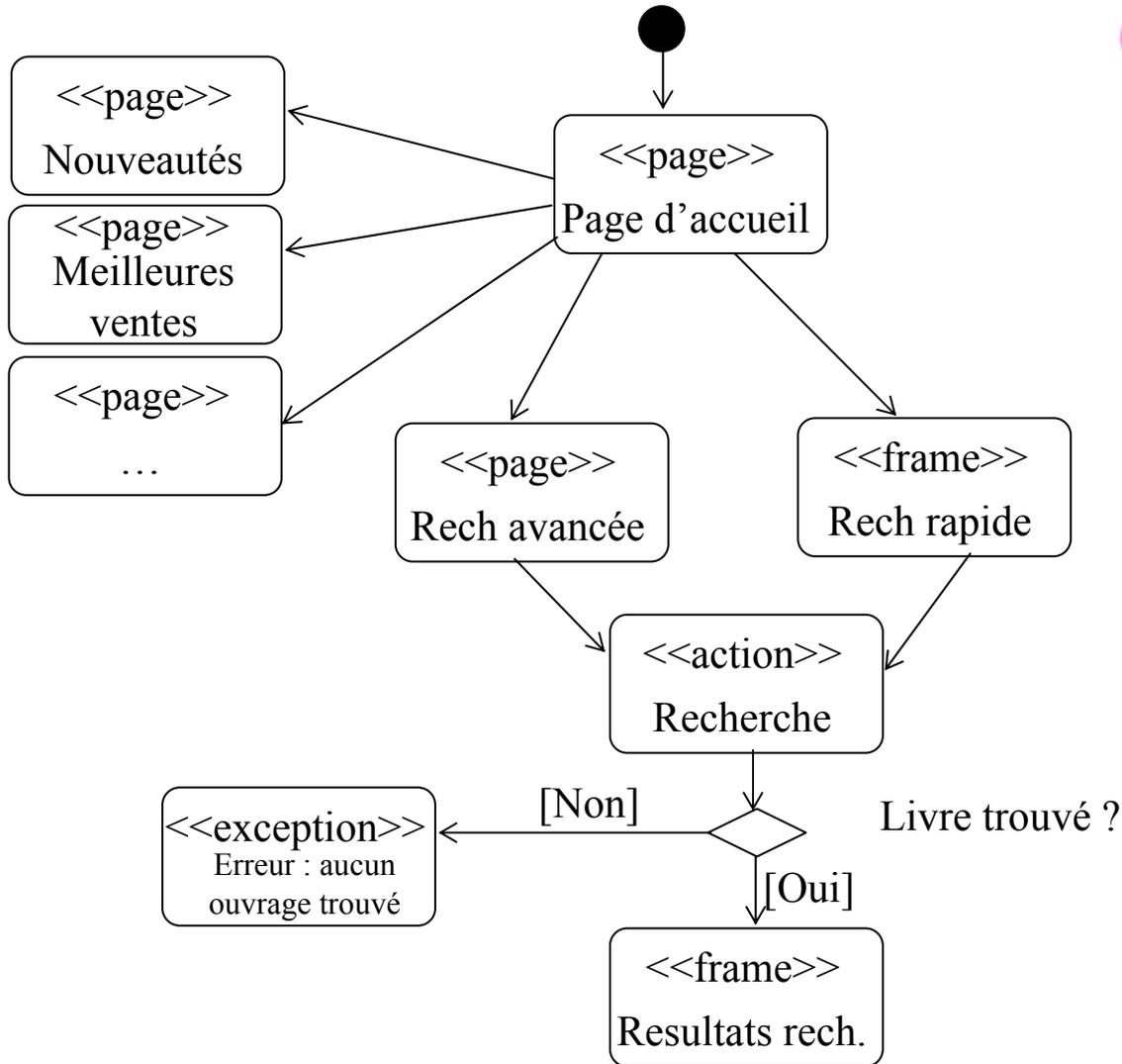
<<connector>>

Une liaison vers un autre diagramme d'activités



Diagrammes d'activités de navigation

Exemple : navigation de la recherche (début)



Diagrammes d'activités de navigation

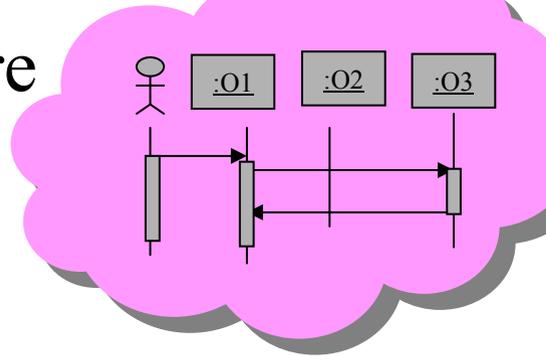
Plusieurs transitions sortant d'une même activité représentent un choix pour l'utilisateur.

Les tâches doivent être séparées en ensembles maîtrisables et les plus indépendants possibles.

➔ La modélisation de la navigation peut se structurer par acteur.

II.6 La conception objet préliminaire

- Pour chaque service ou fonction, il faut décider quelle est la classe qui va le contenir.
- Les diagrammes d'interactions sont très utiles pour représenter graphiquement les décisions d'allocations des responsabilités.
- Chaque diagramme va représenter un ensemble d'objets de classes différentes collaborant dans le cadre d'un scénario d'exécution du système. Les objets communiquent en s'envoyant des messages qui invoquent des opérations sur les objets récepteurs. On peut ainsi suivre visuellement les interactions dynamiques entre objets, et les traitements réalisés par chacun d'eux.
- Par rapport aux diagrammes de séquences système, nous remplaçons le système, vu comme une boîte noire, par un ensemble d'objets en collaboration. Des objets des 3 types de classes interviennent.



Diagrammes d'interaction

Le passage de l'analyse à la conception préliminaire

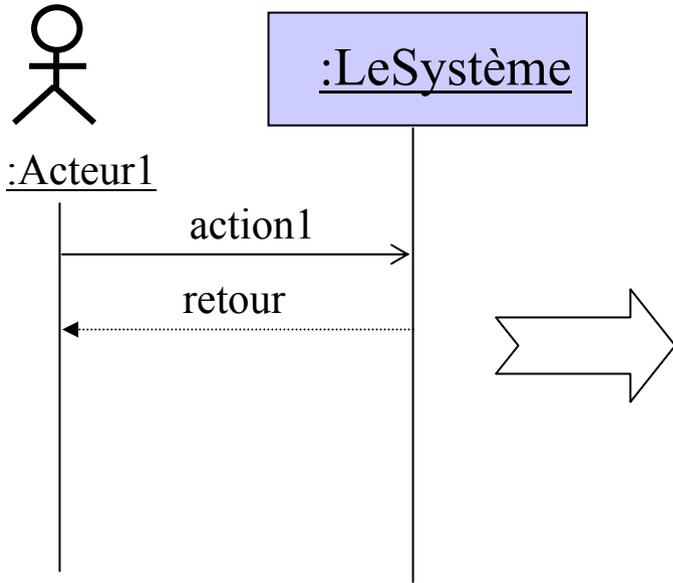


Diagramme de séquence système

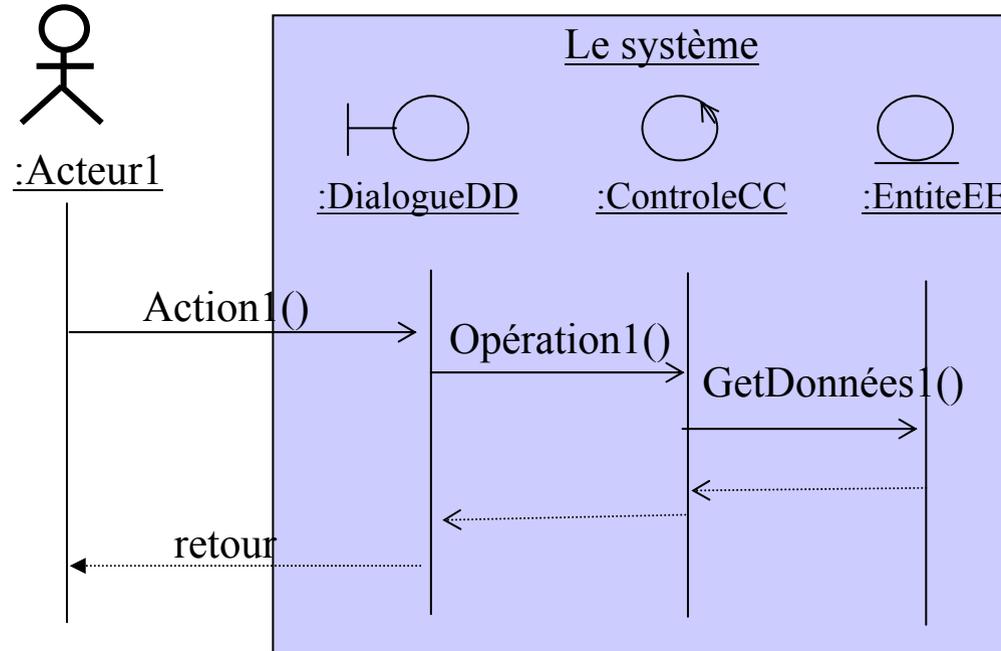
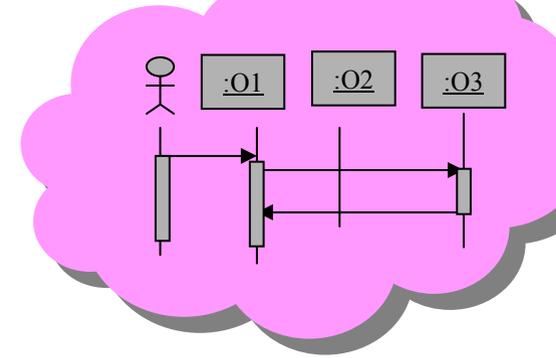
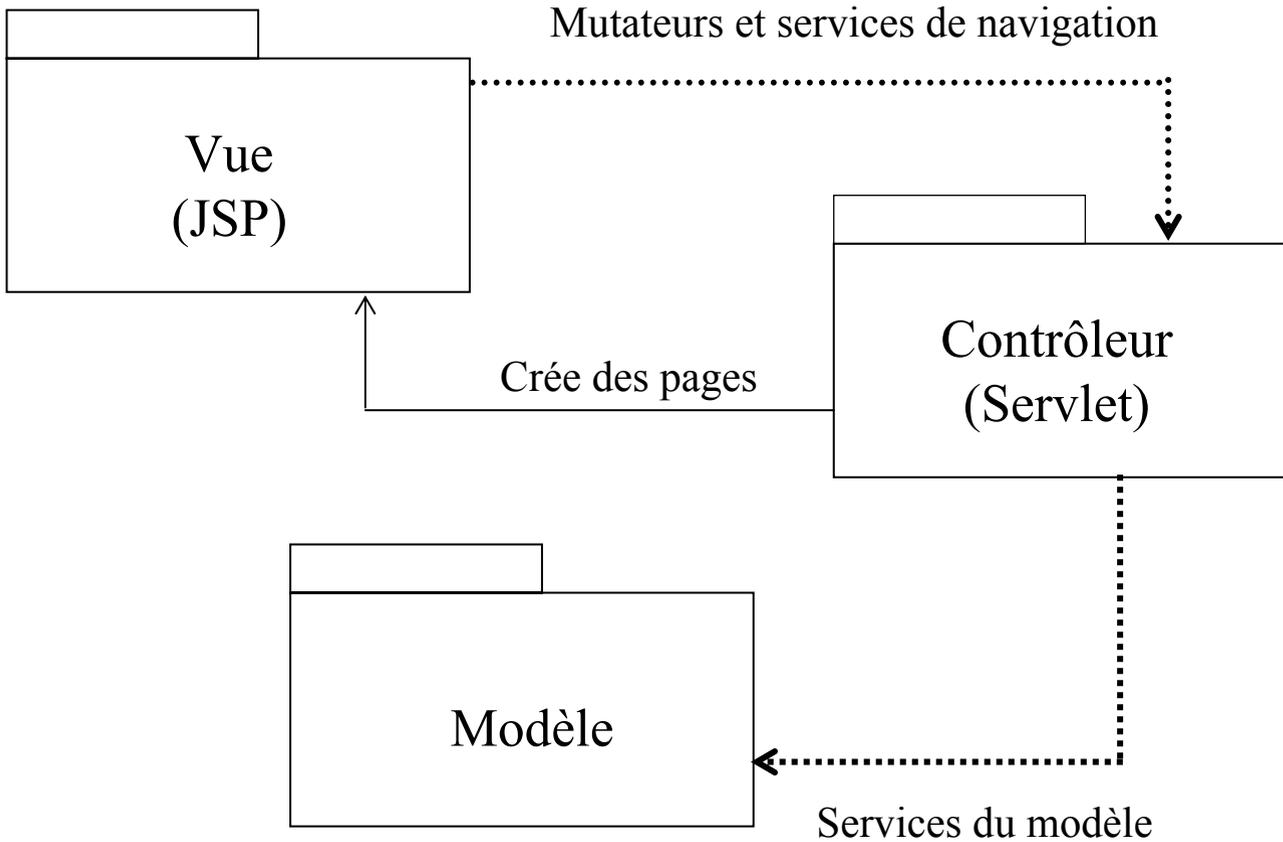


Diagramme de séquence représentant les interactions entre objets

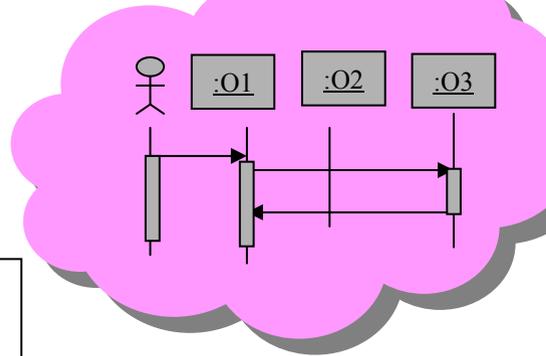
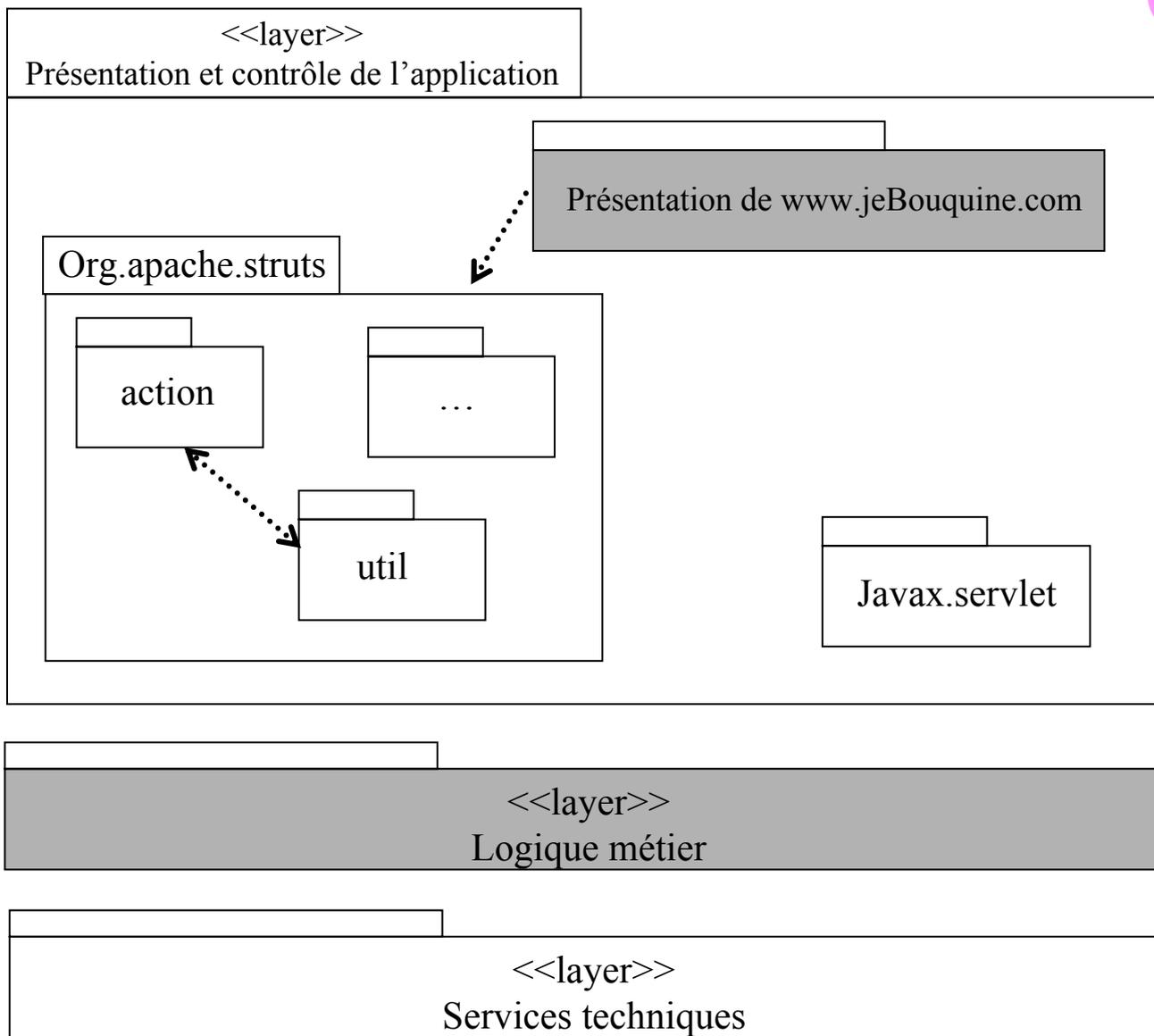
II.7 La conception objet détaillée (solution technique J2EE)



Diagrammes
d'interaction



II.7 La conception objet détaillée (solution technique J2EE)



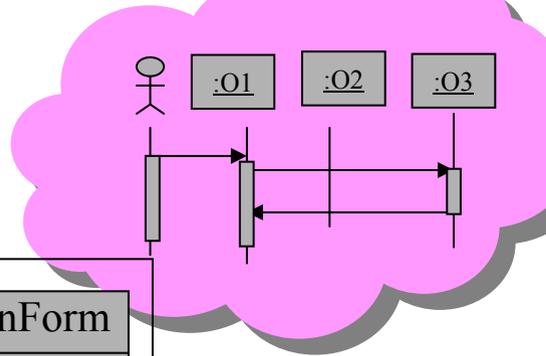
Diagrammes d'interaction

Packages et couches liées à Struts

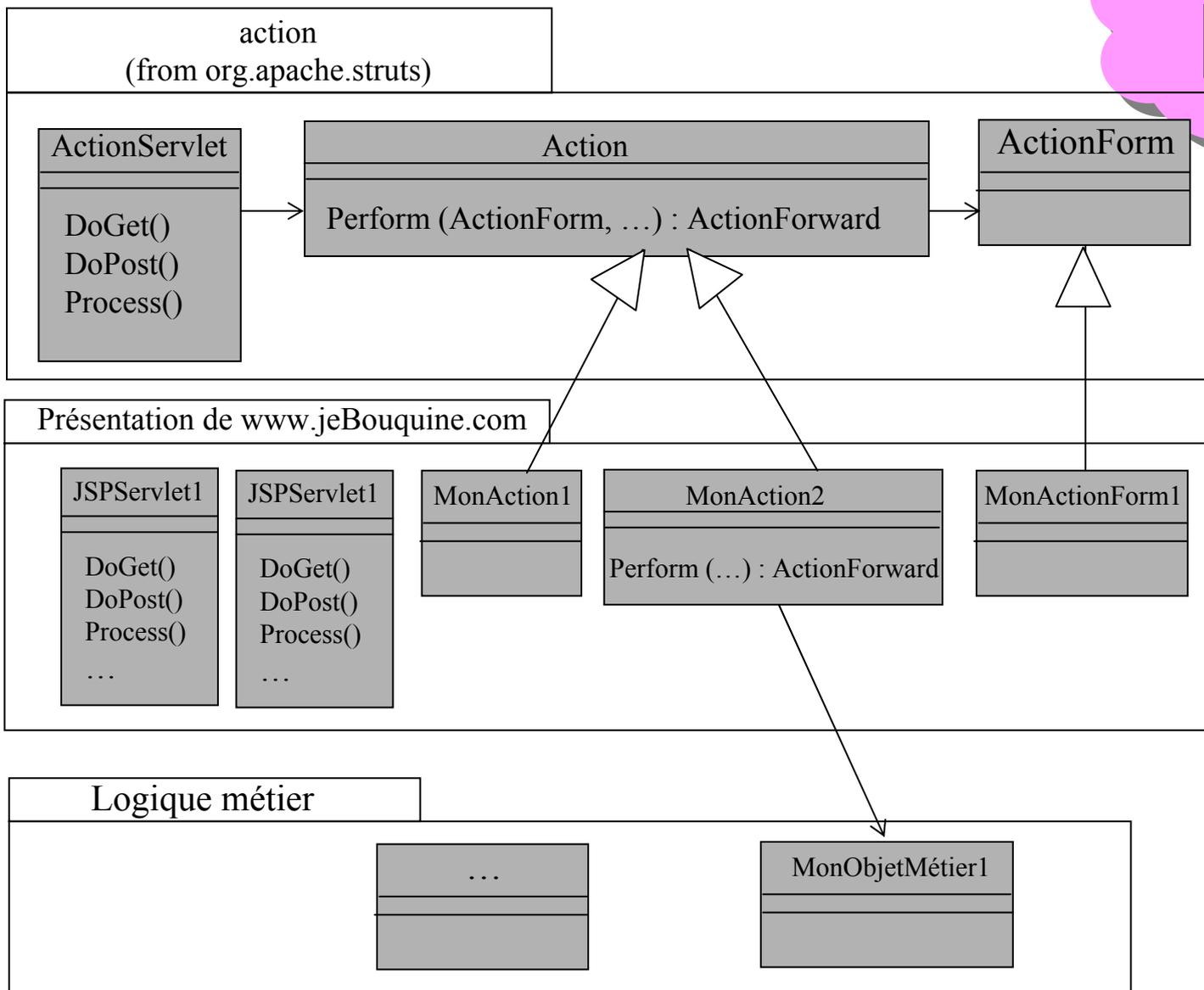
Légende :

- Package en blanc : réutilisable ou standard
- Package en grisé : classes codées par le concepteur

II.7 La conception objet détaillée (solution technique J2EE)



Diagrammes d'interaction



Modèle-
 Vue-
 Contrôleur
 dans Struts

II.7 La conception objet détaillée

(solution technique J2EE)

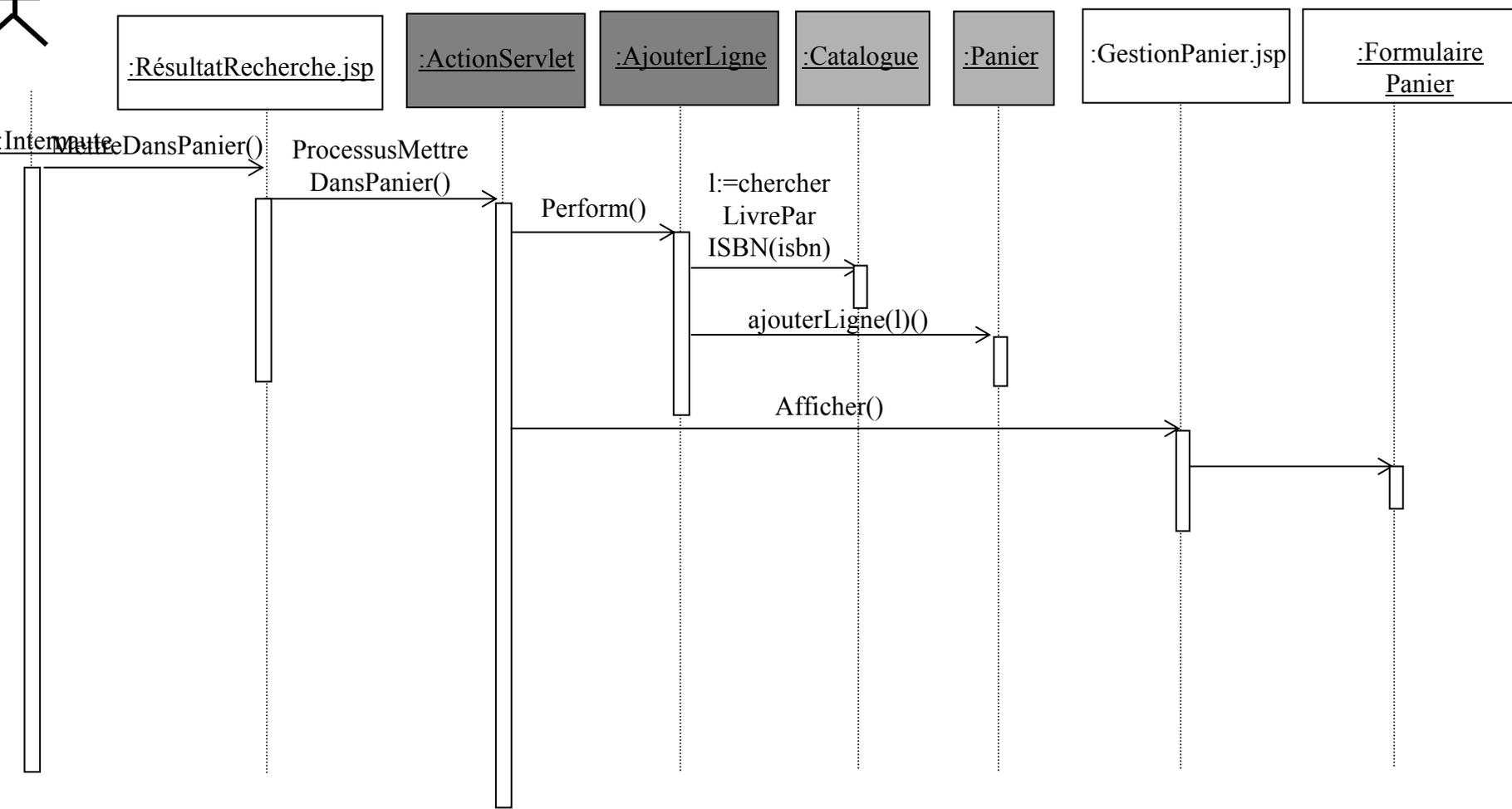
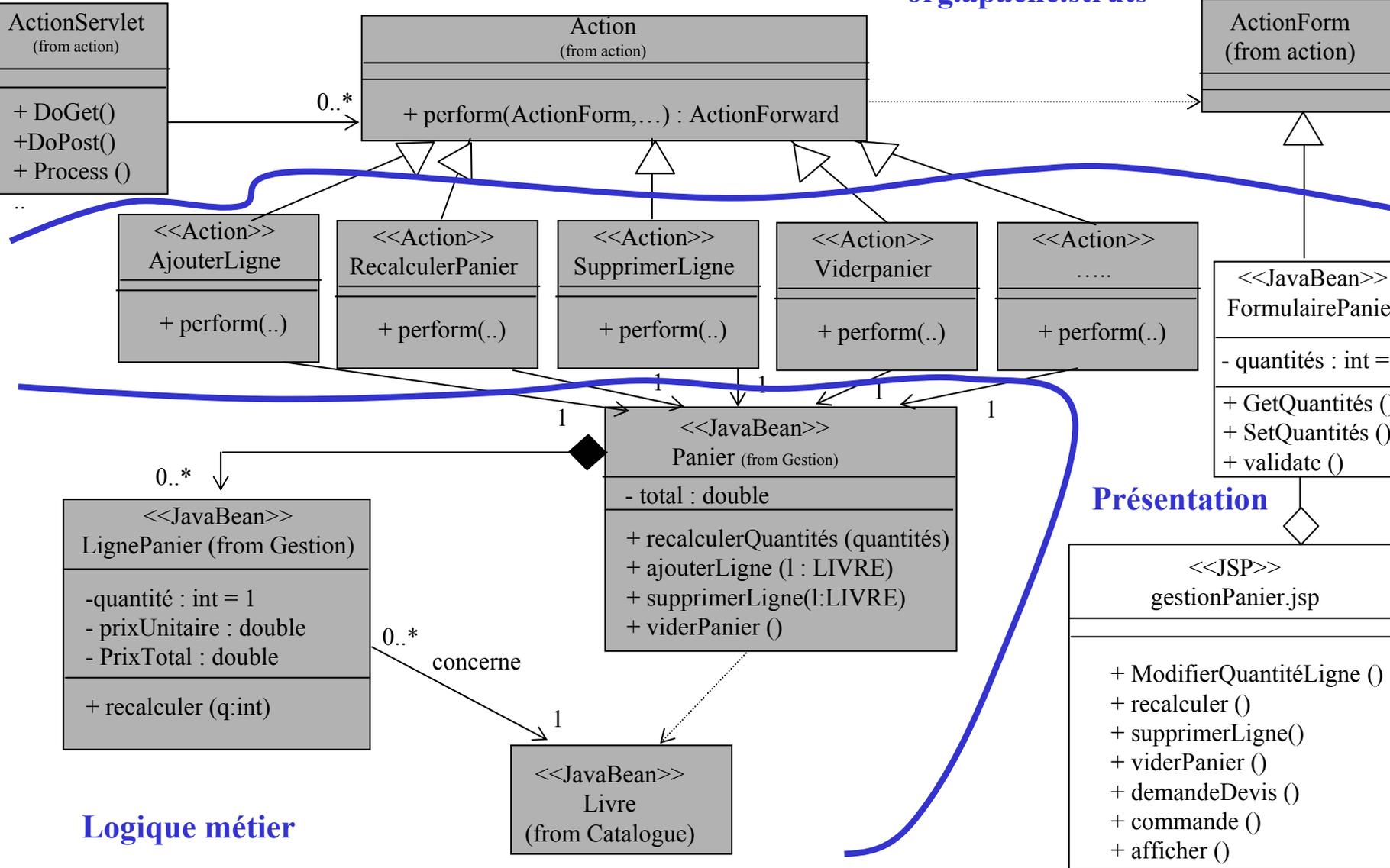


Diagramme de séquence de la création d'une ligne du panier avec Struts

II.7 La conception objet détaillée

Ex de diagramme de classes de conception détaillée

org.apache.struts



Logique métier

Présentation

II.7 La conception objet détaillée

(solution technique J2EE)

Exemple de code

```

Package LogiqueMetier.Gestion;
Import LogiqueMetier.Catalogue.Livre;
Import java.util.*;
Public class Panier {
    private double total;
    private List lesLignesPanier = new ArrayList();

    public Panier () {}
    public double getTotal ()
    {
        return total;
    }
    public void recalculer (List quantites)
    {
        ....
    }

    Public void ajouterLigne (Livre l)
    {
        // Ne pas ajouter de ligne s'il en existe déjà une pour ce livre
        Iterator i = lesLignesPanier.iterator();
        While (i.hasNext()){
            LignePanier l = (LignePanier)lignesIt.next();
            int qte = ((Integer) quantiteIt.next().intValue());
            l.recalculer(qte);
            total += l.getTotal();
        }
        ...
    }
}

```

- La relation de dépendance entre la classe Livre du package Catalogue se traduit par une directive « import »
- La classe étant un javaBean, son attribut possède des accesseurs.
- Total est en lecture seule. Il n'y a donc pas de méthode setTotal().