

Systemes d'Information Web La méthode UWE

Fayçal Hamdi

à partir de transparents de J. Akoka et I. Comyn-Wattiau

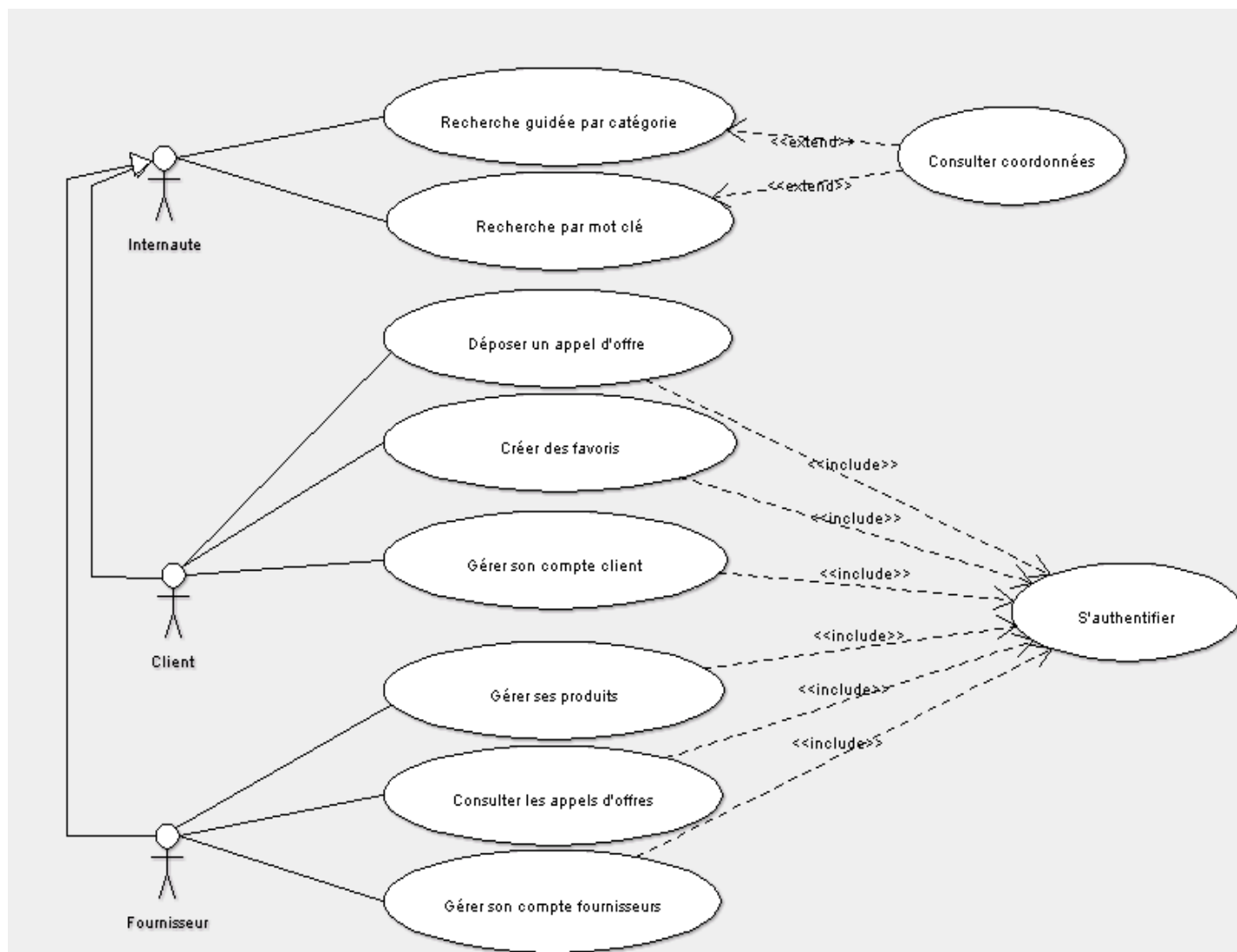
- 3.0 Un premier exemple d'application web
- 3.1 Introduction à La méthode UWE
- 3.2 Les étapes de la méthode
- 3.3 Un deuxième exemple
- 3.4 Modélisation des besoins fonctionnels
- 3.5 Modélisation des contenus
- 3.6 Modélisation de la navigation
- 3.7 Modélisation de la présentation
- 3.8 Modélisation de l'adaptation

3.0 Un premier exemple d'application web

a. Le contexte

- Le site « monfournisseur.com » est le moteur de recherche et de comparaison des prix sur Internet dédiée aux libres services de proximité, un service unique pour les consommateurs.
- Il permet de:
 - Rechercher des produits
 - Rechercher un fournisseur
 - Comparer les prix
 - Lancer des appels d'offre
 - Prendre contact avec le fournisseur pour s'y rendre

b. Diagramme de cas d'utilisation

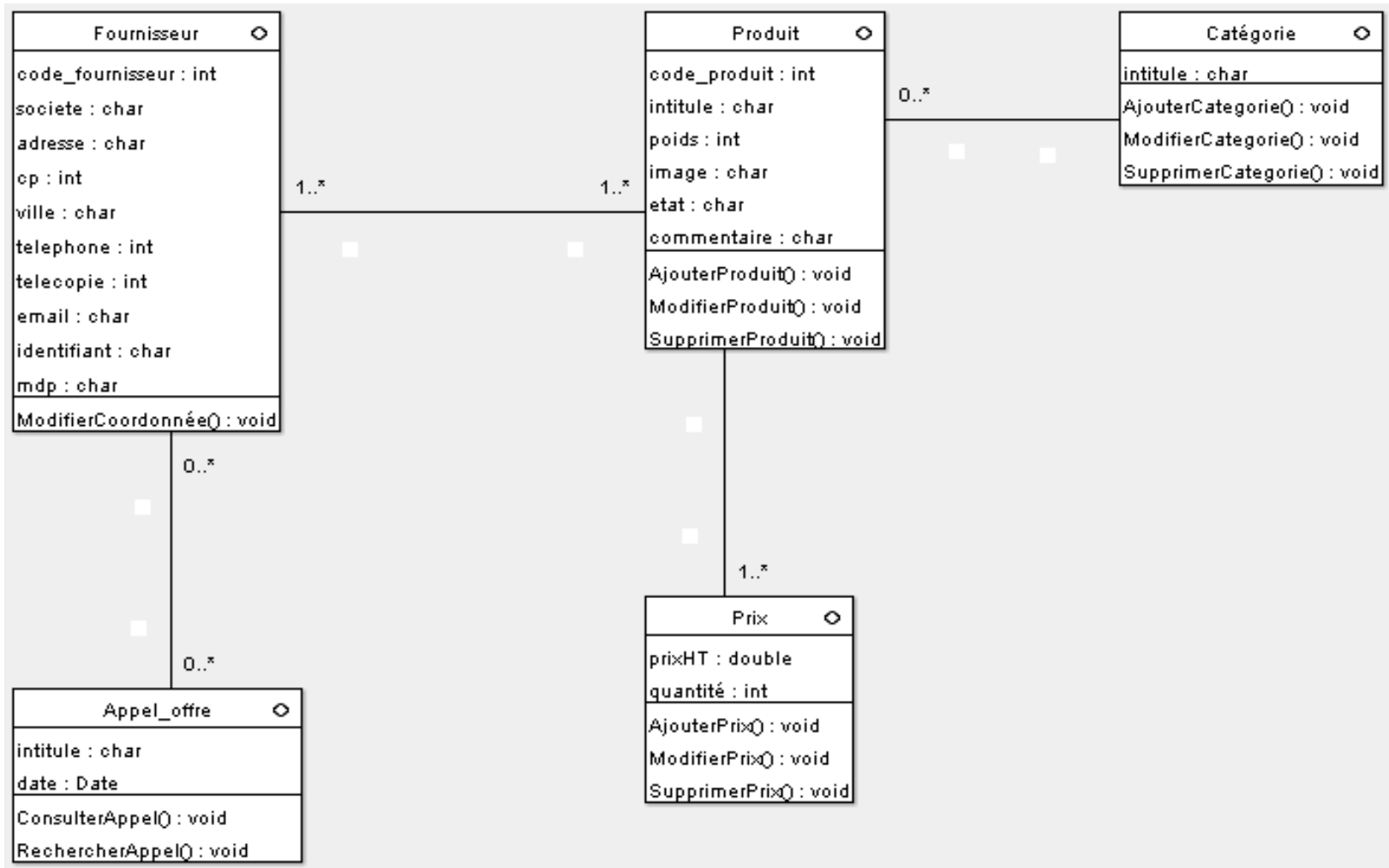


- Il pourra rechercher les prix pratiqués par des fournisseurs d'un produit spécifique en effectuant :
 - Une recherche directe (par mots clé)
 - Une recherche guidée par catégorie (boissons, luminaires...)
- Consulter les coordonnées des fournisseurs (adresse, téléphone, mail...)

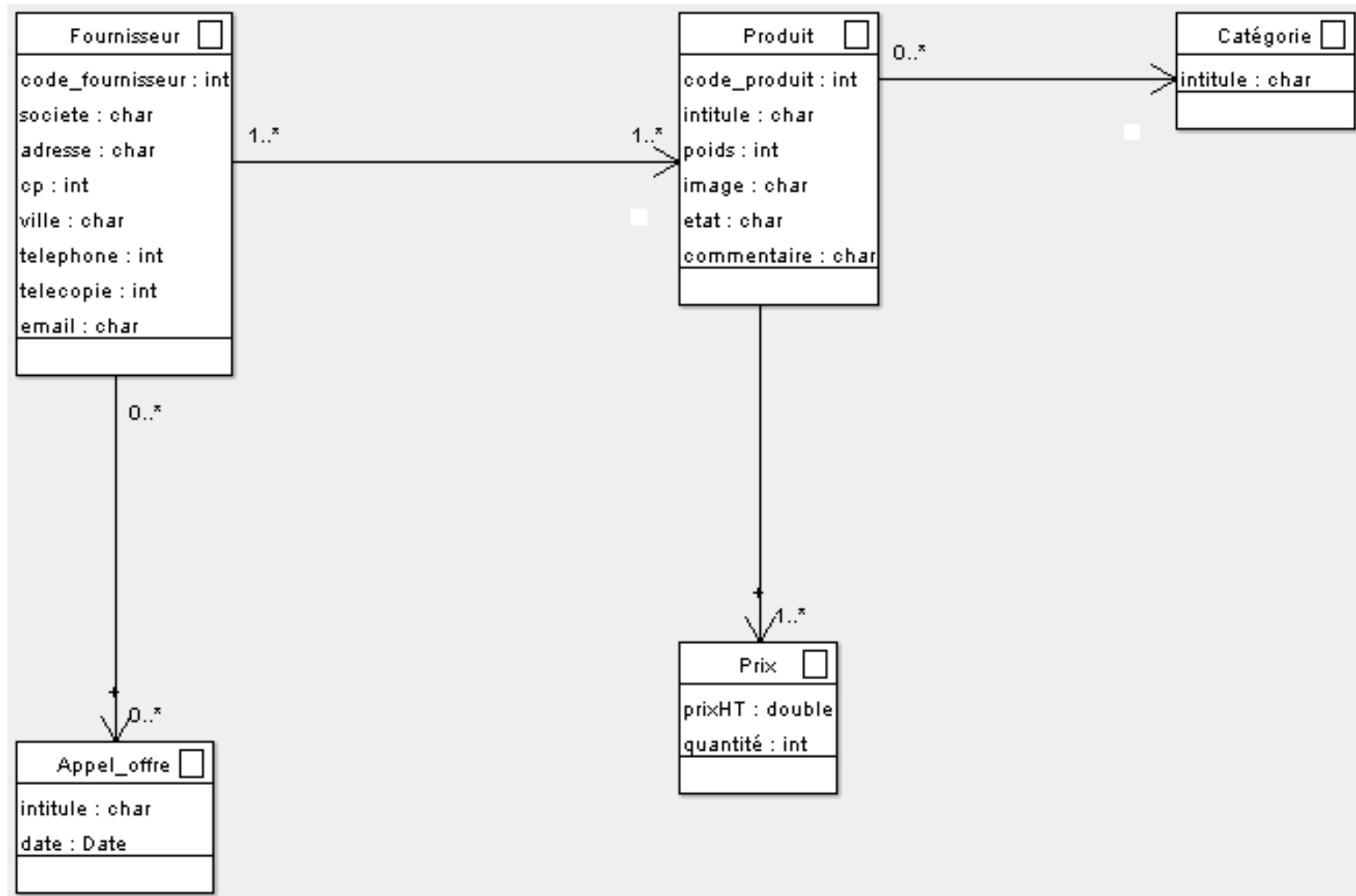
- Déposer des appels d'offre (le client peut déposer une demande auprès des fournisseurs pour un achat particulier, ex. : un gros achat).
- Créer des favoris (c'est-à-dire, classer les produits avec leurs fournisseurs par préférence)
- Gérer son compte client

- Gérer ses produits (ajouter, supprimer, modifier)
- Consulter les appels d'offres
- Gérer son compte fournisseur

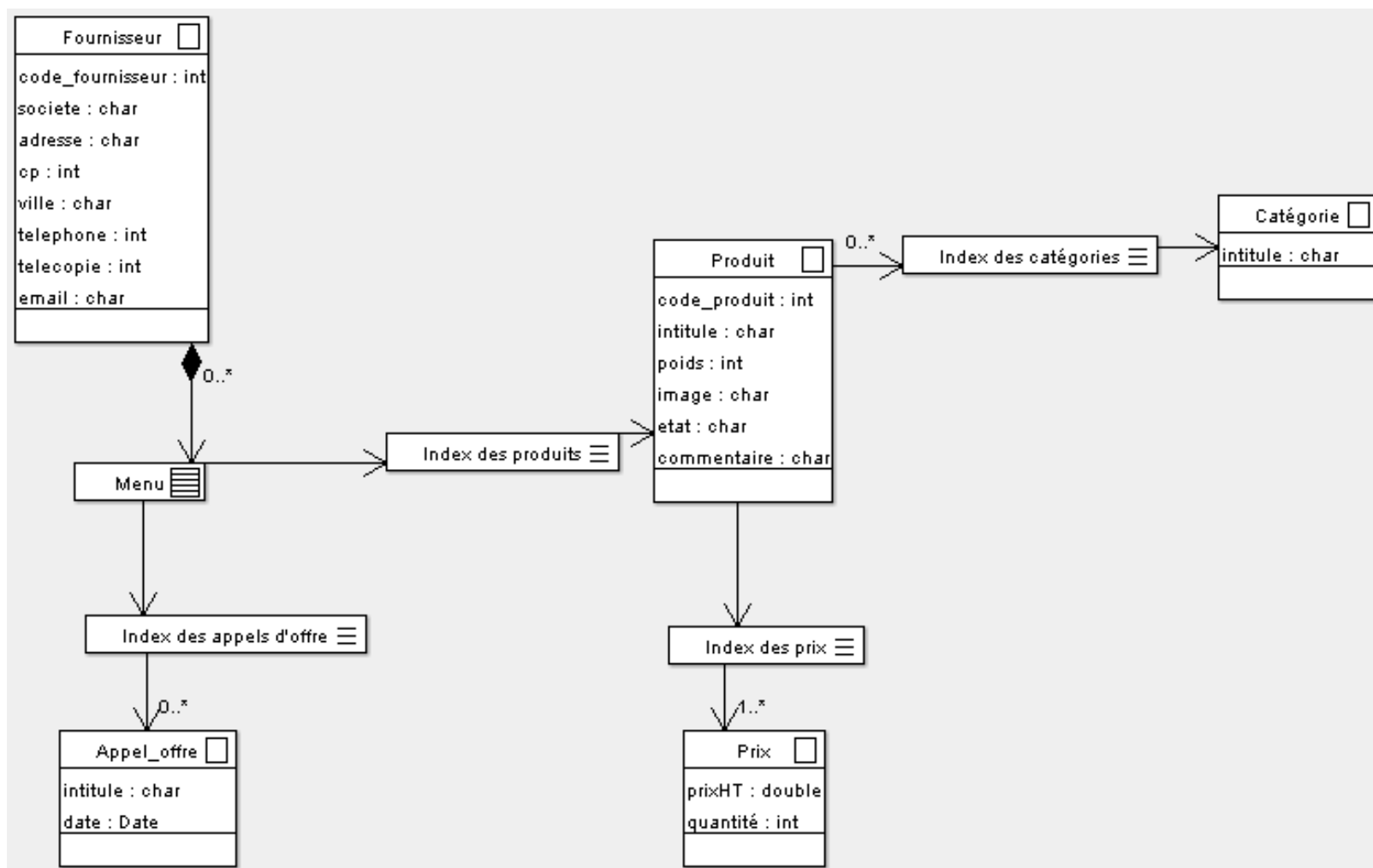
f. Diagramme de contenu (acteur fournisseur)



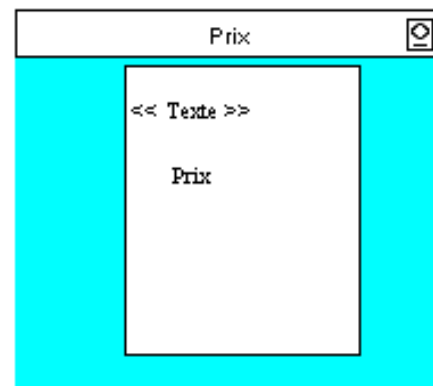
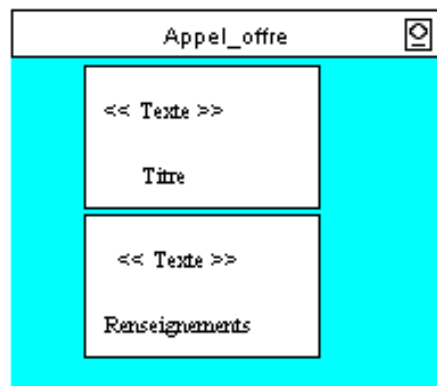
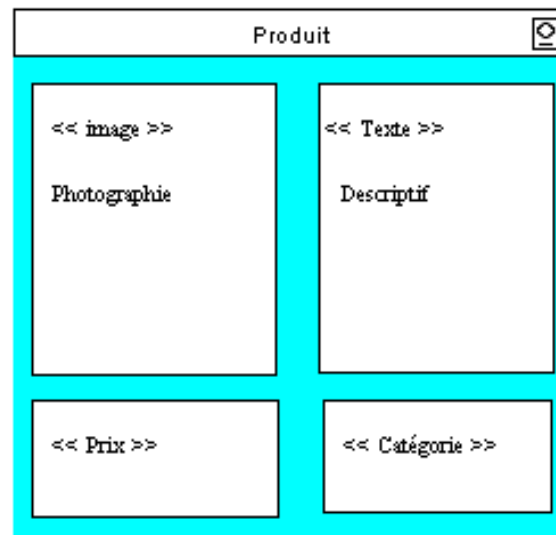
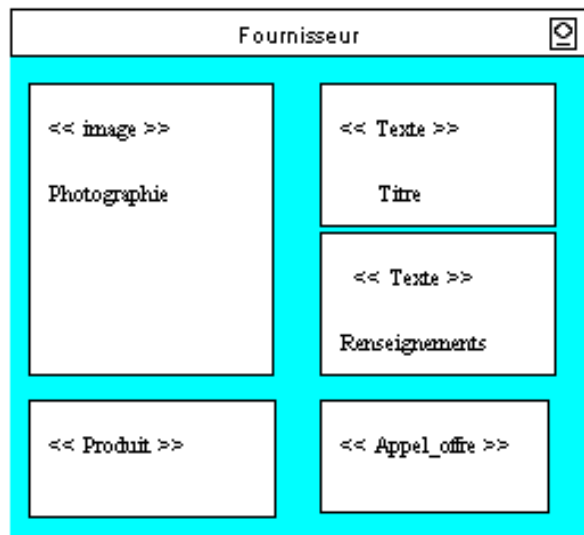
g. Diagramme de la structure de navigation (acteur fournisseur)



h. Diagramme d'accès (acteur fournisseur)



i. Diagramme de présentation (acteur fournisseur)



Exemple 1: site web d'une librairie

Ce site offre la possibilité aux lecteurs de :

- consulter les informations générales sur un ouvrage,
- rechercher un livre:
 - par titre,
 - par auteur,
 - par éditeur
 - par thème.

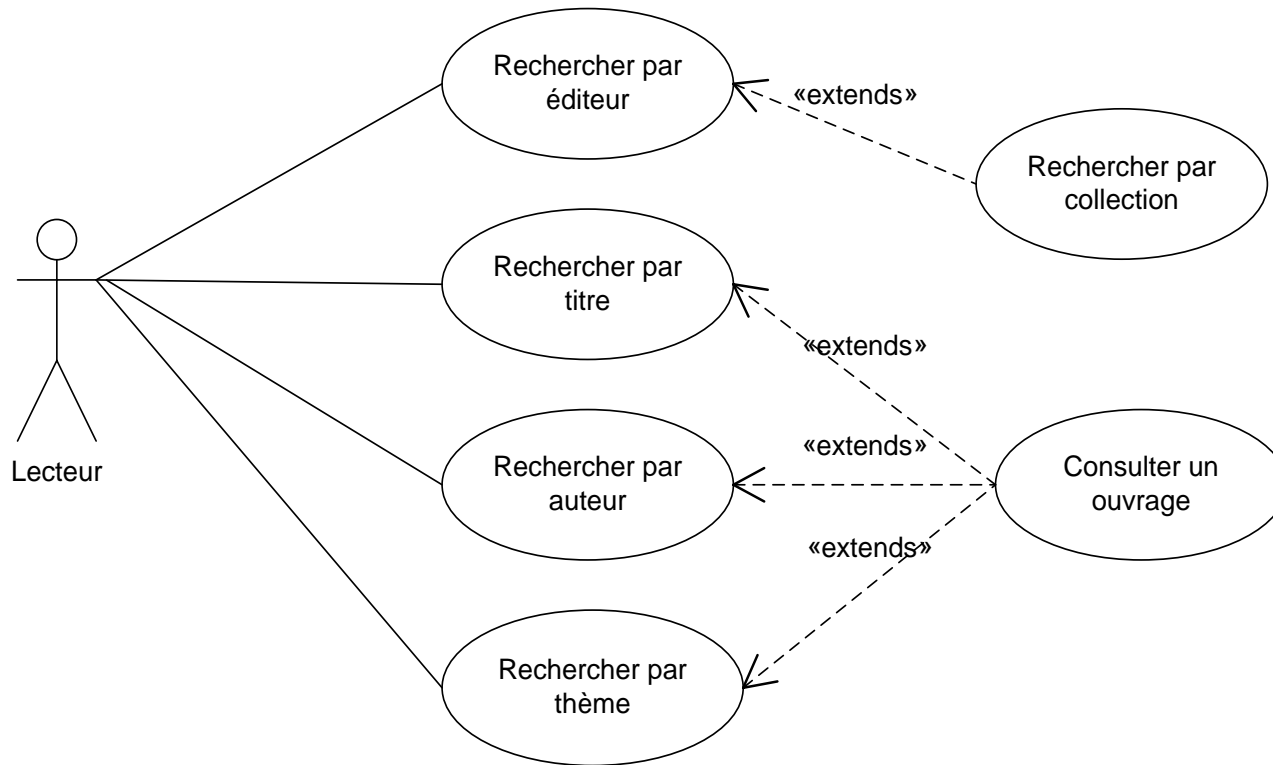


Diagramme de classes

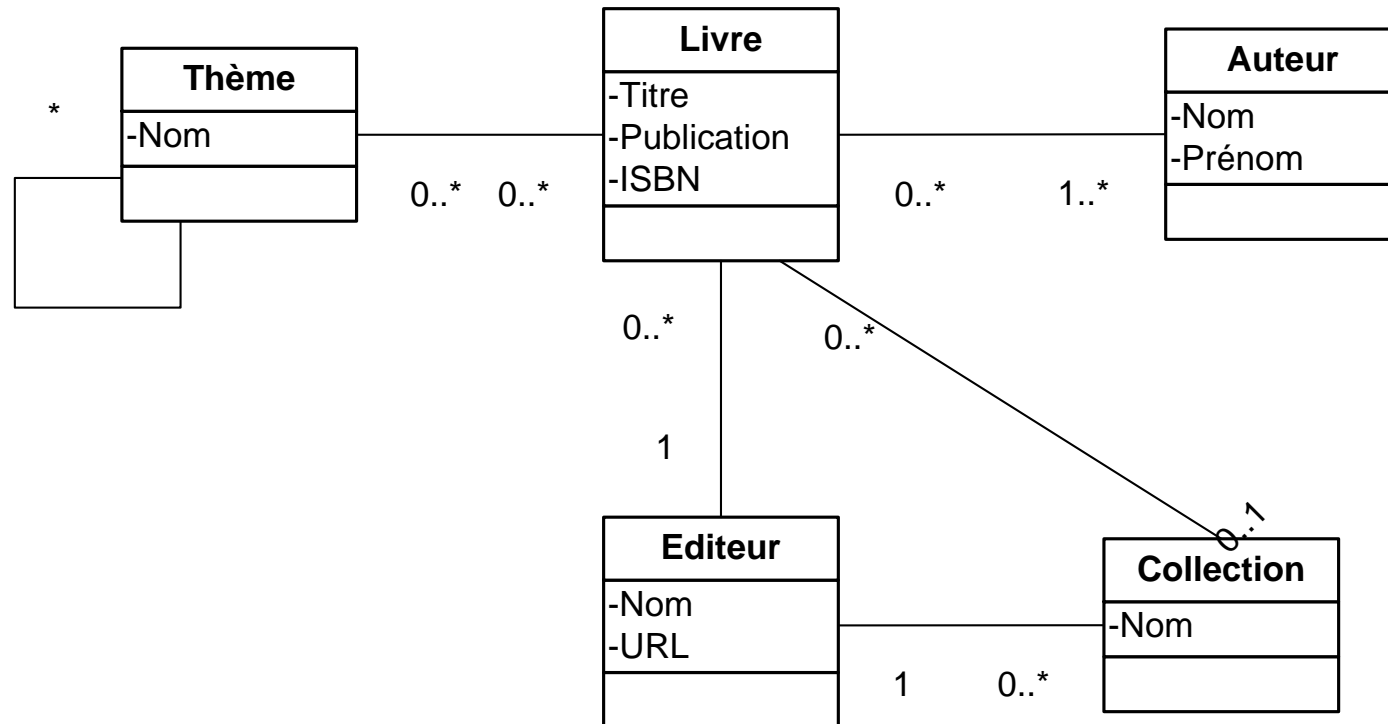


Diagramme de navigation

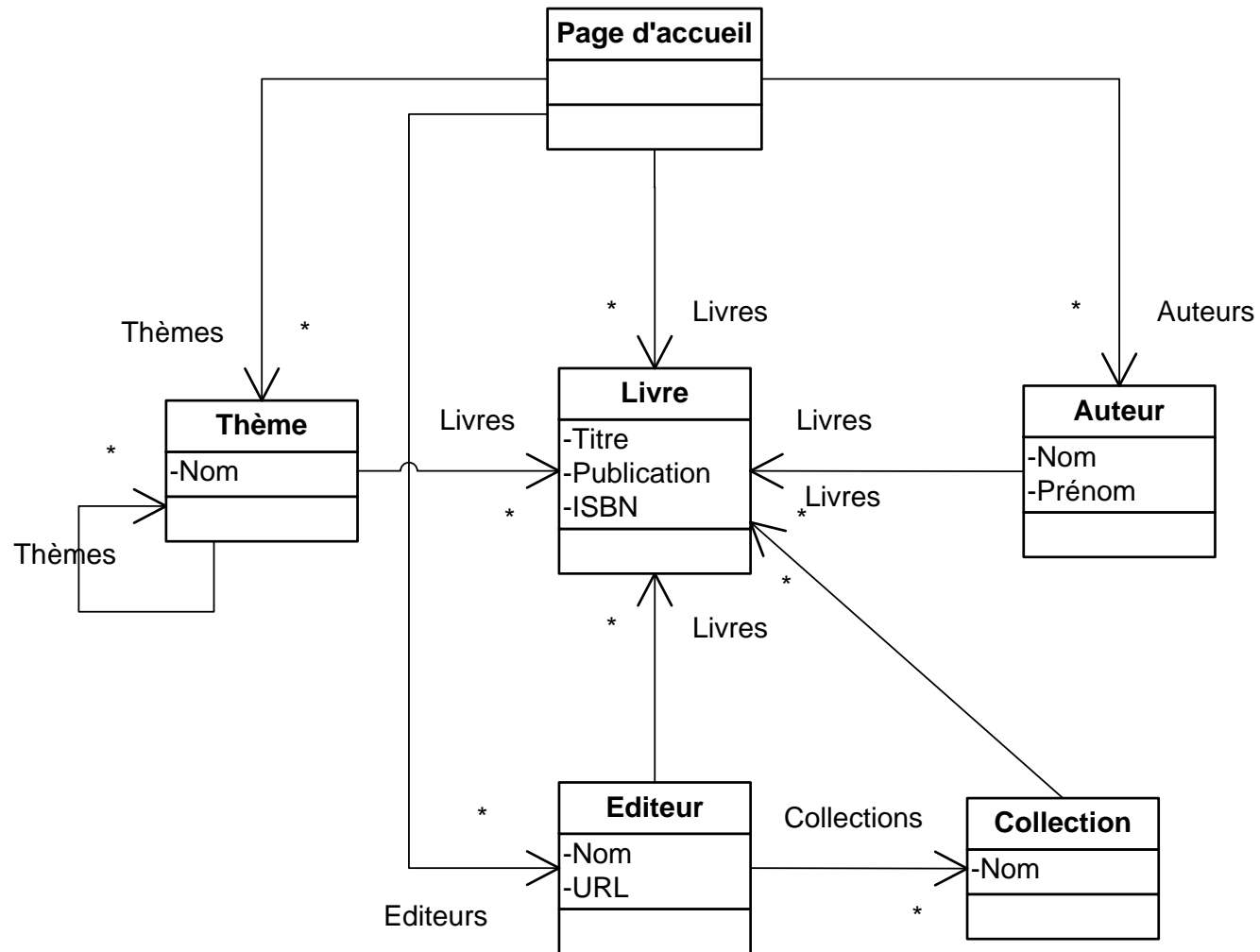
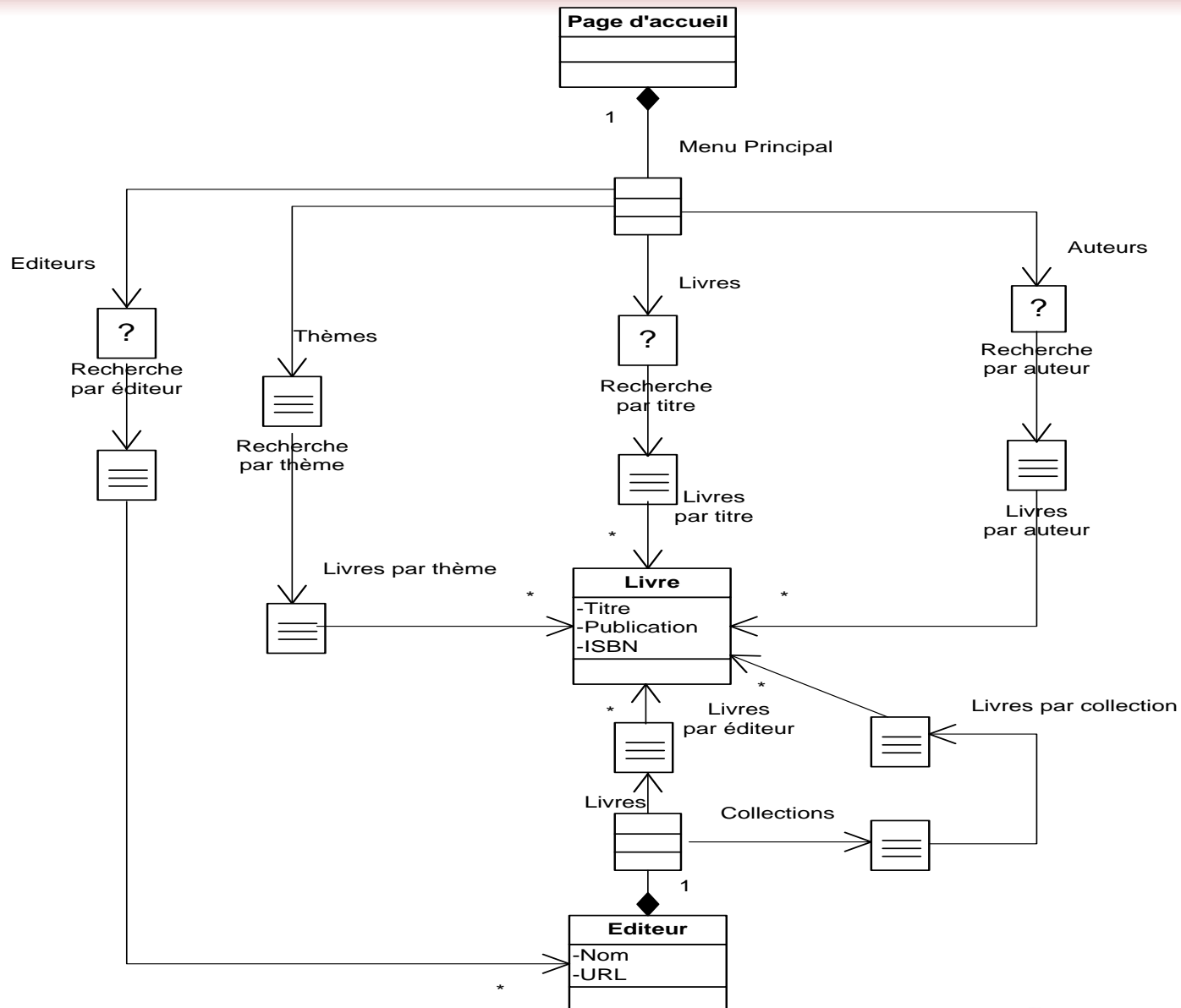


Diagramme d'accès



Exemple 2 : Régie publicitaire

- **Administrateur** : L'administrateur du système d'information
- **Opérateur de régie** : Un opérateur de la société de régie publicitaire qui propose des campagnes et des outils statistiques aux annonceurs
- **Annonces** : Un annonceur propose du contenu pour l'espace publicitaire
- **Internaute** : Un utilisateur occasionnel d'une borne d'accès internet située dans un salon d'aéroport
- **Borne (acteur externe)** : Borne d'accès internet située dans un salon d'aéroport donné

Cas d'utilisation

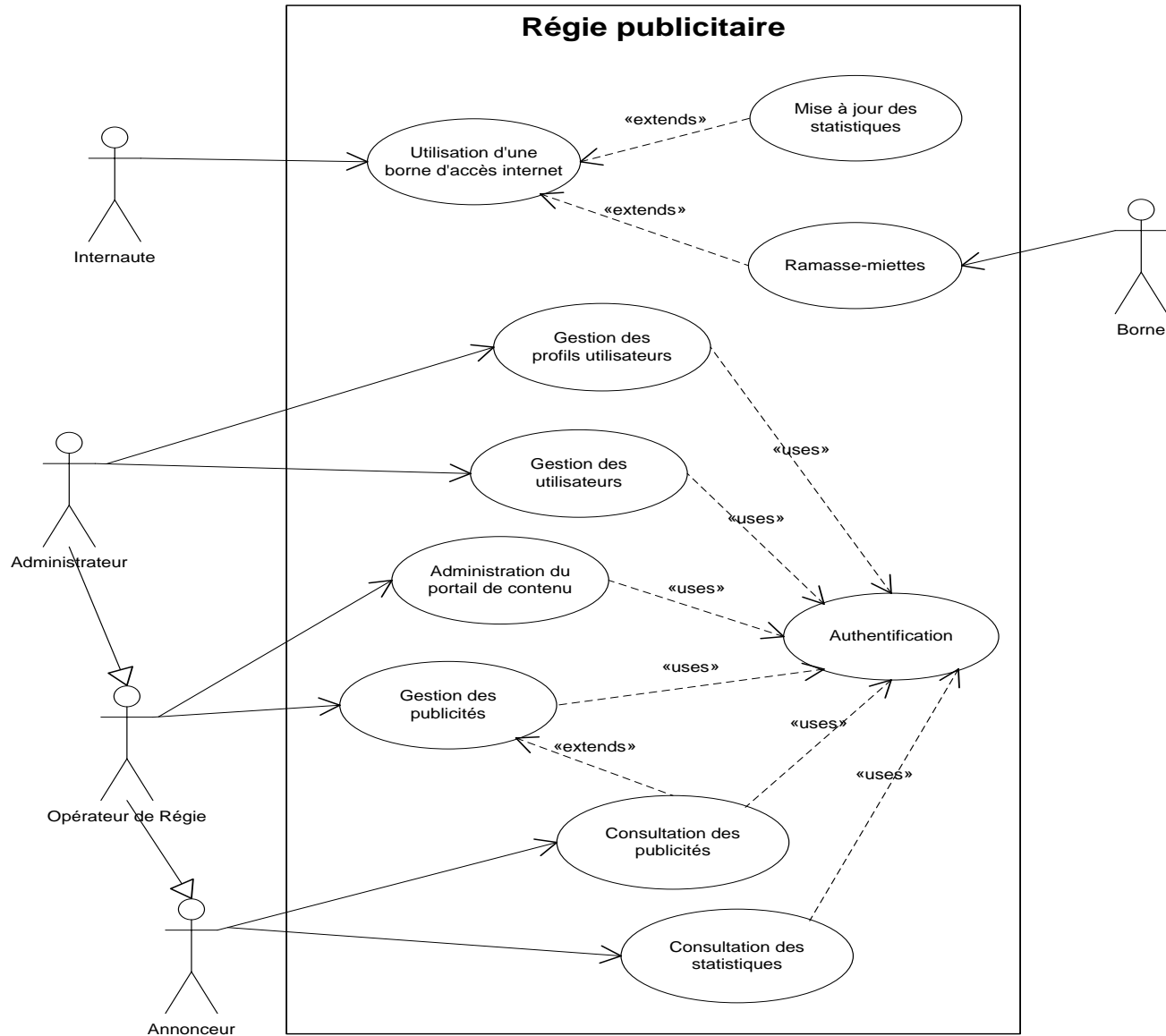


Diagramme de classes

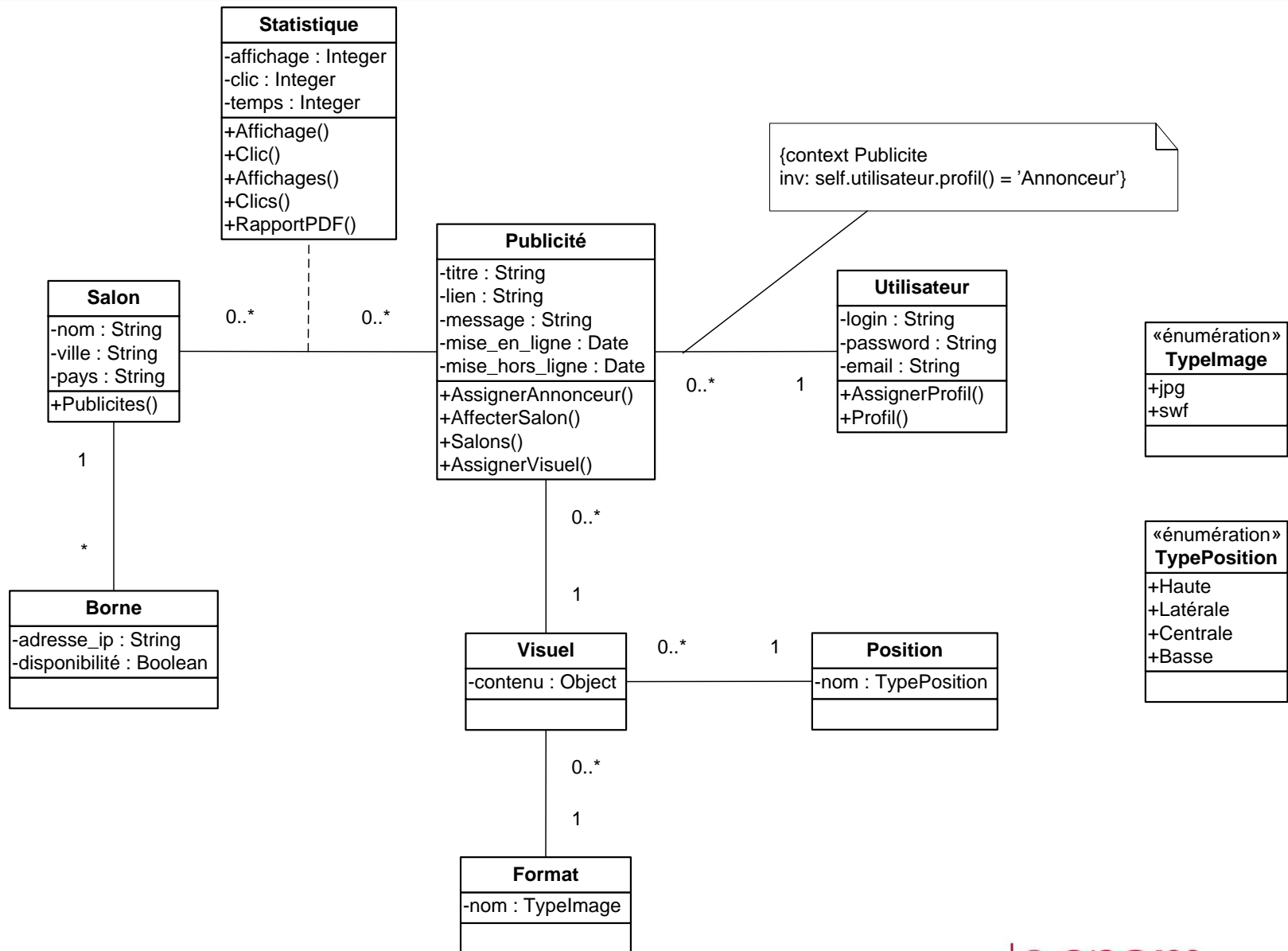


Diagramme de navigation : Rôle Annonceur

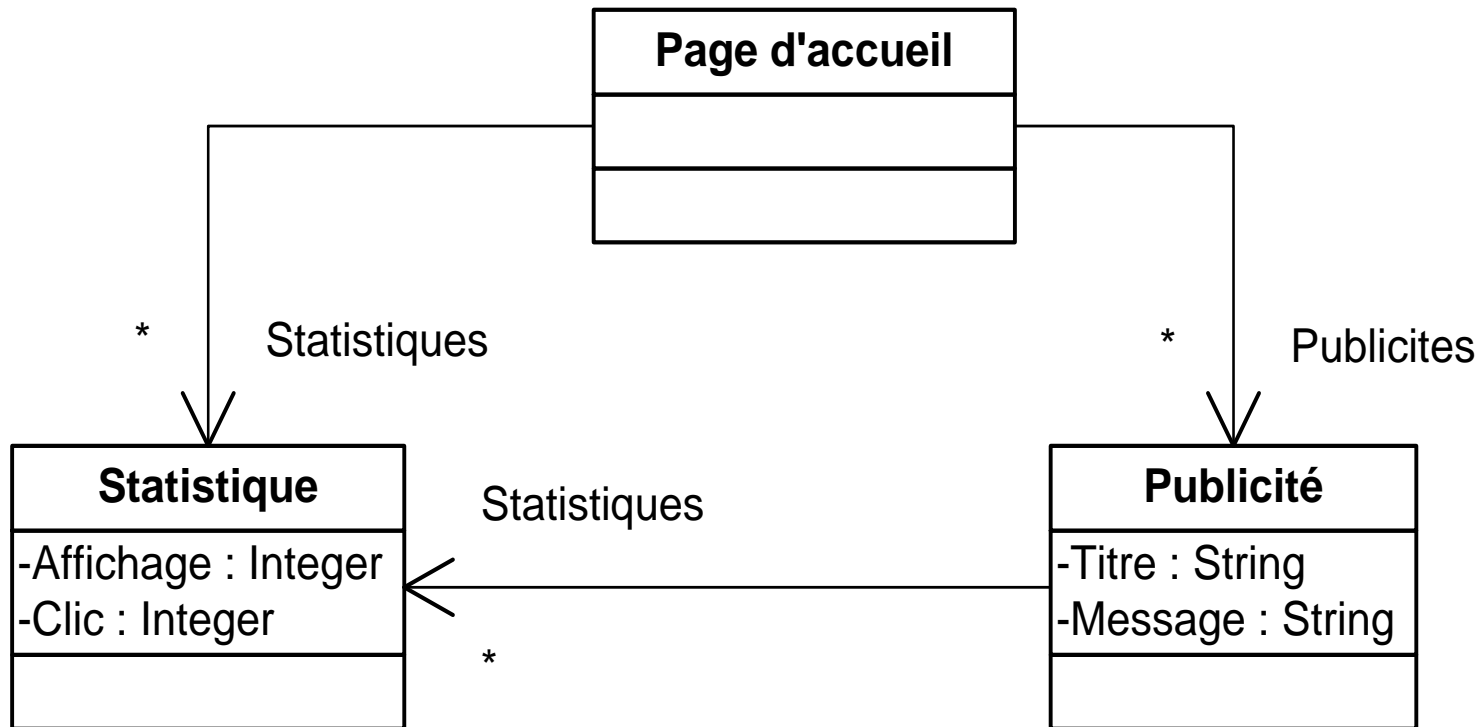


Diagramme d'accès: Rôle Annonceur

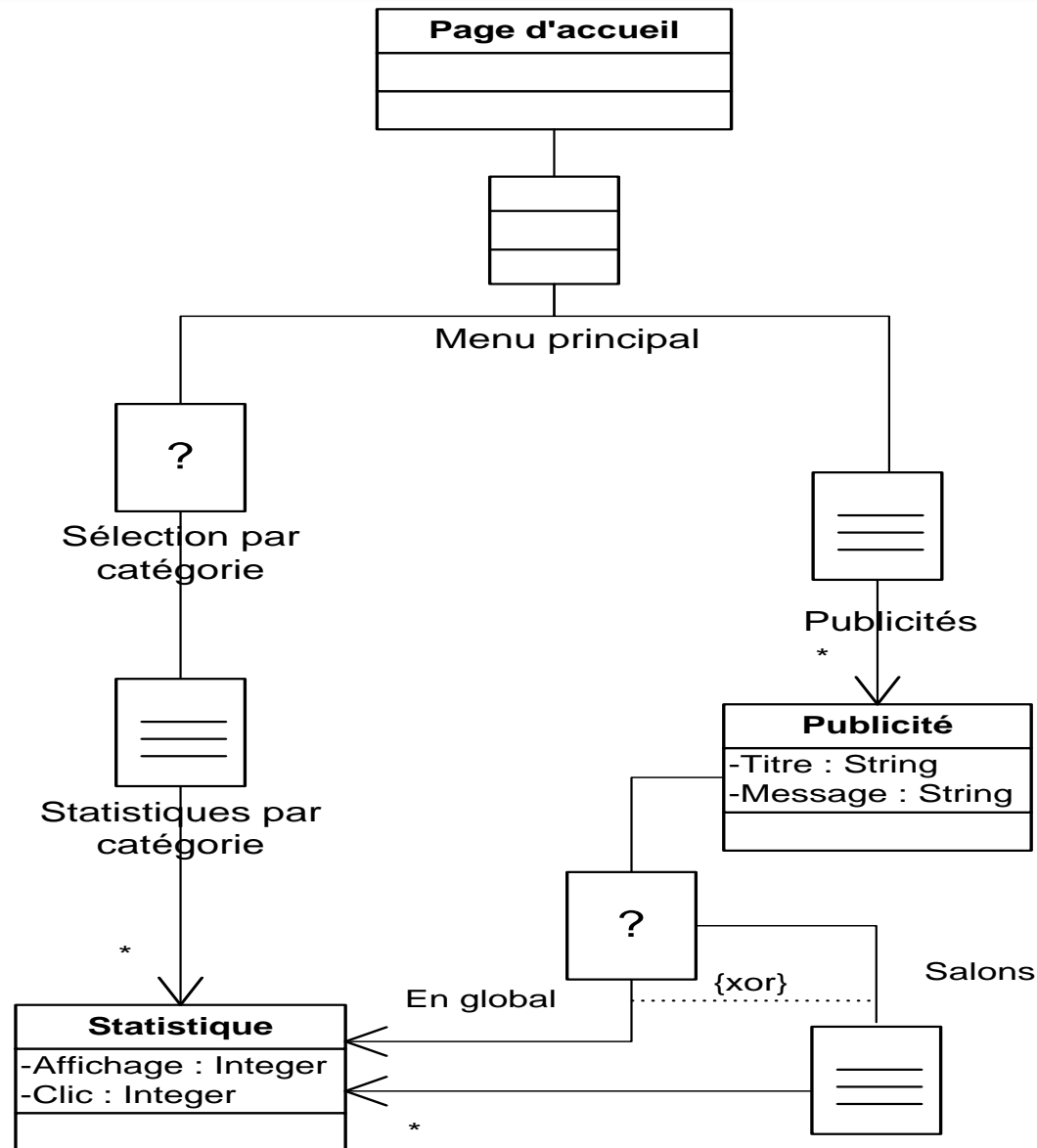


Diagramme de navigation : Rôle Opérateur de Régie

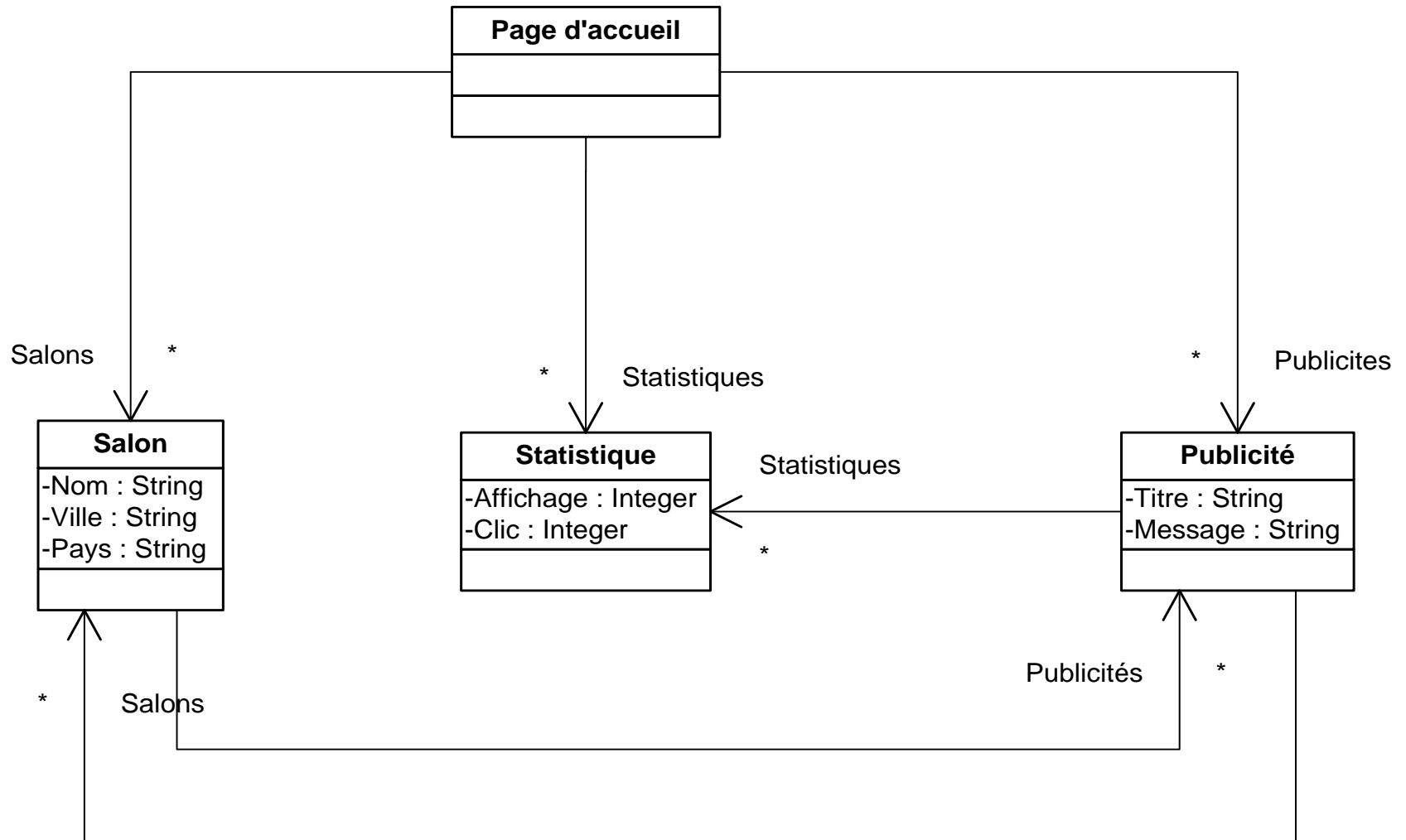
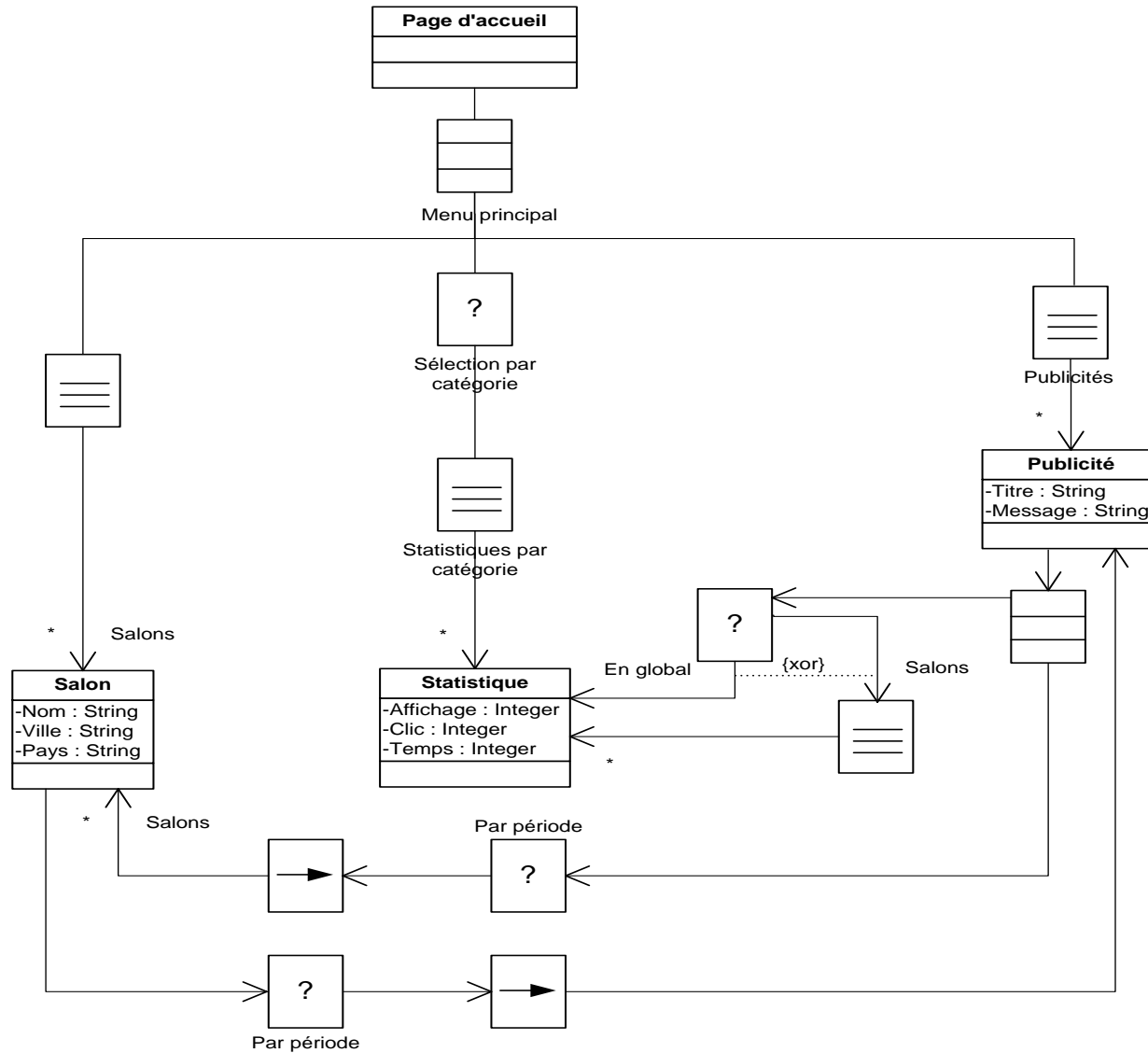


Diagramme d'accès: Rôle Opérateur de Régie



3.1. Introduction à UWE (1)

- UWE : UML-based Web Engineering
- Définie par N. Koch et R. Hennicker
- Modélisation d'applications hypermédia fondées sur le web
- Utilise un profil d'UML
- Reprend la démarche du processus unifié de Jacobson
- Couvre tout le cycle de développement des applications web
- Repose sur un méta-modèle, appelé Munich Reference Model, formalisant les principes de fonctionnement des applications web
- La méthode est accompagnée d'un outil : ArgoUWE

3.1. Introduction à UWE (2)

- Profils et Stéréotypes UML
 - UML peut être adapté pour être un langage spécifique d'un domaine (Domain Specific Language-DSL), appelé UML-Profile
 - Une extension UML est un profil si elle utilise les mécanismes d'extension fournis par UML: **stereotypes, tagged values et contraintes OCL**.
 - Les extensions spécifiques au domaine sont appelés les STEREOTYPES, notées « stereotypes »
 - Les stéréotypes représentent une extension des méta-classes UML
 - Avec des méta-attributs additionnels appelés « tags »
 - Avec des contraintes additionnelles

3.1. Introduction à UWE (3)

- Trois dimensions sont à considérer:
 - Les **phases**: expression des besoins, analyse, conception et mise en œuvre
 - Les **aspects**: structurels et comportementaux
 - Les **niveaux**: contenu, navigation et présentation

3.2 Les étapes de la méthode

- **Modélisation des besoins fonctionnels**
 - Diagramme des cas d'utilisation
 - Diagramme d'activités
- **Modélisation du contenu**
 - Diagramme de classes
 - Diagramme d'états
- **Modélisation de la navigation**
 - Diagramme de la structure de navigation
 - Diagramme d'accès
- **Modélisation de la présentation**
 - Modèle statique
 - Modèle dynamique
- **Modélisation de l'adaptation**

3.3. Illustration: un deuxième exemple

- Système d'évaluation d'articles soumis à une conférence scientifique
- Les acteurs
 - Les utilisateurs du système
 - Auteurs qui soumettent un article
 - Membres du comité de programme qui évaluent les articles
 - Le président du comité de programme
- Les besoins fonctionnels
 - Soumettre un article
 - Affecter un article à des évaluateurs
 - Produire une évaluation
 - Produire la liste des articles acceptés/refusés

3.4. Modélisation des besoins fonctionnels (1)

- Les fonctionnalités de l'application web sont modélisées par les cas d'utilisation UML
- Modélisation fondée sur les vues utilisateurs (acteurs)
- Deux types de besoins pris en compte:
 - Fonctionnels (comme pour les SI classiques) à l'aide de cas d'utilisation standards
 - Navigationnels (spécifiques aux SI web) à l'aide de cas d'utilisation navigationnels
- Modélisation affinée par les diagrammes d'activités UML
 - Donne une représentation visuelle des workflows
 - Modélise la logique fonctionnelle de l'application

3.4. Modélisation des besoins fonctionnels (2)

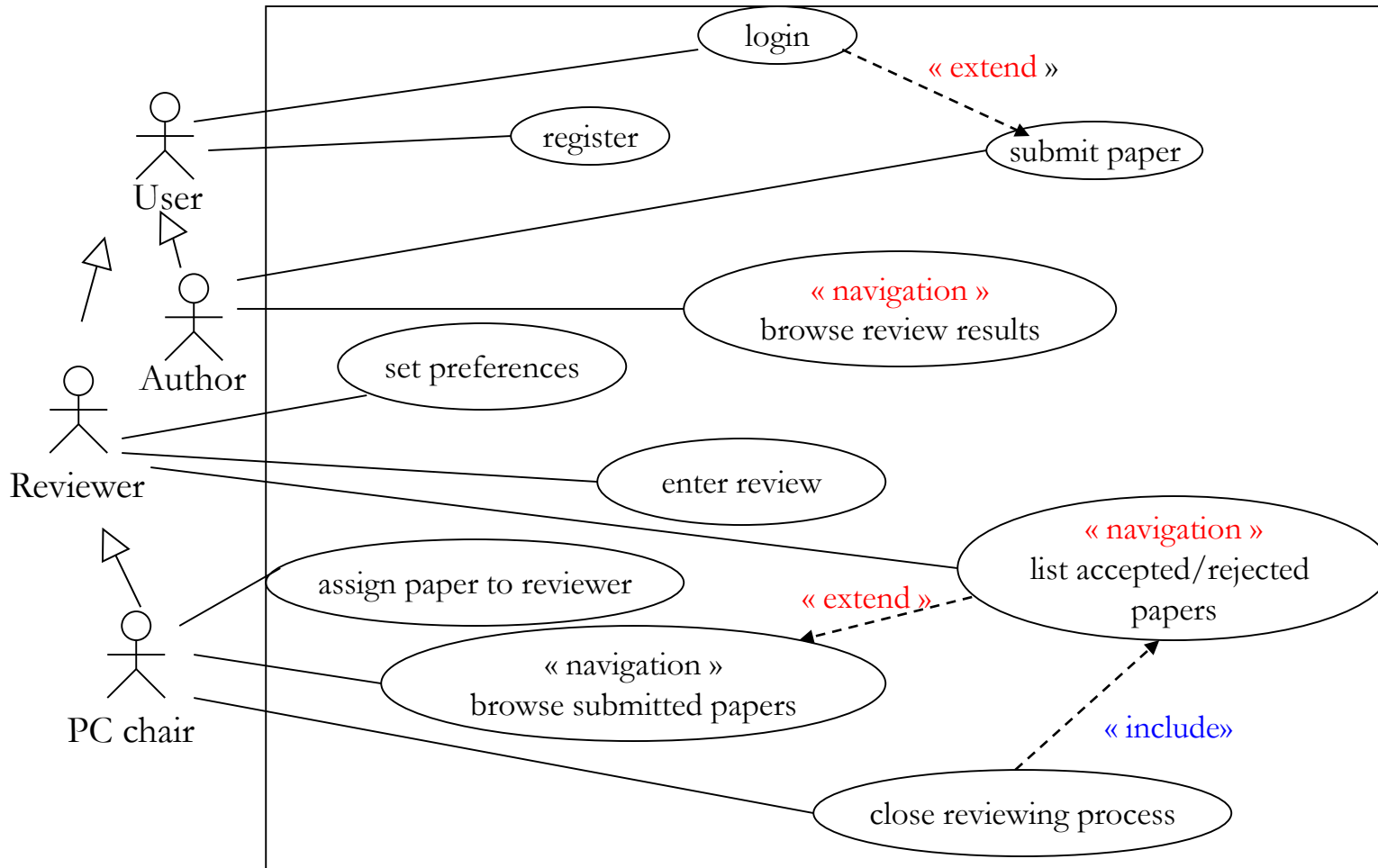
- L'analyse des besoins suit le processus unifié
- Elle identifie les besoins fonctionnels des différents utilisateurs
- Les notions d'acteurs, les relations d'inclusion et d'extension entre cas d'utilisation, les mécanismes de paquetage et de vues conservent la sémantique d'UML
- Les notations graphiques sont celles d'UML2
- Des **relations d'extension** peuvent être définies entre cas d'utilisation
- Des **relations d'héritage** peuvent être définies entre acteurs

3.4. Modélisation des besoins fonctionnels (2) : La pratique

- Elaborer le modèle des cas d'utilisation:
 - Identifier les **acteurs** de l'application
 - Identifier les **cas d'utilisation** définissant les fonctionnalités de l'application
 - Etablir la correspondance entre acteur et cas d'utilisation
 - Définir les relations **includes** et **extends** entre les cas d'utilisation
 - Utiliser **l'héritage** pour les acteurs et les cas d'utilisation
 - Différencier les cas d'utilisation **standards** des cas **navigationnels**
- Construire le diagramme d'activités pour chaque **cas d'utilisation non navigationnels et non trivial**
 - Identifier les **activités**
 - Définir les **transitions** entre activités
 - Affecter les objets relatifs aux activités
- Faire correspondre le diagramme d'activités à chaque cas d'utilisation non navigationnel

3.4. Modélisation des besoins fonctionnels (3)

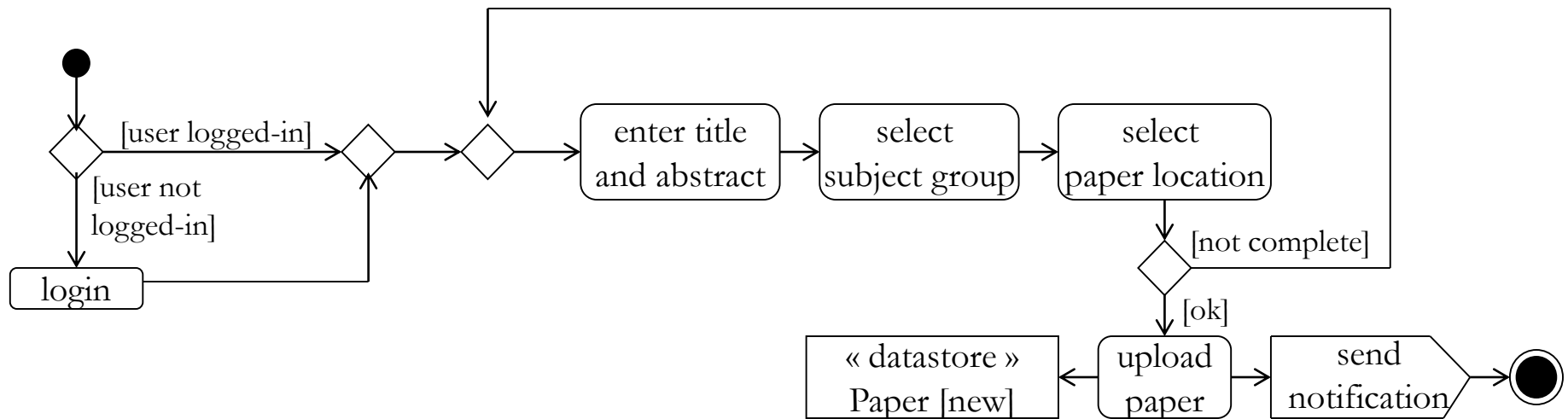
cas d'utilisation



3.4. Modélisation des besoins fonctionnels (3)

Diagramme d'activité

- Exemple : « submit paper » pour l'acteur « Author »



3.5. Modélisation des contenus (1)

- Aspect structurel
 - Modélisation du domaine
 - Diagramme de classes UML
- Aspect comportemental
 - Visualiser les messages entre objets de façon à montrer comment ils collaborent
 - Capturer l'état des objets et montrer les actions menant aux transitions d'états
 - Diagramme d'états/diagramme d'interactions
- Attention!!!
 - On ne présente que le contenu d'un niveau
 - Pas d'hypertexte ou de modélisation de la présentation

3.5. Modélisation des contenus (2) : diagramme de classes

- Trouver les **classes** représentant les unités informationnelles et/ou les éléments multimédia
- Trouver les **associations** et les agrégations représentant les relations entre classes
- Trouver les **héritages** indiquant les hiérarchies entre classes
- Les classes et les associations sont décrites par les attributs et les opérations (méthodes)
- Les classes et les associations peuvent être organisées en paquetages UML

3.5. Modélisation des contenus (3) : diagramme de classes

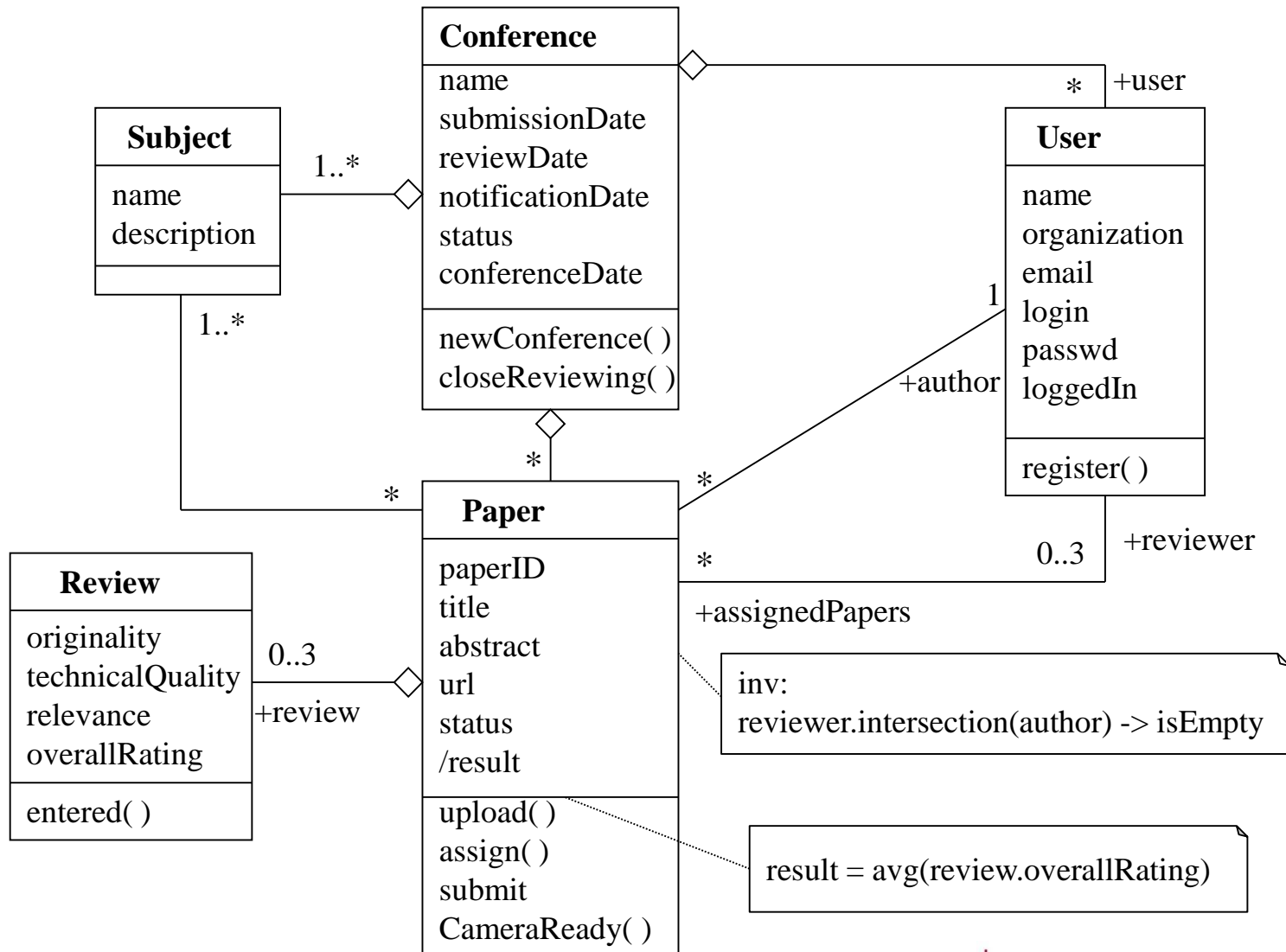
- On représente la sémantique de l'application hypermédia en considérant aussi les aspects navigationnels et présentationnels
⇒ les classes sont équipées d'un compartiment optionnel qui s'ajoute à celui des attributs et des opérations
- Ce compartiment peut recevoir des informations adaptant le contenu de la classe au profil utilisateur
- Ce modèle sert à créer des diagrammes supplémentaires décrivant notamment la navigation

La Pratique

- Construire un diagramme de classe
 - Trouver les **classes** correspondant aux objets identifiés dans les diagrammes d'activités
 - Identifier les **attributs et les méthodes** de ces classes
 - Déterminer les **relations** entre classes (associations, agrégations, compositions)
 - Définir les **hiérarchies d'héritage** si nécessaire
- Identifier les classes conceptuelles pertinentes pour le modèle de navigation
- Générer les classes fondées sur les objets des diagrammes d'activités

3.5. Modélisation des contenus (3) : diagramme de classes

Exemple : système de reviewing

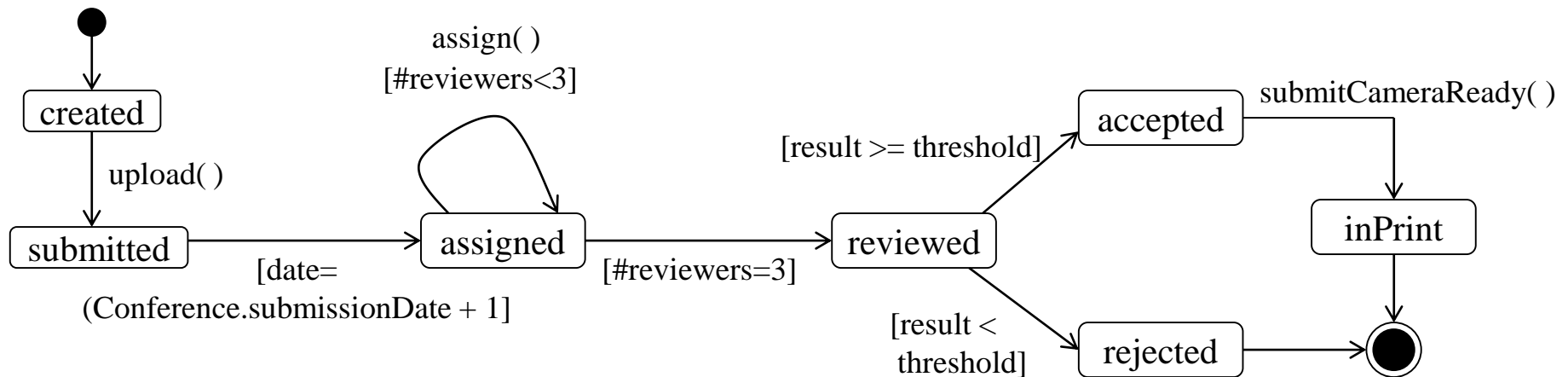


3.5. Modélisation des contenus (5) : diagramme d'états

- Modélisation UML
- A réaliser pour le cycle de vie de l'objet considéré
- Vérifier la cohérence avec le diagramme de classes

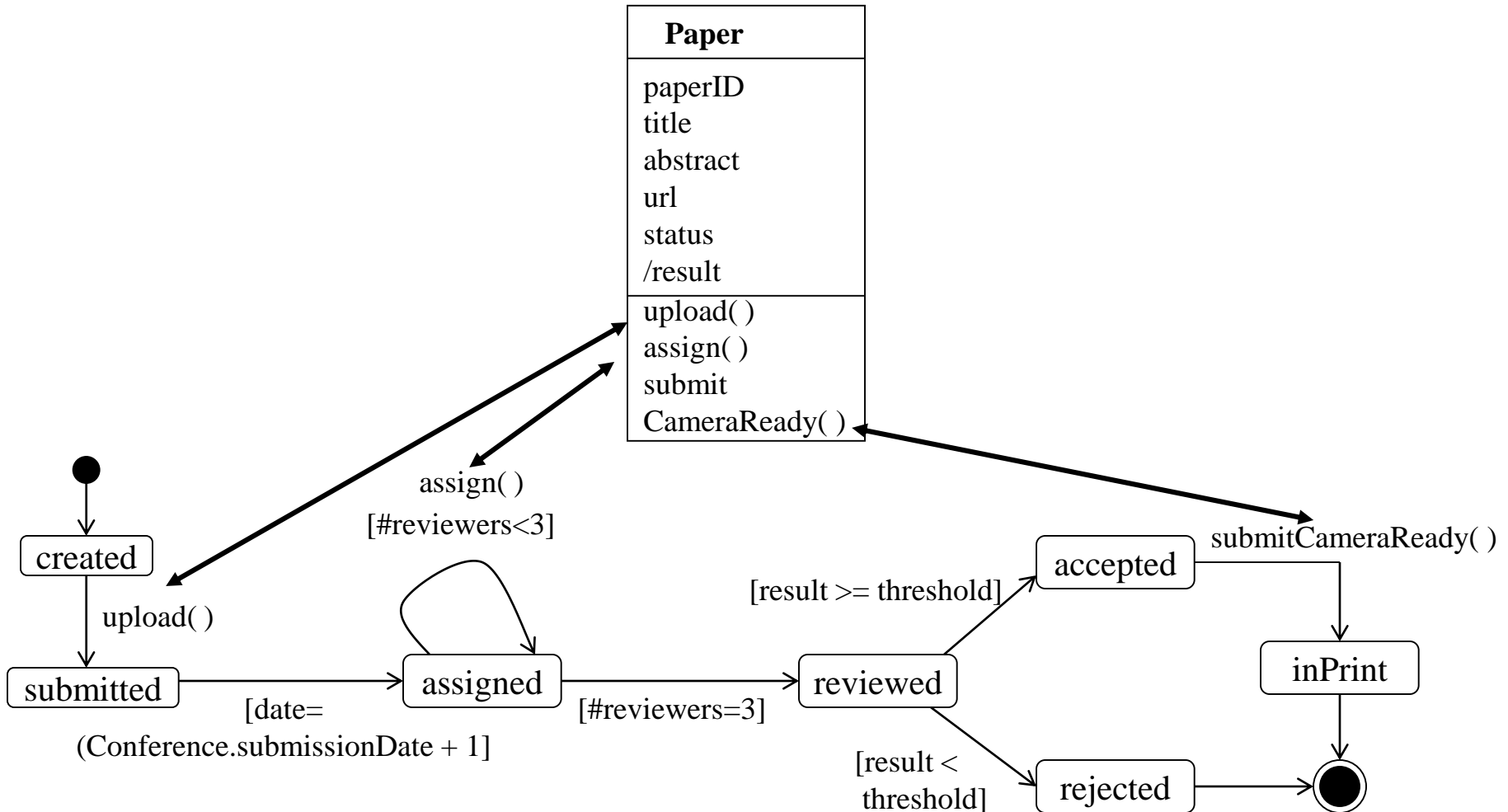
3.5. Modélisation des contenus (6) : diagramme d'états

- Exemple : cycle de vie d'un papier (pour la classe « papier »)



3.5. Modélisation des contenus (7) : diagramme d'états

- cohérence avec le diagramme de classe



3.6. Modèles de navigation (1)

- Deux modèles :
 - Diagramme de la structure de navigation qui définit le QUOI (les objets visités)
 - Diagramme d'accès qui définit le COMMENT (ils sont visités)
- La modélisation est organisée en NŒUDS et LIENS
 - Les nœuds sont les objets
 - Les liens relient les objets

3.6. Modélisation de la navigation (2)

- Objectifs
 - Modélisation des nœuds et des liens de la structure de navigation
 - Un nœud est appelé page ou document
 - Modélisation des parcours de navigation
- Résultats
 - **Modèle de la structure de navigation :**
 - quelles classes du modèle de contenu peuvent être visitées par navigation?
 - **Modèle d'accès :**
 - Affiner le modèle de la structure de navigation en y ajoutant les éléments relatifs aux accès
 - **Pour chaque rôle utilisateur** (acteur et ses use cases), **développer la structure de navigation et le modèle d'accès**

3.6. Modélisation de la navigation (3)

- 3.6.1. Structure de navigation - deux stéréotypes:
 - « classes de navigation » pour les noeuds
 - « liens de navigation » pour les liens
- 3.6.2. Structure d'accès - quatre primitives:
 - « « menu » » accès aux nœuds de classes différentes
 - « « index » » accès aux nœuds individuels d'une classe
 - « « guidedTour » » accès séquentiel à une liste de nœuds
 - « « query » » recherche d'un nœud et accès direct

3.6.1. Diagramme de la structure de navigation (1)

- Indique les classes du diagramme de classes qui peuvent être visitées lors de la navigation
- Composé d'un ensemble de classes et d'associations de navigation obtenues à partir du diagramme de classes
- Une classe de navigation est définie comme une classe stéréotypée « classe de navigation » avec le même nom que la classe correspondante du diagramme de classes
- Elle modélise une classe dont les instances (appelées objets de navigation) sont visitées par l'utilisateur
- Les objets de navigation sont reliés en termes UML par des liens qui représentent les instances des associations du modèle de classe de navigation

3.6.1. Diagramme de la structure de navigation (2)

- Le modèle de classe de navigation peut être vu comme un sous-graphe du diagramme de classes
- Les classes non nécessaires à la navigation sont éliminées, ou réduites à des attributs d'autres classes. On utilise la notion d'attribut dérivé noté **/nomattribut**. Les valeurs de ces attributs sont calculables à partir d'une expression OCL.

3.6.1. Diagramme de la structure de navigation (3)

- Une association de navigation exprime une possibilité d'accès direct à une classe de navigation « cible » à partir d'une classe de navigation « source ».
- Les associations du diagramme de classes sont transformées en associations de navigation (il faut alors que les classes du diagramme de classes aient une correspondance dans le modèle de la structure de navigation)

3.6.1. Diagramme de la structure de navigation (4)

- Des associations de navigation peuvent être créées pour offrir un accès direct à certaines informations. Il faut alors:
 - Préciser la sémantique associée
 - Préciser la façon d'obtenir la ou les classes de navigation cibles associées

3.6.1. Diagramme de la structure de navigation (5) : Méthode

- Les classes et les associations de navigation sont représentées graphiquement dans un diagramme de classes UML appelé « diagramme de classes de navigation »
- La navigabilité est indiquée pour une association par une flèche située à l'extrémité de la ligne de cette association

3.6.1. Diagramme de la structure de navigation (6) : Méthode

- Si une flèche est localisée aux deux extrémités de l'association, l'utilisateur peut se diriger dans les deux directions
- Chaque lien comporte une « source » et une « cible » navigationnelles
- Chaque extrémité de l'association navigationnelle est renseignée à l'aide d'une multiplicité explicite et éventuellement d'un nom de rôle

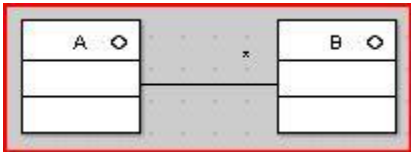
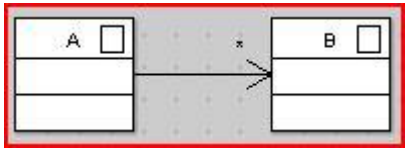
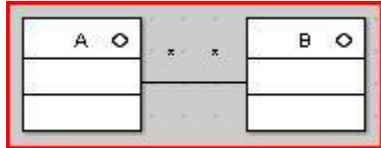
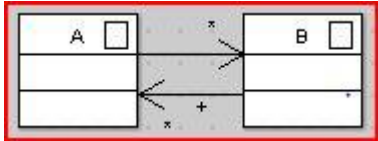
3.6.1. Diagramme de la structure de navigation (7) : Méthode

- Si aucun nom de rôle ne peut être donné, alors on utilise la règle suivante :
 - Si la multiplicité de la classe cible est inférieure ou égale à 1, alors le nom de rôle donné est celui de la classe
 - Si la multiplicité de la classe cible est supérieure à 1, alors le nom de rôle donné est celui de la classe mais avec la marque du pluriel

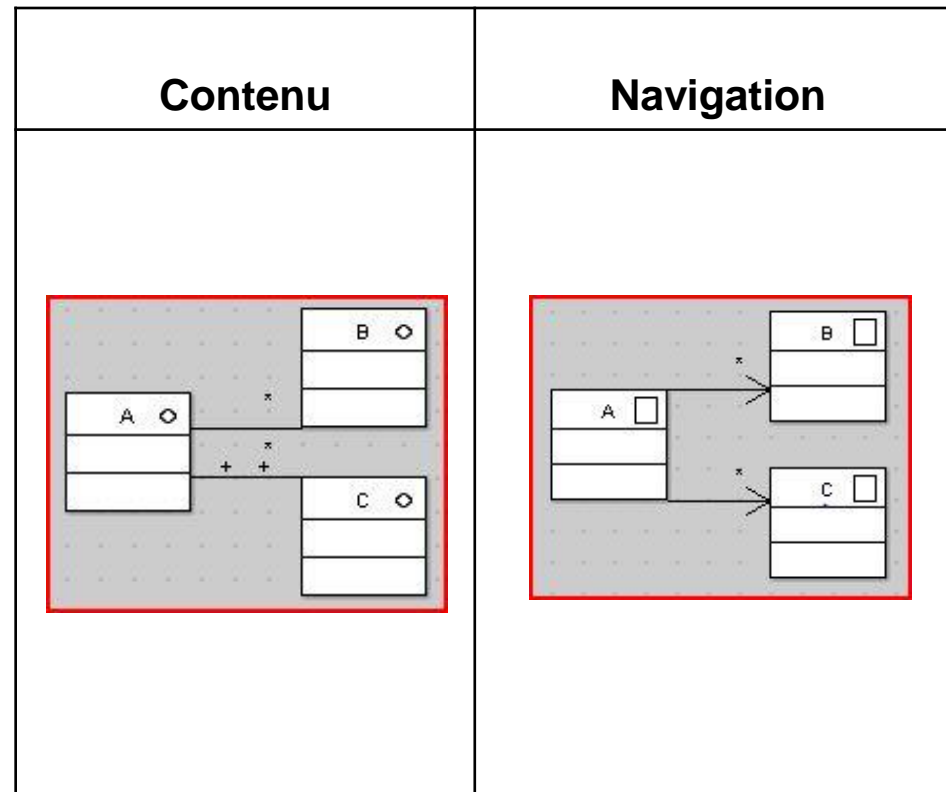
3.6.1. Diagramme de la structure de navigation (8) : Pratique

- Règles de détermination du modèle de navigation
 - Définir une **classe de navigation** pour chaque classe du diagramme de classes pertinente
 - Ajouter les **attributs des classe omises** (dans le diagramme de classes) aux classes de navigation
 - Définir des **liens de navigation** pour les association du diagramme de classes
 - Définir des **liens de navigation** pour les agrégations et compositions du diagramme de classes
 - Ajouter les **attributs des classes** (du diagramme de classes) aux classes de navigation
 - Transférer les **multiplicités et les noms de rôle des associations, agrégations et compositions** du diagramme de classes vers les éléments correspondant du modèle navigationnel
 - Les flèches de navigation sont orientées vers les cardinalités supérieures. En cas de cardinalités multiples, les flèches de navigation sont dans les deux sens
 - Ajouter des liens de navigation supplémentaires du fait de scénarii relatifs aux cas d'utilisation
 - Ajouter des liens de navigation supplémentaires comme raccourcis (Shortcuts) afin de réduire la longueur du chemin de navigation (généralement de longueur 1)

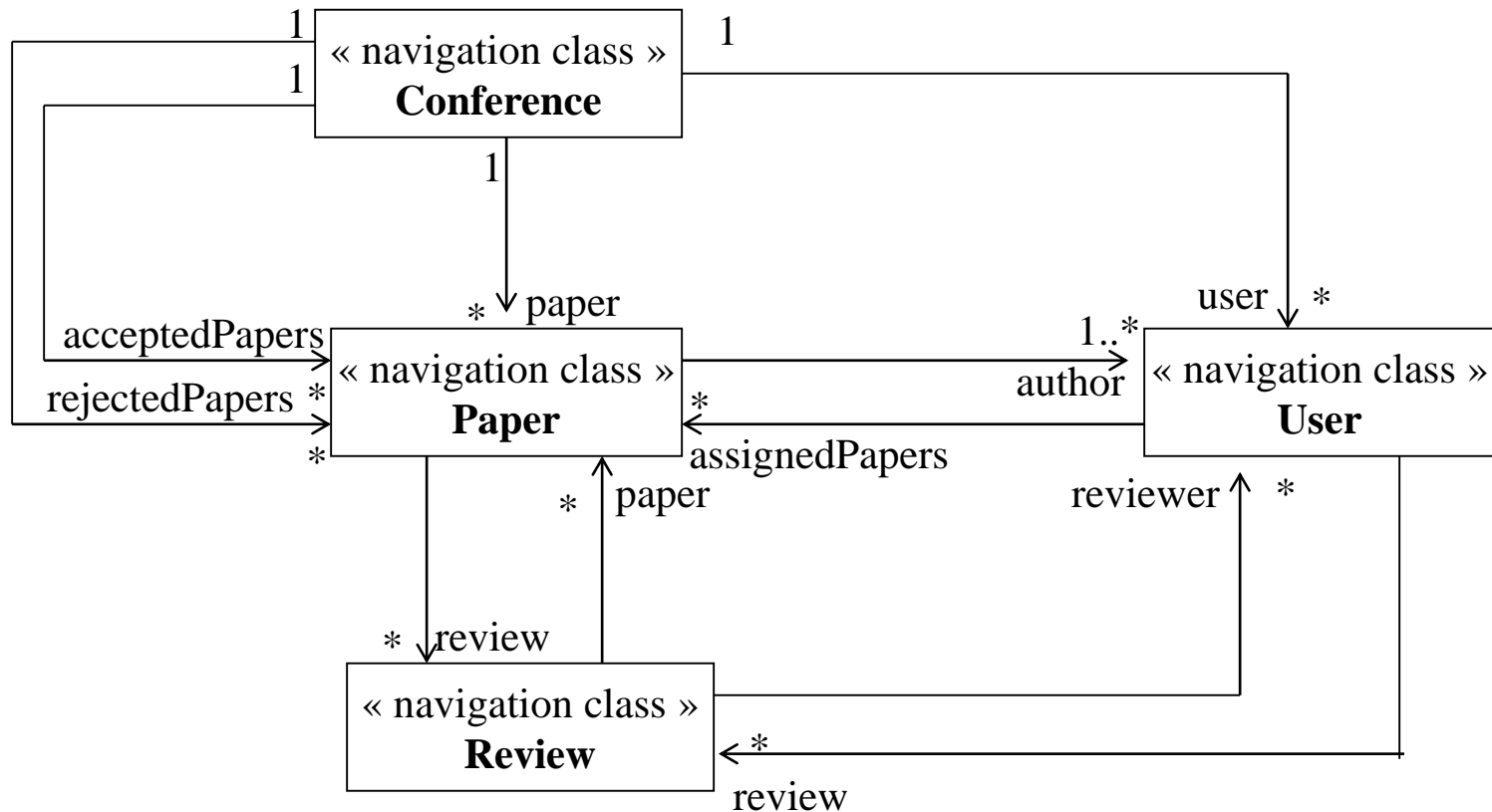
3.6.1. Règles de génération du modèle de navigation (9)

Contenu	Navigation
« classe »	« classe de navigation », nœud
« liens »	« lien de navigation »
<p>lien 1---1 *</p> 	
<p>lien *---*</p> 	

3.6.1. Règles de génération du modèle de navigation (10)



3.6.1. Diagramme de la structure de la navigation (11)



3.6.2. Le diagramme d'accès (1)

- A définir à partir du diagramme de la structure de navigation
- Détermine comment les objets de navigation sont visités
- Des éléments additionnels sont exigés pour exécuter la navigation: ce sont des primitives d'accès
- Primitives d'accès: menus, index, visites guidées, nœuds externes
- Il faut définir aussi les contextes de navigation

3.6.2. Le diagramme d'accès (2) : les primitives d'accès

- Un **menu** est un objet composite qui contient un nombre fixe d'items. Chacun d'entre eux a un nom et possède un lien vers une instance de classe de navigation ou un élément d'accès

3.6.2. Le diagramme d'accès (3) : les primitives d'accès

- Un **GuidedTour** (visite guidée) donne accès au premier objet d'un contexte de navigation.
- Le déplacement d'un objet à l'autre est dirigé séquentiellement.
- La visite guidée peut être commandée par l'utilisateur ou par le système.

3.6.2. Le diagramme d'accès (4) : les primitives d'accès

- Un **index** est un objet composite qui contient un nombre arbitraire d'articles d'index.
- Chaque article d'index est un objet qui possède un lien vers une instance de classe de navigation
- Il est membre d'une classe d'index qui en définit la structure
- Stéréotype UML: « index »

3.6.2. Le diagramme d'accès (5) : les primitives d'accès

- Une **Query** est un objet qui a un attribut composé d'une chaîne de caractères correspondant à une requête.
- La Query peut être spécifiée sous forme OCL.

3.6.2. Le diagramme d'accès (6) : une autre primitive d'accès

- Un **nœud externe** est un nœud de navigation appartenant à une autre application hypermédia
- Permet de se déplacer vers une autre application à travers le web via une adresse URL

3.6.2. Le diagramme d'accès (7)

Règles de transformation pour permettre le passage du modèle de navigation à celui du diagramme d'accès:

- Remplacer les associations de navigation bi-directionnelles dont la multiplicité à chacune des deux extrémités est supérieure à 1 par deux associations de navigation uni-directionnelles
- Remplacer les associations de navigation bi-directionnelles dont la multiplicité à l'une des deux extrémités est supérieure à 1 par une association de navigation uni-directionnelle orientée vers l'extrémité de multiplicité supérieure à 1

3.6.2. Le diagramme d'accès (8)


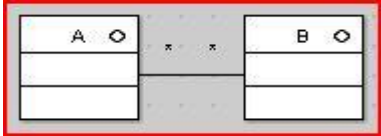
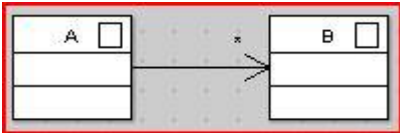
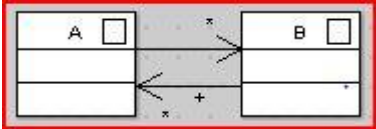

Règles de transformation pour permettre le passage du modèle de navigation à celui du diagramme d'accès (suite)

Pour les associations unidirectionnelles du modèle de structure de navigation ayant une multiplicité supérieure à 1 du côté de l'extrémité qui porte la flèche, choisir une ou plusieurs primitives d'accès pour concrétiser la navigation. Les primitives d'accès sont insérées entre la classe de navigation source et celle qui est cible. Le nom de rôle migre vers le niveau de l'élément représentant la primitive

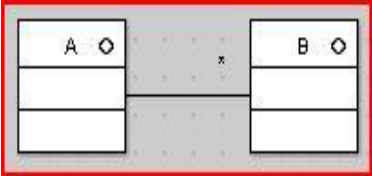
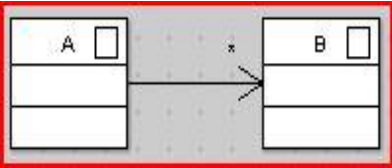

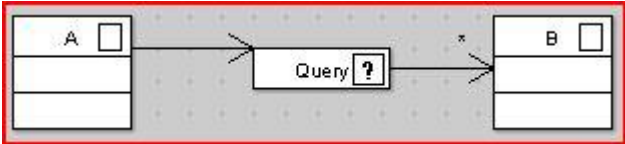
3.6.2. Diagramme d'accès(9) : Démarche de construction

1. Ajouter un **index** à tous les liens de navigation ayant une multiplicité supérieure à 1 à l'extrémité de l'association orientée
2. Remplacer index par **GuidedTour** (optionnel)
3. Ajouter un **Query** pour choisir un sous ensemble des objets de navigation (optionnel)
4. Ajouter un **lien de navigation supplémentaire** à partir d'une primitive d'accès pour cibler une classe de navigation
5. Déplacer les **noms de rôle** des classe de navigation vers la primitive d'accès
6. Mettre les **multiplicités à 1**
7. Introduire un **menu** pour chaque classe ayant plus d'un lien de navigation sortant
8. Utiliser les **noms de rôle** des liens de navigation sortants comme des **items de menu**
9. Ajouter des contraintes pour les invariants et les pre/post conditions
10. A noter que les étapes 1, 5, 6 sont automatisables

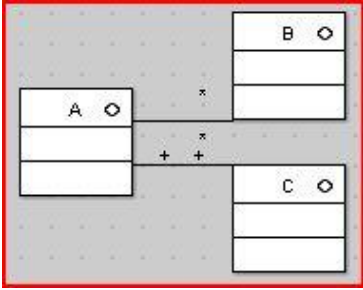
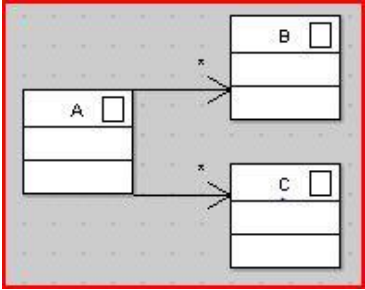
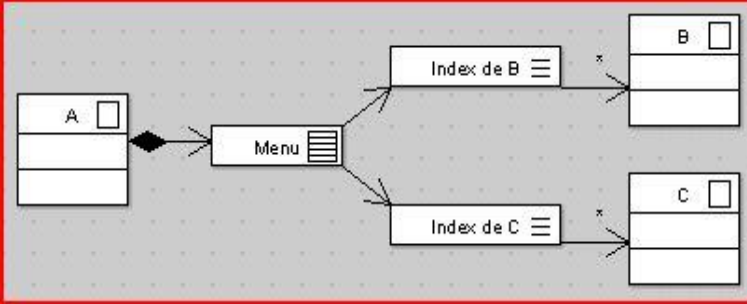
3.6.2. Règles de génération du modèle d'accès (10)

Contenu	Navigation	Accès
« classe »	« classe de navigation », nœud	nœud
« liens »	« lien de navigation »	index, menu, query, guidedTour
<p>lien 1---1 *</p>  <p>lien *----*</p> 	 	<p>« index » accès aux nœuds individuels d'une classe</p> 

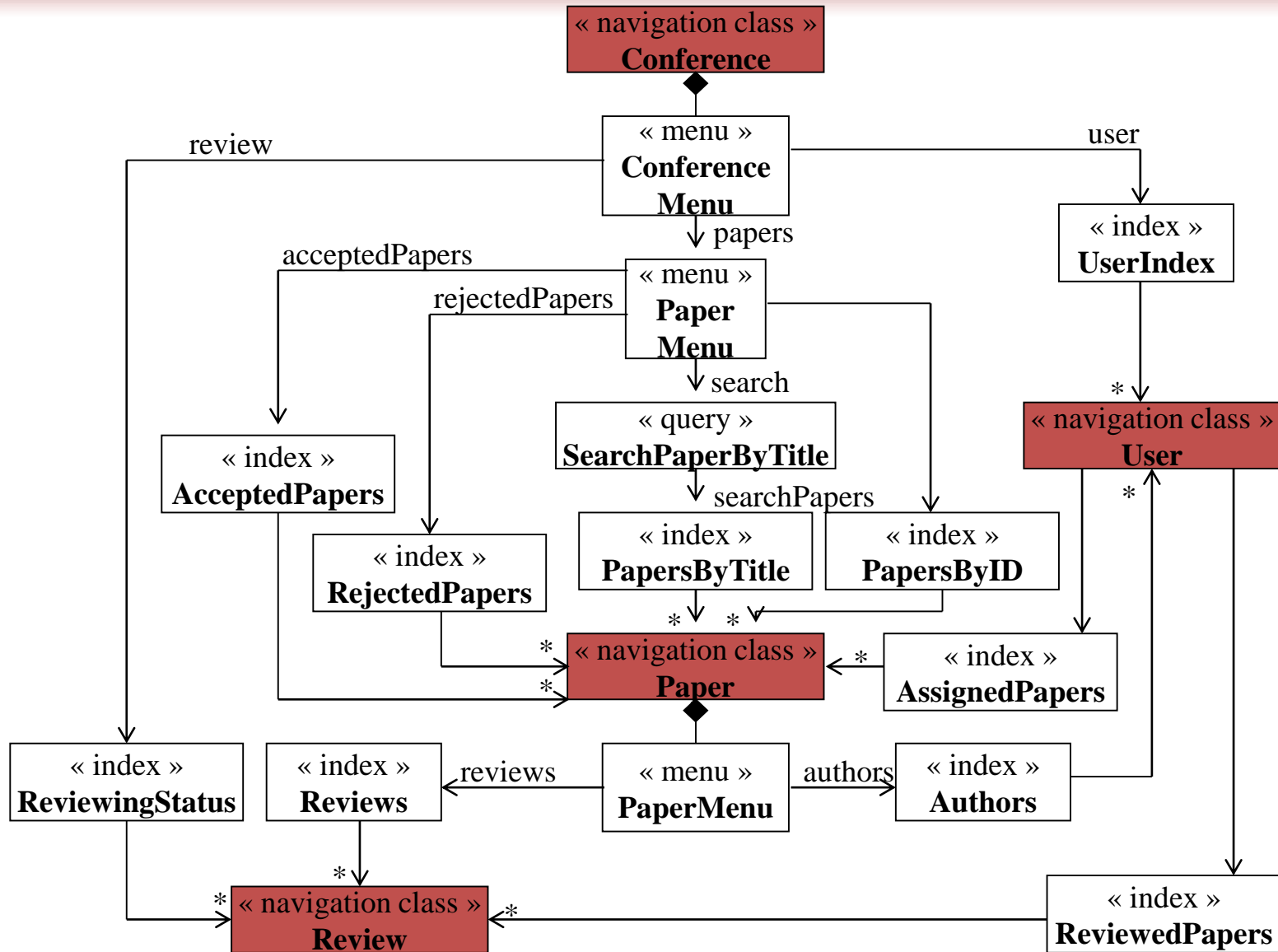
3.6.2. Règles de génération du modèle d'accès (11)

Contenu	Navigation	Accès
		<p>« guidedTour » accès séquentiel à une liste de nœuds</p>  <p>« query » recherche d'un nœud et accès direct</p> 

3.6.2. Règles de génération du modèle d'accès (12)

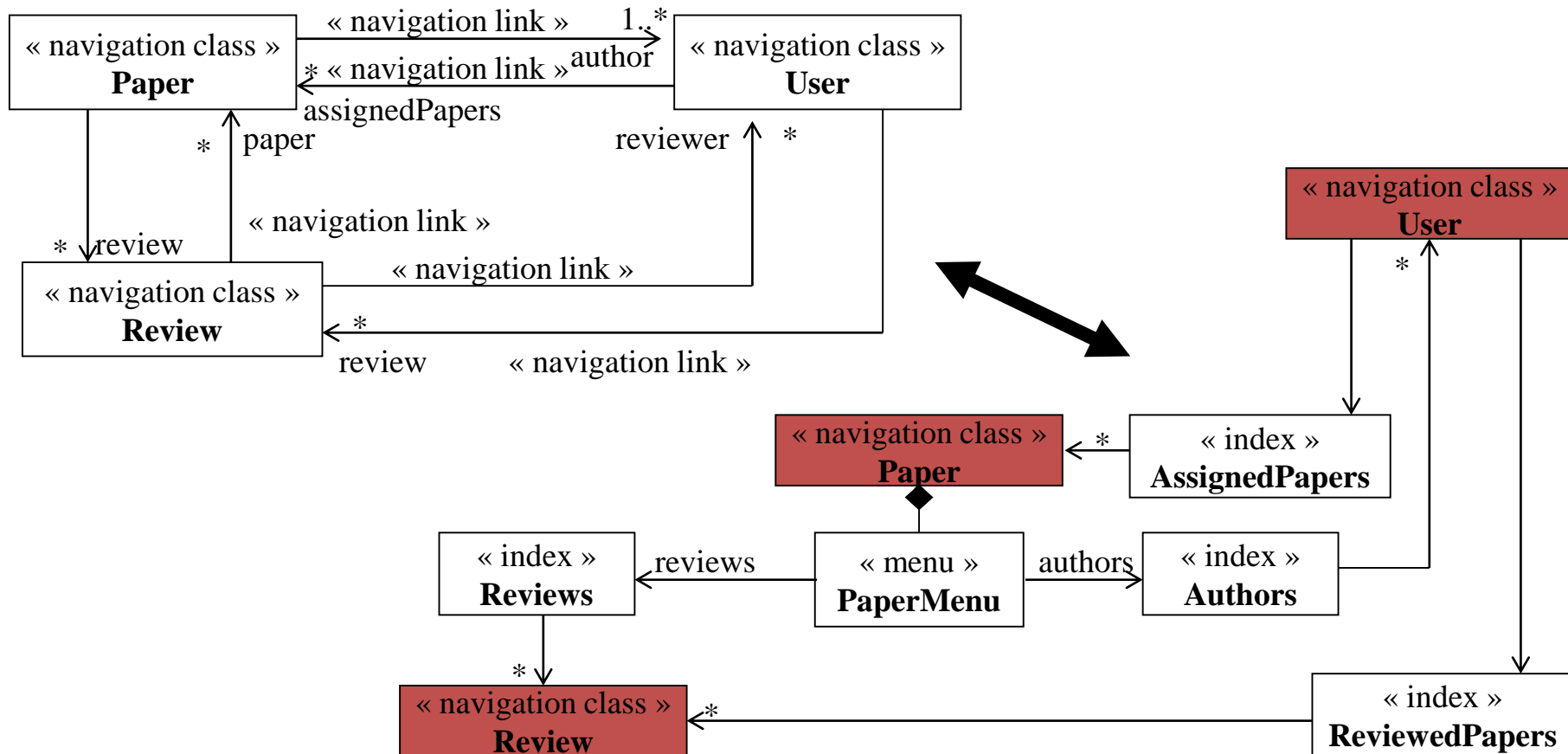
Contenu	Navigation	Accès
		<p>« menu » accès aux nœuds de classes différentes</p> 

3.6.2. Exemple



3.6.2. modélisation de la navigation (14)

- Cohérence des diagrammes de navigation et d'accès



3.6.2. Le diagramme d'accès (15) : le contexte de navigation

- Se compose d'une séquence de nœuds de navigation sur lesquels il impose une méthode de parcours
- Dans un même contexte, des liens relient chaque nœud de navigation au précédent et au suivant
- Un contexte de navigation est dépeint comme un objet avec le stéréotype «contexte de navigation »
- Il est associé à une contrainte OCL qui définit sa séquence de nœuds de navigation
- Les changements de contexte sont possibles

3.6.2. Le diagramme d'accès (18) : le contexte de navigation

- Les types de contexte sont :
 - Contexte de navigation simple
 - Ex: tout le projet à partir des projets par noms
 - Contexte de navigation groupé : séquence extraite des séquences de nœuds de navigation
 - Ex: des projets par département
 - Contexte de navigation filtré : choix dynamique d'une collection d'éléments tirés d'un contexte

3.6.2. Le diagramme d'accès (19) : le contexte de navigation

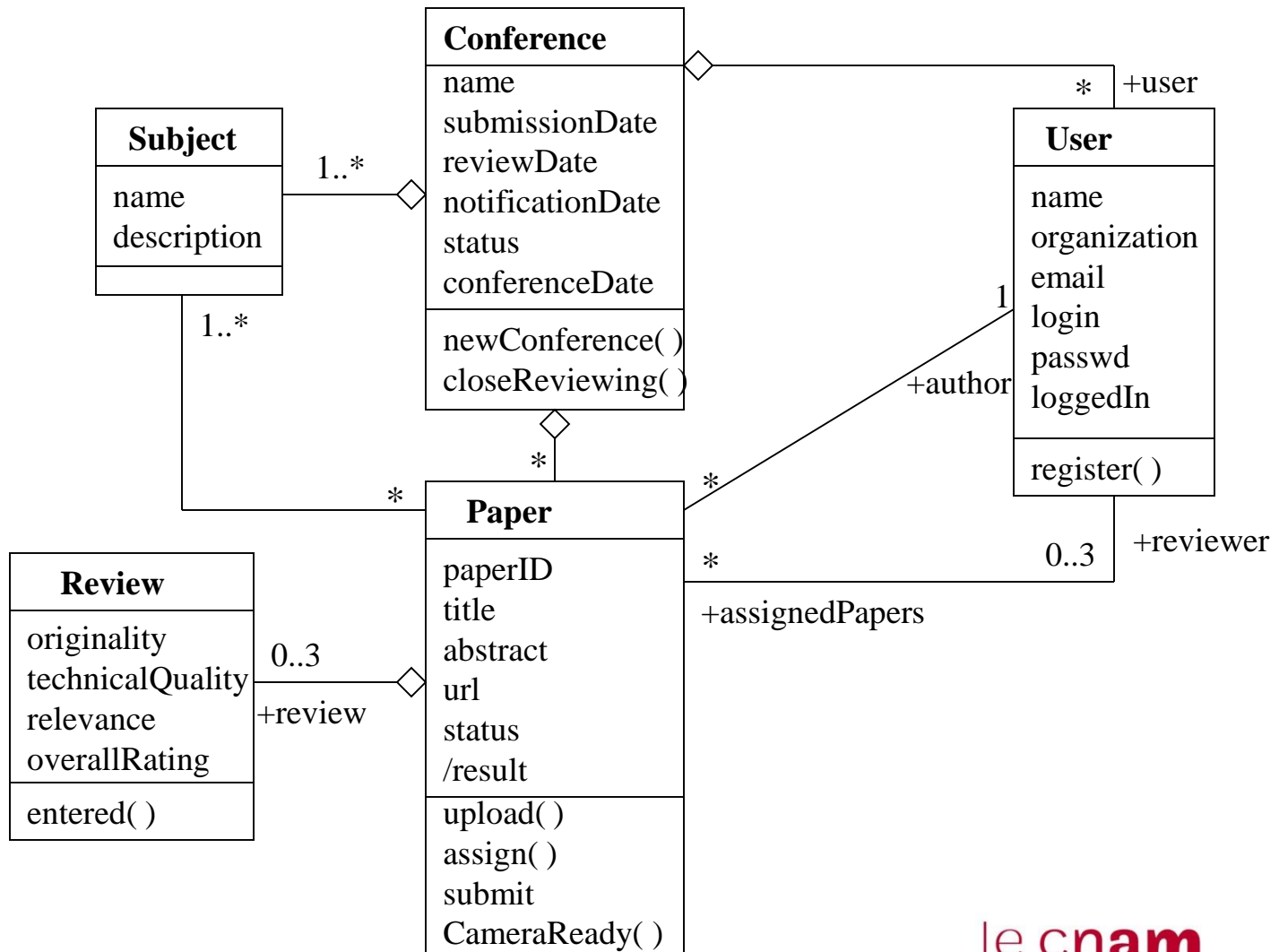
- La notation pour les différents contextes de navigation n'est pas imposée.
- Elle peut être adaptée à l'environnement

3.6.2. Le diagramme d'accès (20) : le contexte de navigation

- Les contextes d'une même classe de navigation peuvent être groupés dans un paquet UML appelé « contexte »
- Ils peuvent être reliés par des associations spéciales qui indiquent les changements de contexte de navigation possibles.
- Une association stéréotypée est définie et est appelée « changement »

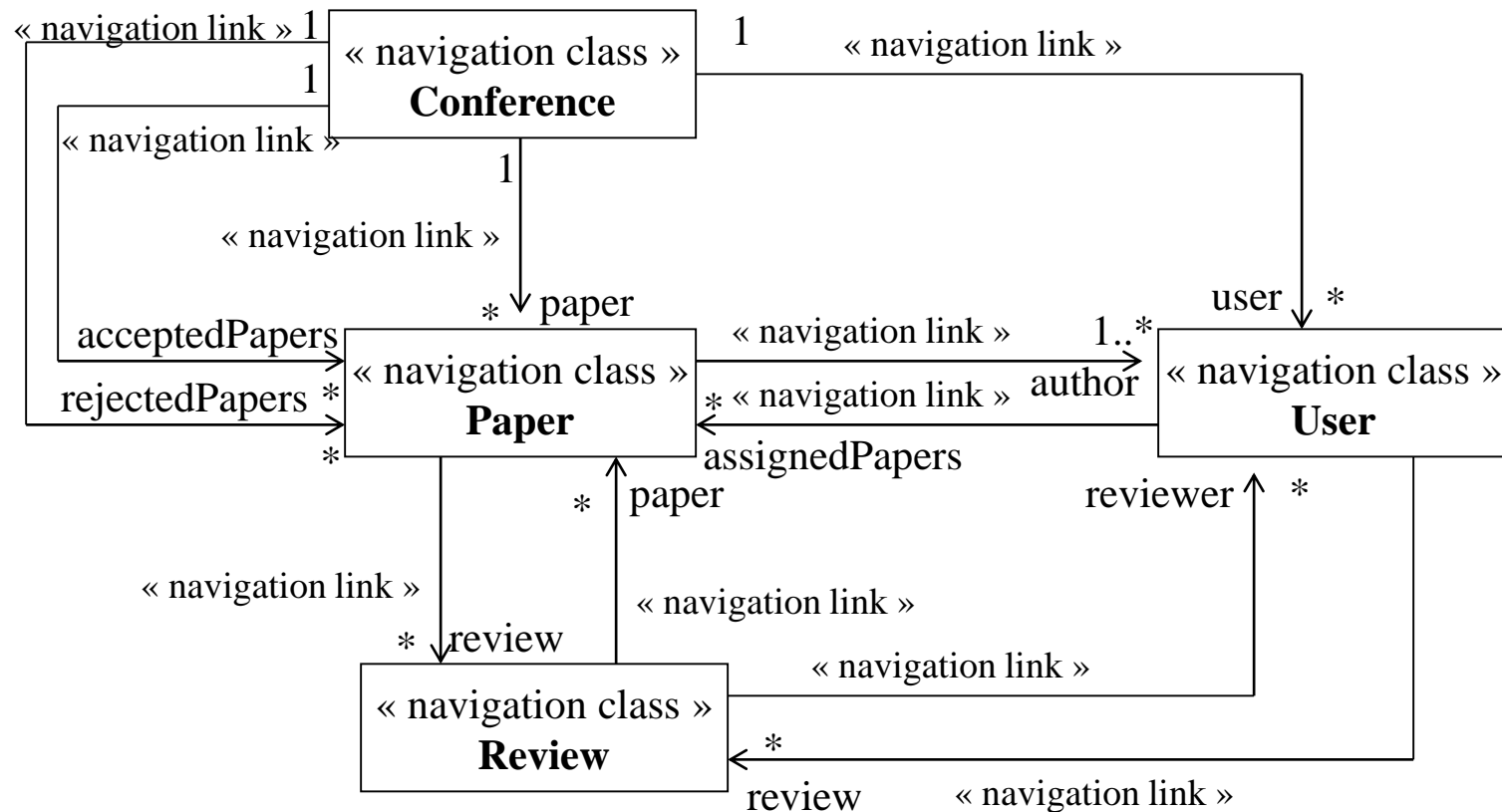
3.6.2. Retour sur le deuxième exemple (21)

- Diagramme de classes UML: exemple du système de reviewing

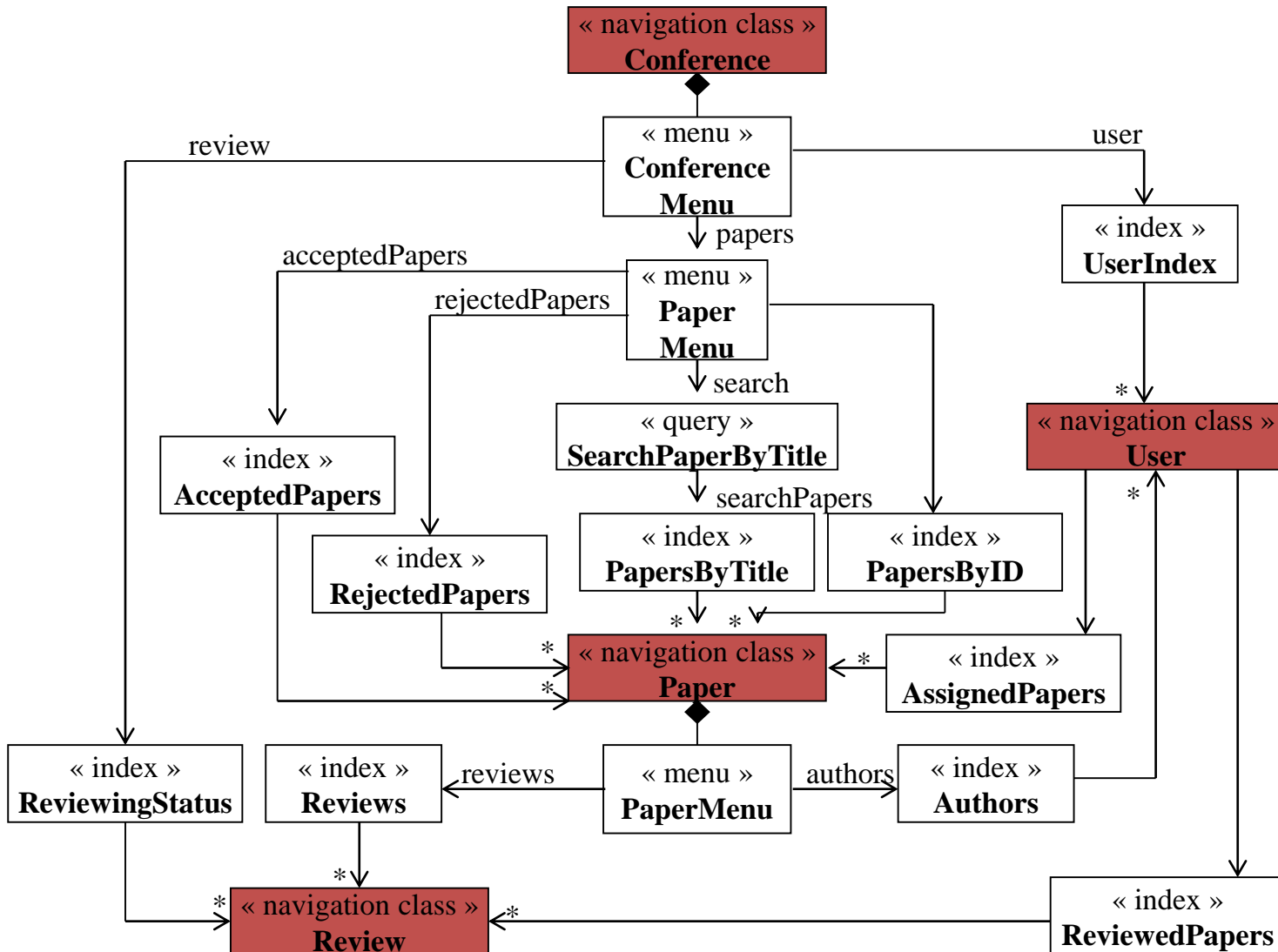


3.6.2. Retour sur le deuxième exemple (22)

- Diagramme de la structure de la navigation: vue du PC chair sur le système de reviewing

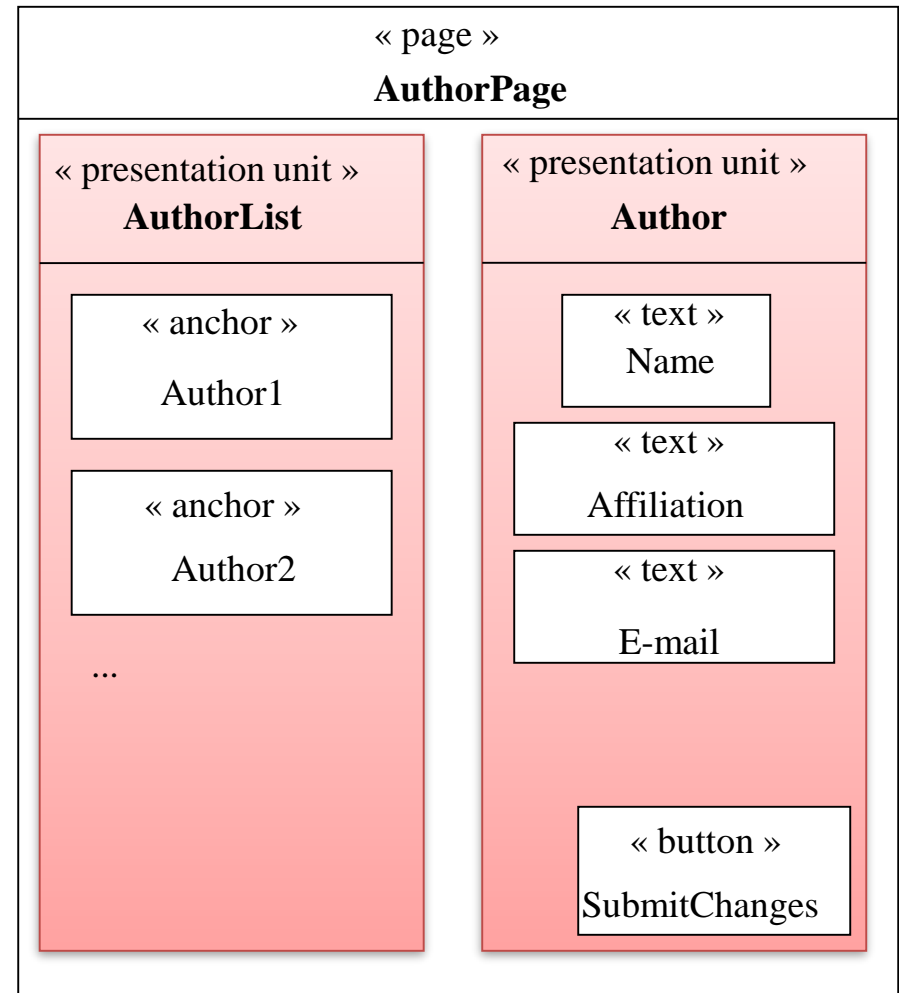
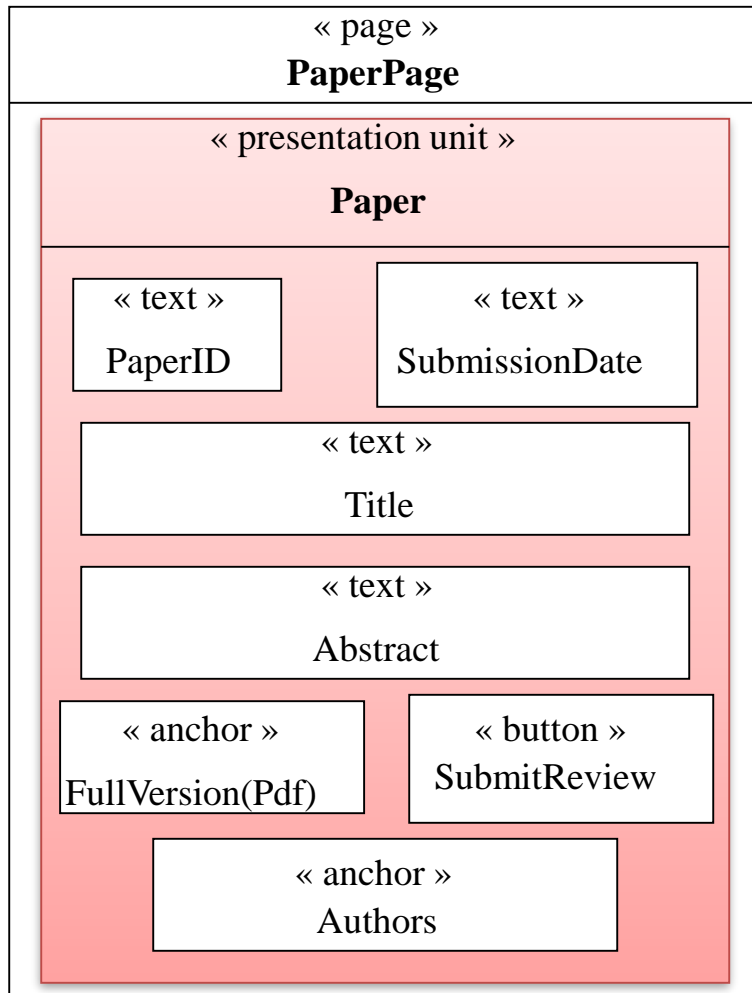


3.6.2. Retour sur l'exemple: Diagramme d'accès (23)



3.6.2. Modélisation de la présentation (24)

- Modèle statique des pages du système de reviewing



3.7. Modélisation de la présentation (1)

- Objectif:
 - Représentation explicite de la structure et du comportement de **l'interface utilisateur**
 - Définition de **pages** sous forme d'une composition hiérarchique d'éléments de présentation
- Approches: peu de méthodes
 - Modélisation statique
 - Modèles de classe pour la structure de présentation (container UML)
 - Modélisation dynamique
 - Diagrammes de séquence pour l'aspect comportement
- Résultats:
 - Modèle de personnalisation

3.7. Modélisation de la présentation (2)

- Définit la manière dont la structure de navigation est présentée à l'utilisateur
- Elle comprend le modèle de la présentation statique et le modèle de présentation dynamique

3.7. Modélisation de la présentation (3)

- Modèle de présentation statique :
 - Définit comment les nœuds de navigation du modèle de navigation sont présentés à l'utilisateur
 - Consiste en une collection d'objets d'interface utilisateur représentés par les objets composites UML
 - Un objet de représentation cadre sert à regrouper ces éléments

3.7. Modélisation de la présentation (4)

Modèle de présentation statique :

- Un **objet d'interface utilisateur** est construit en combinant des primitives appelées **objets de présentation** qui dépend de l'état d'un objet de navigation
- Les objets d'interface utilisateur peuvent avoir leur propre état
- Leur comportement est décrit dans le modèle de présentation dynamique

3.7. Modélisation de la présentation (5)

- Modèle de présentation statique :
 - La sémantique des objets d'interface utilisateur est:
 - **Objet de présentation** : composé à partir d'autres objets et dépend de l'état d'un nœud de navigation
 - **Une ancre** : secteur cliquable, point de départ d'une navigation. Se compose d'une présentation (texte, bouton, forme, image, vidéo) ainsi que d'un lien
 - **Un texte**, séquence de caractères avec une information de formatage
 - **Un bouton**, secteur cliquable qui a une action associée
 - **Une forme**, employée pour demander une information à l'utilisateur. Est composée de champs d'entrée, de menus, de checkboxes
 - **Les images**, acoustique et vidéo peuvent être mises en marche

3.7. Modélisation de la présentation (6)

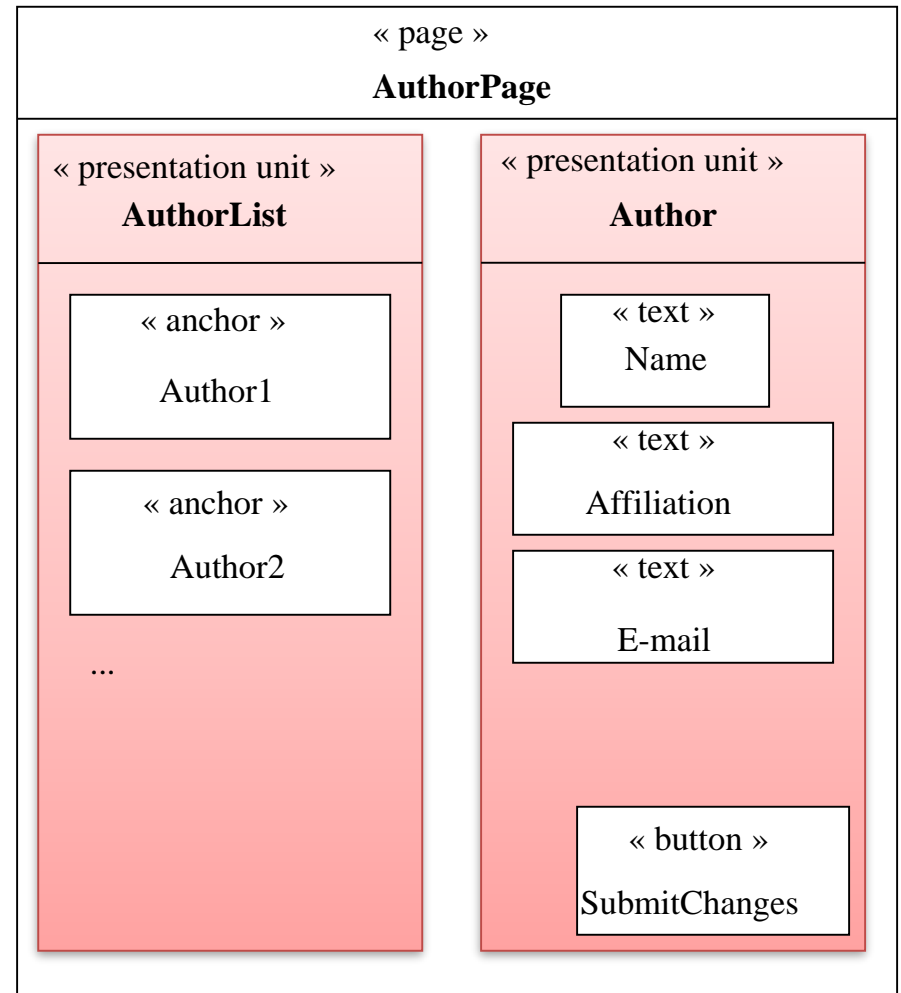
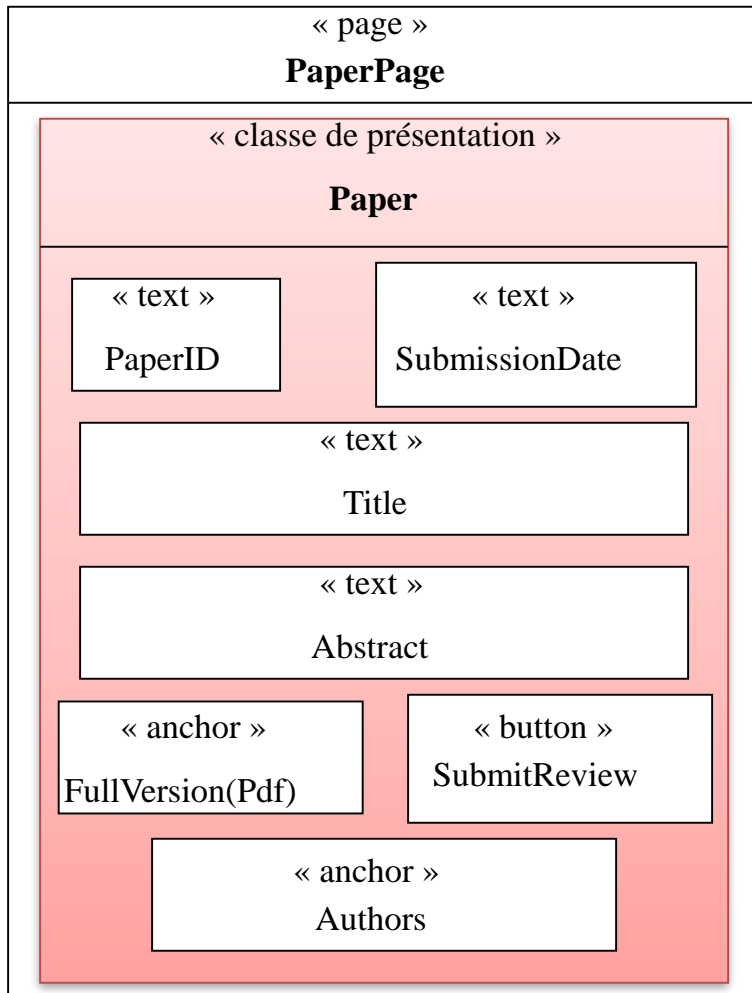
- Concepts :
 - « page »
 - représente une unité de visualisation contenant tous les éléments à présenter à l'utilisateur en réponse à sa requête
 - Peut être composée de plusieurs **classes de présentation**
 - « classe de présentation »
 - Sert à regrouper des unités de présentation reliées (fragments logiques d'une page) représentant des **éléments de l'interface utilisateur**. C'est une logique de présentation
 - « élément de présentation »
 - « Anchor »
 - « Text »
 - « Image »
 - « button »

3.7. Modélisation de la présentation (7) : La pratique

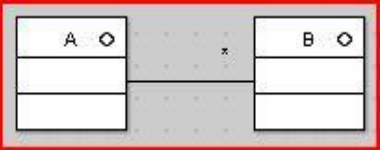
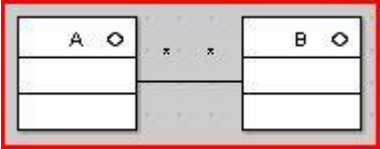
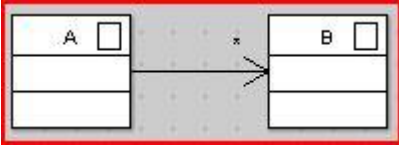
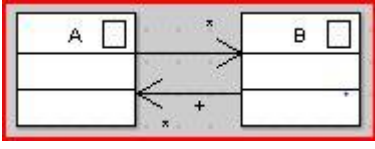


- Construire une **classe de présentation pour chaque classe du modèle d'accès** (utiliser les stéréotypes « text », « image », « audio », « video »)
- Construire une **classe de présentation pour chaque menu et index du modèle d'accès** (utiliser les stéréotypes « anchor » ou « anchored collection »)
- Construire une **classe de présentation pour chaque query** (utiliser le stéréotype « form » et pour chaque guidedtour utiliser un menu avec pour items « next » et « previous »)
- Ajouter des « anchors » aux classes de présentation pour permettre la création, la suppression et l'exécution d'opérations sur les objets du modèle conceptuel
- Déterminer l'ensemble des éléments de présentation à proposer à l'utilisateur (utiliser le stéréotype « UI view »)
- Construire des scénarii (storyboard) représentés par les séquences des UI views

3.7. Modélisation de la présentation statique (7)

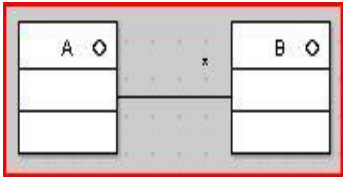
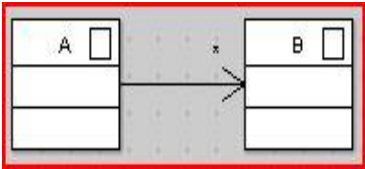

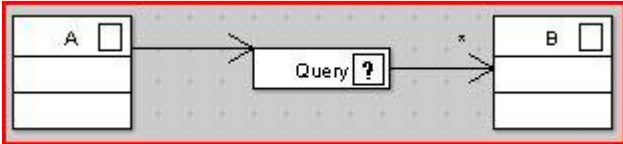
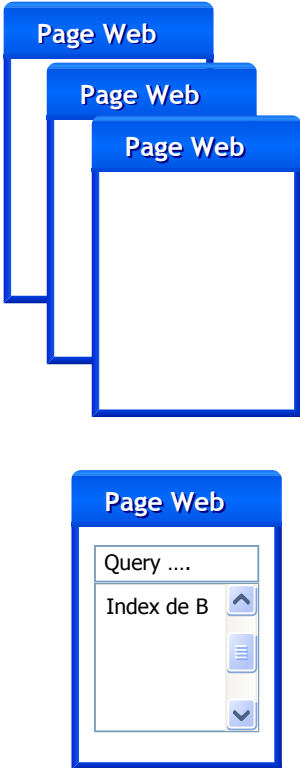
- Exemple des pages de présentation du système de reviewing



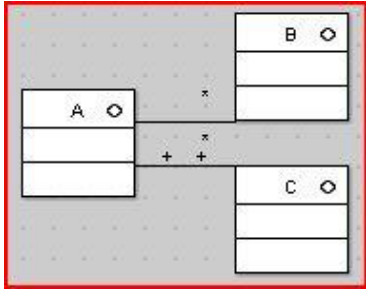
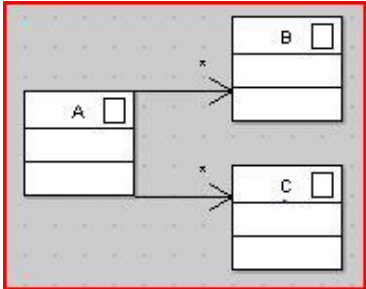
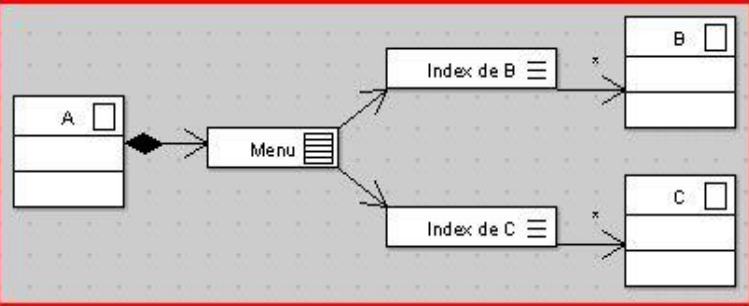
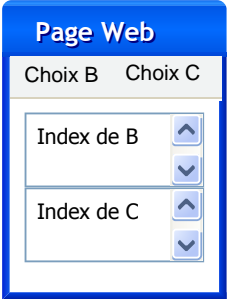
3.7. Règles de génération des modèles (8)

Contenu	Navigation	Accès	Présentation
« classe »	« classe de navigation », nœud	nœud	objet de présentation
« liens »	« lien de navigation »	index, menu, query, guidedTour	
<p>lien 1---1 *</p>  <p>lien *----*</p> 	 	<p>« index » accès aux nœuds individuels d'une classe</p> 	

3.7. Règles de génération du modèle de présentation (9)

Contenu	Navigation	Accès	Présentation
		<p>« guidedTour » accès séquentiel à une liste de nœuds</p>  <p>« query » recherche d'un nœud et accès direct</p> 	

3.7. Règles de génération du modèle de présentation (10)

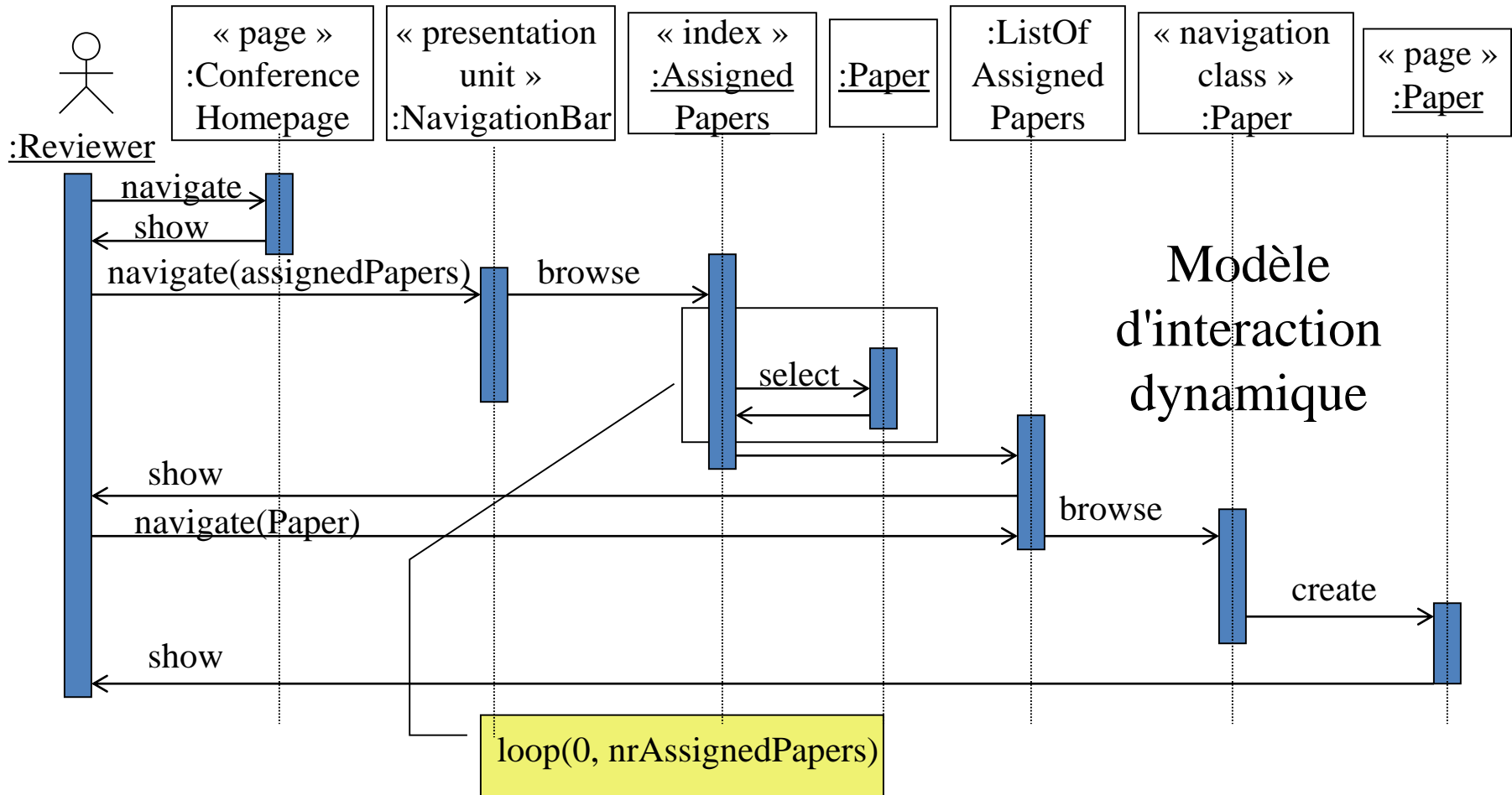
Contenu	Navigation	Accès	Présentation
		<p>« menu » accès aux nœuds de classes différentes</p> 	

- **Modèle de présentation dynamique**

- Emploie des diagrammes de machines d'états UML
- Sert à définir la réaction des objets d'interface utilisateur du modèle de présentation statique sur des événements externes d'utilisateur (mouvements de souris, clics de souris, touches de clavier enfoncées)
- Il en est de même pour la réaction aux événements internes (arrêts, activations, désactivations)
- Exemple d'utilisation des machines d'états UML: contrôler l'accès à une zone protégée par un mot de passe dans un fenêtrage

3.7. Modélisation de la présentation dynamique (12):

- Exemple du scénario pour le système de reviewing



3.7. Modélisation de la présentation (13)

- Modèle de présentation dynamique
 - Fonctionnement des objets d'interface utilisateur
 - A la réception d'un événement, il peut y avoir:
 - changement de la perception et de l'activation de variables,
 - génération de nouveaux événements,
 - envoie de messages à d'autres objets d'interface utilisateur ou à de objets de navigation
 - Un objet d'interface utilisateur composite délègue à ses composants les événements qu'il reçoit
 - Les objets d'interface utilisateur ont un comportement par défaut

3.7. Modélisation de la présentation (14)

- Modèle de présentation dynamique
 - Deux sortes d'objets interface utilisateur :
 - Ceux qui sont **perceptibles** par l'utilisateur
 - Ceux qui sont **en activité** (peuvent recevoir des événements de la part de l'utilisateur)
 - Les variables « **perception et activation** » sont employées respectivement dans chacun des deux cas

3.7. Modélisation de la présentation (15)

- Modèle de présentation dynamique : **La variable perception**
 - Elle contient la liste des objets perceptibles par l'utilisateur
 - Elle recense plusieurs mécanismes: le lien avec les événements internes SHOW et HIDE
 - Chaque fois qu'un élément est mis dans la variable perception, l'événement interne SHOW est produit
 - Chaque fois qu'un élément est enlevé, l'événement HIDE est produit

3.7. Modélisation de la présentation (16)

- Modèle de présentation dynamique : **La variable activation**
 - Elle contient l'objet d'interface utilisateur de la liste des objets perceptibles qui recevra les événements externes d'interface utilisateur
 - Chaque fois qu'un objet d'interface utilisateur est assigné à la variable (activation)
 - il recevra l'événement interne (activé)
 - et l'objet précédemment assigné (activation) recevra l'événement interne (désactivé)
 - L'exception: le bouton. Le modèle de présentation dynamique doit définir les actions qui sont effectuées quand un bouton est appuyé

3.7. Modélisation de la présentation (17)

- Modèle de présentation dynamique : **Méthode de construction d'un objet de présentation**
 - On peut composer un objet de présentation pour une classe de navigation. C'est un diagramme d'objet composite
 - Un élément de niveau supérieur est un cadre modélisé par un objet composite contenant des objets de présentation de plus bas niveau
 - Il peut contenir un nombre arbitraire de sous-cadres
 - Un cadre est une instance d'une classe stéréotypée par «cadre » avec une icône correspondante

3.7. Modélisation de la présentation (18)

- Modèle de présentation dynamique : **Méthode de construction d'un menu de présentation- La pratique**
 - Construire une présentation pour chaque classe de navigation et pour chaque classe d'index du modèle d'accès
 - Choisir une classe de navigation comme racine pour la navigation
 - Considérer tous les chemins possibles dans le modèle d'accès de la classe racine à la classe réelle pour chaque classe de navigation et pour chaque index
 - Combiner les résultats des étapes 1 et 3 de façon à ce que n'importe quel cadre contienne:
 - Un sous cadre droit avec la présentation de la classe de navigation ou de la classe d'index (étape 1)
 - Et un sous cadre gauche représentant l'arbre de navigation (étape 3)

3.8. Modélisation de l'adaptation (1)

- Objectifs : représentation explicite de l'information de contexte et ses implications sur la présentation
- Approches :
 - Modélisation statique (différents modèles de contextes)
 - Modélisation dynamique (un modèle + des règles d'adaptation)
- Résultats : un modèle d'adaptation

3.8. Modélisation de l'adaptation (2)

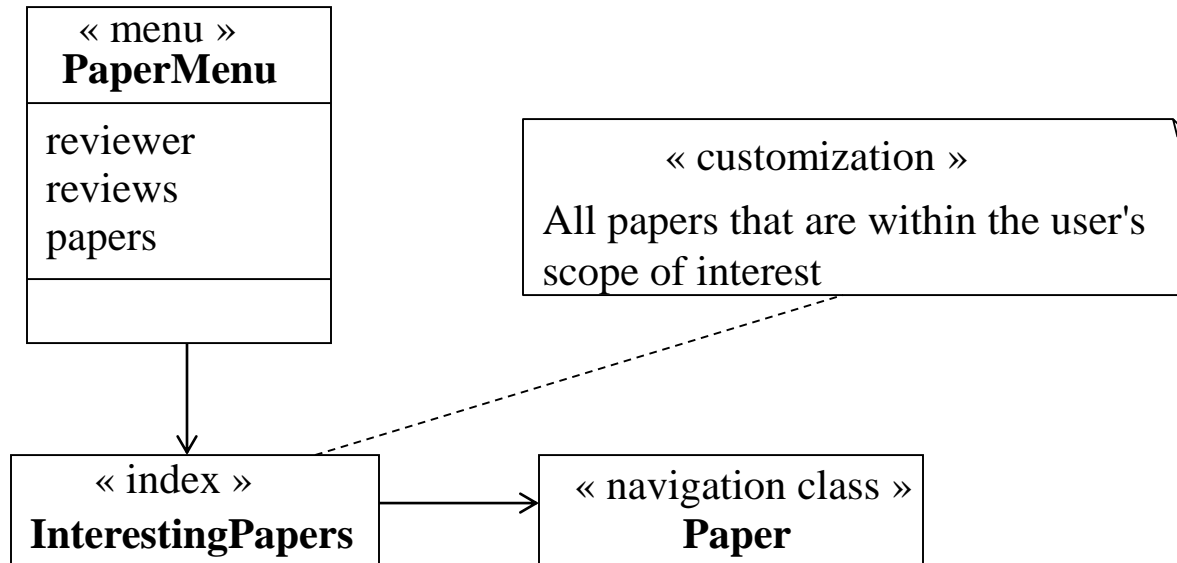
- Fondée sur la notion de règles
- Une règle est modélisée par une classe stéréotypée « rule »
- Elle est liée par une composition à une classe « condition »
- Les conditions et les actions sont liées à des classes du modèle utilisateur et du diagramme de classes

3.8. Modélisation de l'adaptation (3)

- Les règles d'adaptation spécifient les conditions d'adaptation des contenus, de la navigation et de la présentation
- Les règles d'acquisition décrivent le mode d'acquisition des informations au sujet de l'utilisateur et donc la MAJ du modèle utilisateur
- Pour décrire le comportement observé du système, on utilise la classe stéréotypée «UserBehaviour »
- Les règles (conditions et actions) sont exprimées en OCL
- Des diagrammes de communication sont construits pour mettre en évidence les enchaînements éventuels de règles

3.8. Modèle d'adaptation (4)

- Exemple : adaptation dynamique d'un index dans le modèle hypertexte



3.8. Modèle d'adaptation (5)

- Exemple :

