



Développement du modèle dynamique

LES DIAGRAMMES DE SEQUENCES OBJETS

De l'analyse à la conception

DÉVELOPPEMENT DU MODÈLE DYNAMIQUE

Il constitue la troisième activité de l'étape d'analyse. Pour le modèle 2TUP, elle se situe sur la branche gauche du cycle en Y. Il s'agit d'une activité itérative et incrémentale, fortement couplée avec le modèle statique (DCL), s'effectuant quasiment en parallèle. Le développement du modèle dynamique précède l'étape de conception préliminaire.

CONCEPTION D'UN CAS D'UTILISATION

Nous avons vu au chapitre 2 qu'un cas d'utilisation décrit un ensemble de scénarios. Lors de l'étape de détermination des besoins fonctionnels, un scénario représente une séquence d'interactions entre le système et ses acteurs. Le système est alors considéré comme une boîte noire. Une fois que nous avons développé le modèle statique d'analyse (DCL), nous allons remplacer le système par une collaboration d'objets dans chaque scénario.

UN SCÉNARIO D'UTILISATION

Un scénario décrit une exécution particulière d'un cas d'utilisation du début à la fin. Il correspond à une sélection d'enchaînements du cas d'utilisation. On peut distinguer plusieurs types de scénarios :

1. nominaux : ils réalisent les postconditions du cas d'utilisation, d'une façon naturelle et fréquente ;
2. alternatifs : ils remplissent les postconditions du cas d'utilisation, mais en empruntant des voies détournées ou rares ;
3. aux limites : ils réalisent les postconditions du cas d'utilisation, mais modifient le système de telle sorte que la prochaine exécution du cas d'utilisation provoquera une erreur ;
4. d'erreur : ne réalisent pas les postconditions du cas d'utilisation.

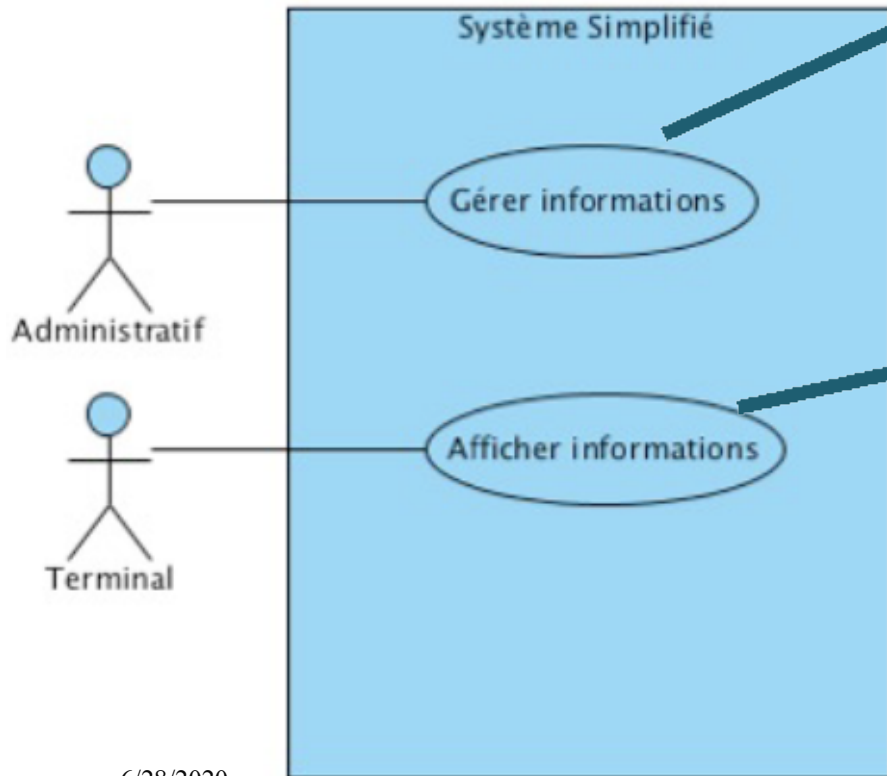


DE L'ANALYSE À LA CONCEPTION

Diagramme de séquences système

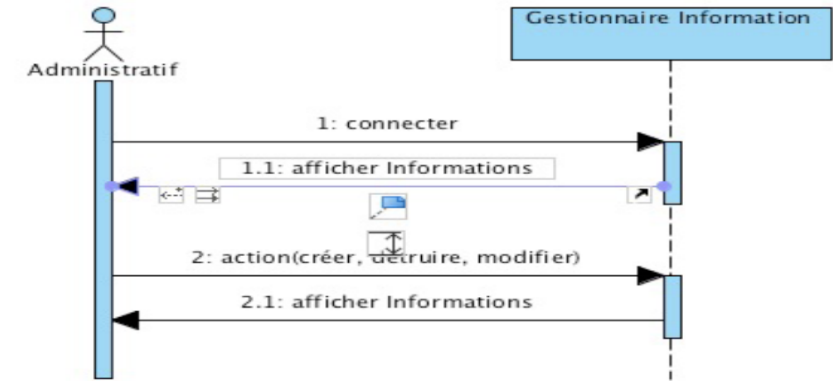
NIVEAU ANALYSE

Diagramme de cas d'utilisation

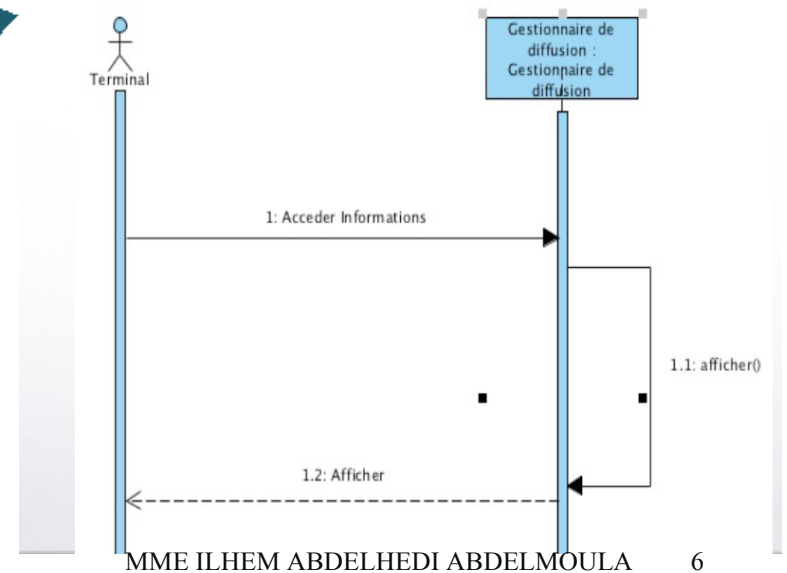


6/28/2020

Gérer Informations : niveau Analyse



Afficher Informations : niveau Analyse



MME ILHEM ABDELHEDI ABDELMOULA

DE L'ANALYSE À LA CONCEPTION

Cas d'utilisation

Diagramme de séquences :

Opérations nécessaires dans le diagramme de classe:

CHOIX DE L'ARCHITECTURE

Présentation

Gestion des Informations

Liste des informations

- nuit de l'info [2010-11-12 13:40:18,3]
- Devint [Fri, 26 Nov 2010 22:53,44]
- Rendu Projet ACSI [Sat, 27 Nov 2010 17:55,53]

Modifier Détruire

Titre de l'information

Créer une nouvelle information

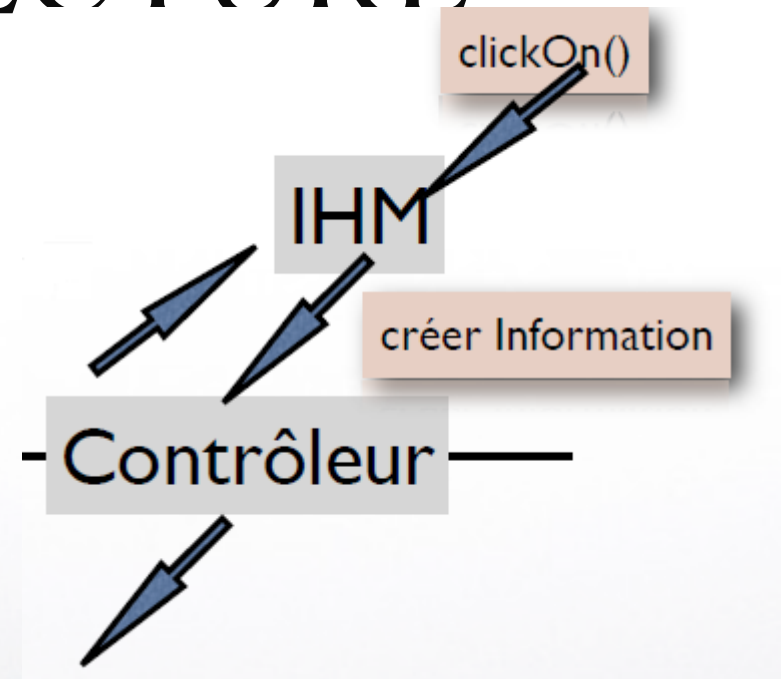
Gérer les informations : métiers

Stockage

```
CREATE TABLE `information` (  
  `titre` varchar(20) NOT NULL,  
  `date` varchar(22) NOT NULL,  
  `identifiant` int(11) NOT NULL auto_increment,  
  PRIMARY KEY (`identifiant`))
```



ORACLE



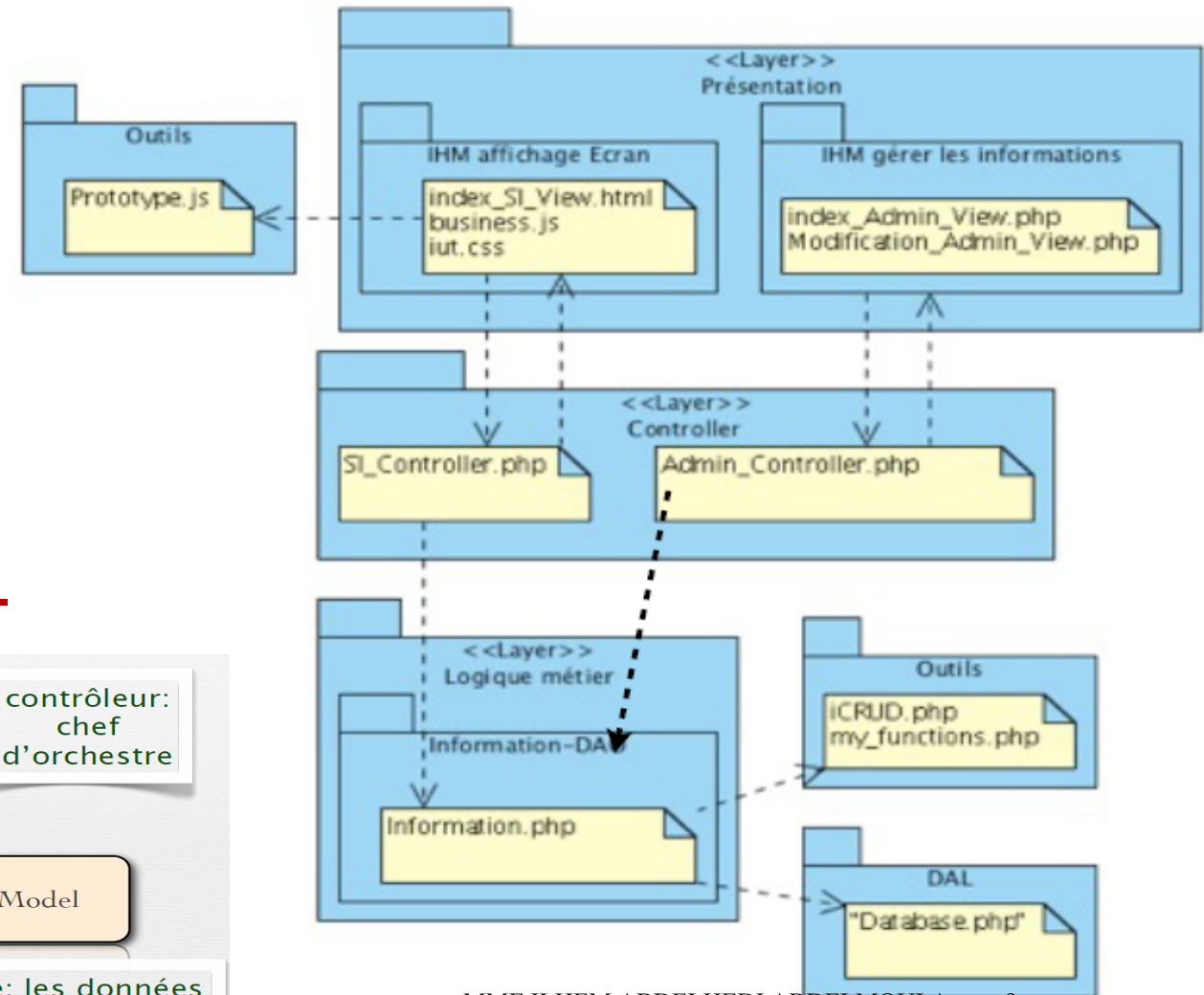
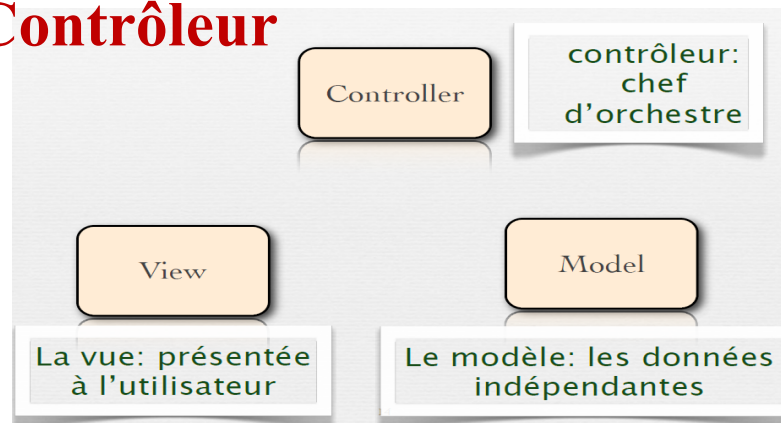
ARCHITECTURE EN COUCHES

- MVC

Architecture en couches

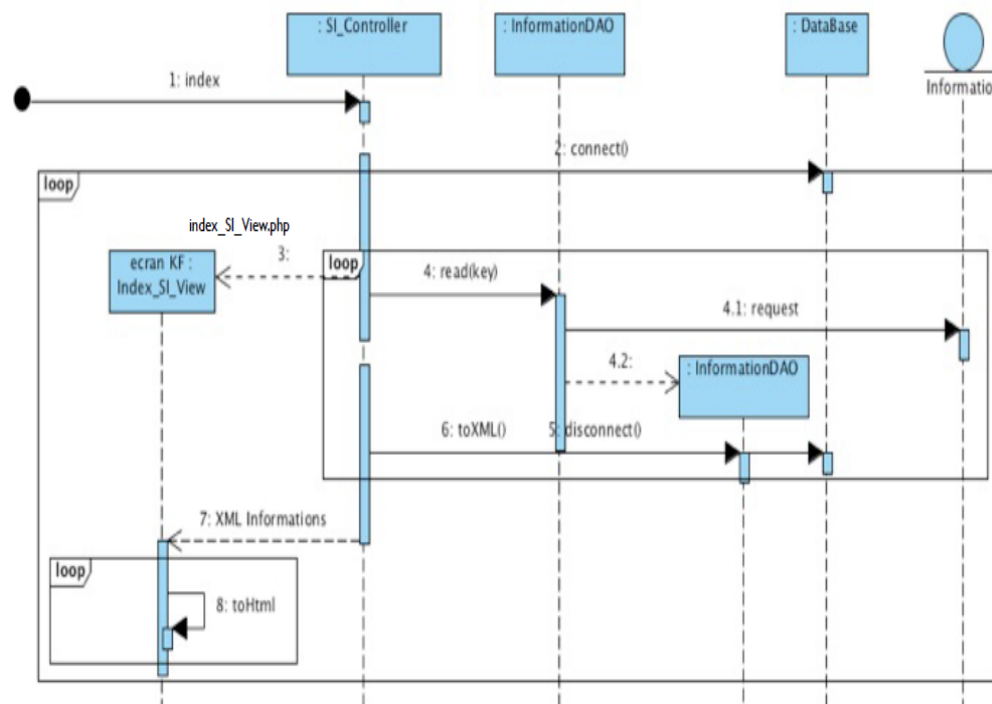
Point de vue	Classes	Couches (Tiers)	MVC
Statique	Métier – entité	Stockage - Accès aux données	Model
Fonctionnel	Interface – IHM	Présentation	View
Dynamique	Control – Manager	Logique Applicative	Control

MVC : Modèle – Vue – Contrôleur



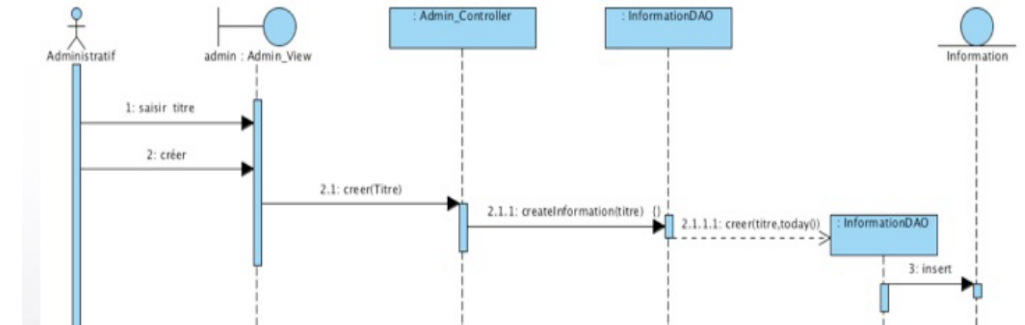
NIVEAU CONCEPTION – DIAGRAMME DE SÉQUENCES OBJET

Afficher Informations

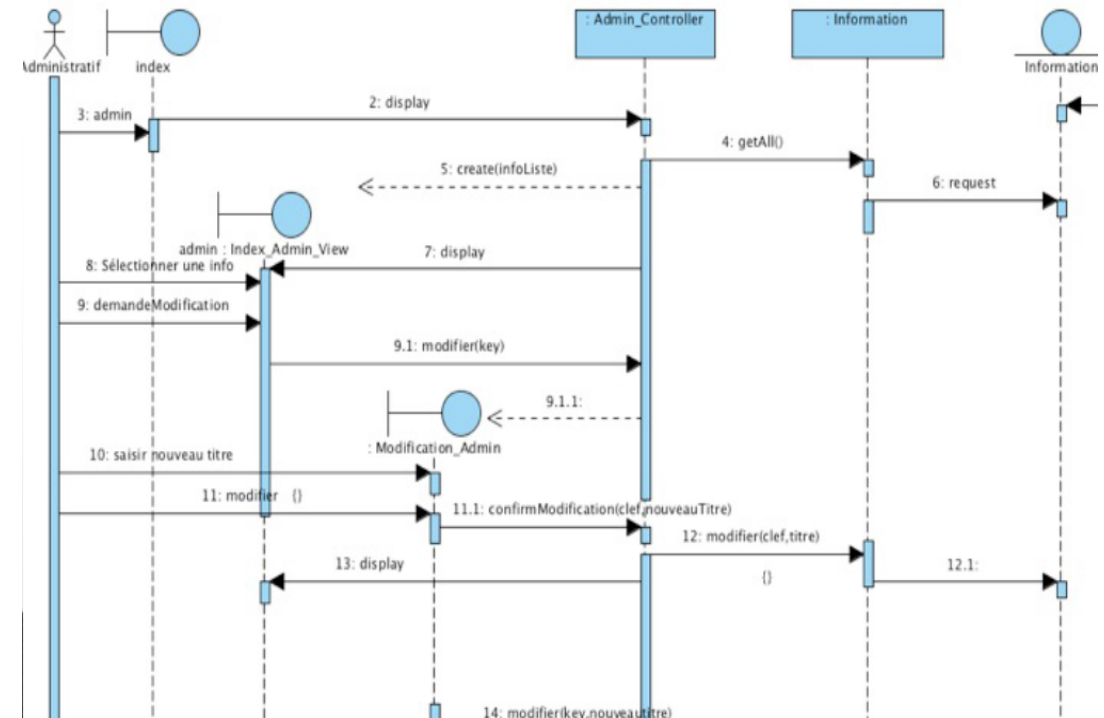


6/28/2020

Créer Informations



Modifier Informations



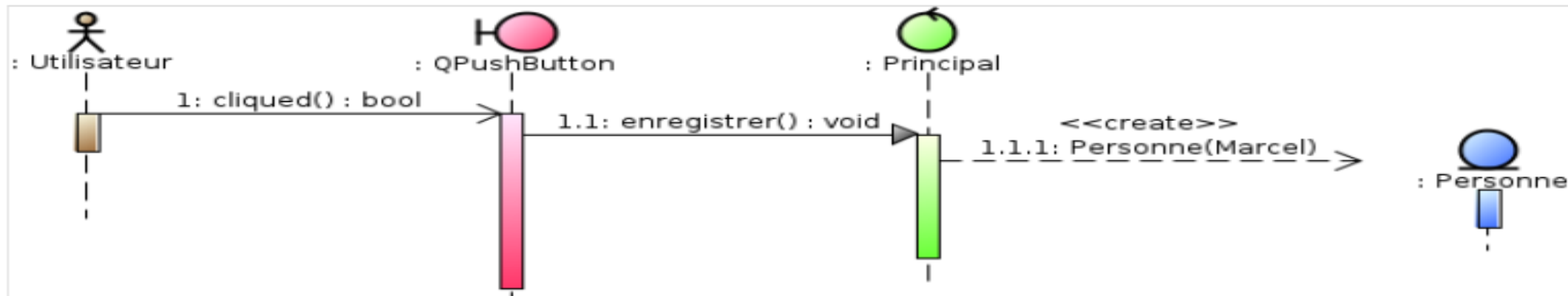
MME ILHEM ABDELHEDI ABDELMOULA

10

LES STÉRÉOTYPES DE JACKOBSON – ICONES ASSOCIÉES

Jacobson distinguent les trois stéréotypes suivants :

- o « **boundary** » : classes qui servent à modéliser les interactions entre le système et ses acteurs.
- o « **control** » : classes utilisées pour représenter la coordination, l'enchaînement et le contrôle d'autres objets.
- o « **entity** » : classes qui servent à modéliser des informations durables et souvent persistantes.



RÉALISATION ANALYSE CAS D'UTILISATION

- Dans un diagramme de séquence, on a généralement un acteur qui interagit avec le système.
- Ces interactions se font à travers une classe avec le stéréotype « **boundary** ».
- Une « boundary » est une interface graphique ou une autre interface Homme/Machine (écran tactile, PocketPC, ...)

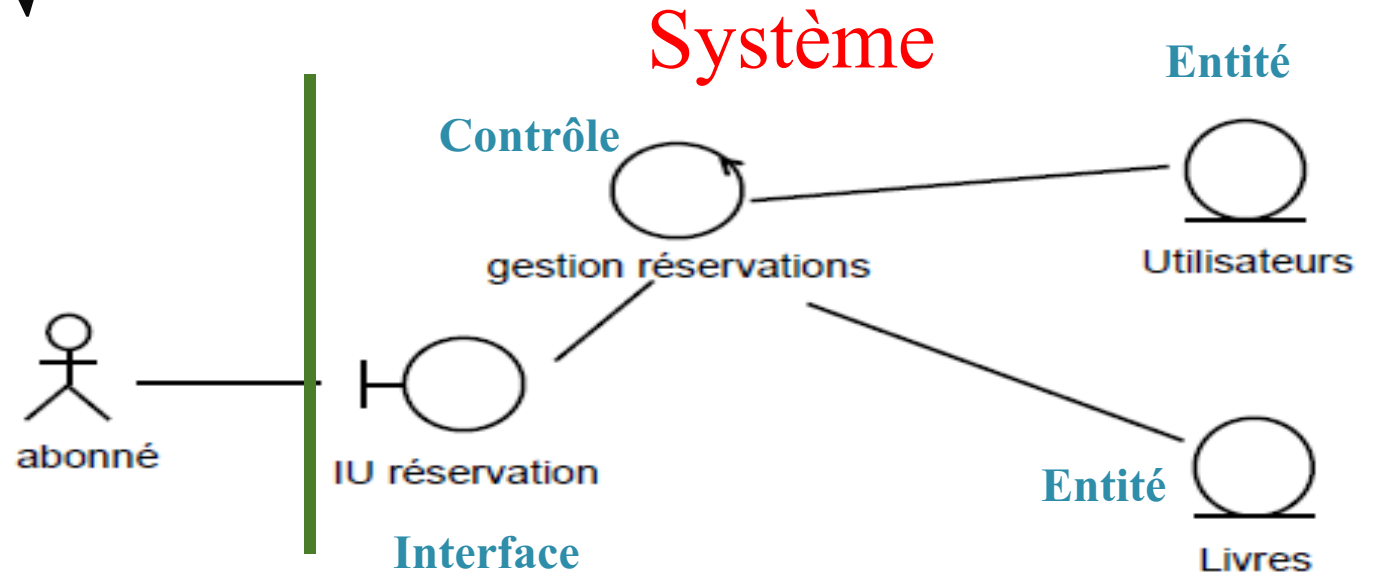
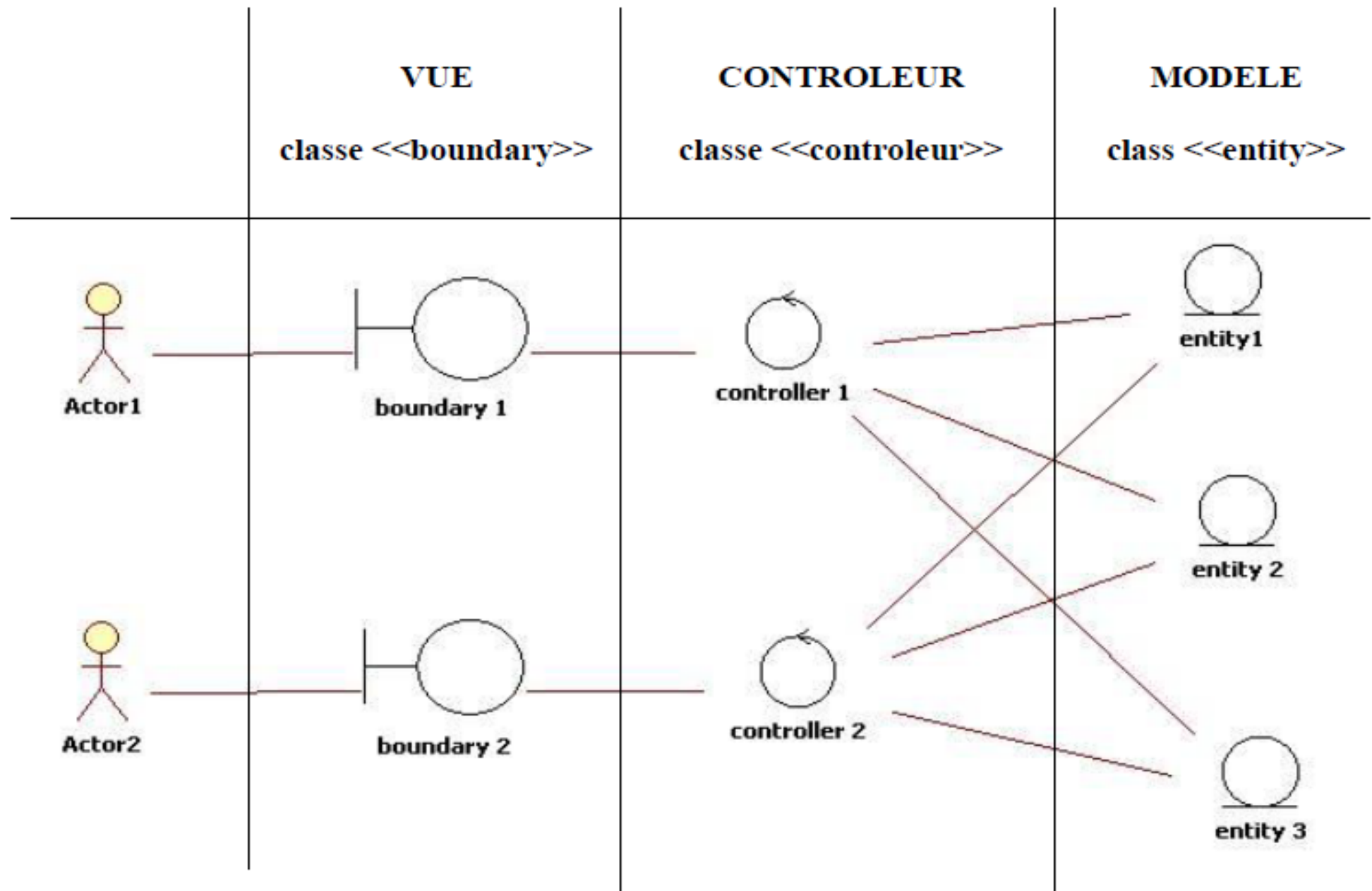


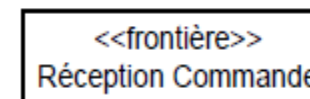
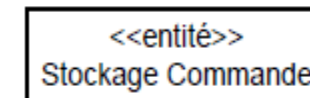
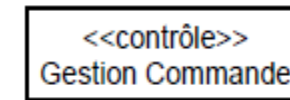
Diagramme de séquences objet ou Boundary



DÉFINITION DE CLASSES D'ANALYSES

On s'aperçoit que dans l'analyse d'un problèmes, on distingue trois types de classes; ce qui nous conduit aux couches applicatives classiques (3 tiers) et au modèle MVC.

1. **Classe frontière (ou Boundary):** La classe qui permet au système de communiquer avec le monde réel, elle est à la frontière du système, elle se conçoit en général par une interface graphique
2. **Classe métier (ou Entité) :** La classe qui mémorise et gère des données, par exemples les livres présents, les prêts effectués...
3. **Classe contrôle (ou Manager) :** La classe qui réalise le contrôle nécessaire pour interpréter le scénario décrivant un cas d'utilisation



CLASSE ENTITÉ – POINT DE VUE STATIQUE

Principe

- Le point de vue statique met l'accent sur les **concepts du domaine** et les associations qui les relient.
- A la place de classe « entité » on dit plutôt **classe « métier »**
- C'est le point de vue des données **persistantes** : celles qu'on retrouve **en BD**.

Méthode de construction du diagramme de classe métier

- Analyse du type de celle faite pour un MCD. Il s'agit de trouver les classes (entités), les attributs et les associations à partir de la description « métier » du domaine.
- Ensuite, on peut s'intéresser aux responsabilités des classes.
- On n'a pas forcément intérêt à ajouter les classes « interface » et « control » car finalement, ça charge le modèle.

Formalisme UML

- Les **classes métier** sont stéréotypées << entité >>
- Inutile dans le diagramme de classes métier qui contient que des classes << entité>>.
- Utile dans des diagrammes mélangeant les 3 types de classe.
- On trouve aussi ce symbole UML pour les classes métier :



CLASSE INTERFACE – DE POINT DE VUE DYNAMIQUE

Principe

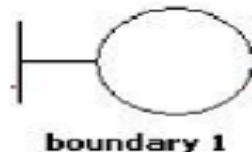
- Il met l'accent sur les interfaces : interfaces utilisateur ou interfaces avec d'autres systèmes actifs.
- Les classes « interface » constituent la partie du système qui dépend de l'environnement.
- Elles correspondent à l'IHM.

Méthode pour trouver les classes interface

- Examiner les paires acteur-cas d'utilisation
- Les classes contiennent les méthodes appelées lors du lancement d'un cas d'utilisation.
- D'où l'intérêt de regrouper les cas d'utilisation.

Formalisme UML

- Les classes interface sont stéréotypée << interface >> ou << boundary >>
- Utile dans des diagrammes mélangeant les 3 types de classe.
- On trouve aussi ce symbole UML pour les classes interface :



CLASSE CONTRÔLE – POINT

Principe

- Il met l'accent sur le lien entre l'interface et les classes métier.
- Il peut ne consister qu'à faire le lien : c'est alors uniquement de la « colle ».
- C'est aussi le lieu de calculs divers.

Méthode pour trouver les classes contrôle

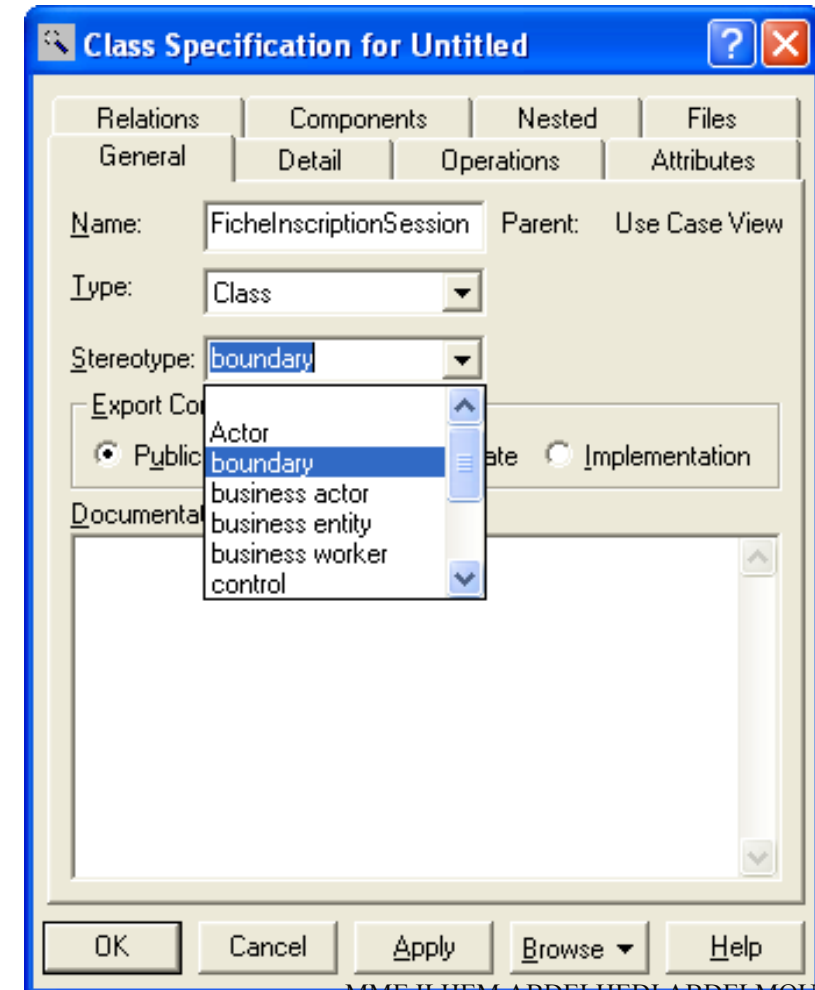
- On associe une classe contrôle à chaque classe interface.
- La classe contrôle peut ensuite utiliser plusieurs classes métier.

Formalisme UML

- Les classes controle sont stéréotypée << controle >> ou << manage >>
- Utile dans des diagrammes mélangeant les 3 types de classe.
- On trouve aussi ce symbole UML pour les classes interface :

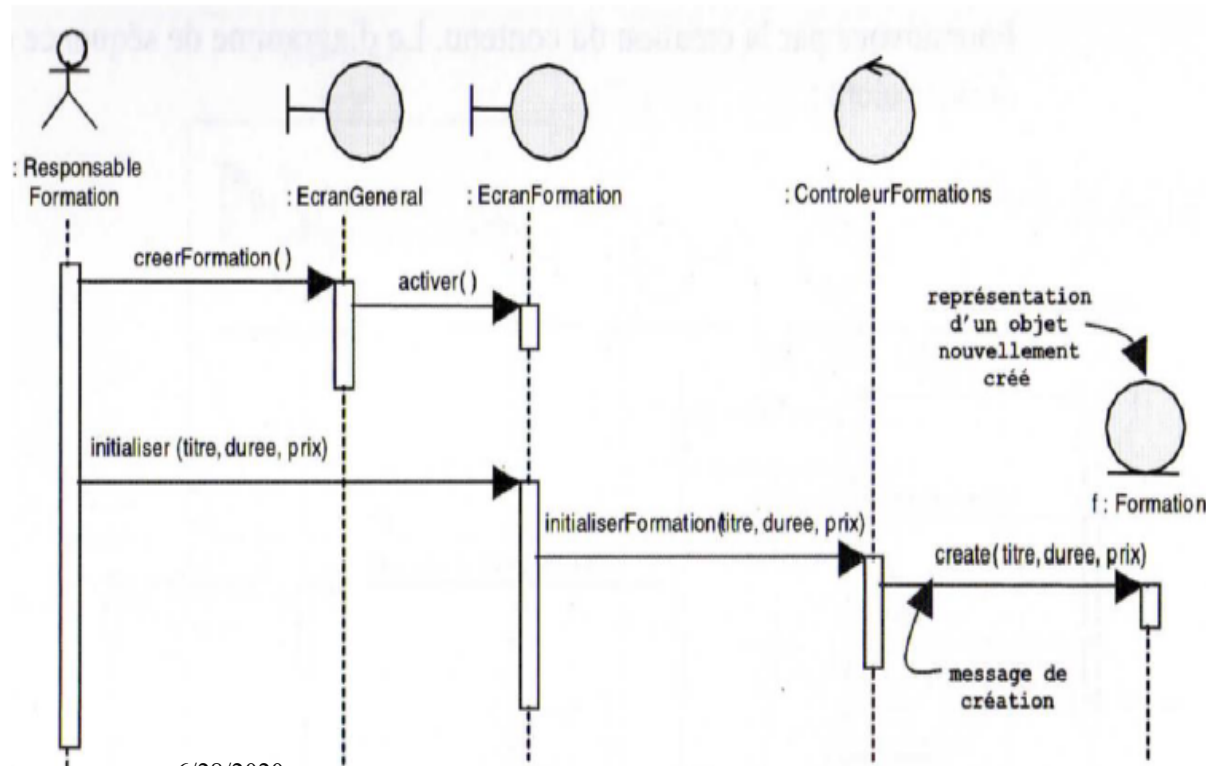


DIAGRAMMES DE SÉQUENCE : « BOUNDARY »



EXEMPLE 2 – NIVEAU CONCEPTION

Diagramme de séquences objet



6/28/2020

Diagramme de communication

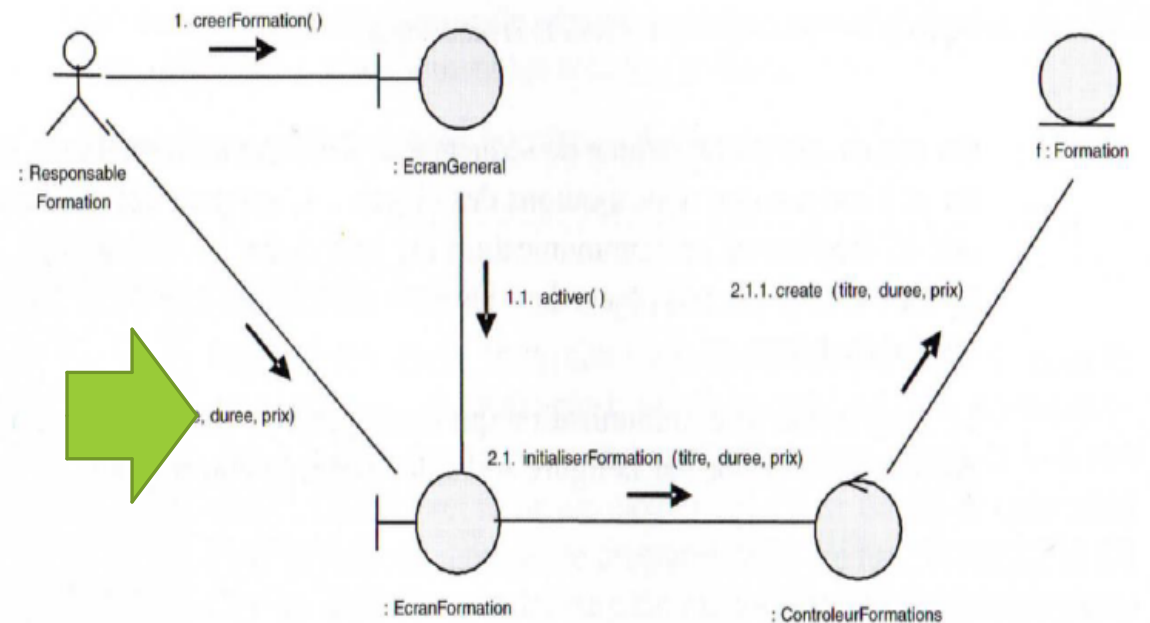


DIAGRAMME DE SÉQUENCE

OBJET

