

Université Mohammed Premier

École Nationale des Sciences Appliquées

Oujda

Formation : Génie Informatique
Niveau : 5^{ème} année

Séance n° I

Unified Modeling Language 2.x

UML 2.x

Présentée par :
Redouane ESBAI

Plan du cours

➤ UML 2.x

- ✎ Notion du diagramme
- ✎ Les 14 diagrammes d'UML 2.0
- ✎ Les nouveaux diagrammes d'UML 2.0

➤ OCL 2.x

- ✎ Historique d'OCL 2.x
- ✎ Principes d'OCL 2.x
- ✎ Différentes sortes de contraintes OCL 2.x

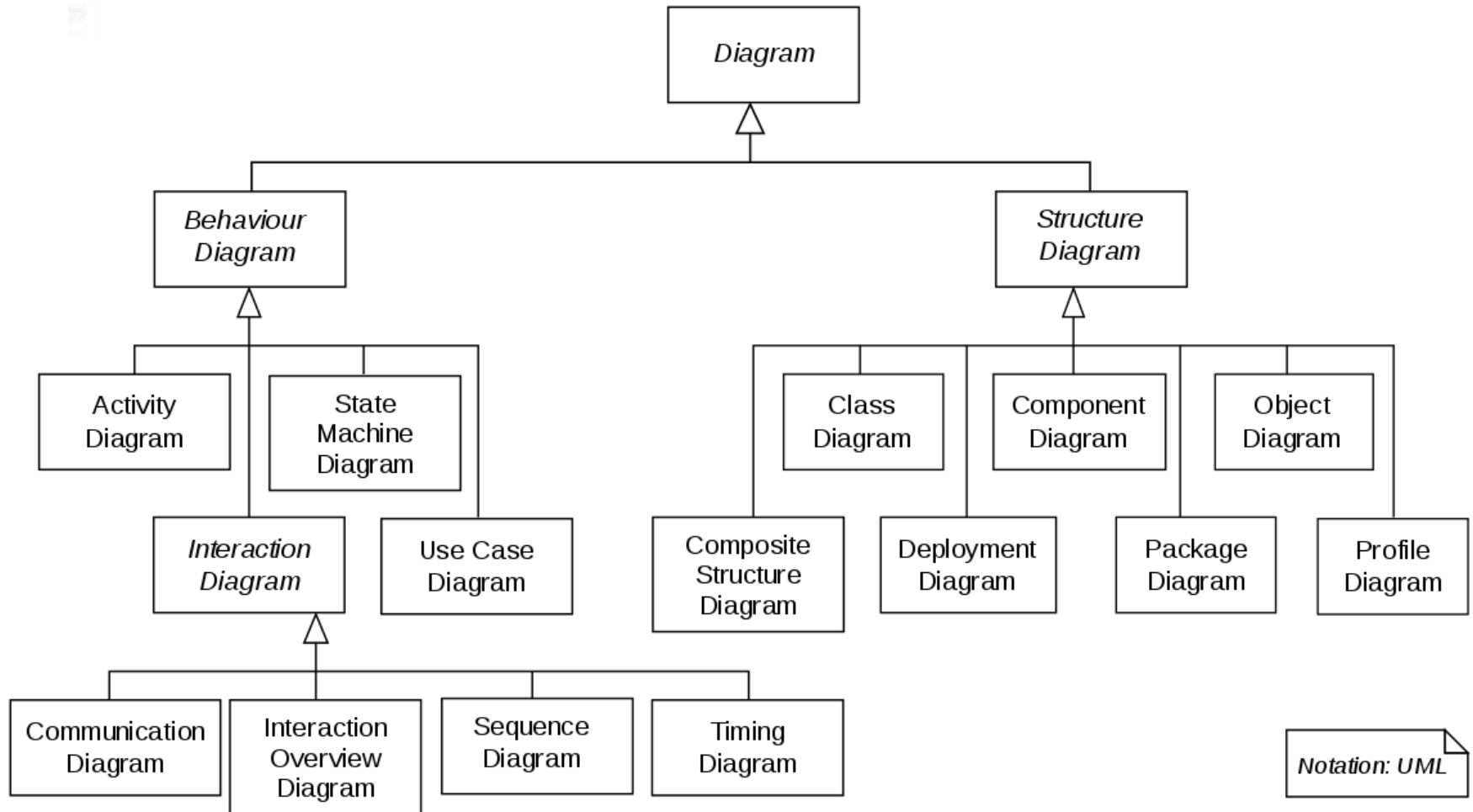
➤ MDA

- ✎ Concepts et principes de MDA
- ✎ Techniques de transformation MDA
- ✎ Langage de transformation QVT 2.0
- ✎ Applications du MDA

Notion de diagramme

- Un diagramme regroupe des éléments de modélisation ayant un même objectif de représentation
- Un diagramme illustre un point de vue sur le système
- La modélisation d'un système est la réunion des éléments de modélisation représentés une ou plusieurs fois sur un ou plusieurs diagramme;
- Les diagrammes peuvent être classés suivant leur type, c'est-à-dire une catégorisation de ce que chacun cherche à exprimer;

Les diagrammes UML 2.x



Les diagrammes UML 2.x

➤ Diagrammes structurels

- ☐ Diagramme de classes
- ☐ Diagramme de structure composite
- ☐ Diagramme d'objets
- ☐ Diagramme de déploiement
- ☐ Diagramme de composants
- ☐ Diagramme de paquets

➤ Diagrammes comportementaux

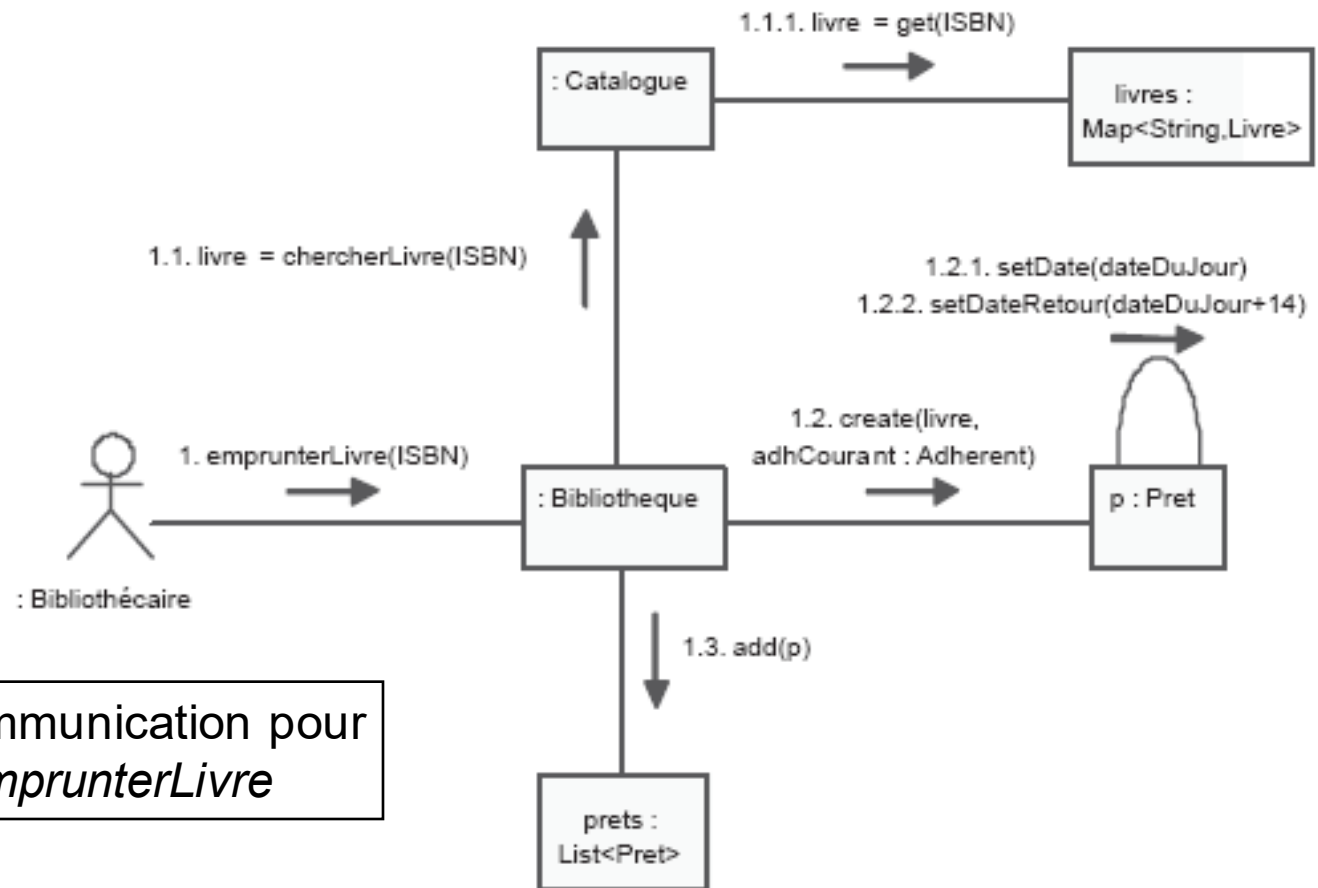
- ☐ Diagramme d'activités
- ☐ Diagramme de communication
- ☐ Diagramme d'interactions d'ensemble
- ☐ Diagramme de séquences
- ☐ Diagramme de machine finie à états
- ☐ Diagramme de chronométrage
- ☐ Diagramme de cas d'utilisation
- ☐ Diagramme de profils

Nouveaux diagrammes

- Diagramme de communication
- Diagramme d'interaction hybrides (overview of interaction)
- Diagramme temporel (timing diagrams)
- Diagramme de structure composite

Diagramme de communication

Le diagramme de communication (appelé collaboration en UML 1.x) montre comment des instances envoient des messages à d'autres instances. Pour qu'un message puisse être reçu par un objet, une opération de même nom doit être déclarée publique dans la classe concernée.



un diagramme de communication pour l'opération système *emprunterLivre*

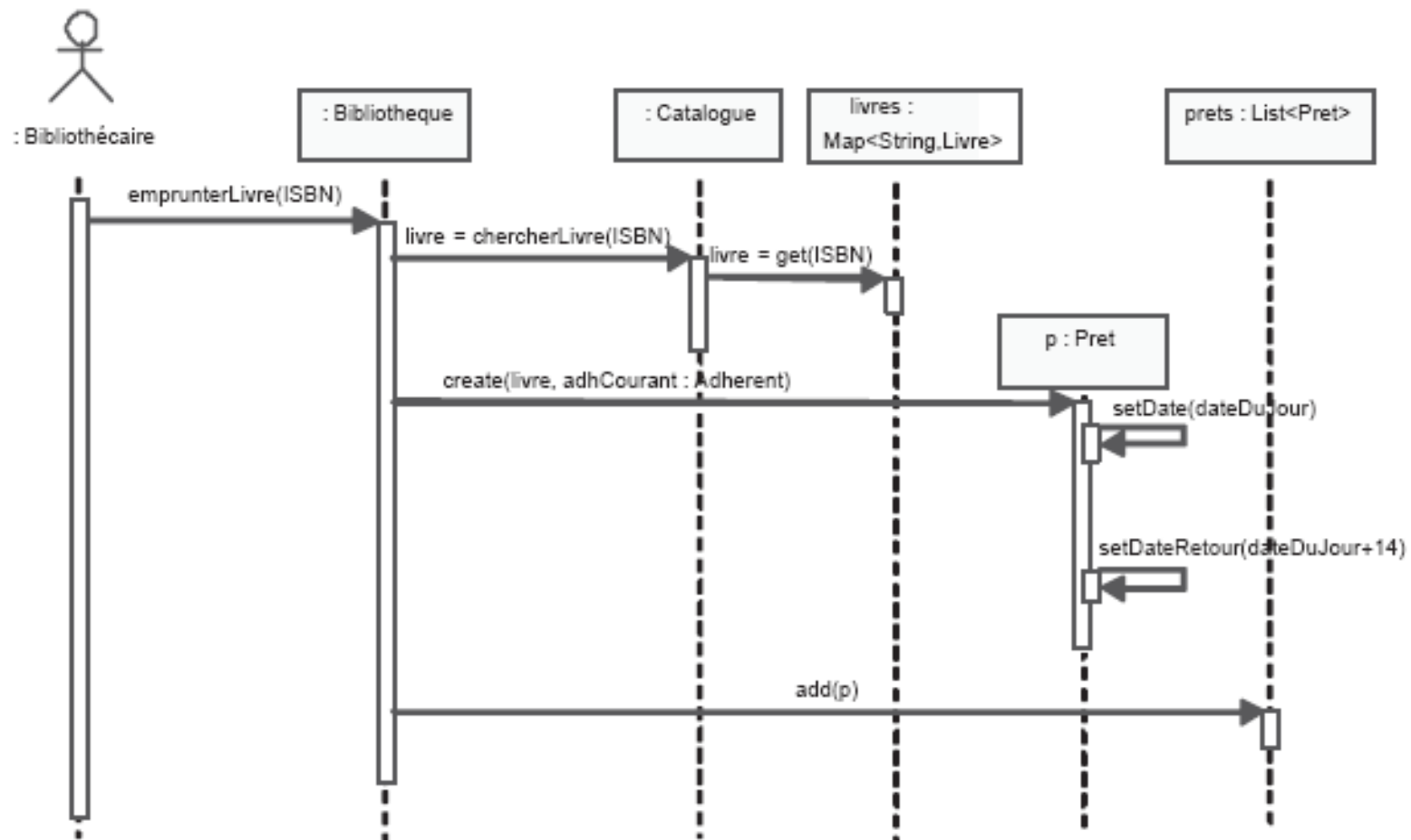


Diagramme d'interaction d'ensemble

- Greffe de diagrammes de séquence et diagramme d'activité
- Les actions sont remplacées par des DS ou des références à ces derniers

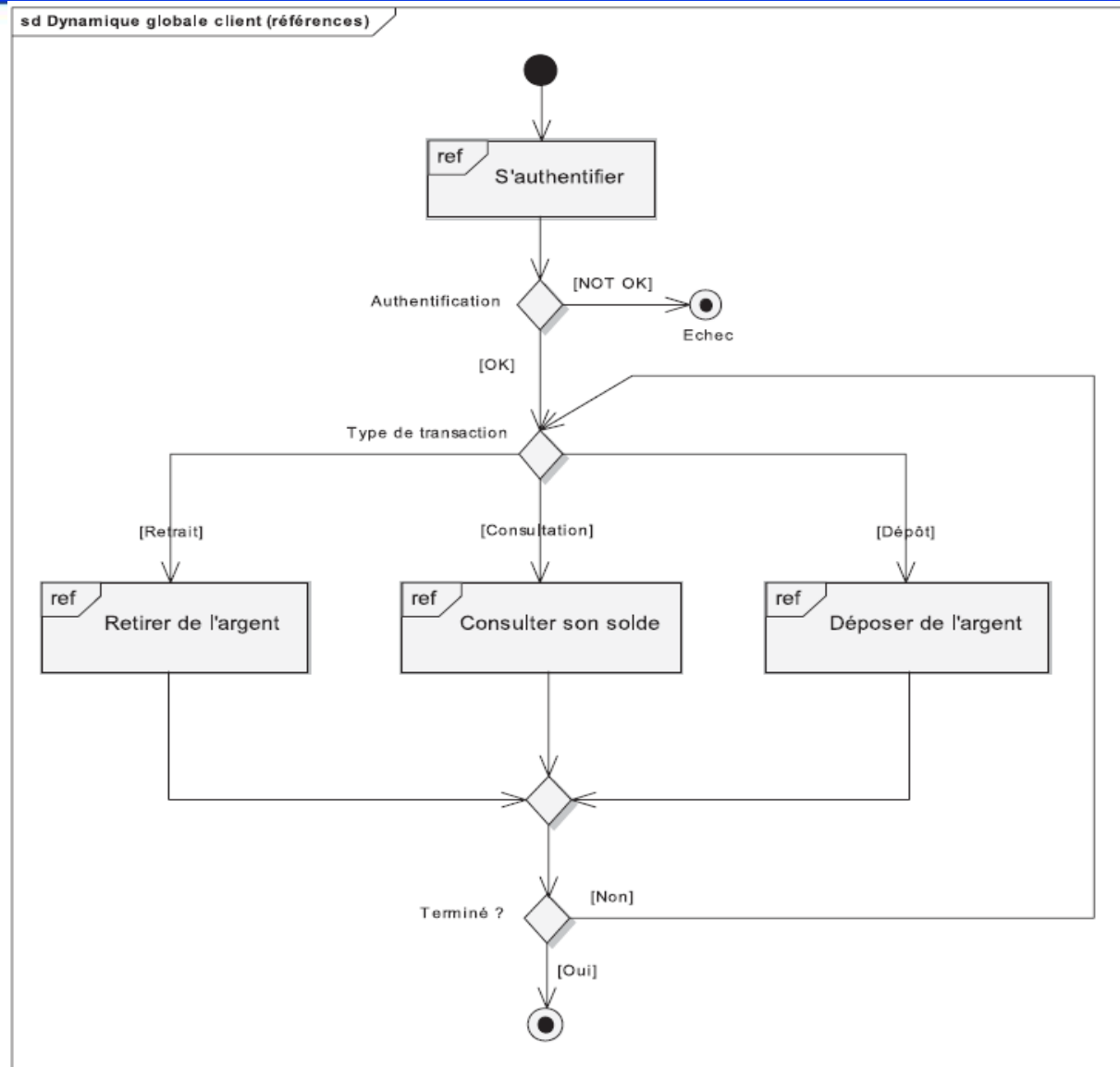
Exemple :

Représentation de la dynamique globale du GAB dans le point de vue de l'acteur Client banque, en représentant les interactions de ses cas d'utilisation dans un *Interaction Overview Diagram*.

On Modélise en particulier le fait que le client peut enchaîner plusieurs transactions (retrait, dépôt, etc.) sans avoir à s'authentifier de nouveau.

Diagramme d'interaction d'ensemble

Il est également possible de remplacer chaque référence par un diagramme de séquence « inline ». Nous l'avons fait pour l'interaction s'authentifier. Il est clair que le souci de lisibilité du diagramme empêche de remplacer chaque référence par un diagramme de séquence entier.



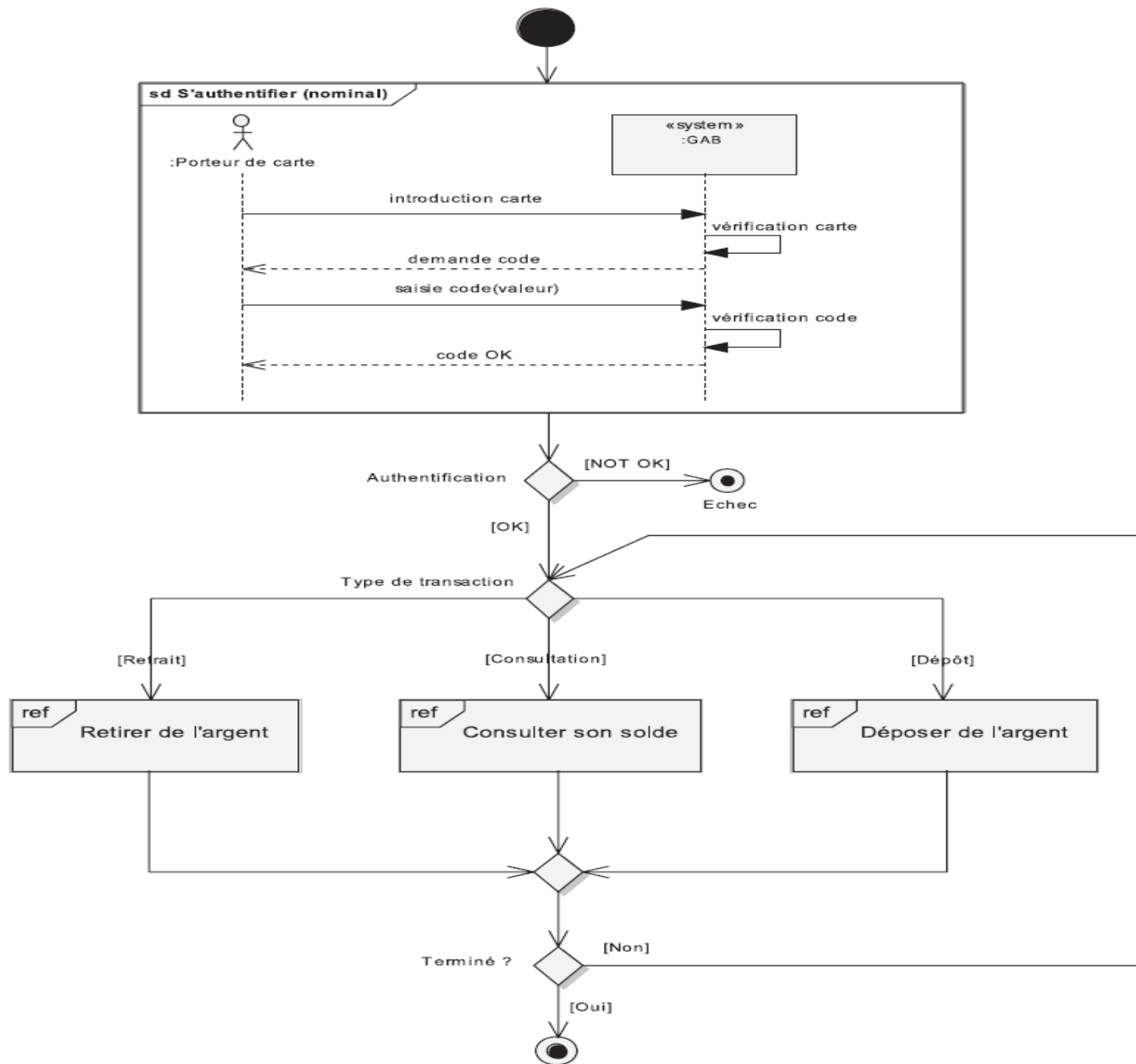


Diagramme de chronométrage ou de temps

Montre l'évolution de l'état d'un objet ou d'un groupe d'objets en fonction d'évènements temporels

Ce diagramme combine le diagramme de séquence et d'état pour proposer un point de vue sur l'évolution de l'état d'un objet au fil du temps, et sur les messages qui modifient cet état.

Diagramme de chronométrage ou de temps

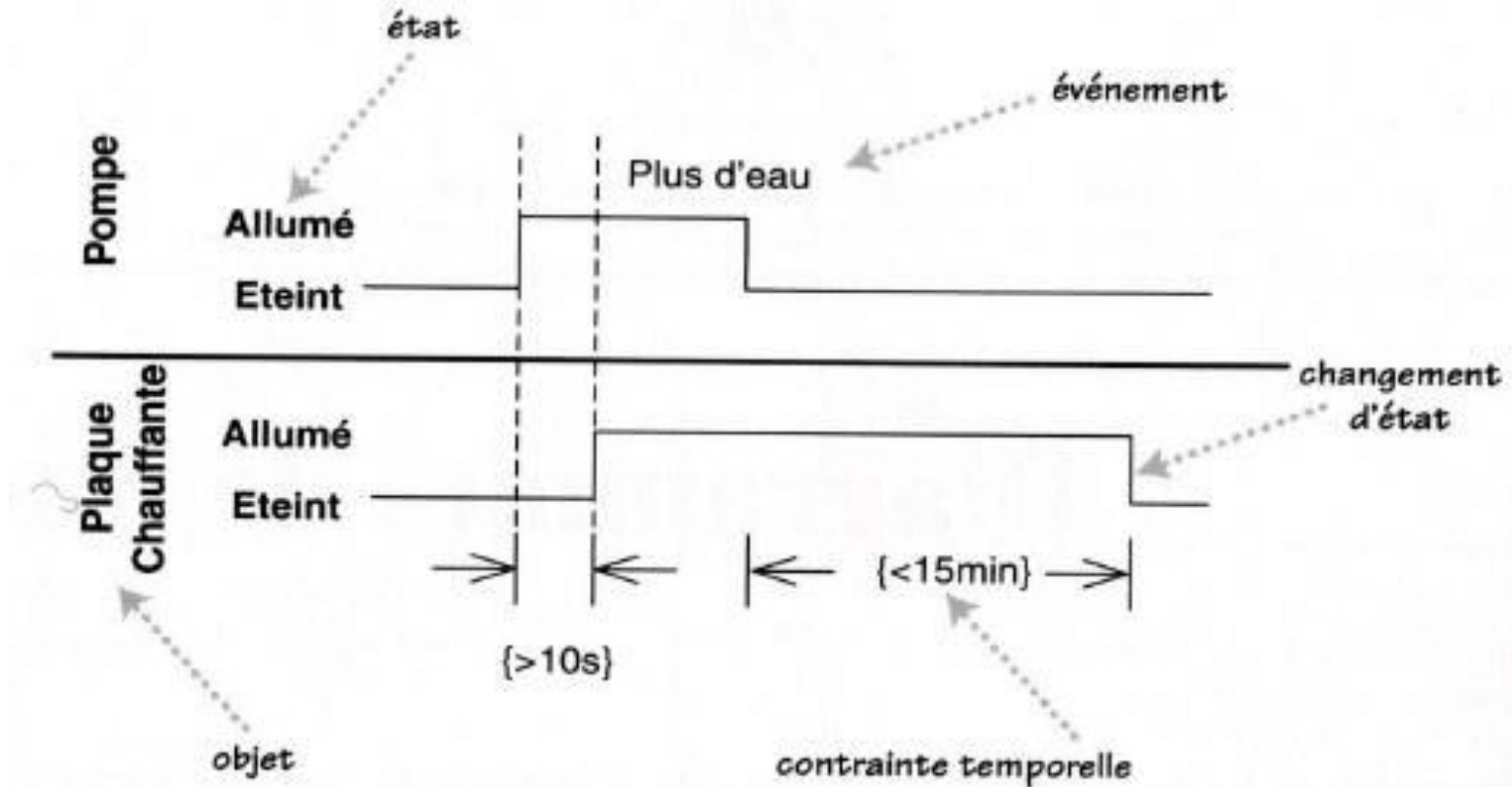
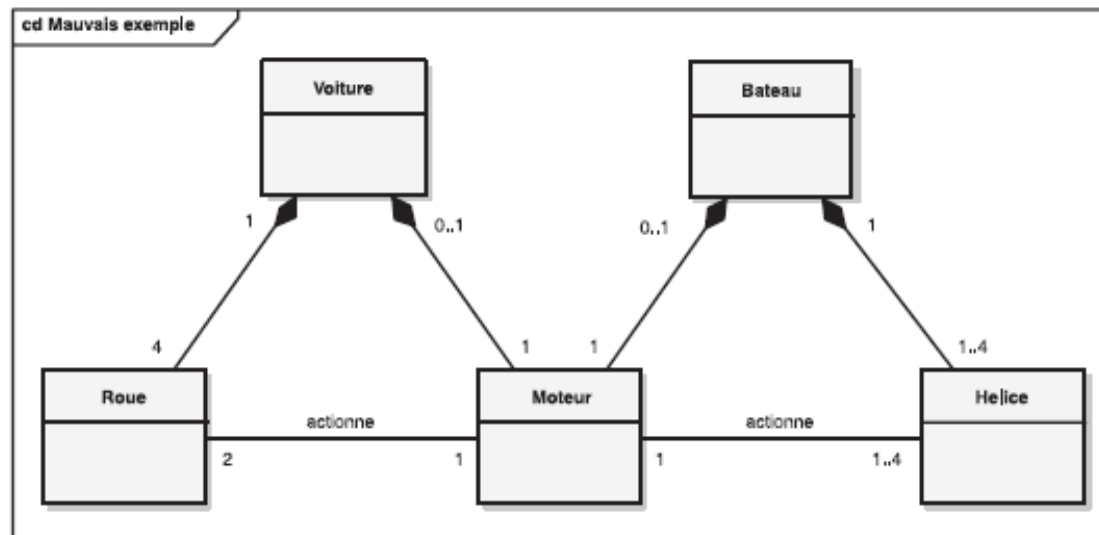


Diagramme de structure composite

explique que les voitures et les bateaux possèdent un moteur, que les voitures ont des roues alors que les bateaux ont des hélices, et que les moteurs se décomposent tous de façon similaire. La composition exprime également le fait que la même instance de *Moteur* ne peut pas appartenir simultanément à une instance de *Voiture* et une instance de *Bateau*, même si la classe *Moteur* est partagée entre les classes *Voiture* et *Bateau*. La composition signifie encore qu'une instance de *Moteur* est détruite quand son instance conteneur de *Voiture* ou de *Bateau* est détruite.



La relation de composition telle qu'elle existait dans UML 1.x fonctionne bien pour modéliser de la décomposition hiérarchique. Cependant, elle présente des limitations significatives lorsqu'il s'agit de relier des éléments au même niveau de décomposition.

Diagramme de structure composite

les associations sont définies globalement pour tous les moteurs, pas dans le contexte de voitures ou de bateaux particuliers. Cela signifie donc que :

- Le moteur d'une certaine voiture peut actionner les hélices d'un bateau, ou les roues d'une autre instance de voiture.
- Chaque moteur est censé actionner à la fois des roues et des hélices. Si l'on avait mis une multiplicité 0..1 au lieu de 1 du côté *Moteur*, on aurait pu avoir cette fois-ci des moteurs n'actionnant rien du tout.
- Un moteur peut actionner les deux roues gauches d'une voiture, au lieu des roues avant. L'association ne spécifie pas quel couple de roues est actionné.

On pourrait dessiner un grand nombre de digrammes d'objets cohérents avec le diagramme de classes précédent, mais complètement stupides !

UML 2 traite le problème en introduisant un nouveau type de diagramme appelé « diagramme de structure composite », avec les nouveaux concepts de classe structurée, participant et connecteur.

Diagramme de structure composite

- Le diagramme de structure composite décrit la composition d'un objet complexe lors de son exécution. Ce diagramme est propre à UML 2 ; il introduit la notion de structure d'un objet complexe, tel qu'il se présente en phase de run-time.
- Diagramme conceptuellement assez proche d'un diagramme de composants
 - ❑ Définit l'architecture interne d'une classe
 - ❑ Les éléments qui la forment (les *parts*)
 - ❑ Les interactions entre ces éléments (d'une manière proche des diagrammes de collaboration)

Diagramme de structure composite

CLASSE STRUCTURÉE

Une classe structurée est une classe dotée d'une structure interne. Elle contient des participants (*parts*) reliées par des connecteurs. Une instance d'une classe structurée contient un objet ou un ensemble d'objets correspondant à chaque participant. Au sein de son conteneur, un participant possède un type et une multiplicité. Tous les objets d'un objet structuré unique sont implicitement reliés par le fait qu'ils sont contenus par le même objet.

CONNECTEUR

Un connecteur est une relation contextuelle entre deux participants dans une classe structurée. La différence avec une association classique est la suivante : chaque association est indépendante des autres, alors que tous les connecteurs d'une classe structurée partagent un contexte unique.

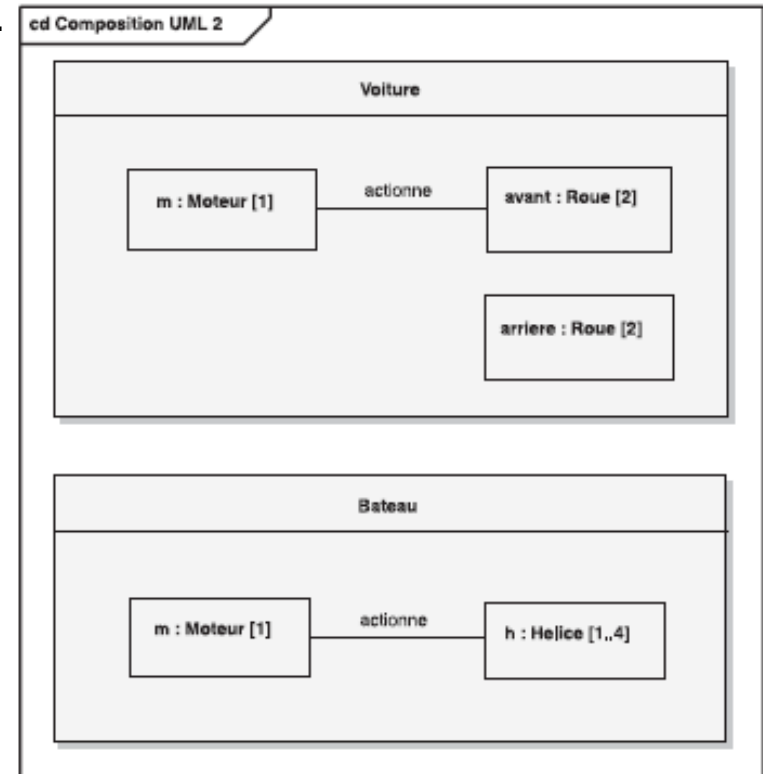


Diagramme de structure composite

NOTION DE PORT

Un port est un point d'interaction individuel entre une classe structurée et son environnement. La classe (et ses ports) peut avoir des interfaces requises et fournies pour définir son comportement visible de l'extérieur. Lorsqu'on crée une instance de classe structurée, ses ports sont créés avec elle. À sa destruction, ses ports sont détruits. On peut également connecter des ports à des participants internes ou à la spécification de l'objet dans son ensemble.

Diagramme de structure composite

