

La Notation UML

Les diagrammes dynamiques

Cedric Dumoulin

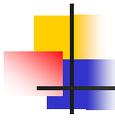
Compilation de présentations de :
Julie Vachon
Jeanine Leguy

1

Les diagrammes dynamiques

- Les diagrammes d'interactions
 - Séquences
 - Communication (uml1 == Collaboration)
- Les diagrammes d'états transitions
- Les diagrammes d'activités

2



Diagrammes d'interactions

- Les diagrammes d'interactions sont des diagrammes d'objets qui communiquent par envoi de messages. Ils mettent l'accent sur les flots de contrôle d'un objet à un autre.
 - Diagramme de séquence
 - Met en exergue le déroulement séquentiel
 - Diagramme de communication
 - Met en exergue les échanges de messages (communication)

3



Diagrammes d'interactions

Introduction

- Permet de décrire (entre autres) les cas d'utilisation
 - met en évidence les interactions entre les instances des classes (objets) du logiciel
- Un cas d'utilisation est réalisé par une collaboration
 - **Collaboration** = Ensemble d'objets qui s'échangent des messages et travaillent ensemble pour accomplir une tâche.
 - **Interaction** = Échange d'un message entre deux objets du logiciel.

4

Diagrammes d'interactions

Introduction

- UML propose deux type de diagrammes d'interaction
 - **Diagramme de séquence** (interactions projetées sur une ligne de temps)
 - **Diagramme de communication** (interactions projetées sur un diagramme d'objets)
- Pour décrire les scénarios des cas d'utilisation on privilégiera
 - **Analyse** : diagrammes de séquence (peu détaillé), d'activités, ou d'états.
 - **Conception** : diagrammes de séquence ou de communication.

5

- Les diagrammes de séquence

6

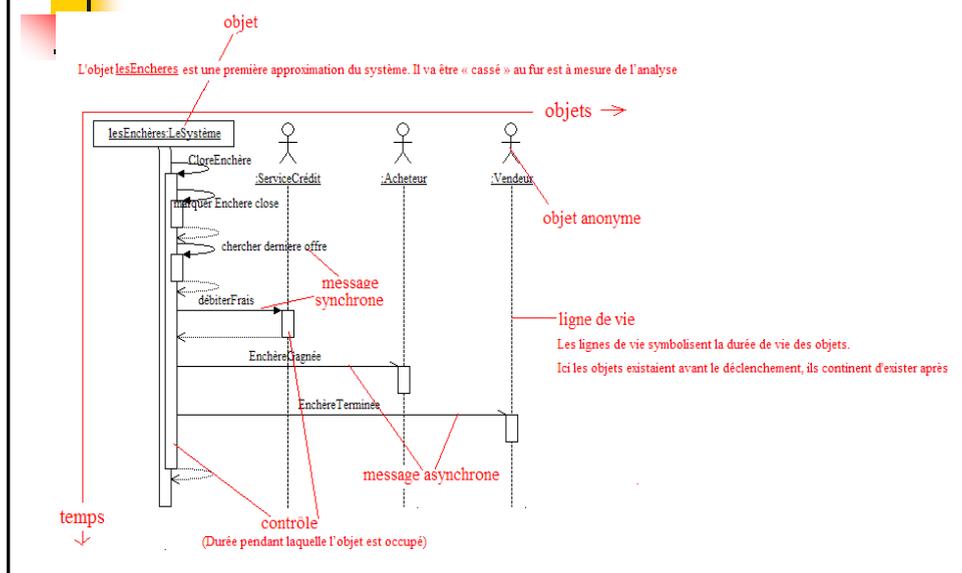
Les diagrammes de séquence

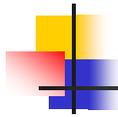
Utilisation

- Utilisés dans différents contextes:
 - description des cas d'utilisation
 - modélisation d'une opération,
 - en particulier quand cette opération met en œuvre des messages asynchrones et des instances de classes actives (pas traité dans ce cours).

7

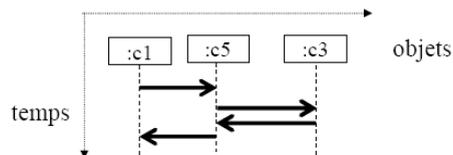
Les diagrammes de séquence





Caractéristiques

- Diagramme de séquence
 - Montre la **séquence dans le temps** des **interactions entre les objets** participant à un scénario.
- Un diagramme de séquence a **deux dimensions**
 - Dimension verticale : **le temps**
 - L'ordre d'envoi d'un message est déterminé par sa position sur l'axe vertical du diagramme ; le temps s'écoule « de haut en bas »
 - Dimension horizontale : **les objets (et les acteurs)**
 - L'ordre de disposition des objets sur l'axe horizontal est sans importance

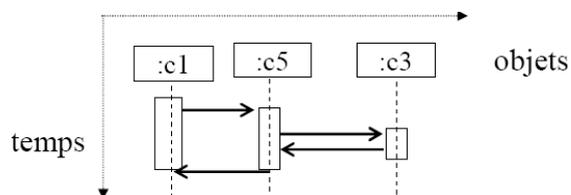


9



Caractéristiques Éléments (graphiques) de base

- Les **objets** qui interagissent dans le scénario
- Représentation graphique de la **ligne de vie** de chaque objet et de ses activations
- Les différents **types de messages envoyés**
 - (simple, synchrone, asynchrone)
- Les indications de contrôle
 - (branchement conditionnel et itération, création et destruction d'objets, délais de transmission, contraintes temporelles)



10

Les diagrammes de séquence

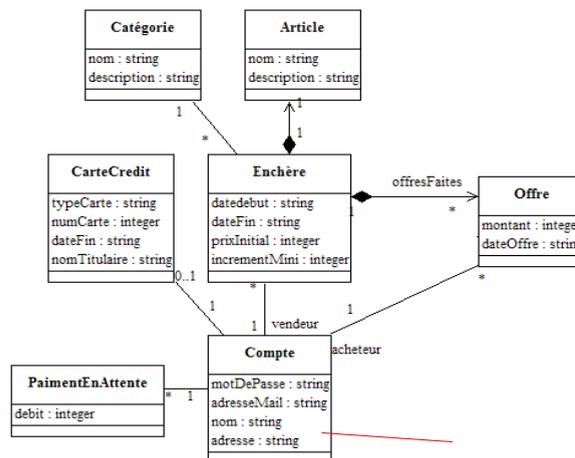
Utilisation

- Les diagrammes de séquence **permettent de décrire des scénarii.**
 - Ils sont **axés sur les classes et leurs interactions**
 - Ils permettent de découvrir de nouvelles classes
- **Méthodologie:**
 - On **part de la description textuelle d'un scénario** (ici le scénario nominal du UC CloreEnchere)
 - **On raffine tant que nécessaire**
 - en « cassant » des objets
 - en détaillant un message
 - Rappel: On dispose d'une description(incomplète) des objets métier

11

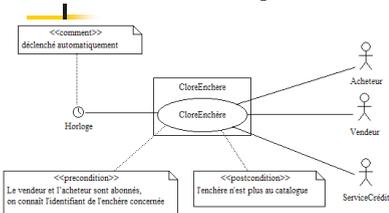
Exemple : CloreEnchere

Diagramme de classes avant



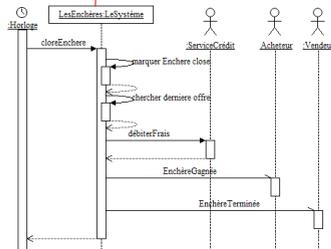
Abonné en tant objet₁₂

Exemple : CloreEnchere

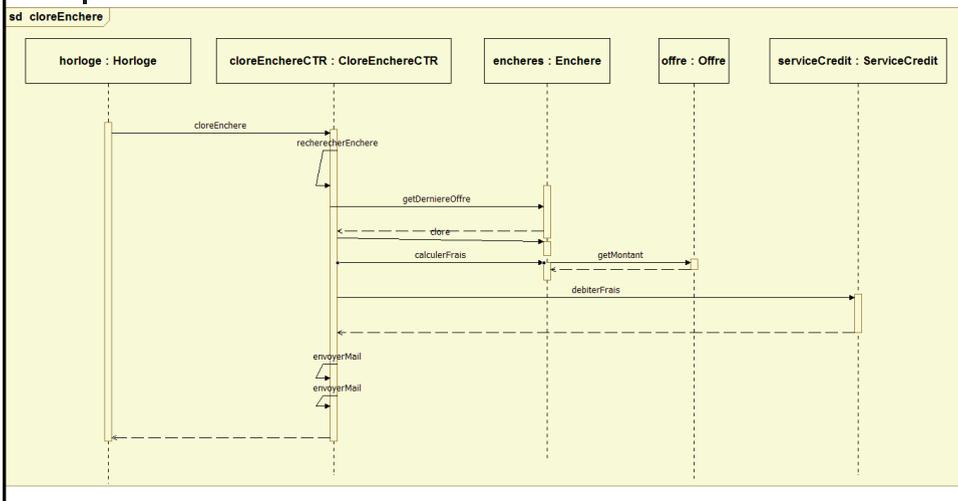


- .Déclenchement à un moment défini.
- .Le système marque l'enchère « close » => aucune offre la concernant ne peut plus être acceptée.
- .Le système récupère la dernière offre
- .Le système soumet au service de crédit le paiement des charges
- .Le système envoie un message à l'acheteur pour lui dire qu'il a gagné l'enchère
- .Le système envoie un message au vendeur pour lui dire que l'enchère est close, à quel prix et que son compte va être débité.
- .Fin du scénario

Prend en charge le UC CloreEnchere, nommons le CloreEnchereCTR

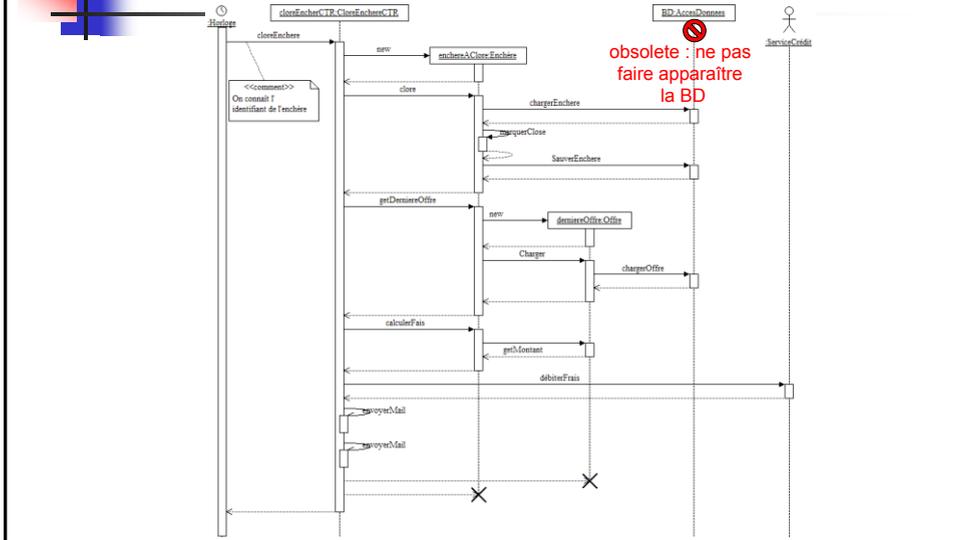


Exemple : CloreEnchere Diagrammes de séquence



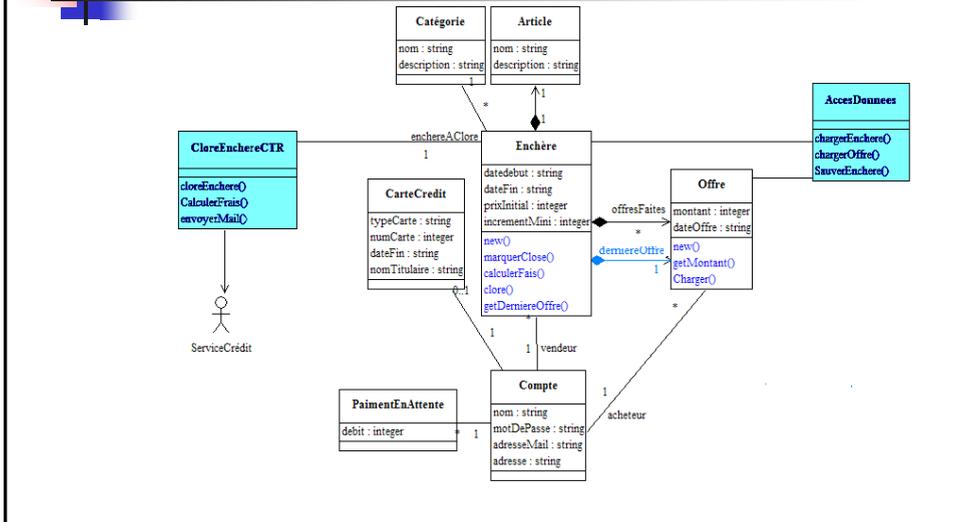
Exemple : CloreEnchere

Diagrammes de séquence (BD)



Exemple : CloreEnchere

Diagramme de classes après

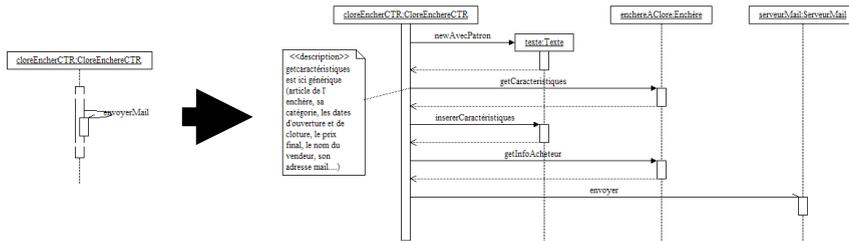


Exemple : CloreEnchere

Raffinement du message

- Raffinement du message « envoyerMail »

- scénario:
 - **cloreEnchereCTR** crée un texte formaté à trous
 - **cloreEnchereCTR** ajoute les infos spécifiques à l'enchère
 - **cloreEnchereCTR** recherche l'adresse mail du destinataire
 - **cloreEnchereCTR** envoie le texte au destinataire via le serveur de mail

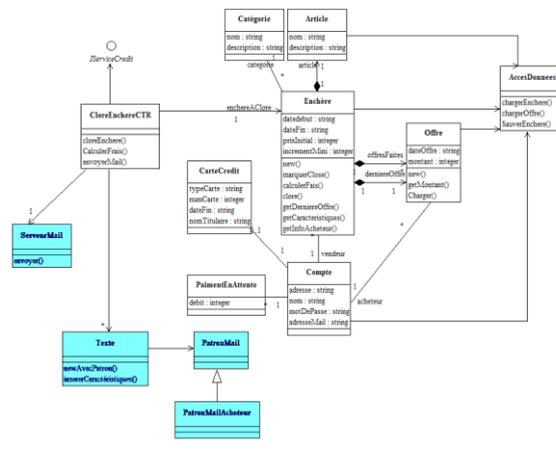


- Approche plus objet:

- Créer une classe se chargeant de la création et de l'envoi du message (a faire)

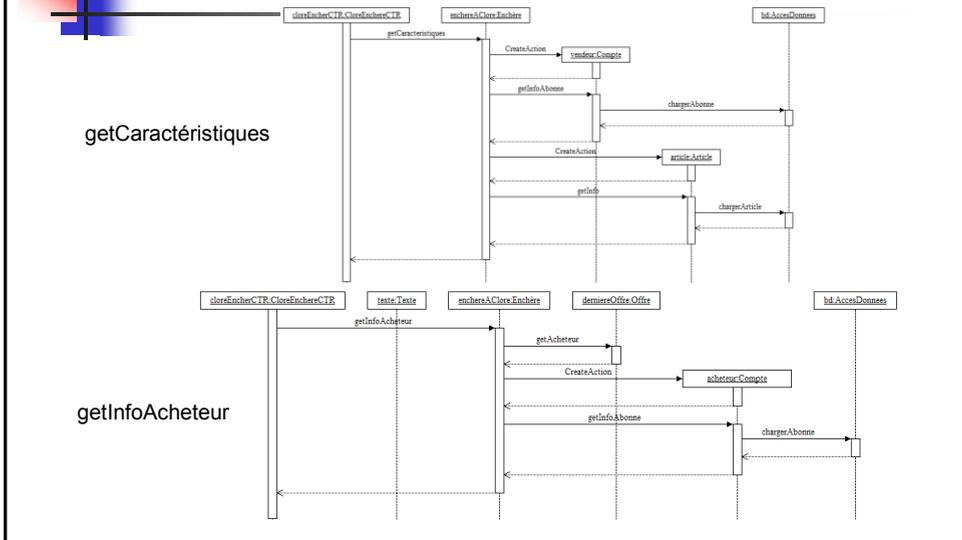
Exemple : CloreEnchere

Diagramme de classes après



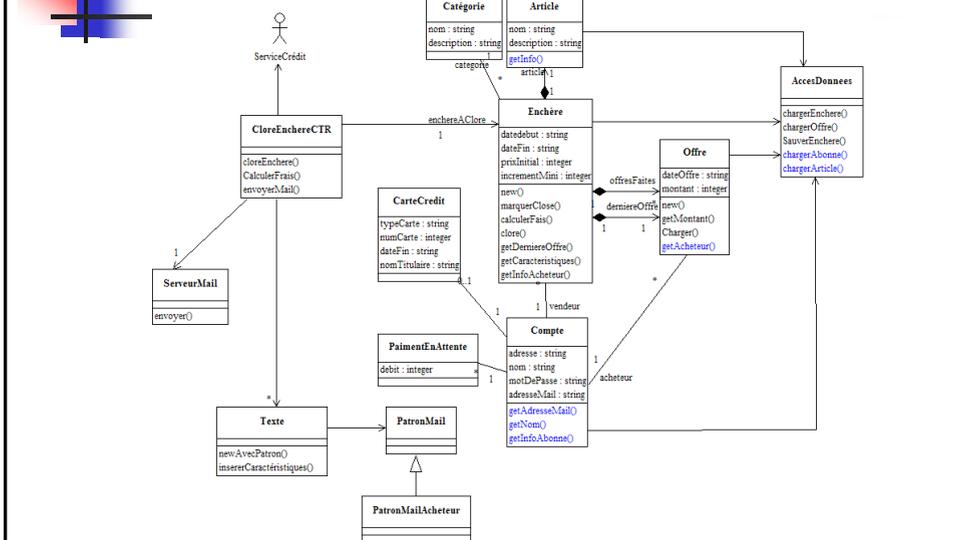
Exemple : CloreEnchere

Détail de méthodes (avec BD)



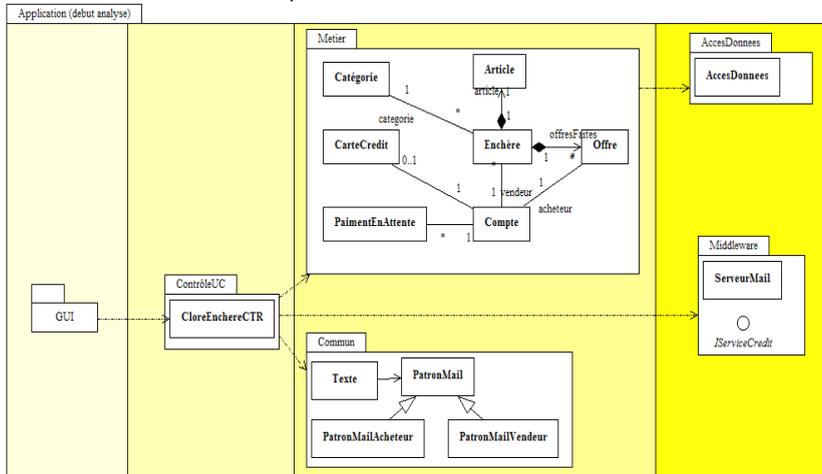
Exemple : CloreEnchere

Diagramme de classes après

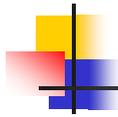


Structuration en paquetage

- Les classes se multiplient => il faut structurer!



- Diagrammes de Communication (ex uml1.x Collaboration)



Diagrammes de communication

Introduction

- Diagramme illustrant les **interactions** entre les instances d'un logiciel en mettant l'accent sur l'**organisation structurelle** des objets participants (et non pas sur l'ordre temporel des messages)
- Diagramme généralement utilisé pour
 - Décrire en détail le déroulement d'un cas d'utilisation, *i.e.*, **décrire la conception d'un scénario d'un cas d'utilisation communication**
 - Ensemble d'instances qui interagissent ensemble pour réaliser une tâche commune
 - Ensemble de liens (pertinents) existant entre ces instances

23



Diagrammes de communication

- Un diagrammes de communication est un diagrammes d'objets communicants.
- Il est composé de :
 - objets
 - de liens entre objets avec message <<stéréotypé>> et typé
 - Notes, contraintes,...
 - Les messages peuvent être numérotés.

Diagrammes de communication

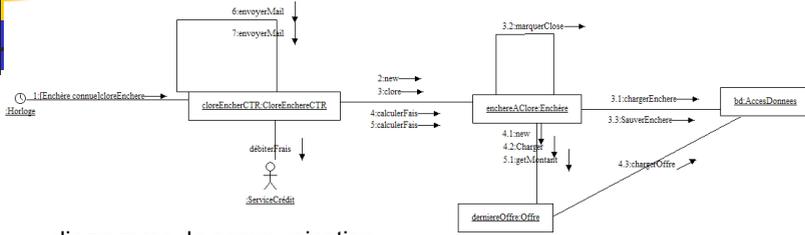
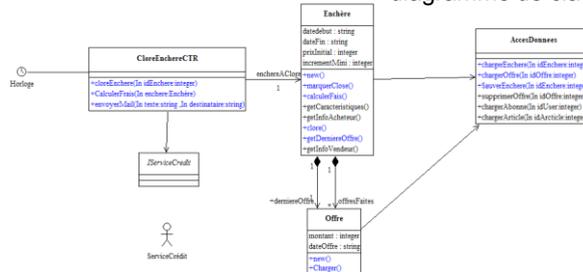


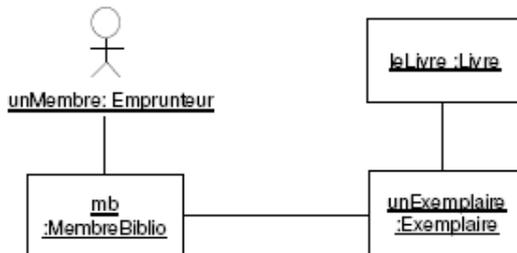
diagramme de communication

diagramme de classes



Communication

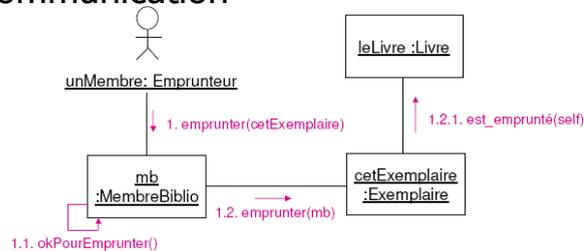
- Deux parties : statique et dynamique
- Partie statique
 - Illustre les instance (des classes) et les liens (instances d'associations) impliqués dans la réalisation d'une communication donnée.



Communication

■ Partie dynamique

- Illustre les **interactions** (les messages envoyés par les instances) nécessaires pour réaliser la tâche décrite par la communication



27

Communication

Partie statique du diagramme

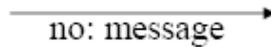
- (*i.e.*, sans interactions)
- Éléments composant une communication
 - Objets : instance d'une classe
 - Rectangle avec étiquette de la forme nomObjet:nomClasse ou :nomClasse
 - Uniquement les instances pertinentes, *i.e.*, interagissant dans le cas d'utilisation ou le scénario
 - Liens : instances d'associations du diagramme de classes
 - Uniquement les instances d'associations pertinentes pour la communication
 - Acteurs : acteurs participant au cas d'utilisation ou scénario
 - L'acteur initiant un cas d'utilisation est appelé initiateur

28

Communication

Partie dynamique du diagramme

- Ajout des messages sur les liens



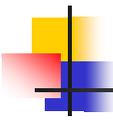
- Indique les messages à côté des liens appropriés sur le diagramme de communication
 - La flèche est issue de l'émetteur et pointe vers le destinataire
 - L'association correspondante dans le diagramme de classes doit être navigable dans la même direction
 - Le destinataire doit pouvoir comprendre le message (opération appropriée ?)

29

Représentation des interactions

- Numéro du message : en mode procédural, lorsqu'un objet O reçoit un message, le numéro de ce message est utilisé comme préfixe pour tous les messages envoyés par O par la suite... jusqu'à ce que O réponde à ce message.
- En général, les messages de retour n'apparaissent pas explicitement dans les diagrammes de communication
- Types de messages
 - Synchrones 
 - Asynchrones 
- Aussi : garde, itération, etc.

30

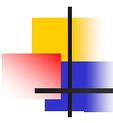


Représentation des interactions

Remarques

- Comme le diagramme de séquence, le diagramme de communication permet de découvrir de nouvelles propriétés aux classes. Il peut aider à compléter et développer le diagramme de classes
 - Ajout d'opérations, d'associations, etc.
- Le diagramme de classes doit être mis à jour pour demeurer « correct » et « cohérent » avec les diagrammes de séquence et de communication

31

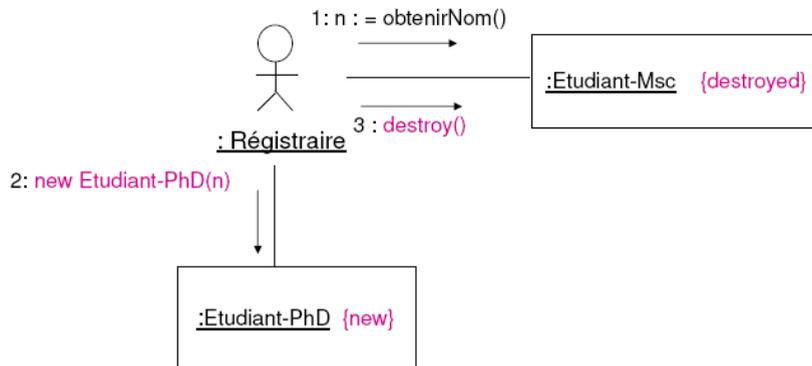


Création et destruction dynamiques

- Ajout d'une contrainte (`{new}` ou `{destroyed}`) après l'étiquette dans le rectangle représentant l'instance
 - Si au cours des interactions représentées par le diagramme de communication, une instance est créée puis détruite, on utilise la contrainte `{transient}`
- Utilisation des messages de création et de destruction d'instance : `new`, `destroy`

32

Création et destruction dynamiques Exemple



33

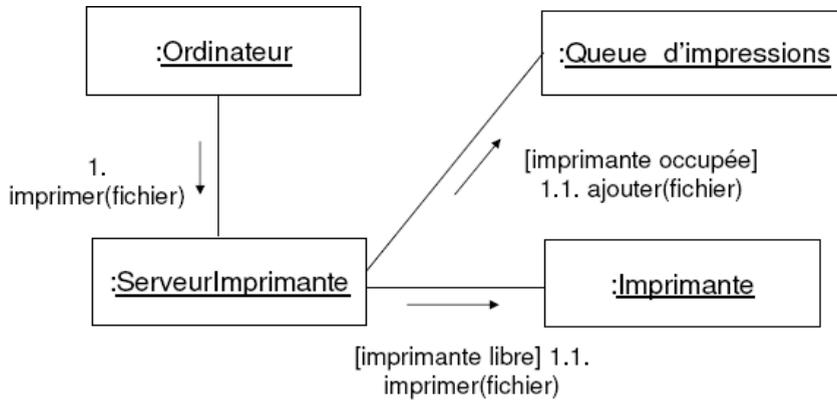
Notation utilisée

- On peut utiliser (comme pour les diagrammes de séquence)
 - Messages avec garde (condition)
 - Ex. :
[i=0] 1: a(), [i=1] 1: b()
 - Messages avec clause itérative
 - Ex. :
1: *[i:=1..n] a()

34

Notation utilisée

Exemple – Message avec garde



35

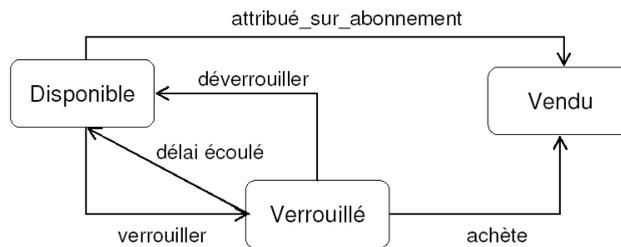
- Diagrammes d'états transitions

36

Diagrammes d'états transitions

Introduction

- Permet de **décrire le comportement d'une instance d'une classe** (ou d'un ensemble d'objets) en fonction des messages reçus
- Peut également être utilisé pour **décrire l'exécution d'une opération**, d'un cas d'utilisation, etc.



- **Diagramme d'état d'un billet de spectacle à vendre sur Internet**

37

Principaux concepts

- État

Nom_état

- Transition

événement / action →

- Marqueur d'état initial



38

Principaux concepts

État

- Décrit un moment de la vie d'une instance Nom_état
- Une instance ne se trouve que dans un seul état à la fois
- Un état est généralement décrit par un nom
- Toutes les instances d'une classe qui se trouvent dans un même état réagissent de façon identique aux événements
- Les instances qui se trouvent dans un état donné, à un moment donné
 - Ont des valeurs d'attributs similaires
 - Attendent un événement particulier
 - Exécutent une activité particulière
- En UML, il existe différents types d'état : simple, concurrent,

39

Principaux concepts

Transition

- Une transition (sortante) définit la réponse d'une instance, dans un état donné, à un événement donné
- Les transitions sont étiquetées par un événement et (optionnellement) par une action

événement / action

- **Événement** : tout ce qui survient et peut affecter une instance (élément **déclencheur de la transition**)
- **Action** : **opération** réalisée lorsqu'une transition est réalisée

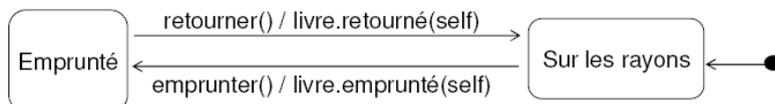
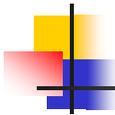


Diagramme d'état pour la classe Exemple_de_livre

40

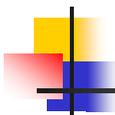


Principaux concepts

- Types d'événements
 - Un événement peut être paramétré

Type d'événement	Description	Syntaxe
Appel / Signal	Réception d'un message synchrone (pour lequel l'émetteur attend une réponse), invocation d'une opération	<code>op(p1:type, p2:type, ...)</code>
Changement	Changement de valeur d'une condition <code>when(exp)</code> booléenne, satisfaction soudaine de cette condition, condition fausse qui devient vraie	<code>when(exp)</code>
Temps	Temps absolu atteint ou passage d'un certain <code>after(time)</code> intervalle de temps, peut signaler le temps écoulé depuis l'entrée dans un état donné	<code>after(time)</code>

41



Principaux concepts

- Types d'actions
 - Les actions peuvent prendre des arguments

Type d'action	Description	Syntaxe
Affectation	Assigne une valeur à une variable	<code>cible := expression</code>
Appel / Envoi de message	Invocation (synchrone) d'une opération d'un objet, peut retourner une valeur	<code>opname(arg1, arg2, ...)</code> <code>object.opname(arg1, arg2, ...)</code>
Création	Création d'une nouvelle instance	<code>new Cname(arg1, arg2, ...)</code>
Destruction	Destruction d'une instance	<code>object.destroy()</code>
Divers	Action décrite dans un autre langage	<code>[description]</code>
Séquence	Séquence d'action	<code>action1; action2; ...</code>

42

Actions d'entrée et de sortie

- Certaines actions peuvent être rattachées à un état au lieu d'une transition
 - Action d'entrée : exécutée chaque fois qu'on entre dans l'état
 - Notation : `entry / action`
 - Action de sortie : exécutée chaque fois qu'on sort de l'état
 - Notation : `exit / action`

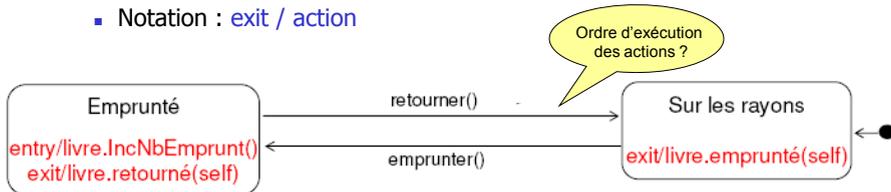
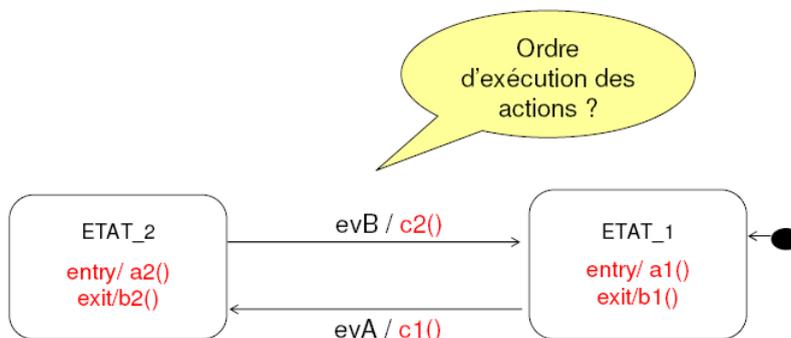


Diagramme d'état d'un exemplaire de livre

43

Actions d'entrée et de sortie



44

Gardes

- Une transition peut être conditionnelle à l'évaluation d'une garde
- Si la garde est vraie, la transition est réalisée
- Si la garde est fausse, la transition n'a pas lieu
- Notation : **événement [garde]**

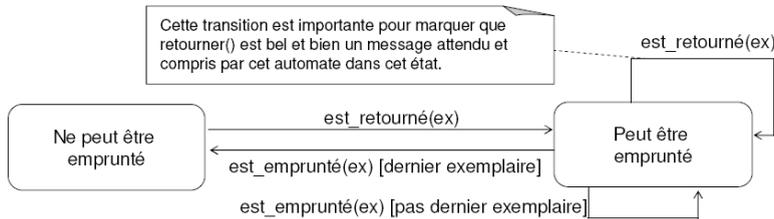


Diagramme d'état d'un livre

45

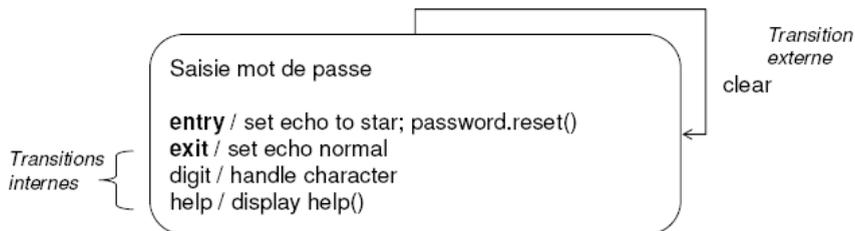
Gardes

- Garde
 - Expression conditionnelle
 - Évaluée uniquement quand l'événement est déclenché
 - Peut contenir des attributs de l'instance ou des paramètres de l'événement associé
- Lorsqu'un même événement est associé à plusieurs transitions, une garde (condition) peut être ajoutée pour préciser le contexte et **déterminer la transition à effectuer**
- Les gardes associées à un même événement sur les transitions sortantes d'un état donné, doivent être **mutuellement exclusives**

46

Types de transitions

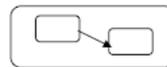
- **Transition externe** : transition standard qui engendre un changement d'état et toutes les actions correspondantes (actions d'entrée et de sortie, ainsi que celles de la transition)
- **Transition interne** : transition qui n'engendre pas de changement d'état et ne déclenche que les actions associées à cette transition (pas les actions d'entrée et de sortie, cf. figure ci-dessous)
- **Transition de complétion** : transition qui n'est pas activée par un événement; est implicitement activée lorsque est atteint la fin d'un flux d'activités



47

État composite

- État « spécialisé » composé de plusieurs sous-états
 - Sous-états **séquentiels**
 - Sous-états **concurrents**



48

État composite séquentiel

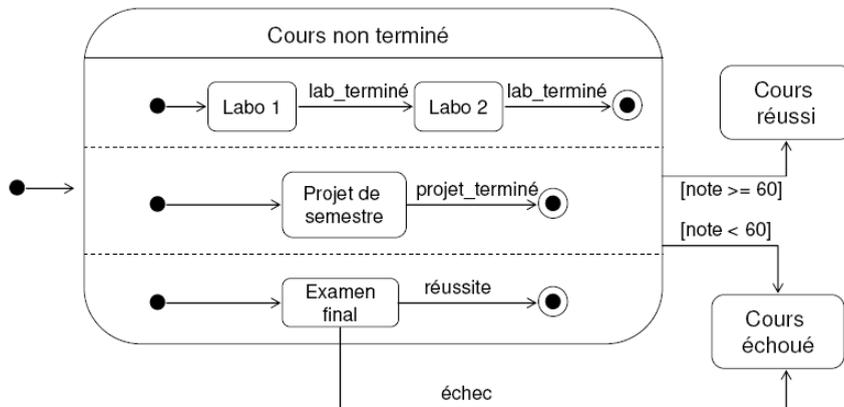


État composite

- Lorsqu'un état composite est activé, un de ses sous états est nécessairement activé
- Entrer et sortir d'un état composite
 - Une transition entrant dans un état composite est implicitement conduite vers son état initial
 - Une transition vers l'état final d'un état composite active implicitement une transition de complétion sortant de l'état composite
- Lorsqu'une transition entre/sort en traversant un ou plusieurs états composites imbriqués, toutes les actions d'entrées/sortie sont exécutées
 - Actions de sortie de l'état le plus interne en premier
 - Actions d'entrées de l'état le plus externe en premier

État composite (concurrent)

Diagramme d'état d'un cours

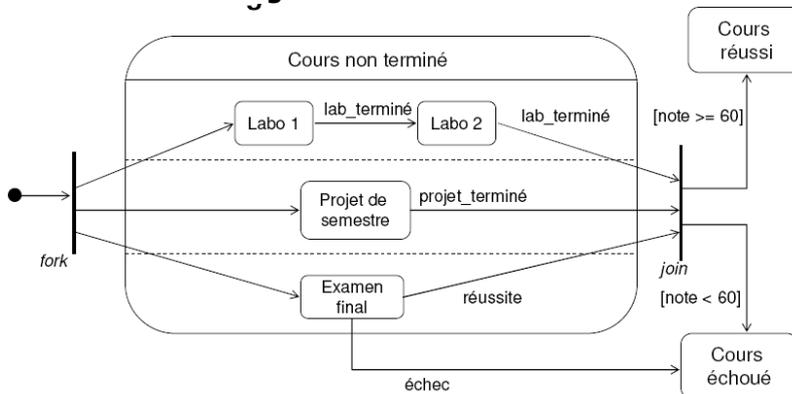


51

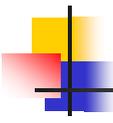
État composite concurrent

- Autre notation possible

Diagramme d'état d'un cours



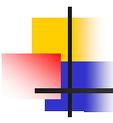
52



Représentation des comportements valides

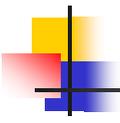
- Un diagramme d'états doit représenter **tous et seulement** les états et les transitions **valides** des instances d'une classe
- Si un diagramme d'état, pour un état donné E, n'a pas de transition étiquetée par un événement X, cela signifie que l'instance n'a pas à réagir à cet événement dans cet état.

53



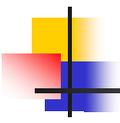
Diagrammes d'états transitions

- Ils sont avant tout centrés sur les états pris par un ensemble d'objets.
- On les utilise pour décrire le comportement d'un ensemble d'objets c'est-à-dire décrire l'enchaînement des états par lesquels ces objets peuvent passer durant leur durée de vie en réponse à des événements ou actions
- Associés à une classe, ils permettent de visualiser les états possibles que peut prendre une instance de cette classe.



- Les diagrammes d'activités

61



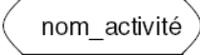
Diagrammes d'activités

Introduction

- Utilisé pour **décrire les séquences d'activités**
 - Un cas d'utilisation (phase d'analyse)
 - Un algorithme particulier (phase de conception)
- Permet de décrire les activités d'un cas d'utilisation en restant à un haut niveau d'abstraction
- Permet d'exprimer des activités séquentielles et concurrentes

62

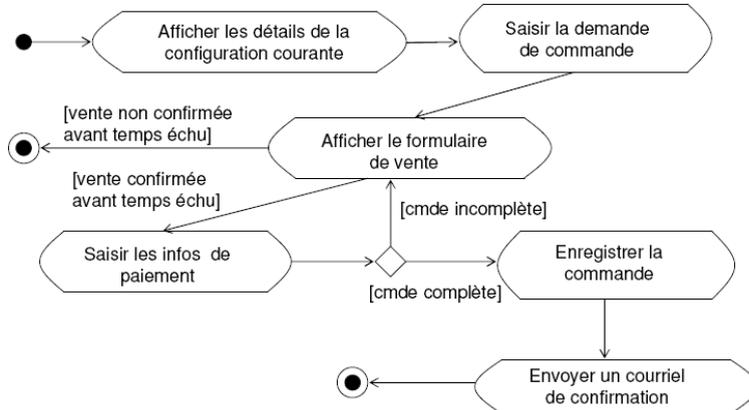
Principaux éléments de notation

- Activité  nom_activité
- Transition 
- Marqueur d'état initial 
- Barre de synchronisation 
- Branchement 
- Condition 

63

Diagrammes d'activités Exemples

Système de vente en ligne
Cas d'utilisation : commander ordinateur (à configurer)

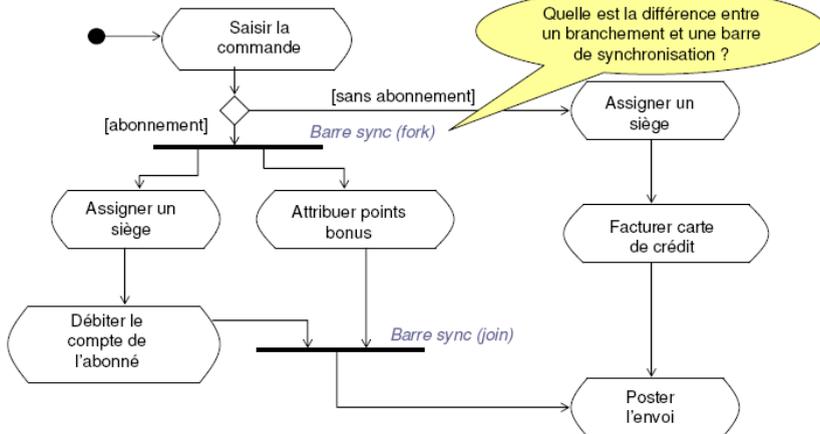


64

Diagrammes d'activités

Exemples

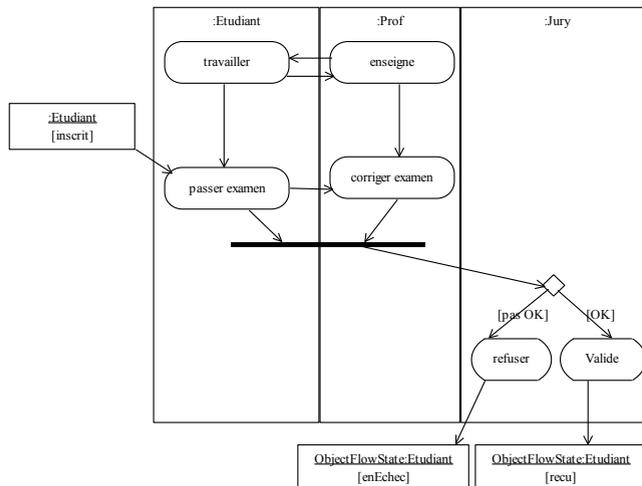
Cas d'utilisation: Traiter une commande de billet de théâtre



65

Diagrammes d'activités

Couloirs (swimlane)



Diagrammes d'activités Desc. de processus (workflow)

- Un processus peut impliquer plusieurs personnes, il peut également se dérouler sur une période de temps plus ou moins longue qui nécessite de connaître des états intermédiaires des objets impliqués.
- En ingénierie du besoin on modélise des workflows encore appelés processus métier qui correspondent au savoir faire de la structure qu'on modélise.
- La modélisation des processus métier permet de trouver les cas d'utilisation (métier) candidats à être des cas d'utilisation de l'application logicielle.
 - → Les diagrammes d'activités vont permettre de visualiser l'enchaînement des cas d'utilisation.
- Un cas d'utilisation ne peut pas être "suspendu" puis repris.
- Un diagramme d'activités peut "attendre" entre activités.

Diagrammes d'activités Exemple: suivi commande client

