

Diagramme d'activités (DAC)

Dr. I. Sarr
UCAD/FST/DMI
Idrissa.sarr@ucad.edu.sn

Termes et concepts (1)

- Les diagrammes d'activité offrent une manière graphique et non ambiguë pour modéliser les traitements.
 - Comportement d'une méthode
 - Déroulement d'un cas d'utilisation
- En théorie, tous les mécanismes dynamiques pourraient être décrits par un diagramme d'activités, mais seuls les mécanismes complexes ou intéressants méritent d'être représentés.
- Les DACs offrent une vision proche des langages de programmation impératifs comme C++ ou Java.
 - Les états représentent des calculs
 - Il n'y a pas d'événements externes mais des attentes de fins de calculs
 - Il peut y avoir de la concurrence entre activités

2

Présenté par I. SARR

Cours d'UML

Termes et concepts (2)

- Diagramme d'interaction décrit l'interaction des objets pour effectuer un traitement, mais s'intéresse uniquement aux messages échangés.
- Diagramme d'activités décrit les traitements en s'affranchissant partiellement de la structuration de l'application.
→ **vision transversale découplée de la vision structurelle classe/composant**
- Modélise le flot de contrôle des traitements
 - Curseur d'exécution (*program counter*)
- Composants fondamentaux d'un DAC
 - Activités
 - Transitions
 - Objets

3

Présenté par I. SARR

Cours d'UML

Action

- Une action est le plus petit traitement qui puisse être exprimé en UML. Elle a une incidence sur l'état du système ou en extrait une information.
- Les actions sont des étapes discrètes à partir desquelles se construisent les comportements.
- Une action peut être, par exemple :
 - une affectation de valeur à des attributs ;
 - un accès à la valeur d'une propriété structurelle ;
 - la création/suppression d'un objet ou de pointeur ;
 - un calcul arithmétique simple ;
 - l'émission d'un signal ;
 - la réception d'un signal ;
 - ...

4

Présenté par I. SARR

Cours d'UML

Action

- Une action peut être définie en langage naturel, en pseudo code ou avec un langage de programmation
- Exemples

Ouvrir porte

Dossier.ouvrir(fichier)

i:=i+1

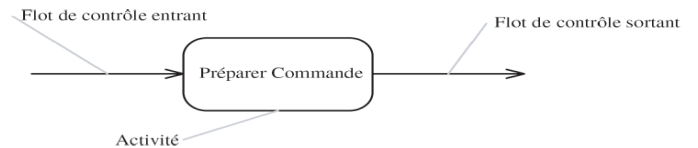
5

Présenté par I. SARR

Cours d'UML

Activités (1)

- Une activité est un comportement qui décrit un séquençage organisé d'actions.
 - Actions de base, boucles, et conditionnelles, appel de méthodes, ...
- Le flot d'exécution est modélisé par des noeuds reliés par des arcs (transitions).
- Le flot de contrôle reste dans l'activité jusqu'à la fin de l'exécution des traitements



6

Présenté par I. SARR

Cours d'UML

Activités (2)

- On peut définir des variables locales à une activité et manipuler les variables accessibles depuis le contexte de l'activité.
- Les activités peuvent être imbriquées hiérarchiquement : on parle alors d'activités composites.
- Il existe trois familles de noeuds d'activités
 - les noeuds exécutables (noeud d'exécution);
 - les noeuds objets ;
 - et les noeuds de contrôle

7

Présenté par I. SARR

Cours d'UML

Transition

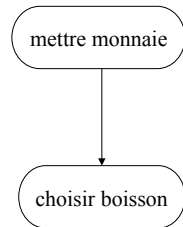
- Les transitions matérialisent le passage d'une activité vers une autre. Elles sont représentées par des flèches en traits pleins qui connectent les activités entre elles.
- Les transitions peuvent être **automatiques**
 - Elles sont déclenchées dès que l'activité source est terminée et provoquent automatiquement et immédiatement le début de la prochaine activité à déclencher (l'activité cible).
 - Elles ne possèdent pas de signature d'événement
- Les transitions peuvent inclure des actions telles que définies dans les automates d'états (DET)
- Les transitions peuvent prendre des conditions de garde booléennes, mutuellement exclusives appelées décisions

8

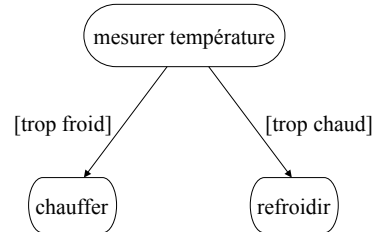
Présenté par I. SARR

Cours d'UML

Transition :Exemples



Transition automatique



Transition conditionnelle

9

Présenté par I. SARR

Cours d'UML

Nœuds exécutables (1)

- Un nœud exécutable est un nœud d'activité qu'on peut exécuter (i.e. une activité).
- 2 types de nœuds
 - **Nœud d'actions**
 - Unité fondamentale de fonctionnalités exécutables dans une activité.
 - L'exécution d'une action représente une transformation ou un calcul quelconque dans le système modélisé.
 - Les actions sont généralement liées à des opérations qui sont directement invoquées.
 - Un nœud d'action doit avoir au moins un arc entrant.

Saisir code

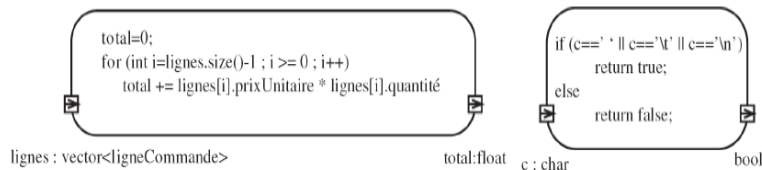
10

Présenté par I. SARR

Cours d'UML

Nœuds exécutables: Activités structurées

- Les activités structurées utilisent les structures de contrôle usuelles (conditionnelles et boucle) à travers une syntaxe.
- La syntaxe précise de ces annotations n'est pas définie dans la norme UML: on utilise du pseudo-code ou la syntaxe d'un langage de programmation.
- Dans une activité structurée, on définit les arguments d'entrée et de sorties par des flèches encadrées.



Ce type d'activité est fourni par commodité, mais son utilisation rend peu visible le cheminement du flot de contrôle.

11

Présenté par I. SARR

Cours d'UML

Nœuds de contrôle

- Un nœud de contrôle est un nœud d'activité abstrait utilisé pour coordonner les flots entre les nœuds d'une activité.
- Il existe plusieurs types de nœuds de contrôle :
 - nœud initial ;
 - nœud de fin d'activité ;
 - nœud de fin de flot ;
 - nœud de décision ;
 - nœud de fusion ;
 - nœud de bifurcation ou de débranchement ;
 - nœud d'union ou de jointure.

12

Présenté par I. SARR

Cours d'UML

Nœuds de contrôle

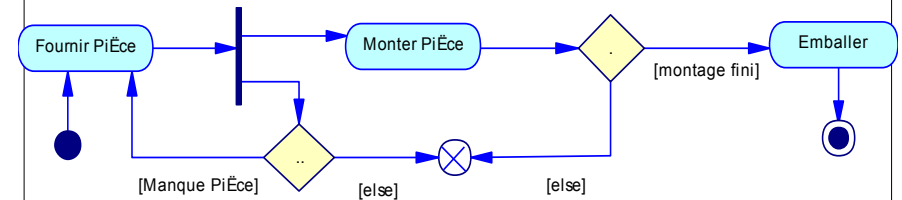
- **Noeud initial**
 - noeud de contrôle à partir duquel le flot débute lorsque l'activité enveloppante est invoquée. Il est représenté par un petit cercle plein.
- **Noeud de fin d'activité**
 - Lorsqu'un flot d'exécution atteint un noeud de fin d'activité, l'exécution de l'activité enveloppante s'achève et tout noeud ou flot actif au sein de l'activité enveloppante est abandonné.
 - Si l'activité a été invoquée par un appel synchrone, un message contenant les valeurs sortantes est transmis en retour à l'appelant.
- **Noeud de fin de flot (flow final)**
 - Lorsqu'un flot d'exécution atteint un noeud de fin de flot, il se termine, mais cette fin de flot n'a aucune incidence sur les autres flots actifs de l'activité enveloppante.
- **Noeud de décision**
 - permet de faire un choix entre plusieurs flots sortants. Il possède un arc entrant et plusieurs arcs sortants. Il est représenté par un losange.

13

Présenté par I. SARR

Cours d'UML

Exemple



Les pièces nécessaires à l'emballage sont produites séquentiellement par l'activité **Fournir Pièce**.

Si une pièce est prête, elle peut être montée

Si la barre **fork** est franchie, 2 jetons de contrôle sont produits: l'un réalise l'activité monter pièce et l'autre de fournir la pièce suivante.

Quand il ne reste plus de pièce à fournir, le flot se termine.

14

Présenté par I. SARR

Cours d'UML

Nœuds de contrôle

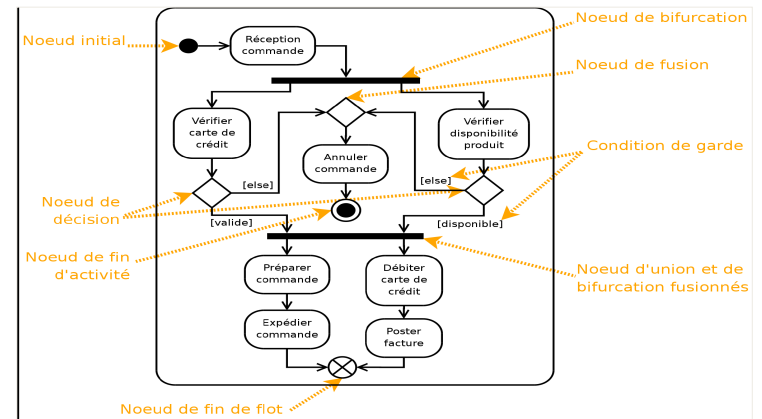
- **Noeud de fusion**
 - sert à accepter un seul flot (sortant) parmi plusieurs flots (entrants)
 - Il est représenté par un losange et il est possible de fusionner un noeud de fusion et un noeud de décision.
- **Noeud de débranchement (fourche): transition de type fork**
 - sépare un flot en plusieurs flots concurrents. Il possède donc un arc entrant et plusieurs arcs sortants. Il est représenté par un trait plein.
- **Noeud d'union ou de jointure**
 - synchronise des flots multiples. Il possède donc plusieurs arcs entrants et un seul arc sortant. Il est représenté par un trait plein.
 - Il est possible de fusionner un noeud de bifurcation et un noeud de union

15

Présenté par I. SARR

Cours d'UML

Nœuds de contrôle: illustration



16

Présenté par I. SARR

Cours d'UML

Nœuds d'objets

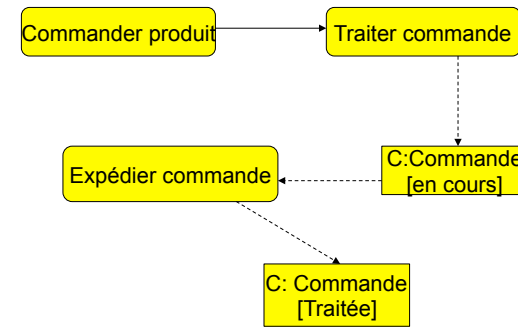
- Un nœud d'objet permet de définir un flot de données dans un diagramme d'activités.
- Il représente l'existence d'un objet généré par une action dans une activité et utilisé par d'autres actions.
- On peut montrer, le rôle de l'objet, et comment son état et ses valeurs sont modifiés

17

Présenté par I. SARR

Cours d'UML

Exemple



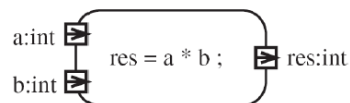
18

Présenté par I. SARR

Cours d'UML

Arguments et valeurs retournées

- Si une activité est bien adaptée à la description d'une opération d'un classeur, il faut un moyen de spécifier les arguments et valeurs de retour de l'opération.
 - C'est un rôle que peut jouer les pins
- Un pin représente un point de connexion pour une action.
 - L'action ne peut débuter que si l'on affecte une valeur à chacun de ses pins d'entrée.
 - Les valeurs sont passées par copie.
- Quand l'activité se termine, une valeur doit être affectée à chacun des pins de sortie



19

Présenté par I. SARR

Cours d'UML

Partitions (Travées) (1)

- Utilisée lors de la modélisation des workflows des processus métier afin de diviser les états d'activités en groupe de diagrammes d'activités.
- Chaque groupe ou partition représente le département responsable des activités.
- Chaque travée possède un nom, unique dans le diagramme.
- Chaque activité appartient à une seule travée.

20

Présenté par I. SARR

Cours d'UML

Partitions (2)

- Les partitions permettent d'attribuer les activités à des éléments particuliers du modèle.
- Une partition peut elle-même être décomposée en sous-partitions.

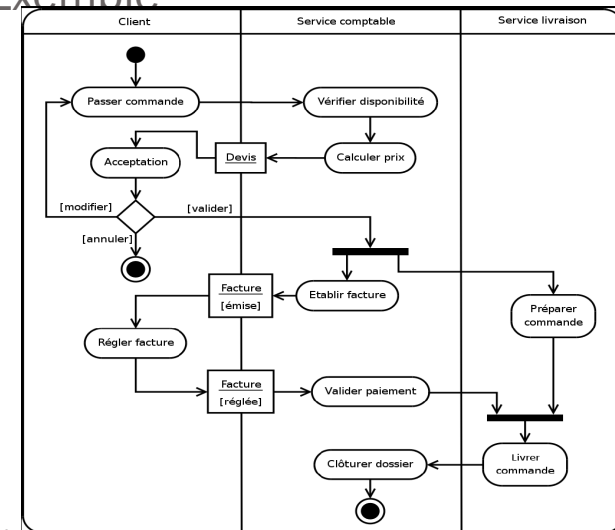
«external»	«attribute» libelléService : Service	
Client	Service comptable	Service livraison

21

Présenté par I. SARR

Cours d'UML

Exemple

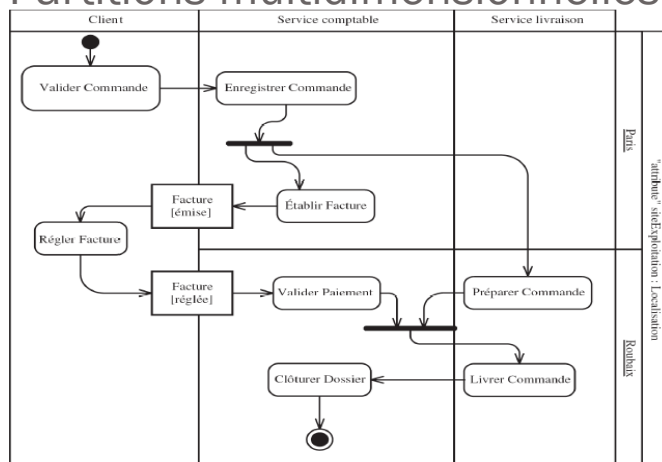


22

Présenté par I. SARR

Cours d'UML

Partitions multidimensionnelles



23

Présenté par I. SARR

Cours d'UML

Exceptions

- Concept important en programmation orientée objet
 - permettent d'interrompre un traitement quand une circonstance qui dévie de la spécification normale advienne
 - assurent une gestion plus élaborée des erreurs au cours du traitement
- Exemple:
 - Division par zéro
 - Si on a *float diviser (float dividende, float diviseur)*
 - Comment signaler à l'appelant que si le diviseur est égal à zéro, une erreur se produit?
 - Solution**
 - Recourir à une exception qui ne correspond pas à une valeur de retour de la signature normale de l'opération.
 - L'appelant a le choix de traiter l'exception ou de la laisser remonter à son propre appelant.

Attention!!! Une exception qui n'est traitée à aucun niveau correspond à une faute de programme, et induit la fin du processus

24

Présenté par I. SARR

Cours d'UML

Gestion des exceptions (1)

- Toute activité peut avoir plusieurs pins d'exceptions (représenté par un triangle).
- Une levée de l'exception se matérialise par:
 - une transition qui vise le pin d'exception
 - annotation textuelle, dans le cas d'activité structurée (*throw* par exemple)
 - Lors de la levée, l'exécution de l'activité en cours est interrompue sans générer des valeurs de sorties.
 - Par contre un jeton de données représentant l'exception est généré.
 - Le mécanisme d'exécution recherche alors un *gestionnaire* d'exception susceptible de traiter ce type d'exception ou une classe de ses classes parentes.
 - Le gestionnaire d'exception (clause catch des langages de programmation) protège une activité.

25

Présenté par I. SARR

Cours d'UML

Gestion des exceptions (2)

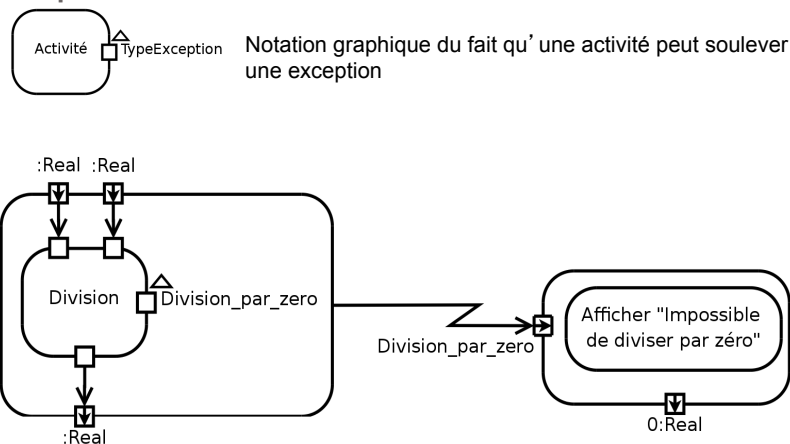
- Un gestionnaire d'exception est représenté par une activité ordinaire avec
 - un *pin d'entrée* du type de l'exception gérée
 - et *liée* avec le bloc activité qu'il protège, par un arc en zigzag
 - Il doit avoir les *même pins de sortie*, en nombre et en type, que le bloc qu'il protège
- Quand l'exécution du gestionnaire se termine, l'exécution se poursuit comme si l'activité protégée s'était terminée normalement.
 - Avec les valeurs de sortie du gestionnaire en lieu et place de celles du bloc protégé.

26

Présenté par I. SARR

Cours d'UML

Exemple



27

Présenté par I. SARR

Cours d'UML

Région interruptible et interruption

- Mécanisme analogue à la gestion des exceptions, mais moins précis
→ *gestion des interruptions* ou *régions interruptibles*.
- Adapté aux phases de modélisation conceptuelle ou de modélisation métier qui ne nécessitent pas une grande précision.
- **Objectif**
 - Représenter graphiquement un enchaînement nominal et une alternative qui interrompt le cours du traitement

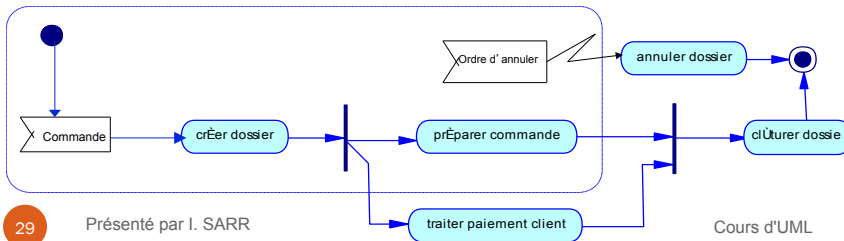
28

Présenté par I. SARR

Cours d'UML

Représentation d' une région interruptible

- Si l' évènement d' interruption survient, toutes les activités en cours dans la région interruptible sont stoppées et le flot de contrôle suit la flèche en zigzag qui quitte la région.
- Aucune contrainte sur la suite du traitement n' est imposée.
- Exemple
 - après réception d' un ordre d' annulation, et durant l' activité *annuler dossier*, l' activité *traiter paiement* peut continuer à s' exécuter. Cependant, quand l' activité *annuler dossier* a fini, on passe au nœud de fin d' activité, ce qui interrompt tous les flots de l' activité.



29

Présenté par I. SARR

Cours d'UML

Exercices pratiques

- Élaborez – une activité swap qui prend trois arguments, un tableau tab et deux indices a et b, et échange le contenu de la case tab[a] et tab[b]

30

Présenté par I. SARR

Cours d'UML

Exercices pratiques (suite)

Un site de vente en ligne propose des produits qui peuvent être placés dans un panier virtuel tandis que l' utilisateur navigue. Pour valider ses achats, il clique sur le bouton Sortir du magasin. On lui propose alors de se connecter à un compte existant, ou d' en créer un s' il n' en a pas encore.

Pour créer un nouveau compte, l' utilisateur doit fournir une adresse de messagerie, qui sert également de login, son nom et son adresse, éventuellement une adresse de livraison, et ses coordonnées bancaires. On prévoit le cas où l' adresse de messagerie est déjà associée à un compte. Si la validation de ces informations réussit, on crée un nouveau compte et l' on propose l' utilisateur de s' y connecter.

On passe ensuite à la confirmation de l' achat.

Modélisez cette procédure à l' aide d' un diagramme d' activités.

Indication: Se concentrer sur la partie authentification et création de compte le cas échéant.

31

Présenté par I. SARR

Cours d'UML