

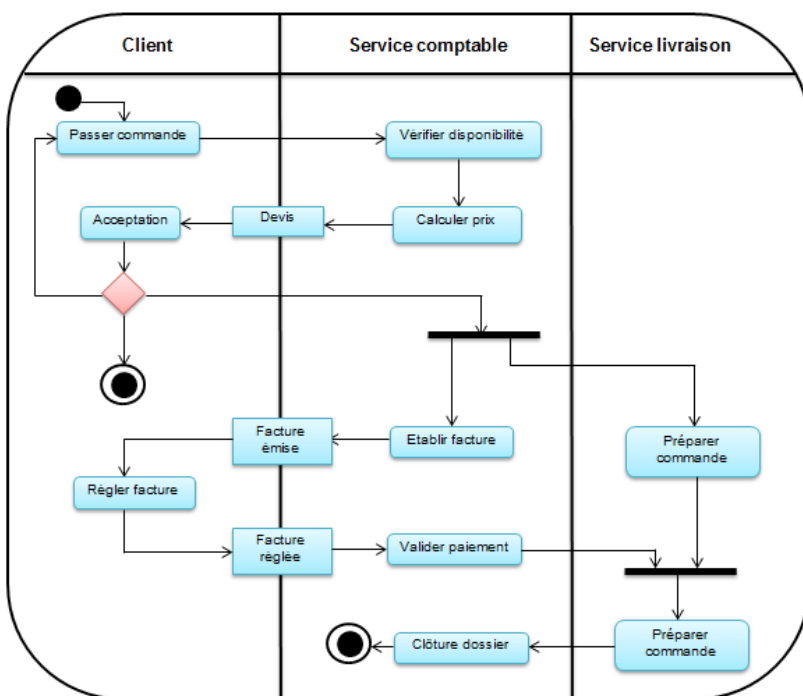
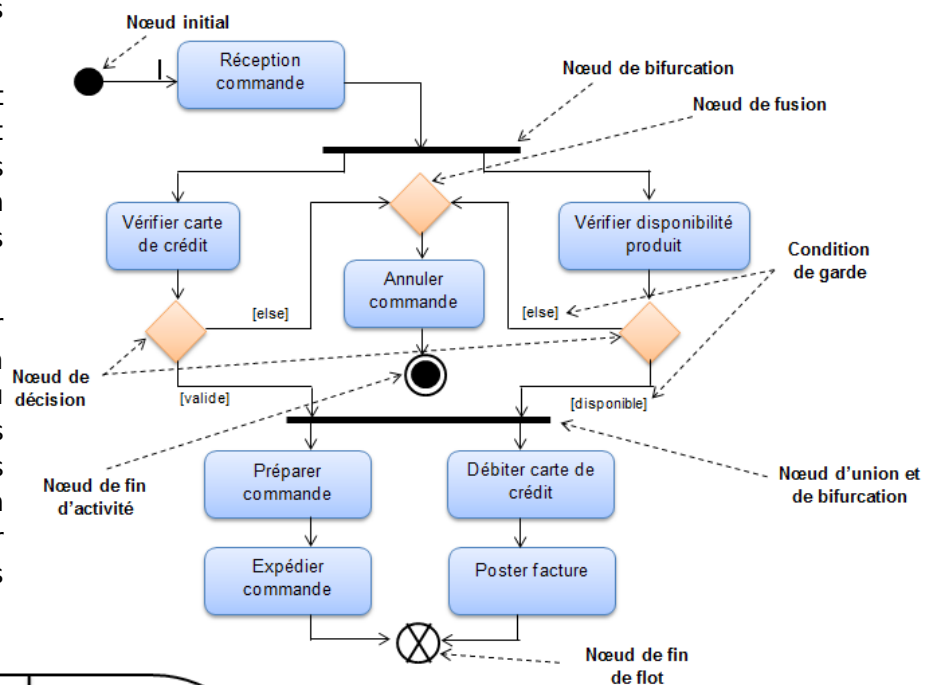
Cours n°11 : Diagramme d'activités

1. Rôle du diagramme d'activités :

Le diagramme d'activité (**Activity Diagram**) fait parti des diagrammes comportementaux. Il est utilisé pour modéliser les aspects dynamiques d'un système. Il s'agit de représenter les opérations d'un processus et leurs conséquences sur les objets (logiciels ou matériels). La modélisation peut être utilisée pour décrire le déroulement d'un cas d'utilisation ou d'une méthode.

Les diagrammes d'activité affichent le flux de travail d'un point de départ à un point d'arrivée en détaillant les nombreux chemins de décision existant dans la progression des événements contenus dans l'activité.

Ils peuvent être utilisés pour détailler des situations dans lesquelles un traitement parallèle peut avoir lieu lors de l'exécution de certaines activités. Les diagrammes d'activités sont utiles pour la modélisation métier car ils sont utilisés pour détailler les processus impliqués dans les activités métier.



Les diagrammes d'activités permettent de mettre l'accent sur les traitements. Les diagrammes d'activités ne sont pas spécifiquement rattachés à un classeur particulier.

Nous pouvons attacher un diagramme d'activités à n'importe quel élément de modélisation afin de visualiser, spécifier, construire ou documenter le comportement de cet élément.

Dans la phase de conception, les diagrammes d'activités sont particulièrement adaptés pour une plus grande description des cas d'utilisation. Plus précisément, ils viennent illustrer et consolider la description textuelle.

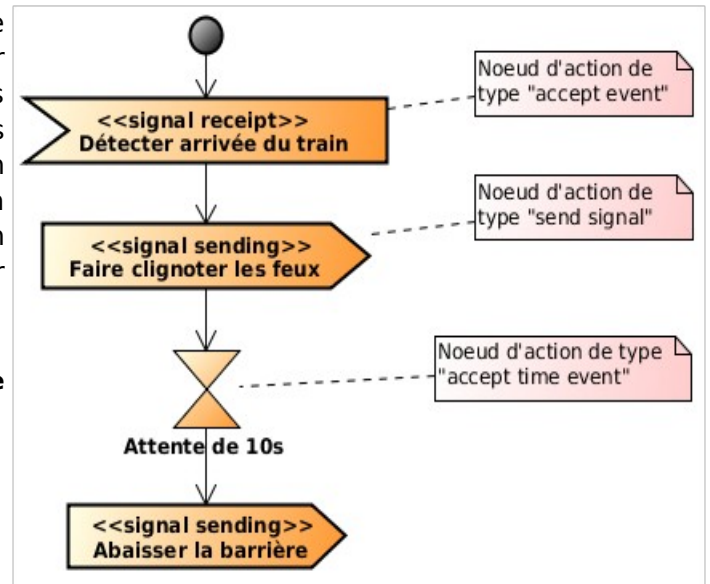
De plus, leur représentation sous forme d'organigrammes nous les rend facilement intelligibles. Nous nous concentrons ici sur les activités telles que les voient les acteurs qui

collaborent avec le système dans le cadre d'un processus métier. La modélisation du flot d'objets est souvent importante dans ce type d'utilisation des diagrammes d'activités.

2. Les actions (action):

Une action est le plus petit traitement qui puisse être exprimé en **UML**. Une action 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. La notion d'action est à rapprocher de la notion d'instruction élémentaire d'un langage de programmation (comme **C++** ou **Java**). Une action peut être, par exemple :

- une affectation de valeur à des attributs ;
- un accès à la valeur d'une propriété structurelle (attribut ou terminaison d'association) ;
- la création d'un nouvel objet ou lien ;
- un calcul arithmétique simple ;
- l'émission d'un signal ;
- la réception d'un signal ;
- une mise en attente...

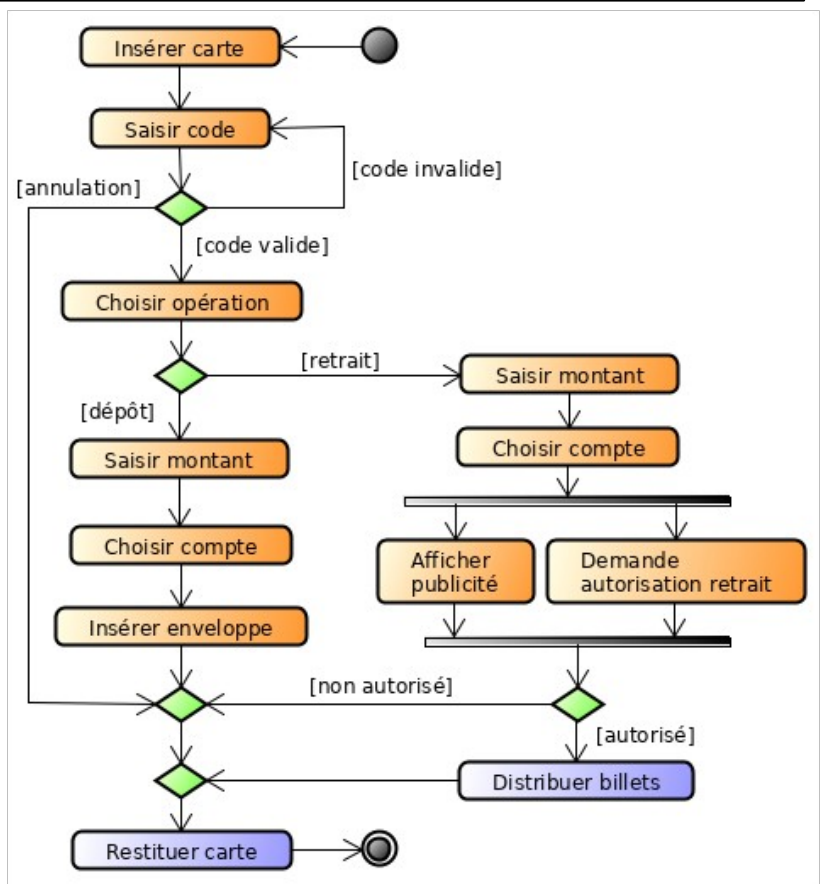


Nous distinguons ainsi graphiquement les actions associées à une communication: send signal, accept event et accept time event. Cela permet de mieux mettre en valeur les échanges entre les diagrammes de la spécification.

3. Les activités (activity) :

Une activité définit un comportement décrit par un séquençage organisé d'unités dont les éléments simples sont les actions. Le flot d'exécution est modélisé par des nœuds reliés par des arcs (**transitions**). Le flot de contrôle reste dans l'activité jusqu'à ce que les traitements soient terminés.

Ce diagramme ci-contre représente le fonctionnement d'une borne bancaire qui illustre les différentes représentations des actions. Après avoir saisi le code, deux activités sont déclenchées: nous choisissons l'opération souhaitée si le code est valide ou la carte est restituée si nous annulons l'opération. Après avoir choisi l'opération, nous trouvons deux alternatives: choisir une somme à déposer (dépôt) ou retirer de l'argent (retrait). Dans les deux cas, nous spécifions le compte par la suite. Si nous avons choisi de déposer des billets, nous devons insérer une enveloppe ce qui nous donne droit à la restitution de notre carte à la fin de l'opération. Si nous avons choisi d'effectuer un retrait de billets, nous lançons deux actions simultanées: afficher une publicité et demander en même temps une autorisation de retrait. Dans tous les cas de figure, la carte bancaire est restituée et nous arrivons à la fin du processus.



Une activité possède éventuellement des paramètres en entrée ou en sortie, ainsi que des variables locales au même titre qu'une méthode d'un objet. Une activité peut regrouper des nœuds et des arcs, c'est ce que nous appelons « **groupe d'activités** ».

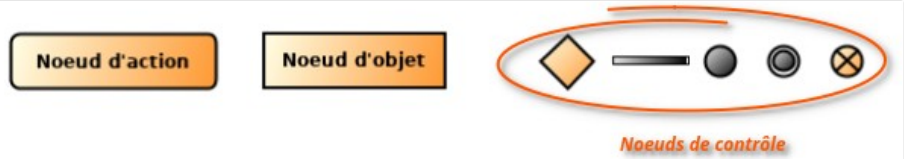
Une activité est un traitement complexe et décomposable en activités ou en actions. Une activité peut être interrompue par un événement. Elle peut faire l'objet d'une description par un diagramme d'activités. Une action est un traitement simple et non décomposable. Elle ne peut pas être interrompue.

4. Les nœuds activités :

Un nœud d'activité est un type d'élément abstrait permettant de représenter les étapes le long du flot d'une activité. Il existe trois familles de nœuds d'activités :

- **les nœuds d'exécutions (executable node en anglais) ;**
- **les nœuds objets (object node en anglais) ;**
- **et les nœuds de contrôle (control nodes en anglais).**

Représentation graphique des nœuds d'activité : De la gauche vers la droite, nous trouvons : le nœud représentant une action, qui est une variété de nœud exécutable, un nœud objet, un nœud de décision ou de fusion, un nœud de bifurcation ou d'union, un nœud initial, un nœud final et un nœud final de flot.



5. Les transitions :

Le passage d'une activité vers une autre est matérialisé par une transition. Graphiquement les transitions sont représentées par des flèches en traits pleins qui connectent les activités entre elles. 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). Contrairement aux activités, les transitions sont franchies de manière atomique, en principe sans durée perceptible.

Les transitions spécifient l'enchaînement des traitements et définissent le flot de contrôle.



6. Les nœud d'actions (action node) :

Un nœud d'action est un nœud d'activité exécutable qui constitue l'unité fondamentale de fonctionnalité exécutable 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.



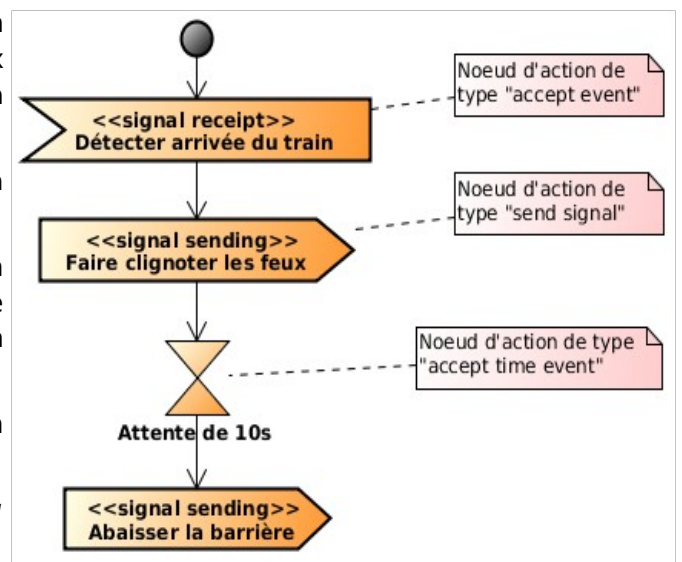
Graphiquement, un nœud d'action est représenté par un rectangle aux coins arrondis qui contient sa description textuelle.

Cette description textuelle peut aller d'un simple nom à une suite d'actions réalisées par l'activité.

UML n'impose aucune syntaxe pour cette description textuelle, nous pouvons donc utiliser une syntaxe proche de celle d'un langage de programmation particulier ou du pseudo-code.

Certaines actions de communication ont une notation spéciale comme montré ci-contre :

Dans l'exemple ci-contre, nous détectons l'arrivée du train; cette action représente l'action "accept event" c'est-à-dire que le système reçoit le signal de l'arrivée du train. Deuxièmement, "faire clignoter les feux" est une action "send signal", cela veut dire que le système envoie un signal qui est transmis à un objet cible sans attendre que ce dernier ait bien reçu ce signal. Ensuite, l'action "time event" est un événement temporel déclenché après l'écoulement d'une certaine durée. Enfin, "abaisser la barrière" est une action "send signal", un message est envoyé et transmis à la cible.



7. Les nœud d'actions (action node) :

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 (initial node en anglais) ;**
- **nœud de fin d'activité (final node en anglais) ;**
- **nœud de fin de flot (flow final en anglais) ;**
- **nœud de décision (decision node en anglais) ;**
- **nœud de fusion (merge node en anglais) ;**
- **nœud de bifurcation (fork node en anglais) ;**
- **nœud d'union (join node en anglais).**

7.1. Nœud initial : ●

Un nœud initial est un nœud de contrôle à partir duquel le flot débute lorsque l'activité enveloppante est invoquée. Une activité peut avoir plusieurs nœuds initiaux. Un nœud initial possède un arc sortant et pas d'arc entrant.

Graphiquement, un nœud initial est représenté par un petit cercle plein.

7.2. Nœud final:

Un nœud final est un nœud de contrôle possédant un ou plusieurs arcs entrants et aucun arc sortant. Il existe deux types de nœud finaux :

Nœud de fin d'activité : ● Lorsque l'un des arcs d'un nœud de fin d'activité est activé, l'exécution de l'activité enveloppante s'achève et tout nœud ou flot actif au sein de l'activité enveloppante est abandonné. Graphiquement, un nœud de fin d'activité est représenté par un cercle vide contenant un petit cercle plein.

Nœud de fin de flot : ⊗ Lorsqu'un flot d'exécution atteint un nœud de fin de flot, le flot en question est terminé, mais cette fin de flot n'a aucune incidence sur les autres flots actifs de l'activité enveloppante. Graphiquement, un nœud de fin de flot est représenté par un cercle vide barré d'un X.

7.3. Nœud de décision (decision node) : ◇

Un nœud de décision est un nœud de contrôle qui permet de faire un choix entre plusieurs flots sortants. Il possède un arc entrant et plusieurs arcs sortants. Ces derniers sont généralement accompagnés de conditions de garde pour conditionner le choix. Graphiquement, nous représentons un nœud de décision par un losange.

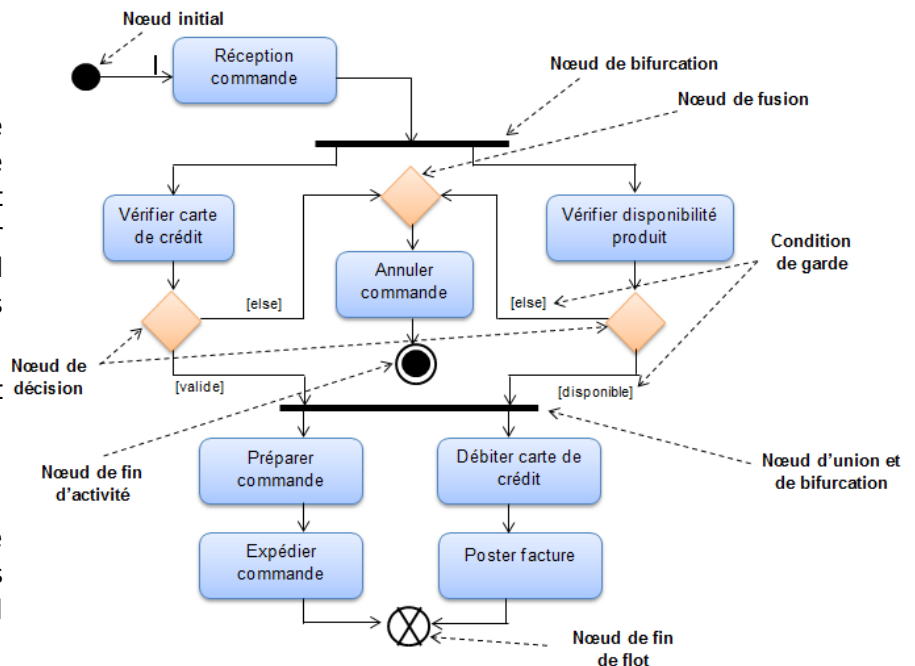
Condition de garde [entre crochets]. Il est possible d'associer une condition à une transition, qui est alors appelée condition de garde. Pour que la transition soit franchie, il faut que la condition soit remplie. C'est une expression booléenne qui doit être vraie pour que la transition soit déclenchée.

Quand le nœud de décision est atteint, si aucun arc en aval n'est franchissable (aucune condition de garde n'est vraie), c'est que le modèle est mal formé. L'utilisation d'une garde [else] est recommandée après un nœud de décision, car elle garantit un modèle bien formé. En effet, la condition de garde [else] est validée si et seulement si toutes les autres gardes des transitions ayant la même source sont fausses.

Ce nœud est aussi appelé BRANCHEMENT CONDITIONNEL : Le losange représente un branchement conditionnel qui offre des possibilités d'alternatives. Il permet d'introduire des conditions booléennes (vrai / faux) **mutuellement exclusives**, notées entre crochets (notion de **condition surveillée** ou **expression de garde**).

7.4. Nœud de fusion (merge node) : ◇

Un nœud de fusion est un nœud de contrôle qui rassemble plusieurs flots alternatifs entrants en un seul flot sortant. Il n'est pas utilisé pour synchroniser des flots concurrents (c'est le rôle du nœud d'union) mais pour n'accepter un flot parmi plusieurs. Graphiquement, nous représentons un nœud de fusion, comme un nœud de décision, par un losange.



Ce nœud est aussi appelé NŒUD DE CONVERGENCE (CONFLUENCE) : Le losange est utilisé pour modéliser un point de convergence : endroit où deux ou plusieurs chemins d'exécution **mutuellement exclusifs** se rejoignent. A partir de ce point de convergence la dernière partie du trajet est commune.

7.5. Nœud de bifurcation ou de débranchement (fork node) :

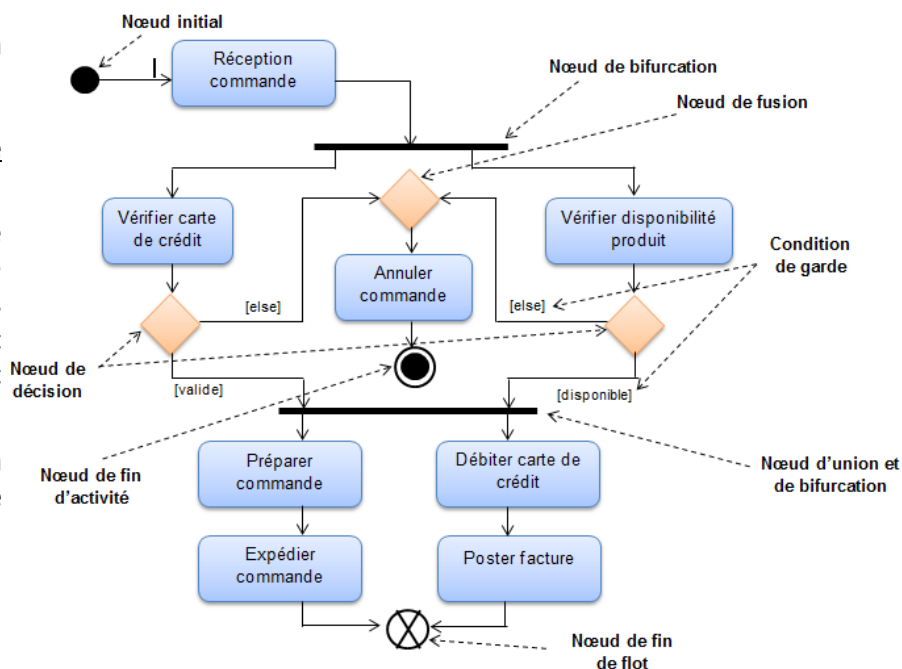
Un nœud de bifurcation, également appelé nœud de débranchement est un nœud de contrôle qui sépare un flot en plusieurs flots concurrents. Un tel nœud possède donc un arc entrant et plusieurs arcs sortants.

Graphiquement, nous représentons un nœud de bifurcation par un trait plein.

7.6. Nœud d'union ou de jointure (join node) :

Un nœud d'union, également appelé nœud de jointure est un nœud de contrôle qui synchronise des flots multiples. Un tel nœud possède donc plusieurs arcs entrants et un seul arc sortant.

Graphiquement, nous représentons un nœud d'union, comme un nœud de bifurcation, par un trait plein.



7.7. Synchronisation : Disjonction & Conjonction d'activités :

Ces traits pleins que nous utilisons dans les nœuds de bifurcation et d'union s'appellent aussi barres de synchronisation qui permettent de réaliser des traitements parallèles (**simultanéité**) et ensuite de la synchronisation pour représenter la fin de ces traitements parallèles.

Les barres de synchronisation permettent d'ouvrir et de fermer des branches (chemins) parallèles au sein d'un flot d'exécution d'une méthode ou d'un cas d'utilisation.

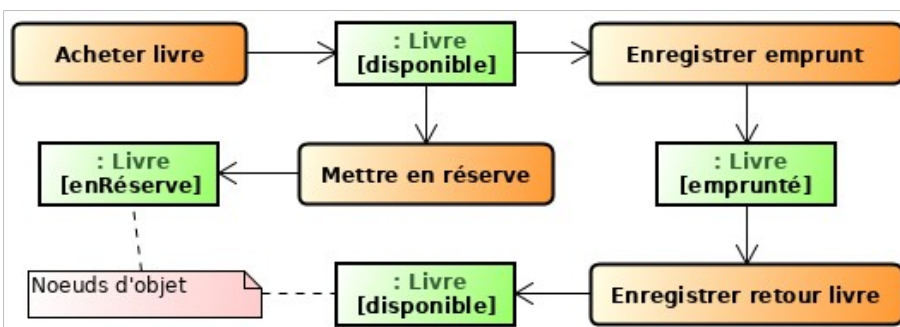
- Dans la **bifurcation**, les transitions de débranchement au départ d'une barre de synchronisation sont déclenchées simultanément, et par là même, les actions qui suivent démarrent ensembles.
- Dans l'**union**, la barre de synchronisation ne peut être franchie que lorsque toutes les transitions en entrée sur la barre sont déclenchées, et donc lorsque toutes les actions en amonts sont toutes finies (d'où la notion de synchronisation).

8. Les nœud d'objet (object node) et les flots d'objet :

Jusqu'ici, nous avons montré comment modéliser le comportement du flot de contrôle dans un diagramme d'activités. Or, les flots de données n'apparaissent pas et sont pourtant un élément essentiel des traitements (arguments des opérations, valeurs de retour...).

Les nœuds d'objet permettent de définir un flot d'objets (flot de données) dans un diagramme d'activités. Chaque nœud représente l'existence d'objet généré par une action dans une activité et utilisé par d'autres actions.

Graphiquement, un tel nœud d'objet est représenté par un rectangle dans lequel est mentionné le type de l'objet. Des arcs viennent ensuite relier ce nœud d'objet à des activités sources



et cibles. Le nom d'un état, ou d'une liste d'états, de l'objet peut être précisé entre crochets après ou sous le type de l'objet. Nous pouvons également préciser des contraintes entre accolades, soit à l'intérieur, soit en dessous du rectangle du nœud d'objet.

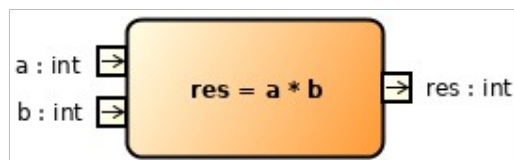
Dans le diagramme d'activités, les objets peuvent être représentés. Ils sont utilisés ou modifiés par des actions. Ils peuvent aussi initier des actions. Il arrive aussi que différentes activités manipulent un même objet qui change alors d'état selon le degré d'avancement du mécanisme comme dans le diagramme précédent.

9. Pins d'entrée/sortie :

Pour spécifier les valeurs passées en argument à une activité et les valeurs de retour, nous utilisons des nœuds d'objets appelés pins (pin en anglais) d'entrée ou de sortie. L'activité ne peut débuter que si nous affectons une valeur à chacun de ses pins d'entrée. Quand l'activité se termine, une valeur doit être affectée à chacun de ses pins de sortie.

Les valeurs sont passées par copie : une modification des valeurs d'entrée au cours du traitement de l'action n'est visible qu'à l'intérieur de l'activité.

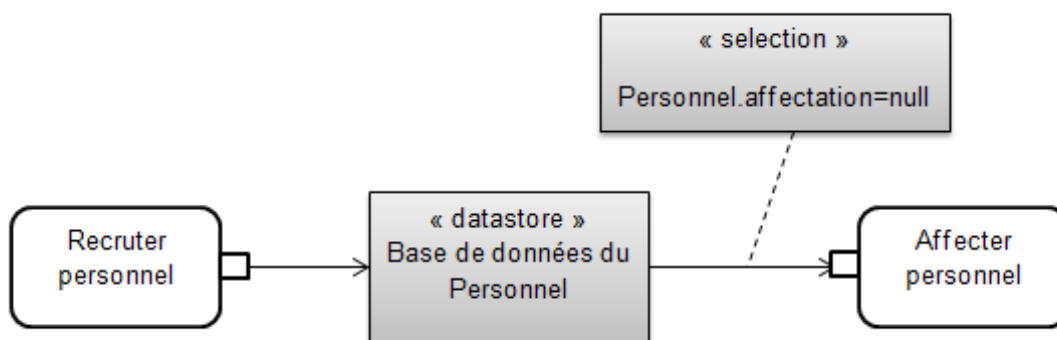
Graphiquement, un pin est représenté par un petit carré attaché à la bordure d'une activité. Il est typé et éventuellement nommé. Il peut contenir des flèches indiquant sa direction (entrée ou sortie) si l'activité ne permet pas de le déterminer de manière univoque.



10. Nœuds de stockage de données (data store node) :

Un nœud de stockage des données est un nœud tampon central particulier qui assure la persistance des données. Lorsqu'une information est sélectionnée par un flux sortant, l'information est dupliquée et ne disparaît pas du nœud de stockage des données. Lorsqu'un flux entrant véhicule une donnée déjà stockée par le nœud de stockage des données, cette dernière est écrasée par la nouvelle.

Ce nœud est représenté comme un nœud d'objet détaché avec le stéréotype « **datastore** ». Voici ci-dessous un exemple de nœud de stockage de donnée:



Après avoir recruté le personnel, il est stocké dans le nœud de stockage des données de façon permanente, appelé dans ce cas "Base de données du Personnel". Ceux qui n'ont pas été affectés sont disponibles pour être affectés par l'activité "Affecter personnel". L'étiquette "selection: personnel.affectation=null" permet de sélectionner ceux qui n'ont pas été affectés.

11. Les partitions ou couloirs d'activités (travées - swimlanes) :

Les diagrammes d'activités indiquent ce qui se passe sans préciser qui fait quoi (en terme de programmation, ils ne précisent pas quelle classe est responsable et en terme de processus métier, ils ne précisent pas quelle partie de l'organisation exécute chaque action).

Il est alors possible de diviser un diagramme d'activités en partitions ou couloirs d'activités. Chaque partition montre ainsi quelles actions sont exécutées par une classe ou par une unité organisationnelle.

Chaque travée représente les responsabilités d'une classe ou d'un service particulier. Les travées sont intéressantes car elles combinent la façon dont les diagrammes d'activités décrivent la logique des traitements avec celle des diagrammes d'interaction qui représentent les responsabilités.

Les partitions, souvent appelées couloirs ou lignes d'eau (**swimlane**) du fait de leur notation, permettent d'organiser les nœuds d'activités dans un diagramme d'activités en opérant des regroupements.

Les partitions n'ont pas de signification bien arrêtée, mais correspondent souvent à des unités d'organisation du modèle. Nous pouvons, par exemple, les utiliser pour spécifier la classe responsable de la mise en œuvre d'un ensemble de tâches. Dans ce cas, la classe en question est responsable de l'implémentation du comportement des nœuds inclus dans ladite partition.

Graphiquement, les partitions sont délimitées par des lignes continues. Il s'agit généralement de lignes verticales, mais elles peuvent être horizontales ou même courbes. Dans le cas d'un diagramme d'activités partitionné, les nœuds d'activités appartiennent forcément à une et une seule partition. Les transitions peuvent, bien entendu, traverser les frontières des partitions.

