

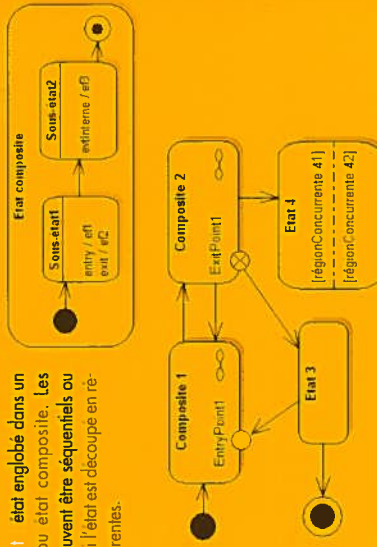
Diagramme d'états

- **État** : action ou activité qui s'exécute lorsqu'une transition se déclenche. L'exécution de l'effet est unitaire et ne permet de traiter aucun événement supplémentaire pendant son déroulement.

Trucs et astuces

- Le pseudo-état initial ne peut avoir que des flèches qui sortent, le pseudo-état final ne peut avoir que des flèches qui entrent.
- Tous les éléments d'une transition sont optionnels. Même l'événement déclencheur est optionnel : cela signifie que l'activité finie de l'état source s'est terminée normalement.
- Un état peut contenir une activité durable *do-activity*. Cette activité durable est interrompible par un événement, mais peut également se terminer d'elle-même. On parle dans ce cas d'activité finie, par opposition à une activité continue. La transition de complétion d'une activité finie, aussi appelée transition automatique, est représentée sans nom d'événement ni mot-clé.
- UML propose de déclarer les événements internes à un objet par les mots-clés *when condition*, *after*, *done*, ou *at date*.
- Un événement *comme une transition* est par convention instantané, ou en tout cas interrompible *atomique*.
- Attention, sur une transition, l'effet est toujours déclenché après l'évaluation à vrai de la condition de garde.

- Sous-état **état englobé** dans un super-état ou état composite. Les sous-états peuvent être séquentiels ou concurrents si l'état est découpé en régions concurrentes.



Trucs et astuces

- Les effets d'entrée *entry* et de sortie *exit* sont exécutés respectivement lors de toute entrée ou de toute sortie de l'état concerné.
- Les états peuvent réagir à des événements sans utiliser de flèche de transition. On parle alors de transition interne, notée à l'intérieur de l'état. Une transition interne est semblable à une transition qui reboucle sur l'état, mais sans effet secondaire *entry*, *exit*, etc.
- Utilisez le concept d'état composite ou *super-état* pour factoriser des transitions déclenchées par le même événement et amenant au même état.
- L'intérieur d'un état composite complexe peut être décrit dans un autre diagramme. Dans ce cas, une notation particulière $\circ \rightarrow$ symbolisant deux états reliés est présente en bas à droite de l'état composite non exposé.
- Les régions concurrentes permettent d'exprimer du parallélisme au sein d'un objet.
- Ne perdez pas de temps à dessiner des diagrammes d'états qui ne contiennent que deux états de type « on / off », voire un seul...
- N'utilisez pas forcément toutes les subtilités du diagramme d'états : pensez à votre lecteur !

Diagramme d'activité

Montre l'enchaînement des actions au sein d'une activité à l'aide de flots de contrôle et d'objets.



- Action : unité fondamentale de spécification comportementale qui représente un traitement ou une transformation. Les actions sont contenues dans les activités, qui fournissent leur contexte.

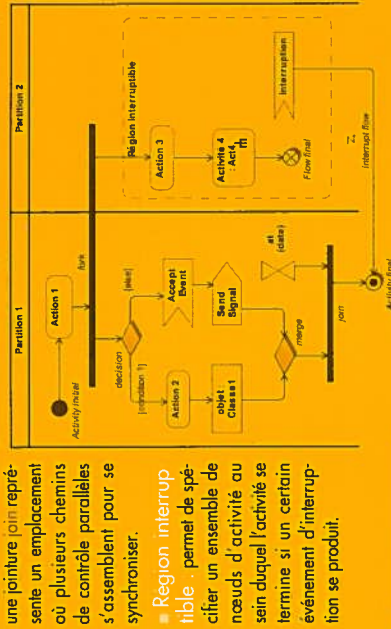
- Flot : contrôle de séquençage pendant l'exécution de nœuds d'activité. Les flots de contrôle sont de simples flèches reliant deux nœuds actions, décisions, etc... Le diagramme d'activité permet également d'utiliser des flots d'objets reliant une action et un objet consommé ou produit.

- Décision : nœud de contrôle structuré représentant un choix dynamique entre plusieurs conditions. Il possède un arc entrant et plusieurs arcs sortants. À l'inverse, une fusion *merge* représente un emplacement où plusieurs chemins de contrôle alternatifs s'assemblent.

Trucs et astuces

- Une activité représente l'exécution d'un calcul complexe, alors qu'une action est un comportement élémentaire.
- Les activités servent à modéliser à la fois le comportement séquentiel et simultané.
- Une action est activée dès que tous ses flots d'entrée possèdent un jeton, s'exécute de façon unitaire et met à disposition un jeton sur chaque flot de sortie quand elle est terminée *semantique inspirée des réseaux de Petri*.

- **Debranchement** *fork* : nœud de contrôle structuré représentant le démarrage d'actions concurrentes. Il possède un arc entrant et plusieurs arcs sortants. À l'inverse, une jonction *join* représente un emplacement où plusieurs chemins de contrôle parallèles s'assemblent pour se synchroniser.



Trucs et astuces

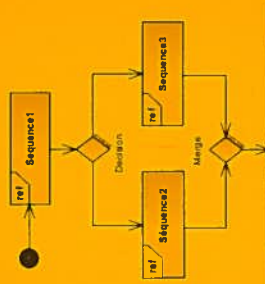
- Un nœud initial possède un arc sortant et pas d'arc entrant. La fin d'activité possède un ou plusieurs arcs entrants et aucun arc sortant. Dès que l'un des arcs entrants est activé, l'exécution de l'activité s'interrompt complètement. Une fin de flot *flow final* permet de terminer une branche autonome sans interrompre complètement l'activité englobante.
- Un flot d'objet représente l'existence d'un objet, générée par une action et utilisée par d'autres. Il est noté par un rectangle contenant un nom et indiquant optionnellement un type et un état.
- Trois actions particulières ont des notations graphiques spécifiques : la réception d'événement *AcceptEvent*, l'envoi de signal *SendSignal* et le timer *AcceptTimeEvent*.

Diagramme d'activité

- Une action peut représenter l'appel d'une autre activité. On le représente par un symbole de râteau en bas à droite de l'action.
- Les nœuds d'une activité peuvent être organisés en partitions *verticales* ou *horizontales* permettant de séparer les responsabilités. C'est particulièrement utile pour modéliser des processus d'entreprise.
- N'utilisez pas forcément toutes les subtilités du diagramme d'activité : pensez à votre lecteur !

Diagramme de vue globale d'interaction

- Combine les diagrammes d'activité et de séquence pour organiser des fragments d'interaction avec des décisions et des flots



Trucs et astuces

- L'utilisation de ce nouveau type de diagramme *Interaction Overview* paraît prometteuse au niveau des cas d'utilisation, voire de la dynamique du système global. Il permet de relier des diagrammes de séquence entre eux afin de montrer le cadre global dans lequel ils s'exécutent.
- Il peut être aussi une manière plus lisible d'exprimer des alternatives en comparaison avec l'opération *alt* du diagramme de séquence.

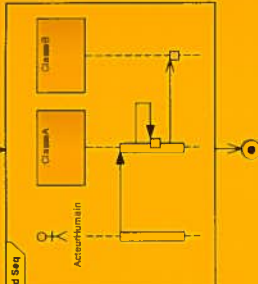


Diagramme de temps

- Combine les diagrammes d'états et de séquence pour montrer l'évolution de l'état d'une ligne de vie au cours du temps et les messages qui modifient cet état

Trucs et astuces

- Le diagramme de temps *timing diagram* est une autre variante de diagramme d'interaction dans lequel l'accent est mis sur l'expression des contraintes temporelles.
- Il ressemble à un diagramme de séquence dans lequel les lignes de vie seraient horizontales et le temps proportionnel. Il montre les oscillations entre différents états ou conditions, avec une échelle de temps concrète.
- Il est surtout destiné aux systèmes informatiques dits « temps réel ».

