

UML

Diagramme  
d'Activité  
(DAC)

## 1.1 - CONTEXTE

Un cas d'utilisation peut être représenté par un Diagramme d'Activité (DAC).

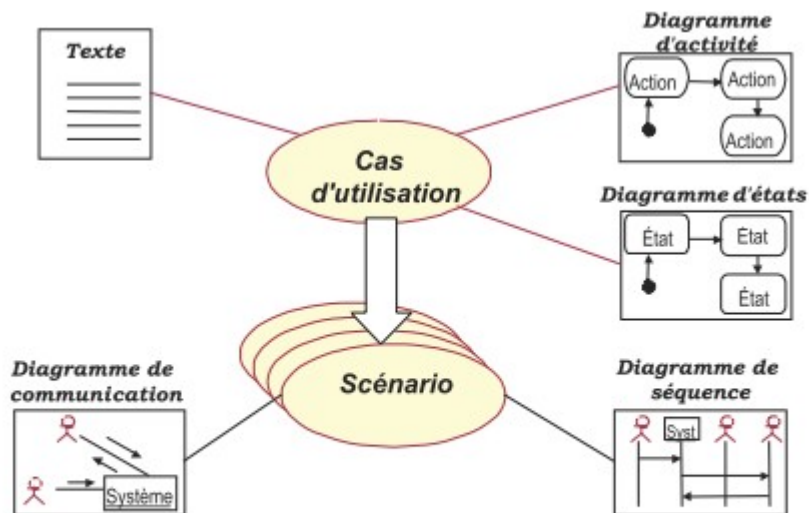
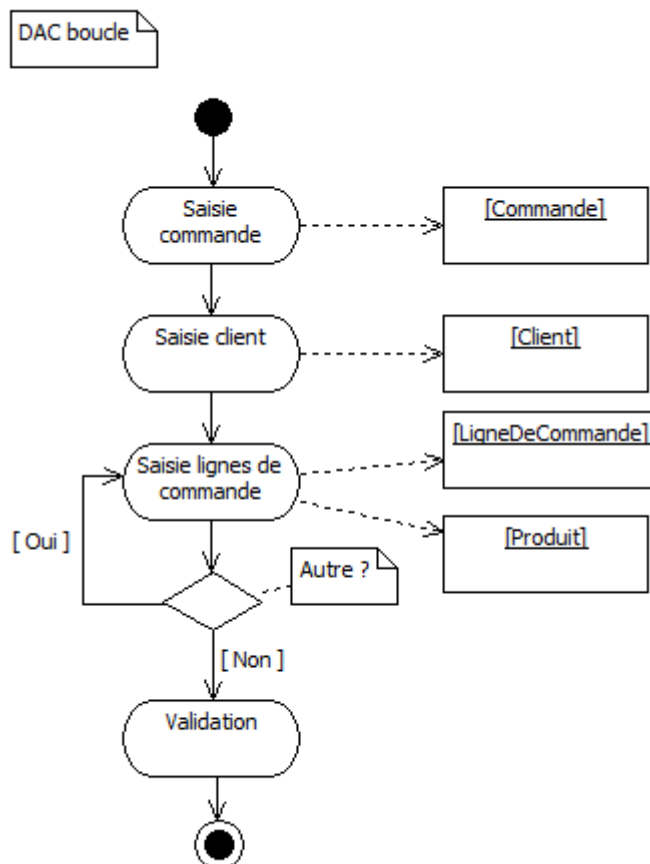
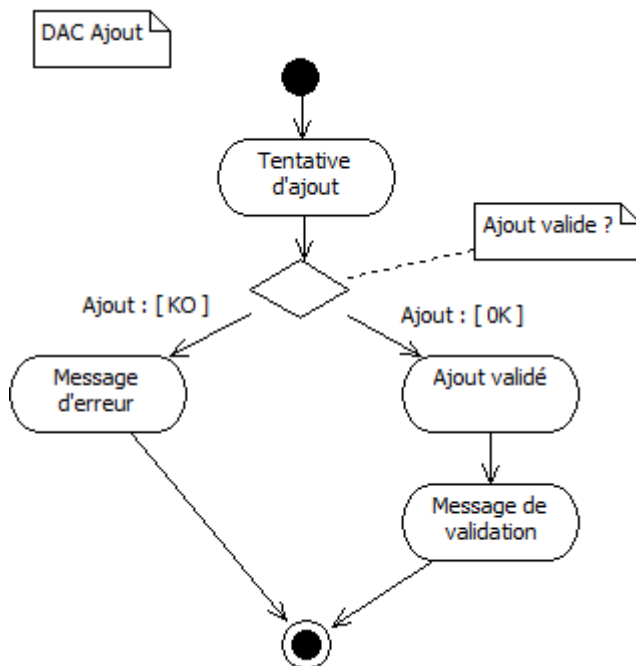


Figure 4-5 : Types de diagrammes dynamiques utilisables pour documenter les cas d'utilisation

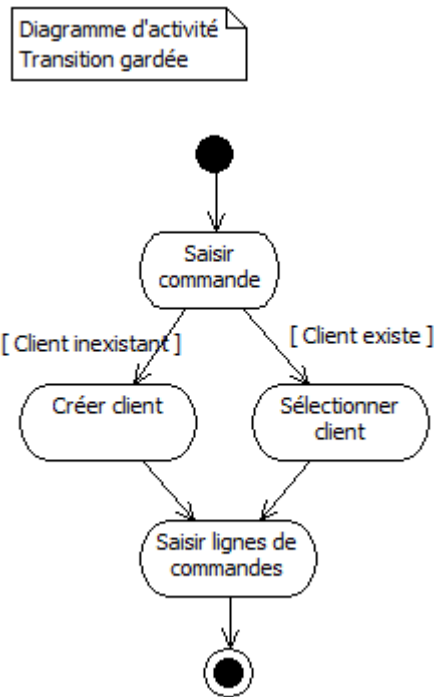
Figure empruntée à Pascal Roques « UML 2 en action », page 87 version papier, et page 75 version PDF.

## 1.2 - DEUX EXEMPLES ...



### 1.3 - DÉFINITION

Un diagramme d'activité représente une partie de la dynamique du système (les processus, les traitements) ou d'un objet (les méthodes).



Un diagramme d'activité représente graphiquement le déroulé d'un cas d'utilisation du point de vue de l'utilisateur, d'une opération d'une classe ou un DET (Diagramme d'Etat-Transition).

Il est organisé par rapport aux activités (unité d'un traitement divisible en plus petites unités appelées actions).

Un DAC (Diagramme d'Activité) représente l'exécution d'un mécanisme, sous la forme d'un déroulement d'étapes regroupées séquentiellement et/ou conditionnellement et/ou itérativement. Les activités s'exécutent le plus souvent avec des transitions instantanées. Mais dans certains cas il représente des communications (via des signaux) inter diagrammes d'activité qui nécessitent des délais d'attente. UML2 introduit même des la notion général d'action pouvant se réaliser à n'importe quel moment (At any time, ni StarUML 1 ni StarUML 2 ne permettent de le représenter !)

**Définitions officielles (UML 2 - User Guide.pdf) :**

Modeling a workflow  
Modeling an operation  
Forward and reverse engineering

Page 311 :

Activity diagrams are one of the five diagrams in the UML for modeling the dynamic aspects of systems. An activity diagram is essentially a flowchart, showing flow of control from activity to activity.

Unlike a traditional flowchart, an activity diagram shows concurrency as well as branches of control.

You use activity diagrams to model the **dynamic aspects of a system**. For the most part, this involves modeling the sequential (and possibly concurrent) steps in a computational process. With an activity diagram, you can also **model the flow of values** among steps. Activity diagrams may stand alone to visualize, specify, construct, and document the dynamics of a society of objects, or they may be used to model the flow of control of an operation. Whereas interaction diagrams emphasize the flow of control from object to object, activity diagrams emphasize the flow of control from step to step. An activity is an ongoing structured execution of a behavior. The execution of an activity ultimately expands into the execution of individual actions, each of which may change the state of the system or communicate messages.

Activity diagrams are not only important for modeling the dynamic aspects of a system, but also **for constructing executable systems** through forward and **reverse engineering**.

Un diagramme d'activité possède un pseudo état initial (rond plein).

Un diagramme d'activité possède un ou plusieurs pseudo états finals (rond plein entouré d'un cercle).

Les activités sont représentées par des rectangles aux coins arrondis.

Les transitions sont représentées par des flèches.

Les gardes – conditions - sont représentées par des losanges ou/et par des "gardes" sur les transitions.

Les synchronisations en entrée et en sortie sont représentées par des lignes horizontales.

Les états de classes sont représentés par des rectangles en parallèle des activités.

Les flots d'objets sont représentés par des rectangles intermédiaires reliés par des flèches entre les activités.

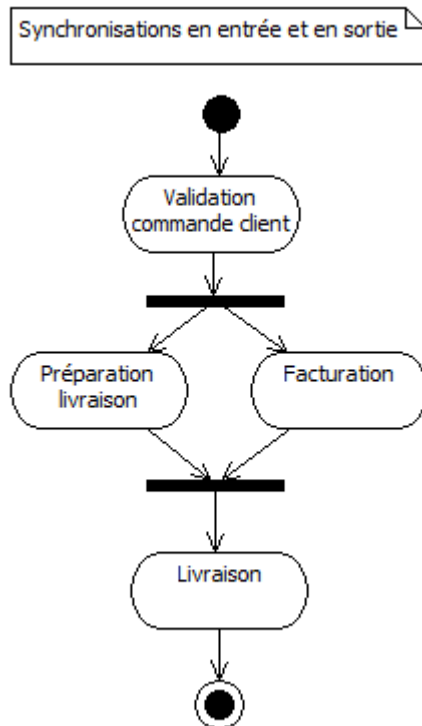
**Notes :**

Il peut être utilisé pour représenter la navigation entre écrans tout comme le diagramme d'états-transitions.

Il est plus usuel d'utiliser des verbes pour nommer les actions mais ... un substantif avec un complément de nom ...

## 1.4 - SYNCHRONISATION

Prise de commande, préparation et facturation, puis livraison.



Une **synchronisation en entrée** (matérialisée par la barre horizontale, aussi nommée nœud de bifurcation) permet d'ouvrir des **transitions parallèles** pour des activités parallèles.

Exemple : suite à la validation d'une commande 2 activités sont déclenchées; la préparation de la commande et la préparation de la facture.

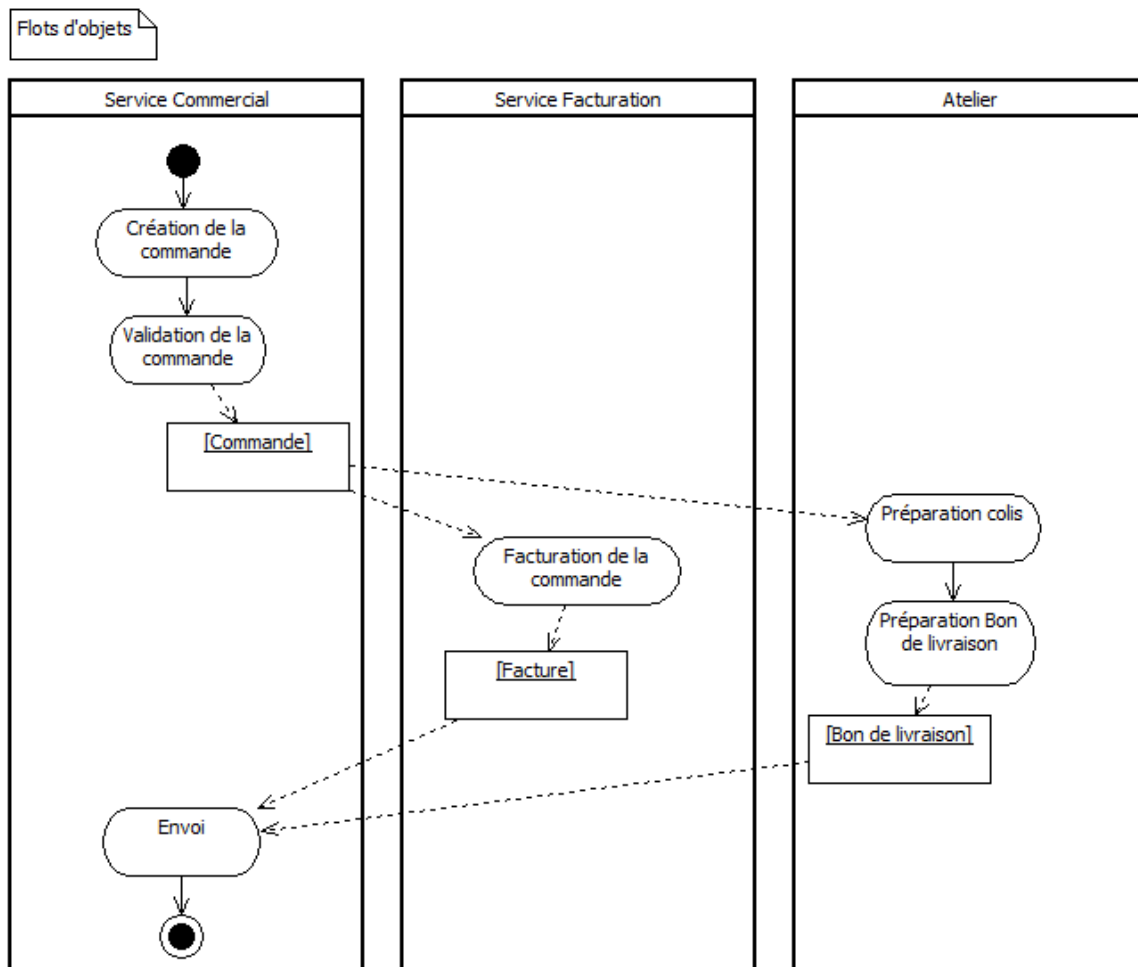
Une **synchronisation en sortie** (matérialisée par la barre horizontale, aussi nommée nœud de jointure) permet de fermer des transitions parallèles. L'activité suivante ne sera exécutée que lorsque les transitions en entrée seront toutes déclenchées. **Cela correspond à un ET logique.**

Exemple : lorsque la commande est préparée et que la facture est prête la livraison pourra avoir lieu.

## 1.5 - SWINLANE (VERTICAL), COULOIR

### | Swimlane (Vertical)

Couloir vertical : représente une partition, une séparation, lorsque plusieurs acteurs – par exemple – participent à l'activité.



Note : selon les cas une synchronisation en entrée et une autre en sortie pourraient être nécessaires.



## 1.6 - MOTS-CLÉS

Pseudo état initial, pseudo état final, activité, transitions, garde.

Synchronisation en entrée, synchronisation en sortie.

Etat de classe, flots d'objets.

Flot final.

Couloir.

Signaux.

Sous-activité.