

## Chapitre VI : Diagramme d'activités

### 1- Introduction

Les diagrammes d'activités permettent de spécifier des traitements a priori séquentiels. Ils offrent un pouvoir d'expression très proche de langage de programmation objet : spécification des actions de base (déclaration de variables, affectation...), structures de contrôle (conditionnelles, boucles), ainsi que les instructions particulières à la programmation orientée objet (appels d'opérations, exceptions...).

Ils sont donc bien adaptés à la spécification détaillée des traitements en phase de réalisation. On peut aussi les utiliser de façon plus informelle pour décrire des enchaînements d'actions de haut niveau, en particulier pour la description détaillée des cas d'utilisation.

### 2- Principes et définition de base

#### 2.1- Activité et transition

Les traitements complets sont décrits par des activités, qui offrent une manière concise et sans ambiguïté de présenter graphiquement un traitement séquentiel, avec tous l'arsenal proposé par un langage de programmation orienté objet (actions de base, boucles et conditionnelles, appels de méthode, gestion des exceptions...)

Les activités donnent donc une description complète des traitements associés à des comportements au sens interaction d'UML. On les utilise couramment pour étiqueter les autres diagrammes (traitements associés aux messages des diagrammes de séquence, transitions des diagrammes d'état-transitions).

#### ***Comment représenter ?***

La vision des diagrammes d'activités est centrée sur les flots de contrôle. On y trouve deux éléments fondamentaux :

- Les activités, représentées par un rectangle aux coins arrondis, décrivent un traitement. Le flot de contrôle reste dans l'activité jusqu'à ce que les traitements soient terminés. On peut définir des variables locales à une activité et manipuler les variables accessibles depuis le contexte de l'activité (classe contenant en particulier). Les activités peuvent être imbriquées hiérarchiquement : on parle alors d'activité composite.
- Les transitions, représentées par des flèches pleines qui connectent les activités entre elles, sont déclenchées dès que l'activité source est terminée et déterminent la prochaine activité à déclencher. Contrairement aux activités, les transitions sont franchies de manière atomique, en principe sans durée perceptible.

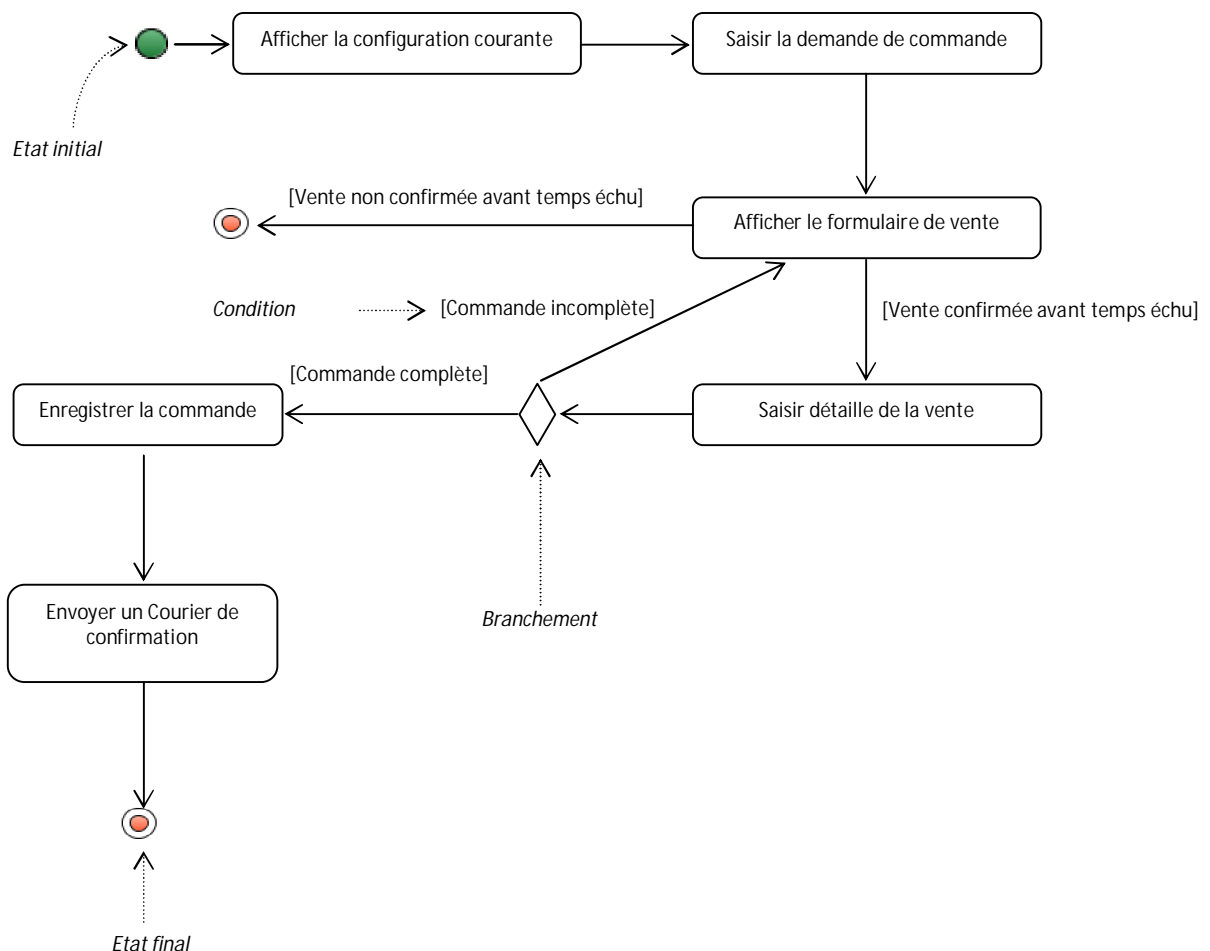


Figure VI.1 : Système de vente en ligne  
Cas d'utilisation : Commander ordinateur

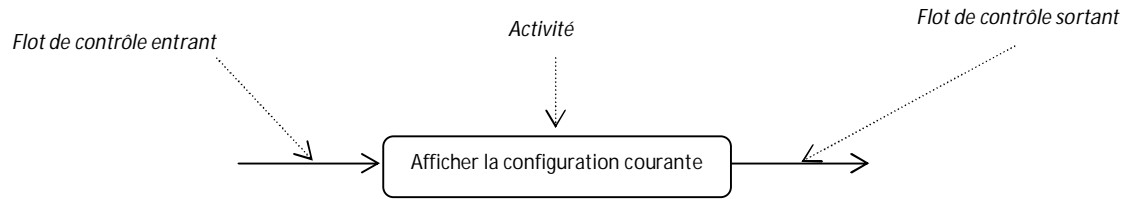


Figure VI.2 : Notation des activités.

## 2.2- Exercice : recette de cuisine

Recette simplifiée : commencer par casser le chocolat en morceaux, puis le faire fondre.  
En parallèle, casser les œufs en séparant les blancs des jaunes.

Quand le chocolat est fondu, ajouter les jaunes d'œuf.

Battre les blancs en neige jusqu'à ce qu'ils soient bien fermes.

Les incorporer délicatement à la préparation chocolat sans les briser.

Verser dans des ramequins individuels.

Mettre au frais au moins 3 heures au réfrigérateur avant de servir.

Représentez par un diagramme d'activité la recette de la mousse au chocolat.

### ***Solution***

Le diagramme d'activité simple est donné par la figure suivante (figure VI.3).

Notez l'utilisation des barres d'embranchement (fork) et de jointure (join), permettant de représenter précisément le parallélisme dans les flots de contrôle de l'activité.

Notez également l'utilisation du nouveau symbole graphique UML 2.0 pour l'action « Attendre 3h » qui est de type : accept time event.

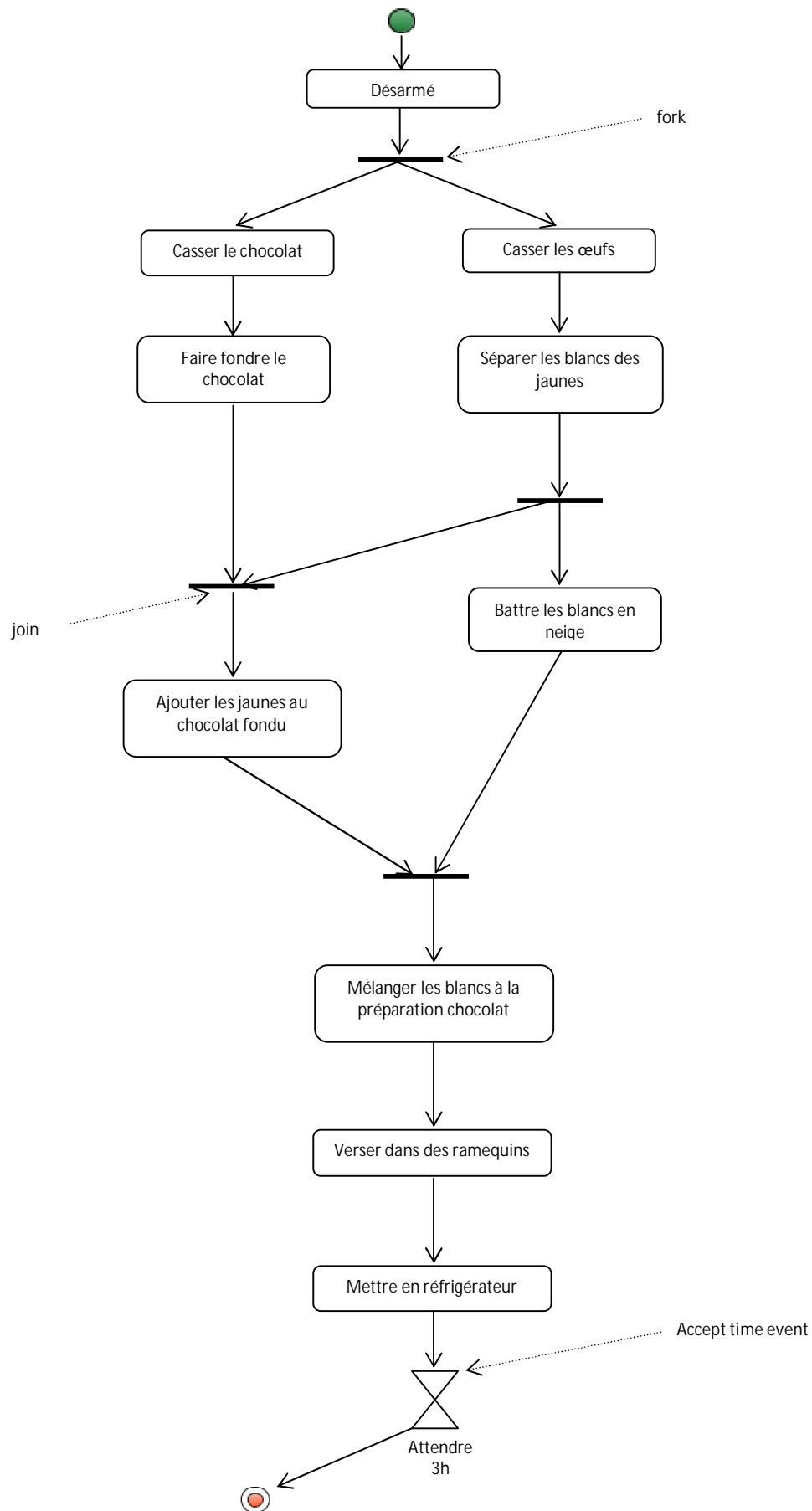


Figure VI.3 : Exemple de diagramme d'activité simple.