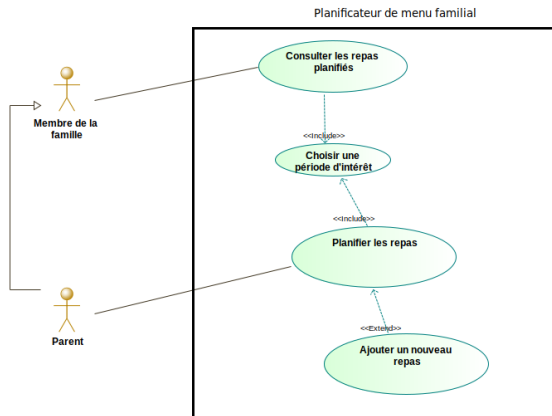


# Mémo UML

## diagrammes de cas d'utilisation, activités, états

Septembre 2021

# Diagrammes de cas d'utilisation



- ▶ frontières du système
- ▶ acteurs
- ▶ cas d'utilisation
- ▶ relations entre cas d'utilisation : extension, inclusion, héritage
- ▶ relation entre acteurs : héritage

# Les machines à états

Les machines à états, aussi appelés diagrammes d'état-transition, servent à modéliser la dynamique d'un sous-système, souvent d'une classe. Une machine à états décrivant le comportement d'une classe décrit en fait la dynamique de toutes ses instances à la réception ou à l'envoi de messages.

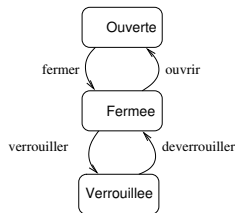


FIGURE – Diagramme d'état-transition très simple pour une porte

- ▶ Un état modélise une situation où un certain invariant (généralement implicite) est maintenu (la porte est fermée, un compte bancaire a un solde positif, ...)
- ▶ Transition : passage d'un état à un autre
  - ▶ Il peut y avoir plusieurs événements déclencheurs possibles, auquel cas on les liste tous (en les séparant par des virgules).
  - ▶ L'action peut être une affectation d'attribut, un appel de méthode, ...
  - ▶ Quand aucun événement déclencheur n'est spécifié, la transition est dite spontanée.

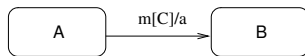


FIGURE – Une transition

# États initial et final

Un pseudo-état initial (noté graphiquement par un petit disque noir) représente un sommet qui est la source d'une seule transition vers l'état "par défaut" d'une machine à état ou d'un état composite. La transition initiale peut être munie d'une action. L'état final matérialise le fait qu'une région (une machine à état ou une région d'état composite) est "terminée" (voir notation figure 3).



FIGURE – États initial et Final

Un état composite :

- ▶ soit contient une seule région
- ▶ soit est décomposé en 2 ou plusieurs régions orthogonales

Un état inclus dans une région d'un état composite est appelé un sous-état de cet état composite. C'est un sous-état direct quand il n'est pas contenu par un autre état, et sinon un sous-état indirect.

# Exemple de machine à état avec état composite

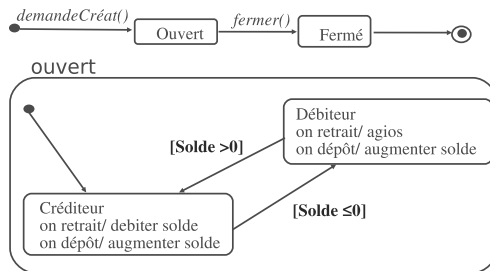


FIGURE – Exemple de machine à état avec état composite

# Exemple avec état à régions orthogonales

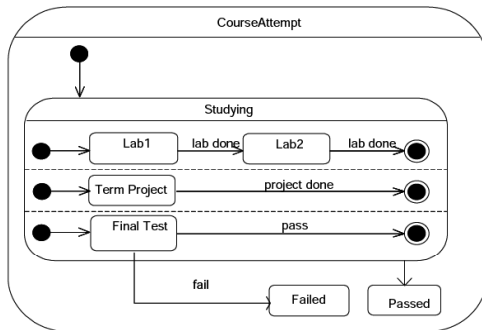


FIGURE – État composite orthogonal, extrait du document de spécification d'UML 2.0



# Comportement d'entrée et de sortie, comportement dans un état

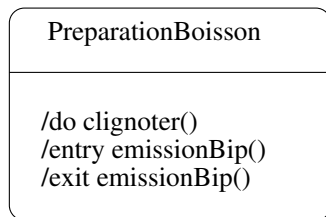


FIGURE – Actions d'entrée et de sortie des états, comportement dans les états

+ on evenement / action

# États Historiques

Il existe des états dits “mémoire” qui permettent de rentrer dans un état composite dans le même sous-état que quand on en est sorti. Il y a deux états mémoire : historique superficiel et historique profond.

**Historique superficiel (Shallow history)** (noté H). L'historique superficiel représente le sous-état actif le plus récent (mais pas les sous-états de ce sous-état).

**Historique profond (Deep history)** (noté H\*). L'historique profond représente la configuration active la plus récente de l'état composite qui contient directement l'historique profond (c'est-à-dire la configuration active la dernière fois qu'on a quitté l'état composite).



FIGURE – États ShallowHistory et DeepHistory

# Autres pseudo-états

Il existe d'autres pseudo-états comme les jonctions, les choix ou les branchements, nous ne les détaillerons pas ici.

# Les diagrammes d'activité

Les diagrammes d'activités permettent de représenter des flots de contrôle et de données. Ils permettent donc par exemple de représenter le comportement d'une opération ou d'un cas d'utilisation. Les diagrammes d'activité sont des graphes, avec différents types de nœuds et d'arcs. Ils mettent en jeu principalement :

- ▶ des nœuds actions
- ▶ des nœuds de contrôle permettant de spécifier l'enchaînement des actions (synchronisation, branchement, ...)
- ▶ des nœuds d'objet permettant de représenter les objets créés ou utilisés au cours d'une activité
- ▶ des arcs de transition permettant de relier les nœuds.

# Exemple

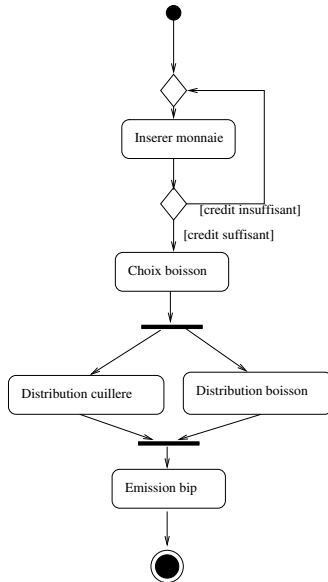







FIGURE – Exemple de diagramme d'activités



# Représentation de graphes de flot de contrôle

- ▶ Nœud initial (Initial node)  . Point d'entrée pour invoquer une activité. Un jeton de contrôle est placé au nœud initial quand l'activité commence.
- ▶ Nœud de fin d'activité (Activity final node)  . Stoppe tous les flots dans une activité. Un jeton atteignant un nœud de fin d'activité fait avorter tous les flots en cours, l'activité est donc terminée et le jeton est détruit (ainsi que tous les jetons circulant dans l'activité).
- ▶ Nœud de fin de flot (Flow Final node)  . Termine un flot. Le nœud de flot final détruit les jetons y entrant.

# Représentation de graphes de flot de contrôle



- ▶ Nœud d'action (Action node) . Unité fondamentale de la fonctionnalité exécutable d'une activité. Une action s'exécute quand toutes les contraintes sur ses *flots de contrôle* entrants sont satisfaites (jonction implicite). L'exécution consomme les jetons de contrôle entrants puis présente un jeton sur chaque flot sortant (branchement implicite).
- ▶ Flot de contrôle (Control flow) . Passage des jetons. Les jetons offerts par le nœud source sont offerts au nœud destination.

# Représentation de graphes de flot de contrôle

- ▶ Nœud de décision (Decision node)  . Choix parmi les flots sortants. Chaque jeton arrivant sur un nœud de décision ne peut traverser qu'un seul flot sortant. Les jetons ne sont pas dupliqués. Ce sont les gardes sur les flots sortants qui permettent le choix (les gardes doivent assurer le déterminisme du choix).
- ▶ Nœud de branchement (Fork node)  . Partage d'un flot en flots concurrents. Les jetons arrivant d'un branchement sont dupliqués sur les flots sortants.



# Représentation de graphes de flot de contrôle

- ▶ Nœud de jonction (Join node)  . Synchronisation de plusieurs flots. Si un jeton de contrôle est offert sur chaque flot entrant, alors un jeton de contrôle est offert sur le flot sortant.
- ▶ Nœud de fusion (Merge node)  . Rassemblement de plusieurs flots. Tous les jetons offerts sur les flots entrants sont offerts sur le flot sortant sans synchronisation.
- ▶ Partition d'activité (Activity Partition). Identifie des actions ayant une caractéristique commune. Les partitions n'affectent



pas le flot des jetons.

