

## **Annexe de la 48<sup>e</sup> Compétition Nationale des Métiers**

# **MÉTIER N°09 SOLUTIONS LOGICIELLES EN ENTREPRISE**

### **Diagrammes de cas d'utilisation (Use Case)**

Soumis par :

Paul NGO, Expert WorldSkills France

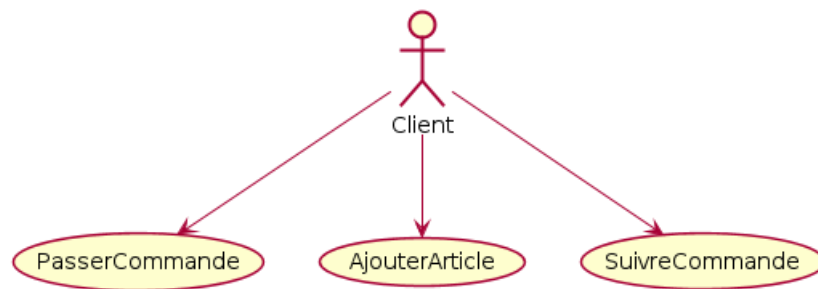
Geoffrey DA ENCARNACAO, Expert adjoint WorldSkills France

## 1. DIAGRAMMES DE CAS D'UTILISATION

- UML (Unified Modeling Language) est un langage de modélisation standardisé par l'OMG.
- Objectifs : représenter la structure, le comportement, et les interactions d'un système.
- Diagrammes principaux :
  - Structurels : classes, composants, déploiement
  - Comportementaux : cas d'utilisation, activités, états
  - D'interaction : séquences, communication
- Le diagramme de cas d'utilisation sert à analyser et communiquer les besoins fonctionnels.

## 2. DEFINITION D'UN CAS D'UTILISATION

- Un **use case** décrit une interaction entre un **acteur** (rôle externe) et le **système** pour atteindre un objectif métier.
- Caractéristiques :
  - Centré sur la valeur pour l'utilisateur
  - Exprimé en langage métier (pas technique)
  - Indépendant de l'implémentation



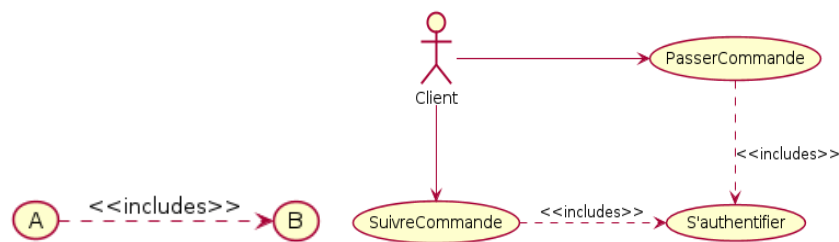
## 3. ÉLEMENTS CONSTITUTIFS

- **Acteurs** : rôles externes au système (humains ou systèmes).
- **Cas d'utilisation** : fonctionnalités ou services offerts.
- **Frontière système** : rectangle englobant les use cases.
- **Relations** :
  - **Association** : acteur ↔ use case
  - **<include>** : inclusion obligatoire d'un cas
  - **<extend>** : extension optionnelle/conditionnelle d'un cas
  - **Généralisation** : hiérarchie acteur → acteur ou use case → use case

## 4. LES RELATIONS <INCLUDE> ET <EXTEND>

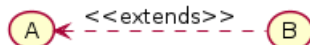
### <include> (inclusion)

- Inclusion **obligatoire** d'un use case dans un autre.
- Sémantique : le scénario du cas inclus est **toujours exécuté**.
- Exemple :
  - Cas base : **Passer commande**
  - Cas inclus : **Vérifier stock**
  - Relation : Passer commande <<include>> Vérifier stock



### <extend> (extension)

- Extension **optionnelle** ou conditionnelle d'un use case.
- Sémantique : l'extension s'exécute **seulement si la condition est remplie**.
- Exemple :
  - Cas base : **Payer commande**
  - Cas extension : **Appliquer remise**
  - Relation : Payer commande <<extend>> Appliquer remise si le client a un code promo valide



## 5. GENERALISATION

### 5.1 Généralisation d'acteurs

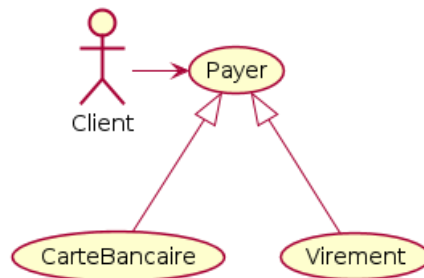
- Permet de factoriser des rôles communs et de créer des **profils**.
- Exemple :
  - Acteur général : *Utilisateur*
  - Acteurs spécialisés : *Client, Administrateur*
- Les acteurs spécialisés héritent des **cas d'utilisation** du général et peuvent ajouter des cas spécifiques.

## 5.2 Généralisation de cas d'utilisation

- Permet de modéliser des variantes d'un même objectif métier.
- Exemple :
  - Cas général : **Payer réservation**
  - Cas spécialisés : **Payer par carte bancaire**, **Payer via PayPal**
- Le cas spécialisé hérite du scénario principal et peut ajouter des étapes spécifiques.

## 5.3 Schéma conceptuel

- Triangle UML blanc pointe vers l'élément général.
- Contrairement à <<include>> / <<extend>>, il ne s'agit pas d'une séquence ou d'une dépendance conditionnelle, mais d'une **hiérarchie**.



## 6. BONNES PRATIQUES POUR LA MODELISATION

- Ne pas confondre généralisation et <<include>> / <<extend>>.
- Limiter la complexité : 5–15 use cases par diagramme.
- Nommer les use cases en verbe + complément : Réserver vol, Payer facture.
- Documenter chaque use case avec une fiche textuelle.
- Utiliser <<extend>> uniquement pour les scénarios vraiment conditionnels.

## 7. METHODOLOGIE DE CONSTRUCTION

1. Identifier les acteurs (utilisateurs, systèmes externes).
2. Lister les objectifs métiers (services attendus).
3. Définir les use cases correspondants.
4. Tracer les associations acteur ↔ use case.
5. Détecter les comportements communs → <<include>>.
6. Repérer les variantes conditionnelles → <<extend>>.
7. Compléter avec des descriptions textuelles (scénarios principaux, alternatifs, extensions).