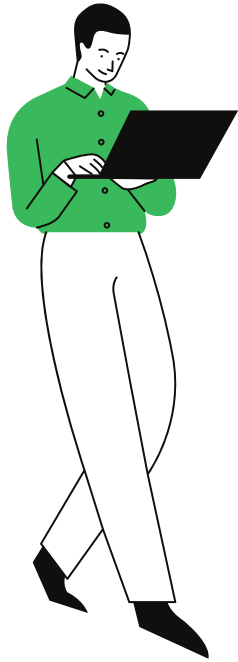


LA NOTATION UML & SES APPLICATIONS PRATIQUES

Exposé





Introduction

Face à la complexité croissante des systèmes d'information, de nouvelles méthodes ont émergé, notamment la programmation orientée objet (P.O.O.), qui a marqué une avancée majeure au cours des 15 dernières années.

Les méthodes de modélisation classiques, comme MERISE, ont dû s'adapter pour répondre aux exigences de cette nouvelle approche. Plusieurs méthodes orientées objet, telles que Booch et OMT, ont vu le jour.

Pour standardiser ces méthodes, l'Object Management Group (OMG) a créé UML (Unified Modeling Language), une notation unifiée issue de la fusion de Booch, OMT et OOSE.



1

Qu'est-ce que l'UML ?

2

Pourquoi utiliser l'UML ?

1 Qu'est-ce que la Notation UML ?

L'UML, ou Unified Modeling Language, est un langage standardisé qui utilise des diagrammes pour représenter visuellement la structure et le comportement d'un système informatique. C'est un peu comme un plan architectural pour un logiciel.

Les principaux avantages de l'UML :

- Amélioration de la qualité
- Réduction des coûts
- Facilitation de la maintenance
- Amélioration de la collaboration

2 Pourquoi modéliser ?

- Comprendre la complexité : Les systèmes logiciels peuvent être très complexes. L'UML permet de décomposer ces systèmes en éléments plus simples et de visualiser les relations entre ces éléments.
- Communiquer efficacement: L'UML offre un langage commun à toutes les parties prenantes d'un projet (développeurs, chefs de projet, clients), facilitant ainsi la communication et la collaboration.
- Détecter les erreurs en amont : En modélisant un système avant de le développer, on peut identifier et corriger les erreurs potentielles plus tôt dans le cycle de développement, ce qui réduit les coûts et les délais.

Les Principaux Types de Diagrammes UML

1

**Diagramme de
classes**

2

**Diagramme de
séquence**

3

**Diagramme
d'activités**

4

**Autres
Diagrammes**

Diagramme de classe : La Structure de Votre Système

Qu'est-ce que c'est ?

- Représentation graphique de la structure statique d'un système.
- Montre les classes, leurs attributs (caractéristiques) et leurs opérations (comportements).
- Les relations entre les classes sont aussi représentées.

À quoi ça sert ?

- Modéliser la structure interne du système.
- Identifier les objets et leurs interactions.
- Servir de base à la conception du code.

Éléments clés:

- **Classe:** Un modèle d'objet avec des attributs et des opérations.
- **Attribut:** Propriété d'une classe.
- **Opération:** Action qu'une classe peut effectuer.
- **Relations:** Association, généralisation, agrégation, composition.

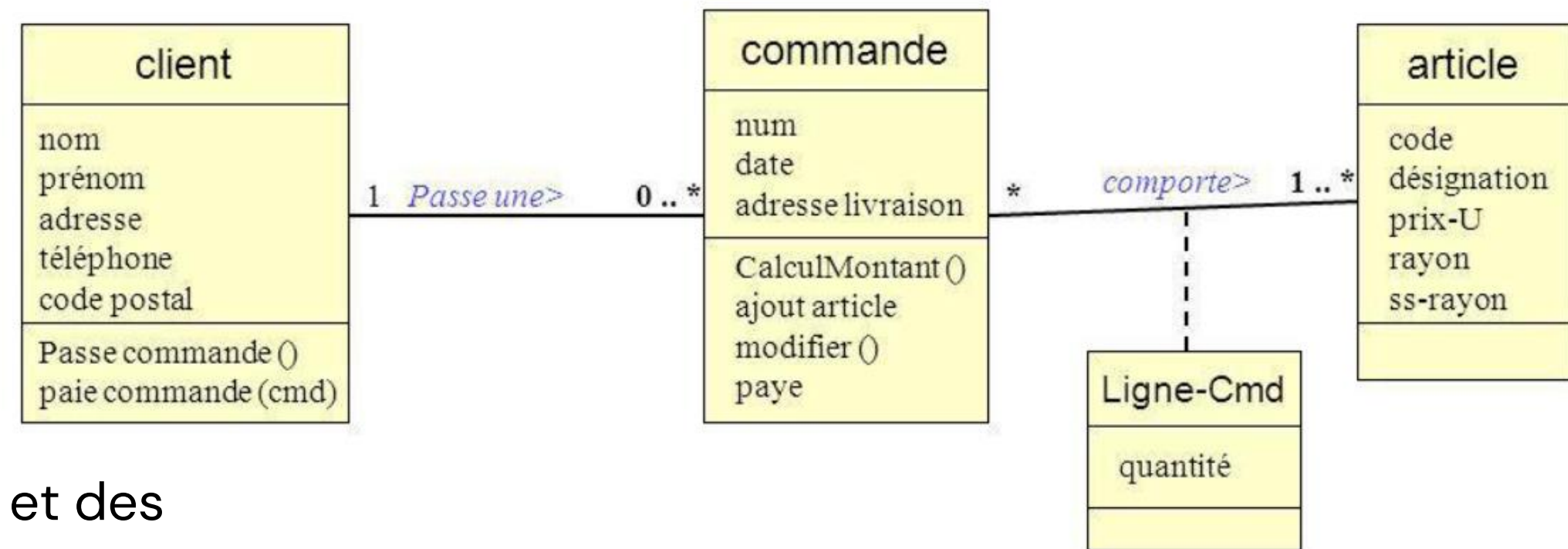


fig. 2 : Exemple de Diagramme de classe pour la gestion de commande

Diagramme de séquence : L'Ordre des Événements

Qu'est-ce que c'est ?

- Représentation graphique de l'interaction entre différents objets au cours du temps.
- Montre l'ordre chronologique des messages échangés.

À quoi ça sert ?

- Visualiser le déroulement d'un scénario.
- Analyser les flux de contrôle.
- Identifier les problèmes de coordination.

Éléments clés:

- **Objet:** Représenté par une ligne de vie verticale.
- **Message:** Interaction entre deux objets, représenté par une flèche.
- **Ligne de vie:** Représente l'existence d'un objet au cours du temps.

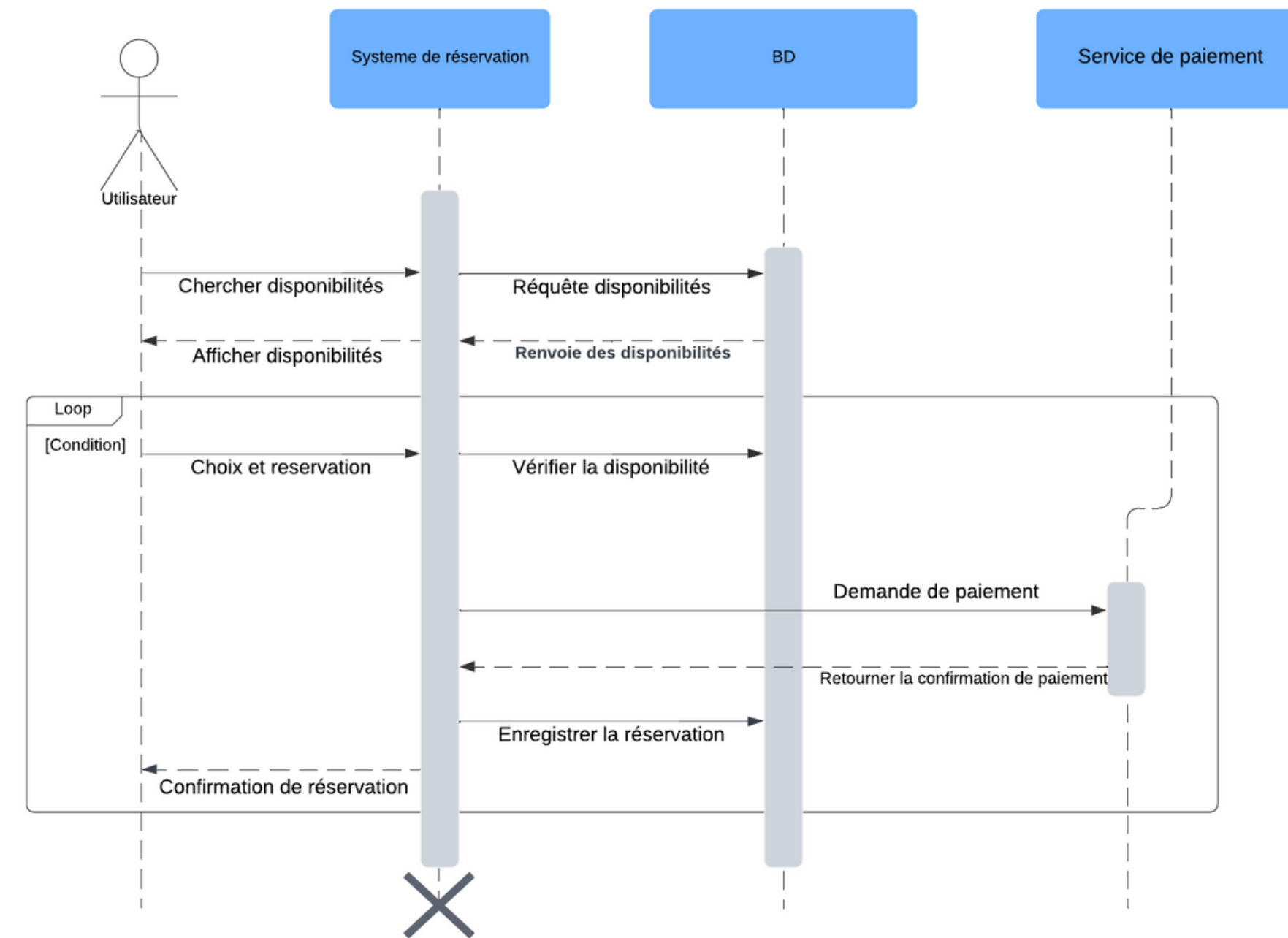


fig. 3 : Exemple de Diagramme de séquence qui représente les interactions entre objets dans un processus de réservation en ligne

Diagramme d'activité : Le Flux des Processus

Qu'est-ce que c'est ?

- Représentation graphique du déroulement d'un processus.
- Montre les différentes étapes d'un processus et les conditions qui régissent le passage d'une étape à l'autre.

À quoi ça sert ?

- Modéliser les flux de travail.
- Analyser les processus métier.
- Identifier les goulots d'étranglement.

Éléments clés:

- **Activité:** Tâche à accomplir.
- **Flèche:** Indique le flux de contrôle.
- **Décision:** Point de branchement conditionnel.
- **État initial et final:** Démarrage et fin du processus.

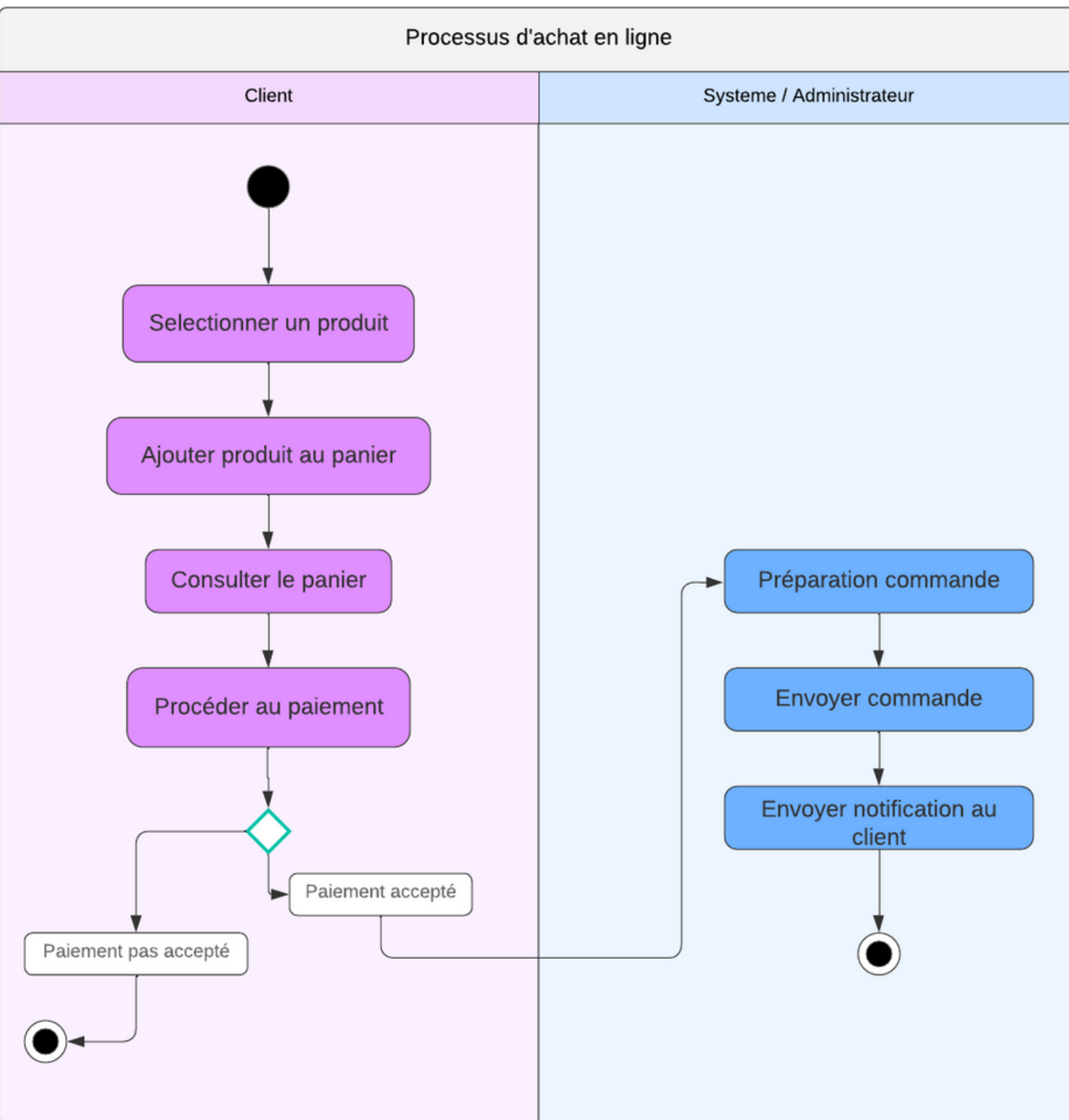


fig. 4 : Exemple de Diagramme d'activité décrivant le processus d'achat en ligne

Autres diagrammes (Diagrammes de structure statiques & dynamiques)

Diagrammes de structure (statiques)

- **Diagramme d'objets:** Montre une instance d'un diagramme de classes à un moment donné.
- **Diagramme de composants:** Représente la structure physique d'un système.
- **Diagramme de déploiement:** Montre la configuration matérielle et le déploiement des logiciels.
- **Diagramme de paquetages:** Organise les éléments d'un modèle en groupes logiques.
- **Diagramme de structure composite:** Montre la structure interne d'une classe ou d'un composant.
- **Diagramme de profil:** Permet d'étendre le métamodèle UML.

Diagrammes de comportement (dynamiques)

- **Diagramme de cas d'utilisation:** Représente les interactions entre un système et ses utilisateurs.
- **Diagramme d'état:** Montre les différents états d'un objet et les transitions entre ces états.
- **Diagramme de communication:** Variante du diagramme de séquence, met l'accent sur les liens structurels.
- **Diagramme de temps:** Montre les changements d'état d'un objet au cours du temps.
- **Diagramme de vue d'ensemble des interactions:** Représente une vue globale d'un ensemble d'interactions.

Merci ! 