Diagramme de cas d’utilisation

Introduction à l’UML

Définition

UML (Unified Modeling Language) est un langage de modélisation **visuelle** utilisé pour représenter et documenter les **systèmes logiciels** et leurs composants.

Historique

- **Développé** dans les **années 90**
- Fruit de la **collaboration** entre **Grady Booch**, **James Rumbaugh** et **Ivar Jacobson**
- Standard **officiel** de l'**OMG** (Object Management Group) depuis **1997**

Importance dans le développement logiciel

- Favorise la **compréhension** des exigences et du comportement des systèmes
- Facilite la **communication** entre les développeurs et les non-développeurs
- Aide à **l'organiser** et **planifier** les projets
- Sert de base pour la **codification**, la **maintenance** et les **tests** du logiciel

---

Composants des cas d'utilisation UML

Acteurs

Les **acteurs** représentent les **entités externes** qui interagissent avec le **système**.

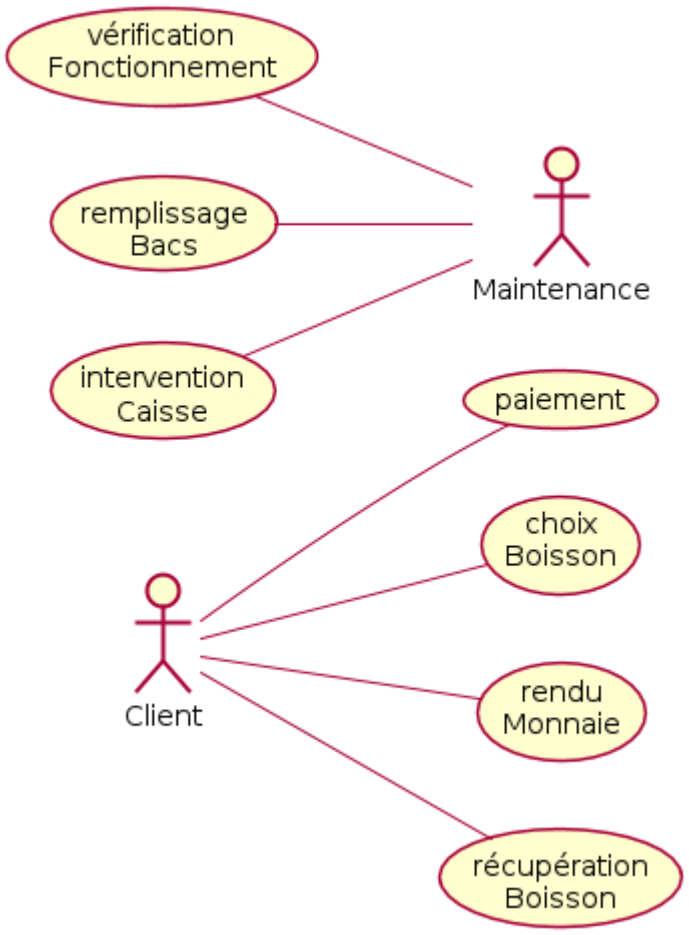
- Utilisateurs
- Autres systèmes
- Bases de données

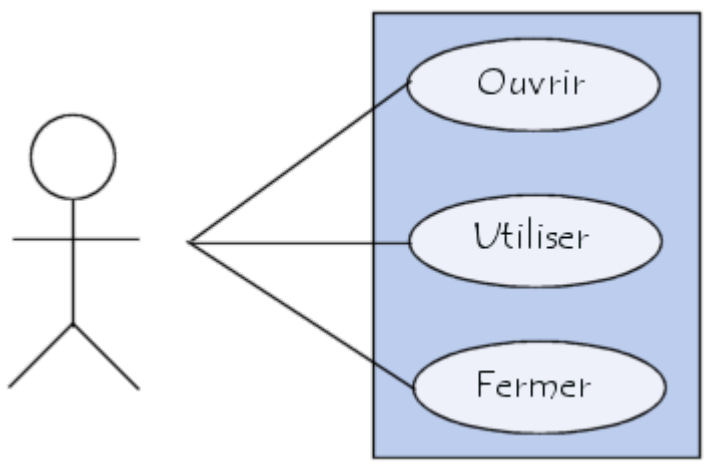
Les acteurs peuvent être physiques (comme les employés) ou logiques (comme des processus automatisés, des services).

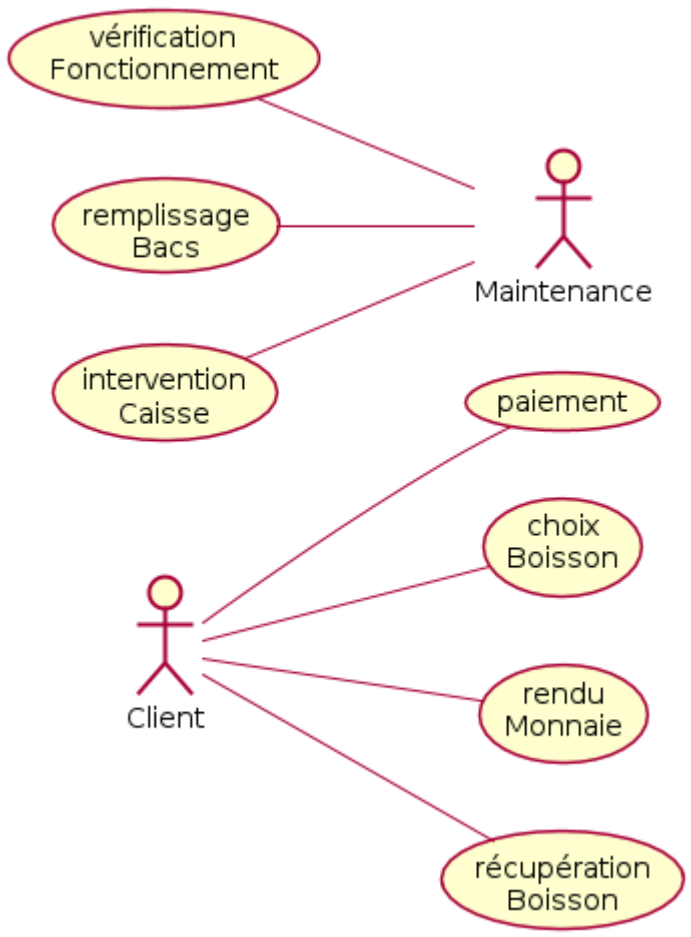
Définition

Un **acteur** est une entité (personne, organisation, ou un autre système) qui interagit avec le système pour **atteindre un objectif**.

Exemple de representation d’un acteur :







Rôles

Les acteurs ont différents **rôles** à jouer dans le système, comme un **utilisateur**, un **administrateur**, ou un **fournisseur de services**.

Types d'acteurs

- **Utilisateur :** Les personnes qui utilisent le système
- **Système externe :** Les systèmes externes qui communiquent avec le système étudié
- **Organisation :** Les organisations (internes ou externes) impliquées dans le système

---

Cas d'utilisation

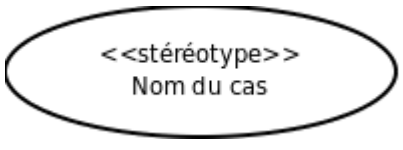
Les **cas d'utilisation** décrivent les **fonctionnalités** du système du point de vue d'un **acteur**.

Définition

Un **cas d'utilisation** est une unité d'interaction entre un **acteur** et le **système**, pour atteindre un objectif spécifique.

Les cas d'utilisation permettent d'identifier les différentes fonctionnalités qui seront présentes dans un système, ainsi que leurs interactions avec les acteurs impliqués. Ils sont essentiels pour définir les exigences fonctionnelles d'un projet.

Exemple de representation de cas d’utilisation :



Objectifs

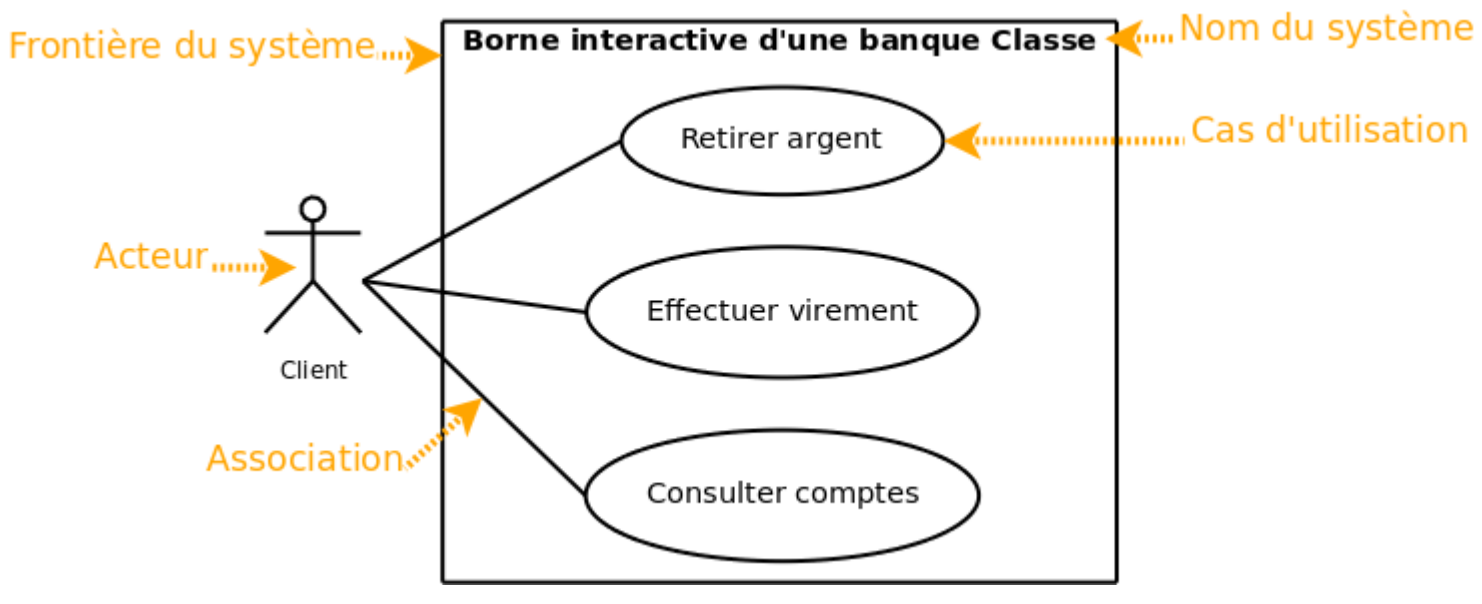
Les cas d'utilisation ont pour objectifs :

- **Décrire les fonctionnalités** du système
- **Communiquer les exigences**
- **Faciliter la conception** du système

Scénario

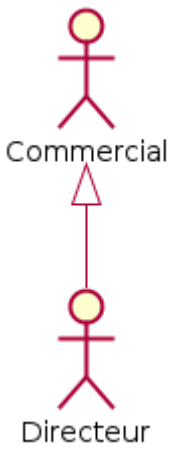
Un **scénario** est une séquence d'**actions** qu'un **acteur** exécute pour réaliser un **cas d'utilisation**.

Representation cas d’utilisation



Relations entre acteurs

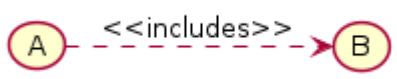
Il existe un seul type de relation possible entre acteurs : la relation de généralisation. Elle indique qu'un acteur est une sorte d'un autre acteur. Par exemple, un directeur est une sorte de commercial et peut effectuer toutes les actions d'un commercial, plus d'autres tâches.



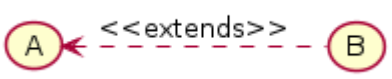
Relations entre cas d'utilisation

Il existe trois types de relations entre les cas d'utilisation :

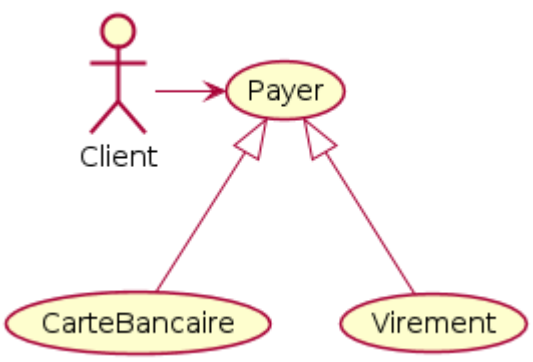
- Inclusion : B est une partie obligatoire de A (A inclut B).



- Extension : B est une partie optionnelle de A (B étend A).



- Généralisation : B est une généralisation de A (B est une sorte de A).



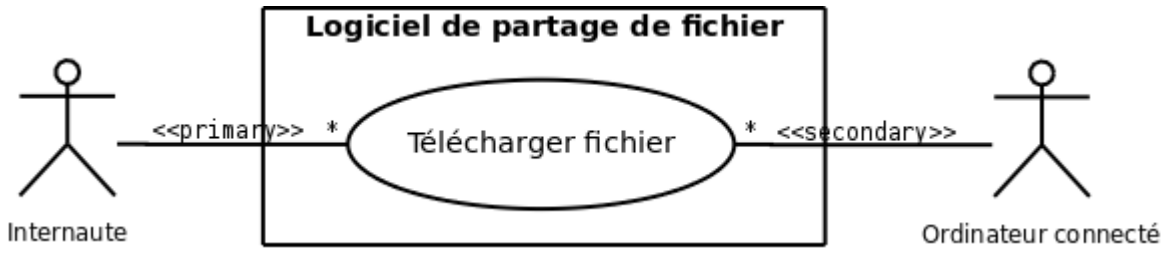
Relations

Les **relations** représentent les liens entre les **acteurs** et les **cas d'utilisation**.

Type de relation	Description
Association	Lien entre un acteur et un cas d'utilisation, indiquant que l'acteur participe au cas d'utilisation.
Héritage	Lien entre deux acteurs, indiquant que l'un hérite des caractéristiques de l'autre.
Include	Lien entre deux cas d'utilisation, indiquant qu'un cas d'utilisation inclut un autre cas d'utilisation.
Extend	Lien entre deux cas d'utilisation, indiquant qu'un cas d'utilisation étend un autre cas d'utilisation.

Association

Une **association** est une relation entre un **acteur** et un **cas d'utilisation**, indiquant que l'acteur participe à ce cas d'utilisation.



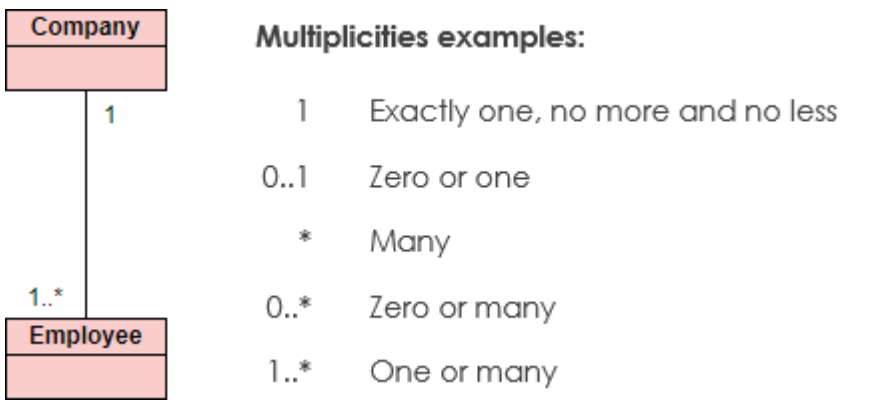
Une relation d'association est chemin de communication entre un acteur et un cas d'utilisation et est représenté un trait continu

Association - Multiplicité

La **multiplicité** indique la quantité d'instances qui peuvent être associées à une autre.

Exemples de multiplicité :

- **0..1** : aucune ou une instance
- **1..**: au moins une instance
- **1** : exactement une instance
- ***** : *n* instances
- **1..*** : une à *n* instances



Héritage

L'**héritage** indique qu'un acteur ou un cas d'utilisation dérive d'un autre, héritant de ses **propriétés** et de ses **relations**.

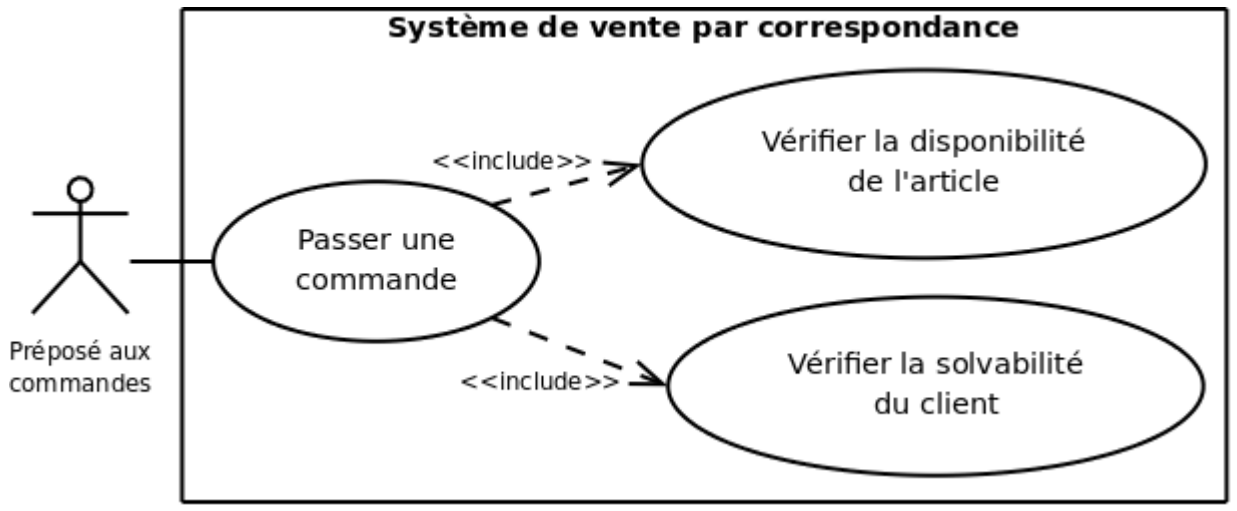
Exemple d'héritage : une classe `Personne` peut donner lieu à la création de sous-classes `Etudiant` et `Enseignant` qui héritent des propriétés et méthodes de la classe `Personne`.

Inclusion

L'**inclusion** permet de décomposer un cas d'utilisation en **sous-cas** d'utilisation, où un cas d'utilisation inclut un autre cas d'utilisation pour accomplir une partie de sa **fonctionnalité**.

L'inclusion est utilisée pour modéliser un scénario commun qui est partagé par plusieurs cas d'utilisation. Il permet de décomposer un cas d'utilisation en sous-cas d'utilisation pour éviter la duplication de scénarios.

La notion d'inclusion est importante pour comprendre comment différents cas d'utilisation peuvent interagir et se compléter pour accomplir une tâche plus complexe.

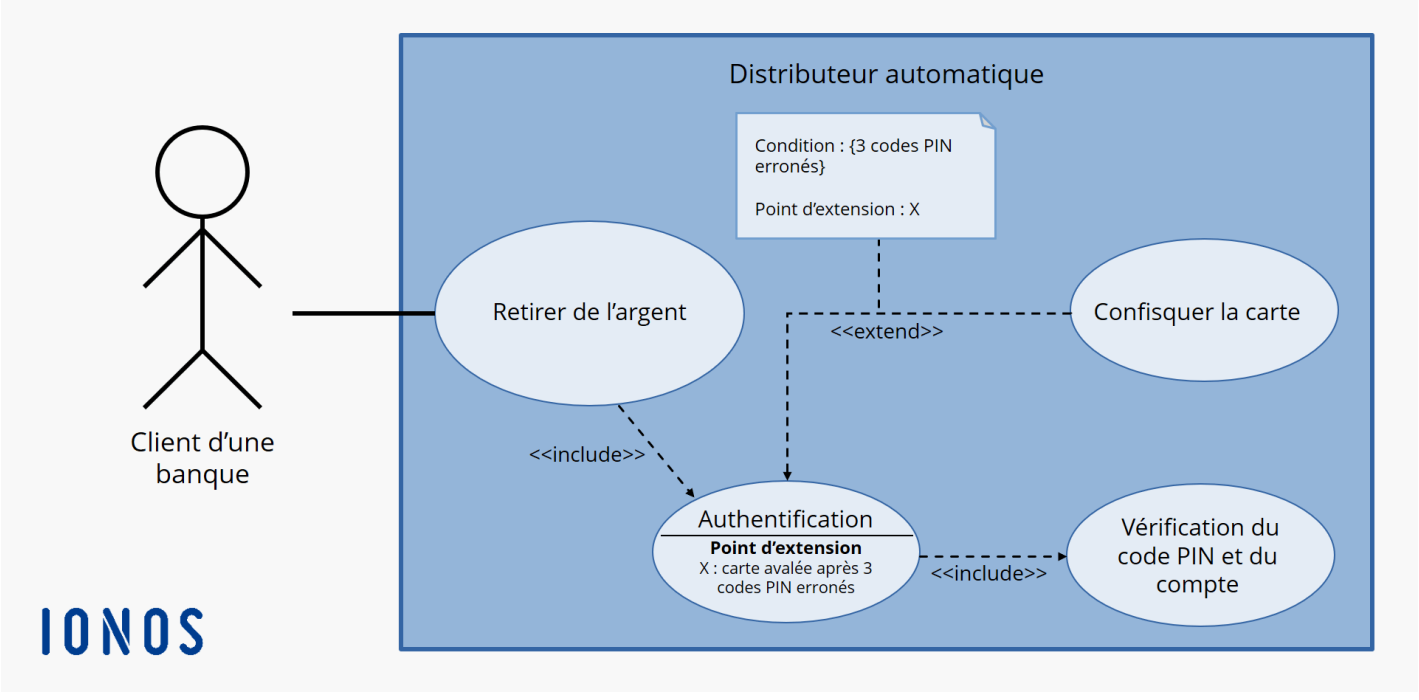


Extension

L'**extension** est une manière d'ajouter des **fonctionnalités** à un cas d'utilisation existant, en définissant des **points d'extension** où les cas d'utilisation secondaires peuvent s'ajouter.

L'extension est utilisée pour modéliser des comportements optionnels ou conditionnels qui étendent le scénario de base d'un cas d'utilisation. Il permet de représenter des scénarios alternatifs qui ne sont pas toujours exécutés.

Diagramme de cas d'utilisation UML : exemple « Retirer de l'argent »



Création d'un diagramme de cas d'utilisation

Identifier les acteurs

Les **acteurs** sont les entités externes qui interagissent avec le **système**.

Il peut s'agir d'utilisateurs, d'autres systèmes ou de matériel. Les acteurs peuvent communiquer avec le système en fournissant des entrées ou en recevant des sorties du système. Pensez à donner des exemples d'acteurs pour aider les stagiaires à mieux comprendre.

Identifier les utilisateurs

- Liste des utilisateurs potentiels du système
  - Administrateurs
  - Utilisateurs finaux
  - Utilisateurs invités
  - BDD
  - ...

- Décrire leurs rôles et responsabilités
  - **Administrateurs** : Gérer les utilisateurs, les permissions et les paramètres généraux
  - **Utilisateurs finaux** : Utiliser les fonctionnalités du système pour effectuer des tâches spécifiques
  - **Utilisateurs invités** : Avoir un accès limité à certaines fonctionnalités du système

L'identification des utilisateurs est la première étape pour définir les besoins et les objectifs du système.

Identifier les cas d'utilisation

Les cas d'utilisation représentent les **fonctionnalités** du système.

- Comprendre les besoins des utilisateurs
- Décrire les scénarios d'utilisation
- Prioriser les fonctionnalités
- Estimer les coûts et les efforts

Déterminer les fonctionnalités principales

- Liste des **fonctionnalités principales** du système
- Associer chaque **fonctionnalité** à un cas d'utilisation

Fonctionnalités	Cas d'utilisation

Ajout d'utilisateurs	Enregistrement d'un nouvel utilisateur
Authentification	Connexion de l'utilisateur
Gestion des tâches	Création, modification et suppression des tâches

---

Déterminer les sous-fonctionnalités

- Liste des sous-fonctionnalités pour chaque **cas d'utilisation principal**
- Associer chaque sous-fonctionnalité à un **cas d'utilisation secondaire**

Il est important d'identifier les sous-fonctionnalités pour mieux comprendre le fonctionnement général du système et faciliter le travail en équipe.

Etablir les relations entre les acteurs et les cas d'utilisation

Les **relations** décrivent comment les **acteurs** et les **cas d'utilisation** interagissent.

Établir des relations d'inclusion, d'extension ou d'héritage si nécessaire

- **Inclusion**: un cas d'utilisation en inclut un autre
- **Extension**: un cas d'utilisation étend un autre avec des fonctionnalités optionnelles
- **Héritage**: un cas d'utilisation hérite des propriétés d'un cas d'utilisation parent

---

UML. Introduction au diagramme de cas d'utilisation