

Génie Logiciel

UML
Modèle des
Use-Cases

Sommaire

The background features a large, faint gear silhouette on the left side. In the top right corner, there is a cluster of various grey gears of different sizes and designs, including some with holes or different tooth patterns. A horizontal grey line runs across the middle of the slide, separating the title from the list.

- Introduction
- Acteurs
- Use-Cases
- Diagrammes de *Use-Cases*

The background features a large, faint, light-gray silhouette of a gear on the left side. In the upper right corner, there is a cluster of various smaller gears in shades of gray, some with different tooth patterns and sizes. A thin, dark horizontal line runs across the middle of the page, positioned just above the main title.

Introduction

Modèle des *use-cases*

- On appelle modèle de *use-cases* (ou encore vue utilisateur) la partie du modèle d'un système qui rend compte de l'utilisation et des fonctionnalités de celui-ci.
- L'objectif du modèle des *use-cases* est double, c'est essentiellement :
 - ◆ décrire le comportement externe (**du point de vue des utilisateurs**) du système,
 - ◆ et déterminer les **limites** du systèmes en identifiant les éléments extérieurs (utilisateurs, matériel ou logiciel externe).

Modèle des *use-cases*

- Les *use-cases* sont utiles tout au long du cycle de développement logiciel.
 - ◆ Dès l'analyse des besoins, ils permettent de décrire ce que le système doit faire, ce qui doit être réalisé. Ils peuvent être associés ou même constituer l'essentiel du cahier des charges. Ils sont adaptés au dialogue entre analystes(MOE) et commanditaires(MOA).
 - ◆ Cette description peut ensuite être précisée et utilisée comme guide dans les étapes ultérieures du développement.



Acteurs

Acteurs



- Un acteur est un élément **extérieur** au système qui communique avec celui-ci dans le cadre de son fonctionnement.
- Les acteurs constituent la frontière du système. Ils sont les premiers éléments extérieurs en rapport avec celui-ci.
- La connaissance des acteurs n'est pas une connaissance de la structure du système mais de son interface.

Acteurs



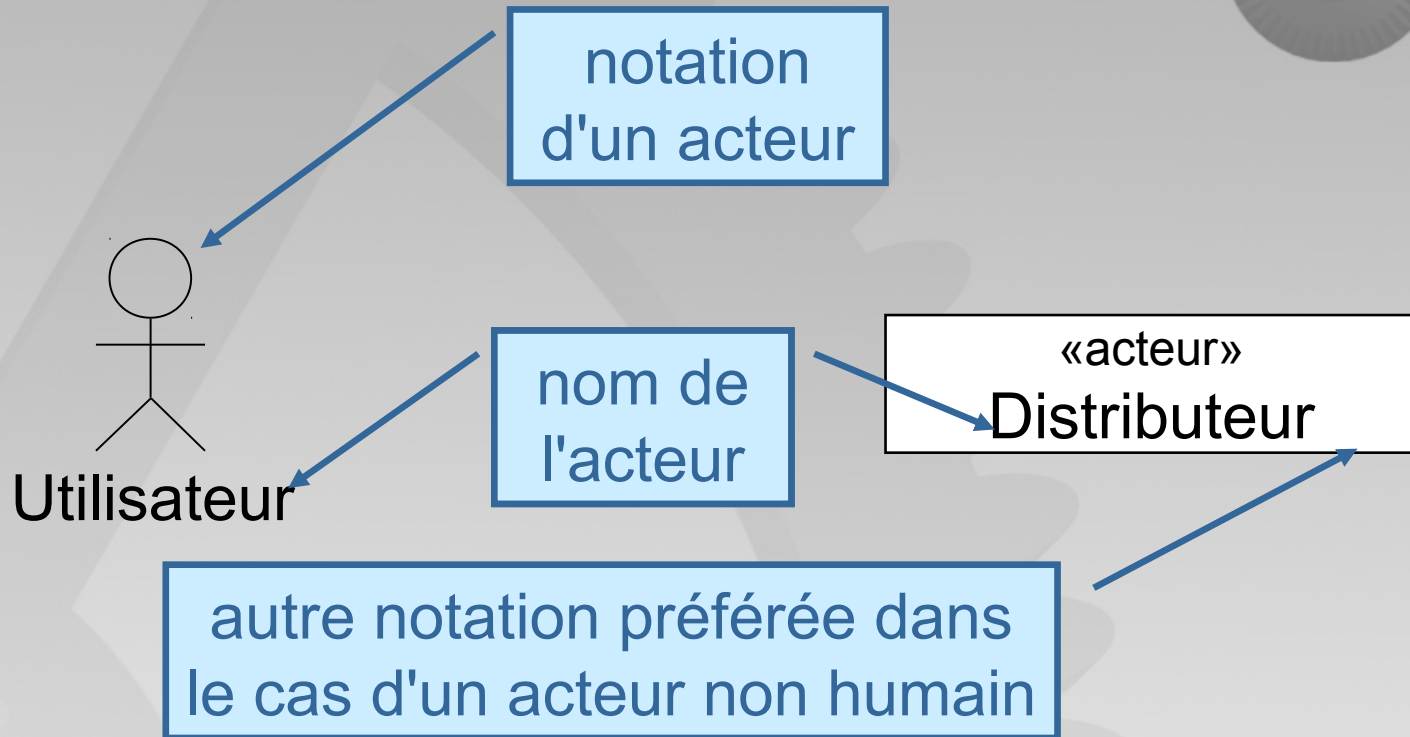
- On peut distinguer **3 types d'acteurs**, en tenant compte de leur nature :
 - ◆ les acteurs **humains**, généralement utilisateurs, ou encore administrateurs du système,
 - ◆ les acteurs **logiciels**, d'autres systèmes logiciels déjà existants, communiquant au travers d'un protocole ou d'une API,
 - ◆ les acteurs **matériels** (robots, automates, machines diverses, ...) pilotés par le système ou utilisant celui-ci.

Acteurs



- On peut aussi regrouper les acteurs en fonction de la finalité de leur interaction avec le système :
 - ◆ un acteur "**primaire**" est un acteur qui utilise les services du système,
 - ◆ un acteur "**secondaire**" est un acteur nécessaire au bon fonctionnement du système (administrateur, matériel ou logiciel requis par le système).

Notations d'acteurs



2 notations possibles pour les acteurs

Acteurs



- Un acteur est la description d'un **rôle**.
 - ◆ Un acteur est la description d'une entité **communicante** avec le système.
 - ◆ Un acteur peut communiquer avec le système dans différents contextes (*use-cases*). Pour chaque contexte, l'acteur représente une entité (personne, objet, ...) qui joue un **rôle** vis à vis du système.
 - ◆ Une même personne (ou objet, ...) peut jouer plusieurs rôles, être plusieurs fois acteur.

Généralisation d'acteurs



- Les acteurs peuvent être liés par une relation de **généralisation** :
 - ◆ l'acteur le plus spécifique hérite de l'acteur le plus général la description du rôle associé,
 - ◆ si une entité joue le rôle décrit par l'acteur le plus spécifique, elle joue au moins le rôle décrit par l'acteur le plus général.

The background features a large, faint, light-gray silhouette of a gear on the left side. In the upper right corner, there is a cluster of various smaller gears in shades of gray, including some with different tooth patterns and sizes. A thin, dark horizontal line runs across the middle of the page, passing behind the text.

Use-cases

Use-cases



- Un *use-case* est la description du dialogue (de l'interaction) entre un acteur et le système dans le cadre de l'accomplissement d'une **fonctionnalité** de ce dernier.
- Un *use-case* est une description abstraite, il décrit un dialogue mais **ne donne pas d'informations sur la structure ou les mécanismes**. Un *use-case* est l'abstraction de plusieurs scénarios.

Use-cases



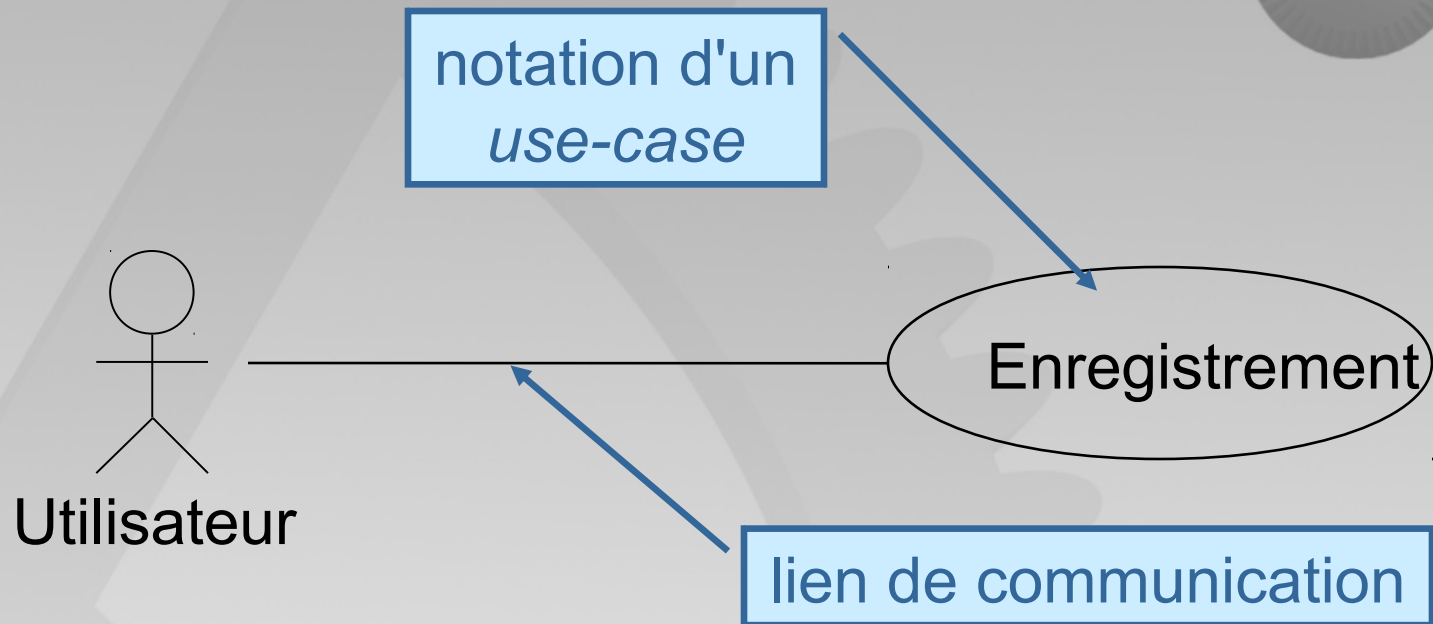
- Un *use-case* est décrit :
 - ◆ de manière informelle (le plus généralement) par des scénarios, chacun d'eux pouvant être décrit par un diagramme de séquence, de collaboration, ou une description textuelle ; il doit y avoir au moins un scénario dit "déroulement principal" et éventuellement plusieurs autres scénarios correspondant aux déroulements exceptionnels,
 - ◆ de manière formelle par un diagramme d'états ou un diagramme d'activités.

Notation des diagrammes de Use Cases



- Use cases
 - ◆ Dessinés comme des **ellipses** avec un nom à l'intérieur ou dessous de chaque ellipse
 - ◆ Décrivent **une séquence d'actions** que le système entreprend pour obtenir un résultat observable par un acteur
 - ◆ Le nom est habituellement un **verbe actif**

Notation de *use-case*



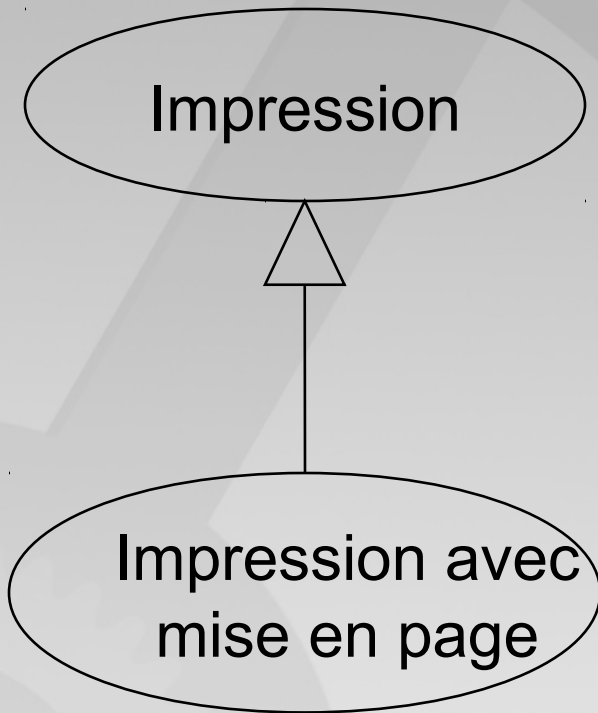
Un acteur communiquant avec un *use-case*

Notation des diagrammes de Use Cases



- Associations par communication
 - ◆ Ligne dessinée entre un acteur et un use-case
 - ◆ Représente un lien de communication entre une instantiation du use-case et une instantiation de l'acteur

Généralisation des *use-cases*

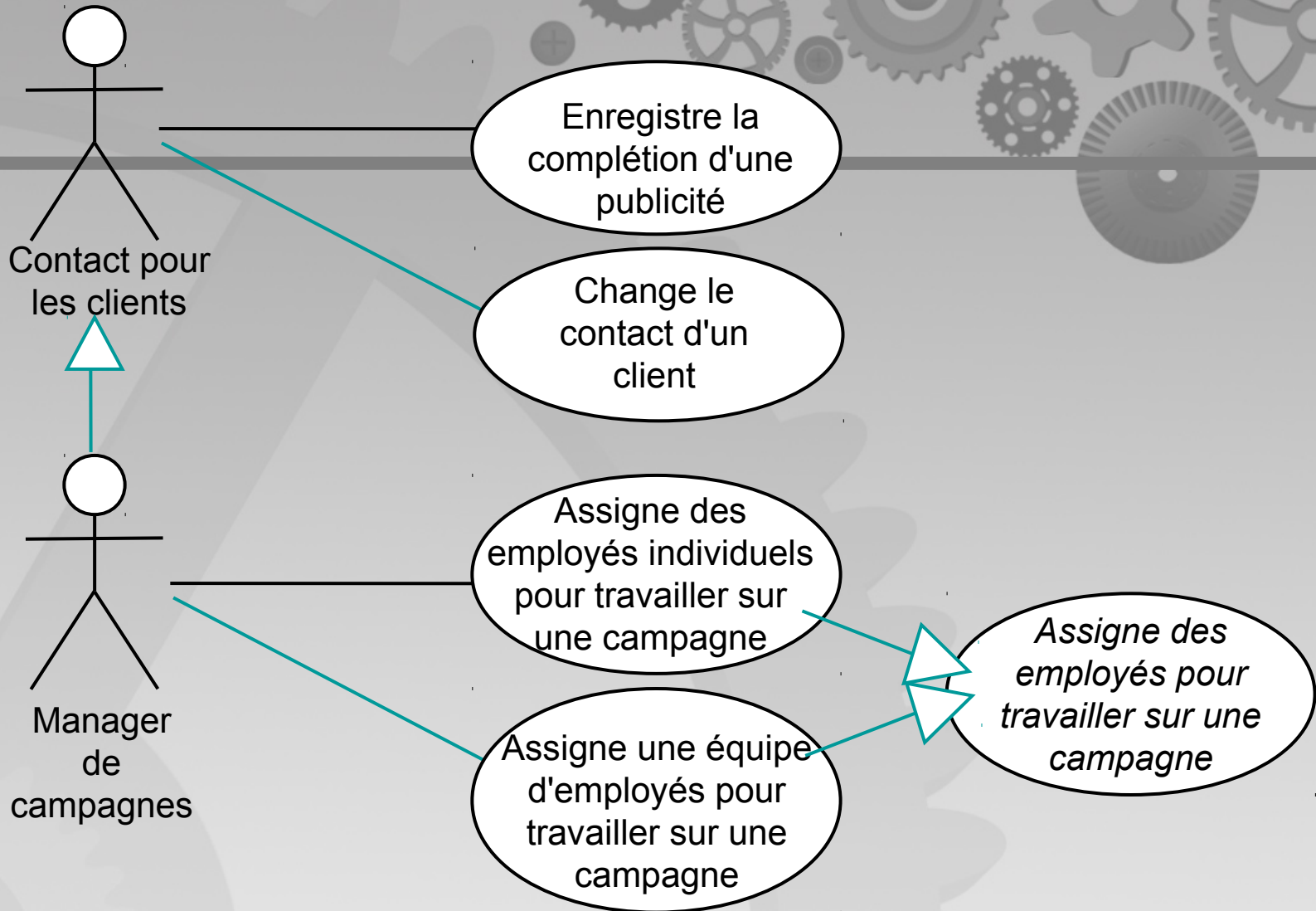


- Les *use-cases* peuvent être liés par une relation de généralisation :
 - ◆ "Impression" est plus général que "Impression avec mise en page",
 - ◆ "Impression avec mise en page" hérite de "Impression", y compris ses liens de communications.

Notation des diagrammes de Use Cases



- Généralisation
 - ◆ Montre qu'un use-case procure toute la fonctionnalité du use-case le plus spécifique et quelque fonctionnalité supplémentaire



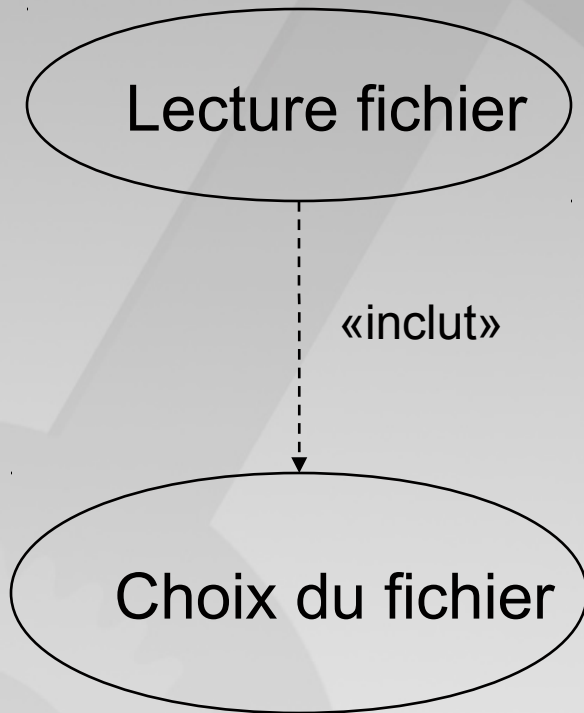
Notation des diagrammes de Use Cases



- Relations

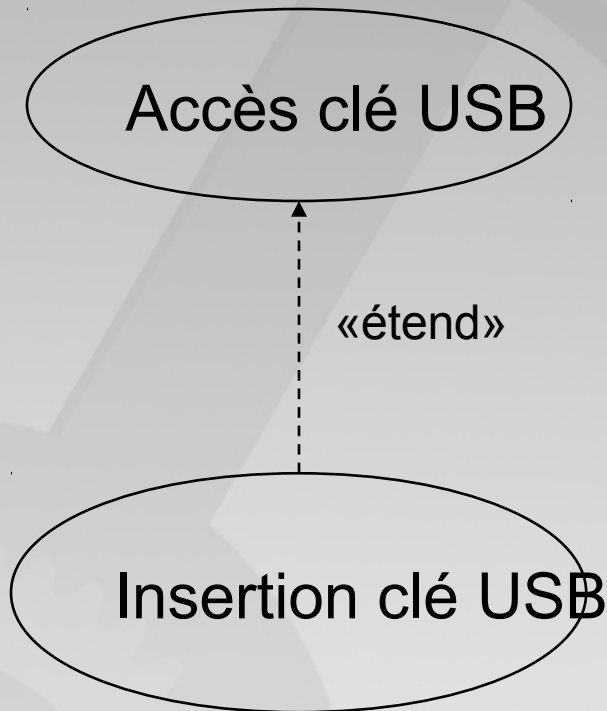
- ◆ Etend et includ
- ◆ Sont montrées comme des dépendances stéréotypées
- ◆ Les stéréotypes sont écrits comme: «étend» et «includ»

Inclusion entre *use-cases*



- La relation d'inclusion entre *use-cases* permet d'exprimer la décomposition fonctionnelle :
 - ◆ "Choix du fichier" **est toujours inclus dans le déroulement** de "Lecture fichier",
 - ◆ "Choix du fichier" est éventuellement partagé par (inclus dans) d'autres *use-cases*.

Extension de *use-cases*



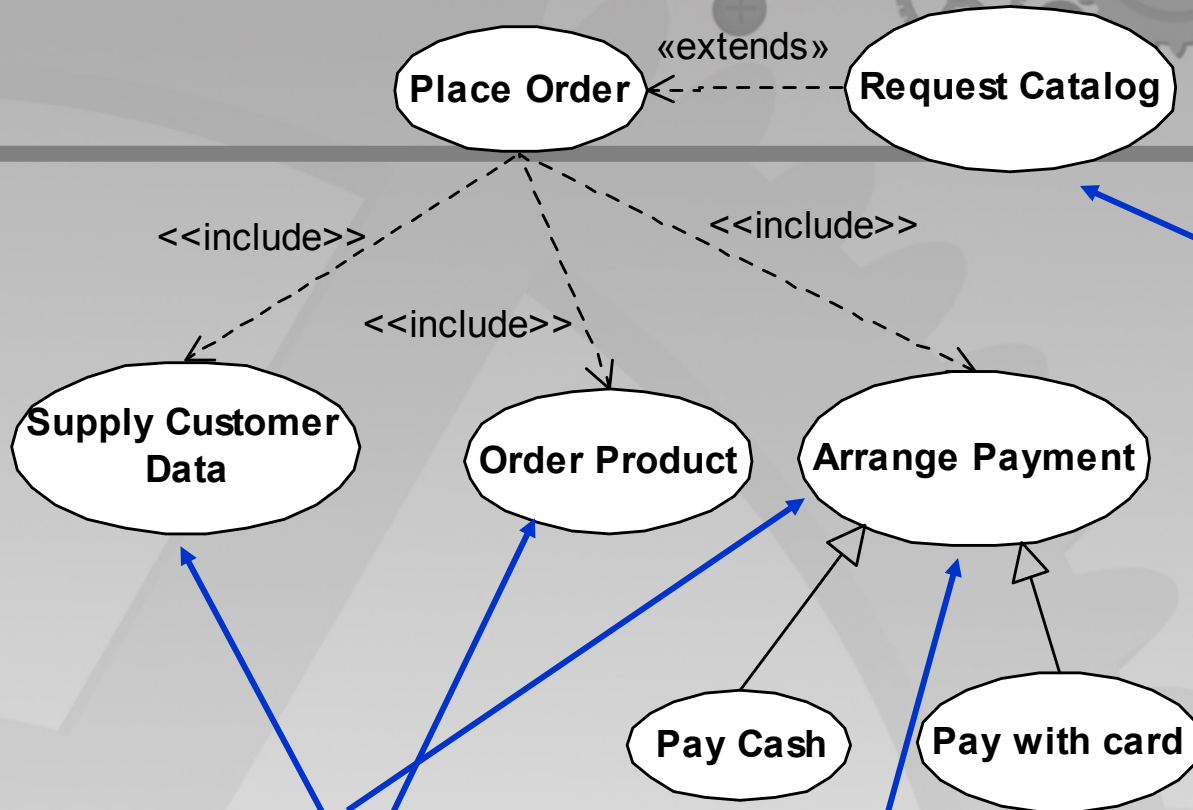
- La relation d'extension entre *use-cases* permet de prendre en compte les déroulements **exceptionnels ou occasionnels** :
 - ◆ "Accès clé USB" peut se dérouler sans que "Insertion clé USB" ne se déroule.

Notation des diagrammes de Use Cases

- Relation d'**extension**
 - ◆ Un use-case procure une fonctionnalité additionnelle qui **peut être** nécessaire dans un autre use-case
- Relation d'**inclusion**
 - ◆ un use case inclus **toujours** la fonctionnalité d'un autre use-case
 - ◆ Peut être utilisée pour séparer le comportement qui est utilisé dans de nombreux use-cases

- 
- Il est facile de confondre les relations d'**extensions**, d'**inclusion**, et la **généralisation**.

Les différentes relations



Selon la demande du client (condition d'extension)

Fonctionnalités toujours incluses

Use case de haut niveau décrivant les possibilités de paiement

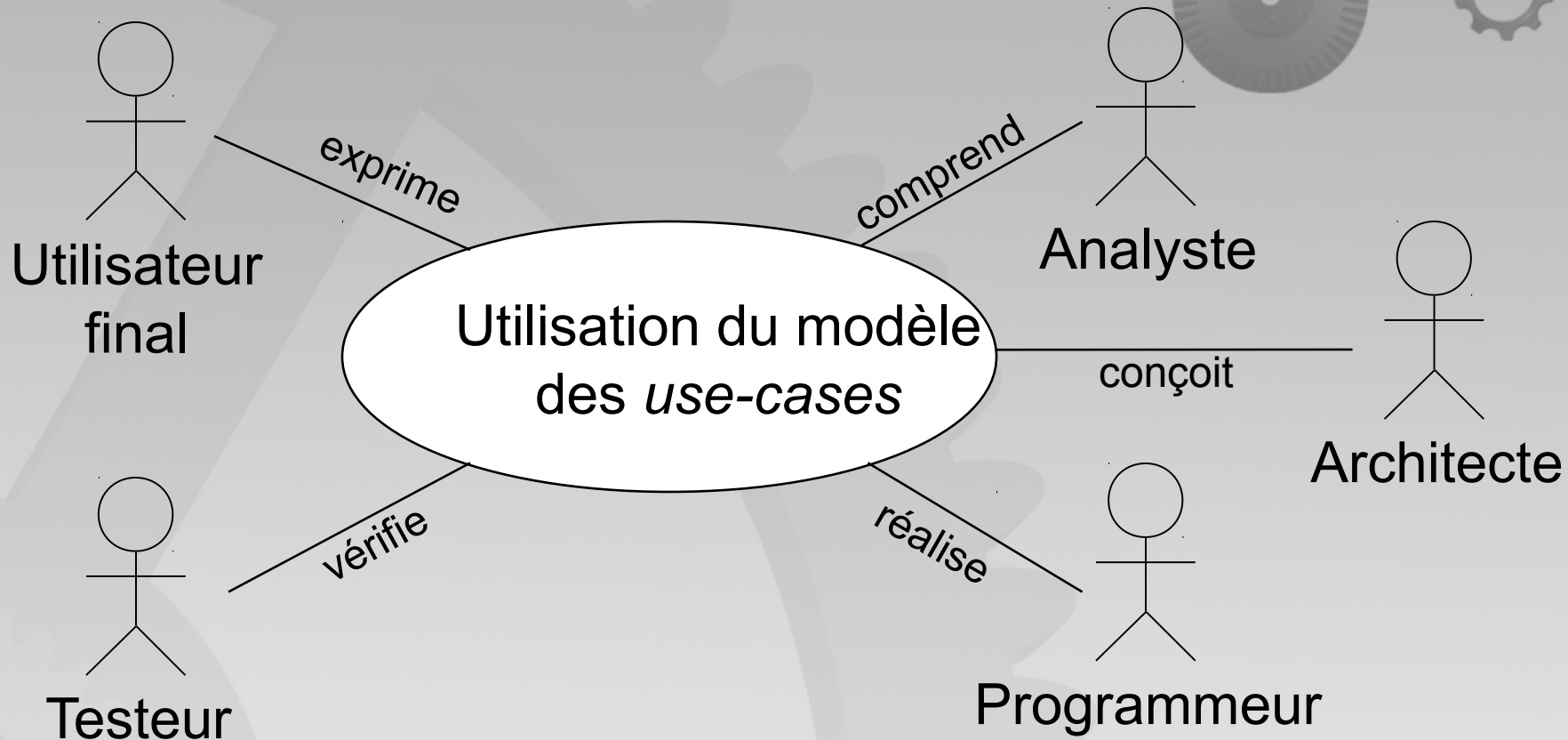


Diagrammes de *Use-cases*

Diagrammes de *use-cases*

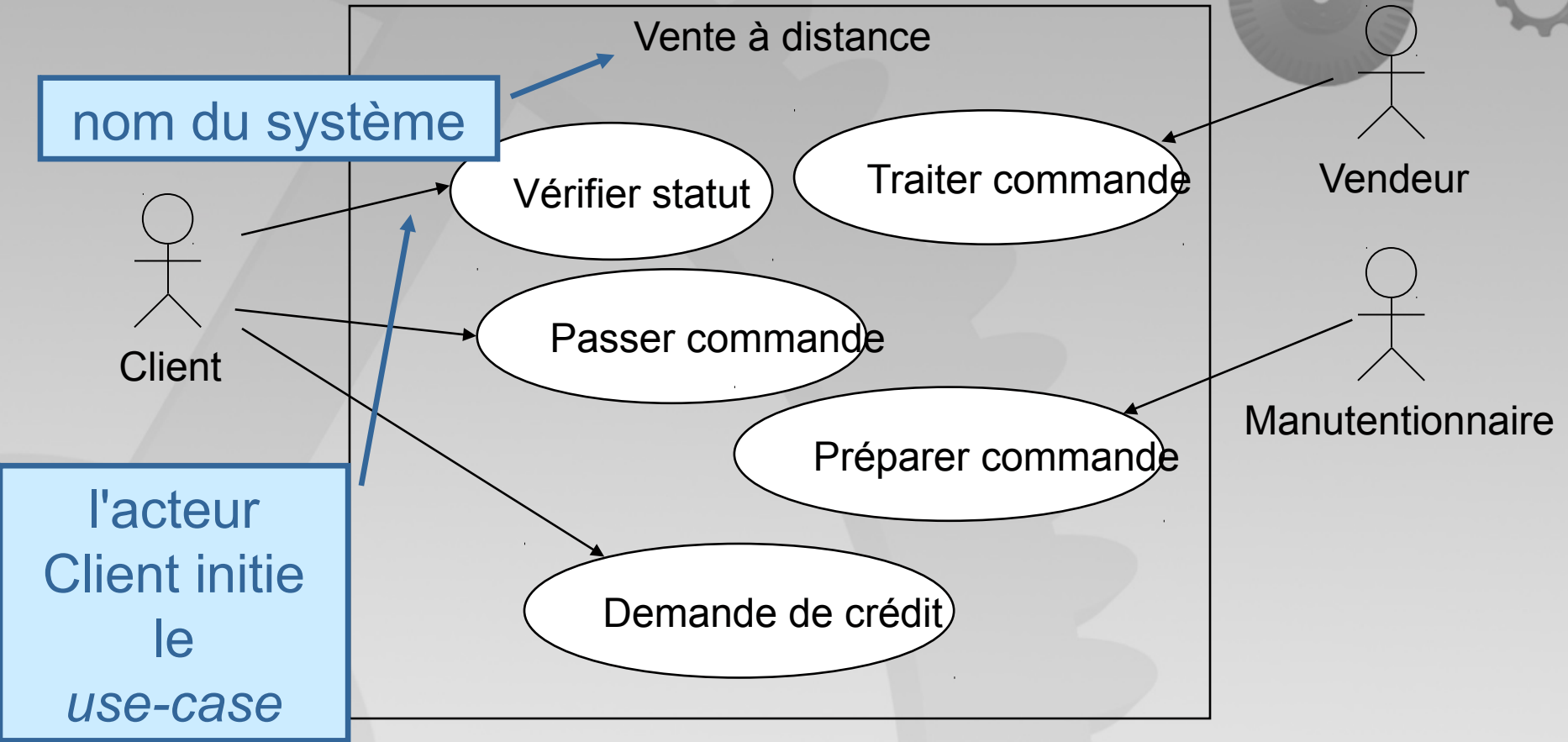
- Un **modèle** de *use-cases* est constitué de la description de chaque acteur et de chaque *use-case* identifiés pour le système, et de diagrammes de *use-cases*.
- Un **diagramme** de *use-cases* décrit les relations entre *use-cases* et les communications entre *use-cases* et acteurs pour la totalité ou une partie du système.
- On peut utiliser plusieurs diagrammes de *use-cases* pour décrire le système à des niveaux de détails différents

Diagrammes de *use-cases*



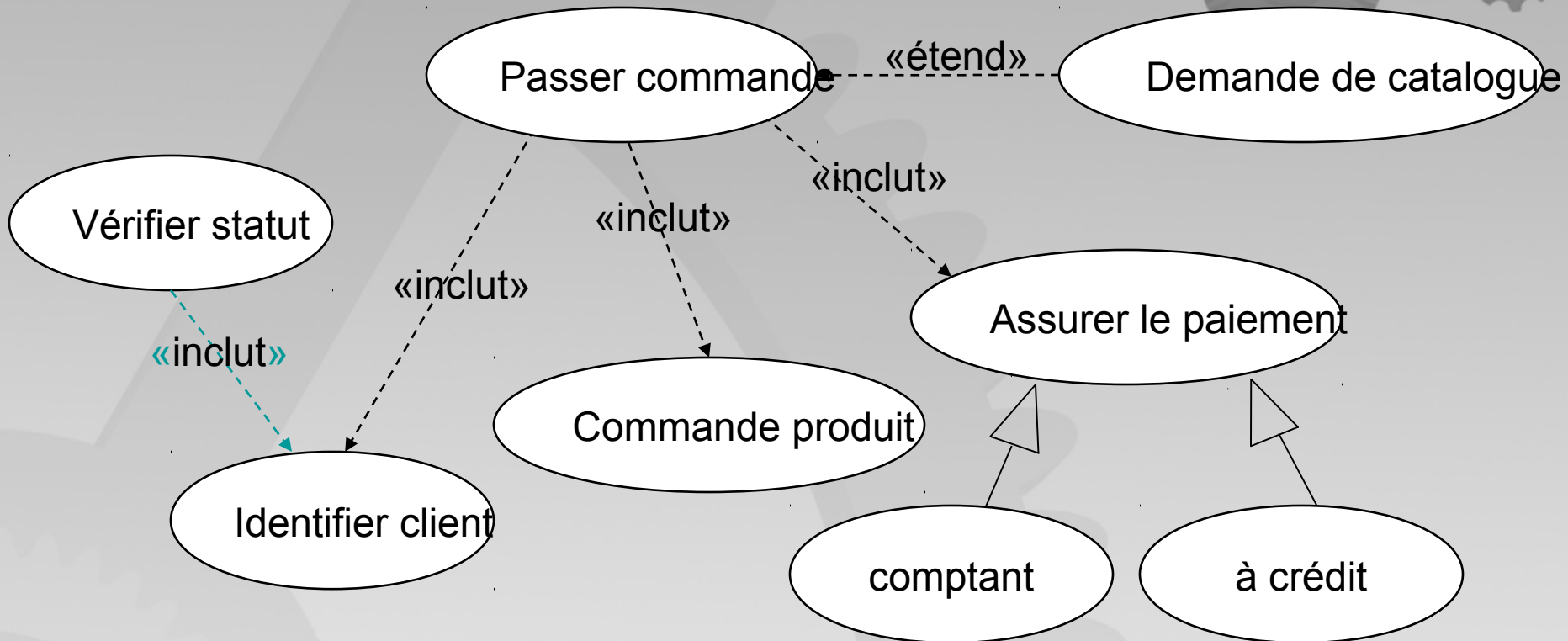
Un diagramme de *use-cases* pour décrire l'utilisation du modèle des *use-cases* tout au long du cycle de développement

Diagrammes de *use-cases*



Un diagramme de *use-cases* pour décrire les fonctionnalités d'un système de vente à distance

Diagrammes de *use-cases*



Un diagramme de *use-cases* de plus bas niveau pour préciser le use-case "Passer commande"

Descriptions de use-cases

Assigne des employés pour travailler sur une campagne

- Peut-être un simple paragraphe

Assigner une équipe pour travailler sur une campagne

- ◆ *Le manager de la campagne désire noter quels employés travaillent sur une campagne spécifique. Cette information est utilisée pour valider des feuilles de pointage d'heures et pour calculer les bonus de fin d'année.*

Descriptions de use-cases

Assigne des employés pour travailler sur une campagne

- Peut être une décomposition pas-à-pas de l'interaction entre acteur et système

Assigner une équipe pour travailler sur une campagne

Action de l'acteur Réponse du système

1. L'acteur tape le nom du client.
2. Liste toutes les campagnes pour ce client.
3. Sélectionne la campagne appropriée.
4. Liste les employés pas encore assignés à cette campagne.
5. Sélectionne les employés
6. Affiche un message confirmant à assigner à cette campagne que les employés ont été alloués.

Chemins différents

Pas 1–3. L'acteur connaît le nom de la campagne et le tape directement.

Spécifications de comportement



- Plutôt que (ou bien en plus de) d'utiliser du texte, un use-case peut-être lié à un autre diagramme qui spécifie son **comportement**
- Typiquement, un diagramme de collaboration, un diagramme de séquences, une machine d'états ou plusieurs d'entre eux

Dessiner des diagrammes de Use-Cases



- Identifier les acteurs et les use-cases
- Etablir des priorités entre les use cases
- Développer chaque use case, en commençant par les prioritaires, en écrivant une description pour chacun
- Ajouter une structure au modèle de use case : généralisation, relations d'inclusion et d'extension, et sous-systèmes

Travailler avec des Use Cases

- Use cases concernent les **fonctionnalités** requises de l'extérieur.
- **Discuter** des use cases avec les utilisateurs du système.
Quels sont les buts de l'utilisateur réel? Considérez les façons alternatives d'atteindre ces buts utilisateurs.
- Ajuster la **granularité** de vos use cases selon la complexité du problème individuel sur lequel vous travaillez, tout en gardant à l'esprit que
 - **trop** de use cases peut être vous surcharger, mais
 - **pas assez de** use cases peut cacher d'important détails.

Travailler avec des Use Cases: planning

Catégoriser les Use Cases

- Les **utilisateurs** devraient indiquer le niveau de **priorité** pour chaque use case.
 - “Je dois absolument avoir cette fonction pour tout système réel.”
 - “Je peux me passer de cette fonction pour une courte période.”
 - “C'est une fonction importante, mais je peux survivre sans elle pendant un moment.”
- Les **développeurs** doivent considérer le risque **architectural**.
 - N'oubliez pas de use cases qui pourront vous demander de refaire beaucoup de travail plus tard pour les incorporer.
 - Concentrez-vous sur les use cases qui sont technologiquement les plus difficiles.
- Les **développeurs** devront être conscient des risques **d'échéances**.
 - “Je suis sûr de savoir combien de temps cela prendra.”
 - “Je peux estimer le temps seulement à un mois-homme près.”
 - “Je n'en ai aucune idée.”

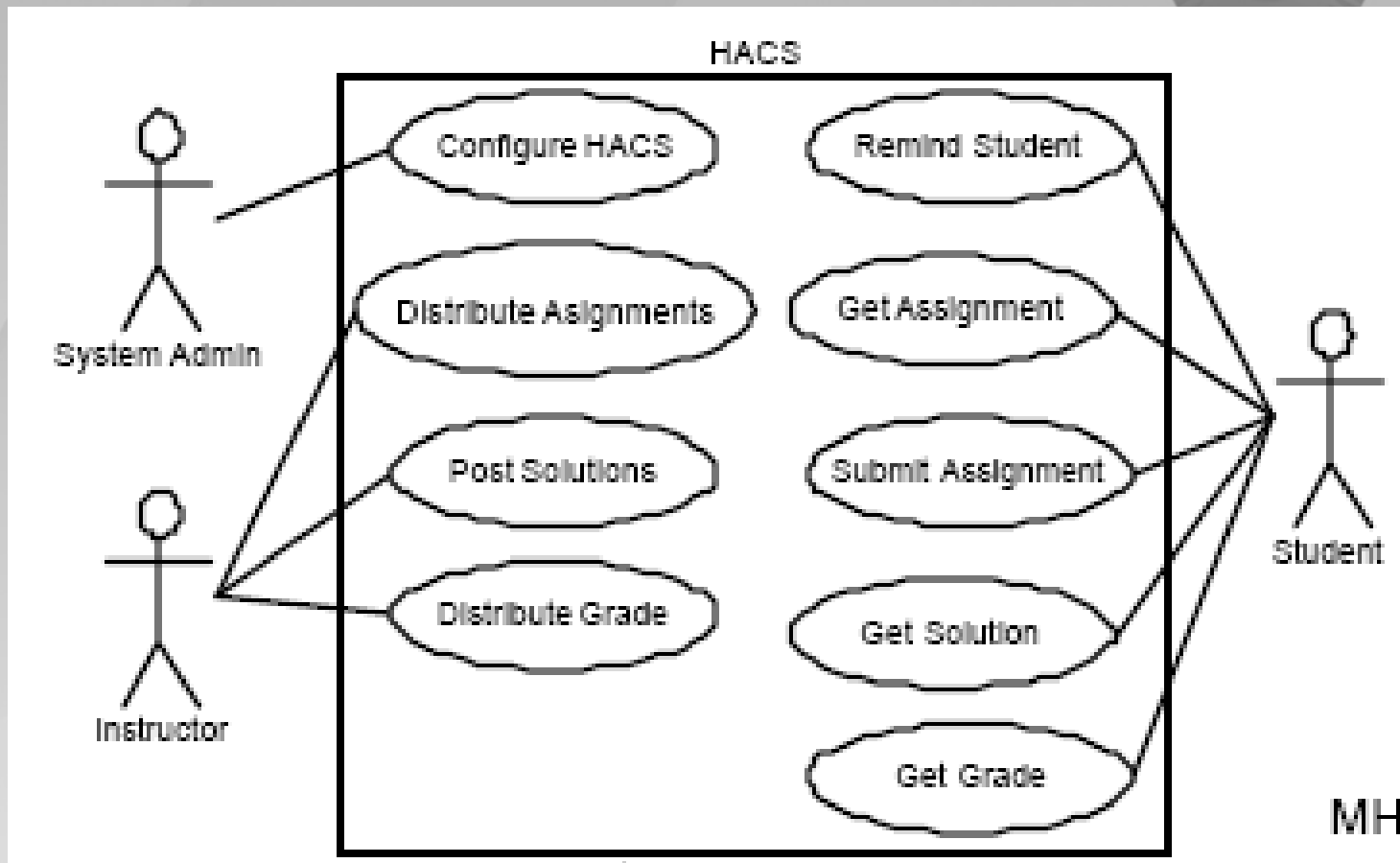
Exercice

Assignment et le ramassage de travail à la maison sont une partie **important** de tout **système éducationnel**. Souvent, cette tâche est entreprise manuellement.

Nous voulons que le **Système de Distribution et de Ramassage de Travail à la Maison (SDRTM)** **automatise ce processus**.

SDRTM sera utilisé par l'enseignant pour distribuer les travaux à la maison, évaluer les solutions des étudiants, distribuer la solution suggérée, et assigner des notes aux étudiants pour chaque devoir.

SDRTM devra aussi aider les étudiants en distribuant automatiquement les énoncés aux étudiants, en procurant un moyen pour que les étudiants soumettent leur solution, en rappelant aux étudiants qu'un devoir est bientôt à rendre, et en rappelant aux étudiants qu'un certain travail devrait déjà être rendu.



MH