Le langage UML

2ème LMD

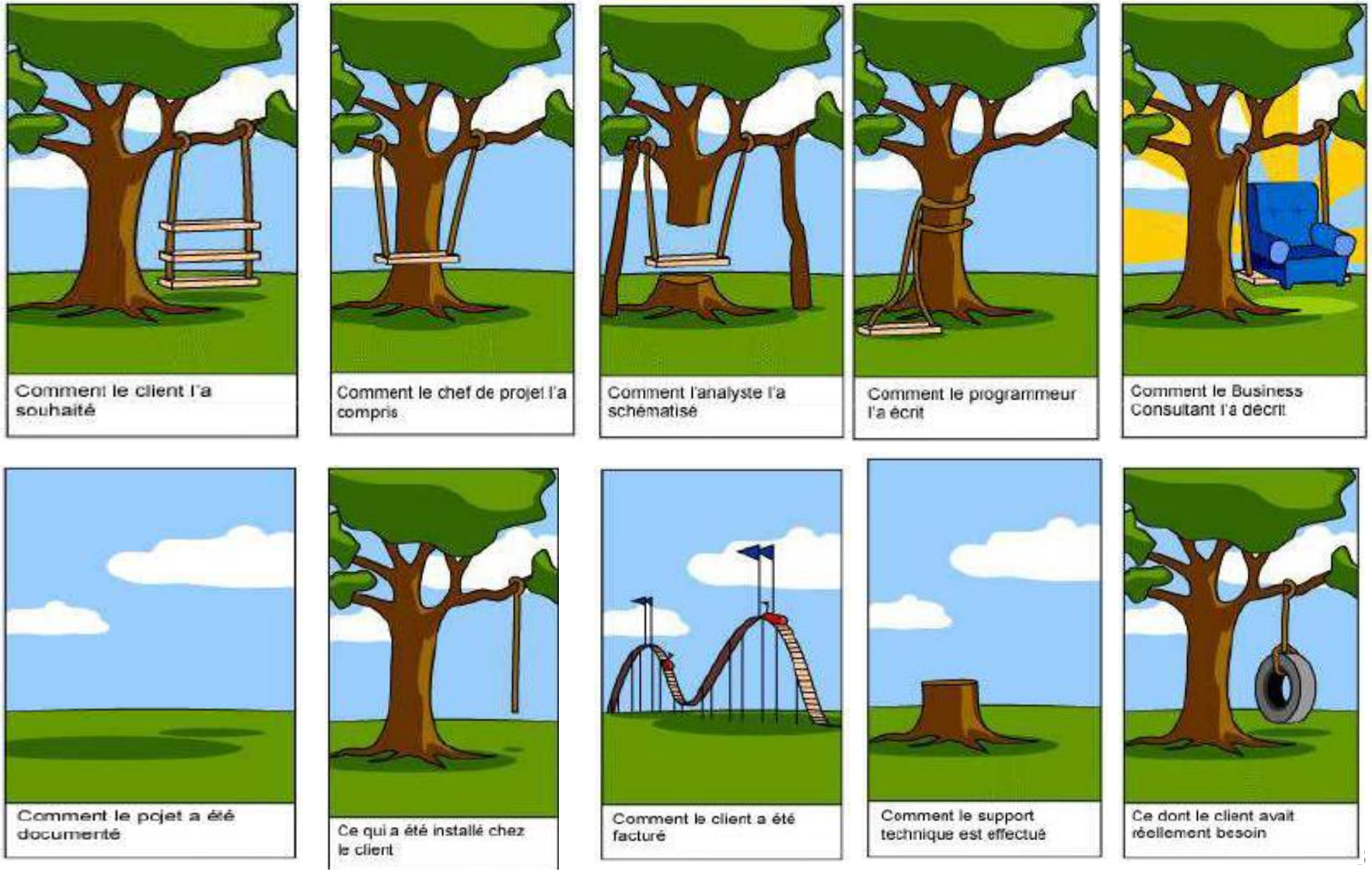
L. Kahloul

2016-2017

plan

- Mais pourquoi documenter?
- Méthodes d’analyse
- Naissance et développement de UML
- Définition de UML
- Entre langage et méthode
- Entre modèle et diagramme, Les diagrammes de UML
- Démarche de conception OO
- Diagrammes de UML: étude détaillée
- Conclusion
- Références

Pourquoi Documenter?



Méthodes d'analyse

Méthodes orientées comportement

- On s'intéresse à la dynamique du système

ex : réseaux de Pétri

Méthodes fonctionnelles :

- On s'inspirent de l'architecture des ordinateurs
- On s'intéresse aux fonctions du système

ex : SADT

Méthodes orientées données :

- On ne s'intéresse pas aux traitements

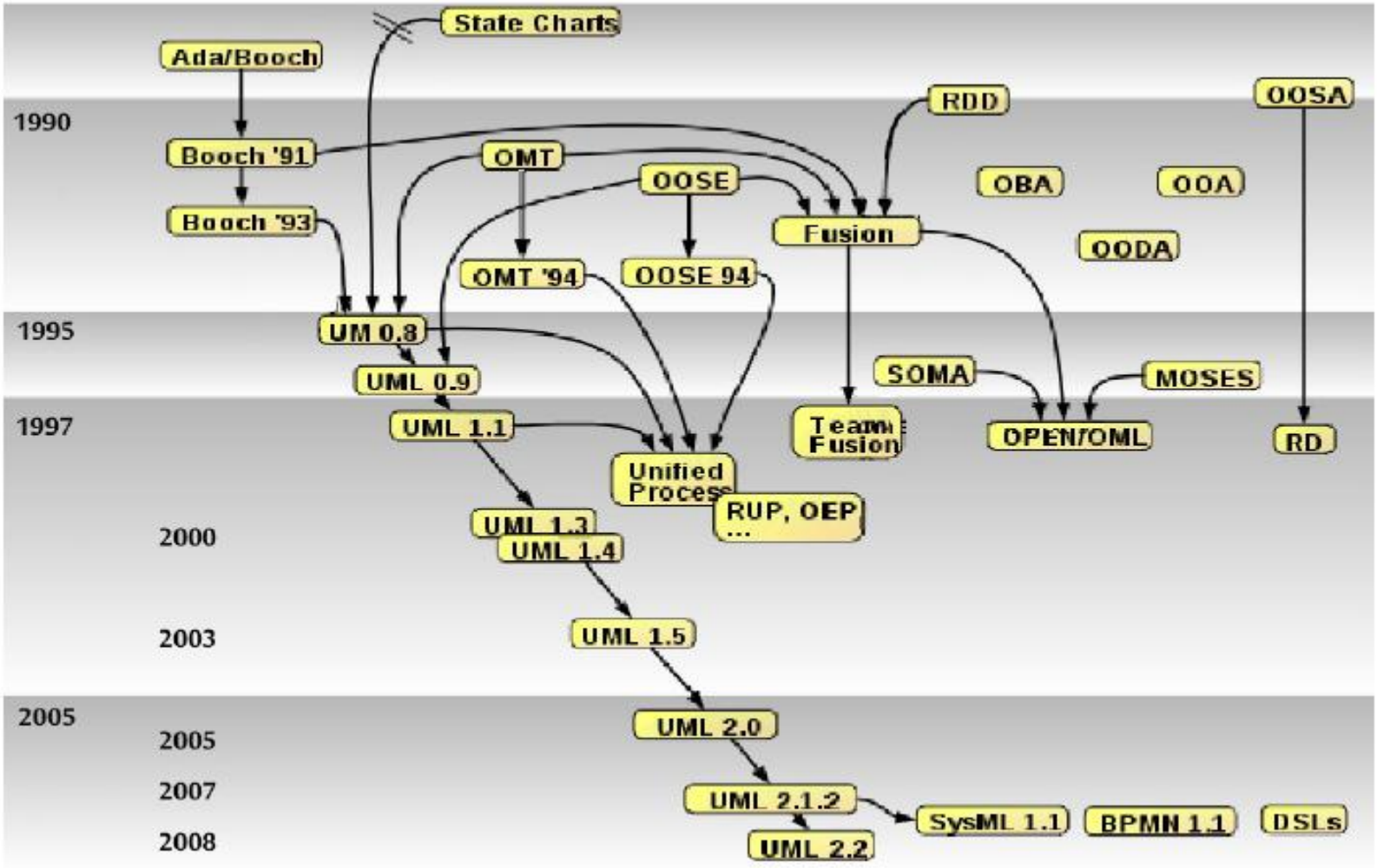
ex : MERISE

Méthodes orientées objets :

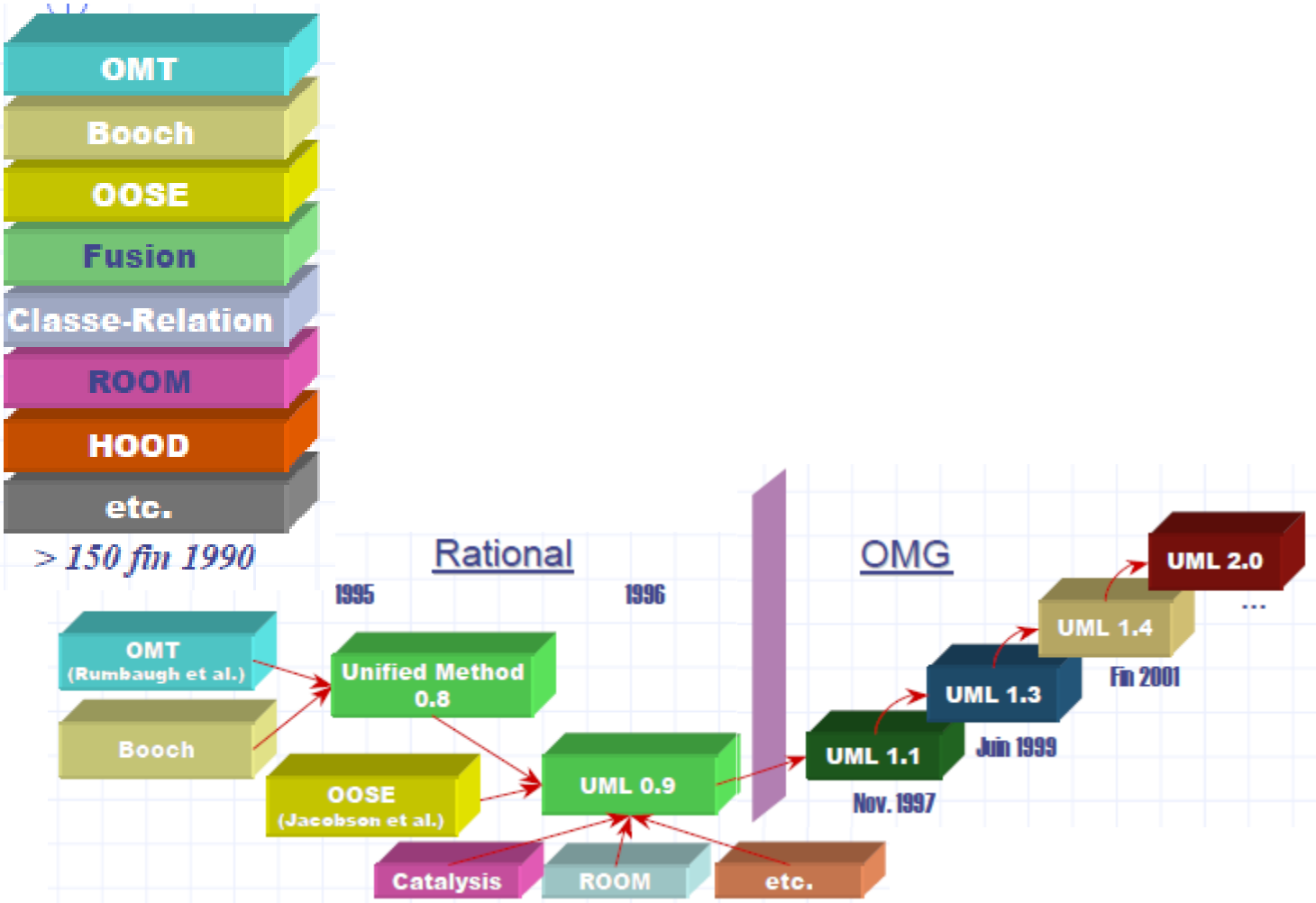
- On ne sépare pas les données et les traitements

ex : Booch, OMT

Naissance UML

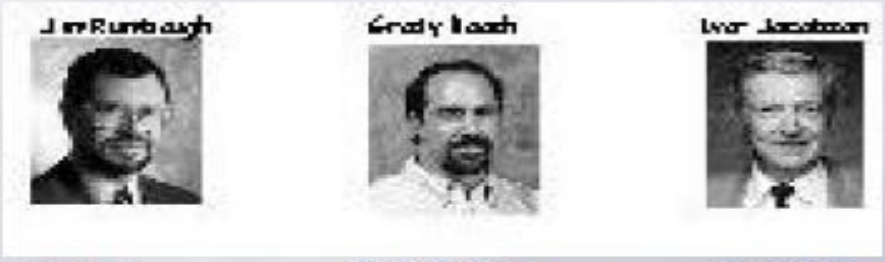


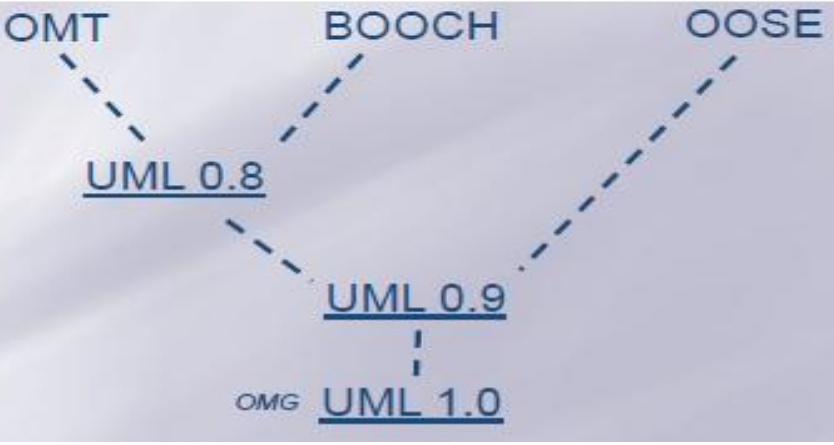
Naissance et développement



Naissance de UML

- Grady BOOCH (BOOCH)
- James Rumbaugh (OMT)
- Ivar Jacobson (OOSE)





UML en 2015

- Maintenu toujours par OMG: http://www.uml.org/
- Version 2.4.1

Définition

- Langage visuel dédié à la **spécification**, la **construction** et la **documentation** des **artefacts** d'un système logiciel, [selon OMG]

UML est un langage

Reste au niveau d'un langage

- ne propose **pas un processus** de développement
- **ni ordonnancement** des tâches,
- **ni répartition** des responsabilités,
- **ni règles** de mise en œuvre

(Certains ouvrages basés sur UML ajoutent cet aspect fondamental en méthodologie)

Modèles d'UML

Modèles descriptifs vs prescriptifs:

- Descriptifs: Décrire **l'existant** (domaine, métier)
- Prescriptifs: Décrire le **futur** système à réaliser

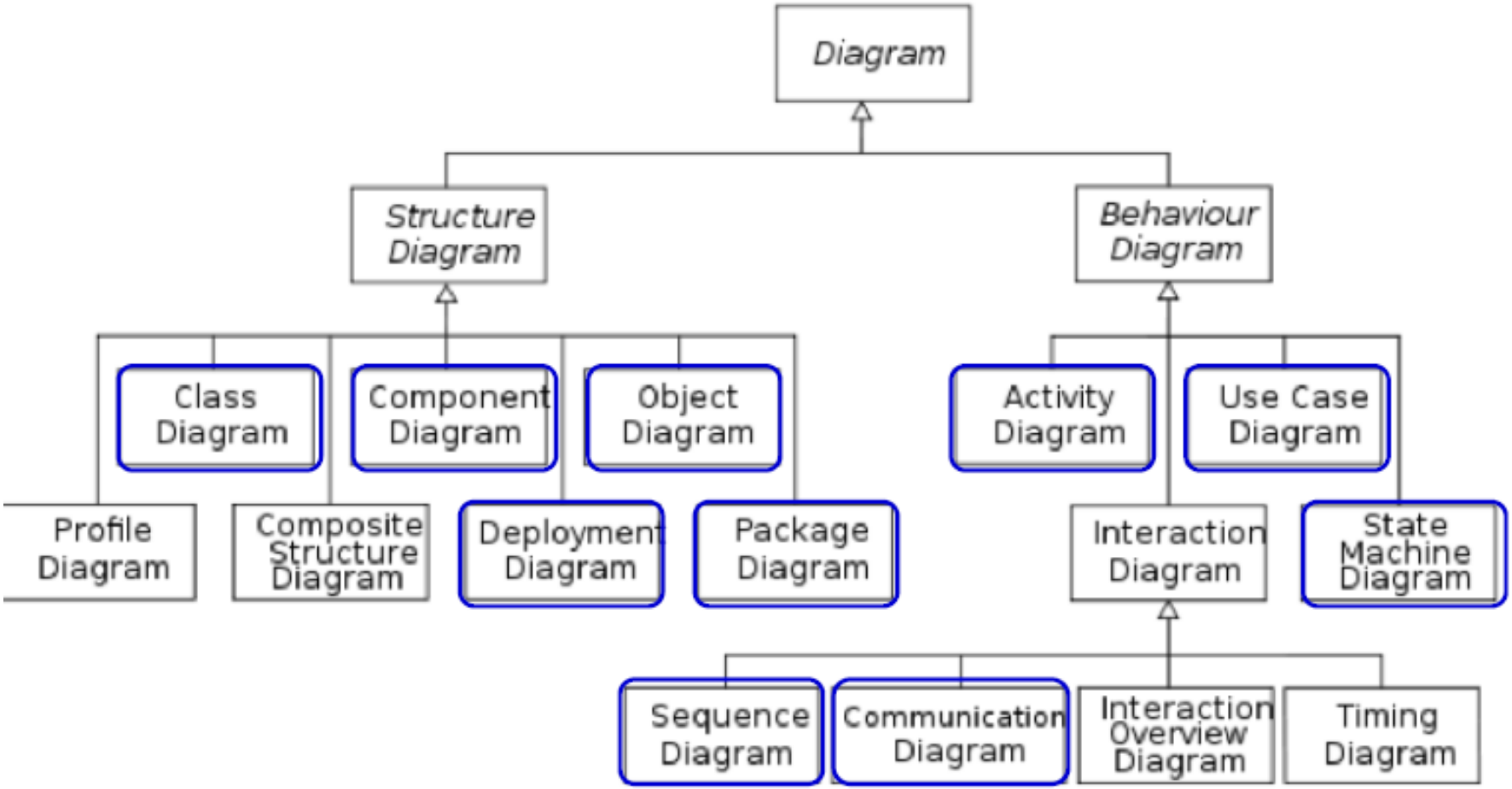
Modèles destinés à différents acteurs:

- Pour l'utilisateur: Décrire le **quoi?**
- Pour les concepteurs/développeurs: Décrire le **comment?**

Modèles statiques vs dynamiques!

- Statiques: Décrire les aspects **structurels**
- Dynamiques: Décrire **comportements et interactions**

Diagrammes UML



9 diagrammes de base

Besoins des utilisateurs:

- **Diagramme des cas d'utilisation**

Structure statique:

- **Diagramme de classes**
- **Diagramme objet**

Dynamique des objets:

- **Diagramme états-transition**
- **Diagramme d'activités**

Interactions entre objets:

- **Diagramme de séquence**
- **Diagramme de collaboration**

Réalisation et déploiement:

- **Diagramme de composants**
- **Diagramme de déploiement**

Démarche possible (**C'est pas de UML !!!**)

- Spécifier le système: **use-cases**
- Modéliser des objets communicants: **objets**, **communication**
- Modéliser la structure de l'application: **Classes**
- Modéliser le **comportement** des objets
- Modéliser les **traitements**
- Modéliser **l'instanciation** de l'application

Diagramme 1: « Use case »

pourquoi?

- Structurer les **besoins** des utilisateurs et les **préoccupations** "réelles" des utilisateurs;
- Montrer les **objectifs** du système;
- Identifier les **utilisateurs** (acteurs) + **interactions** (utilisateur-système);
- Vision générale sur les **fonctionnalités** du système

Diagramme 1: « Use case » modélisation

- Acteur: 
- Cas d'utilisation: 

- Commentaire : 

- Package: 

Diagramme 1: « Use case » concepts de base: **use case**

- **Use case**: **séquences d'actions** réalisées par le système et **produisant un résultat** observable intéressant pour un acteur particulier.

Exemple: Achat billet, Ouverture compte, Livraison Client, Traiter commande, établir facture…

Diagramme 1: « Use case » concepts de base: **acteur**

Acteur:

- **entité externe** qui **agit** sur le système (opérateur, autre système...).
- peut **consulter** |**modifier** l'état du système.
- Le système fournit un service qui correspond au besoin d'un acteur
- Les acteurs peuvent être **classés**

Exemple:

Diagramme 1: « Use case »
liens dans un UCD

- Entre acteurs et use cases: **communication**;
- Entre cas d'utilisation: **généralisation**, **inclusion**, **extension**
- Entre acteurs: **généralisation, extension**.

Diagramme 1: « Use case »
exemple: communication 1

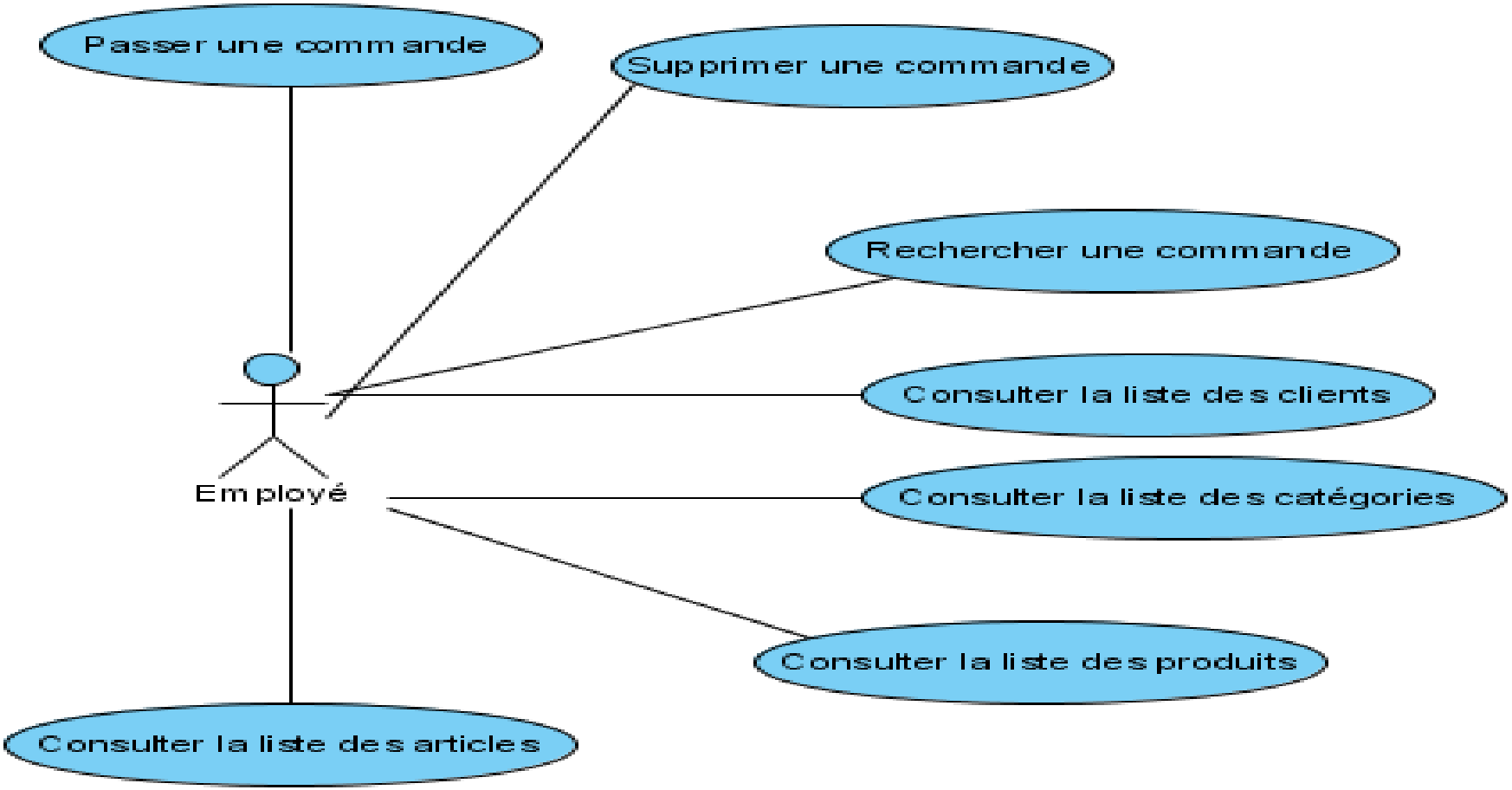


Diagramme 1: « Use case »
exemple: communication 2

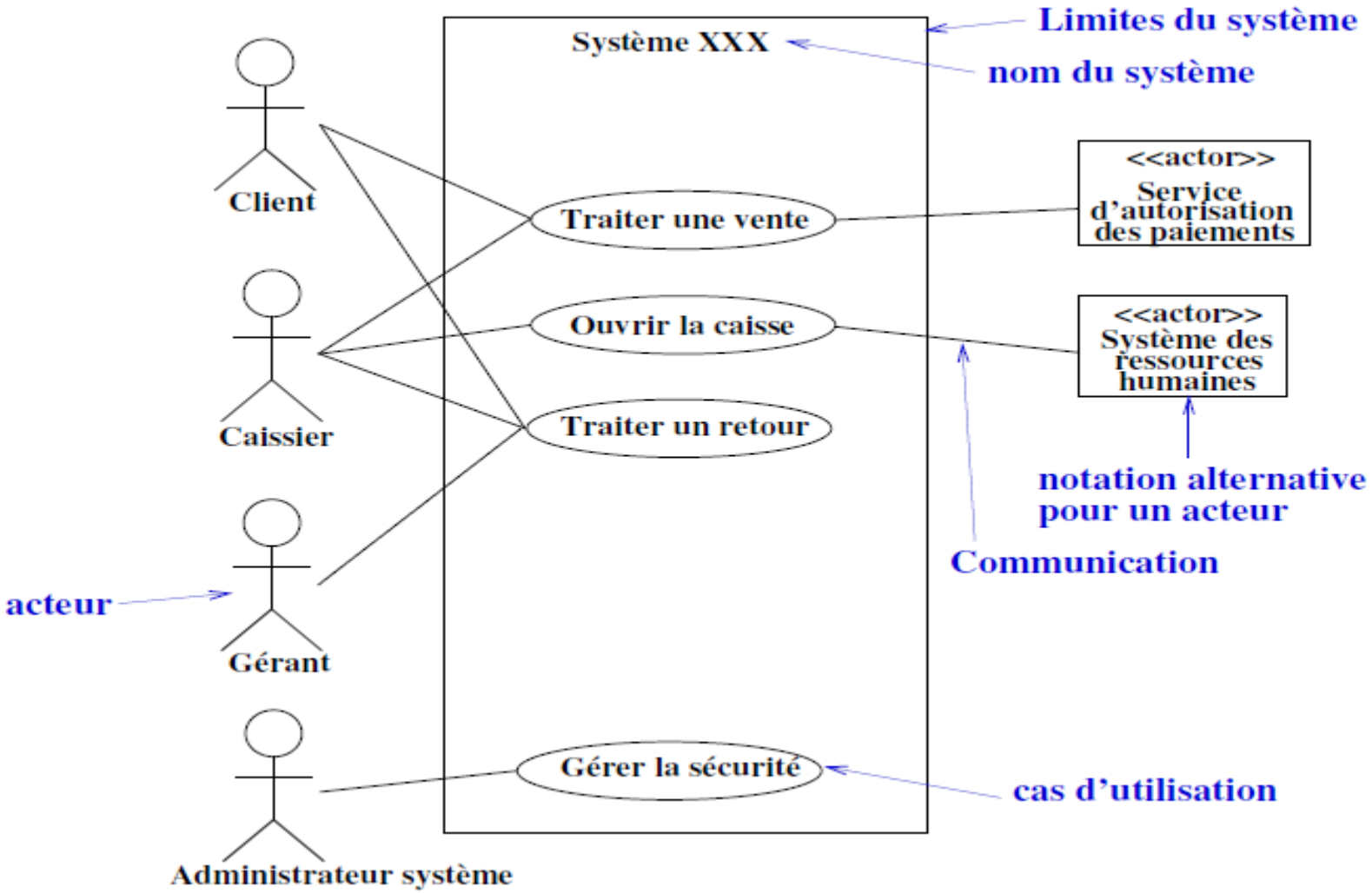
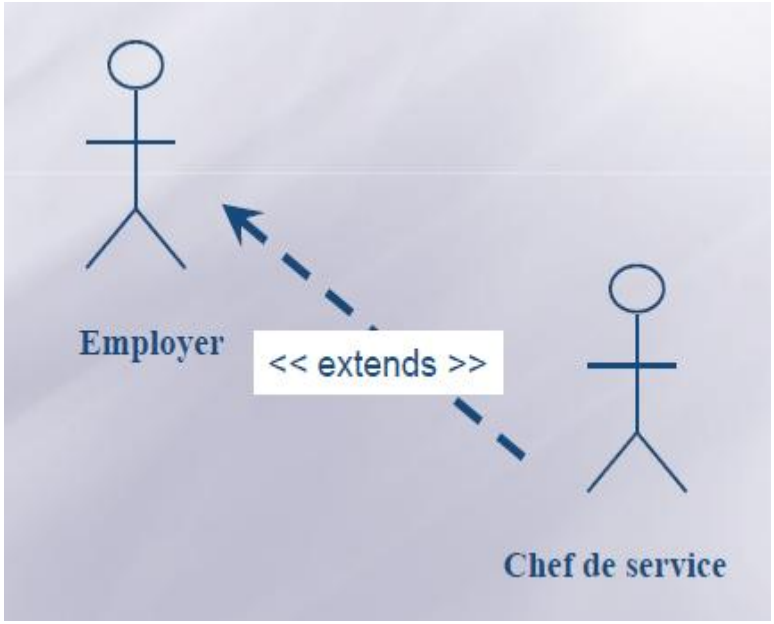


Diagramme 1: « Use case »

Acteurs: généralisation et extension



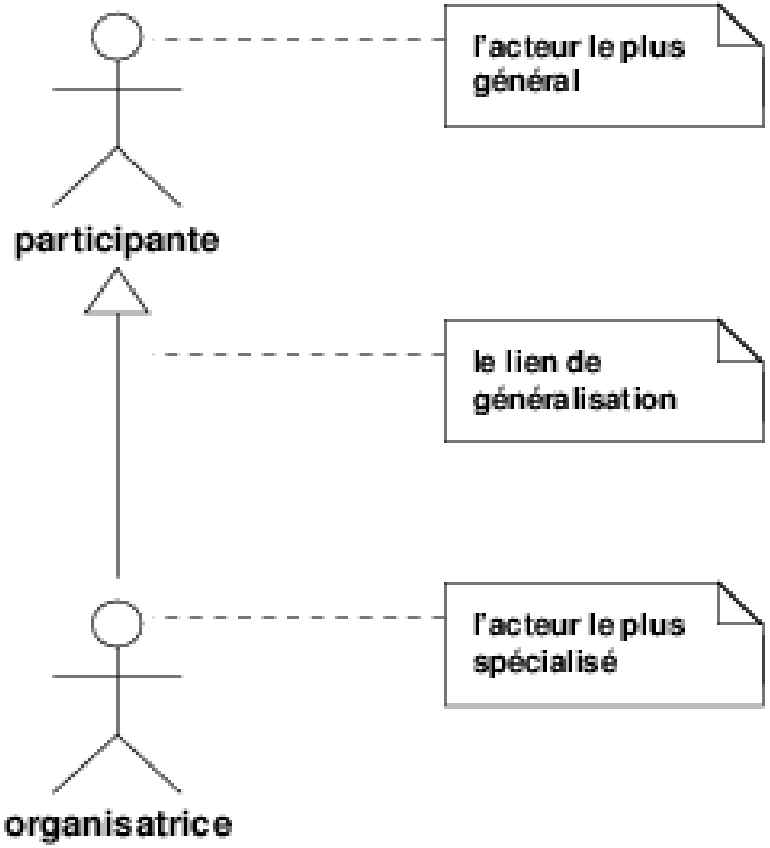


Diagramme 1: « Use case »

exemple: communication et généralisation entre acteurs

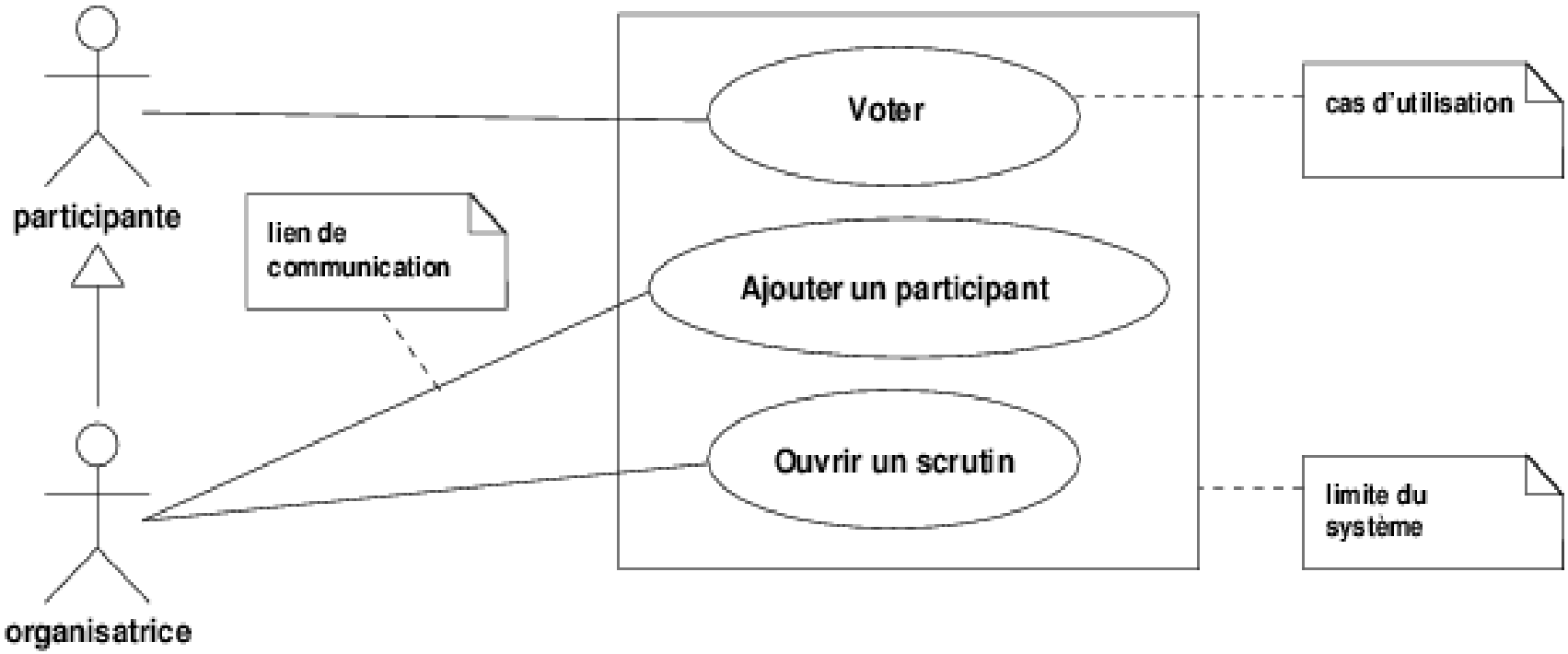
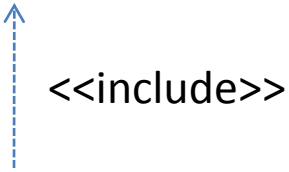


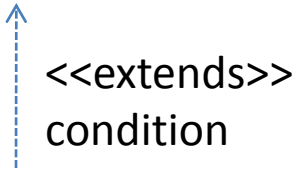
Diagramme 1: « Use case »

liens entre UCs

- **Inclusion:** A inclut B si A contient B dans sa définition.

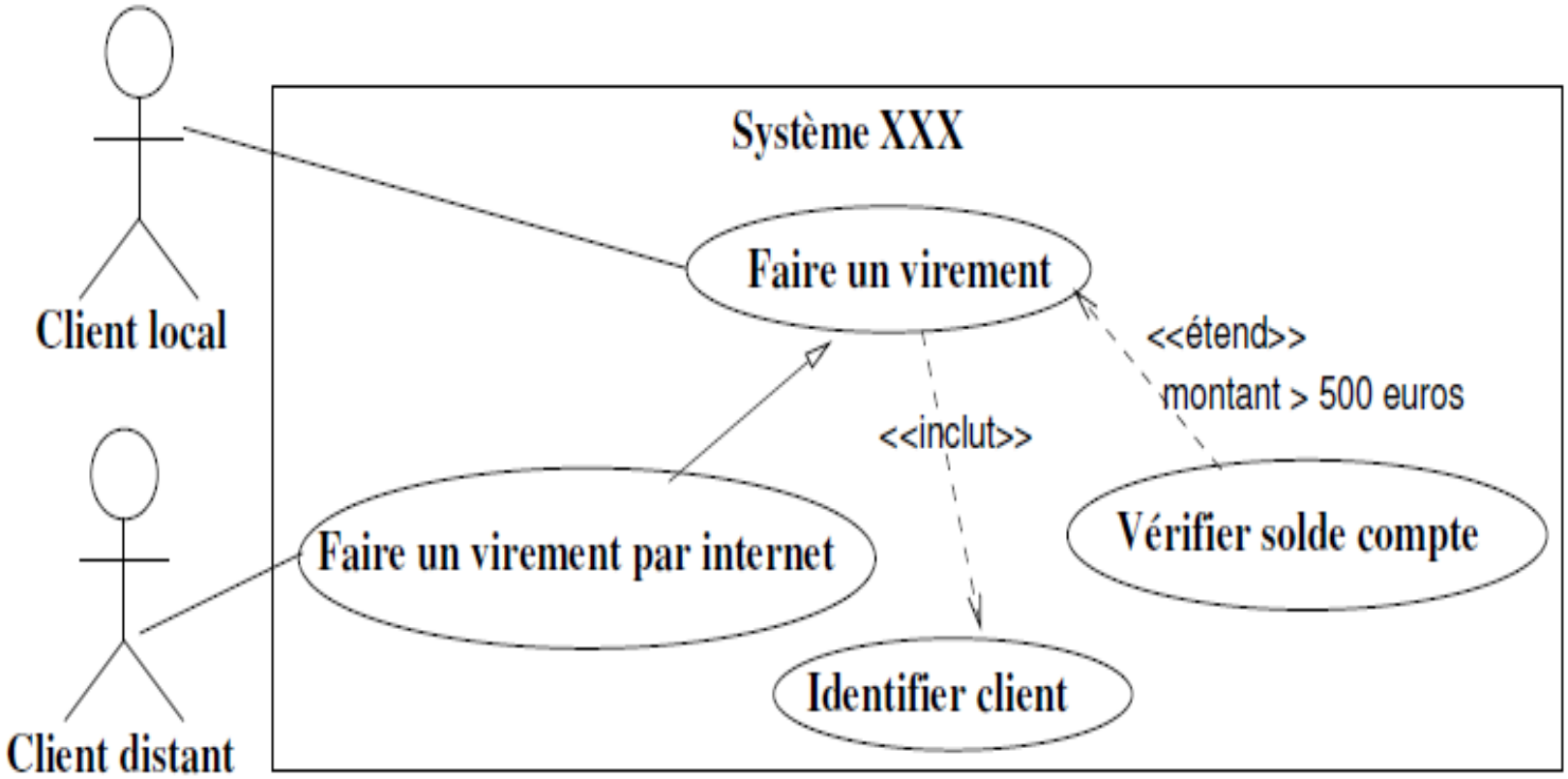


- **Extension:** A étend B si A ajoute des fonctionnalités facultatives à B. On peut avoir une condition que doit vérifier B avant d'avoir l'extension A.



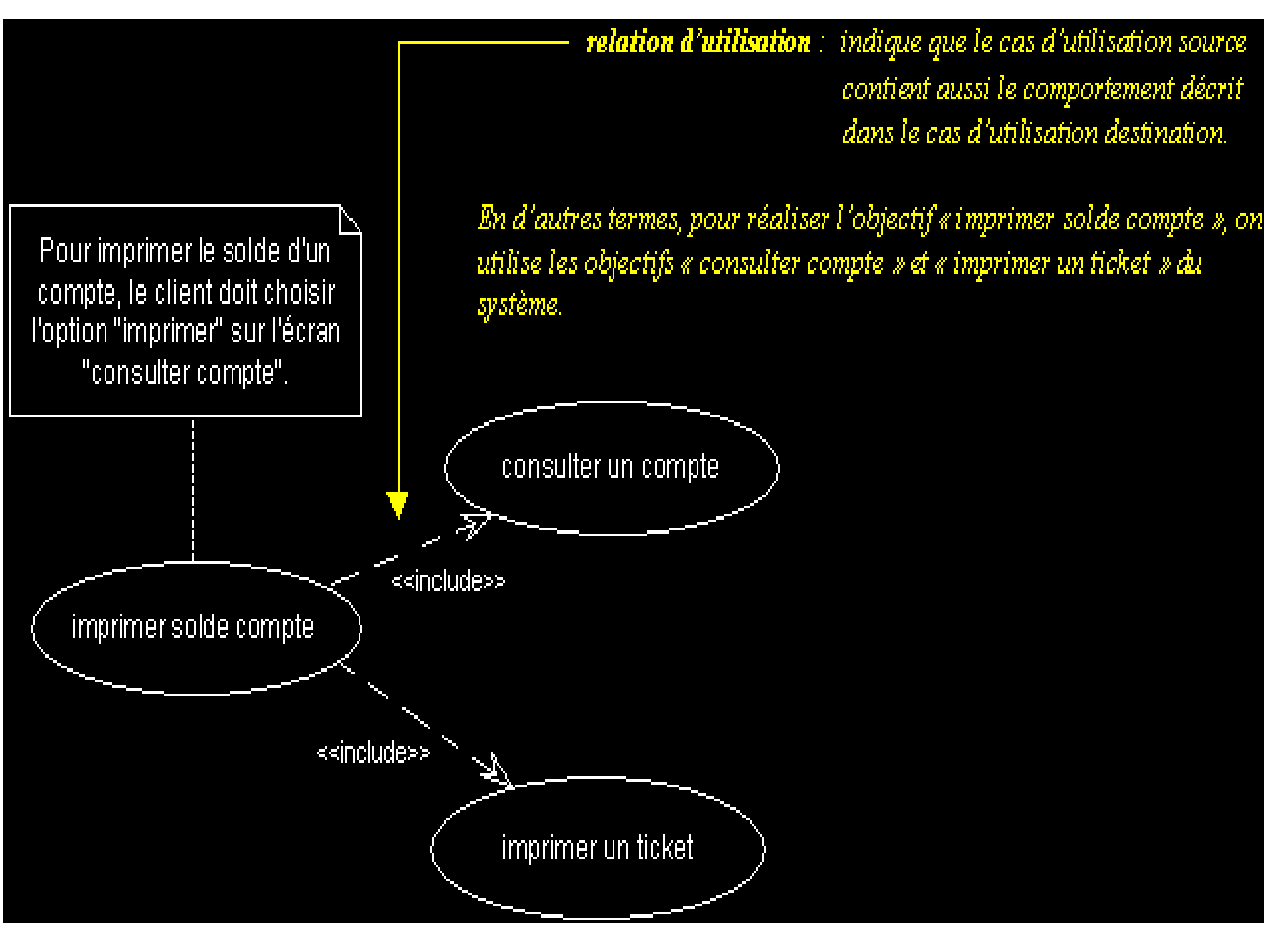
- **Généralisation:** A est une généralisation de B, si A peut être substitué par B pour un cas précis. Spécialisation de certaines actions du cas d'utilisation d'origine

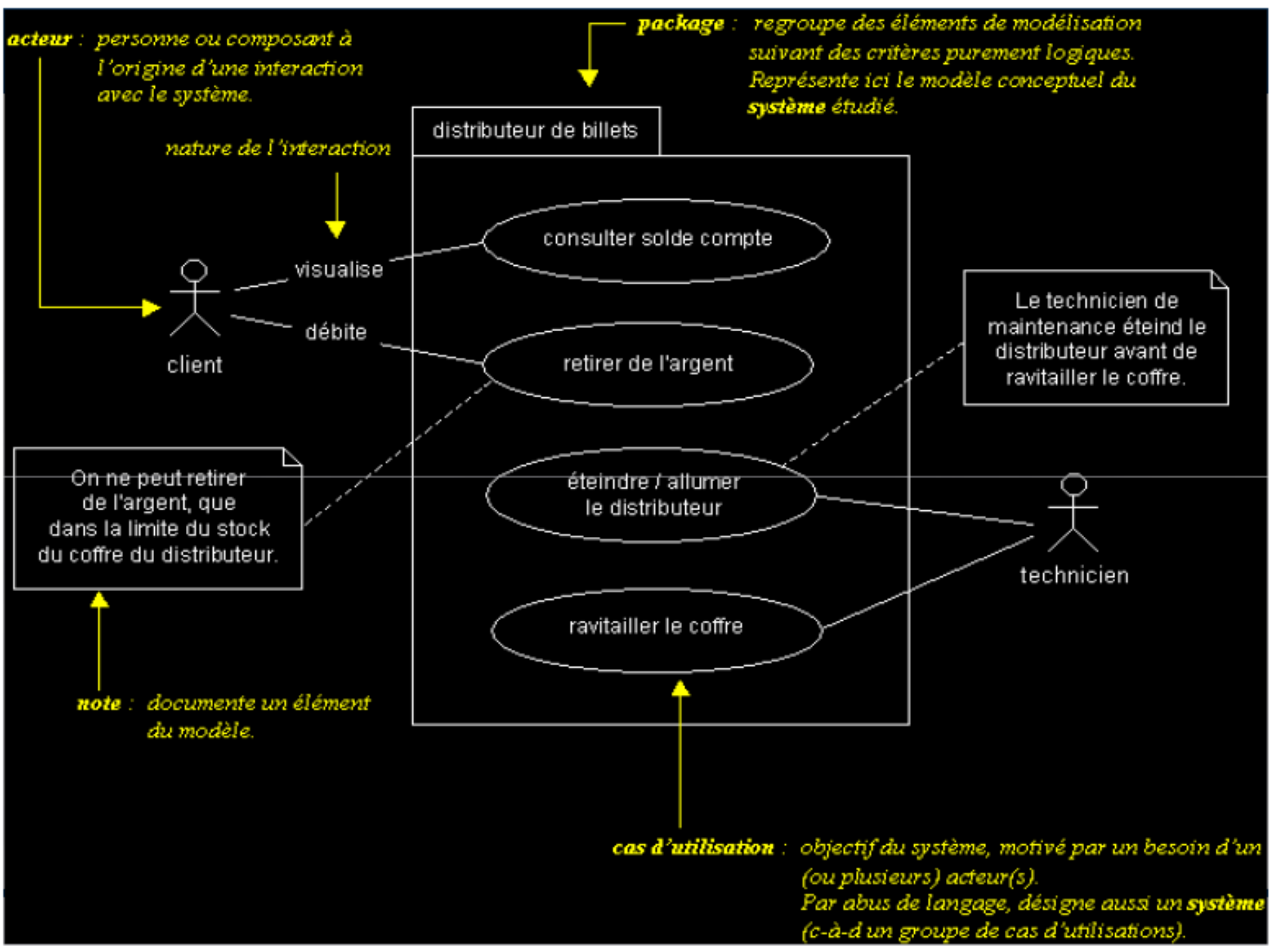
Diagramme 1: « Use case » exemple: extension, inclusion

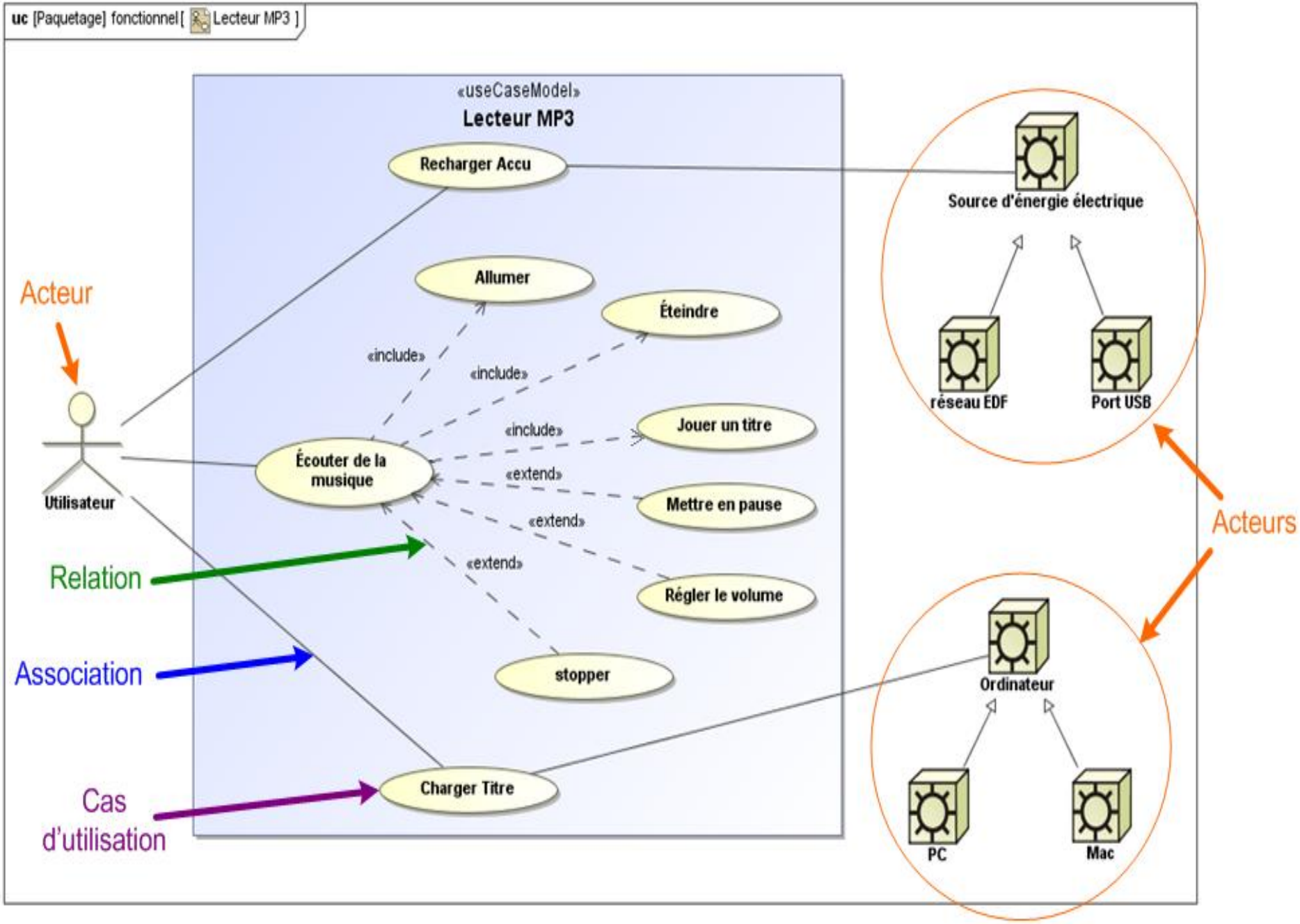


relation d'utilisation : *indique que le cas d'utilisation source contient aussi le comportement décrit dans le cas d'utilisation destination.*

En d'autres termes, pour réaliser l'objectif « imprimer solde compte », on utilise les objectifs « consulter compte » et « imprimer un ticket » du système.




acteur : personne ou composant à l'origine d'une interaction avec le système.
package : regroupe des éléments de modélisation suivant des critères purement logiques. Représente ici le modèle conceptuel du **système** étudié.
distributeur de billets
nature de l'interaction
consulter solde compte
visualise
client
débite
retirer de l'argent
Le technicien de maintenance éteind le distributeur avant de ravitailler le coffre.
On ne peut retirer de l'argent, que dans la limite du stock du coffre du distributeur.
éteindre / allumer le distributeur
technicien
ravitailler le coffre
note : documente un élément du modèle.
cas d'utilisation : objectif du système, motivé par un besoin d'un (ou plusieurs) acteur(s).
Par abus de langage, désigne aussi un **système** (c-à-d un groupe de cas d'utilisations).


uc [Paquetage] fonctionnel [Lecteur MP3]
«useCaseModel»
Lecteur MP3
Recharger Accu
Allumer
Éteindre
«include»
«include»
«include»
Jouer un titre
Écouter de la musique
«extend»
Mettre en pause
«extend»
«extend»
Régler le volume
stopper
Charger Titre
Utilisateur
Source d'énergie électrique
réseau EDF
Port USB
Ordinateur
PC
Mac
Acteur
Relation
Association
Cas d'utilisation
Acteurs

Comment créer un DCU?

Cas d'utilisation: chaque comportement du système attendu par l'utilisateur:

1. **Définir le périmètre du système : Paquetage**
2. **Identifier les acteurs (ceux qui utiliseront le futur logiciel)**
3. **Évaluer les besoins de chaque acteur en CU**
4. **Regrouper ces CU dans un diagramme présentant une vue synthétique du paquetage**
5. **Possibilité de documentation des CU complexes (cahier de charge)**

Remarque: Ne pas descendre trop bas et réinventer l'analyse fonctionnelle