

# **Initiation à UML**

## **AFCEPF**

**[bernard.granier@sfr.fr](mailto:bernard.granier@sfr.fr)**

# UML

- **Introduction**
- **Cas d'utilisation**
- **Diagramme d'activité**
- **Diagramme de classes**
- **Diagramme d'état**
- **Diagramme de séquence**
- **Diagramme de packages**

# UML

- **Diagramme de déploiement**
- **Diagramme de composant**
- **Conclusion**

# Introduction

# Introduction

- **Plan**

- **Objectif du cours**
- **Bibliographie**
- **Qu'est ce qu'UML**
- **A quoi sert UML**
- **Vue d'ensemble**
- **Ce qui n'est pas couvert par UML**

# Introduction

- **Objectif du cours 1/1**
  - **Introduction à UML**
    - Ce n'est pas un cours avancé, ni d'expertise
  - **Fournir les bases essentielles**
  - **Présenter les schémas usuels**
  - **Comprendre la documentation et la bibliographie**

# Introduction

- **Bibliographie 1/1**

- **<http://www.omg.org/> : norme**
- **UML 2.0 de Martin Fowler, ed. Campus Press**
  - Bon livre, didactique, clair
- **Tête la première Analyse et conception orientées objet, ed. O'Reilly**
  - Original, pédagogique, complet

# Introduction

- **Qu'est-ce qu'UML 1/2**
  - **UML : Unified Modeling Language**
  - **UML est un langage**
    - **Ensemble de schémas modélisant les SI**
      - Tous les éléments d'un SI, d'une application
  - **UML sous-tend une méthode**
    - **Mais ce n'est pas une méthode**
    - **Une méthode est à mettre en place (itérative)**



# Introduction

- **Qu'est-ce qu'UML 2/2**
  - **Outils**
    - IBM Rhapsody
    - Modelio
    - Sparx
    - Magic Draw
    - Plug-in UML d'Eclipse : TopCased / Polarsys
    - StarUML / White star UML
    - ObjectAid
    - ArgoUML
    - LabUML, Visual Paradigm

# Introduction

- **À quoi sert une UML 1/3**
  - **À débbugger le logiciel**
    - Une bonne description du logiciel permet de cerner les bugs
  - **A le faire évoluer**
    - Une bonne description du logiciel permet de savoir comment le faire évoluer

Ce sont probablement les points les plus importants

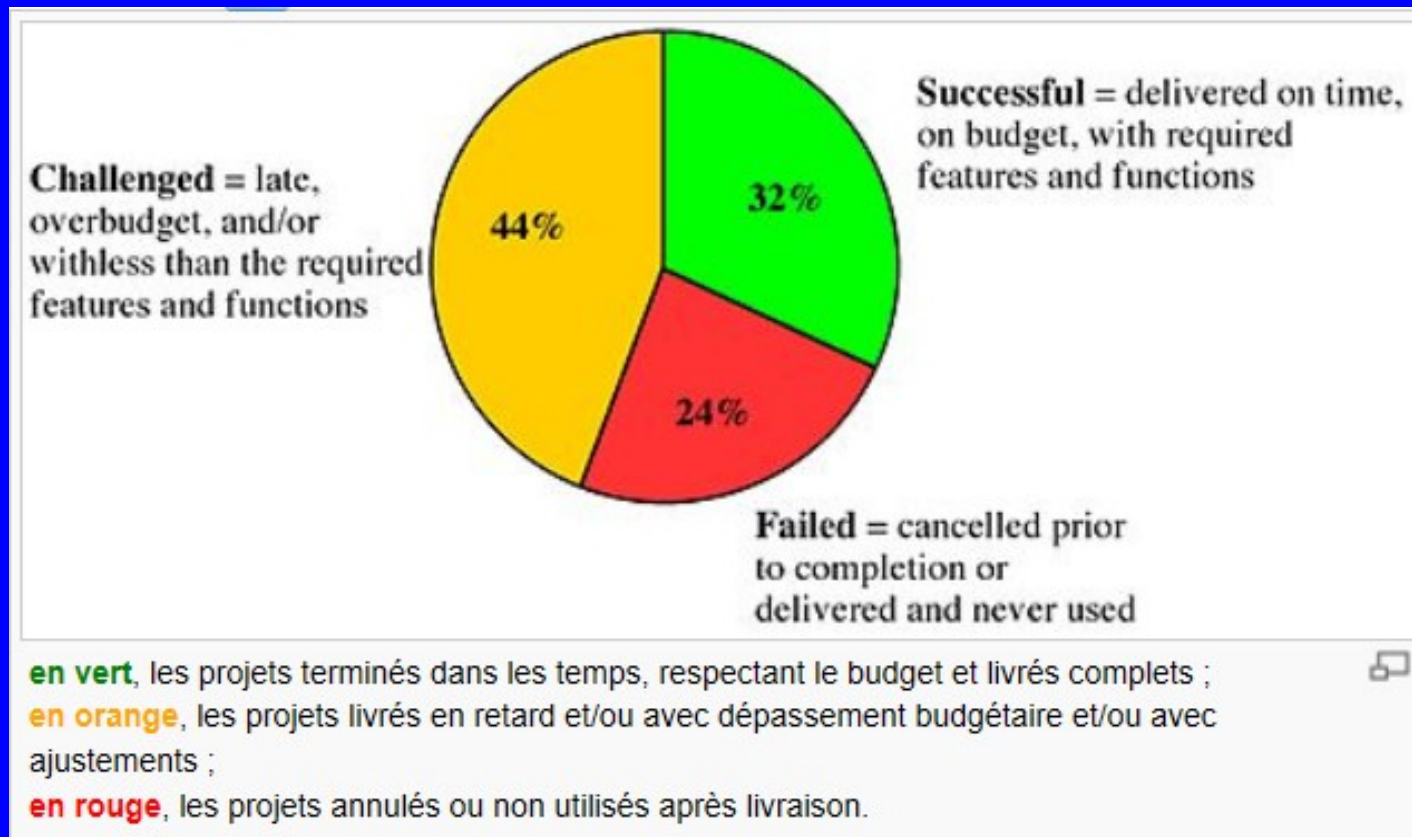


# Introduction

- **À quoi sert une UML 2/3**
  - **A bien concevoir un logiciel**
    - A répondre aux besoins
  - **UML : technique la plus courante dans le monde objet**
    - Utilisable partiellement
    - Sans concurrent dans le monde objet
    - Mais suivant les domaines :
      - Merise
      - SADT
      - SART
      - ...

# Projets informatiques 2/5

- À quoi sert une UML 3/3
  - Standish Group, Chaos report 2009

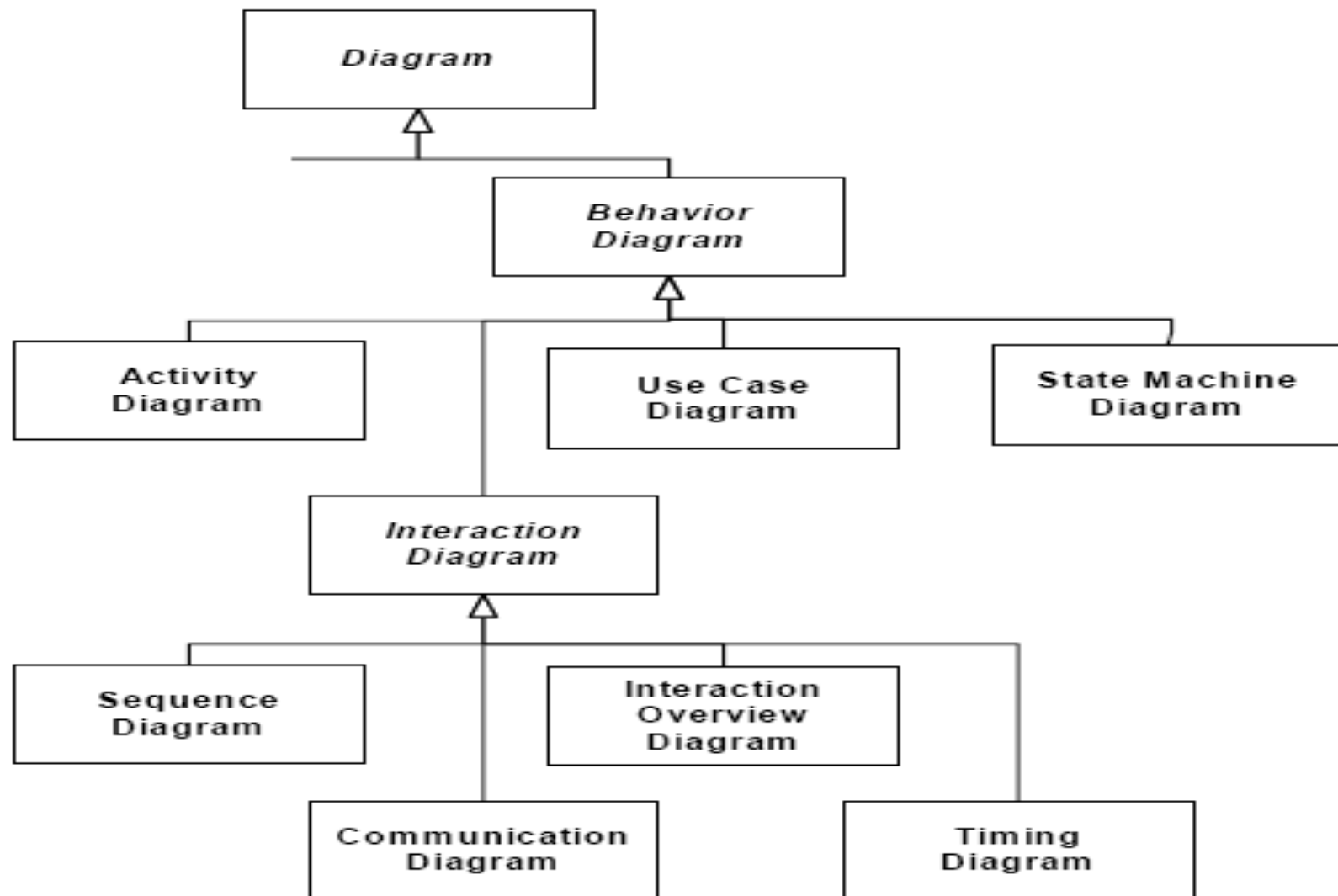


- **Vue d'ensemble 1/2**



# Introduction

- Vue d'ensemble 2/2



# Introduction

- **Ce qui n'est pas couvert par UML ... parce que ce n'est pas son objet 1/2**
  - Règles de codage
  - Gestion de version / configuration
  - Gestion du build
  - Gestion des tests
  - Gestion des traces
  - Gestion des exceptions

# Introduction

- **Ce qui n'est pas couvert par UML ... parce que ce n'est pas son objet 2/2**
  - **La modélisation des IHM**
  - **L'analyse des systèmes**
    - **Point adressé par SysML**
    - **Thème « ancien » ayant ses propres méthodes**
    - **N'utilisant pas UML culturellement**
    - **UML ayant quelques défauts**
      - **Les « preuves mathématiques » d'UML sont à développer**



# Cas d'utilisation

# Cas d'utilisation

- **Plan**

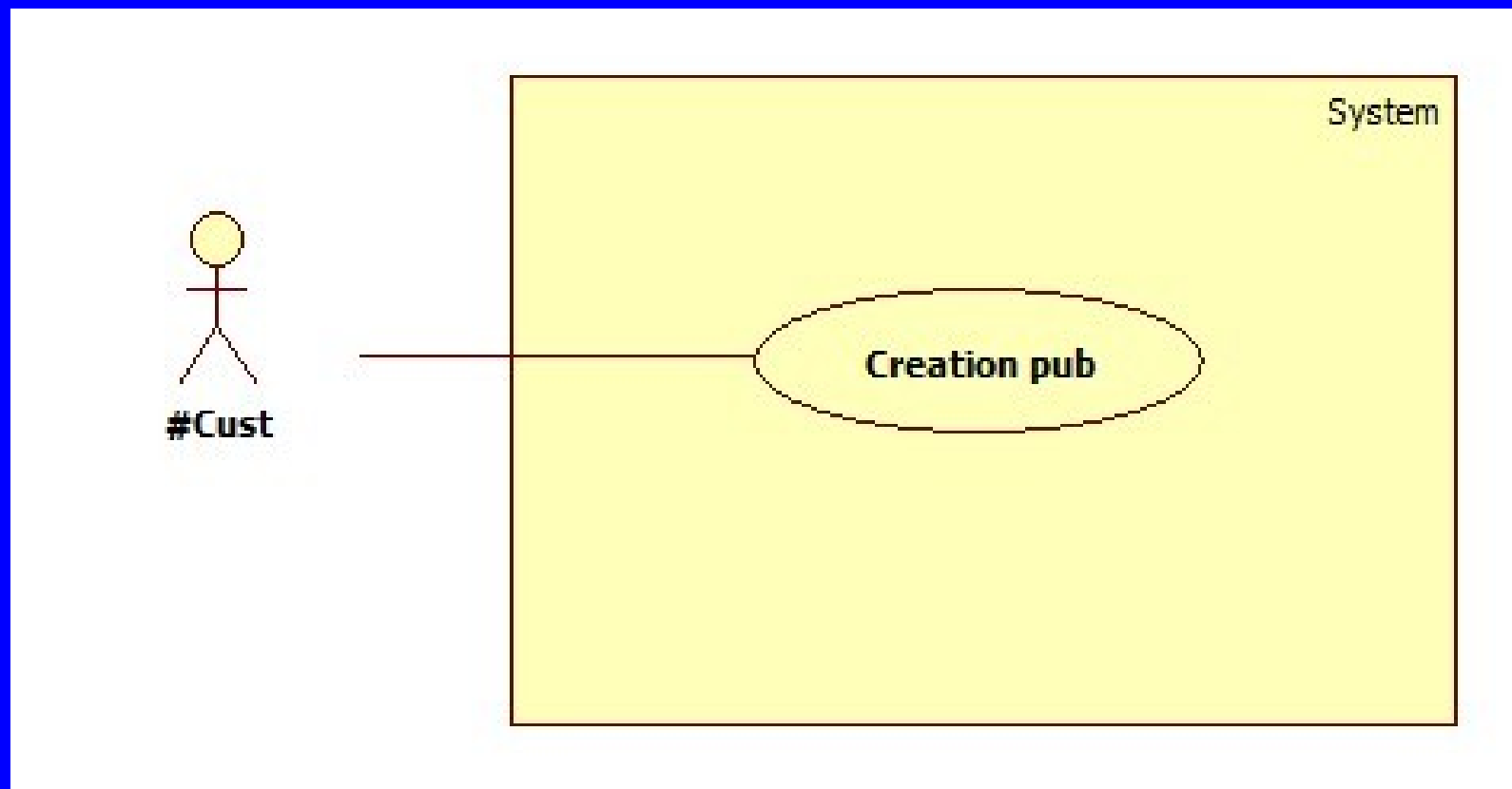
- À quoi cela sert
- Acteurs de cas d'utilisation
- Diagramme de contexte
- Liste de cas d'utilisation
- Description de cas d'utilisation
- Relations entre cas d'utilisation
- Difficultés des cas d'utilisation

# Cas d'utilisation

- **Plan**
  - **Critères d'évaluation**
  - **QCM**

# Cas d'utilisation

- À quoi cela sert 1/6



# Cas d'utilisation

## • À quoi cela sert 2/6

USE-CASE DÉTAILLÉ «CRÉATION PUB»	
Référence	5.2.1
Justification	Un client se connecte au site pour créer sa publicité
Description	1. → #CUST visualise la page d'accueil du site 2. → #CUST s'identifie 3. → #CUST <u>click</u> sur le lien «créer une publicité» 4. → #CUST saisie le texte de sa publicité 5. → #CUST <u>upload</u> une bannière 6. → #CUST saisie la localisation du lieu d'affichage 1. → #CUST se déconnecte
Acteurs	#CUST
Use-cases inclus	N/A
Déclenchement	Après avoir payé son adhésion #CUST saisie une publicité
Pré-conditions	Le site est opérationnel et #CUST est correctement enregistré
<u>Post-conditions</u>	La publicité est stockée
Scénario(s)	Erreur de saisie, taille de la bannière incorrecte, pertes de connexion
Note	N/A

# Cas d'utilisation

- **À quoi cela sert 3/6**
  - **Définissent les besoins des utilisateurs**
    - Les utilisateurs, c'est ce qui utilise le système, pas forcément des personnes
  - **Les cas d'utilisation définissent les spécifications fonctionnelles**
    - Ils racontent une « histoire » d'utilisation du système par un acteur
  - **Définissent les interfaces externes**
  - **Définissent les données échangées**
    - Avec l'extérieur

# Cas d'utilisation

- **À quoi cela sert 4/6**
  - **Mais bien conçus, ils peuvent aussi définir**
    - Les tests d'intégration
    - Le manuel utilisateur
    - Voir les stories de Scrum

# Cas d'utilisation

- À quoi cela sert 5/6



C'est chouette de  
raconter des  
histoires pour  
spécifier un logiciel



Un soft ne répond  
vraiment aux besoins  
qu'après plusieurs  
versions ...



Les clients, ils  
nous ennuiant et  
ils ne savent pas  
ce qu'ils veulent



# Cas d'utilisation

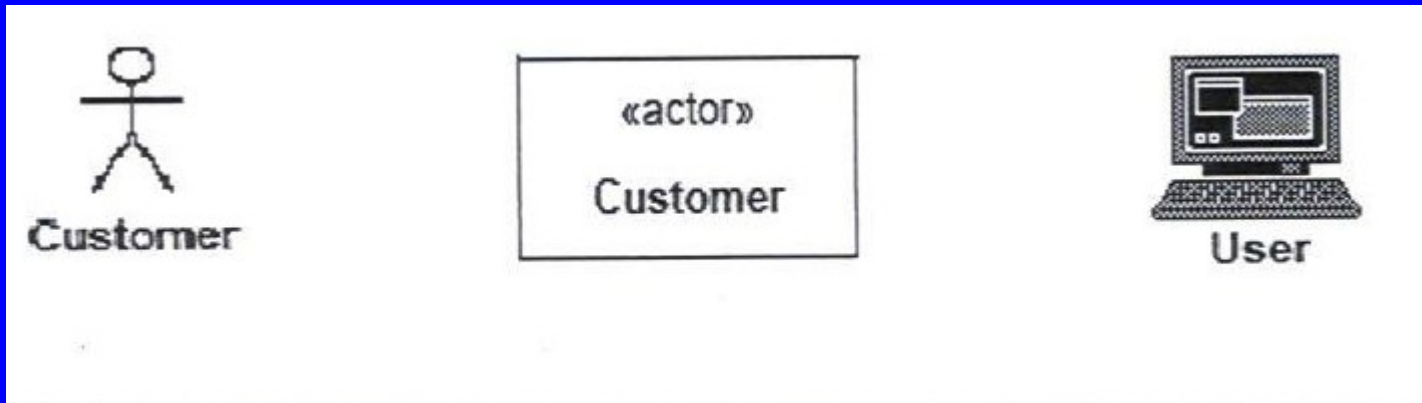
- **À quoi cela sert 6/6**
  - **Deux éléments principaux :**
    - Les acteurs
    - Les cas d'utilisation

# Cas d'utilisation

- **Acteurs de cas d'utilisation 1/5**
  - **Éléments extérieurs interagissant avec le système**
    - **Personne physique, process, serveur, machine**
    - **Définissent les frontières du système**
  - **Deux types**
    - **Primaire s'ils activent un cas d'utilisation**
    - **Secondaire s'ils participent**
  - **Possèdent :**
    - **Un nom, une description, un id**

# Cas d'utilisation

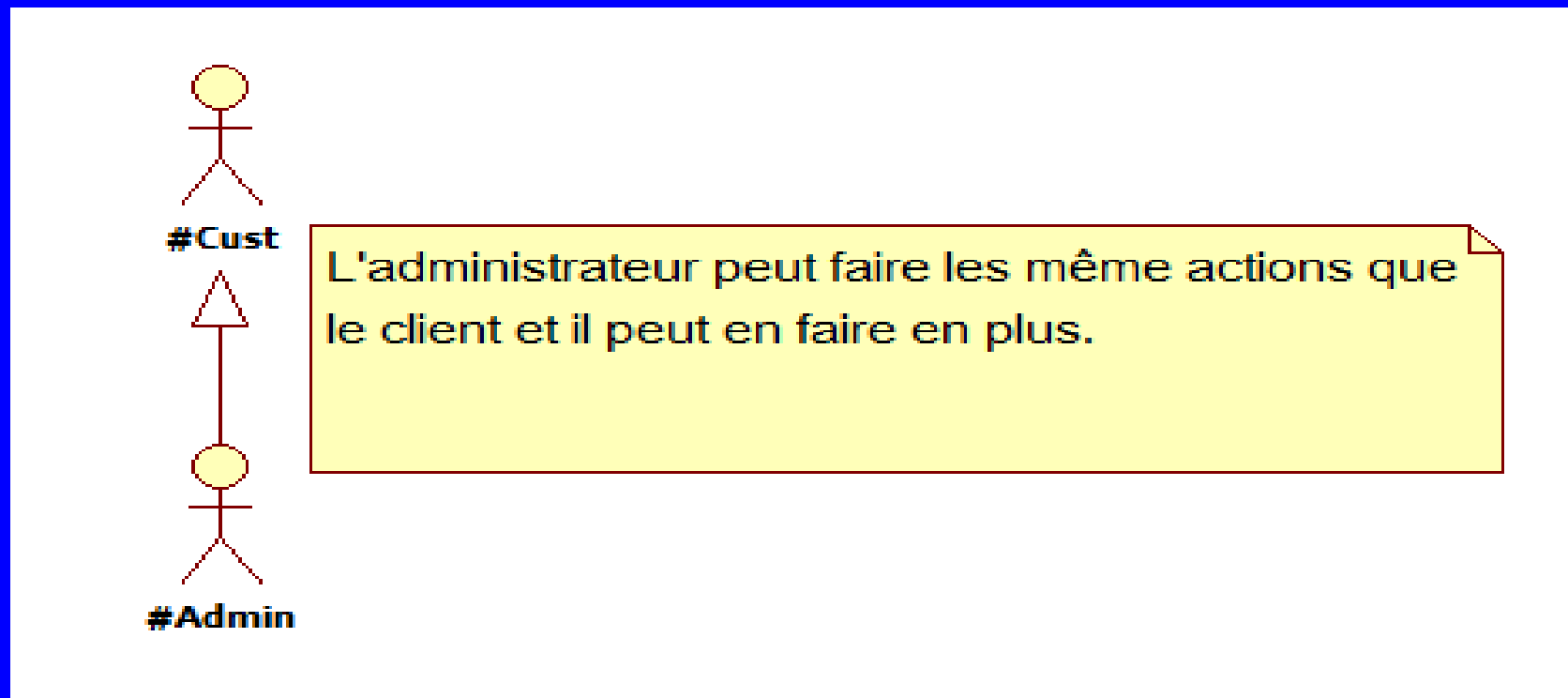
- **Acteurs de cas d'utilisation 2/5**
  - Une abréviation
  - Représentation :



Id	Abréviation	Nom	Description
1	#CUST	Customer	Personne se connectant au site pour acheter un produit

# Cas d'utilisation

- **Acteurs de cas d'utilisation 3/5**
  - Un acteur peut hériter d'un autre acteur si il active les mêmes cas d'utilisation que son père

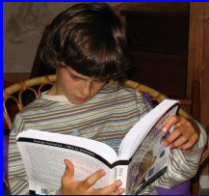


# Cas d'utilisation

- **Acteurs de cas d'utilisation 4/5**
  - **Définir les acteurs**
    - **c'est définir les frontières du système**
      - L'équipe projet ne s'engage que sur l'intérieur du système
    - **C'est ébaucher le modèle de droits du système**
    - **Énumérer les interfaces externes**

# Cas d'utilisation

- Acteurs de cas d'utilisation 5/5



Et si je définis plus d'acteurs qu'il n'en faut ?

Un acteur est justifié par le fait qu'il participe à un cas d'utilisation



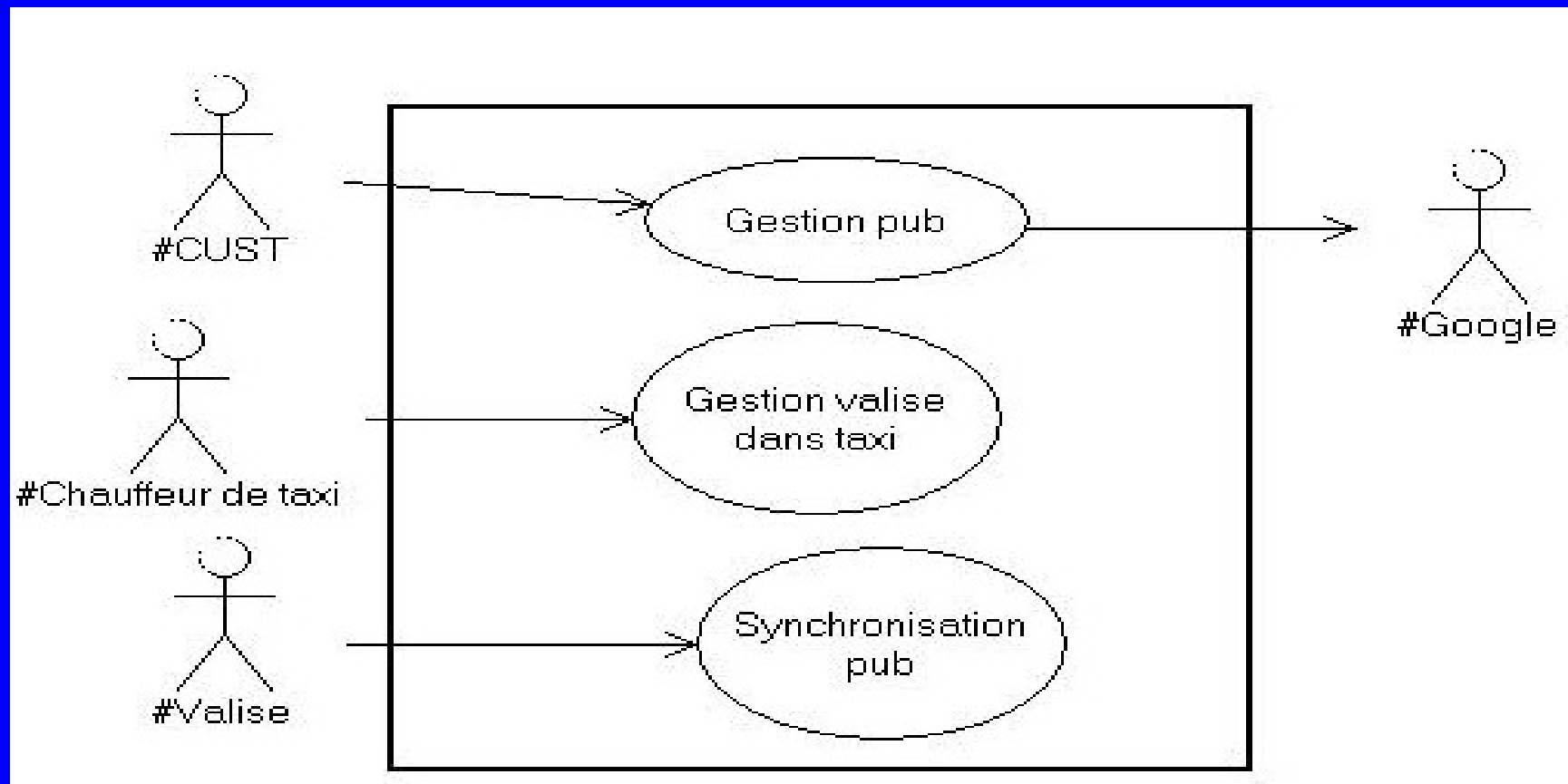
À ce stade, le risque est d'en oublier, donc autant en définir le plus possible et de les filtrer après

# Cas d'utilisation

- **Diagramme de contexte 1/2**
  - Donner une vue de l'application en un schéma
  - Peut ne pas être complet
  - Ne fait pas partie d'UML
    - Mais existe dans beaucoup de méthode
    - Deux possibilités en UML
      - Détourner un diagramme de use cases
      - Utiliser un diagramme de communication

# Cas d'utilisation

- Diagramme de contexte 2/2





# Cas d'utilisation

- **Liste de cas d'utilisation 1/4**
  - **Après avoir trouvé les acteurs : faire une liste des cas d'utilisation**
    - Cette liste doit être exhaustive
    - Si elle est longue, il faut regrouper les cas d'utilisation par thème
  - **Pour établir cette liste, tous les aspects du système doivent être pris en compte**
    - Administration, backup
    - Surveillance
    - Rapports

# Cas d'utilisation

- **Liste de cas d'utilisation 2/4**
  - Cette liste doit être lisible
  - Son objectif : fournir en une lecture simple une vue complète du système
  - Dans la liste, chaque cas d'utilisation possède
    - Un Id
    - Un acteur principal
    - Un nom
    - Une description sommaire
  - La Liste peut être établie à partir des exigences

# Cas d'utilisation

- Liste de cas d'utilisation 3/4

Id	Acteur	Nom	Description
1	#CUST	Connection	Le client se connecte au site
2	#CUST	Recherche rapide	Le client effectue une recherche de produit en saisissant un mot clé

- La liste contient les évolutions possibles, probables, improbables, ce qui va dépendre de clients spécifiques

# Cas d'utilisation

- Liste de cas d'utilisation 4/4



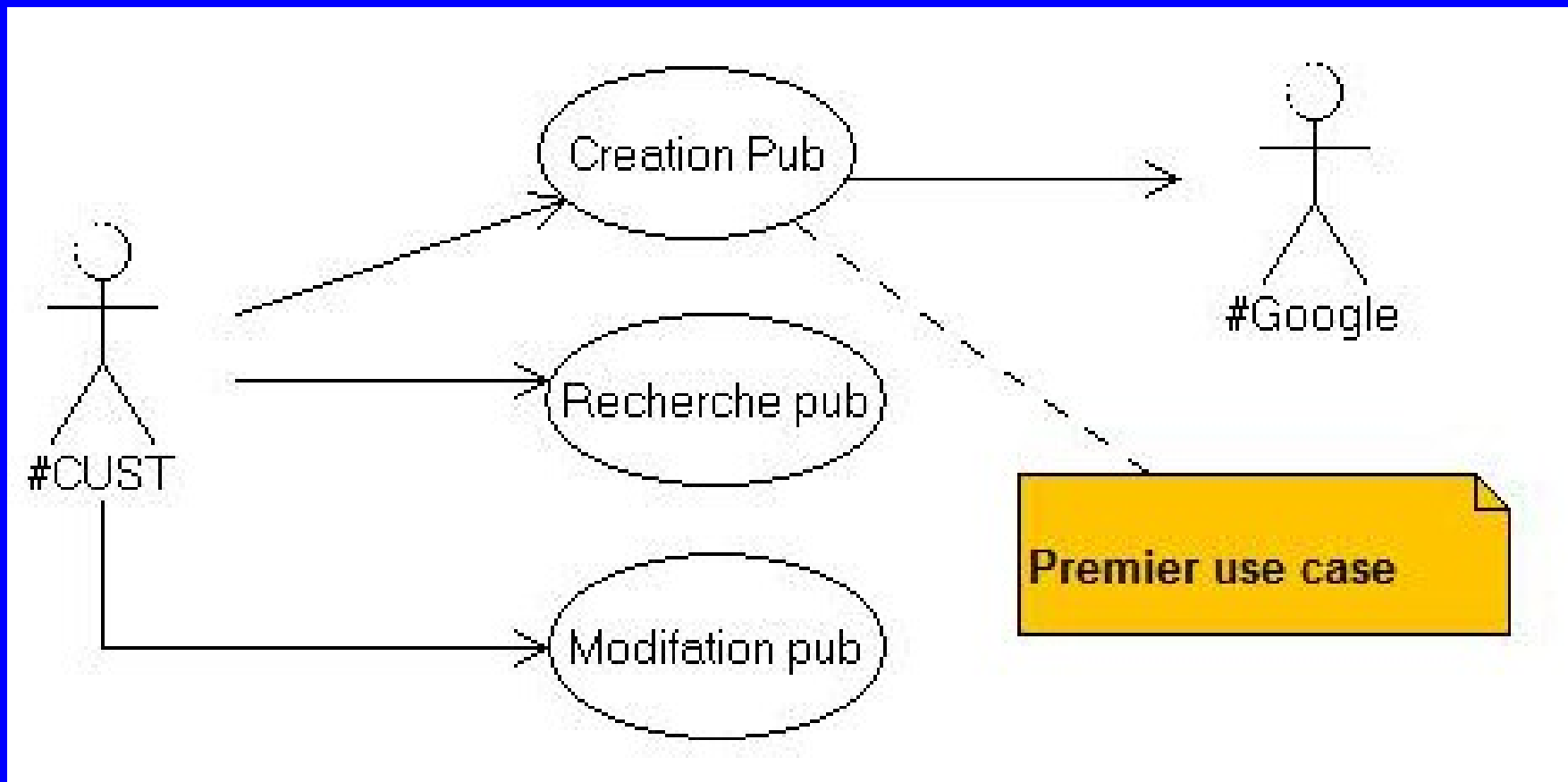
Les clients non seulement on ne comprend pas ce qu'ils veulent, mais en plus il faut imaginer ce qu'ils ne disent pas ?

Si l'on veut que le soft puisse évoluer dans le futur, il faut l'avoir anticiper ...



# Cas d'utilisation

- Description des cas d'utilisation 1/4



# Cas d'utilisation

## - Description des cas d'utilisation 2/4

USE-CASE DÉTAILLÉ «CRÉATION PUB»	
Référence	5.2.1
Justification	Un client se connecte au site pour créer sa publicité
Description	1. → #CUST visualise la page d'accueil du site 2. → #CUST s'identifie 3. → #CUST <u>click</u> sur le lien «créer une publicité» 4. → #CUST saisie le texte de sa publicité 5. → #CUST <u>upload</u> une bannière 6. → #CUST saisie la localisation du lieu d'affichage 1. → #CUST se déconnecte
Acteurs	#CUST
Use-cases inclus	N/A
Déclenchement	Après avoir payé son adhésion #CUST saisie une publicité
Pré-conditions	Le site est opérationnel et #CUST est correctement enregistré
<u>Post-conditions</u>	La publicité est stockée
Scénario(s)	Erreur de saisie, taille de la bannière incorrecte, pertes de connexion
Note	N/A

# Cas d'utilisation

- **Description des cas d'utilisation 3/4**
  - **Un cas d'utilisation contient plusieurs scénarios**
    - **Le scénario nominal et tous les scénarios d'exception ou conditionnels**
  - **Description textuelle « non normalisée »**
  - **Un des objectifs est de déduire des exigences**

N/A	E_098	Besoins sources : UC 4.1.2.1
L'API doit fournir une méthode de destruction des objets métiers		
Description des tests : Réalisation d'un programme qui se connecte à une base contenant des objets métiers et qui détruit un des objets métiers.		
Référence des tests : TST_15		

# Cas d'utilisation

- Description des cas d'utilisation 4/4



Les exigences  
viennent des cas  
d'utilisation, mais  
s'ils sont mal  
rédigés ?



Les exigences  
risquent d'être  
fausses ...

Il faut bien les  
revoir entre  
pairs alors ?



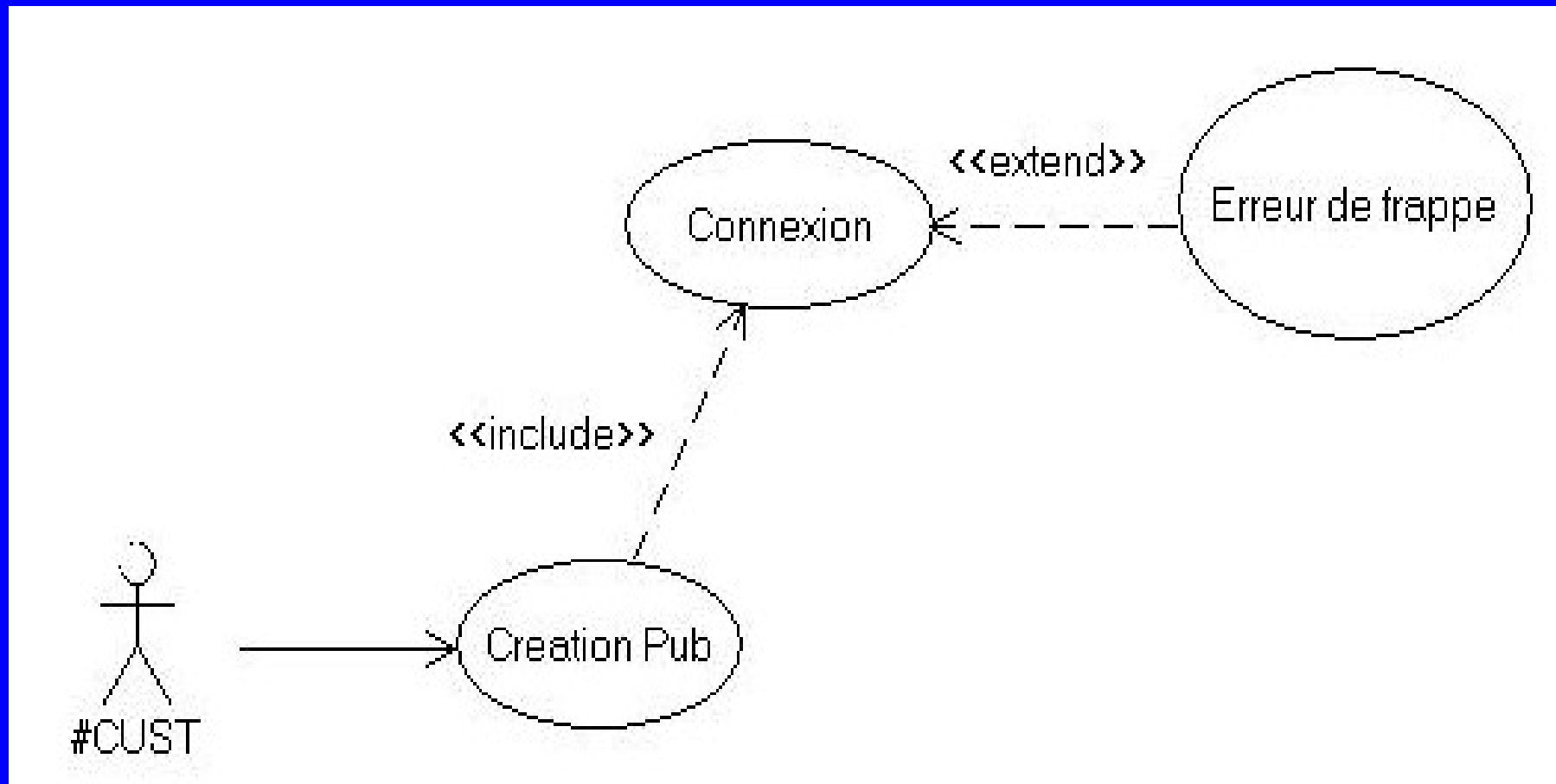


# Cas d'utilisation

- **Relations entre cas d'utilisation 1/3**
  - **Un cas d'utilisation peut :**
    - Inclure d'autres cas d'utilisation
    - Étendre un autre cas d'utilisation
      - Présenter les alternatives
    - Hériter d'autres cas d'utilisation
    - Appartenir à un package

# Cas d'utilisation

- Relations entre cas d'utilisation 2/3



# Cas d'utilisation

- Relations entre cas d'utilisation 3/3

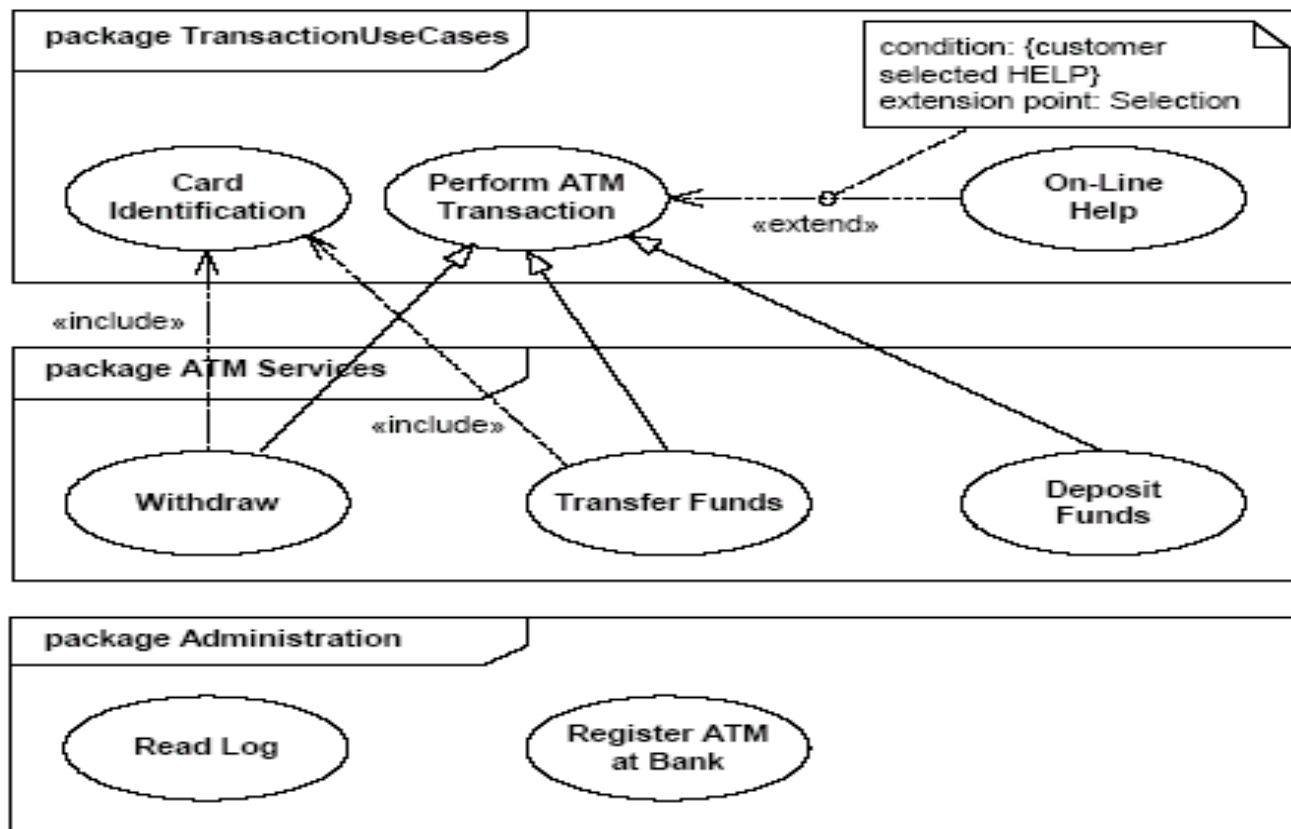


Figure 16.7 - Example of use cases owned by various packages

# Cas d'utilisation

- **Difficultés des cas d'utilisation 1/1**
  - **Les cas d'utilisation sont verbeux**
    - Il y en a beaucoup
  - **Les alternatives ne sont pas simples à décrire**
  - **Les erreurs et exceptions sont difficiles à prendre en compte**
  - **Comprendre le métier des utilisateurs n'est pas aisé**
  - **Il y a le risque de coder en procédural**

# Cas d'utilisation

- **Critères d'évaluation 1/1**
  - **La liste est-elle exhaustive ?**
    - Comprend-t-elle les évolutions, l'avenir ?
  - **Les cas principaux ont-ils été décrits ?**
  - **Les utilisateurs ont-ils été impliqués ?**
    - Ont-ils compris les descriptions ?
  - **Toutes les parties prenantes ont-elles été consultées ?**
  - **Le vocabulaire utilisé est-il précis, clairement défini ?**

# Cas d'utilisation

- **QCM 1/1**

- Les cas d'utilisation ne servent-ils qu'aux spécifications YIN ?, à quoi d'autres ?
- Les cas d'utilisation concernent-ils d'autres fonctionnalités que celles attendues ?
- Quels sont les éléments des cas d'utilisation le plus importants ?
- Citer des éléments difficilement pris en compte par dans les cas d'utilisation ?
- Faut-il décrire tous les cas d'utilisation ?

# Diagrammes d'activité

# Diagrammes comportementaux

- **Plan**
  - **Utilisation**
  - **Diagramme d'activités**
  - **Critères d'évaluation**
  - **QCM**

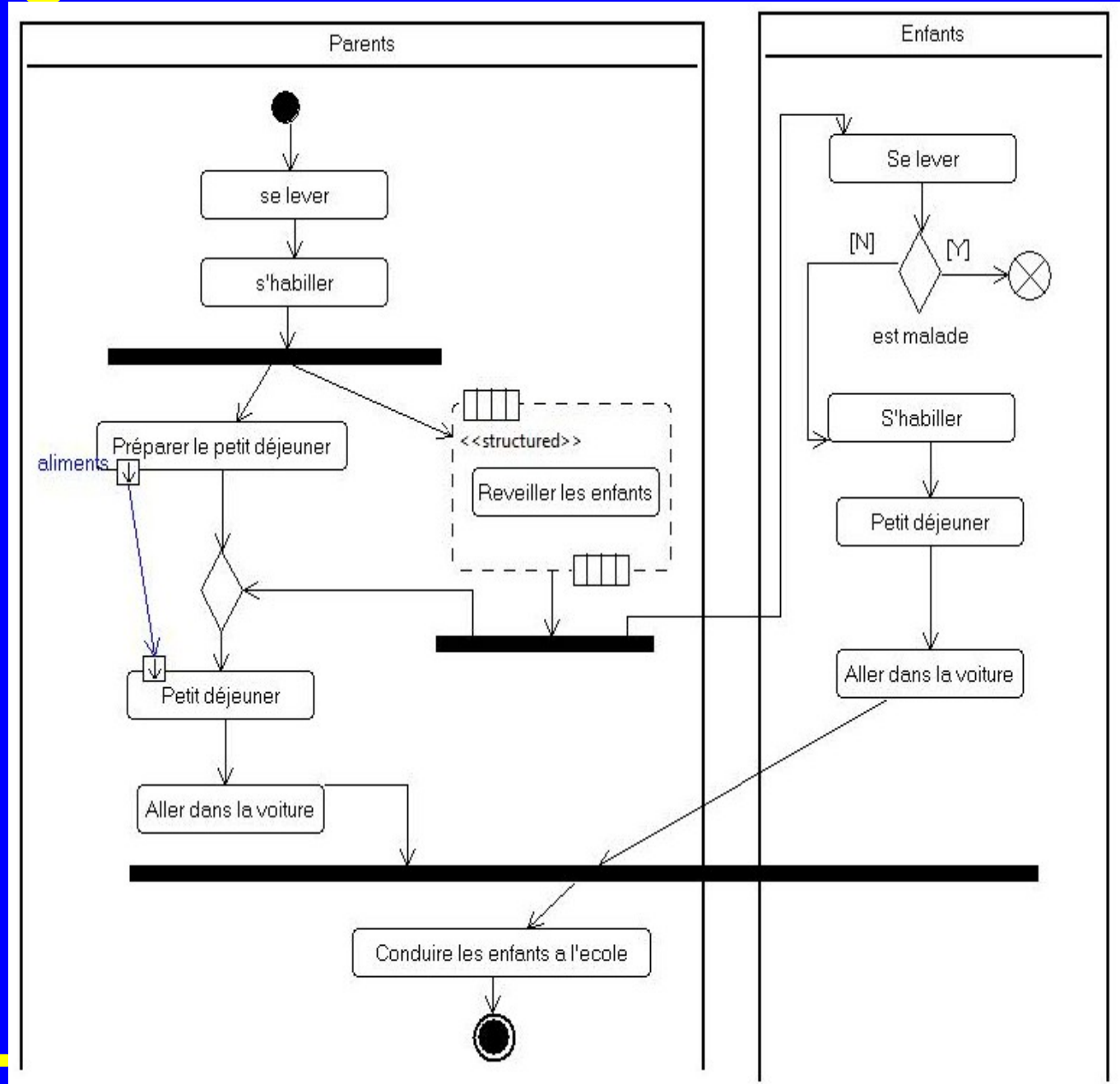


# Diagrammes d'activité

- **Utilisation 1/1**
  - **Modélise:**
    - Les cas d'utilisation
    - Action entre objets
    - Workflow
    - Processus d'entreprise ...
  - **Ressemble le plus aux organigrammes**
  - **Ils sont simples à comprendre**
  - **En général, ils sont appréciés par tous les niveaux hiérarchiques**

# Diagrammes d'activité

- Diagramme D'activités 1/5

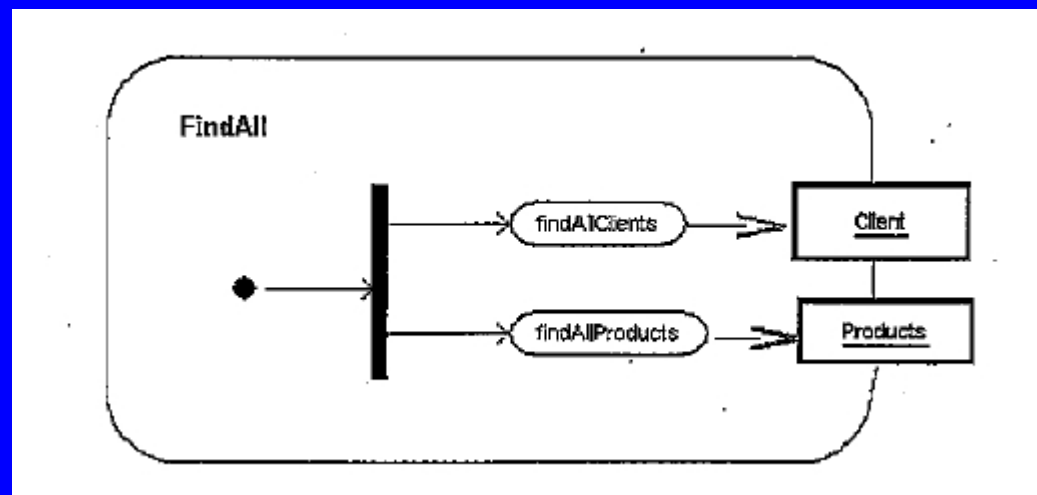
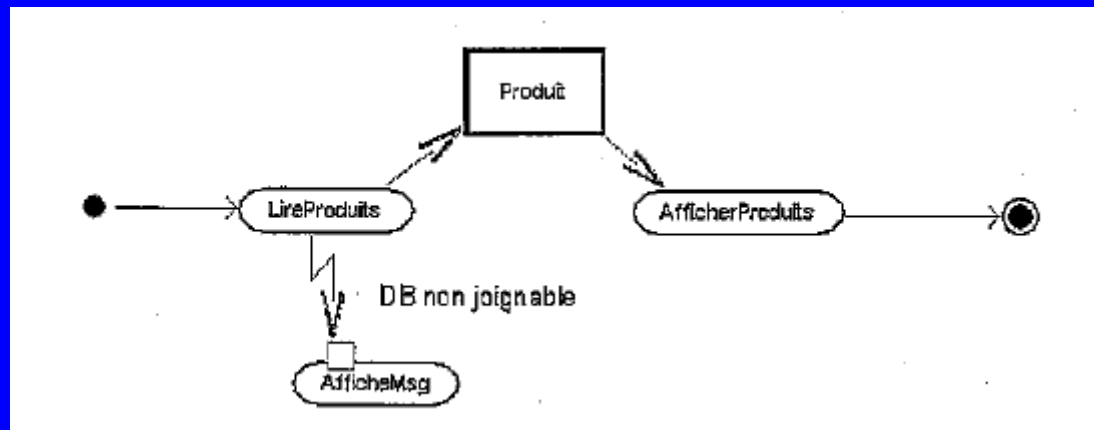


# Diagrammes d'activité

- **Diagramme d'activités 2/5**
  - **Qu'est-il possible de spécifier de plus ?**
    - Des flux d'objets
    - Des paramètres
    - Des actions subsidiaire
    - Des gestions d'exceptions
    - Des événements

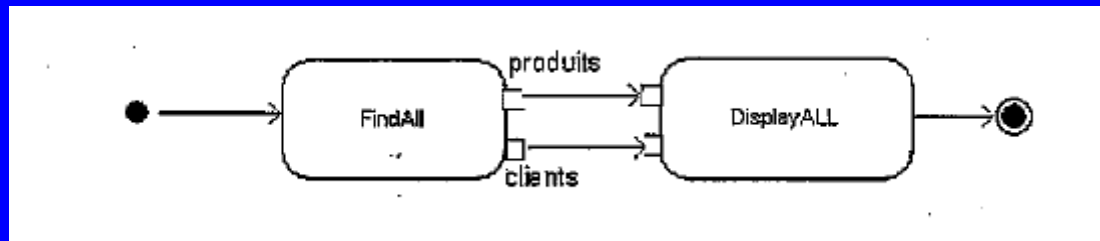
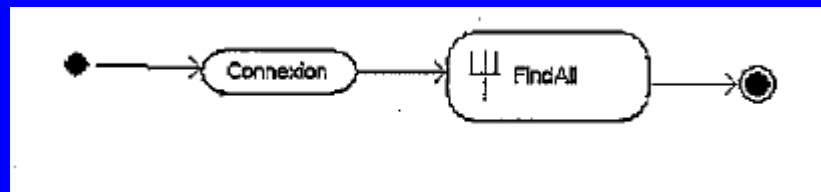
# Diagrammes d'activité

- Diagramme d'activités 3/5



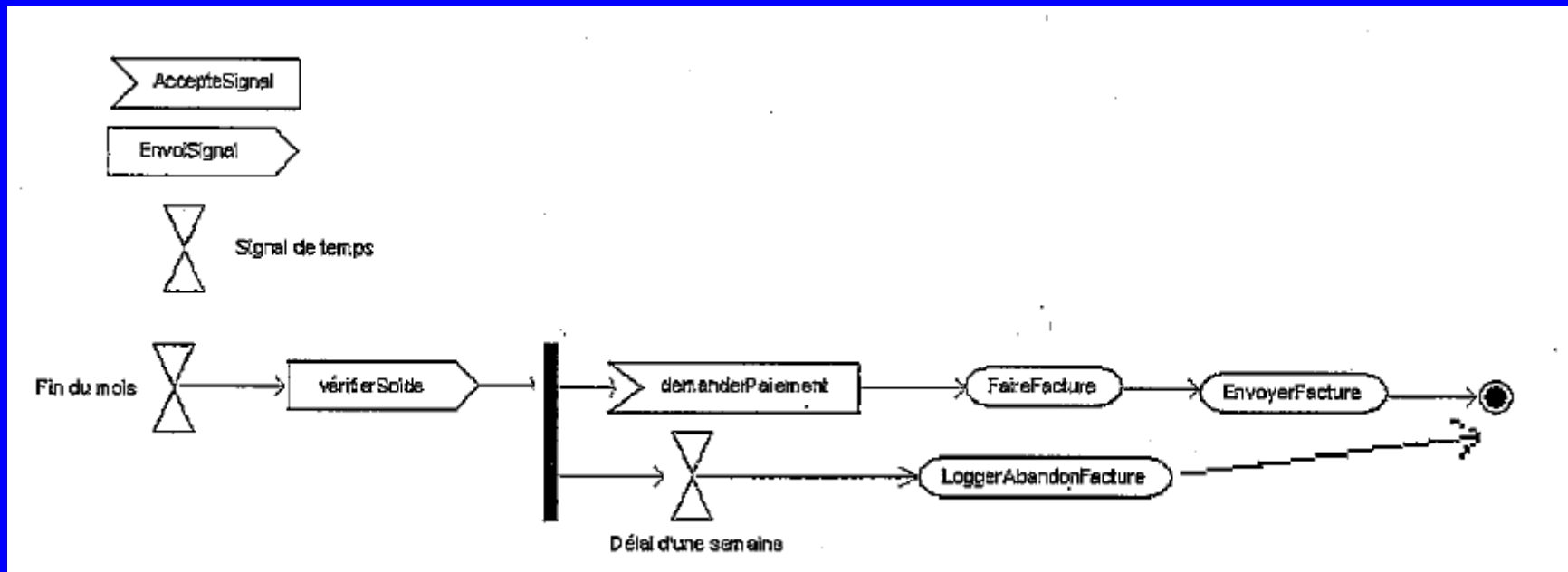
# Diagrammes d'activité

- Diagramme d'activités 4/5



# Diagrammes d'activité

- Diagramme d'activités 5/5



# Diagrammes d'activité

- **Critères d'évaluation 1/1**
  - Les participants ont-ils été correctement identifiés ?
  - Les principaux uses cases ont-ils été modélisés ?
  - Les données et actions principales sont-elles identifiées ?
  - Sont-ils suffisamment complets ?
  - Sont-ils logiquement cohérents ?

# Diagrammes d'activité

- **QCM 1/1**

- **Que peut représenter un diagramme d'activité ?**
- **Qu'est-il intéressant de visualiser sur un diagramme d'activité ?**
- **Comment se concrétise dans le code les actions et les données des actions ?**
- **Tous les uses cases doivent-ils donner lieu à un diagramme d'activité ?**