

# Phases d'un Projet Logiciel et Cas d'utilisation

# PHASES D'UN PROJET LOGICIEL

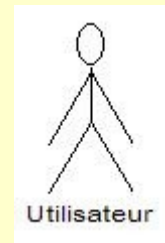
- Expression de besoins
- Cahier des charges
- Analyse : Décrire un problème
- Conception : Décrire la solution du problème
- Implémentation et tests unitaires
- Tests d'intégration
- Recette

# PHASE D'ANALYSE

- On commence par identifier :
  - Le système
  - Les acteurs qui interagissent avec le système
  - Les actions des acteurs sur le système
- On continue en identifiant les objets du système et leurs interactions

# Les acteurs

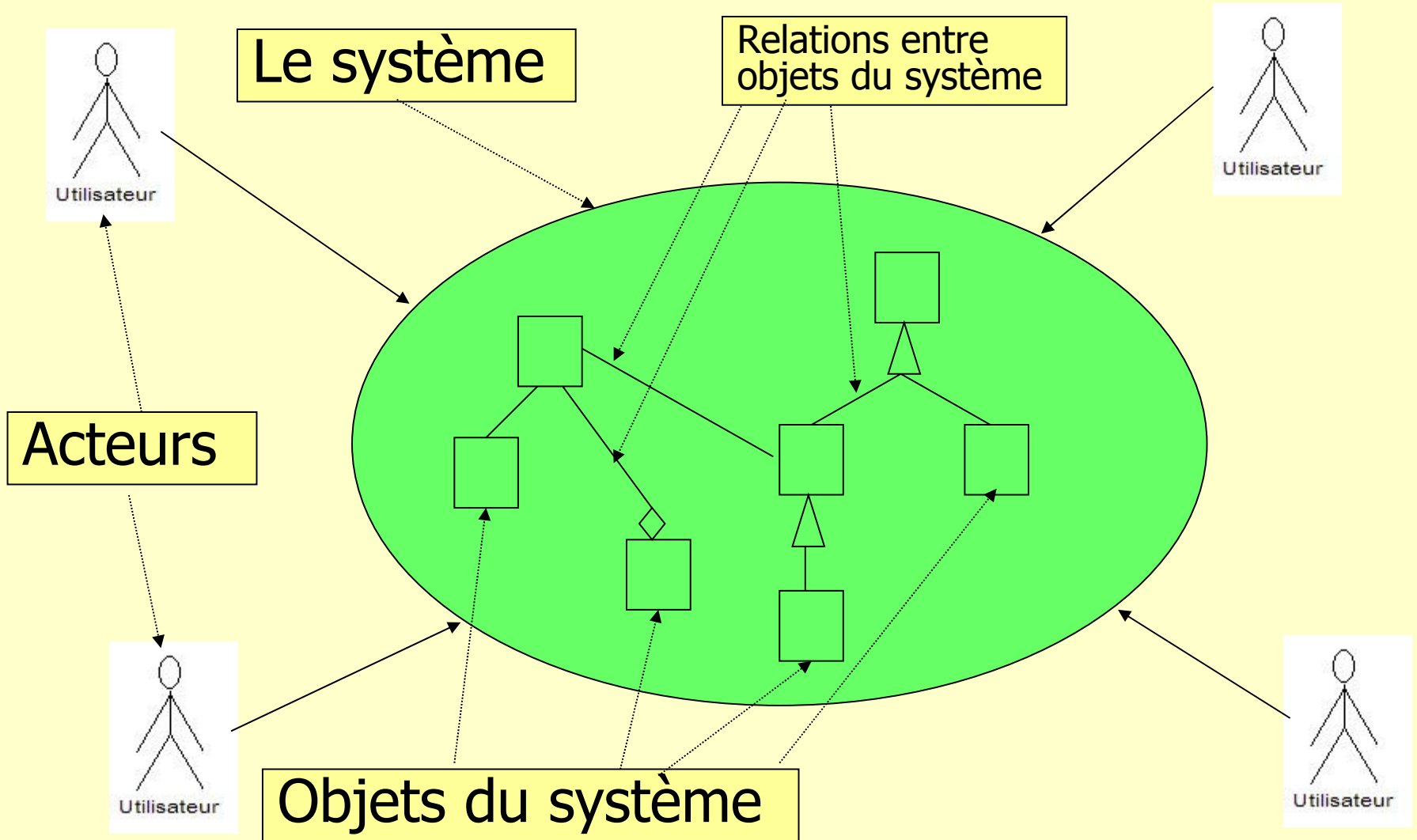
- Un acteur est une entité externe (être humain ou robot) qui interagit avec le système.
- Un même humanoïde peut être plusieurs acteurs.
- On définit donc un acteur par un ensemble de rôles qu'il a sur le système.
- Il agit sur le système en accès lecture et/ou écriture.
- En UML, on utilise le pictogramme



# type d'acteurs

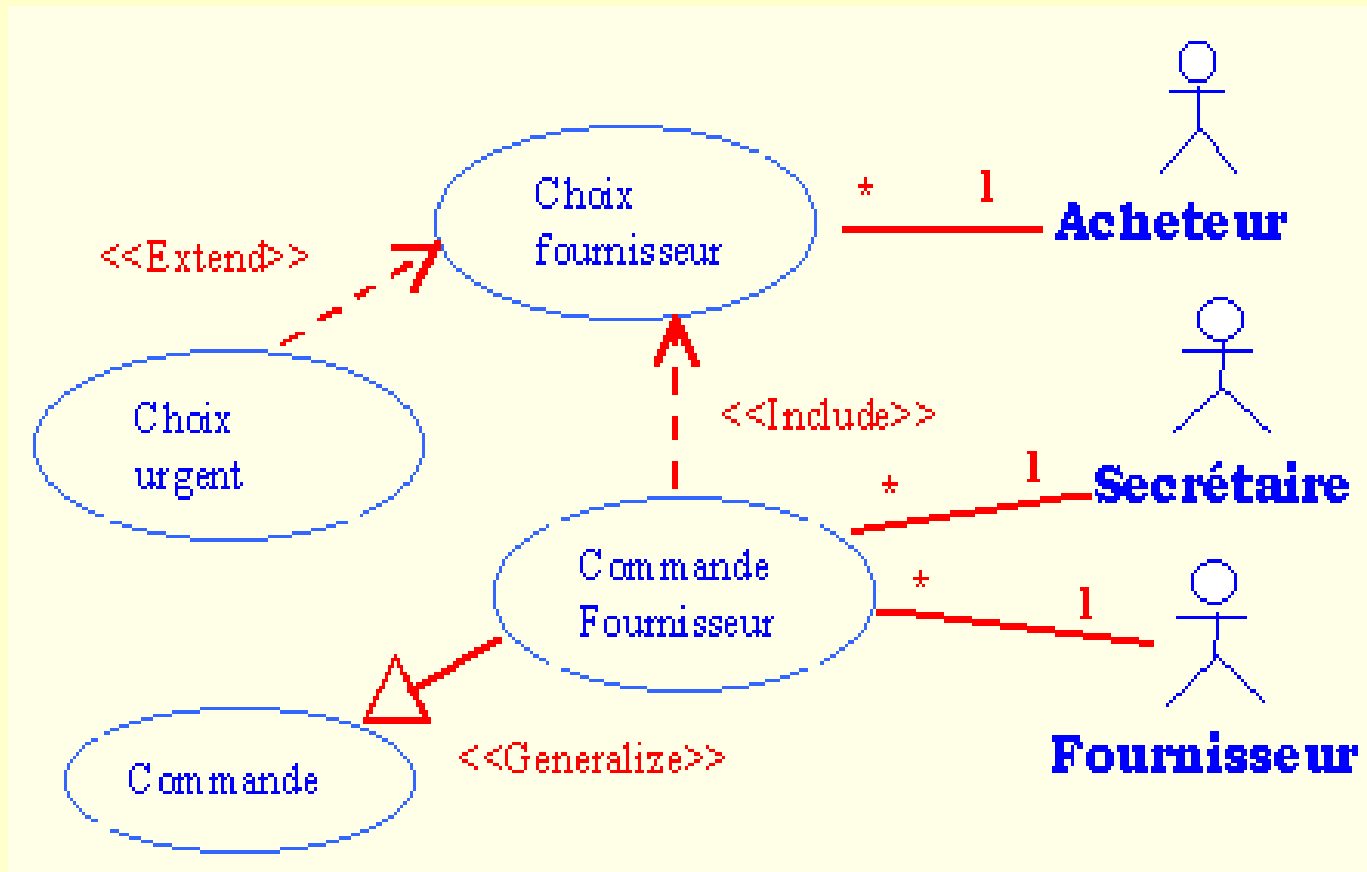
- Les **acteurs principaux** : usager, client, ....
- Les **acteurs secondaires** : administrateur, opérateur de maintenance, ....
- Le **matériel externe** : imprimante, capteur, ....
- Les **autres systèmes** : Serveurs, SGBD

# Acteurs et système



# UML : USE CASE

- Acteur et action  $\Leftrightarrow$  Actor et Use Case en UML



# La relation <<extend>>

- C'est une relation entre 2 instances de Use Case Ajout et Gen notée :

Ajout extend Gen

- Elle signifie que le comportement d'un cas d'utilisation Gen peut être complété par le comportement du use case Ajout sous réserve qu'une certaine condition soit satisfaite.

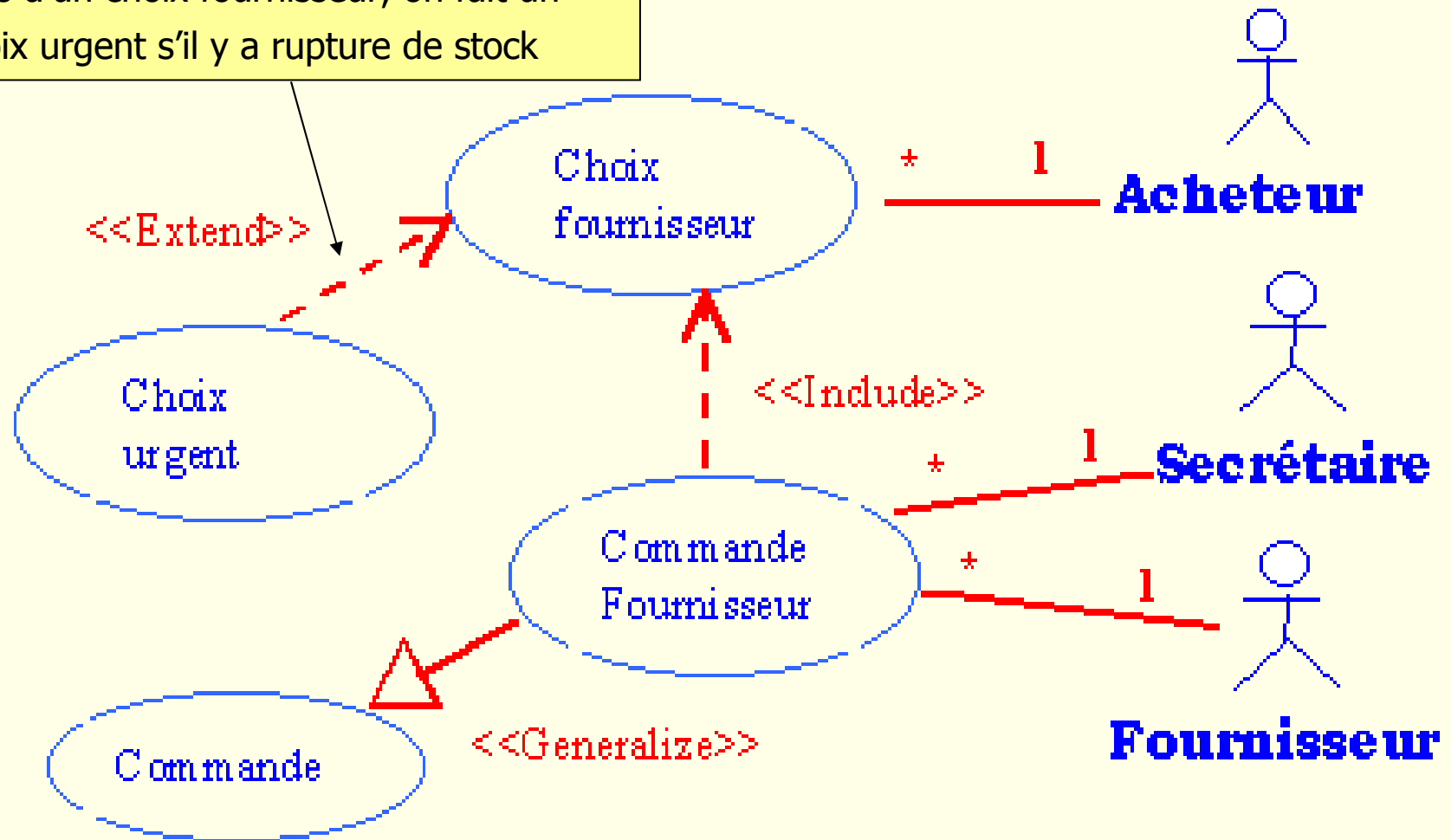


# La relation <<extend>>

- **Dans ce type de relation, on doit spécifier :**
  - **la condition de l'extension : Si cette condition est satisfaite, le use case Ajout sera exécuté lors de l'exécution du use case Gen.**
  - **le point d'extension : le moment où l'extension Ajout doit être faite dans le Use Case Gen.**

# La relation <<extend>>

Lors d'un choix fournisseur, on fait un choix urgent s'il y a rupture de stock



# La relation <<include>>

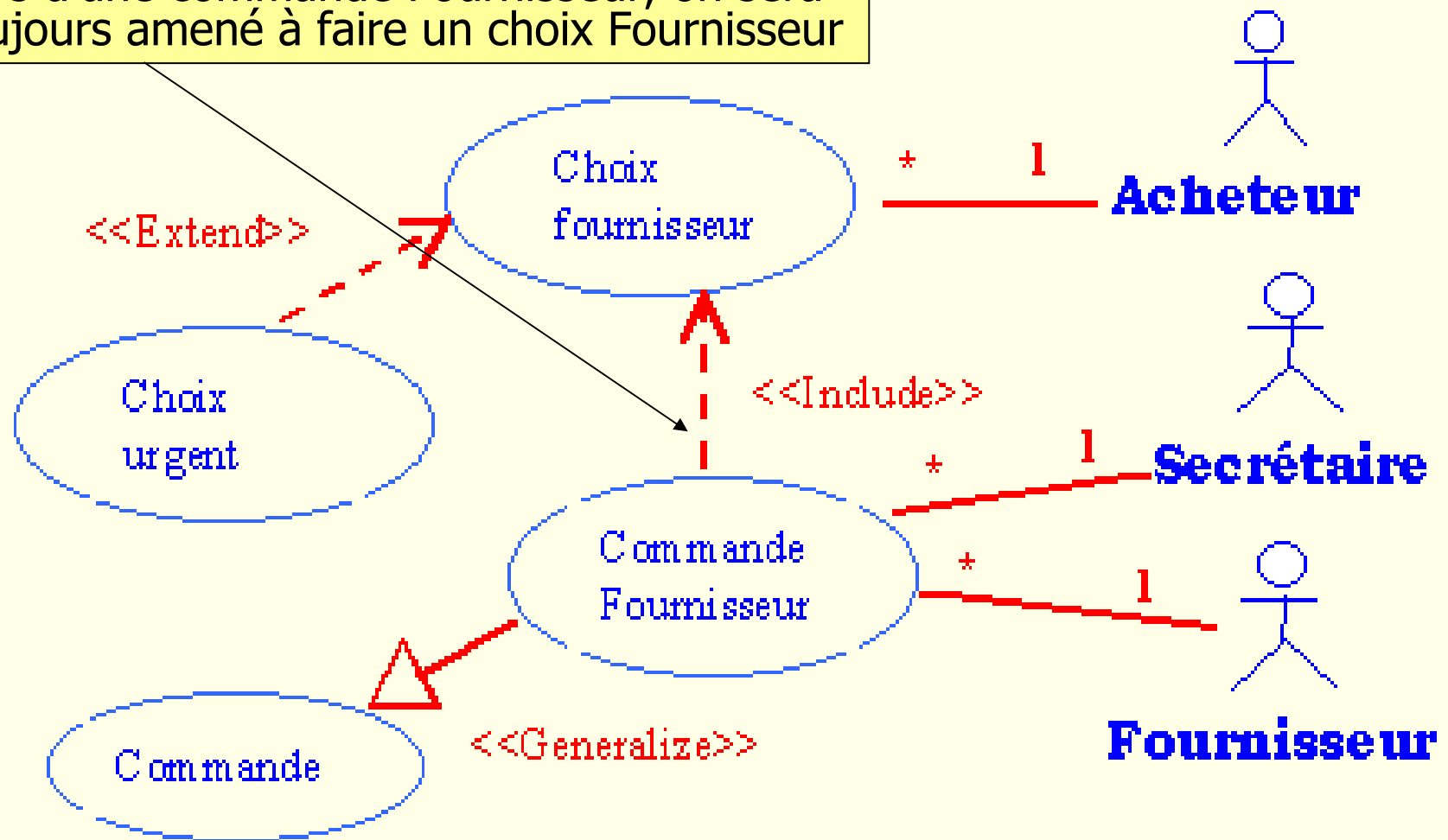
- C'est une relation entre 2 instances de Use Case A et B notée :

B include A

- Elle signifie que le comportement d'un cas d'utilisation B inclura systématiquement le comportement du use case A en un certain point d'extension.

# La relation <<extend>>

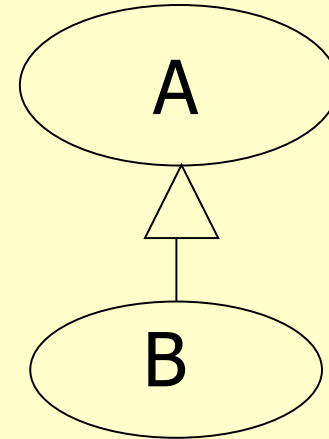
Lors d'une commande Fournisseur, on sera toujours amené à faire un choix Fournisseur



# La relation <<generalize>>

- C'est une relation entre 2 cas d'utilisation A et B notée :

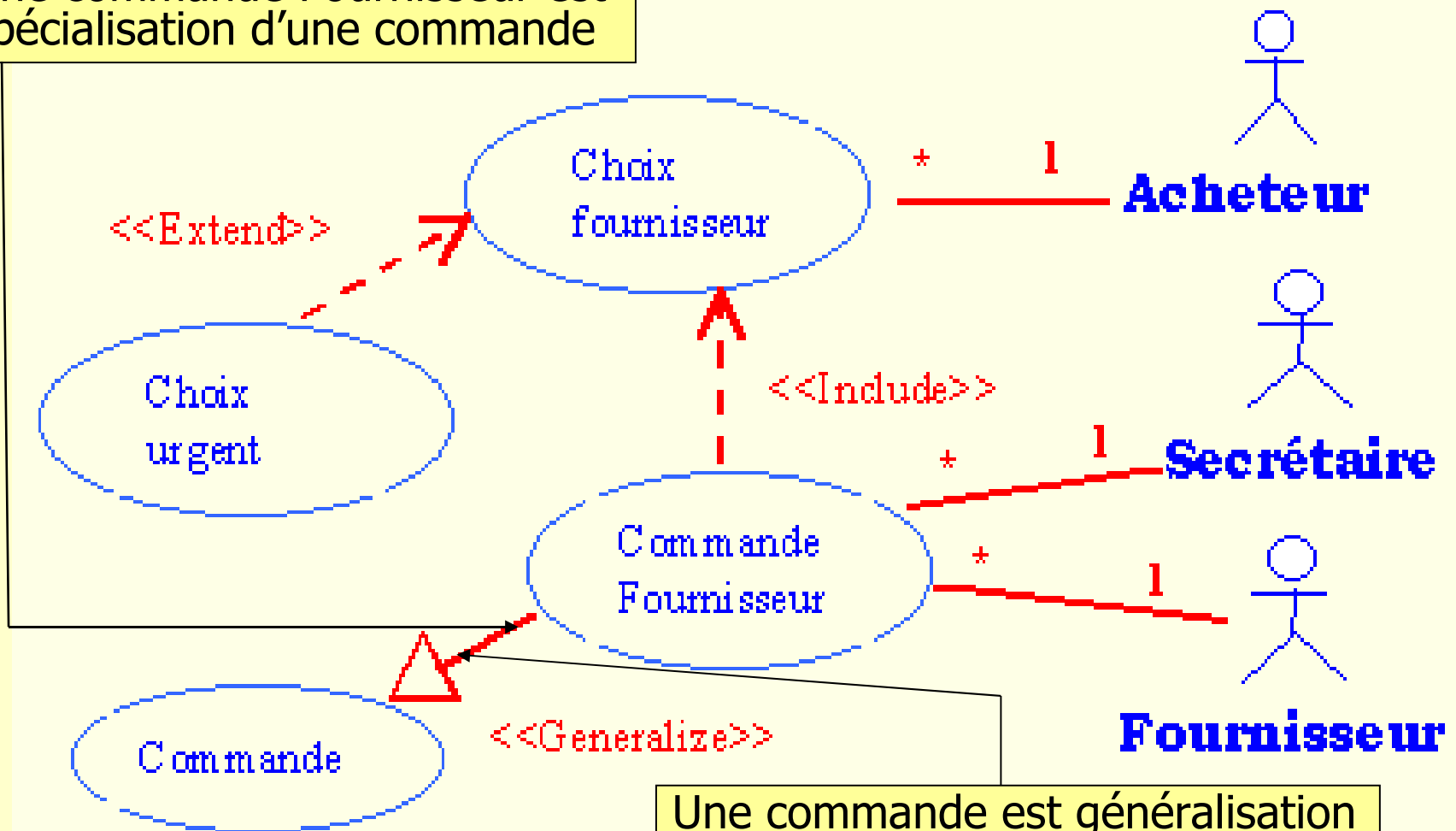
A generalize B ou



- Contrairement aux précédentes, il s'agit d'une relation entre 2 cas et non entre des instances de ceux-ci

# La relation <<extend>>

Une commande Fournisseur est spécialisation d'une commande



Une commande est généralisation d'une commande Fournisseur

# Fiche descriptive d'un cas d'usage

- A chaque cas d'usage, on associe une fiche descriptive.
  - 1) une description du cas d'usage
  - 2) règle d'initiation : qu'est ce qui déclenche la transaction ?
  - 3) règle de terminaison : qu'est ce qui termine la transaction ?
  - 4) règle d'exception qu'est ce qui déclenche le cas d'usage quand la règle d'initiation n'est pas vérifiée ?
  - 5) les relations avec d'autres cas d'usage : extension , inclusion et généralisation

# Fiche descriptive d'un cas d'usage

<b>USE CASE # 1</b>	<i>Forme verbale (infinitif)</i> <span style="float: right;"><i>(ex : Passer une commande)</i></span>	
<b>Résumé</b>		
<b>Acteur principal</b>	X	
<b>Intervenants &amp; Intérêts</b>	<i>Intervenants</i>	<i>Intérêt</i>
	X Y Z	
<b>Préconditions</b>		
<b>Garanties de succès</b>	<i>(Conditions d'arrivée)</i>	
<b>Déclencheur</b>	<i>(Evènement)</i>	
<b>Scénario nominal</b>	<i>Etape</i>	<i>Action</i> <span style="float: right;"><i>(Le cas où tout se passe bien)</i></span>
	1	L'utilisateur
	2	X lance ....
	3	Le système
	4	X lance le processus <i>(Cas d'utilisation 1.1)</i>
	5	Le système.
<b>Extensions</b>	<i>Etape</i>	<i>Action</i> <span style="float: right;"><i>(Variations au scénario et cas d'erreurs)</i></span>
	2a	X reçoit ...
		2a.1 X fait ceci
		2a.2 Y fait cela



The image shows a screenshot of the Rational Rose software interface. The window title is "Rational Rose - (untitled) - [Class Diagram: Logical View / Main]". The menu bar includes File, Edit, View, Format, Browse, Report, Query, Tools, Add-Ins, Window, and Help. The toolbar contains various icons for file operations and navigation. The left sidebar shows a project tree with folders for Use Case View, Logical View, Component View, and Deployment View, each containing a Main sub-folder. A Log window at the bottom left shows a series of "[Customizable Menus]" entries. Three yellow callout boxes with black text are overlaid on the right side of the interface, with arrows pointing to the Use Case View, Logical View, and Component View folders respectively.

**Analyse**  
Description des acteurs et cas d'utilisation

**Analyse : quoi**  
Description du système

**Conception : comment**  
Description du système et de son architecture hardware et software