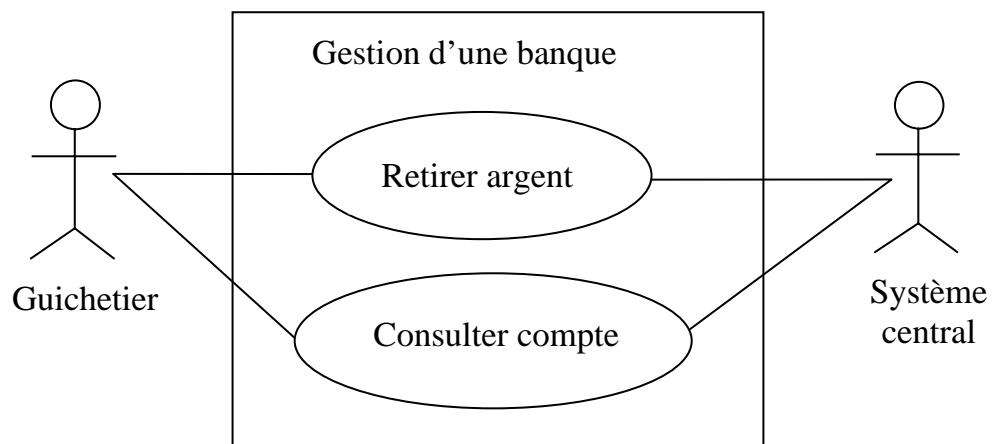


## Chapitre 3: UML (Unified Modeling Language)

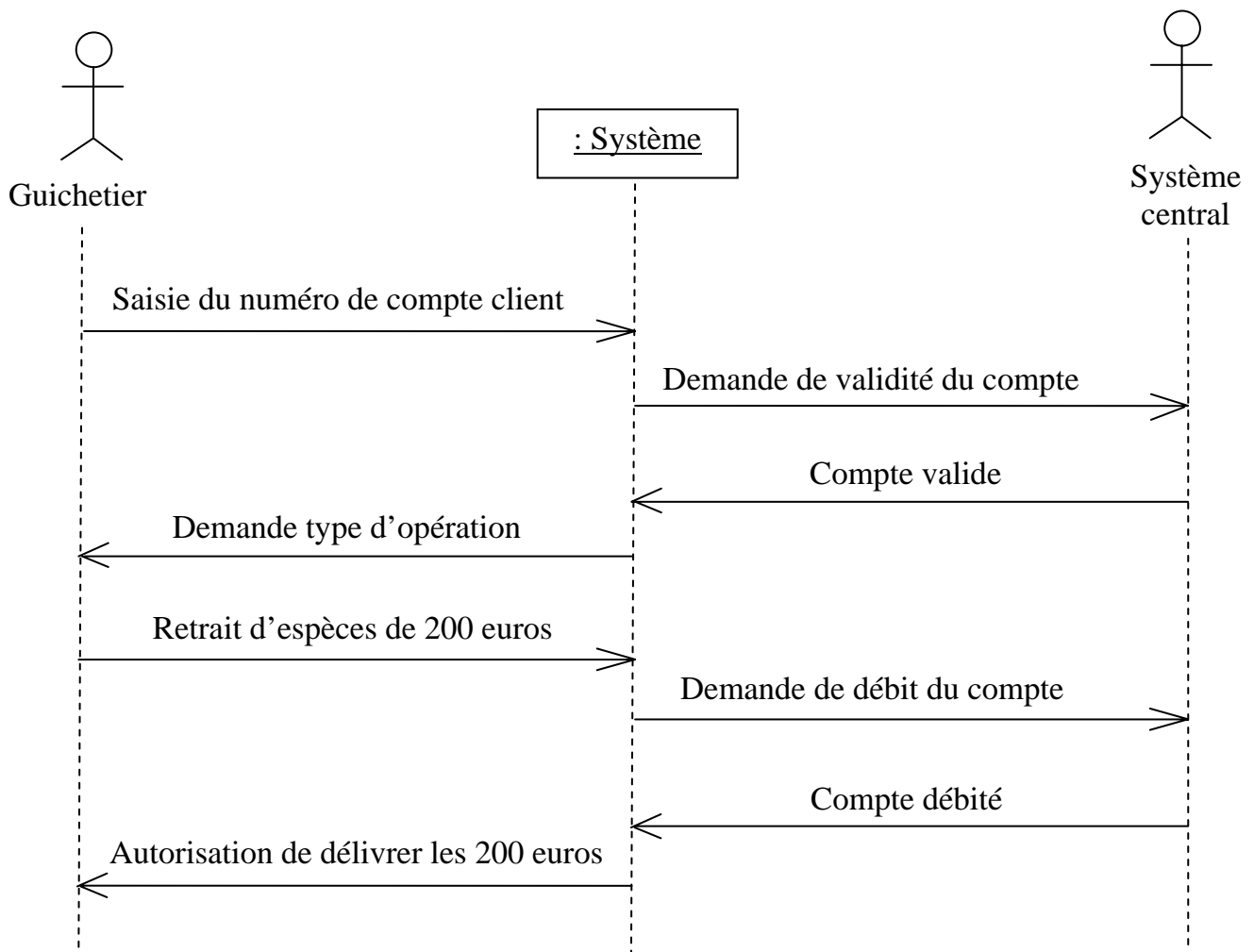
### Partie 7 : Description des cas d'utilisation

#### 1. Description des cas d'utilisation

À partir du diagramme de cas d'utilisation ci-dessous :



La figure suivante montre une façon de décrire les interactions pour le retrait d'argent (retrait d'espèces en euros). On y voit clairement apparaître une séquence de messages.



**Remarque :**

En modélisation objet, les diagrammes de séquence s'utilisent de deux manières différentes, selon la phase du cycle de vie et le niveau de détail désiré.

La première utilisation correspond à la documentation des cas d'utilisation ; elle se concentre sur la description de l'interaction, souvent dans des termes proches de l'utilisation et sans entrer dans les détails de la synchronisation. L'indication portée sur les flèches correspond alors à des événements qui surviennent dans le domaine de l'application. A ce stade de la modélisation, les flèches ne traduisent pas encore des envois de message au sens des langages de programmation.

La deuxième utilisation correspond à un usage plus informatique. Le concept de message unifie toutes les formes de communication entre objets (appel de procédure par exemple).

### ➤ Description textuelle des cas d'utilisation

La description d'un scénario par un diagramme de séquence n'empêche pas de décrire le cas sous forme textuelle, ce qui permet de faire figurer des renseignements supplémentaires.

Une description textuelle couramment utilisée se compose de trois parties. La première partie permet d'identifier le cas. Elle doit contenir :

- le nom du cas ;
- un résumé de son objectif ;
- les acteurs impliqués (principaux et secondaires) ;
- les dates de création et de mise à jour de la description courante ;
- le nom des responsables ;
- un numéro de version.

La deuxième partie contient la description du fonctionnement du cas sous la forme d'une séquence de messages échangés entre les acteurs et le système. Elle contient **toujours une séquence nominale** qui correspond au fonctionnement nominal du cas. Cette séquence nominale commence par préciser l'événement qui déclenche le cas et se développe en trois points :

- Les pré-conditions : Elles indiquent dans quel état est le système avant que se déroule la séquence.
- L'enchaînement des messages.
- Les post-conditions : Elles indiquent dans quel état se trouve le système après le déroulement de la séquence nominale.

Parfois la séquence correspondant à un cas a besoin d'être appelée dans une autre séquence. Signifier l'appel d'une autre séquence se fait de la façon suivante :

« appel du cas X », où X est le nom du cas.

Les acteurs n'étant pas sous le contrôle du système, ils **peuvent** avoir des comportements imprévisibles. La séquence nominale ne suffit donc pas pour décrire tous les comportements possibles. À la séquence nominale s'ajoutent fréquemment des séquences alternatives et des séquences d'exceptions. Ces deux types de séquences se décrivent de la même façon que la séquence nominale mais il ne faut pas les confondre. Une séquence alternative diverge de la séquence nominale (c'est un embranchement dans une séquence nominale) mais y revient toujours, alors qu'une séquence d'exception intervient quand une erreur se produit (le séquençage nominal s'interrompt, sans retour à la séquence nominale).

Dans le cas d'un retrait d'argent, des séquences alternatives se produisent par exemple dans les situations suivantes :

- Le client choisit d'effectuer un retrait en euros ou en dollars.
- Le client a la possibilité d'obtenir un reçu.

Une exception se produit si la connexion avec le système central de la banque qui doit vérifier la validité du compte est interrompue.

La survenue des erreurs dans les séquences doit être signalée de la façon suivante :

« appel de l'exception Y » où Y est le nom de l'exception.

La séquence nominale, les séquences alternatives, les exceptions, etc., font qu'il existe une multitude de chemins depuis le début du cas jusqu'à la fin. Chaque chemin est appelé scénario. Un système donné génère peu de cas d'utilisation, mais, en général, beaucoup de scénarios.

La dernière partie de la description d'un cas d'utilisation est une rubrique optionnelle. Elle contient généralement des spécifications<sup>1</sup> non fonctionnelles (ce sont le plus souvent des spécifications techniques, par exemple pour préciser que l'accès aux informations bancaires doit être sécurisé).

## **Description d'un retrait d'argent**

### **Identification**

Nom du cas : Retirer argent

But : détaille les étapes permettant à un guichetier d'effectuer l'opération de retrait d'euros demandé par un client.

Acteur principal : Guichetier.

Acteur secondaire : Système central.

Date : le 18/02/2015.

Responsable : M. Dupont.

Version : 1.0.

### **Séquencement**

Le cas d'utilisation commence lorsqu'un client demande le retrait d'espèces en euros.

Pré-conditions

Le client possède un compte (donne son numéro de compte).

Enchaînement nominal

1. Le guichetier saisit le numéro de compte client.
2. L'application valide le compte auprès du système central.
3. L'application demande le type d'opération au guichetier.
4. Le guichetier sélectionne un retrait d'espèces de 200 euros.
5. L'application demande au système central de débiter le compte.
6. Le système (application) notifie au guichetier qu'il peut délivrer le montant demandé.

Post-conditions

Le guichetier ferme le compte.

Le client récupère l'argent.

### **Rubriques optionnelles**

Contraintes non fonctionnelles

Fiabilité : les accès doivent être extrêmement sûrs et sécurisés.

Confidentialité : les informations concernant le client ne doivent pas être divulguées.

Contraintes liées à l'interface homme-machine.

Donner la possibilité d'accéder aux autres comptes du client.

Toujours demander la validation des opérations de retrait.

---

<sup>1</sup> Les spécifications peuvent être divisées en deux catégories selon qu'elles sont fonctionnelles ou techniques. Les spécifications fonctionnelles concernent les fonctions du système, la fonction de retrait d'argent par exemple, tandis que les spécifications techniques permettent de préciser le contexte d'exécution du système.