

INSIA – SIGL 2

UML

2 : ANALYSE FONCTIONNELLE

Diagrammes de cas d'utilisation, de séquence, d'activités

Bertrand LIAUDET

SOMMAIRE

<u>LES UC</u>	<u>2</u>
1. Présentation générale de la notion de cas d'utilisation : UC	2
2. Approche par l'exemple	4
3. Les acteurs	10
4. Les UC du logiciel – Formalisme	13
5. Méthode de construction d'un diagramme des UC	18
6. Architecture et diagramme des cas d'utilisation	20
7. Compléments, limites et alternatives du diagramme des cas d'utilisation.	21
<u>LE DIAGRAMME DE SEQUENCE « SYSTEME »</u>	<u>22</u>
1. Le diagramme de séquence système	22
2. Architecture et diagramme de séquence système	27
3. Modularité, test et boucle	28
<u>LE DIAGRAMME D'ACTIVITE DES SCENARIOS</u>	<u>30</u>
1. Les diagrammes d'activité dans l'analyse fonctionnelle	30
2. Diagramme d'activité d'un UC : tous ses scénarios	33
3. Diagramme d'activité d'un traitement MERISE	33

Première édition : janvier 2009

Deuxième édition : décembre 2009

LES UC

Il est facile de décrire la méthode encore que son application exige à coup sûr savoir et pratique.

1. Présentation générale de la notion de cas d'utilisation : UC

ANALYSE FONCTIONNELLE	Diagramme UML
	Cas d'utilisation
	Séquence
	Activités

Cas d'utilisation : UC

Cas d'utilisation = Use Case = UC

Un UC est :

un usage du système (du programme),
une fonctionnalité du système.

Plus précisément :

une fonctionnalité complète du système.

Exemple :

Dans le système « guichet automatique d'une banque », « retirer de l'argent » est un UC. C'est une fonctionnalité complète du système qui va de l'insertion de la carte de retrait par le client jusqu'à la récupération de la carte de retrait par le client.

Du point de vue de l'utilisateur, un UC est :

un ensemble d'activités du système qui produit un résultat intéressant pour un utilisateur

Du point de vue du système lui-même, un UC est :

un ensemble d'activités qui part d'un système au repos pour arriver de nouveau à un système au repos.

Acteur

Les UC sont initiés par des acteurs.

Un acteur est à l'extérieur du système. Il interagit avec le système.

Exemple :

Dans le système « guichet automatique d'une banque », le client qui vient retirer de l'argent est un acteur du système.

Le scénario : une instance d'un UC

Définition

Un scénario est un déroulement concret d'un UC parmi tous les déroulement concrets possibles.

Un UC est donc un ensemble de scénarios.

Le scénario est à l'UC ce que l'objet est à la classe : une instance concrète.

Un scénario est une instance concrète d'un UC (un élément de l'ensemble de tous les scénarios pour le UC considéré).

Scénario nominal

Le scénario nominal est le scénario qui correspond au fonctionnement « normal » du UC, c'est-à-dire à sa finalité première.

Les autres scénarios, alternatifs, sont des cas particuliers.

Exemple :

Il y aura plusieurs scénarios pour retirer de l'argent : si le code de la carte de retrait est faux ; si le client n'est pas autorisé à retirer de l'argent ; si le guichet n'a plus de billets ; etc.

Le scénario nominal décrit le retrait d'argent « normal ».

Acteur et instance d'acteur

L'acteur est une abstraction. L'instance d'acteur est une personne concrète.

L'instance d'acteur est à l'acteur ce que l'objet est à la classe.

Exemple :

M. Dupond qui vient retirer de l'argent est un acteur concret.

2. Approche par l'exemple

Généralisation et spécialisation des UC : UC concrets et abstraits

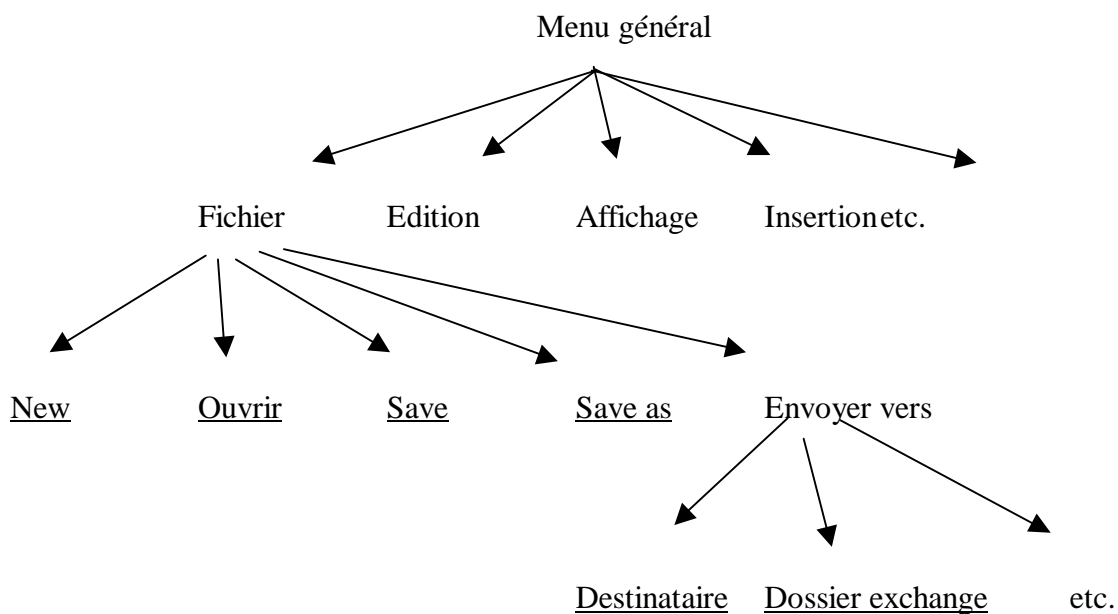
Partons du logiciel Word et décrivons les UC.

Les UC sont donnés par les menus.

Les UC abstraits sont les menus qui contiennent des sous-menus.

Les UC concrets sont les menus qui conduisent à une activité.

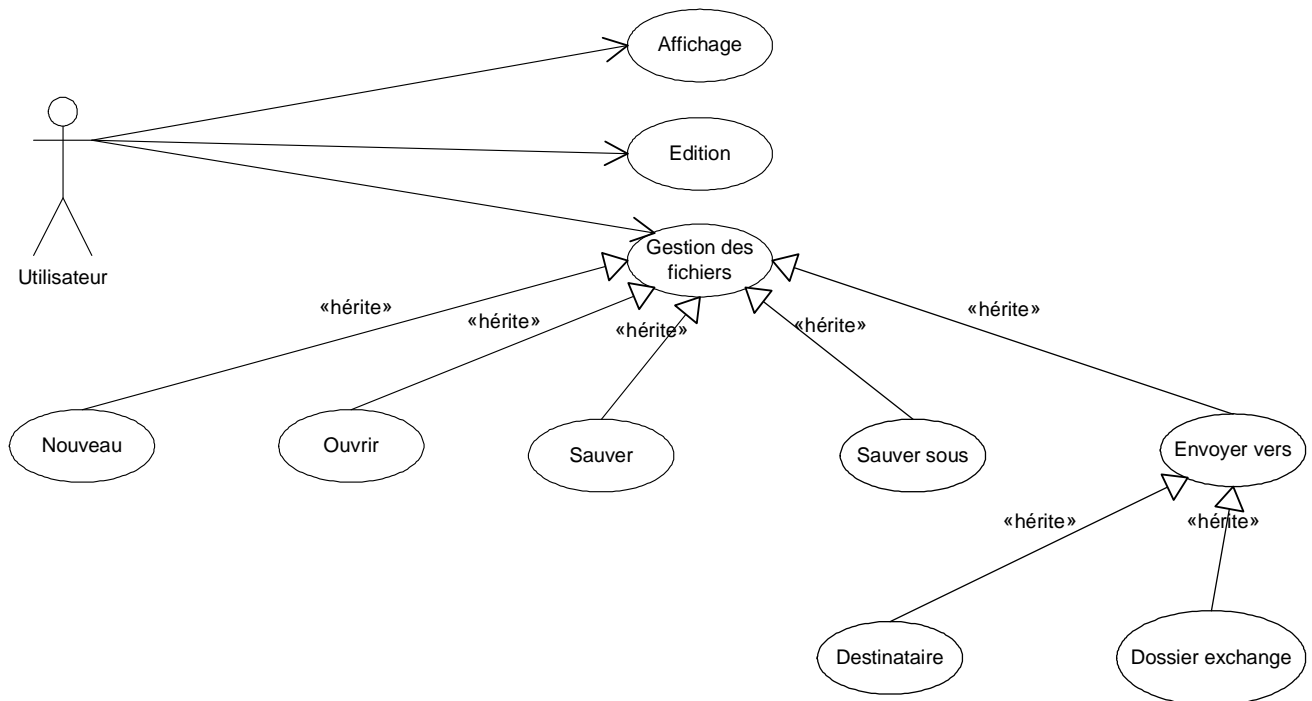
Arborescence des menus



Les UC concrets correspondent aux feuilles de l'arbre. Ils sont soulignés.

Les UC abstraits correspondent aux nœuds non-feuilles.

UC des menus



UC abstraits

Il y a 4 UC abstraits : « Edition », « Affichage », « Gestion de fichier » et « Envoyer vers ».

Ces UC abstraits regroupent des UC concrets.

UC concrets

Il y a 6 UC concrets : « Nouveau », « Ouvrir », « Sauver », « Sauver sous », « Destinataire », et « Dossier exchange ».

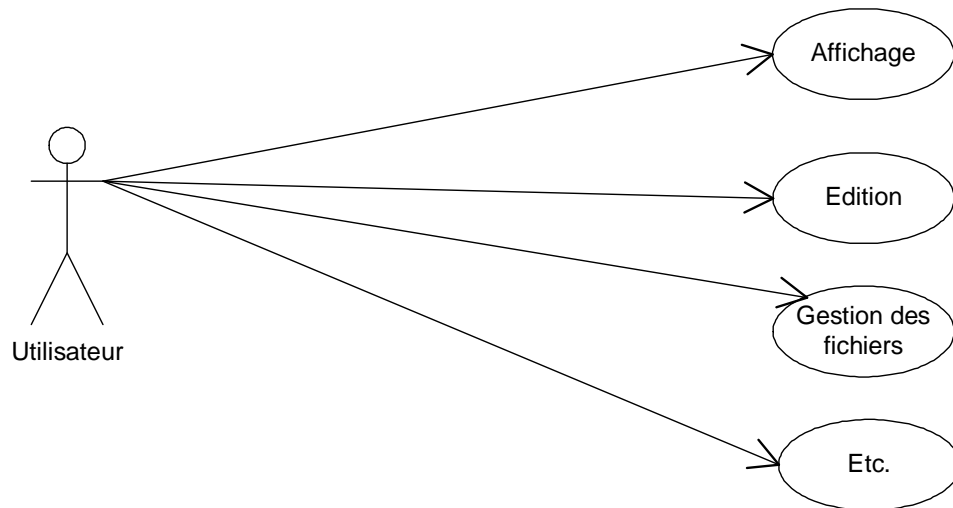
Les UC concrets correspondent à un usage concret du logiciel.

Niveaux de présentation des UC

On peut proposer un seul diagramme des UC : il risque d'être très embrouillé.

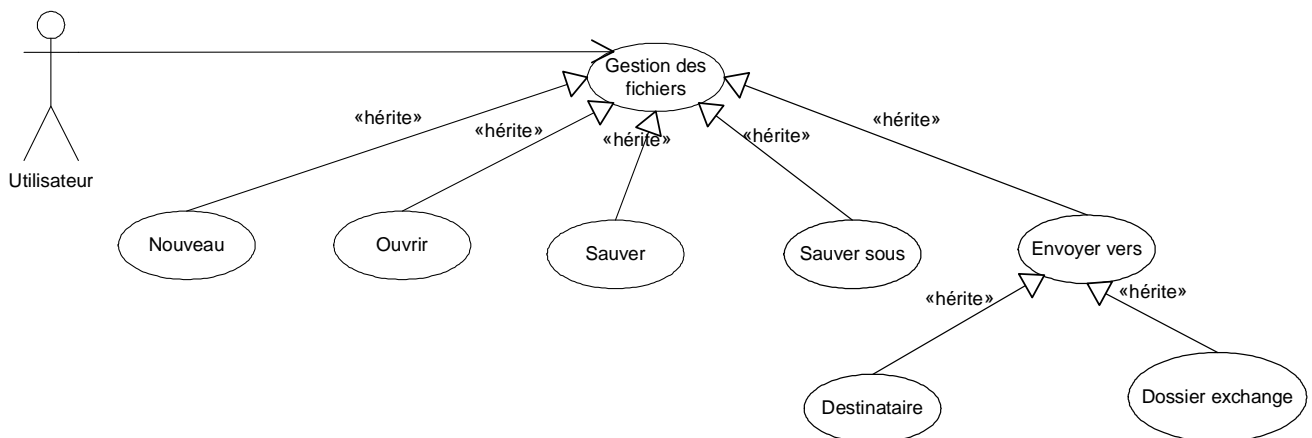
On aura donc intérêt à présenter plusieurs diagrammes d'utilisation par niveau d'abstraction descendant.

➤ *UC du menu général*



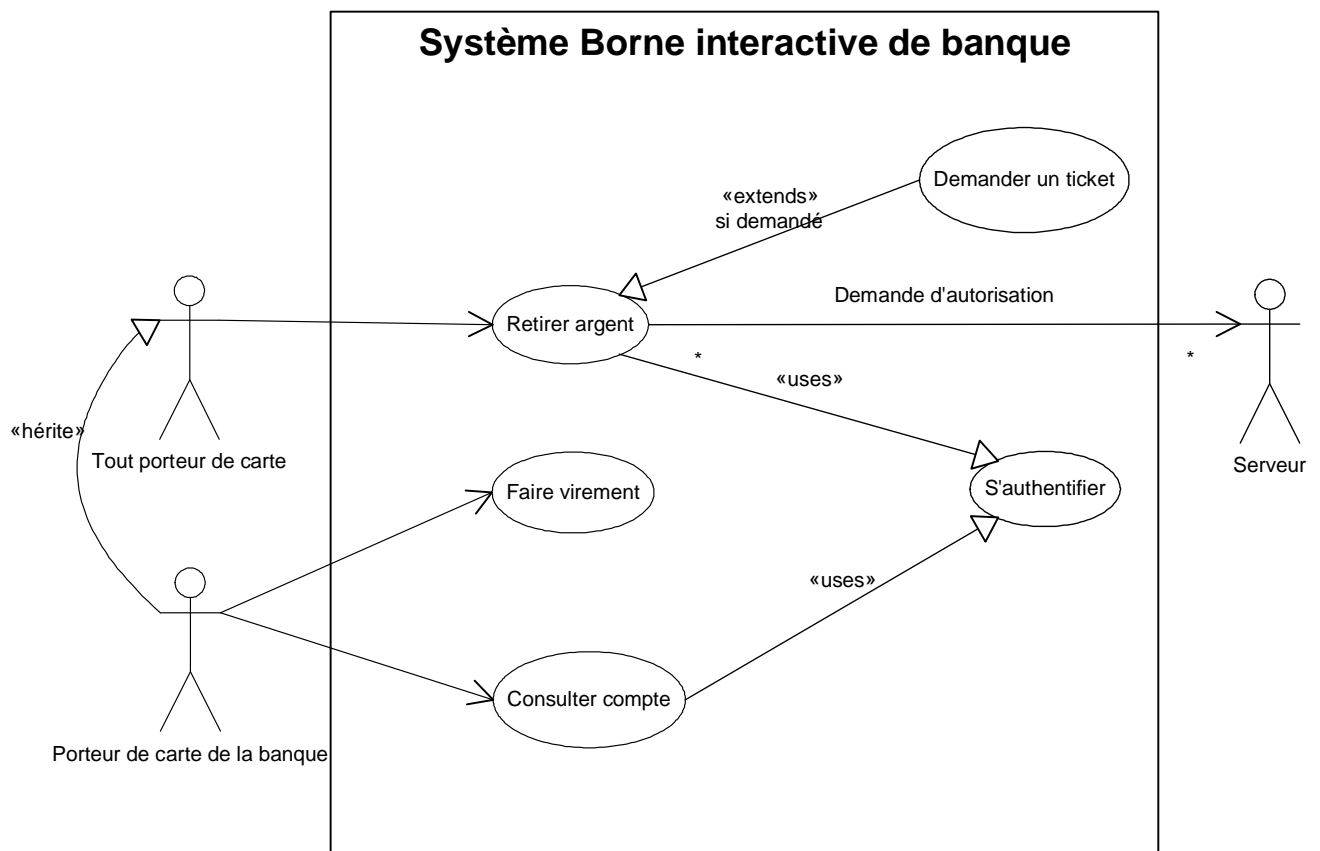
UC du menu général

➤ *UC du menu Gestion de fichier*



UC du menu Gestion de fichier

UC d'une borne de retrait d'argent : inclusion et extension



Remarques:
Le serveur est un acteur passif. C'est un logiciel.
Le lien Retirer argent -> serveur est orienté du fait de la passivité de l'acteur.
On nomme le lien pour faciliter la compréhension (sélectionner le lien, forme / action / option d'affichage)

UC concret

Il y a trois UC concrets: « retirer de l'argent », « effectuer un virement » et « consulter le compte ».

Composant : sous-UC inclus

Ces trois UC incluent le sous-UC « s'authentifier ». Ce sous-UC est un composant du UC concret.

Composant : sous-UC étendu

Le UC « retirer de l'argent » est étendu par le sous-UC « demander un ticket » à la condition que ce soit demandé. Ce sous-UC est un composant du UC concret.

Inclusion de UC

Un UC correspond au « film » du déroulement du programme pour une utilisation donnée. Un UC inclus est un morceau de ce film.

Il faut éviter de mettre trop de UC inclus pour éviter d'alourdir inutilement le diagramme des UC.

On met des UC inclus dans 3 cas :

1. Quand on pense que cela apporte quelque chose à la compréhension du diagramme
2. Quand le UC inclus est partagé par plusieurs UC.
3. Quand le UC inclus est aussi un UC pour un acteur.

Extension de UC

C'est le même principe que pour les inclusions :

Un UC correspond au « film » du déroulement du programme pour une utilisation donnée. L'extension d'un UC inclus est un morceau de ce film. Mais ce morceau ne s'exécute que sous condition.

Il faut éviter de mettre trop d'extension de UC pour éviter d'alourdir inutilement le diagramme des UC.

On met des UC étendus dans 3 cas :

1. Quand on pense que cela apporte quelque chose à la compréhension du diagramme.
2. Quand le UC étendu est partagé par plusieurs UC.
3. Quand le UC étendu est aussi un UC pour un acteur.

Les acteurs

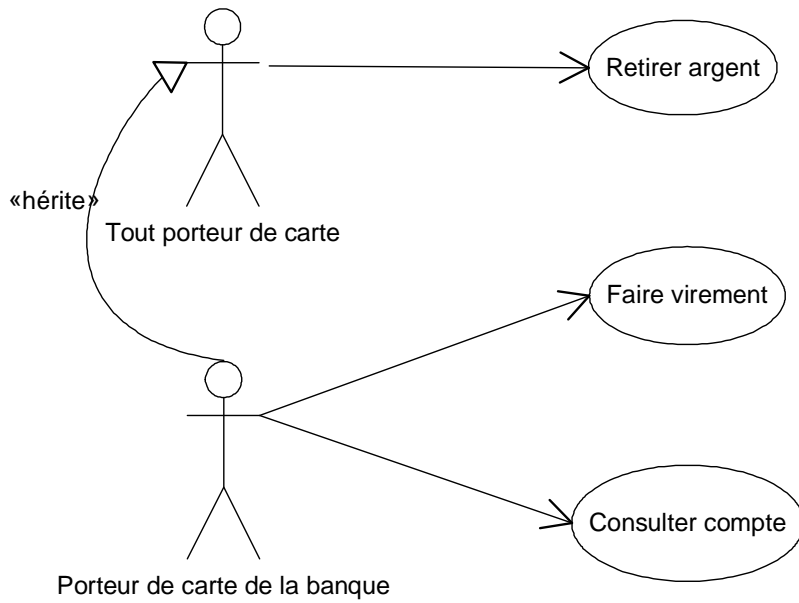
Généralisation des acteurs

Les acteurs peuvent être généralisés et inversement spécialisés.

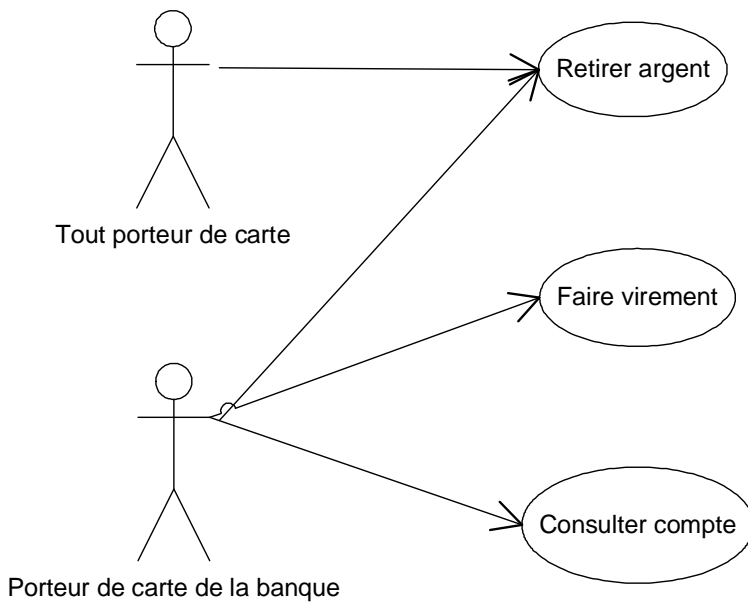
L'intérêt de la généralisation, c'est de montrer que certains acteurs héritent de tous les UC d'autres acteurs, et qu'ils ont en plus leur UC spécifiques.

Dans l'exemple traité, l'acteur « porteur de carte de la banque » peut consulter son compte et faire des virements. En plus de cela, il peut faire ce que peuvent faire tous les porteurs de carte, à savoir retirer de l'argent.

➤ **Version avec héritage**



➤ **Version équivalente sans héritage**



3. Les acteurs

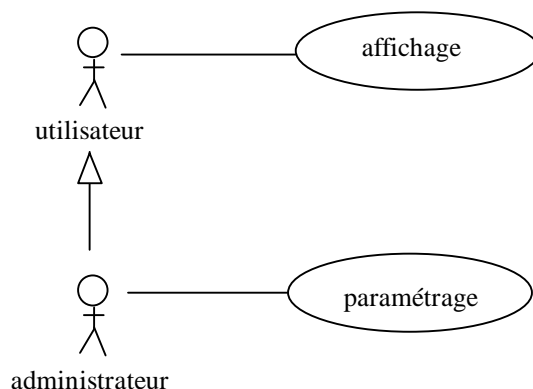
Relation entre les acteurs : la généralisation

Il n'y a qu'une seule relation possible entre les acteurs : la généralisation

L'acteur source est une espèce de l'acteur destination.

L'acteur destination est un genre de l'acteur source.

Exemple et formalisme



L'administrateur est une espèce d'utilisateur. Tous les administrateurs sont des utilisateurs. Donc les administrateurs accèdent aux UC des utilisateurs : ils accèdent à « affichage ». Par contre, les utilisateurs n'accèdent pas au paramétrage.

Catégories d'acteurs

Les distinctions ci-dessous permettent d'analyser complètement tous les acteurs du système.

Acteur principal vs Acteur secondaire

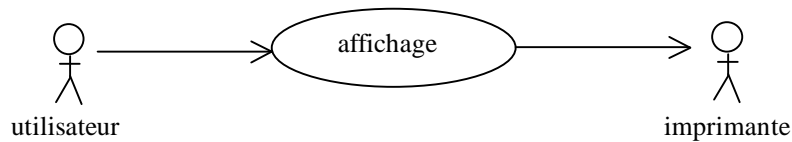
- L'acteur principal : l'utilisateur. Celui pour qui est fait le système.
- L'acteur secondaire : l'administrateur du système, etc.

Acteur humain vs Acteur mécanique

- **Les acteurs humains** sont les fonctions des personnes (le bibliothécaire, le client, l'administrateur) ou les services (la comptabilité).
- **Les acteurs mécaniques** peuvent être **matériels** (des périphériques), **logiciels** (un serveur, un autre système), ou **temporels** (une échéance).

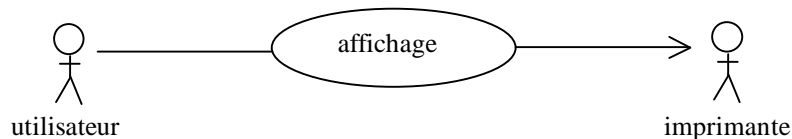
Acteur actif vs Acteur passif

- L'acteur actif est à l'origine du UC. Il utilise le système
- L'acteur passif n'est pas à l'origine du UC. Il est utilisé par le système.
- **Formalisme :**



Les flèches distinguent un acteur actif d'un acteur passif.

ou bien :



Par défaut, l'absence de flèche caractérise les acteurs actifs.

Acteur externe vs Acteur interne

- **Les acteurs externes** sont les acteurs « normal », à l'extérieur du système.
- **Les acteurs internes** sont internes au système. Ils correspondent soit à un acteur mécanique **temporel** (une échéance), soit à un **état particulier du système** nécessitant un usage particulier du système.

Bilan

En général :

L'acteur principal est une personne et est actif

Les logiciels sont des acteurs passifs et secondaires.

Notion d'événement

Un événement est un stimulus qui conduit à un usage du système.

Il y a trois types d'événement :

- **Les décisions externes** : c'est la principale cause de l'intervention d'un acteur. Le client décide de retirer de l'argent. Un système externe « décide » d'envoyer un message au système.

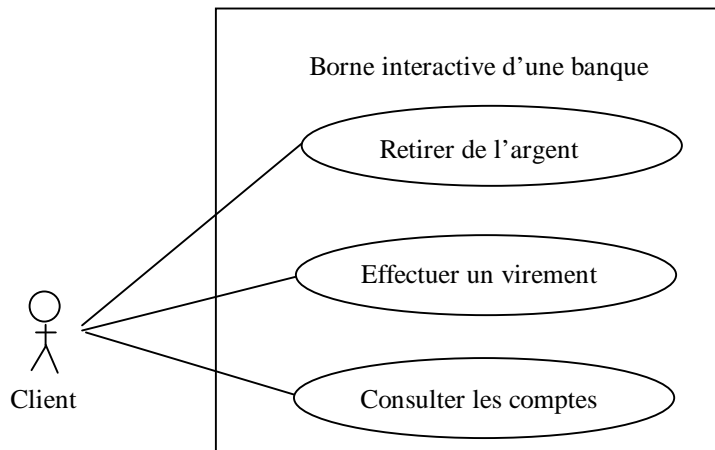
- **Les échéances** : ces événements peuvent conduire à une action automatique du système ou seulement intervenir dans une décision externe.
- **Les états** : ces événements correspondent à une **situation particulière du système** nécessitant un usage particulier du système. Comme pour les échéances, un tel événement peut conduire à une action automatique du système ou seulement intervenir dans une décision externe.

L'analyse des événements est une technique permettant la mise au jour des cas d'utilisation et des acteurs.

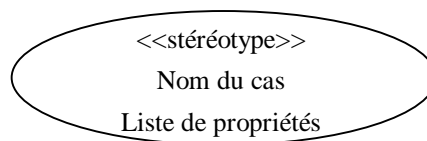
4. Les UC du logiciel – Formalisme

Description des UC

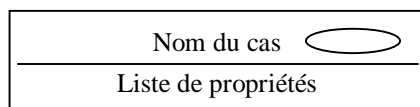
Exemple



UC

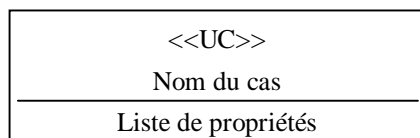


Ou bien



Le petit ovale symbolise le stéréotype <<UC >>

Ou bien



Stéréotype

Un stéréotype est une utilisation particulière d'un élément de modélisation. L'élément stéréotypé a un parent non stéréotypé. La syntaxe est la même pour l'élément stéréotypé et pour son parent, mais la sémantique est différente.

Le rectangle (qui est un classeur) est stéréotypé en <<UC >>

L'ovale (qui est un UC) peut être stéréotypé si on en voit l'utilité.

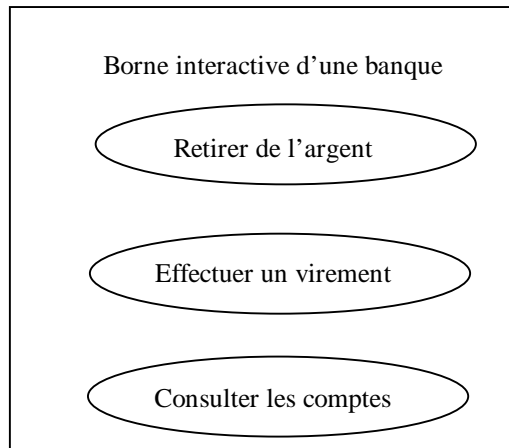
Le classeur

Un classeur est un élément de modélisation qui décrit une unité comportementale ou structurelle.

C'est la forme la plus simple du regroupement.

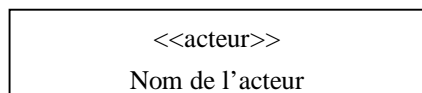
Un classeur se représente par un rectangle.

Le système complet est un classeur.



L'acteur

Ou bien



L'association entre acteur et UC

L'association est un lien entre un acteur et un UC.

Par défaut, l'association n'est pas orientée : cela signifie que la communication se fait dans les deux sens.

En orientant l'association dans un sens, on signifie la priorité d'un sens de communication sur un autre. Ce n'est qu'une priorité. Elle n'exclut pas la communication dans l'autre sens.

Usage :

Pour les acteurs actifs, on laisse une association non orientée.

Pour les acteurs passifs, on oriente l'association vers l'acteur.

Les 3 relations entre les UC

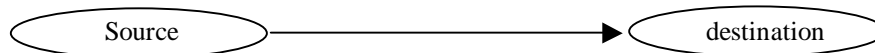
3 relations possibles entre les UC :

- généralisation
- inclusion
- extension

La généralisation correspond à la généralisation classique.

Inclusion et extension sont deux formes de la relation de composition.

La relation relie deux UC par une flèche. Du côté du départ de la flèche, on parle de source, du côté de l'arrivée de la flèche, on parle de destination.



La généralisation

Le UC source est une espèce du UC destination.

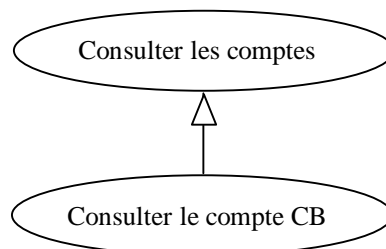
Le UC destination est un genre pour le UC source.

Par exemple

Le UC « consulter le compte carte bleue » est une espèce du UC « consulter les comptes »

Formalisme

Ce formalisme vaudra pour toutes les relations de généralisation / spécialisation.



L'inclusion

Le UC source comprend le comportement du UC destination.

Donc la destination est une partie nécessaire de la source : D inclut dans S.

On peut aussi dire que la destination est une étape obligée de la source.

Donc : si S alors D

si non D alors non S

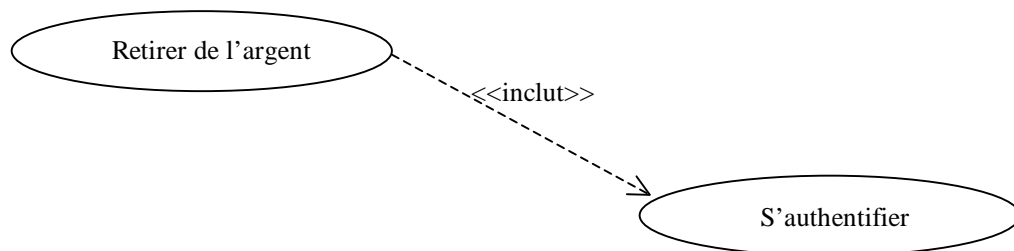
L'inclusion, comme la généralisation, sont des relations structurelles et nécessaires.

Par exemple

Le UC « Retirer de l'argent » inclut le UC « s'authentifier ».

« S'authentifier » est une étape obligée de « retirer de l'argent ».

Formalisme



L'extension

Le UC source ajoute son comportement au UC destination.

Il l'ajoute sous condition.

Remarque :

S'il n'y avait pas de condition, alors l'extension serait une inclusion inversée : le UC destination comprendrait le comportement du UC source.

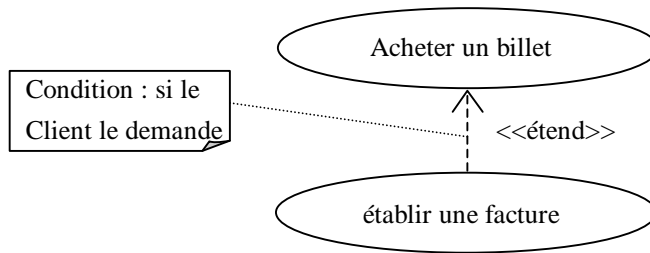
Mise à part la condition, l'extension peut être vue comme équivalente à l'inclusion (mais à l'envers). Ce qui distingue les deux, c'est le caractère nécessaire de l'une (inclusion) et le caractère accidentel de l'autre (extension).

Par exemple

Le UC « Acheter un billet » est étendu par le UC « établir une facture ».

En effet, le UC « établir une facture » sera effectué uniquement à la demande du client.

Formalisme



5. Méthode de construction d'un diagramme des UC

Recherche des acteurs

Est acteur du système tout ce qui est à l'extérieur du système et qui interagit avec le système.

Les principaux acteurs sont **les utilisateurs** du système.

Mais il ne faut pas oublier **ceux qui l'administrent** d'une façon ou d'une autre.

Ni les acteurs mécaniques :

- **Les périphériques** manipulés par le système (imprimantes, système de télétransmission, etc.)
- **Les logiciels** interfacés ou intégrés au système (serveur, etc.)

L'acteur principal d'un UC est celui pour qui le UC produit le résultat utile.

En général, l'acteur principal initie le UC, mais ce n'est pas toujours vrai.

Les autres acteurs du UC sont dits secondaires.

Recherche des UC

Les UC sont les finalités du système : ses objectifs, ce qu'on veut qu'il permette de réaliser.

Les UC doivent couvrir tous les besoins fonctionnels du système.

Par exemple :

Un logiciel de réservation de billet de train sur internet à trois UC :

- La recherche des voyages possibles
- La réservation des places (on peut réserver et payer plus tard).
- Le paiement

Autre exemple :

Dans le cas des retraits d'argent, le système interagit avec le système central qui donne les autorisations. Ce système central doit être représenté comme un acteur.

Dans ce système il y a trois UC :

- Le retrait d'argent
- La consultation de compte
- Les virements de compte à compte

Si on distinguait entre consultation du solde et consultation des opérations, on aurait deux espèces de UC pour le genre consultation de compte. Mais il n'y a pas d'acteur qui soit spécifiquement associé à l'un de ces UC sans être associé aussi au cas général.

Approche par abstraction : de l'UC abstrait à l'UC concret

On peut produire un diagramme des UC les plus abstraits. C'est celui qui donne la vision fonctionnelle du système la plus globale.

Ensuite, pour chaque UC abstrait, on peut entrer dans le détail et aller jusqu'au UC concrets : ceux qui seront finalement concrétisés en scénarios.

UC principaux

Les UC principaux sont les UC les plus abstraits : ce sont ceux qui sont directement associés à un acteur.

La liste des UC principaux doit couvrir exhaustivement toutes les fonctionnalités du système.

Les extensions, les inclusions et les spécialisations sont des précisions apportées au modèles d'UC principaux.

Recherche des UC par la recherche des événements

L'analyse des événements est une technique permettant la mise au jour des cas d'utilisation et des acteurs.

Recherche par le diagramme des flux et les diagrammes d'activité

Pour déterminer les UC et les acteurs externes, on peut reprendre la technique de l'analyse des flux de la méthode MERISE : diagramme organisationnel des flux, diagrammes d'activité correspondants, tableau des phases (préalable au MOT).

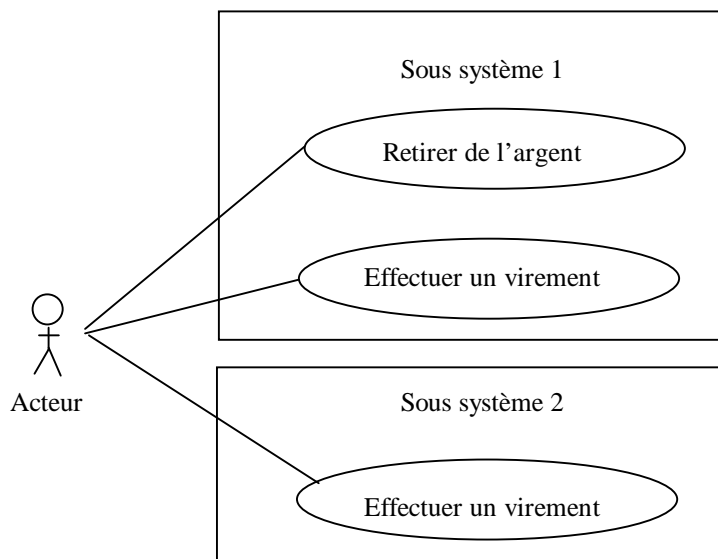
6. Architecture et diagramme des cas d'utilisation

Le diagramme des cas d'utilisation permet de représenter de deux façons les sous-systèmes de l'architecture système

Le sous système en tant que sous ensemble de cas d'utilisation

Les cas d'utilisation d'un sous-système peuvent être regroupés dans un classeur.

Formalisme : celui du classeur

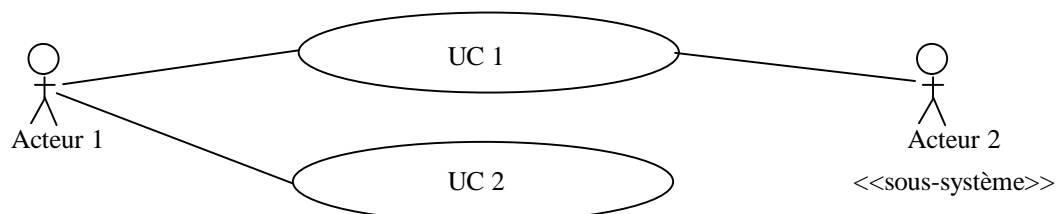


Le sous-système en tant qu'acteur

Un sous-système peut être considéré comme un acteur, et représenté comme tel. Ce sous-système sera acteur pour d'autres sous-système.

Formalisme : stéréotype « sous-système »

Les acteurs en tant que sous-système sont stéréotypés <<sous-système>>.



7. Compléments, limites et alternatives du diagramme des cas d'utilisation.

Compléments

Description du déroulement des cas d'utilisation

Le diagramme des cas d'utilisation est complété par un ensemble de diagrammes de séquence système qui permettent de détailler le déroulement effectif d'un scénario.

A ces diagrammes de séquence on peut aussi associer des diagrammes d'activité.

Maquettage de l'IHM

L'aspect visuel de l'utilisation n'est pas décrit par les diagrammes UML.

Un simple maquettage (dessin des écrans prévus) permet de clarifier l'usage réel du logiciel

Prototype

Sur certains points particulièrement complexes, on peut aussi produire un prototype (une simulation du logiciel final) pour pousser un peu plus loin la concrétisation de la conception.

Alternatives

Le diagramme des cas d'utilisation et les diagramme de séquence peuvent être remplacés par une analyse purement textuelle.

En général, on conserve au moins l'analyse des cas d'utilisation sous forme graphique pour faciliter la communication avec le maître d'ouvrage (le client, l'utilisateur final).

LE DIAGRAMME DE SEQUENCE « SYSTEME »

Il est facile de décrire la méthode encore que son application exige à coup sûr savoir et pratique.

1. Le diagramme de séquence système

ANALYSE FONCTIONNELLE	Diagramme UML
	Cas d'utilisation
	Séquence
	Activités

Présentation

UC, scénario, scénario nominal

Un UC est une abstraction.

Les scénarios correspondent aux instances concrètes d'un UC.

Le scénario nominal est le scénario pour lequel le UC est conçu.

Les scénarios alternatifs sont les scénarios alternatifs au scénario nominal.

Définition d'un diagramme de séquence « système »

Le diagramme de séquence « système » décrit les échanges entre l'utilisateur et le système, c'est-à-dire le logiciel à réaliser (et plus généralement entre le système et tous les acteurs) pour un ou plusieurs scénarios d'un UC donné.

On peut choisir de se limiter au diagramme de séquence système du scénario nominal.

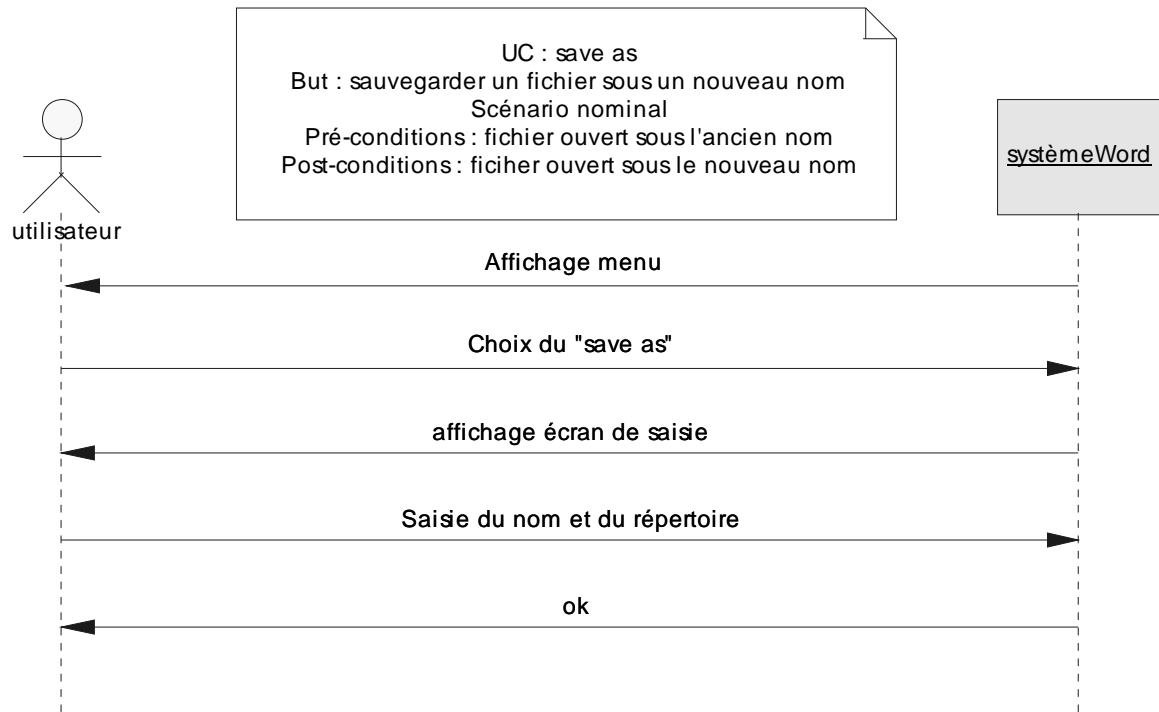
Version textuelle

La séquence des échanges entre l'utilisateur et le système pour un UC donné peut être présentée de façon uniquement textuelle.

Dans ce cas, on présente tous les scénarios de l'UC et non pas seulement le scénario nominal.

Diagramme de séquence « système » du « save as »

Diagramme de séquence du scénario nominal du UC : « save as »



➤ *Syntaxe des diagrammes de séquence « système » :*

En commentaires, on précise les caractéristiques du diagramme : UC concerné, but de l'UC, cas nominal ou alternatif, pré-condition, post-condition.

L'utilisateur communique avec le « système » considéré comme un objet : ici c'est l'objet « systèmeWord ». Il est souligné, la première lettre du nom de l'objet est en minuscule.

Pré-conditions et post-conditions décrivent l'état du système avant et après l'usage de l'UC.

———➔ **Message synchrone** : une fois le message envoyé, l'expéditeur est bloqué jusqu'à ce que le destinataire accepte le message.

———> **Message asynchrone** : le message envoyé n'interrompt pas l'exécution de l'expéditeur.

———➤ Caractère synchrone ou asynchrone est indéterminé. Attention, cette représentation peut aussi être utilisée pour les messages asynchrones.

Remarque : on pourrait se passer des deux premiers échanges : menu et choix du « save as » et entrer directement dans le déroulement du « save as » : affichage de l'écran de saisie.

Description textuelle du UC : « save as »

On peut aussi décrire les scénarios de façon uniquement textuelle (on n'utilisera pas cette méthode).

En général dans ce cas, on décrit à la fois le cas nominal et les cas alternatifs.

Identification

Nom du cas : **Save as**

But : sauvegarder dans un autre fichier

Acteur principal : l'utilisateur

Acteur secondaire : aucun

Séquencement

Un fichier a été ouvert

Pré-conditions (état du système avant l'opération).

Fichier ouvert sous l'ancien nom

Enchaînement nominal

1.0 Affichage du menu

2.0 Choix du save as

3.0 Affichage de l'écran de saisie du save as

4.0 Saisie du nom et du répertoire du nouveau fichier et validation.

5.0 L'application renvoie un message de confirmation de l'opération.

Post-conditions (état du système après l'opération)

Fichier ouvert sous le nouveau nom.

Alternative n°1

4.1 Annulation en cours de saisie (4.1 est une alternative à 4.0)

Post-conditions (état du système après l'opération)

Fichier ouvert sous l'ancien nom.

Alternative n°2

5.1.1 Message d'erreur : le disque n'est pas accessible

5.1.2 Validation du message

Retour en 3.0

➤ *Syntaxe des descriptions textuelles des diagrammes de séquence « système »*

On décrit l'enchaînement nominal, puis les alternatives. Les alternatives peuvent conduire à la fin de l'UC ou à un retour à une étape du cas nominal.

Toute sortie de l'UC s'accompagne d'une description des post-conditions.

Remarques

La description textuelle permet de représenter les alternatives et les répétitions.

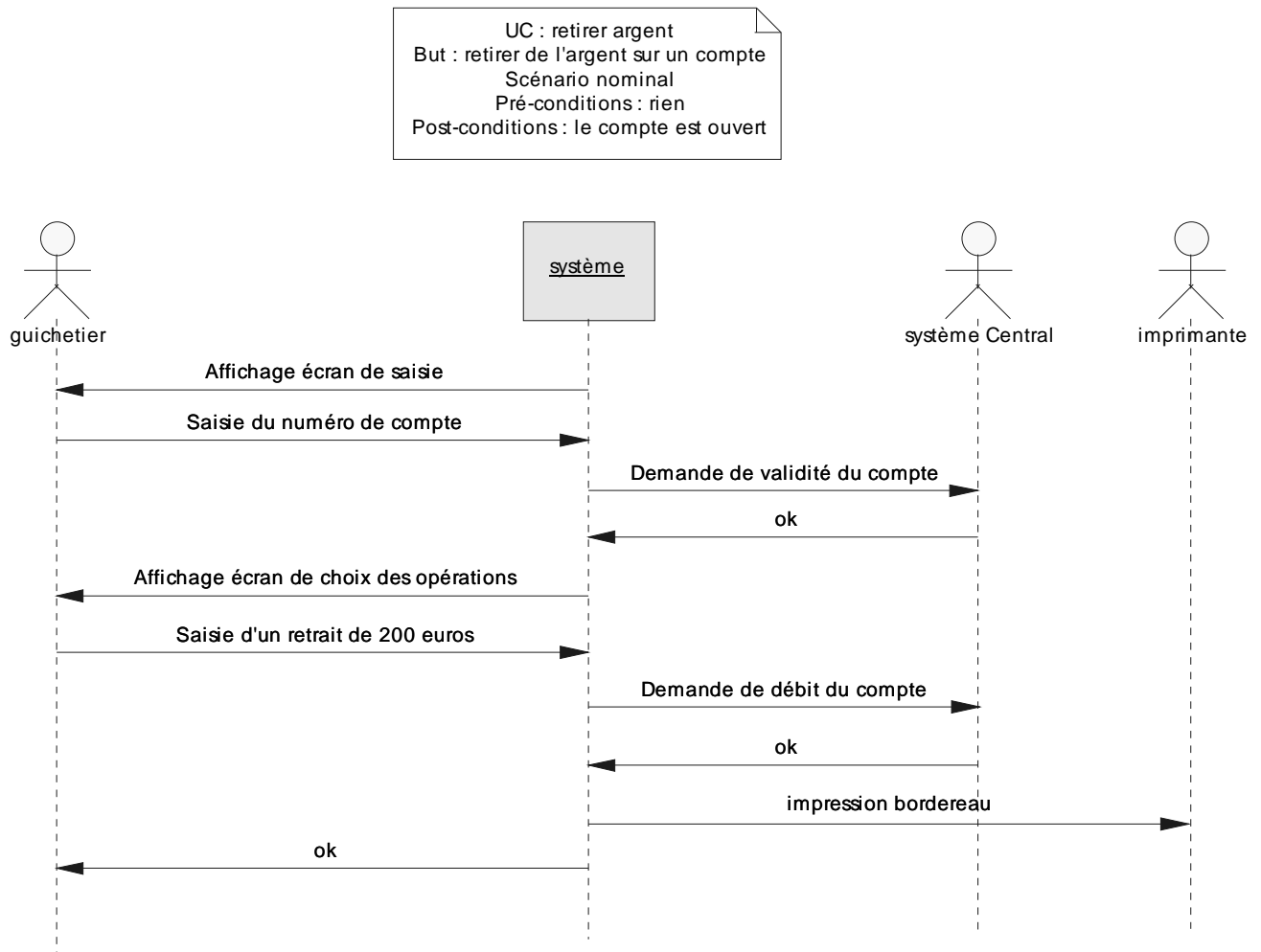
On n'a pas présenter cette possibilité dans les diagrammes de séquence, toutefois la syntaxe UML le permet, quoique ce soit peu recommandé et peu utilisé pour des questions de lisibilité.

Pour représenter les alternatives, on préférera l'utilisation de diagrammes d'activités.

Diagramme de séquence « système » de « retirer de l'argent »

Diagramme de séquence du scénario nominal du UC : retirer de l'argent.

On s'intéresse au logiciel d'une banque : un client vient retirer de l'argent sur son compte. C'est le guichetier qui gère l'opération.



➤ Syntaxe des diagrammes de séquence « système »

Dans cet exemple, on a **2 acteurs secondaires** qui sont des machines : le système central (un logiciel) et l'imprimante (une machine physique).

Remarques

On a choisi d'entrer dans l'UC par l'ouverture du compte et d'en sortir avec un compte ouvert. On pourrait aussi choisir de fermer le compte dans l'UC ou d'entrer dans l'UC avec un compte ouvert.

Description textuelle du UC : retirer de l'argent.

Identification

Nom du cas : **retirer de l'argent**

But : opération de retrait d'argent par un guichetier de banque

Acteur principal : le guichetier

Acteurs secondaires : le système central, l'imprimante

Séquencement

L'UC est utilisable à tout moment

Pré-conditions (état du système avant l'opération).

rien

Enchaînement nominal

- 1.0 Le système affiche un écran de saisie
- 2.0 Le guichetier saisit le numéro de compte du client
- 3.0 L'application demande la validité du compte auprès du système central
- 4.0 Le système central valide le compte
- 5.0 L'application affiche un écran de choix des opérations
- 6.0 Le guichetier sélectionne un retrait d'espèces de 200 euros
- 7.0 L'application demande au système central de débiter le compte.
- 8.0 Le système central valide la demande
- 9.0 Le système imprime un bordereau
- 10.0 Le système notifie au guichetier qu'il peut délivrer le montant demandé.

Post-conditions (état du système après l'opération).

Le compte est ouvert pour d'autres opérations

Alternative n°1

- 4.1 Le système central ne valide pas le compte (alternative à 3.0)
- Retour en 1.0

Alternative n°2

- 8.1 Le système central ne valide pas le retrait (alternative à 7.0)
- Retour en 5.0

Remarques :

La description textuelle permet de représenter les alternatives et les répétitions.

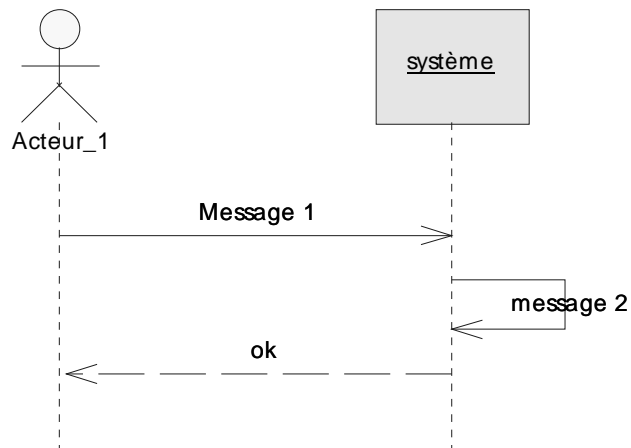
On n'a pas présenté cette possibilité dans les diagrammes de séquence, toutefois la syntaxe UML le permet, quoique ce soit peu recommandé et peu utilisé pour des questions de lisibilité.

Pour représenter les alternatives, on préférera l'utilisation de diagrammes d'activités.

2. Architecture et diagramme de séquence système

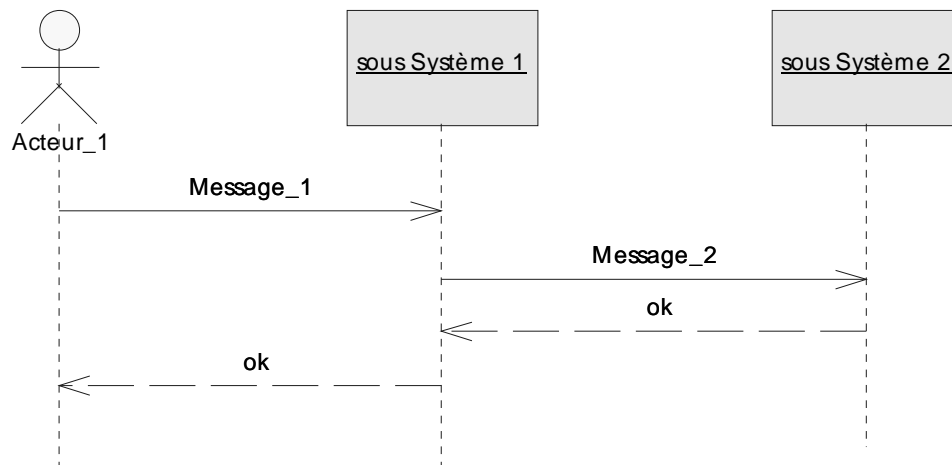
Appel réflexif sur l'objet « système »

On peut montrer des appels réflexifs qui signifient l'existence de communication entre sous-systèmes :



Définition de plusieurs objets « sous système »

On peut représenter plusieurs sous-systèmes dans le diagramme de séquence système :



3. Modularité, test et boucle

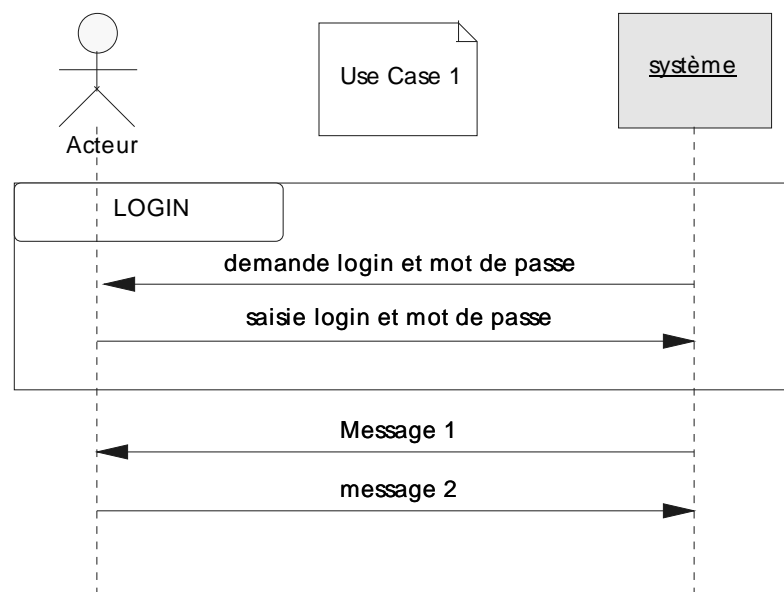
Modularité

On peut regrouper dans un module une séquence d'échanges entre l'utilisateur et le système.

Ce module pourra être utilisé dans d'autres diagrammes de séquence sans avoir à re-détailler son contenu.

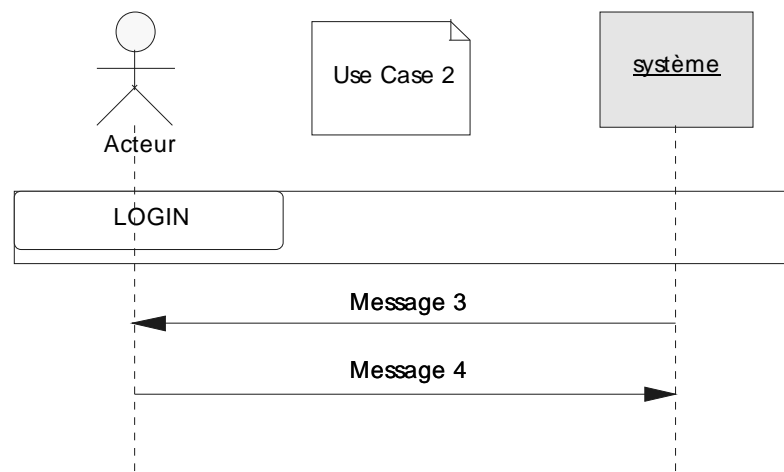
Exemple et syntaxe :

➤ *Diagramme de séquence du Cas d'utilisation 1 :*



➤ *Diagramme de séquence du Cas d'utilisation 2 :*

On réutilise le module « LOGIN » :



Test et boucle

On peut représenter des tests et des boucles dans les diagrammes de séquence. Toutefois, au niveau des diagrammes de séquence, il est préférable de ne pas le faire et de privilégier une lecture facilitée en se limitant à un seul scénario par diagramme de séquence système.

Les diagrammes d'activités sont plus appropriés pour représenter les tests et les boucles. On les utilisera pour représenter la totalité des scénarios d'un cas d'utilisation.

LE DIAGRAMME D'ACTIVITE DES SCENARIOS

Il est facile de décrire la méthode encore que son application exige à coup sûr savoir et pratique.

1. Les diagrammes d'activité dans l'analyse fonctionnelle

ANALYSE FONCTIONNELLE	Diagramme UML
	Cas d'utilisation
	Séquence
	Activités

Présentation

Le diagramme d'activité montre l'enchaînement des actions (une action est une partie de l'activité) et des décisions au sein d'une activité du système, ou au sein de tout le système (d'où son utilisation en analyse fonctionnelle et en analyse organique).

Le diagramme d'activité correspond aux organigrammes de l'algorithmique classique.

L'activité permet de représenter n'importe quel blocs d'instructions, de la simple instruction au programme complet en passant par la fonction ou n'importe quel morceau de code.

Les tests sont représentés par des transitions

Les boucles sont représentées par un retour en arrière dans le flux des activités.

Eléments des diagrammes d'activité et formalisme UML

Activité

Une activité est l'exécution d'un comportement, autrement dit un programme ou un bout de programme qui s'exécute.

Action

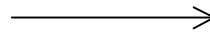
Une action est une partie de l'activité.



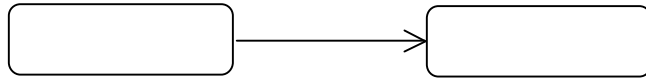
Une activité est par exemple constituée de plusieurs actions qui se déroulent l'une après l'autre : elles seront reliées par des transitions.

L'action permet de représenter n'importe quel blocs d'instructions, de la simple instruction au programme complet en passant par la fonction ou n'importe quel morceau de code.

Transition



Une transition relie deux actions entre elles (source - destination).



Une transition peut être réflexive (ce sera alors forcément une transition conditionnelle).

Elle montre le passage d'une activité à l'autre.

En général, elle est déclenchée par la fin du comportement de l'activité source.

Etat initial



L'état initial montre le point de départ de la première activité.

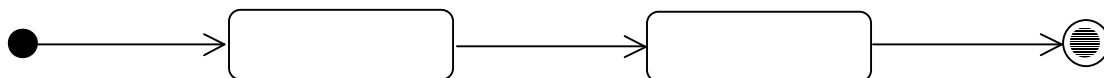
Il n'y a qu'un seul état initial par activité.

Etat final



L'état final montre le point d'arrivée de la dernière activité.

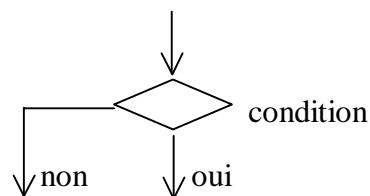
Il peut y avoir plusieurs états finaux pour une activité.



Transition conditionnelle = décision

C'est un aiguillage dans une transition.

On peut préciser la condition de l'aiguillage.



On précise les conditions de passage dans les différentes branches de l'aiguillage.

Attention : la transition conditionnelle indique que le flux va partir d'un côté ou de l'autre. Mais attention : il n'y a pas d'activités (de saisie, d'affichage, d'affectation) à ce niveau. Le choix a été fait au niveau de l'activité qui précède.

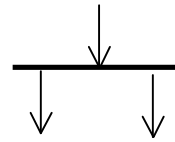
La transition correspond au test de la programmation impérative (C, C++, etc.)

Etat à sous-activité

Un état à sous-activités « encapsule » un diagramme d'activité.

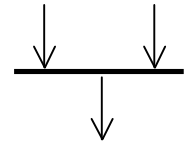
Synchronisation fourche ou Transition fourche

Certaines activités peuvent se dérouler en parallèle :



Synchronisation jonction ou Transition jonction

Certaines activités peuvent être
déclenchées par la fin coordonnée de plusieurs autres

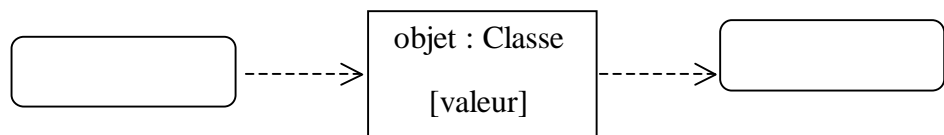


Travée ou Couloir

Les travées répartissent les activités par responsable des activités (personnes ou service).

On peut aussi représenter les acteurs dans les travées.

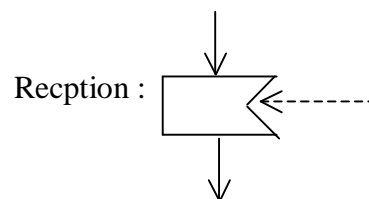
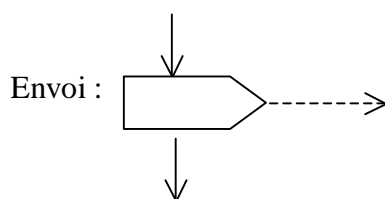
Transition à objet



On peut faire apparaître les objets dans une transition entre deux activités

Les objets représentés sont ceux qui initient des activités, qui sont utilisés par des activités, ou qui sont modifiés par des activités.

Transition à signal : envoi et réception



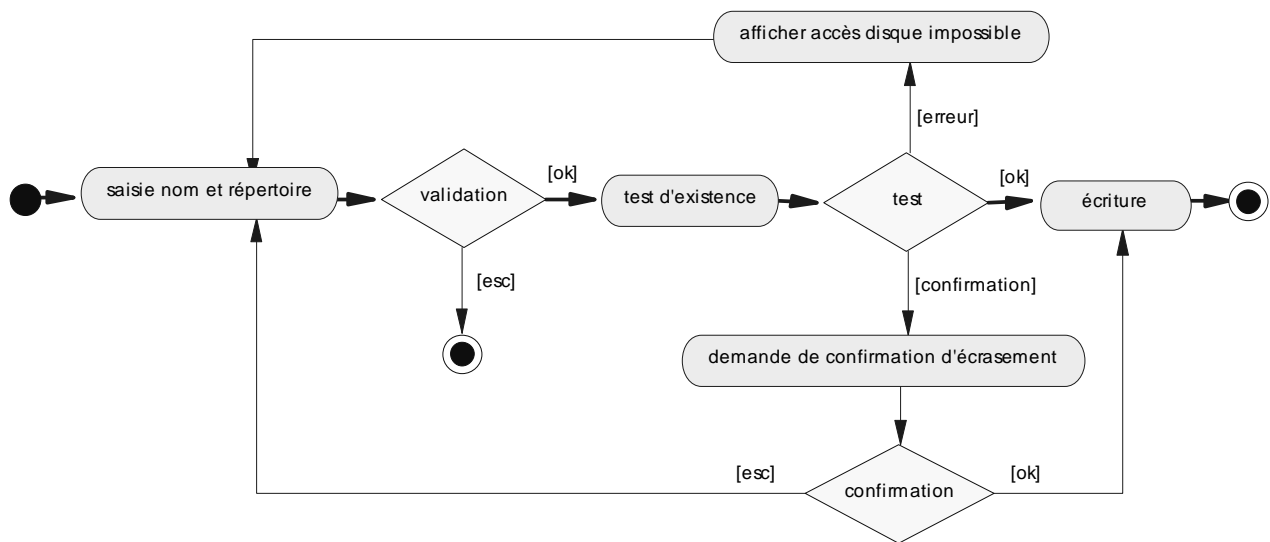
On peut faire apparaître l'envoi ou la réception d'un signal entre deux activités

2. Diagramme d'activité d'un UC : tous ses scénarios

On peut représenter tous les scénarios d'un UC dans un diagramme d'activité.

On utilisera cette possibilité dans le projet.

Exemple : le save as



Dans un tel diagramme, on aura intérêt à essayer de faire apparaître le scénario nominal. Dans notre exemple, il correspond au chemin tout droit. On pourrait aussi mettre des transitions en gras.

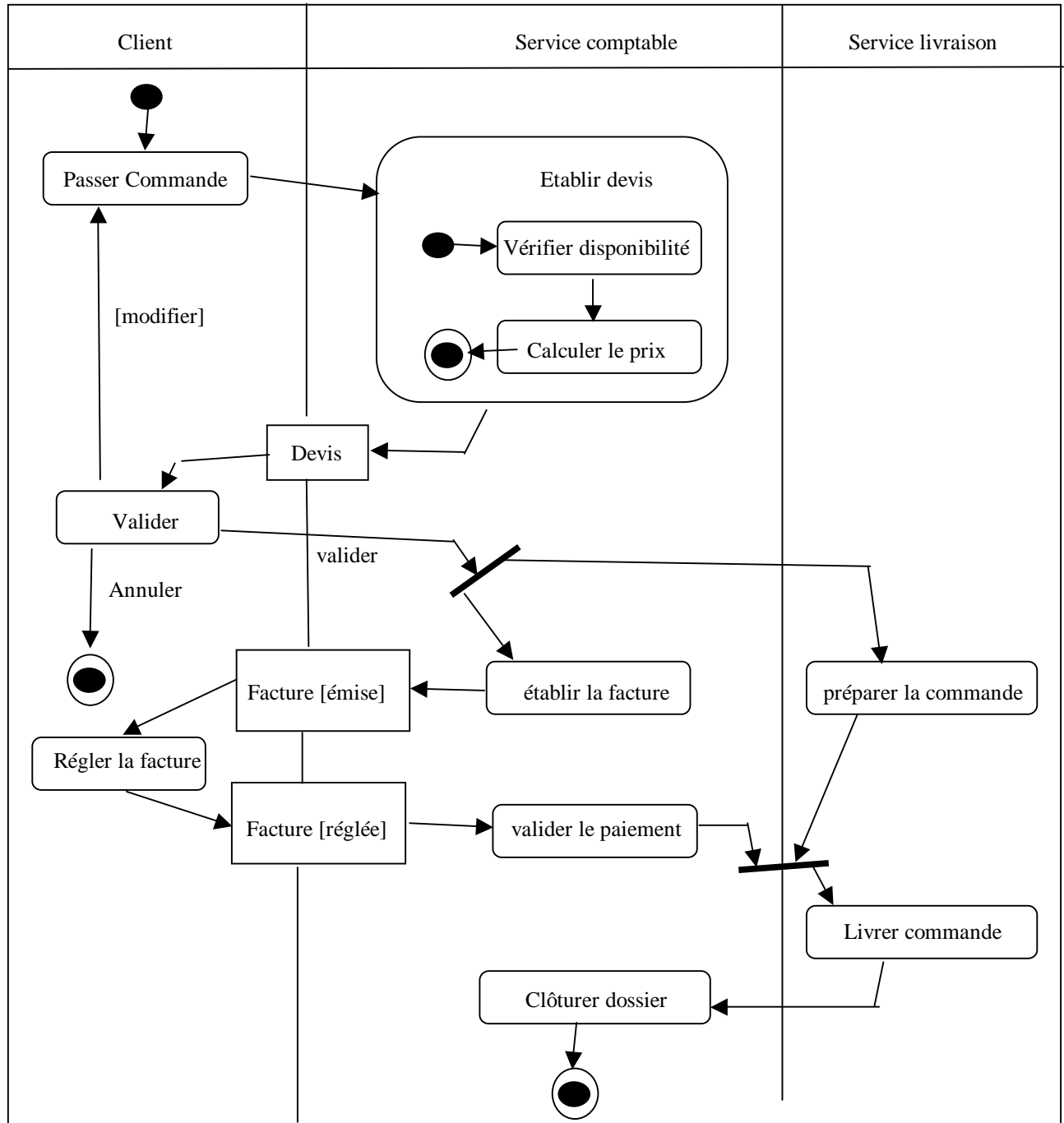
3. Diagramme d'activité d'un traitement MERISE

On peut représenter un traitement au sens MERISE avec un diagramme d'activité.

Un traitement au sens MERISE est un ensemble d'activités coordonnées dans le temps qui pourront être réalisées par plusieurs acteurs humains en utilisant plusieurs logiciels.

L'analyse d'un traitement MERISE permet d'analyser les relations entre les différents acteurs humains indépendamment des relations directes avec le logiciel à réaliser. Dans le cas d'un système d'information complexe, son utilisation peut permettre une meilleure compréhension du cahier des charges et permettre ainsi une mise au jour facilitée des UC du logiciel à réaliser.

Exemple : diagramme d'activité du traitement PasserCommande



UML2, p. 165

● Etat initial

Etat final



Synchronisation

Activité

→ transition

Classe / Objet