

3ème année licence

Option: SCI

Module: CL2

Chapitre 2: Expression des besoins



Dr. Meriem Kern



1. Expression des besoins

- Objectif : Comprendre les besoins du client pour rédiger le cahier des charges
- Principe :
 - ❑ Définir les limites du système.
 - ❑ Définir l'environnement du système : les utilisateurs ou éléments qui interagissent avec le système
 - ❑ Définir les utilisations principales du système : à quoi sert-il ?

1. Expression des besoins

○ **Capture des besoins**

- ❑ collecte des informations : interviews, lecture de documentation
- ❑ chercher à comprendre : (1) le domaine (2) le problème

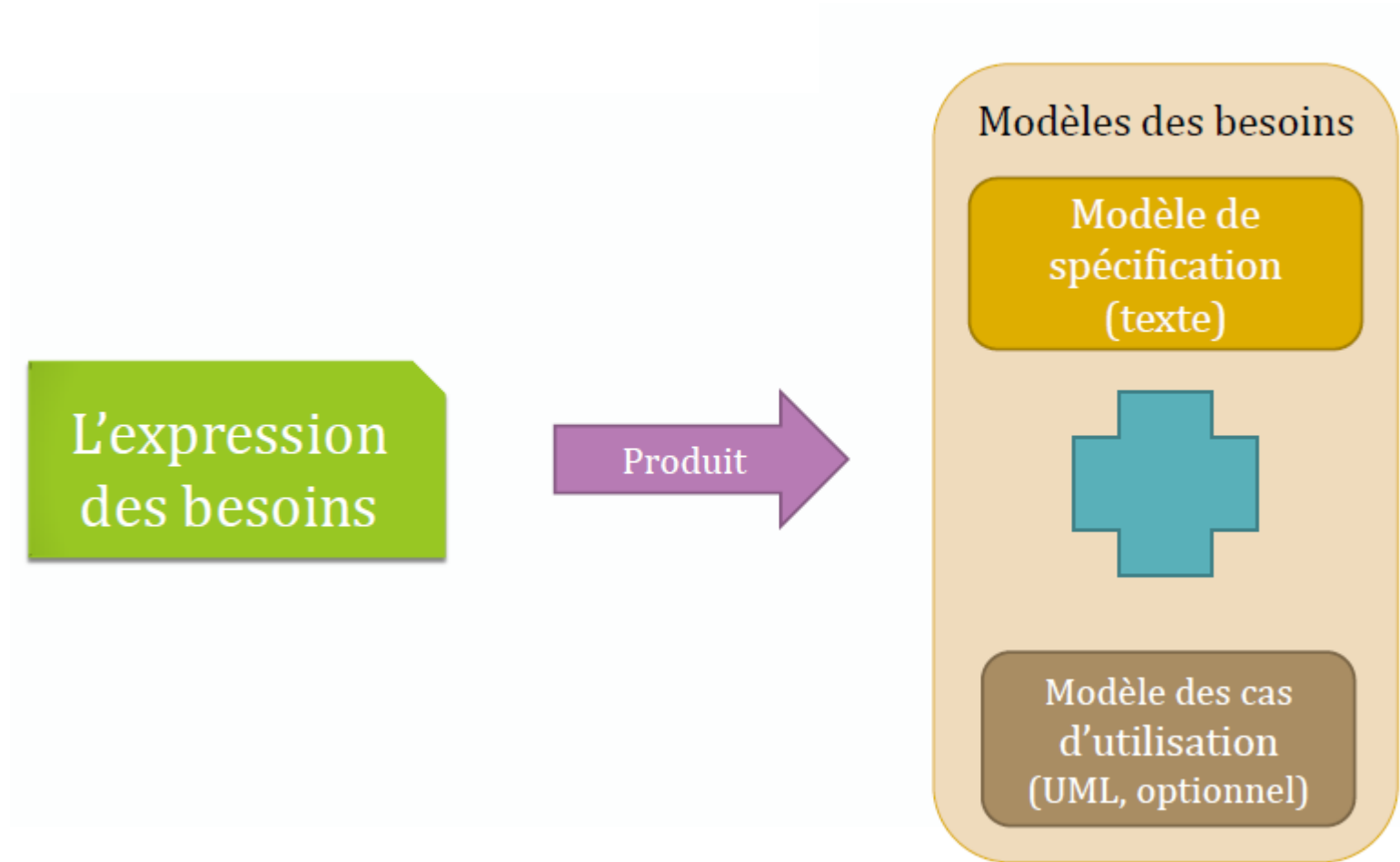
○ **Définition des besoins**

- ❑ Restitution dans un langage compréhensible par le client
- ❑ Identification, structuration, définition d'un dictionnaire

○ **Spécification des besoins**

- ❑ Spécification détaillée des besoins, plus formel
- ❑ Utile pour le client, mais aussi pour les développeurs

1. Expression des besoins



2.Besoins

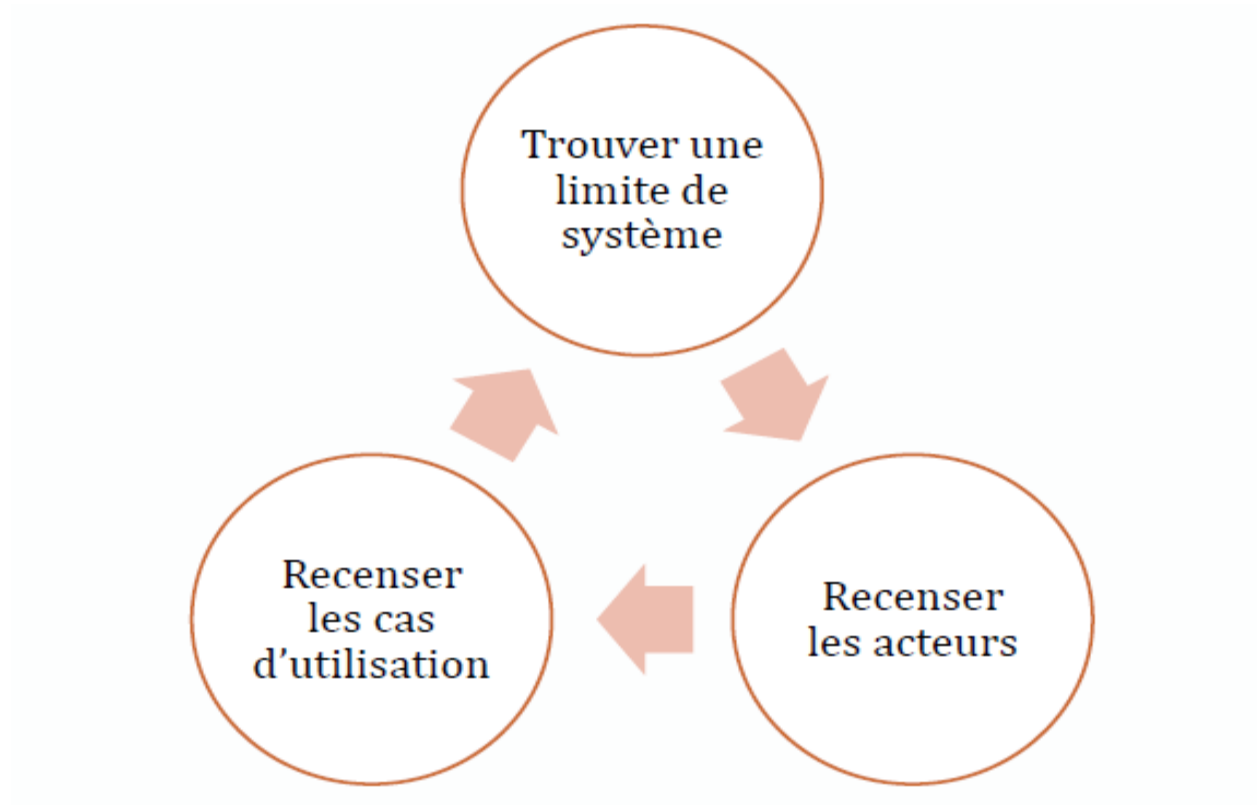
- **Exigences fonctionnelles**

- à quoi sert le système
- ce que doit faire le système, les fonctions utiles.

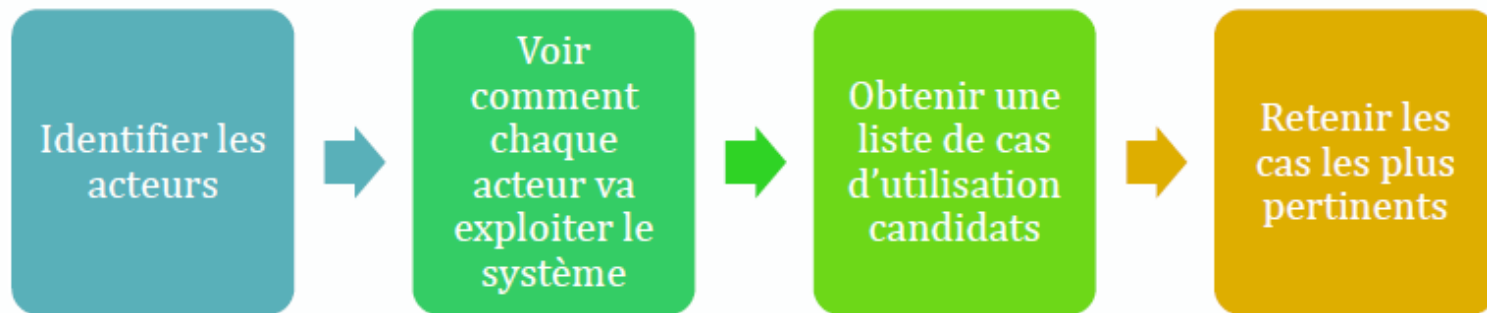
- **Exigences non fonctionnelles**

- Performance, sûreté, confidentialité, portabilité,

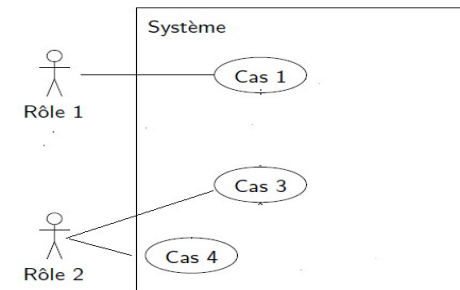
2. Modèle de cas d'utilisation



2. Modèle de cas d'utilisation

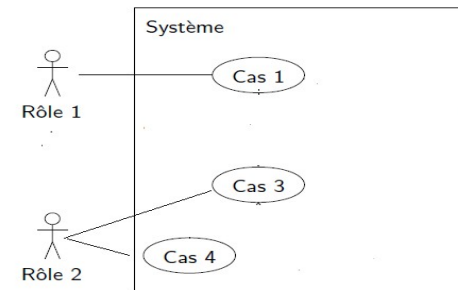


3. Cas d'utilisation



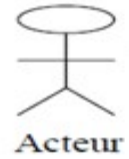
- Un cas d'utilisation est un diagramme qui modélise une interaction entre le système informatique à développer et un utilisateur ou acteur interagissant avec le système.
- Permettent de définir les besoins des utilisateurs et les fonctionnalités du système.
 - Limitation du système
 - Relation avec son environnement
 - Fonctions attendues

3. Cas d'utilisation



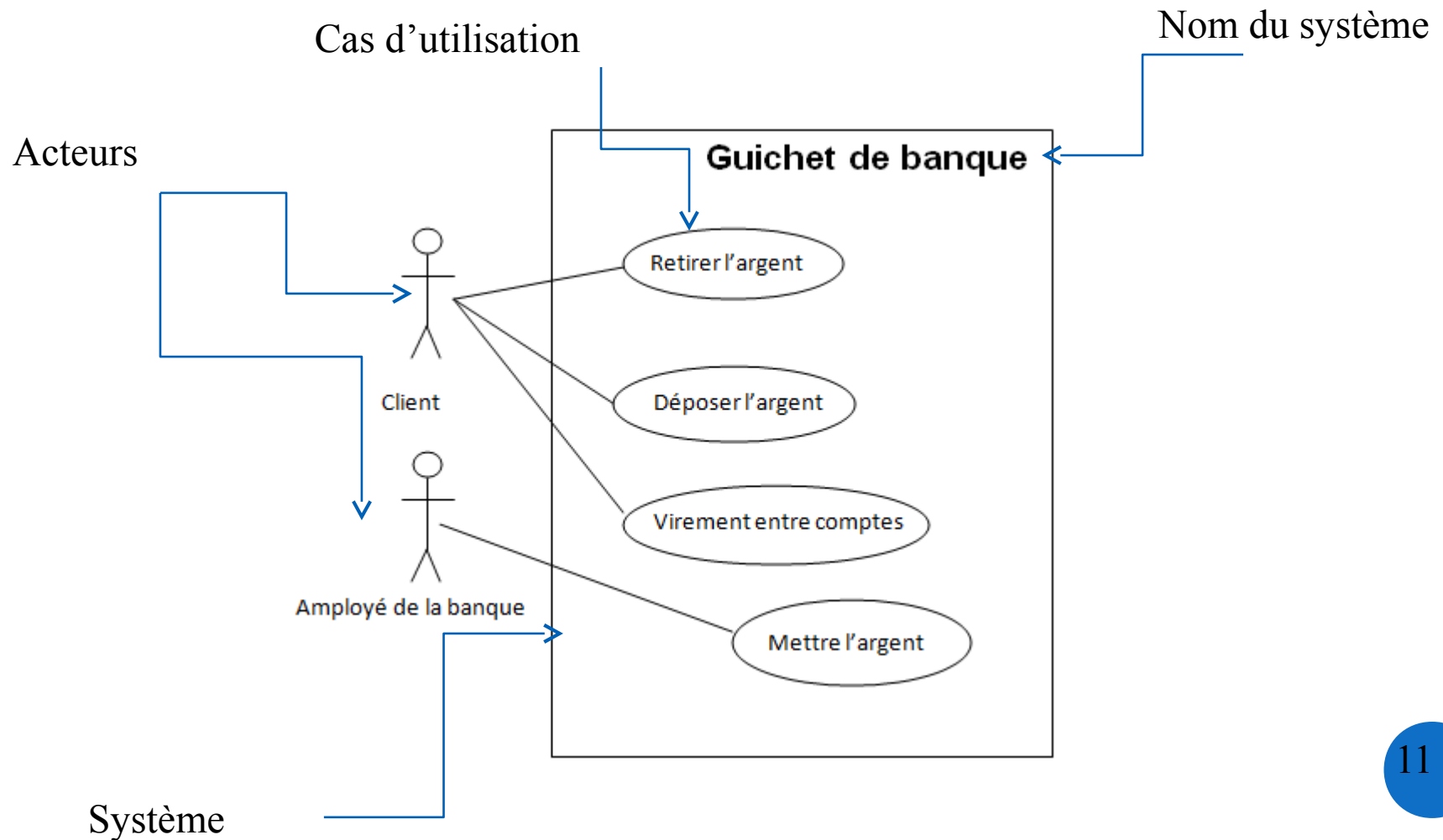
- ❑ Les cas d'utilisation décrivent sous la forme d'actions et de réactions, le comportement du système étudié du point de vue des utilisateurs. Ils définissent les limites du système et ses relations avec son environnement.
- ❑ Entre un utilisateur et le système, un cas d'utilisation décrit les interactions liées à un objectif fonctionnel de l'utilisateur.
- ❑ Les cas d'utilisation font apparaître les besoins fonctionnels et leur ensemble constitue le modèle des cas d'utilisation **qui** décrit les objectifs (fonctionnalités) du système.

4. Acteur



- Personne ou système qui interagit avec le système en échangeant de l'information
- Un utilisateur externe du système peut jouer différents rôles vis à vis du système.
- Il existe 2 catégories principale d'acteurs:
 - Les acteurs primaires: pour lesquels l'objectif du cas d'utilisation est essentiel ;
 - Les acteurs secondaires: qui interagissent avec le cas d'utilisation mais dont l'objectif n'est pas essentiel.

5. Exemple : Guichet de banque



6. Relations entre cas d'utilisation

1- **Une relation d'inclusion** : formalisée par la dépendance « **include** »

- ❑ Dans une relation « include », le cas d'utilisation de base utilise systématiquement les enchaînements provenant du cas inclus.

2- **Une relation d'extension** : formalisée par la dépendance « **extend** »

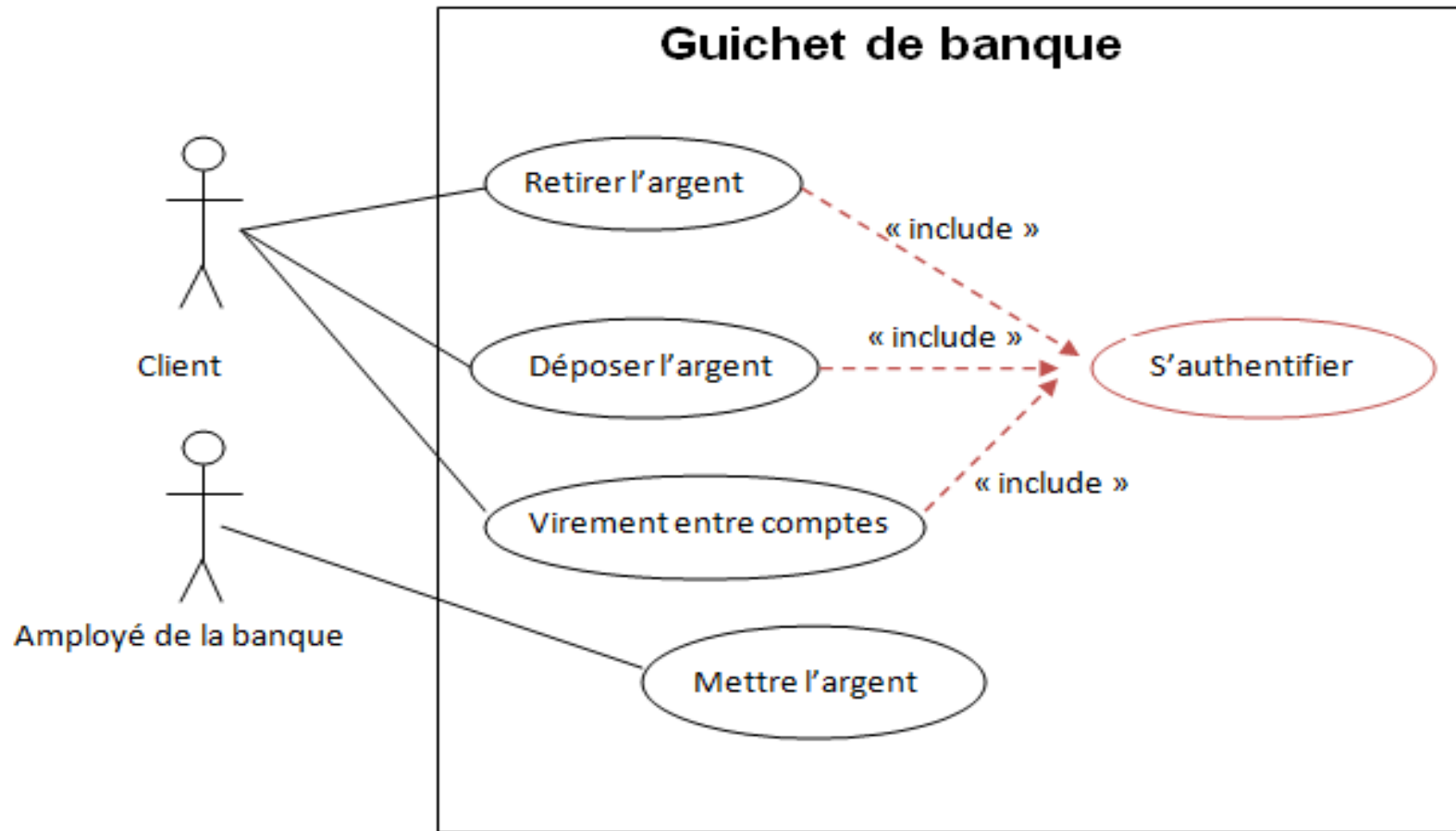
- ❑ On utilise principalement cette relation pour séparer le comportement **optionnel** (les variantes) du comportement obligatoire

6. Relations entre cas d'utilisation

- ✓ La relation <<extend>> montre une possibilité d'exécution d'interactions qui augmenteront les fonctionnalités du cas étendu, mais **de façon optionnelle, non obligatoire, alors que la relation <<include>> suppose une obligation d'exécution des interactions.**

3- Relation Héritage (généralisation/ spécialisation) : le cas d'utilisation dérivé est une spécialisation du cas d'utilisation parent.

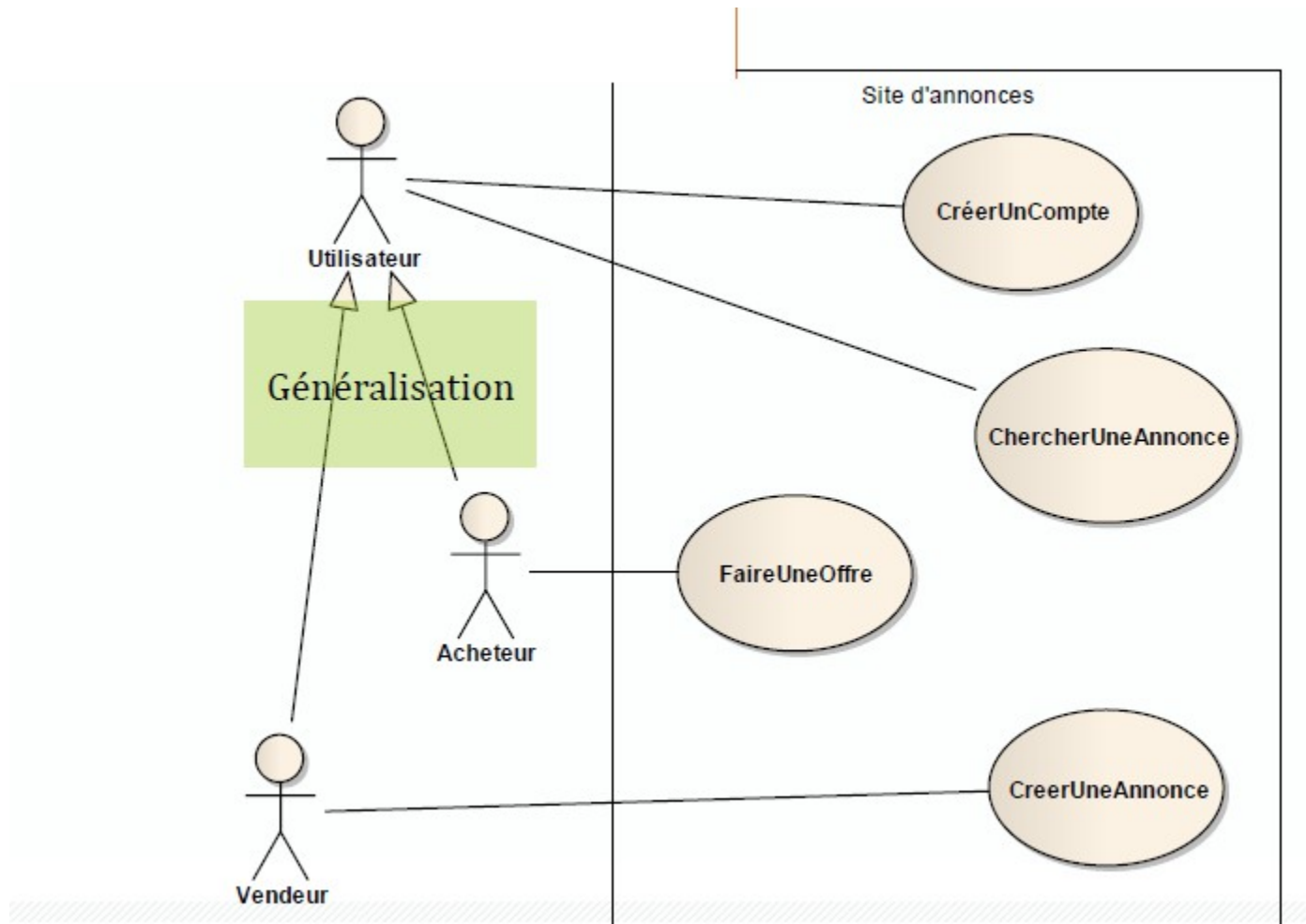
7. Exemple



8. Relation entre acteurs

- La seule relation possible entre deux acteurs est la généralisation.
- Un acteur A est une généralisation d'un acteur B si l'acteur A peut être substitué par l'acteur B. Dans ce cas, tous les cas d'utilisation accessibles à A le sont aussi à B, mais l'inverse n'est pas vrai.

8. Relation entre acteurs



Travail a faire

- Distributeur de banque
 - ❑ Identifier les acteurs,
 - ❑ Identifier les besoins fonctionnelles/ non fonctionnelles
 - ❑ Identifier le diagramme de cas d'utilisation

Travail a faire

□ Spécifications fonctionnelles :

- 1.Le système GAB (distributeur) de biller doit vérifier la validité de la carte CIB insérée
- 2.Le système GAB doit valider le code PIN entré par l'utilisateur
- 3.Le système doit allouer une somme maximum de 20000 dinars à l'utilisateur

□ Spécifications non fonctionnelles :

- 1.Le système du GAB doit être écrit en C++

Fiche descriptive

1. Description des cas d'utilisation

- Décrire un cas d'utilisation consiste à définir son contexte, et à détailler la communication entre le cas et l'acteur .
- On réalise alors une **fiche descriptive** pour chaque cas d'utilisation, de façon à raconter **l'histoire** du déroulement des différentes actions.

1.Description des cas d'utilisation

- La description d'un cas d'utilisation permet de :
 - ❑ Clarifier le déroulement de la fonctionnalité
 - ❑ Décrire la chronologie des actions qui devront être réalisées ;
 - ❑ D'identifier les parties redondantes pour en déduire des cas d'utilisation plus précises qui seront utilisées par inclusion, extension ou généralisation/spécialisation.

1.Description des cas d'utilisation

- Décrire le déroulement des actions pour un cas d'utilisation peut vous paraître simple.
- mais c'est un travail qui demande beaucoup de réflexion et de questionnement. On commence souvent par une première description, basée sur les informations que l'on a obtenu auprès du client (futurs utilisateurs).
- Lors de la réalisation de cette première description, on découvre souvent des questions auxquels nous n'avons pas de réponse.

1. Description des cas d'utilisation

- Exemple Boutique en ligne
- Si une personne n'est pas encore « cliente » sur la boutique en ligne et qu'elle souhaite réaliser un achat :
 - Doit-elle s'inscrire avant de commencer la procédure d'achat ?
 - Peut-elle s'inscrire pendant la procédure d'achat, par exemple juste avant de régler ?

1.Description des cas d'utilisation

- Cette fiche descriptive doit comporter 4 volets :
 - L'identification
 - La description des scénarios
 - La fin et les post-conditions
 - Les compléments

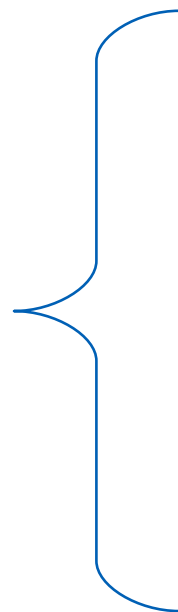
2. Modèle fiche descriptive 1/

Identification	Numéro	
	Nom	
	Acteurs (primaire /secondaire)	
	Objectif	
	Pré-condition	
	Date de création	
	Auteur	

Les pré-conditions : il s'agit des conditions obligatoires au bon déroulement du cas d'utilisation. Par exemple, dans le guiche de banque il fallait obligatoirement s'authentifier avant le cas « retirer l'argent ». Dans ce cas, l'« Authentification » est une pré-condition du cas « retirer argent » ;

2. Modèle fiche descriptive 2/

description des scénarios



Le scénario nominal	décrire le déroulement idéal des actions donc , Étapes à suivre pour réaliser ce cas
Les scénarios alternatifs / exception	Déviations des étapes du scénario principal -scénarios alternatifs, scénarios d'erreur-

2. Modèle fiche descriptive 3/

	Post-condition	nous indiquent un résultat tangible qui est vérifiable après l'arrêt du cas d'utilisation et qui pourra témoigner du bon fonctionnement. Cela pourrait être une information qui a été enregistrée dans une base de données ou dans un fichier, ou encore un message envoyé par mail
--	----------------	---

	Compléments	<ul style="list-style-type: none">▪ l'ergonomie ;▪ la performance attendue ;▪ des contraintes (techniques ou non) à respecter ;▪ des problèmes non résolus (ou questions à poser au client et aux futurs utilisateurs).
--	-------------	--

4. Exemple

- Fiche descriptive : cas d'utilisation « Authentification » des clients dans le guiche de banque

4.Exemple

Nom	Authentification
Acteur	Client de la banque
Objectif	- L'utilisateur souhaite se connecter au système de la banque
Près-condition	- l'utilisateur possède un login et mot de passe
Le scénario nominal	1-l'utilisateur accède au système 2-Le système affiche la page d'authentification 3-l'utilisateur remplis les champs' login & mot de passe' 4-le système affiche le menu
Scénario alternatif	3.a- mot de passe et/ou login incorrect -aller a 2- 3.a.1 . Le système affiche un message d 'erreur
Exception	L'utilisateur annule l'opération
Poste-condition	Accès au système

Diagramme d'interaction

1. Diagrammes d'interaction

- Les diagrammes d'interaction permettent d'établir un lien entre les diagrammes de cas d'utilisation et les diagrammes de classes : ils montrent comment des objets (*i.e.* des instances de classes) communiquent pour réaliser une certaine fonctionnalité. Ils apportent ainsi un aspect dynamique à la modélisation du système.

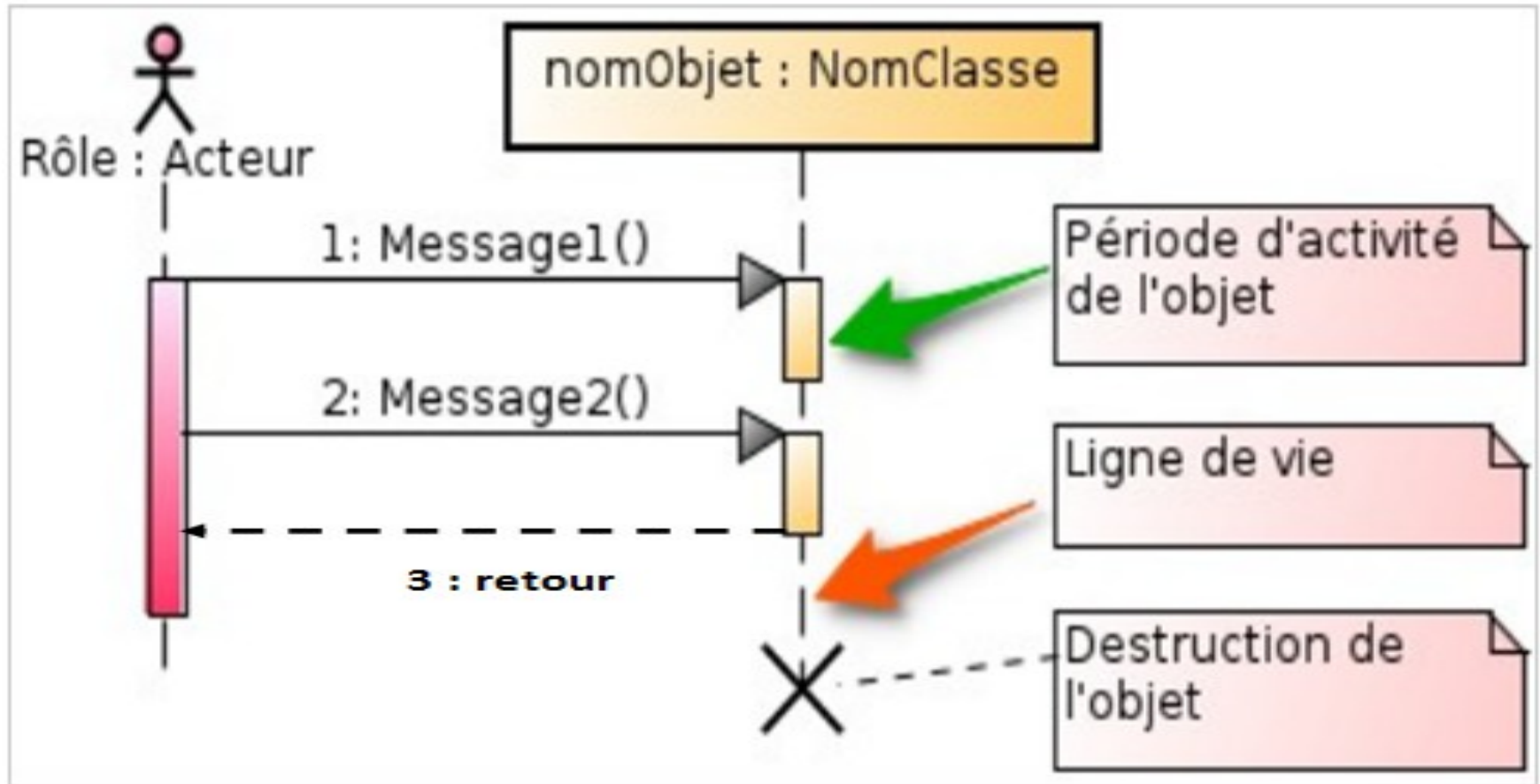
1. Diagrammes d'interaction

- Diagramme d'états-transitions
- Diagramme d'interaction
- Diagramme de séquence
- Diagramme de communication

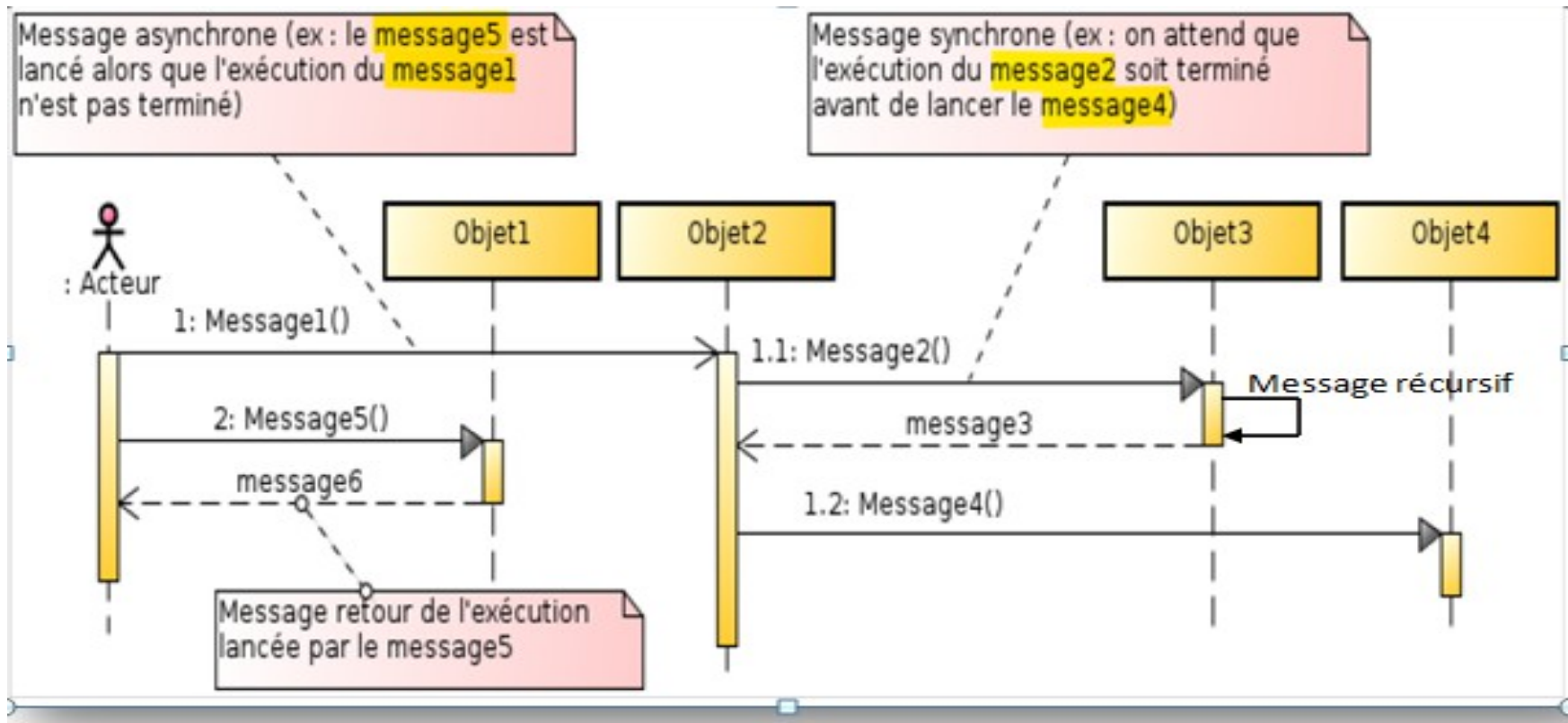
2. Diagramme de séquence

- Les **diagrammes de séquences** sont la **représentation** graphique des interactions entre les acteurs et le système selon un ordre chronologique.
- Description des interactions dans des termes proches de l'utilisateur

3. Eléments du Diagramme de Séquence

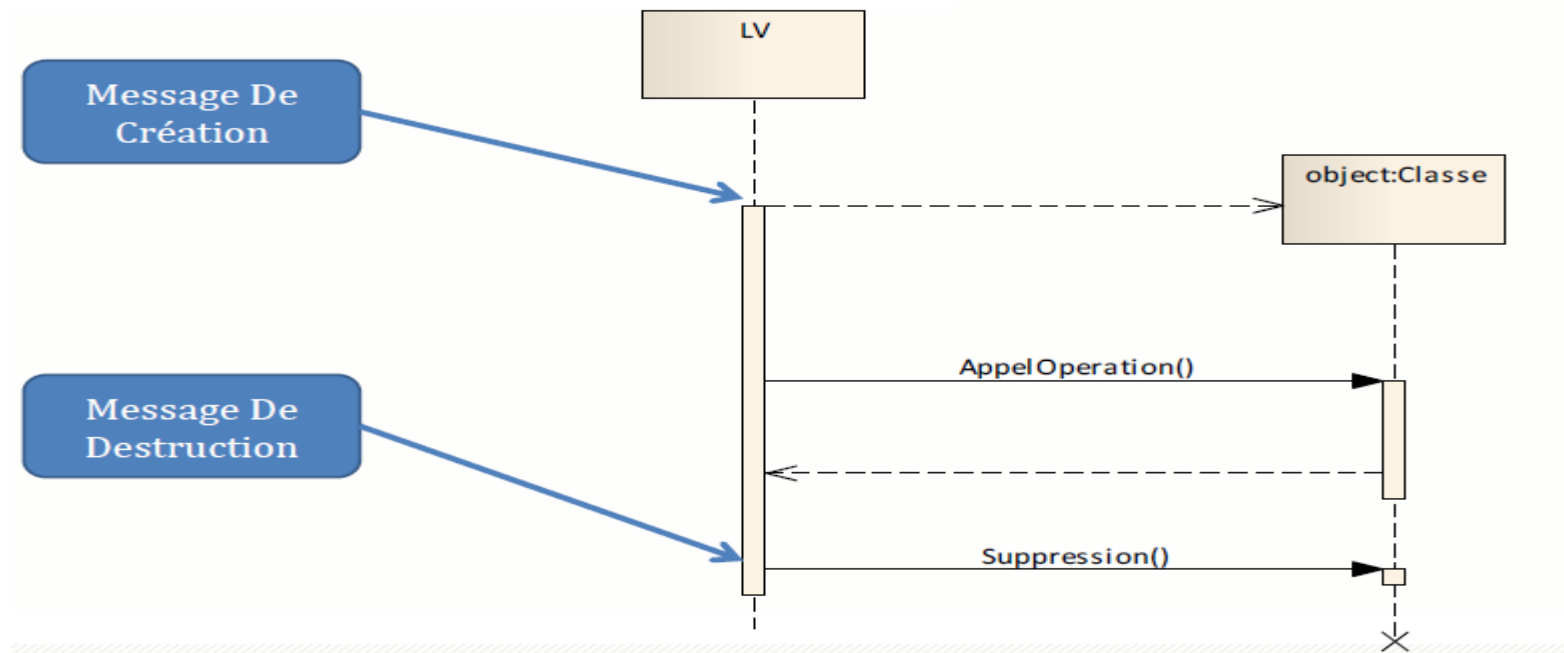


3. Eléments du Diagramme de Séquence

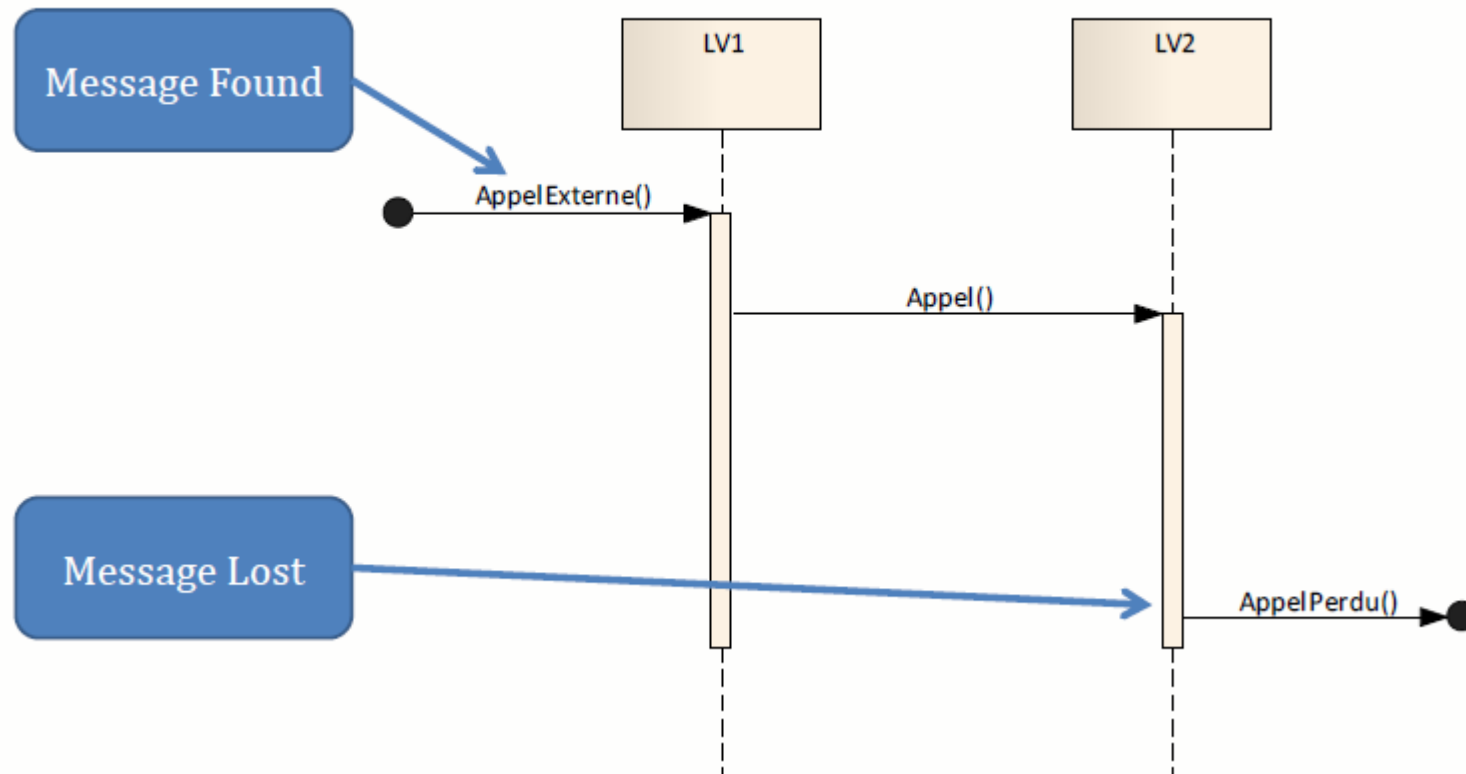


- ❑ Message synchrone : Un message est envoyé par à un objet à un autre, et le premier objet attends jusqu'à ce que l'action ai finie.
- ❑ Message asynchrone : Un message est envoyé par à un objet à un autre, mais le premier objet n'attends pas la fin de l'action.

3. Eléments du Diagramme de Séquence



3. Eléments du Diagramme de Séquence



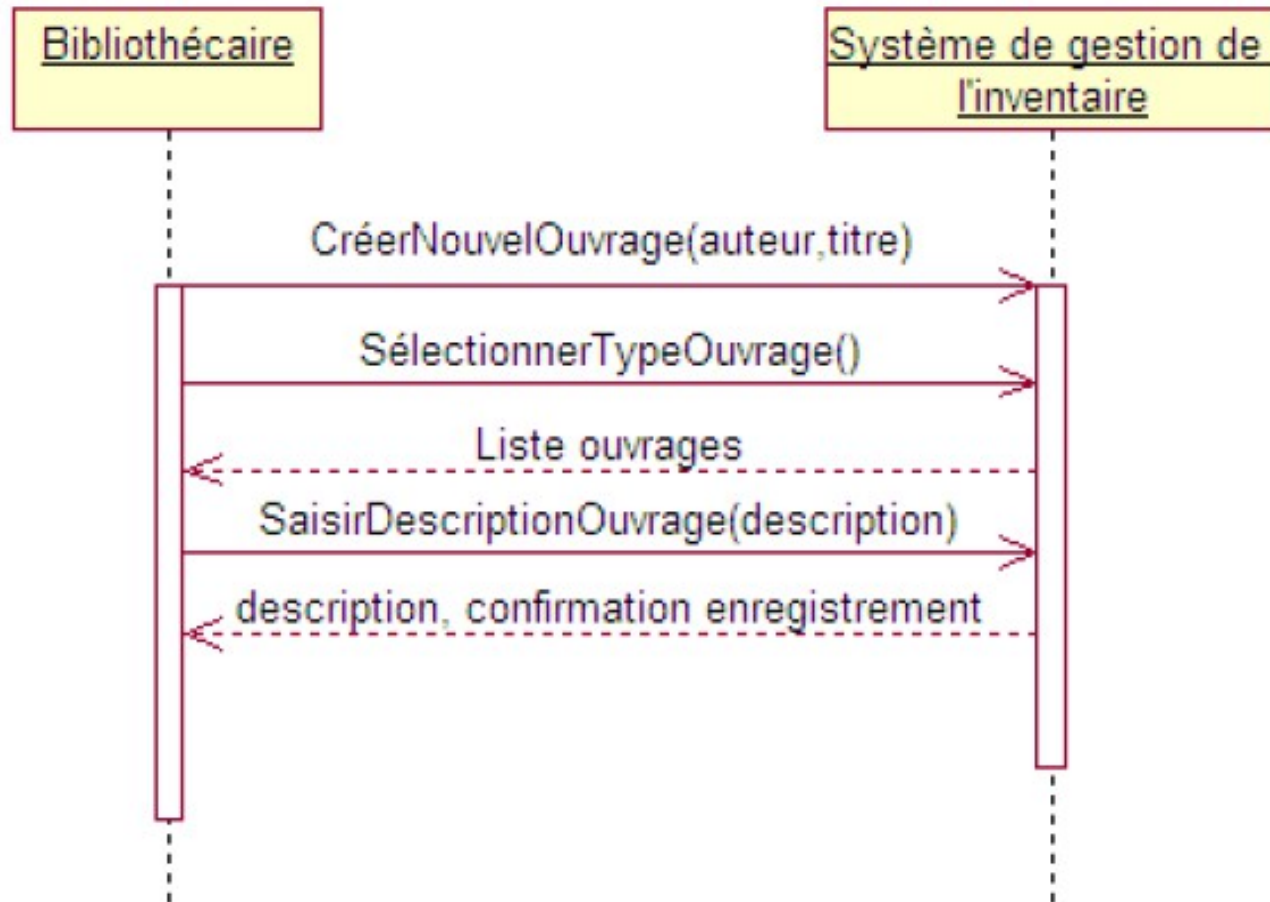
3. Diagramme de séquence système

- **Objectif:** Décrire le comportement du système par des diagrammes de séquence système (DSS) où le système est vu comme une « boîte noire ».
- **Principe** Le système est donc vu de l'extérieur(par les acteurs) sans préjuger de comment il sera réalisé. La « boîte noire » sera ouverte (décrite) seulement en phase de conception. Le fonctionnement d'un cas d'utilisation est notamment décrit sous la forme d'une séquence de messages échangés entre les acteurs et le système.

4. Exemple Diagramme de séquence système

- Le bibliothécaire souhaite enregistrer un nouvel ouvrage dans le système de gestion de l'inventaire de la bibliothèque. Pour cela, il fournit au système le nom de l'auteur et le titre de l'ouvrage.
- Le bibliothécaire sélectionne ensuite le type d'ouvrage à enregistrer. Le système lui retourne la liste des ouvrages de ce type qui ont été écrit par cet auteur et qui sont déjà enregistrés dans le système.
- Le bibliothécaire insère une description complète de l'ouvrage.
- Le système retourne l'ensemble de la description sous un format de lecture et confirme l'enregistrement du nouvel ouvrage.

4. Exemple



3. fragments combinés

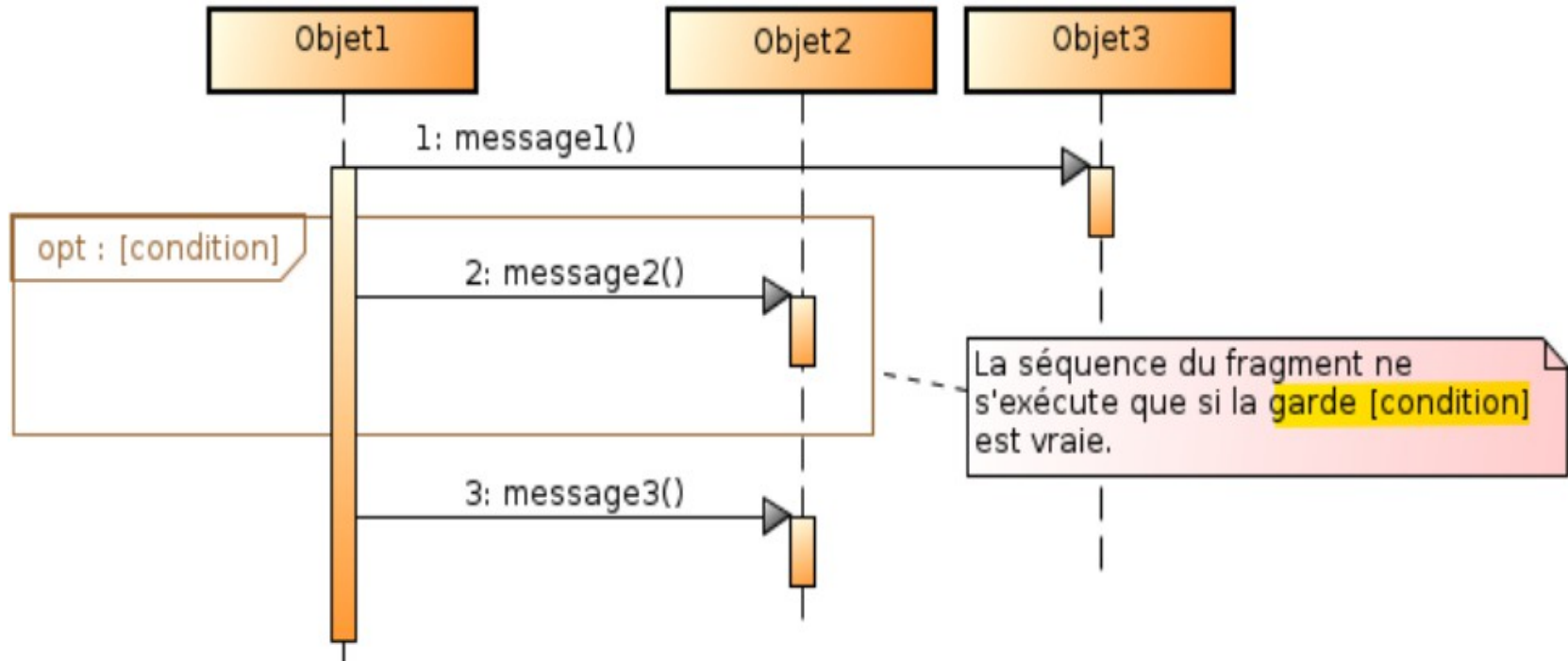
- Un fragment d'interactions est une partie du diagramme de séquence (délimitée par un rectangle) associée à une étiquette (dans le coin supérieur gauche). L'étiquette contient un opérateur d'interaction qui permet de décrire des modalités d'exécution des messages à l'intérieur du cadre.
- Il existe **10 opérateurs** définis dans la notation UML2.0

3. fragments combinés

- La liste suivante regroupe les opérateurs d'interaction par fonctions :
 - les opérateurs de choix et de boucle : *alternative*, *option*, *break* et *loop* ;
 - les opérateurs contrôlant l'envoi en parallèle de messages : **parallel** et **critical region** ;
 - les opérateurs contrôlant l'envoi de messages : **ignore**, **consider**, **assertion** et **negative** ;
 - les opérateurs fixant l'ordre d'envoi des messages : **weak sequencing** « seq », **strict sequencing**. « strict »

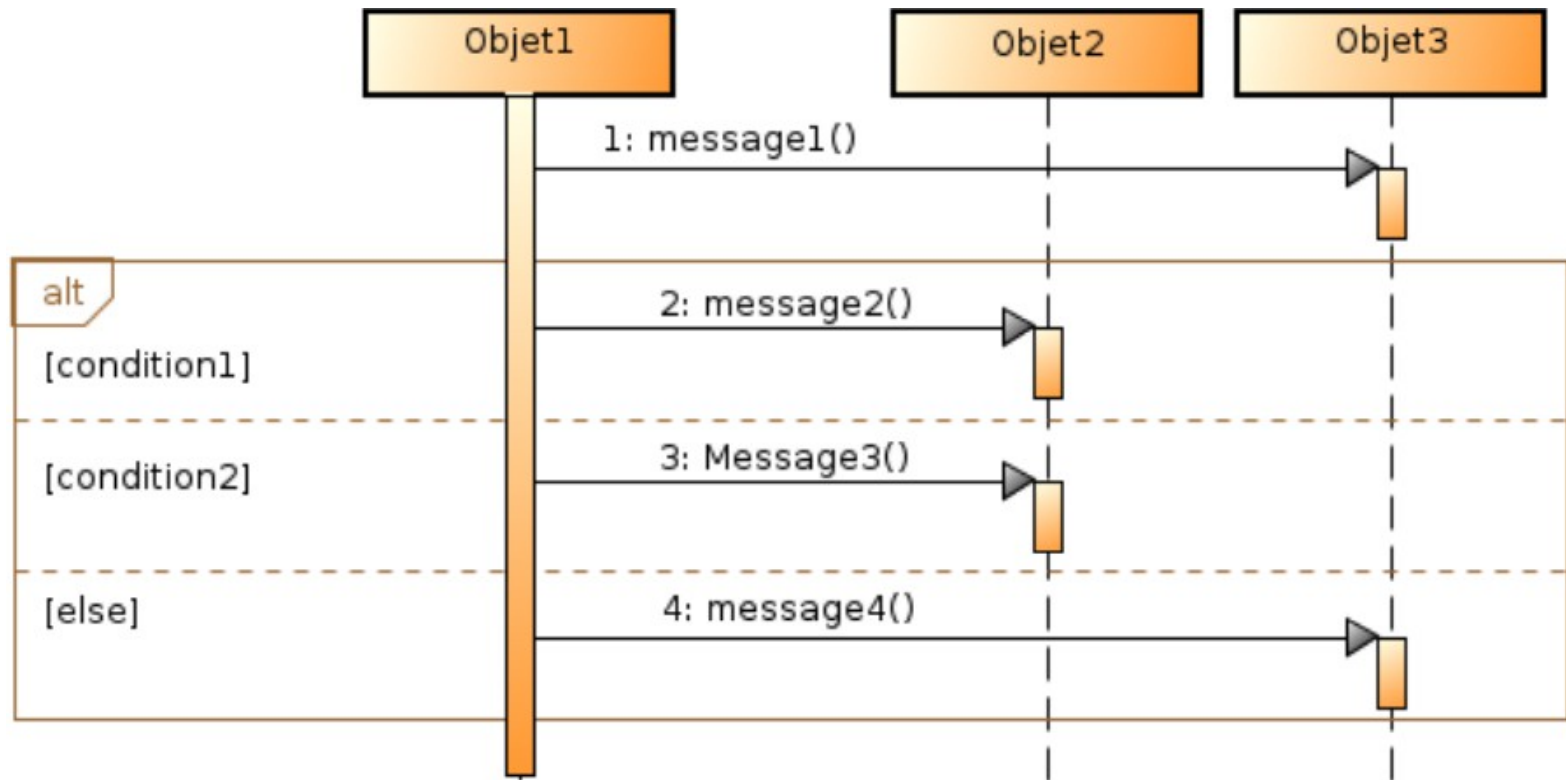
3. fragments combinés

3.1 opérateur «OPT»



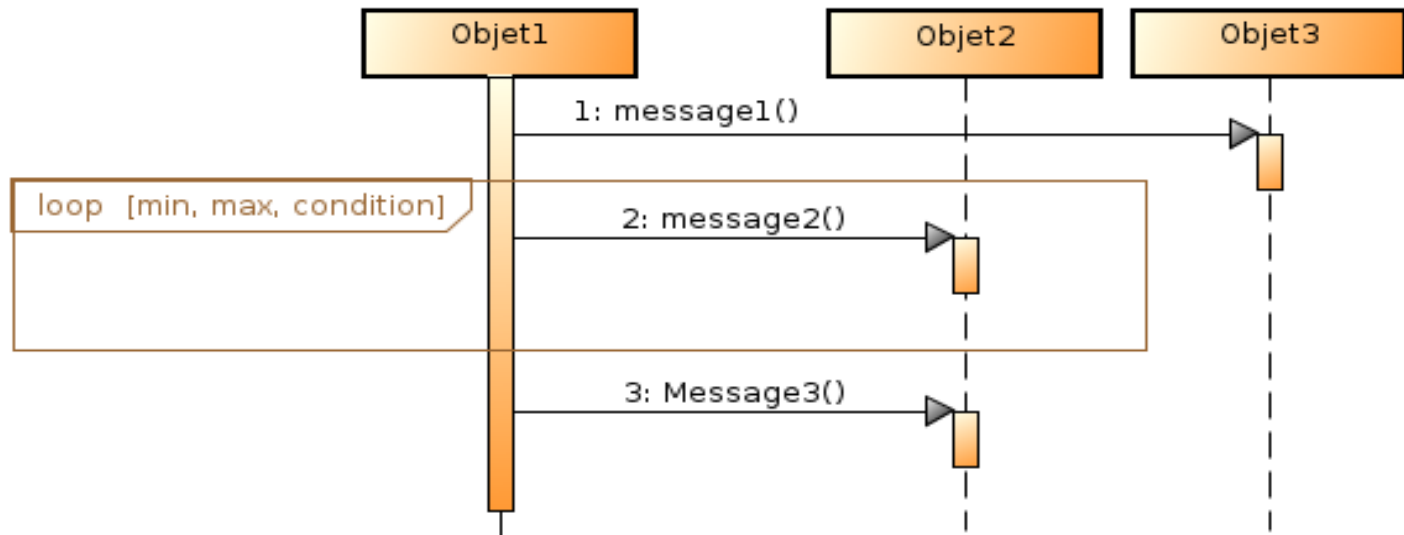
3. fragments combinés

3.2 opérateur « ALT » Sialors...sinon



3. fragments combinés

3.3 opérateur « loop » Boucle

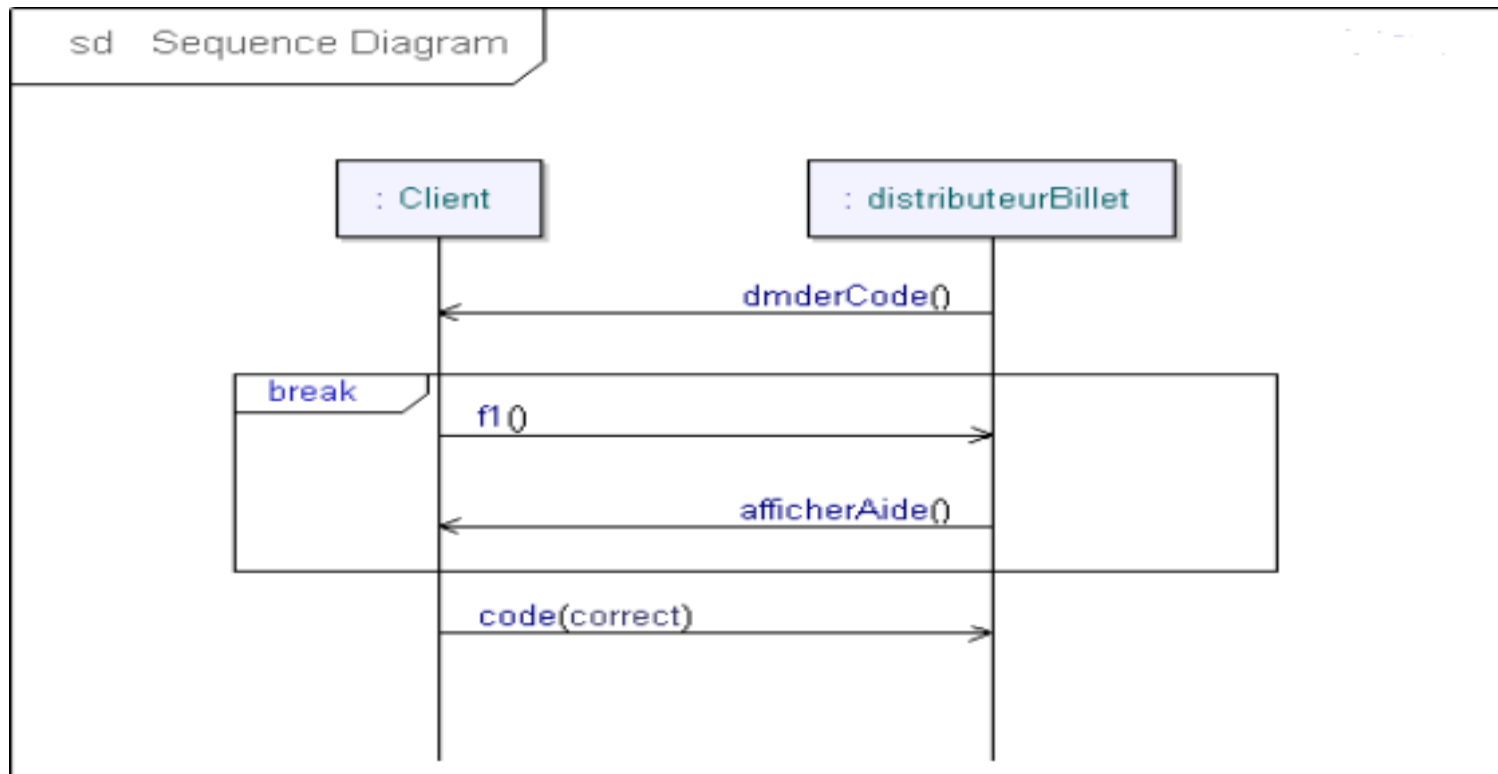


loop [min, max, condition] : Chaque paramètre (min, max et condition) est optionnel.

Le contenu du cadre est exécuté min fois, puis continue à s'exécuter tant que la condition et que le nombre d'exécution de la boucle ne dépasse pas max fois.

3. fragments combinés

3.6 opérateur « Break »

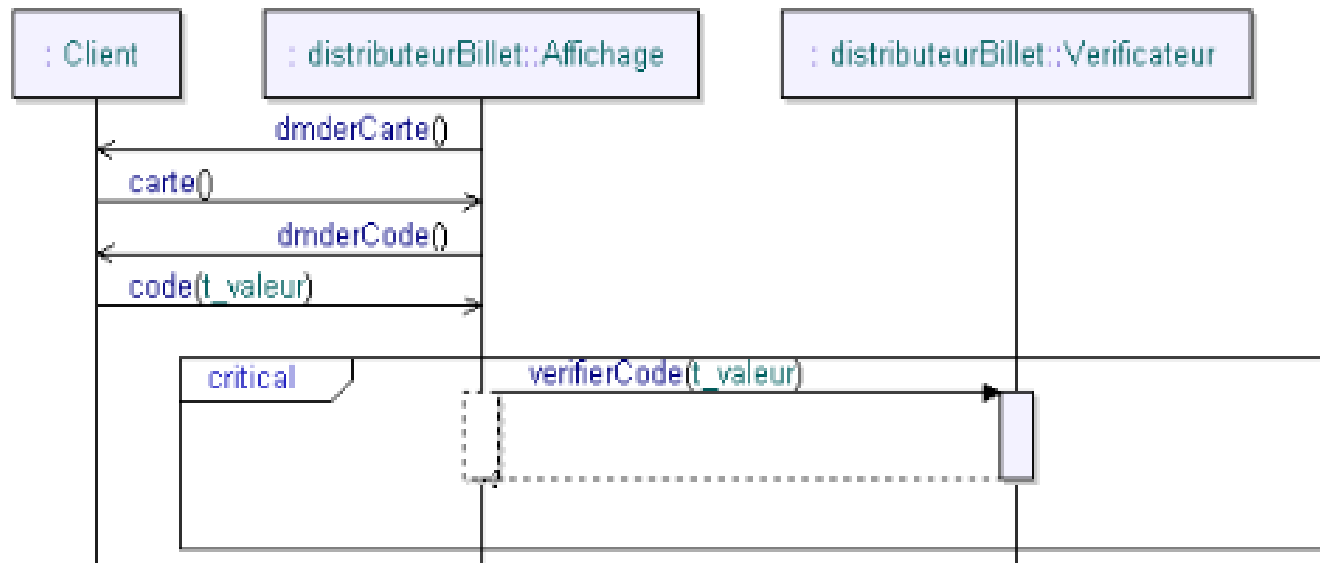


L'opérateur "**break**" est utilisé pour représenter des scenario d'**exception** . Les interactions de ce fragment seront exécutées à la place des interactions décrites en dessous. Il y a donc une notion d'interruption du flot "normal" des interactions.

3.fragments combinés

3.7 opérateur «critical»

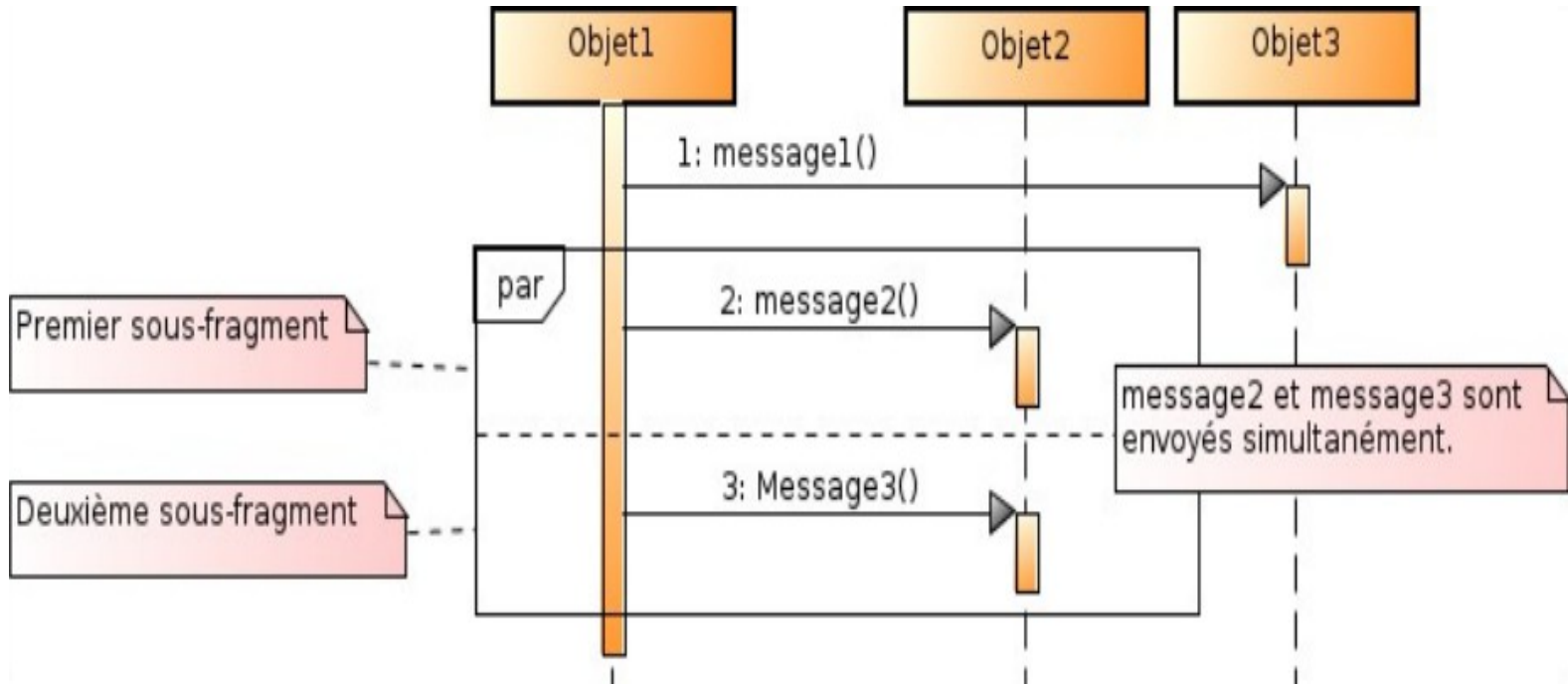
sd RetirerArgent_critical



L'opérateur "**critical**" désigne une **section critique**. Une section critique permet d'indiquer que les interactions décrites dans cet opérateur ne peuvent pas être interrompues par d'autres interactions décrites dans le diagramme. On dit que l'opérateur impose un traitement atomique des interactions qu'il contient.

3. fragments combinés

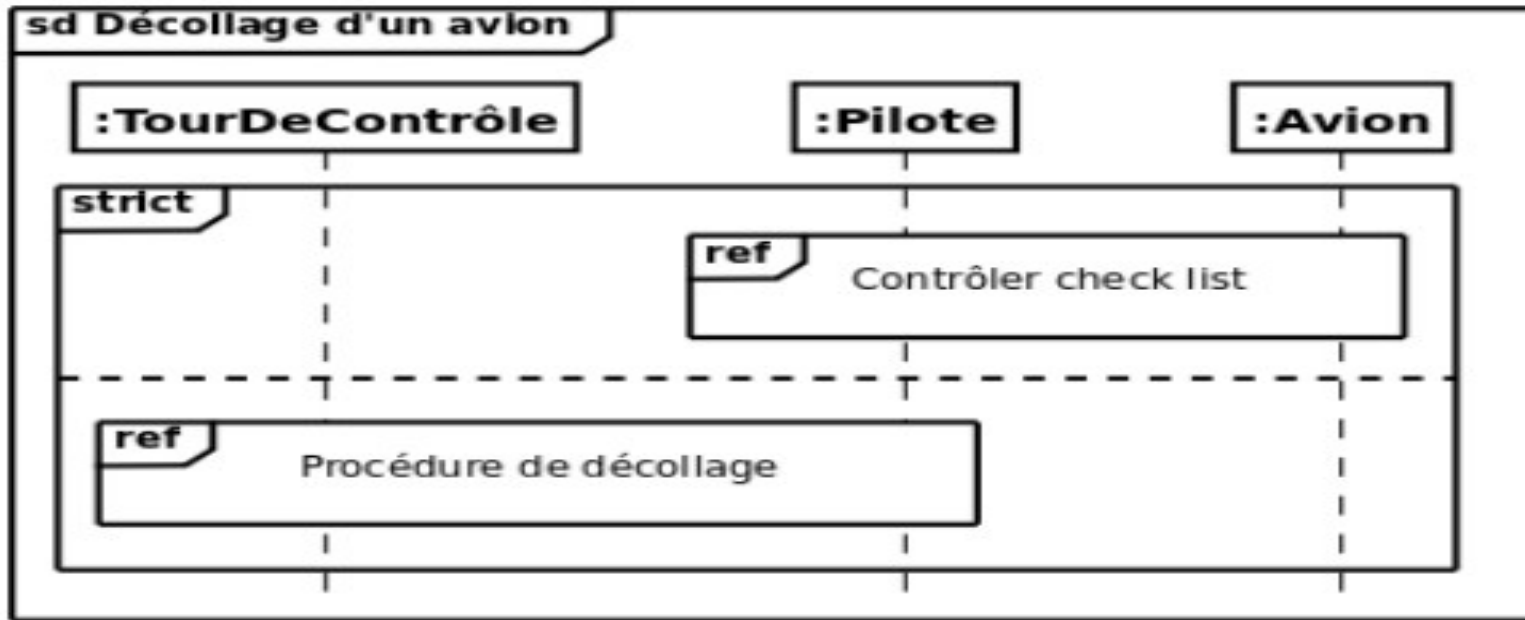
3.4 opérateur « **Par** » parallèles



Un fragment d'interaction avec l'opérateur de traitements parallèles (par) contient au moins deux sous fragments (opérandes) séparés par des pointillés qui s'exécutent simultanément (traitements concurrents).

fragments combinés

3.5 opérateur « strict »



- Un fragment combiné de type *strict sequencing*, ou *strict*, possède au moins deux sous-fragments. Ceux-ci s'exécutent selon leur ordre d'apparition au sein du fragment combiné.
- Opérateur « **Ref** » permet de faire appel à un autre diagramme de séquence.

sd identifierUtilisateur

interaction identifierUtilisateur {1/3}

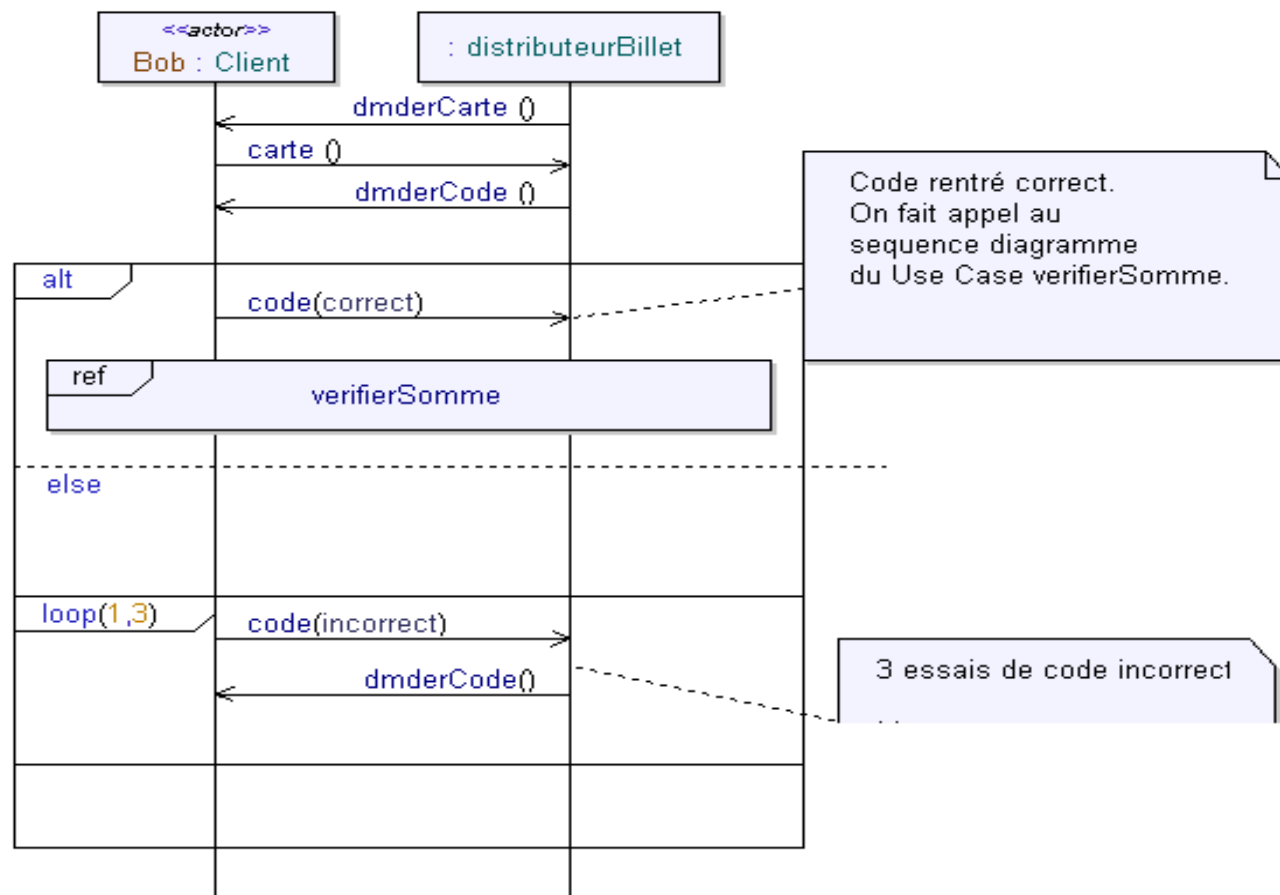


Diagramme de communication

1. Définition

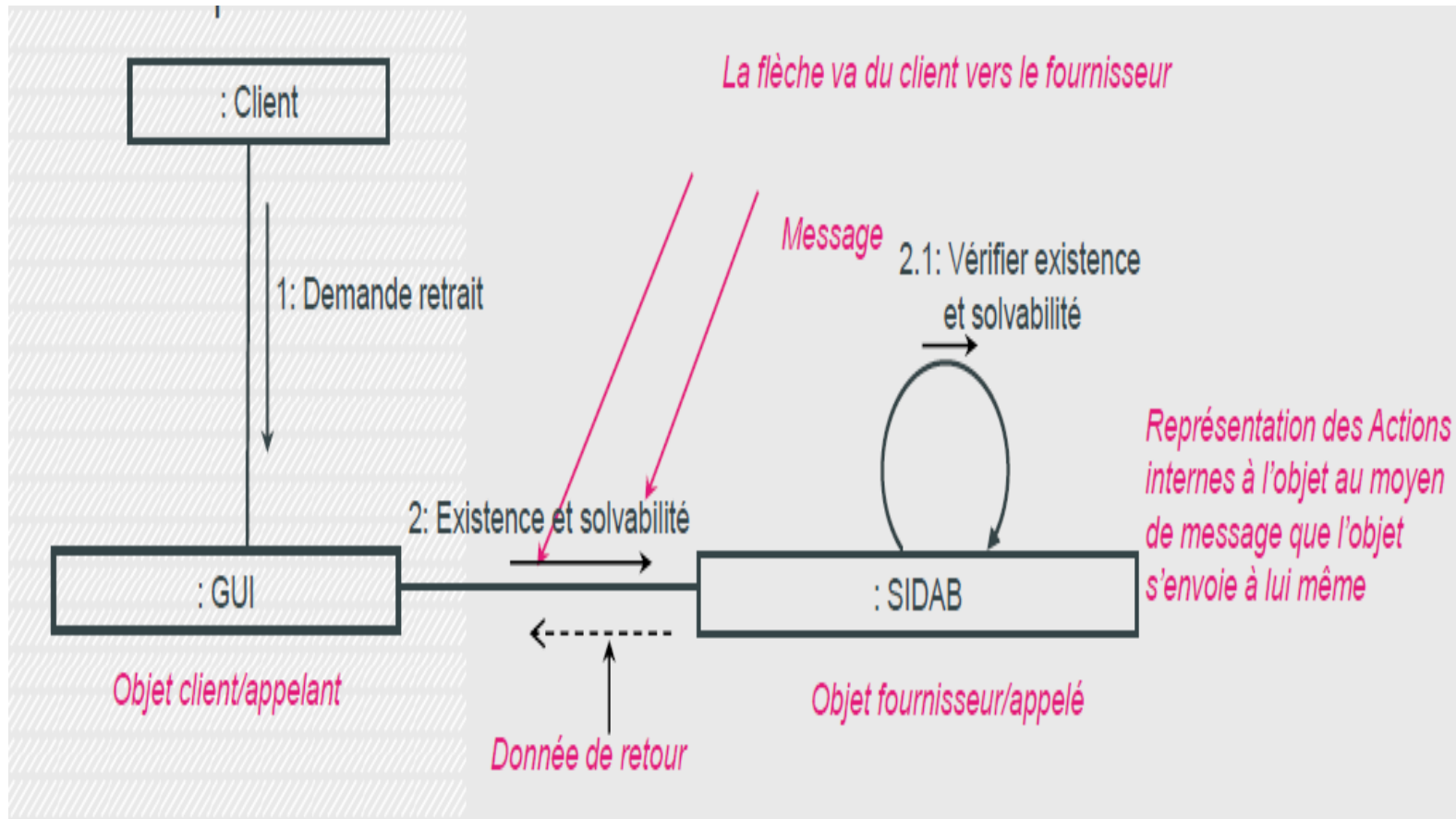
- En UML 1.x s'appelle **diagramme de collaboration**.
- A été renommé **diagramme de communication** en UML 2.0.

2. Formalisme

- **3 CONCEPTS IMPORTANTS :**

- Les Objets: Représenté par un rectangle
- Les liens d'interactions : Indique un chemin de communication entre 2 objets, sur lequel passent les messages
- Les Messages : synchrone, asynchrone, aller et retour

3.Exemple



3.Exemple

