

*Dans la cartographie des diagrammes UML 2,
Vous avez étudié ...*

Dans une première partie :

- ♦ **Les diagrammes de cas d'utilisation** modélisant à **QUOI** sert le système, en organisant les interactions possibles avec les acteurs ;

Dans une seconde partie, vous étudierez quelques-uns des diagrammes d'interaction permettant de :

- ♦ Modéliser **comment** les objets communiquent entre eux (point focal : échange de messages) ;
- ♦ Offrir une vue plus holistique du comportement d'un jeu d'objets.

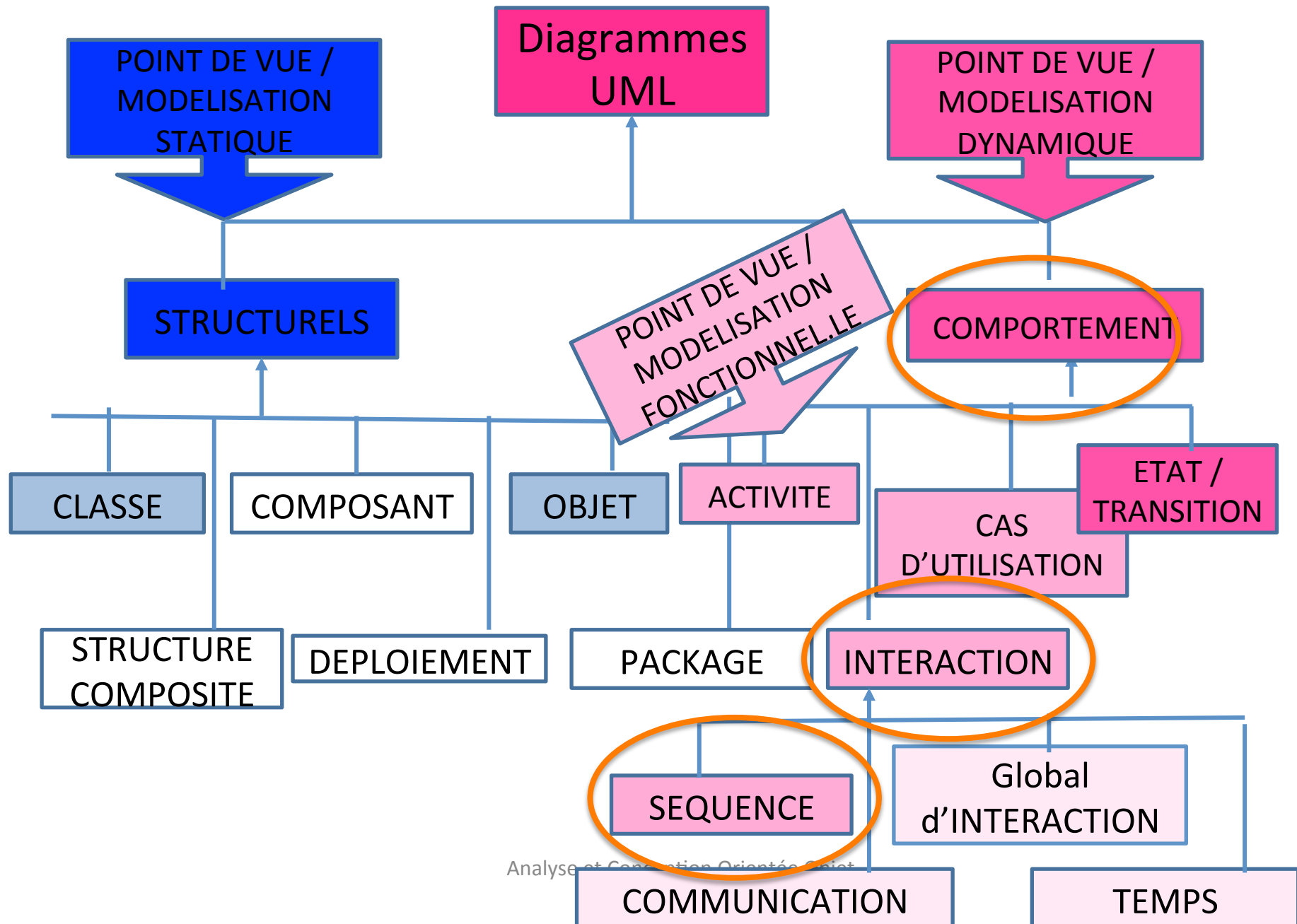
Diagrammes d'Interactions

La norme 2.0 reclasse les diagrammes de la norme 1.X et met les 4 diagrammes suivants dans la catégorie des diagrammes d'interaction :

- ◆ **Diagrammes de séquence**
- ◆ **Diagramme de communication**
- ◆ Diagramme d'interaction globale
- ◆ Diagrammes de temps

Ces types de diagramme tendent à mettre l'accent spécifiquement sur la **séquence** des opérations plutôt que sur les données qui sont véhiculées, bien que certains ajouts (décorations) permettent de le faire.

Rappel - Cartographie des Diagrammes UML 2



Description dynamique

Diagramme de Séquence (DSE) :

Le DSE est une forme de diagramme comportemental, dérivé des scénarios de OMT

L'**objectif** du DSE est de représenter les interactions entre objets en indiquant la **chronologie des échanges**.

Grâce à ces informations, vous pouvez déterminer plus précisément pourquoi deux objets sont liés et voir comment les ils s'utilisent mutuellement.

Cette représentation est réalisée par cas d'utilisation en considérant :

- Forme générique : décrivant **tous** les déroulements possibles d'un scénario et contenant des **branchements**, des **conditions**, et des **boucles**;
- Forme d'instanciation : décrivant **un aspect spécifique** d'un scénario et ne contenant aucun **branchements**, et aucune **condition** ou **boucle**.

En général, un DSE capture le comportement d'un seul scénario.

Description dynamique

Autrement dit :

- Les diagrammes de séquences permettent de décrire **COMMENT** les éléments du système interagissent entre eux et avec les acteurs :
 - ◆ Les objets au coeur d'un système interagissent en s'échangeant des messages ;
 - ◆ Les acteurs interagissent avec le système au moyen d'IHM (Interfaces Homme-Machine).

Description dynamique

Diagramme de Séquence (DSE):

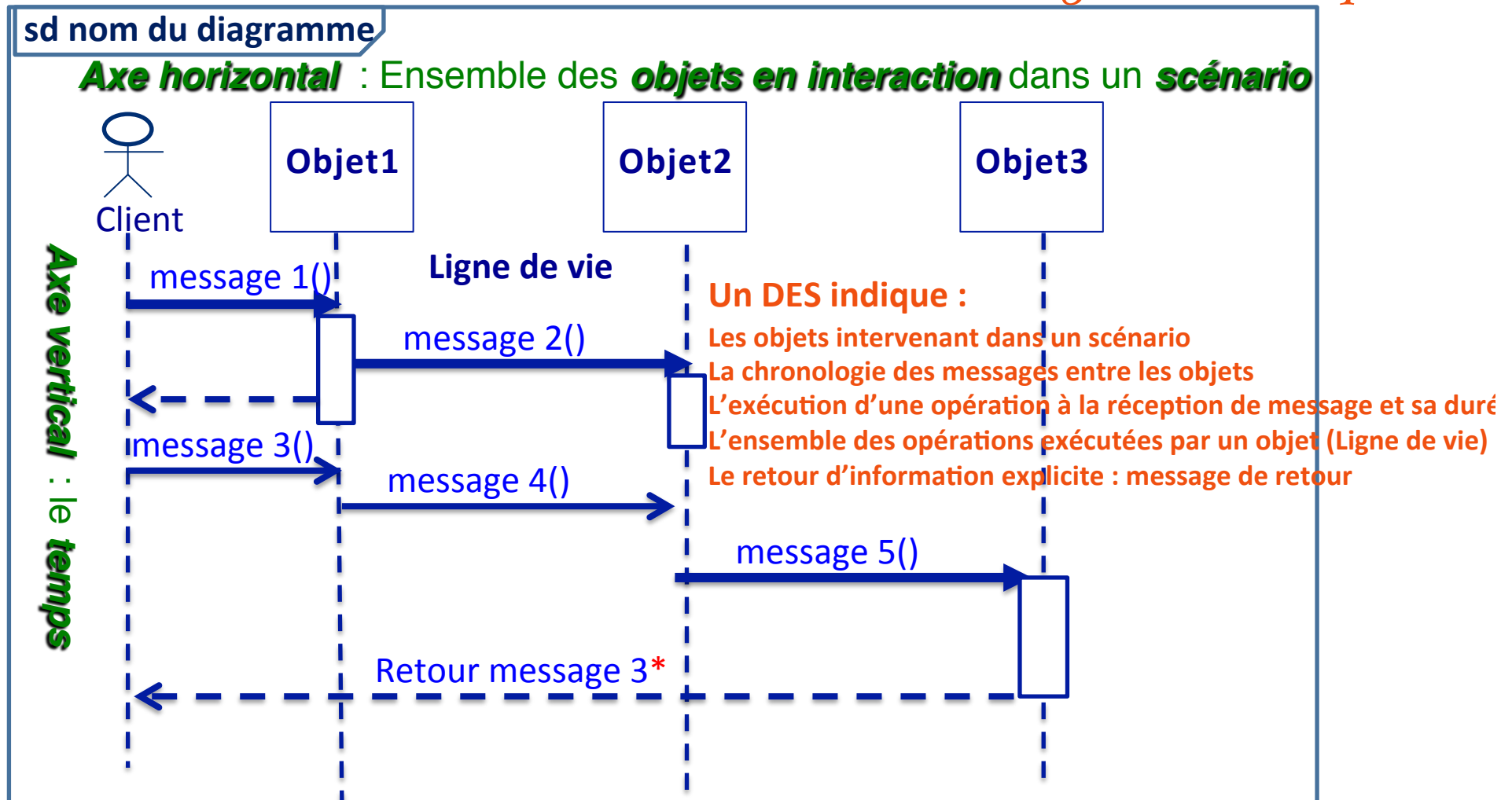
Formalisme général du cadre d'un diagramme de séquence :



sd : abréviation de « sequence diagram »

Description dynamique

Formalisme du diagramme de séquence



** Les messages asynchrones pouvant être reçus dans un ordre différent de l'ordre d'envoi*

Description dynamique *Diagramme de Séquence (DSE):*

Pour illustrer les concepts de base et les opérations d'un DSE, prenons l'exemple d'un scenario de traitement d'une commande, spécifiant :

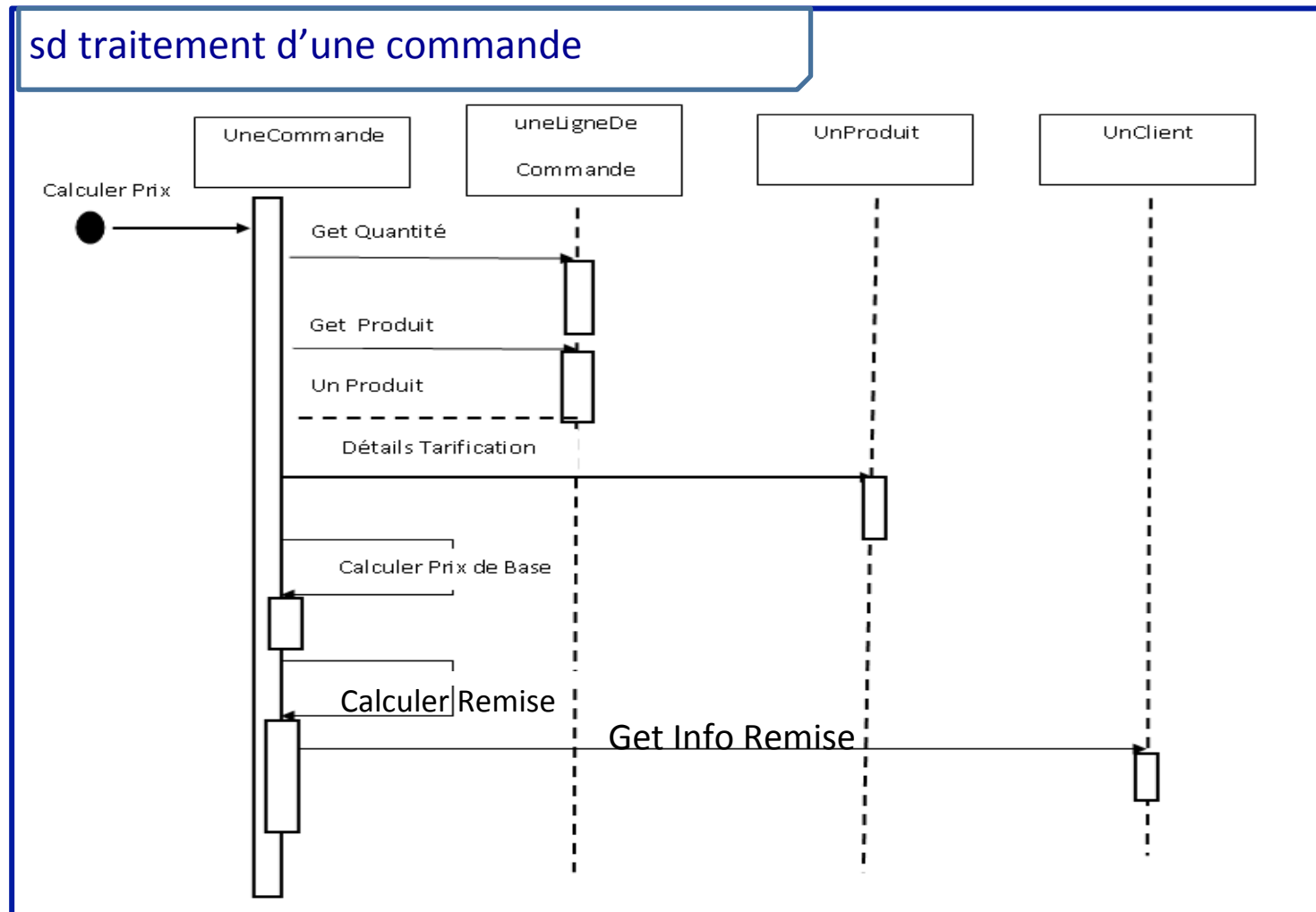
Pour calculer le prix d'une commande, l'objet commande doit :

- ✓ Lire toutes les lignes de la commande
- ✓ Déterminer leur prix, en fonction des tarifications des produits.

Après avoir traité chaque ligne, il doit ensuite :

- ✓ Calculer une remise globale, qui dépend des règles associées à chaque client.

Implémentation du scenario exemple par un DSE

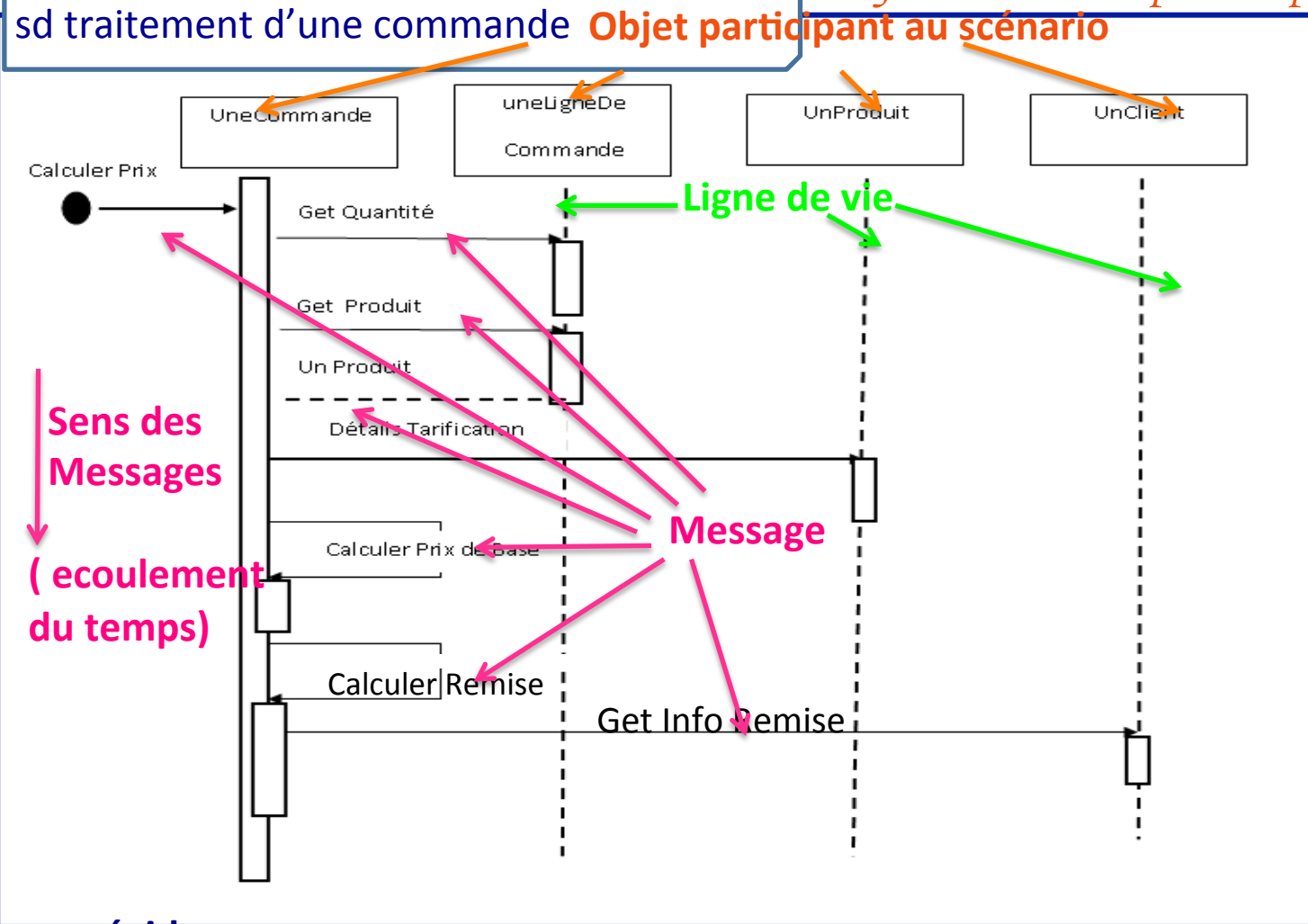


Le DES met bien en évidence :

- L'interaction d'un objet participant au scénario
- Chaque participant ayant une ligne de de vie, qui parcourt verticalement la pag
- L'ordre des messages se faisant de haut en bas.

Implémentation du scénario exemple par un DSE

Identification des composants primaires

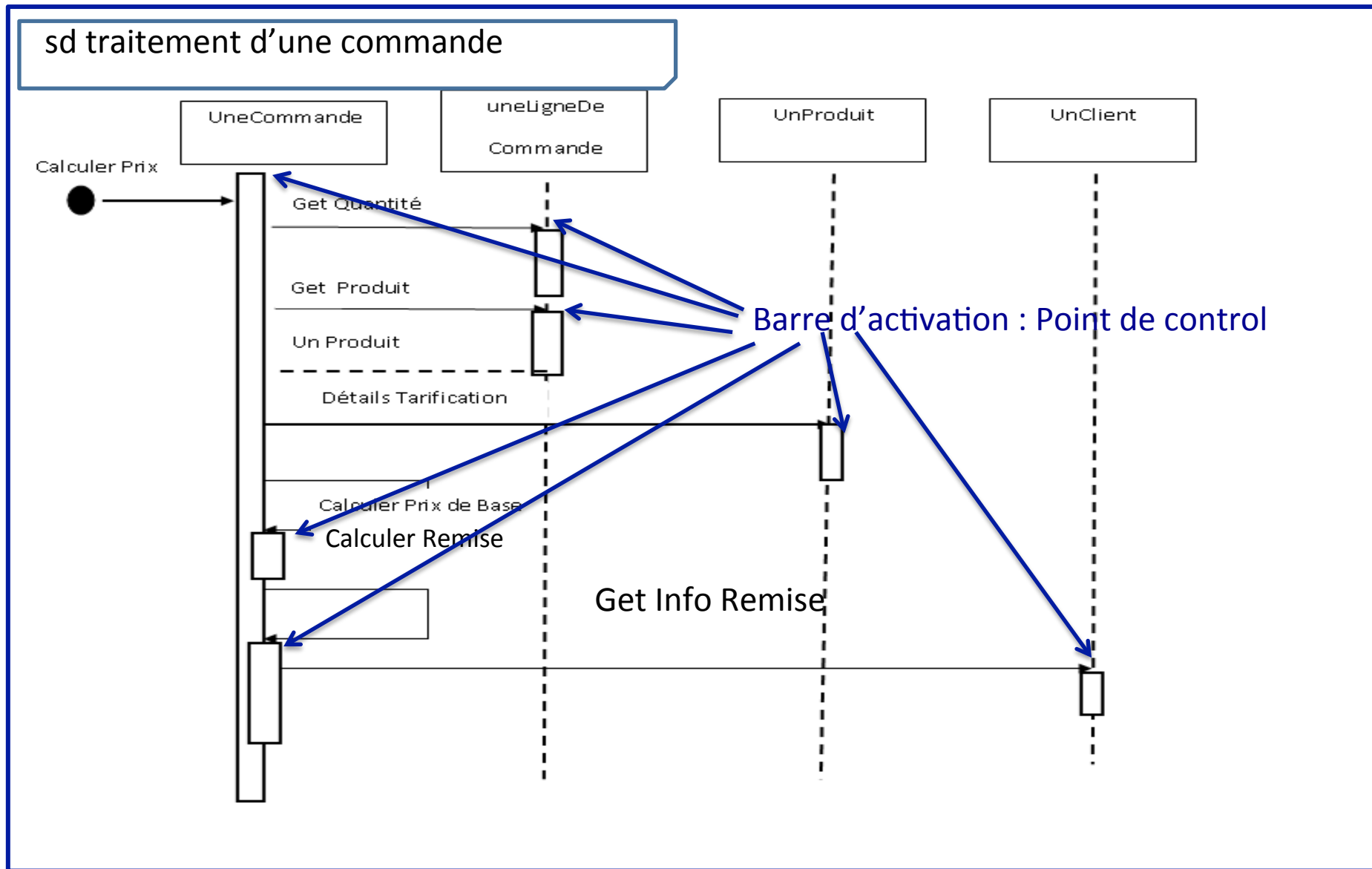


Le DES met bien en évidence :

- **L'interaction d'un objet participant au scénario**
- **Chaque participant ayant une ligne de vie, qui parcourt verticalement la page**
- **L'ordre des messages se faisant de haut en bas.**

Implémentation du scenario exemple par un DSE

Composants optionnels



Description dynamique

Sémantique des composants d'un DSE

Définition des composants primaires d'un DSE

➤ **1. Les objets (Object)** : participants à l'interaction

Les objets apparaissent toujours dans la partie supérieure, ce qui facilite l'identification des classes (en supposant que la notation des classes est utilisée).

Représentation graphique :

UML



UML 2



Les composants primaires (suite) :

- **2. La ligne de vie de l'objet (Object lifeline) :** représentant
- ✓ la vie d'un objet dans le contexte de la séquence **d'événements**
 - ✓ La durée de l'activité du participant dans l'interaction.

Formalisme :

|
|
|
|
|

Un événement se produit généralement lors de la réception explicite d'un signal ou d'un message, lorsqu'une condition devient vraie, écoulement d'une période de temps (expression temporelle)

- ✓ Chaque ligne de vie comporte une activation : point de contrôle (durée de vie de l'activité du participant dans l'interaction)
 - Le point de contrôle correspond au temps pendant lequel l'une des méthodes du participant est au sommet de la pile
 - Les barres d'activation sont facultatives, mais elles clarifient le comportement.

Les composants primaires (suite) :

- **3. Les messages** définissent une communication particulière entre deux instances, présentes sur des lignes de vie.
 - ◆ Représentés par des flèches directionnelles véhiculant une information (message) envoyée entre les objets ;
 - ◆ Les messages anticipent qu'une action sera entreprise ;
 - ◆ Dans la plupart des cas, la réception d'un message est suivie de l'exécution d'une méthode d'une classe (classe réceptrice) ;
 - ◆ L'ordre relatif des messages est matérialisé par l'axe vertical qui représente l'écoulement du temps ;
 - ◆ Le retour de message doit être matérialisé, lorsqu'il existe.

Plusieurs types de messages existent, les plus communs sont :

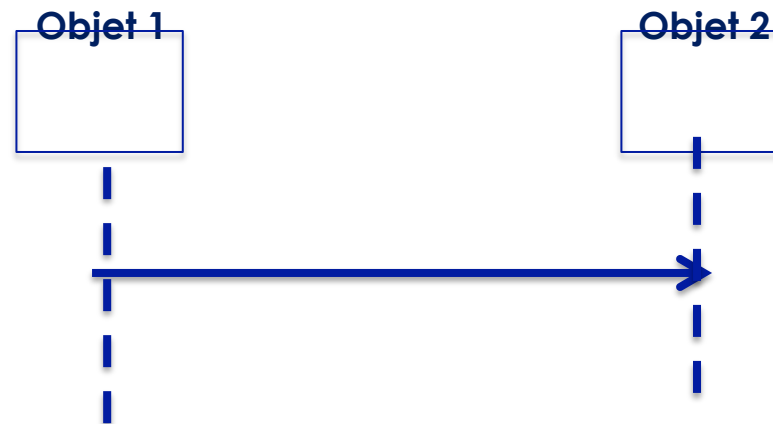
- ◆ L'envoi d'un signal ou d'un événement ;
- ◆ L'invocation d'une opération ;
- ◆ La création ou la destruction d'une instance.

Description dynamique

Sémantique des composants d'un DSE

➤ **3.1 Message asynchrone :**

- ✓ L'émetteur n'attend pas la réponse à son message, il poursuit l'exécution de ses opérations.

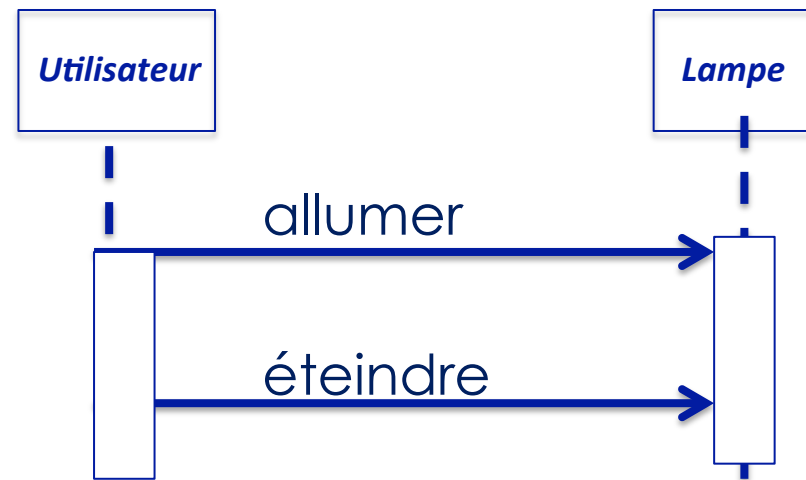


- ◆ Une interruption ou un événement sont de bons exemples de signaux.
- ◆ Ils n'attendent pas de réponse et ne bloquent pas l'émetteur qui ne sait pas si le message arrivera à destination, le cas échéant quand il arrivera et s'il sera traité par le destinataire.
- ◆ Lors de l'envoi de message asynchrone, le retour doit être représenté s'il existe.

Description dynamique
Sémantique des composants d'un DSE

➤ **3.1 Exemple de message asynchrone :**

- ✓ Représentation de l'allumage et de l'extinction de la lampe par un utilisateur



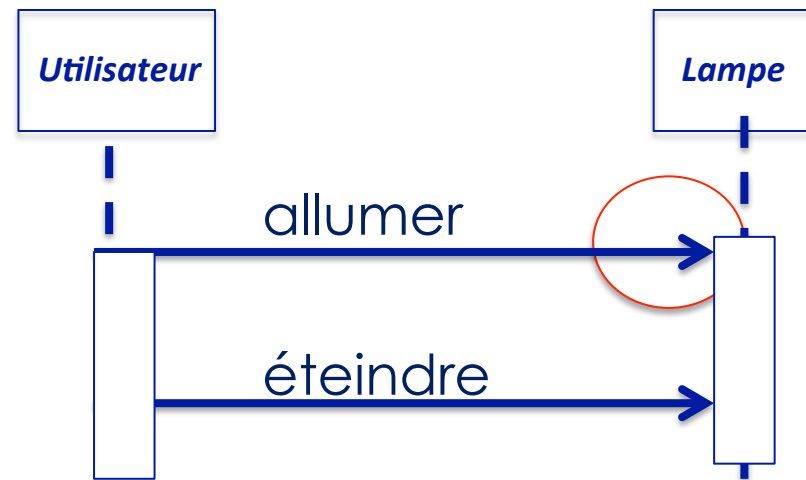
- ◆ Ces deux messages sont asynchrones :
 - L'utilisateur n'attend pas un message de retour de la lampe lui indiquant si elle est allumée (ou éteinte dans le second cas).

Description dynamique

Sémantique des composants d'un DSE

➤ **2.1 Exemple de message asynchrone :**

- ✓ Représentation de l'allumage et de l'extinction de la lampe par un utilisateur



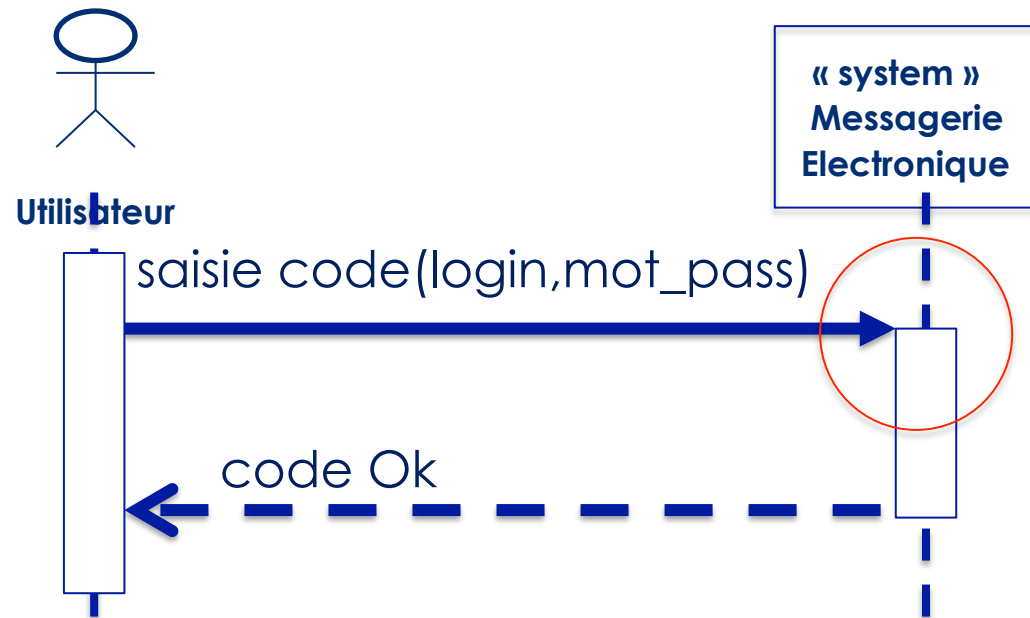
- ◆ Ces deux messages sont asynchrones :
 - L'utilisateur n'attend pas un message de retour de la lampe lui indiquant si elle est allumée (ou éteinte dans le second cas).

Description dynamique

Sémantique des composants d'un DSE

➤ 3.2 Message synchrone :

- ✓ L'émetteur reste en attente de la réponse à son message, avant de poursuivre l'exécution de ses opérations.



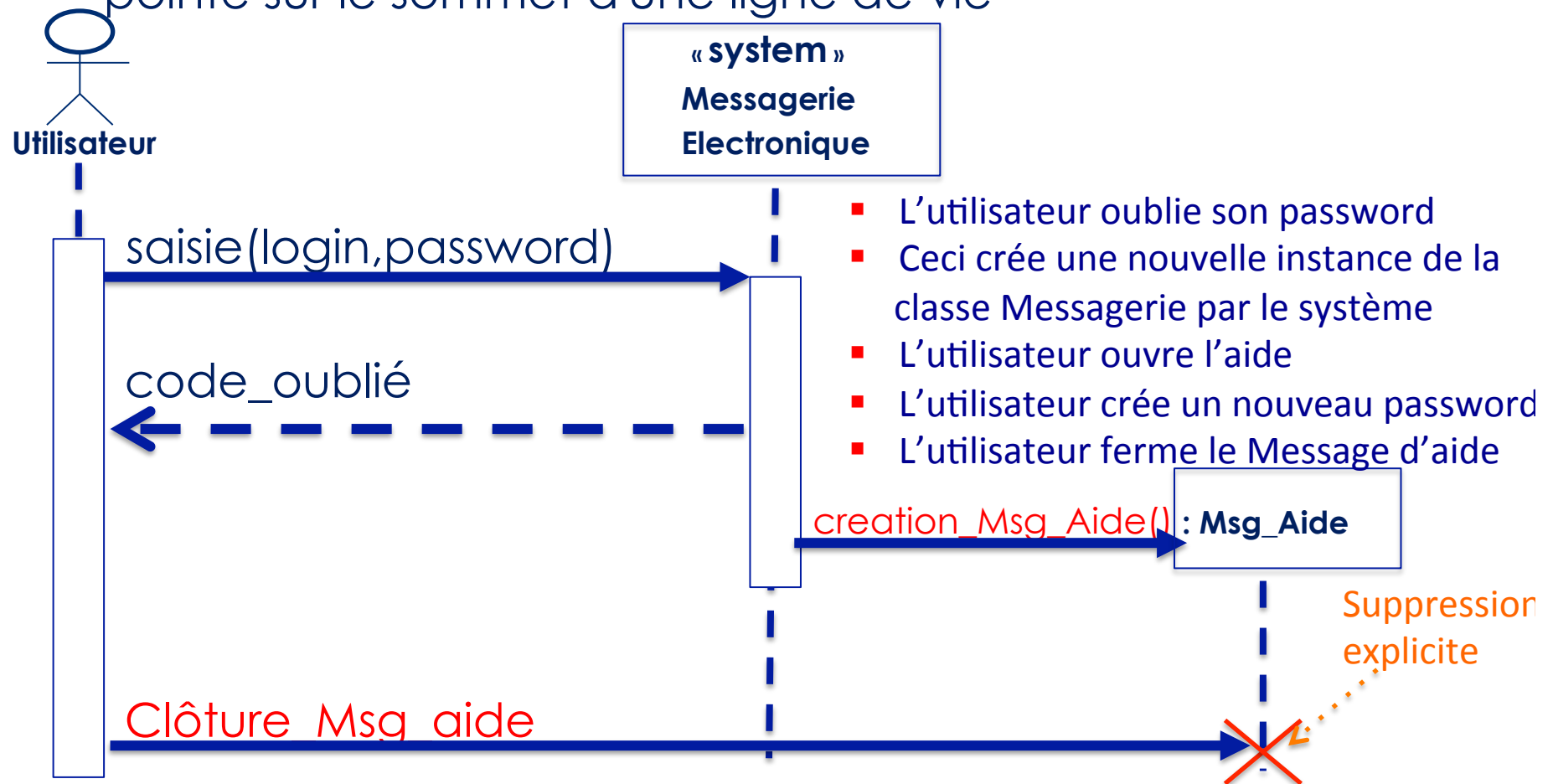
- ◆ L'invocation d'une opération est le type de message le plus utilisé en programmation objet ;
- ◆ Dans la pratique, la plupart des invocations sont synchrones, l'émetteur reste alors bloqué le temps que dure l'invocation de l'opération ;
- ◆ Le message de retour peut ne pas être représenté car il est inclut dans la fin d'exécution de l'opération de l'objet destinataire du message; le retour de message est implicite.

Description dynamique

Sémantique des composants d'un DSE

➤ 3.3 Message de création et de suppression d'instance :

- ✓ La création d'un objet est matérialisée par une flèche qui pointe sur le sommet d'une ligne de vie



Description dynamique

Sémantique des composants d'un DES

➤ 3.3 Message de suppression d'instance :

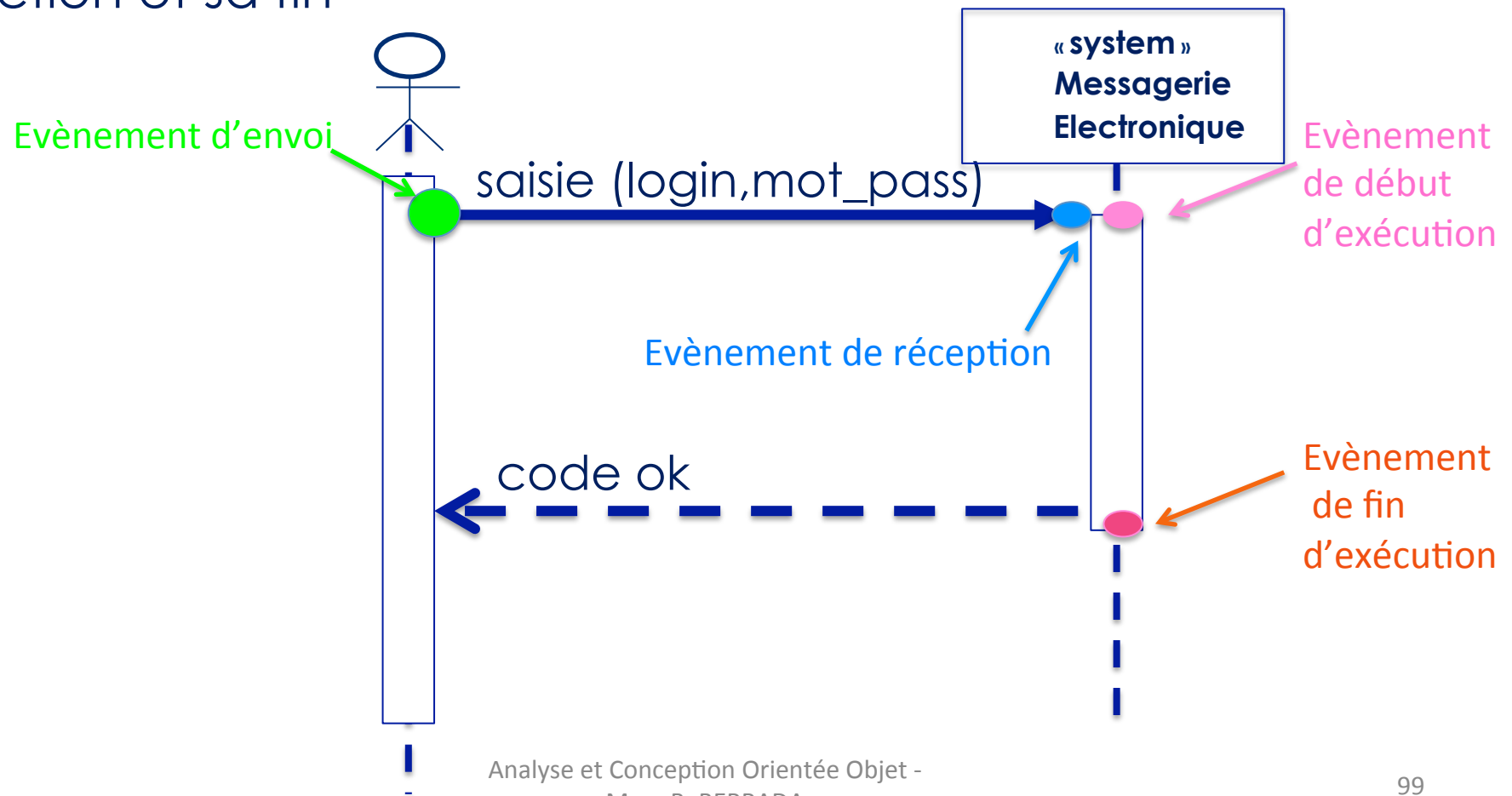
- ◆ La suppression d'un objet (participant) est indiquée par un **X** ;
- ◆ Une flèche de message entrant signifie qu'un objet en supprime explicitement un autre ;
- ◆ Un **X** à la fin d'une ligne de vie, montre que l'objet se supprime lui-même.
- ◆ Il est toujours utile d'utiliser un X pour indiquer qu'un objet n'est plus nécessaire, et qu'il est prêt à être collecté dans un environnement disposant d'un ramasse-miette ;
- ◆ Un **X** est également approprié aux opérations de clôture, pour signifier également que l'objet n'est plus utilisable.

Description dynamique

Sémantique des composants d'un DSE

➤ 3.4 Message et événements :

UML permet de séparer clairement l'envoi du message, sa réception, ainsi que le début de l'exécution de la réaction et sa fin



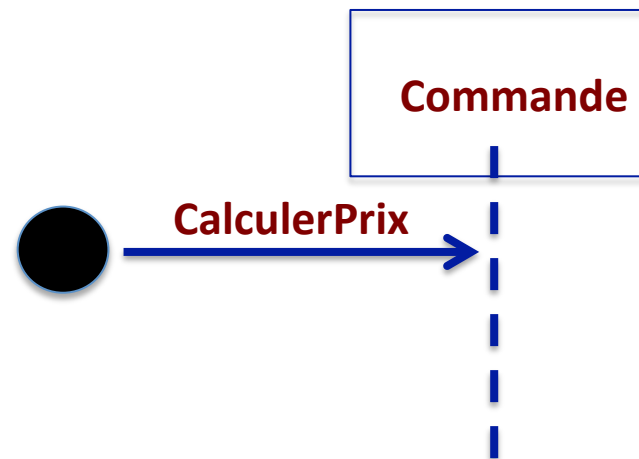
Description dynamique

Sémantique des composants d'un DSE

➤ 3.5 Message trouvé :

Un message trouvé est tel que l'événement de réception est connu, mais pas l'événement d'émission.

Une flèche partant d'une petite boule noire représente un message trouvé, comme illustré :



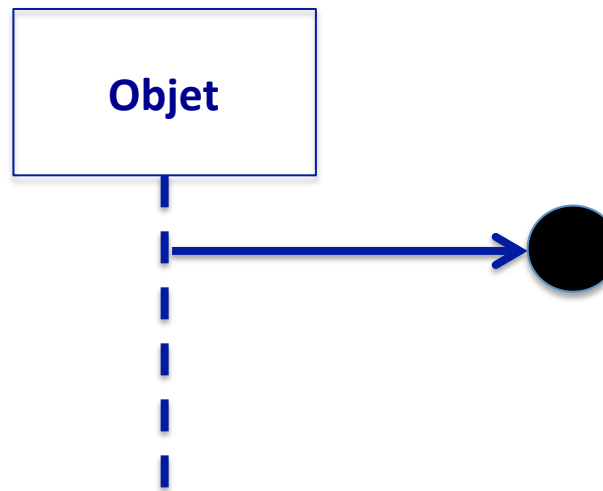
Description dynamique

Sémantique des composants d'un DSE

➤ **3.6 Message perdu :**

Un message perdu est tel que l'événement d'envoi est connu, mais pas l'événement de réception.

Il se représente par une flèche qui pointe sur une petite boule noire tel qu'illustré ci-dessous :

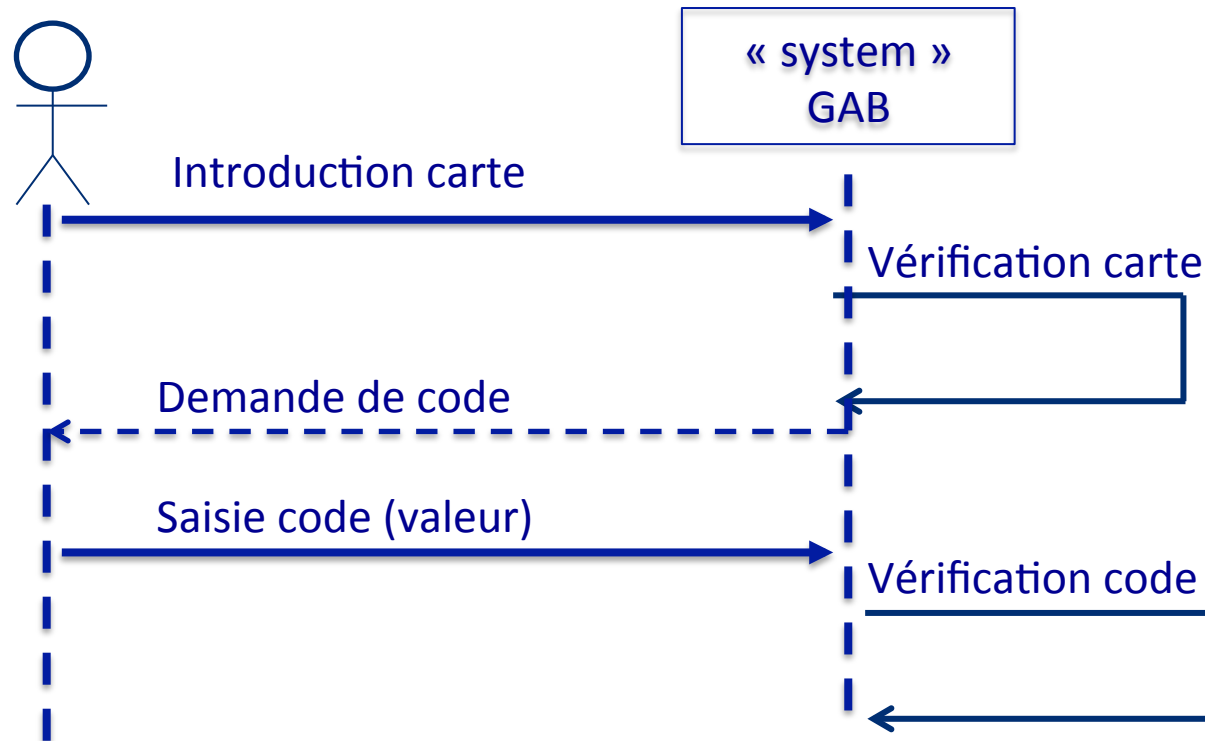


Description dynamique

Diagramme de Séquence (DSE):

➤ 3.7 Message réflexif :

Il représente un message que s'envoie un objet à lui-même (self-call):



Il représente une activité interne à l'objet ou une abstraction d'une autre interaction (qu'on peut détailler dans un autre diagramme de séquence)

Description dynamique

DSE : EXERCICE 1

Représenter un diagramme de séquence, décrivant le traitement suivant :

1. Un gestionnaire reçoit une requête;
2. Le gestionnaire formule une requête à l'aide d'une instruction dans la base de données ;
3. L'instruction retourne un résultat ;
4. Après extraction des informations utiles, l'instruction est clôturée et la requête est terminée.

[Solution ?](#)

Description dynamique

DSE : EXERCICE 2

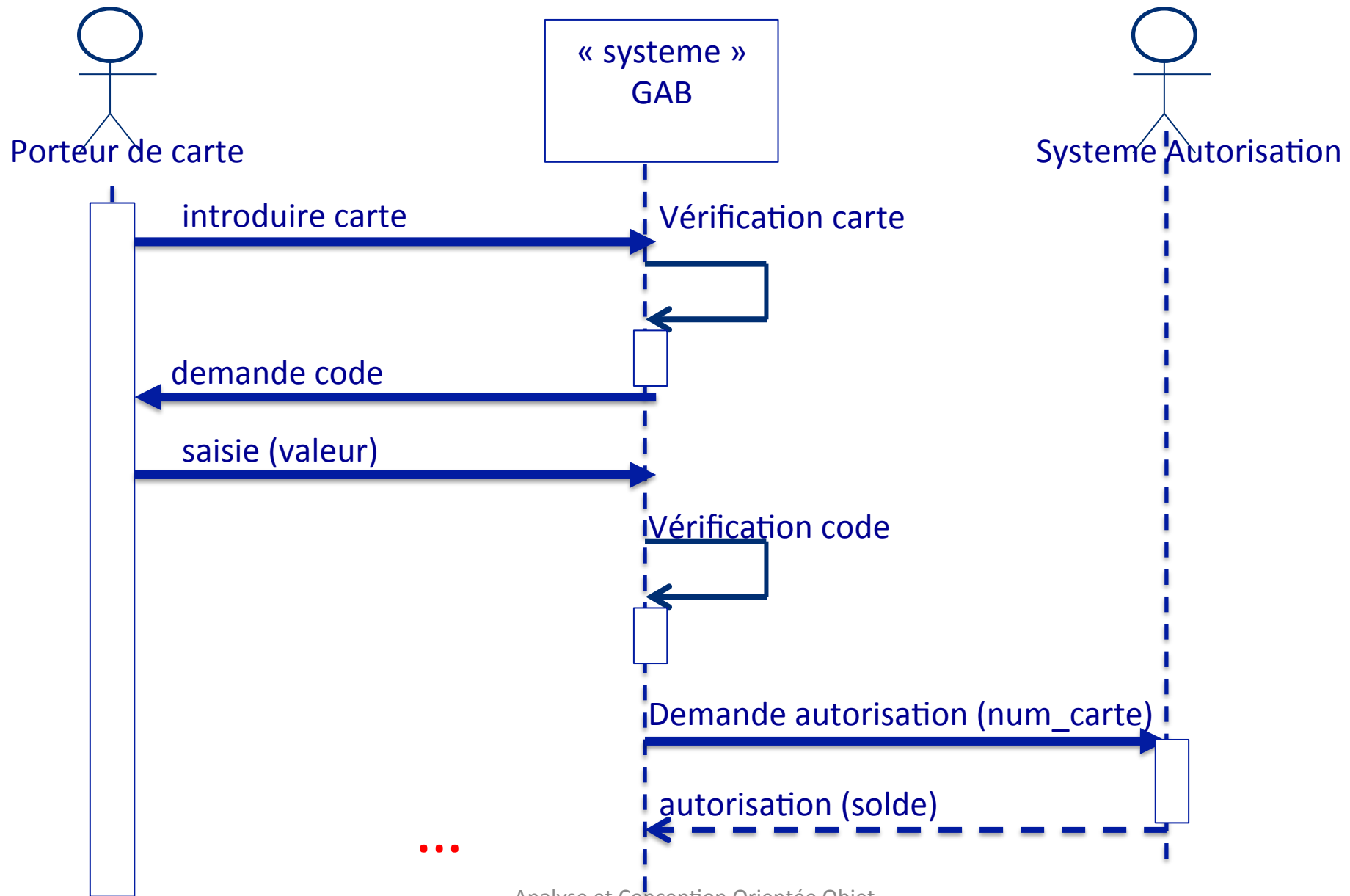
Exemple de diagramme de séquence qui décrit le scenario nominal du cas d'utilisation RETIRER ARGENT, proposé dans le TD 1 traitant de l'étude d'un Guichet Automatique de Banque (GAB).

1. Il suffit de transcrire sous forme de diagramme de séquence les interactions citées dans la description textuelle du scenario nominal du cas d'Utilisation RETIRER ARGENT (pour un porteur de carte non client de la banque), précisée dans la diapositive suivante.
2. Utiliser les conventions graphiques :
 - Acteur Principal Porteur de carte, à gauche
 - Un participant représentant le GAB au milieu
 - L'acteur secondaire Sys. Auto à droite du GAB

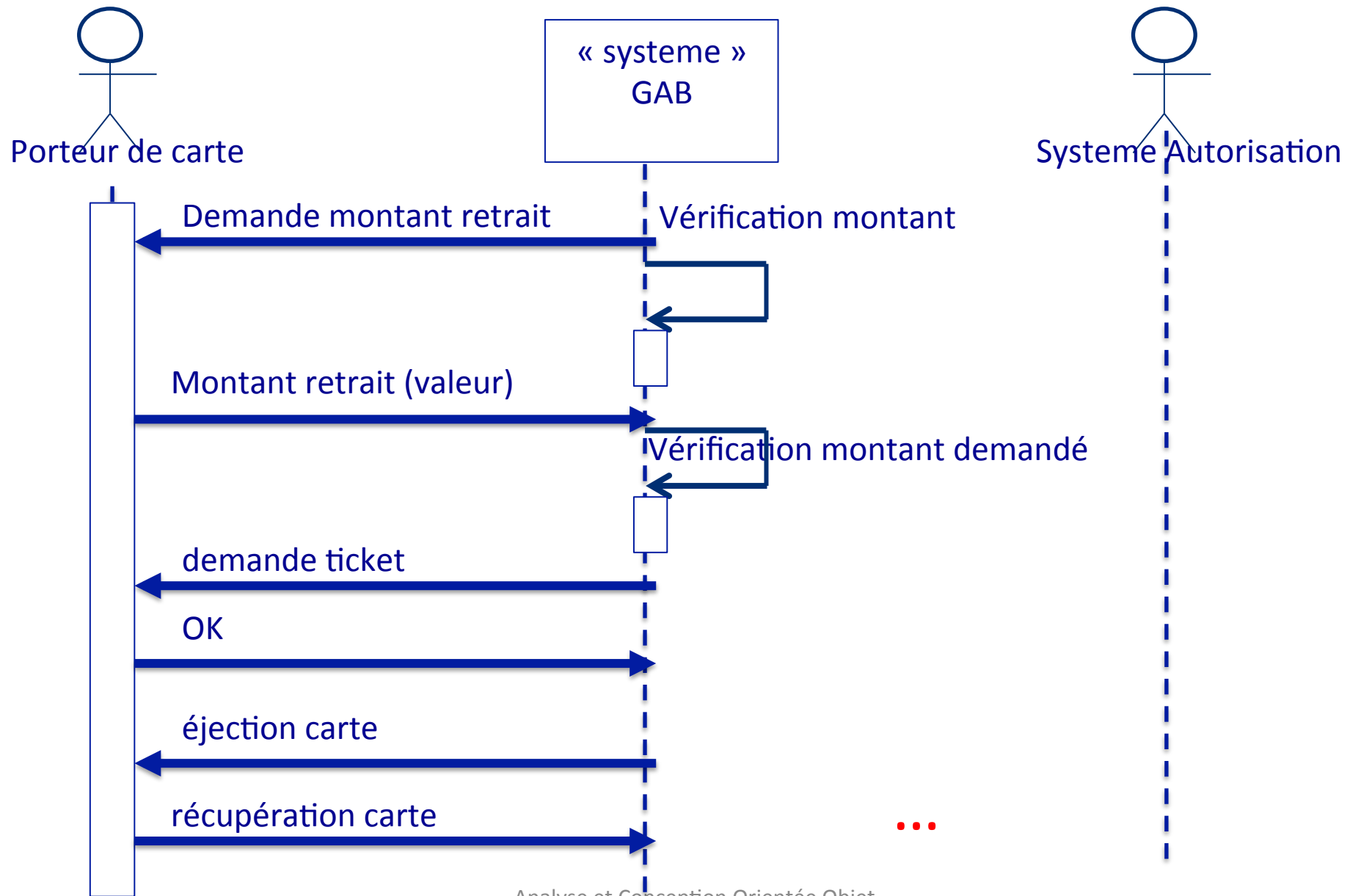
Cas d'Utilisation RETIRER ARGENT

1. Le Porteur de carte introduit sa carte dans le lecteur de cartes du GAB
2. Le GAB vérifie que la carte introduite est bien une carte bancaire
3. Le GAB demande au Porteur de carte de saisir son code d'indentification
4. Le Porteur de carte saisit son code d'identification
5. Le GAB compare le code d'identification avec celui qui est codé sur la puce de la carte
6. Le GAB demande une autorisation au Système d'autorisation
7. Le Système d'Autorisation donne son accord et indique le crédit hebdomadaire
8. Le GAB demande au Porteur de carte de saisir le montant désiré du retrait
9. Le Porteur de carte saisit le montant du retrait
10. Le GAB contrôle le montant demandé par rapport au crédit hebdomadaire
11. Le GAB demande au Porteur de carte s'il veut un ticket
12. Le Porteur de carte demande un ticket
13. Le GAB rend sa carte au Porteur de carte
14. Le Porteur de carte reprend sa carte
15. Le GAB délivre les billets et un ticket
16. Le Porteur de carte prend les billets et le ticket.

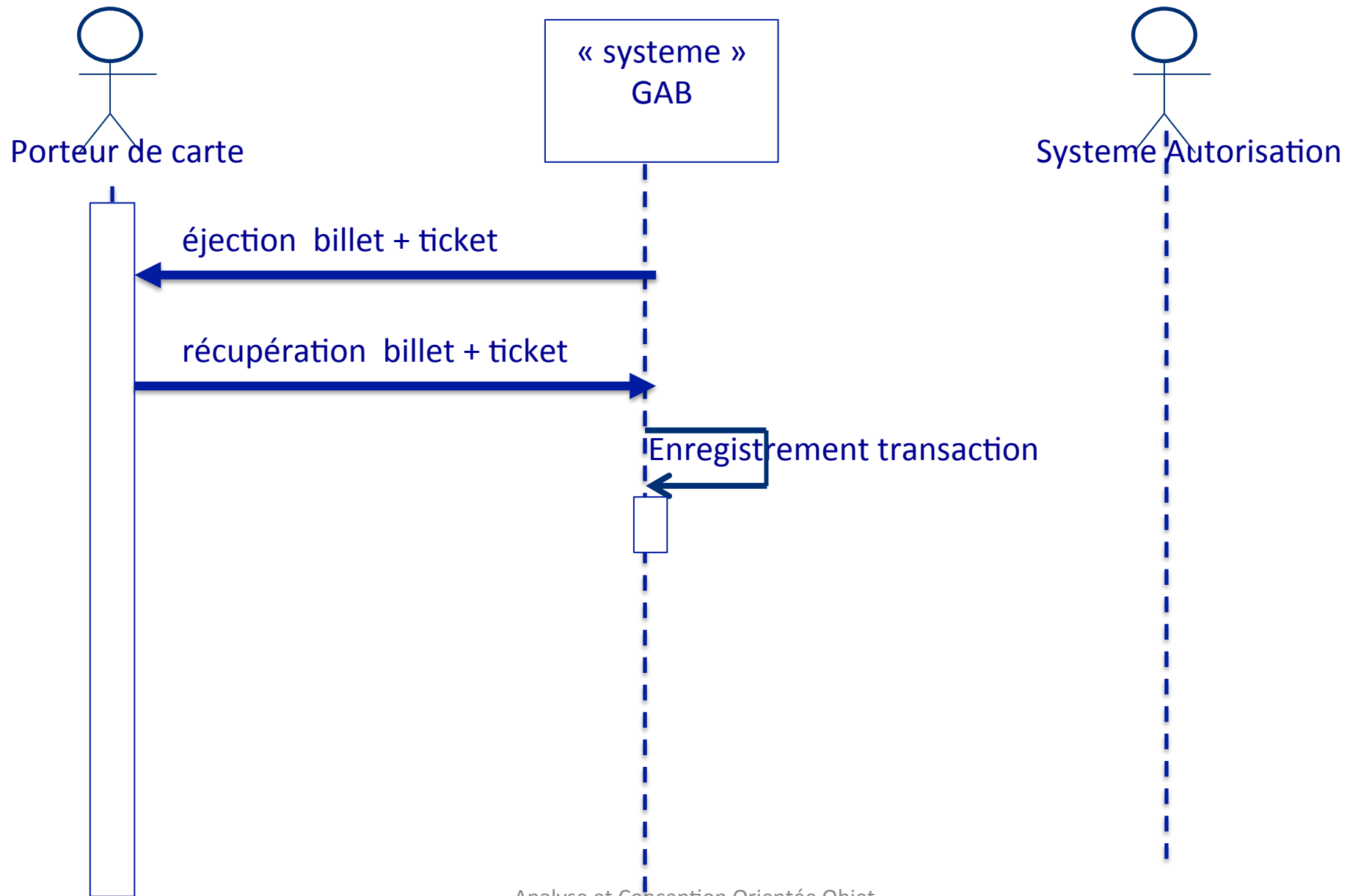
DSE : EXERCICE 2



DSE : EXERCICE 2



DSE : EXERCICE 2



Exercice :

1. le Diagramme de Contexte préliminaire de l'utilisation du GAB, montrant les cas d'utilisation (ovales) reliés par des associations (lignes) avec leurs acteurs principaux (icônes ou « stick-man ») et secondaires.
2. Le diagramme de Séquence pour le scénario étudié
3. *Le diagramme de séquence en ajoutant des relations d'inclusion, d'extension pour illustrer les fragments d'interaction*

Suite des Exercices :

Pour l'exercice 3,

3. Le diagramme de séquence en ajoutant des relations d'inclusion, d'extension pour illustrer les fragments d'interaction

3.1 Abordons tout d'abord la relation d'inclusion : le cas d'utilisation de base en incorpore explicitement un autre de façon obligatoire, à un endroit spécifique dans ses enchaînements.

On utilise cette relation pour décrire plusieurs fois le même enchaînement, et donc permettre de factoriser le comportement commun dans un cas d'utilisation à part.

Suite des Exercices :

Le début de la description textuelle du scénario nominal du cas d'utilisation **Retirer de l'argent**, (étapes de 1..5) va être similaire à tous les cas d'utilisation du client de la banque, en substituant « Porteur de carte » par « Client de la banque ».

1. Le Client de la banque introduit sa carte dans le lecteur de cartes du GAB
2. Le GAB vérifie que la carte introduite est bien une carte bancaire
3. Le GAB demande au Client de la banque de saisir son code d'identification
4. Le Client de la banque saisit son code d'identification
5. Le GAB compare le code d'identification avec celui qui est codé sur la puce de la carte

Cet enchaînement est en outre complété par enchaînements alternatifs ou d'erreur A1 (code provisoirement erroné), E1 (carte non valide) et E2 (code d'identification définitivement erroné)

Suite des Exercices :

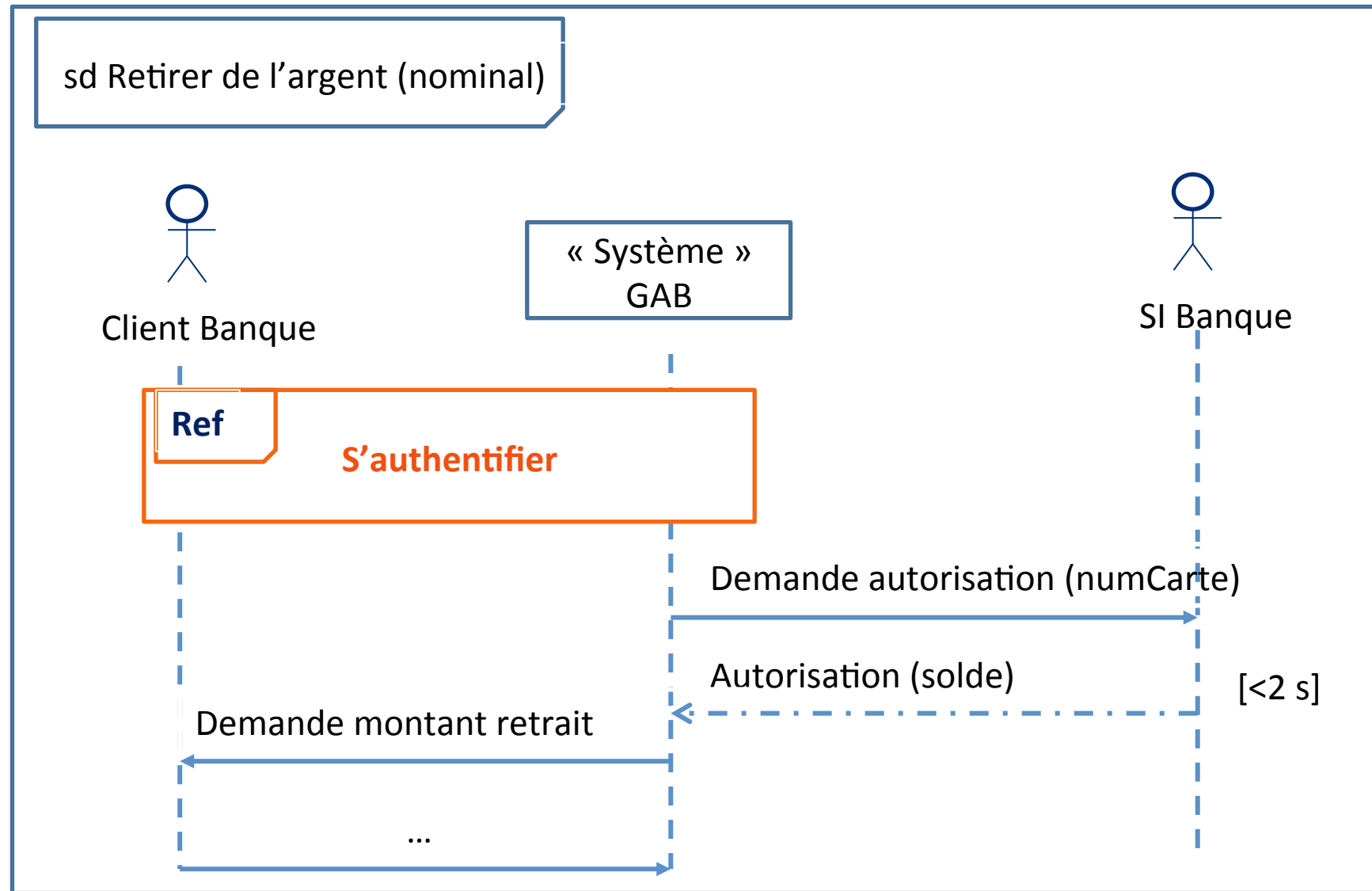
Un nouveau cas d'utilisation : S'Authentifier, inclus dans les précédents cas d'utilisation, entre autres Consulter Solde, Déposer Argent...

Avantages :

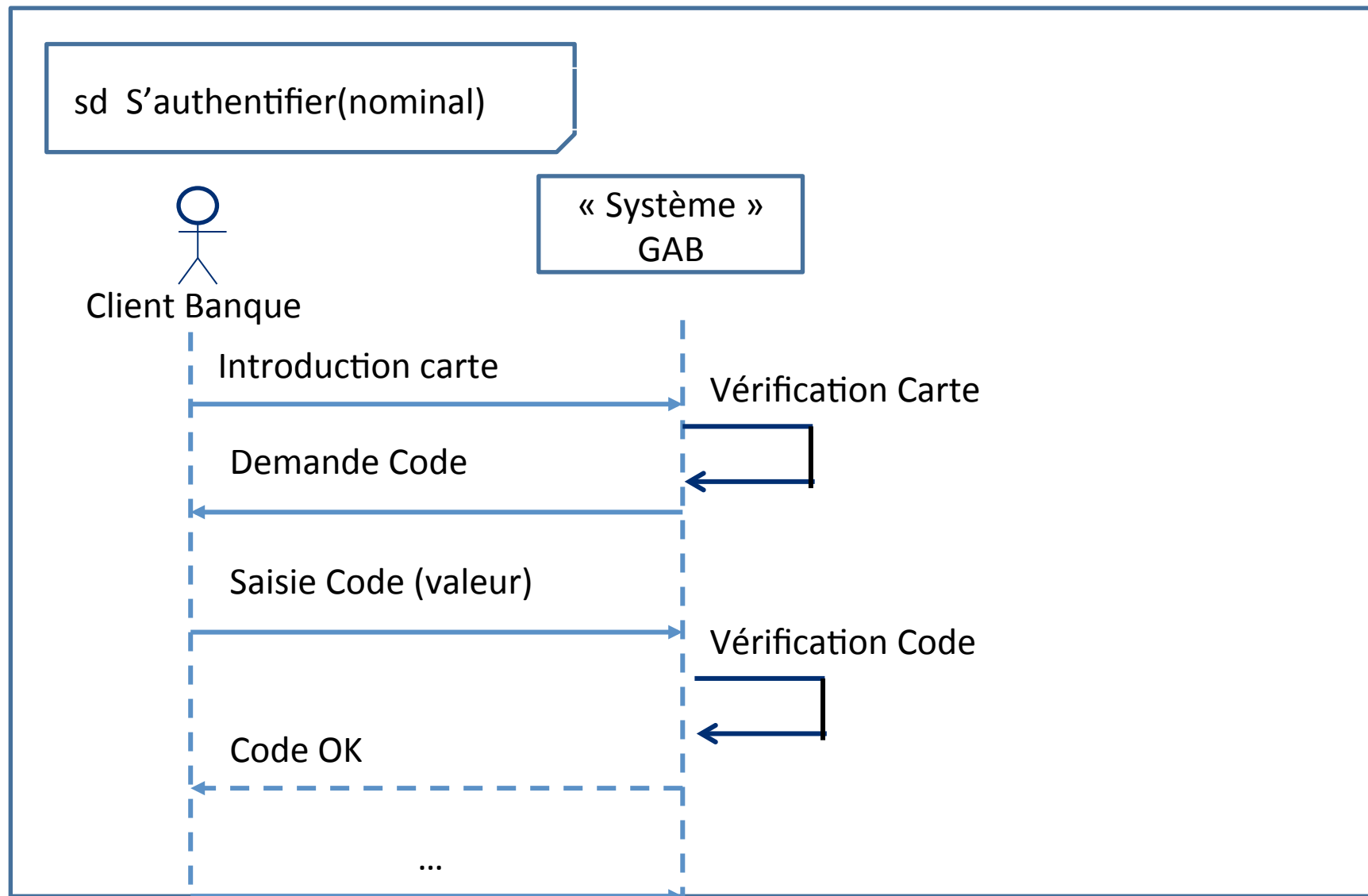
- **S'Authentifier** va contenir tous les enchaînements de 1 à 5
- Les descriptions textuelles des autres cas d'utilisation vont être simplifiées et focalisées sur les spécificités fonctionnelles
- Les descriptions redondantes seront éliminées.

TP 3.1

Nouveau diagramme de séquence du système avec référence au cas inclus



Le diagramme de séquence du fragment référencé



Suite des Exercices :

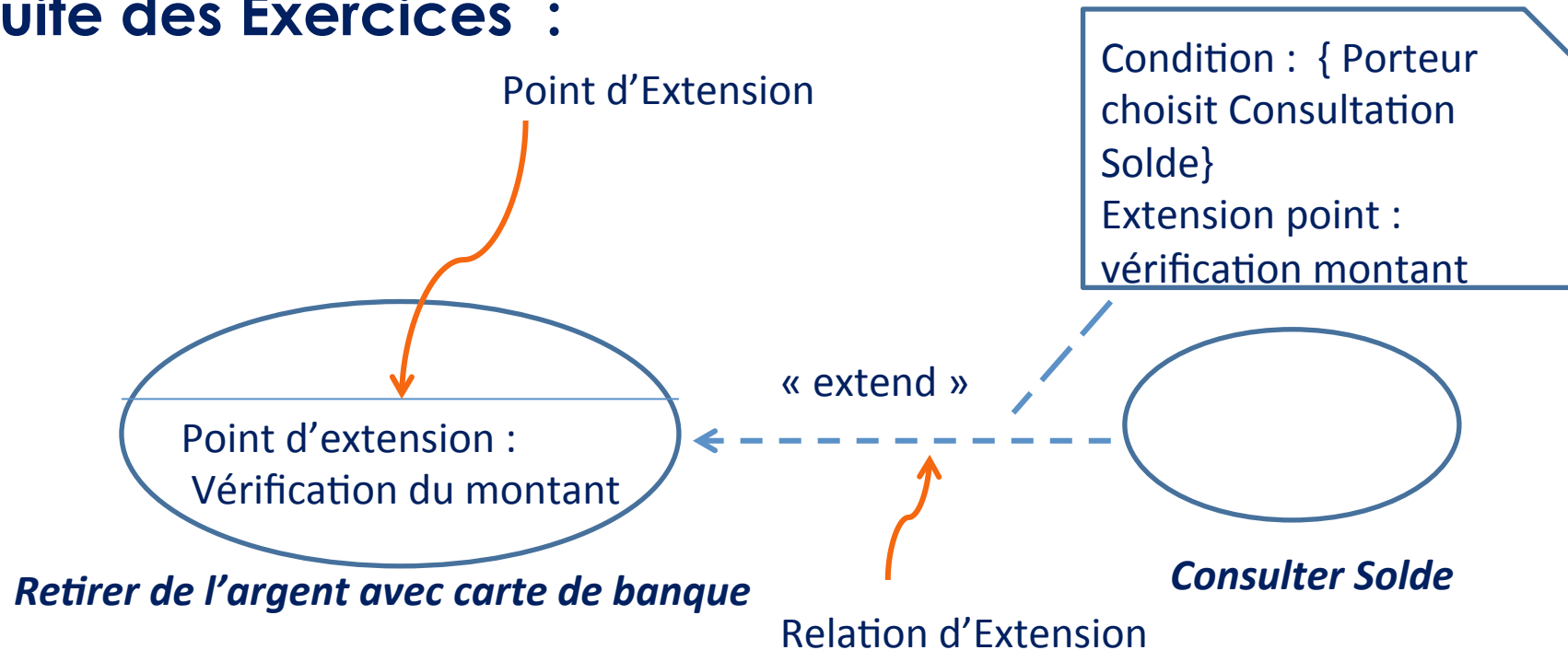
Pour l'exercice 3,

3. Le diagramme de séquence en ajoutant des relations d'extension

3.1 Rappelons la relation d'extension : le cas d'utilisation de base en incorpore implicitement un autre de façon optionnelle, à un endroit spécifique dans le cas d'utilisation qui procède à l'extension.

On utilise cette relation pour séparer un comportement optionnel ou rare du comportement obligatoire.

Suite des Exercices :



Ce point d'extension doit être déclaré dans la description textuelle, par exemple en modifiant L'enchaînement nominal :

...

7. Le SI banque donne son accord et indique le solde hebdomadaire

8. Le GAB demande au Client banque de saisir le montant désiré du retrait

Point d'extension : Vérification montant

9. Client banque saisit le montant désiré

10. Le GAB contrôle le montant demandé par rapport au solde hebdomadaire.

...

Nouveau diagramme de séquence du système avec référence au cas étendu

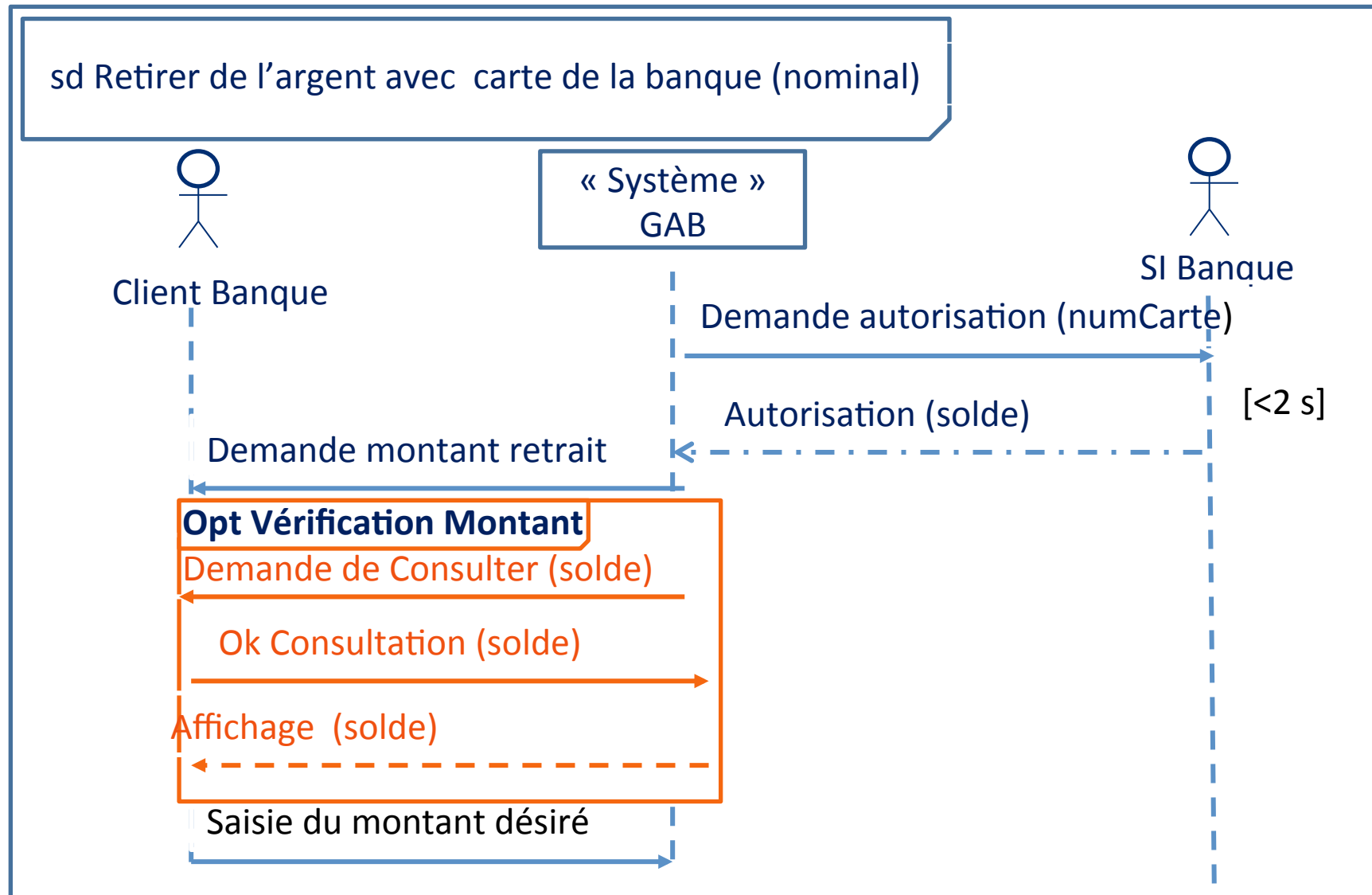


Diagramme de Séquence

Fragment d'interaction

Les cadres d'interactions sont utilisés comme ceux représentés, pour distinguer des sous-ensembles qui constituent des Fragments d'interactions.

Une indication dans le coin gauche, dans les cas précédents, ref ou opt portant le nom de l'interaction.

Treize opérateurs de fragments d'interaction (combiné) existent :

ref, opt, loop, alt, par, strict/weak, break, ignore/consider, critical, et assertion

Diagramme de Séquence

Fragment d'interaction

A l'image des deux premiers, l'opérateur :

alt : correspond à une instruction de test avec une ou plusieurs alternatives possibles. Seul le fragment dont la condition est vraie s'exécutera

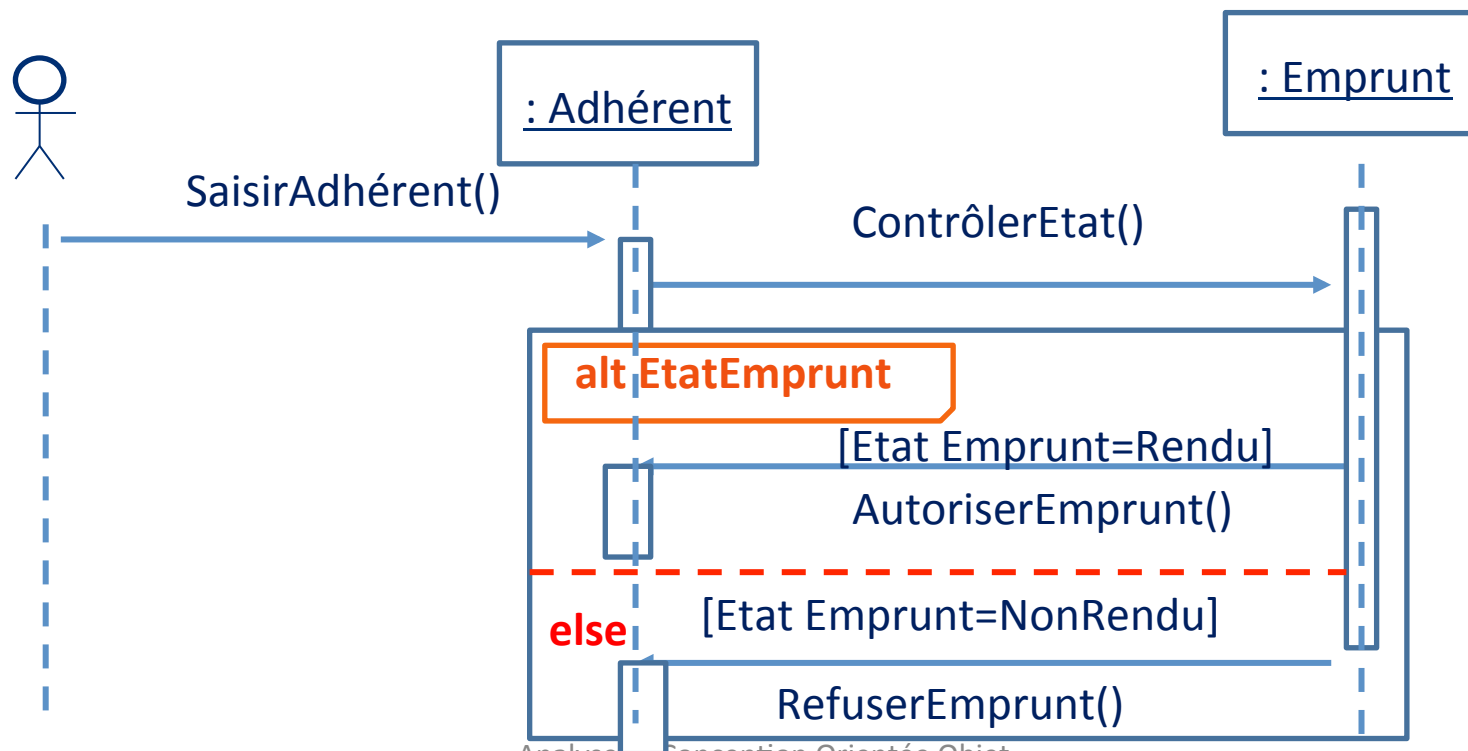


Diagramme de Séquence

Fragment d'interaction

A l'image des deux premiers, l'opérateur :

loop : correspond à une instruction de boucle qui permet d'exécuter une séquence d'interaction tant qu'une condition n'est pas satisfaite. Une garde indique la condition de l'iteration.

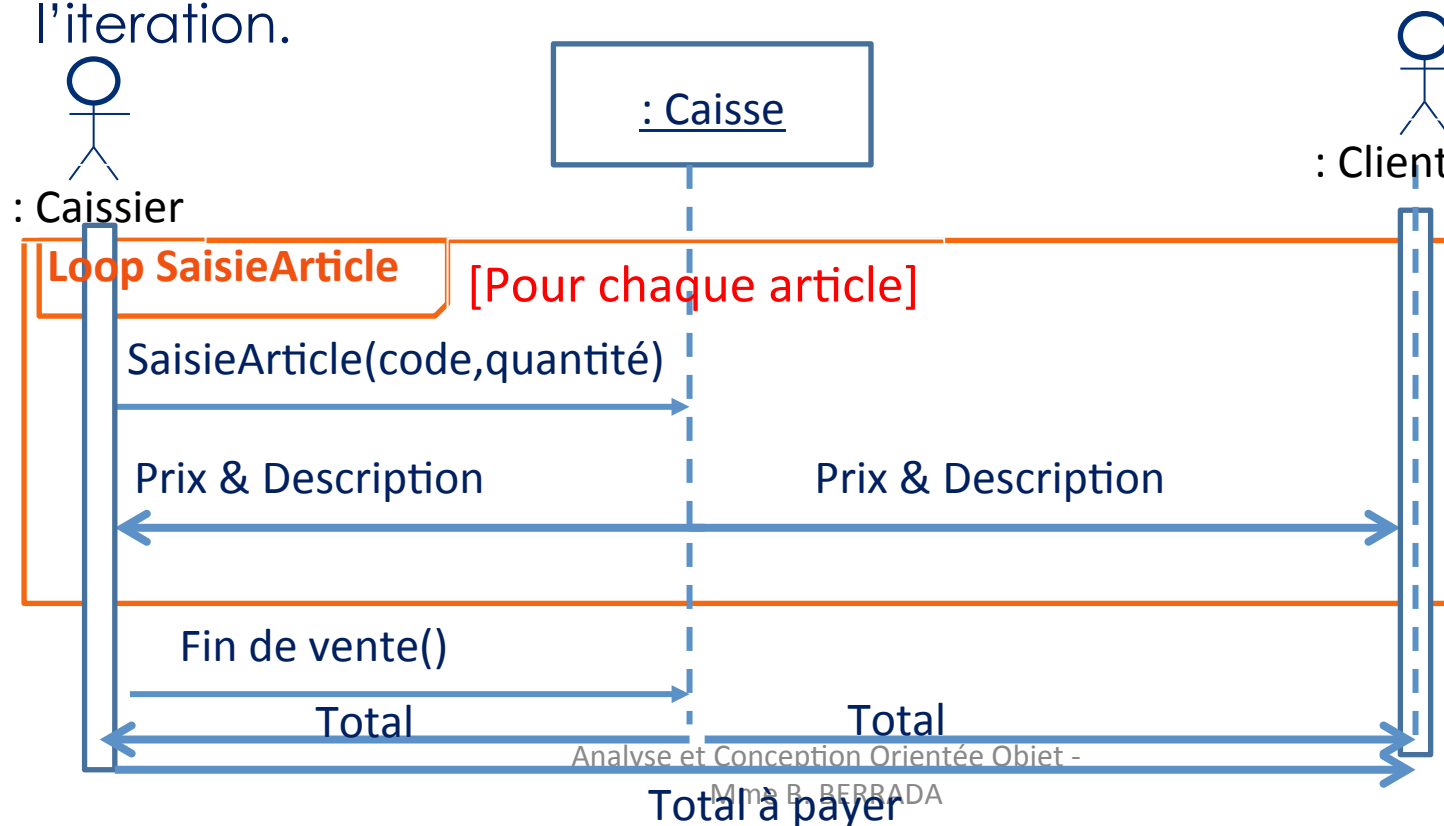


Diagramme de Séquence

Fragment d'interaction : par

par : correspond à une instruction qui permet de représenter deux séries d'interactions qui se déroulent en parallèle.

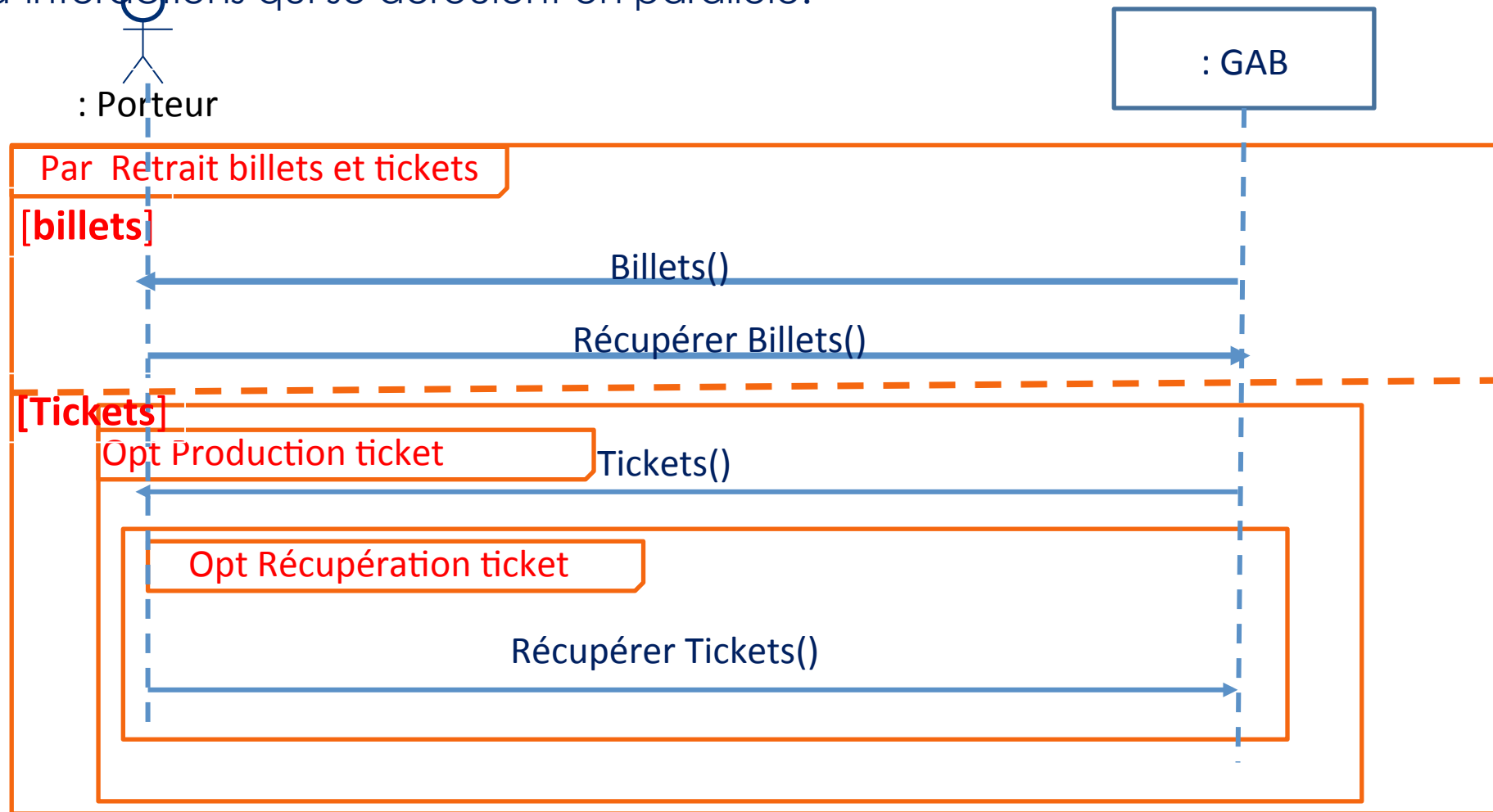


Diagramme de Séquence

Fragment d'interaction

Les autres opérateurs :

strict/weak sequencing : permettent de préciser que l'ordre des opérations(strict) ou pas important

Break : permet de représenter une situation exceptionnelle correspondant à un scénario de rupture (avec une condition de garde, par rapport au scénario général.

Consider/ignore : expriment respect. que des messages doivent être soit obligatoirement présents (consider), soit absents (ignore), sans incidence sur le déroulement des interactions.

Critical : indique une séquence d'interaction critique, donc **ne pouvant être interrompue** de l'importance des opérations traitées.

Diagramme de Séquence

Fragment d'interaction

Suite des autres opérateurs :

assertion : indique qu'une séquence d'interaction est l'unique séquence possible en considérant les messages échangés dans le fragment d'interaction. Toute autre configuration de message est invalide.

négative : indique le fragment représente une interaction invalide.

ARRET 3^{ème} Séance