

UML : DIAGRAMME DE SÉQUENCE



ECOLE NORMALE SUPÉRIEURE DE L'ENSEIGNEMENT
TECHNIQUE DE MOHAMMEDIA
UNIVERSITÉ HASSAN II DE CASABLANCA

المدرسة العليا لأساتذة التعليم التقني المحمدية

2^{ème} année Cycle Ingénieur

GLSID 2 ICCN 2 & IIBDCC 2
2024/2025

Pr. SARA RETAL

INTRODUCTION

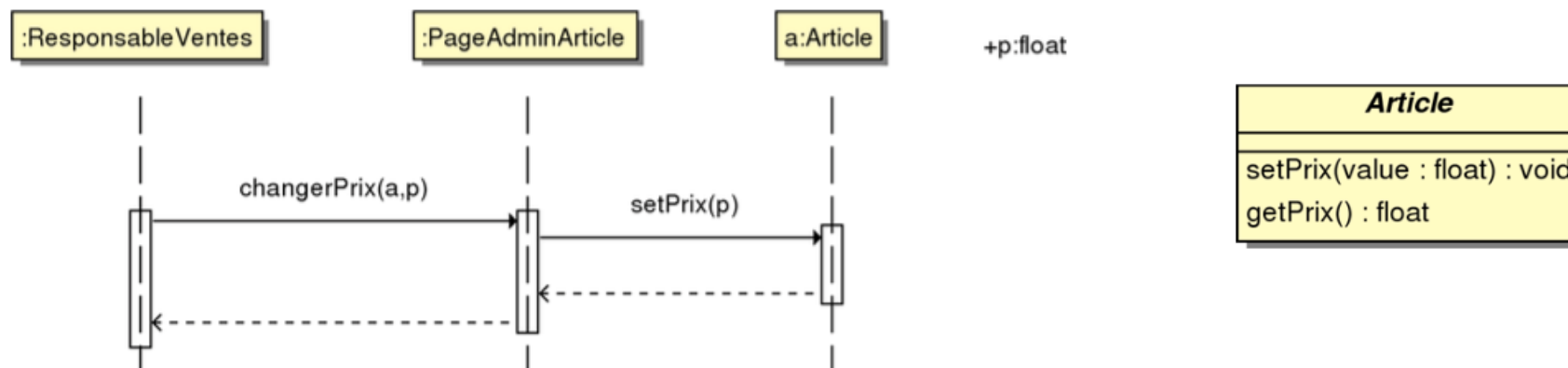
- Les **diagrammes de cas d'utilisation** modélisent à **QUOI** sert le système, en organisant les interactions possibles avec les acteurs.
- Les **diagrammes de classes** permettent de spécifier la structure et les liens entre les objets dont le système est composé : ils spécifient **QUI** sera à l'œuvre dans le système pour réaliser les fonctionnalités décrites par les diagrammes de cas d'utilisation.
- Par contre, les **diagrammes de séquences** permettent de décrire **COMMENT** les éléments du système interagissent entre eux et avec les acteurs. Les objets au cœur d'un système interagissent en échangeant des messages. Les acteurs interagissent avec le système au moyen d'IHM (Interfaces Homme-Machine).

EXEMPLE D'INTERACTION

- ❑ Soit le cas d'utilisation « ChangerPrixArticle » qui est représenté dans la figure ci-dessous:

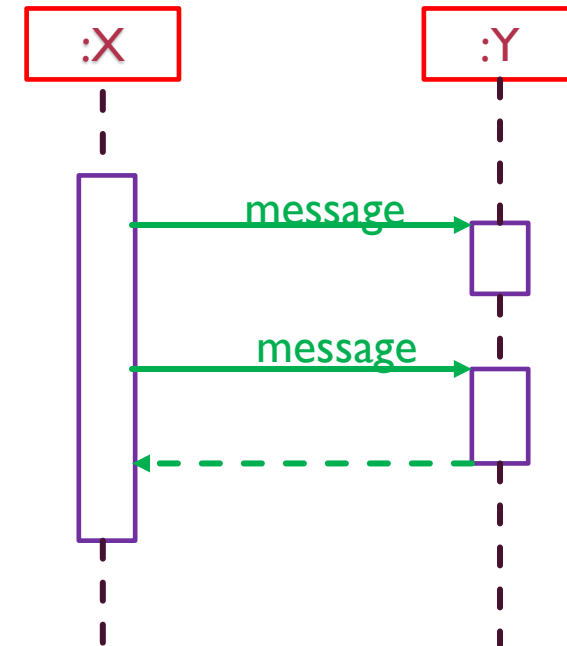


- ❑ Son diagramme de séquences correspondant est le suivant:



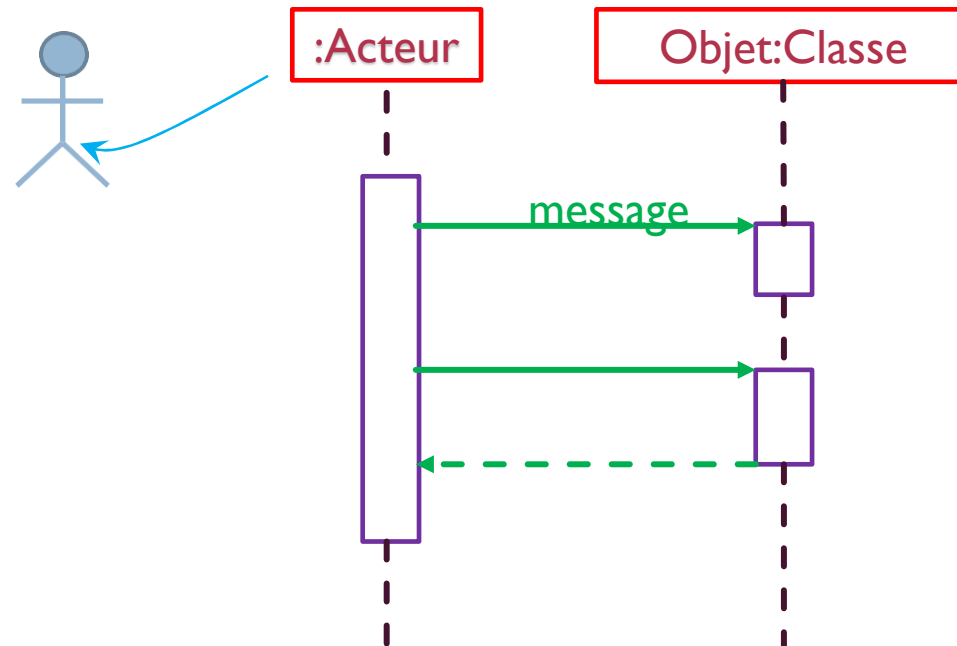
CONCEPTS PRINCIPAUX

- Les objets / acteurs
- La ligne de vie de l'objet/acteur
- Les messages
- Le point de contrôle (barre d'activation)



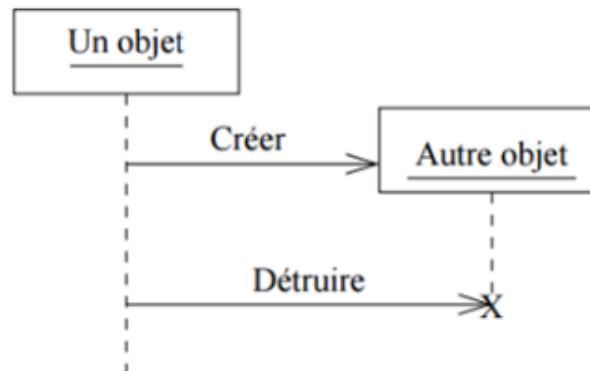
LES OBJETS / ACTEURS

Sur un diagramme de séquence, les objets apparaissent toujours dans la partie supérieure, ce qui facilite l'identification des classes qui participent à l'interaction.



LIGNE DE VIE DES OBJETS

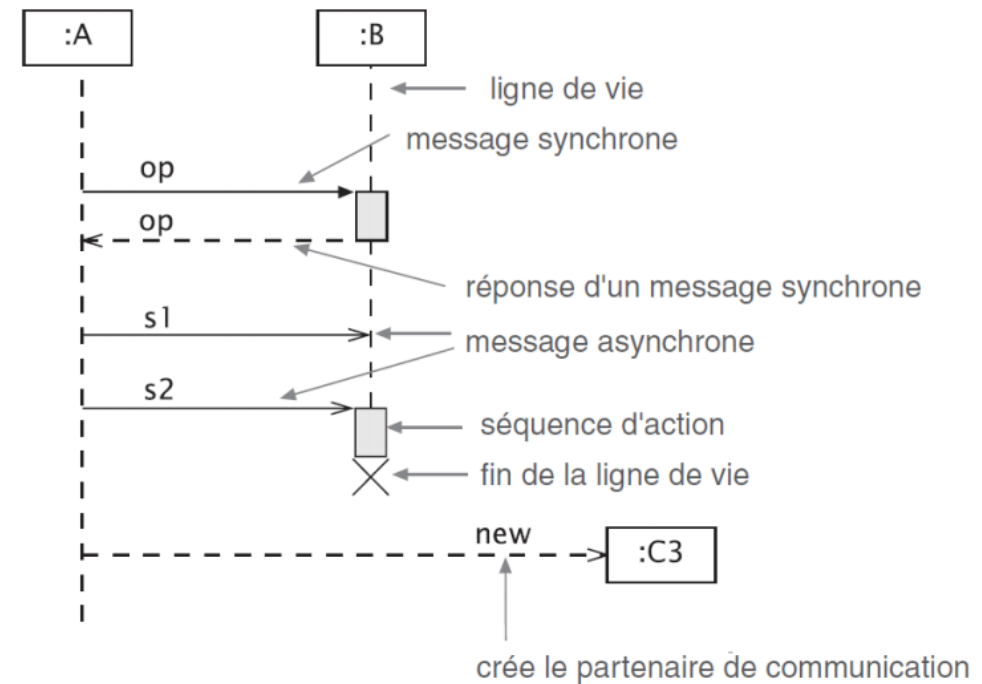
- ❑ Ce concept représente la vie d'un objet dans le contexte de la séquence d'événements. Les objets qui sont créés vers la fin de la séquence n'apparaissent pas toujours en haut du diagramme, mais peuvent apparaître à l'endroit où ils sont créés.
- ❑ La ligne de vie peut se terminer à l'endroit où l'objet est détruit. De même, la création d'un objet peut être représentée simplement par l'envoi d'un nouveau message à un objet.



LES MESSAGES

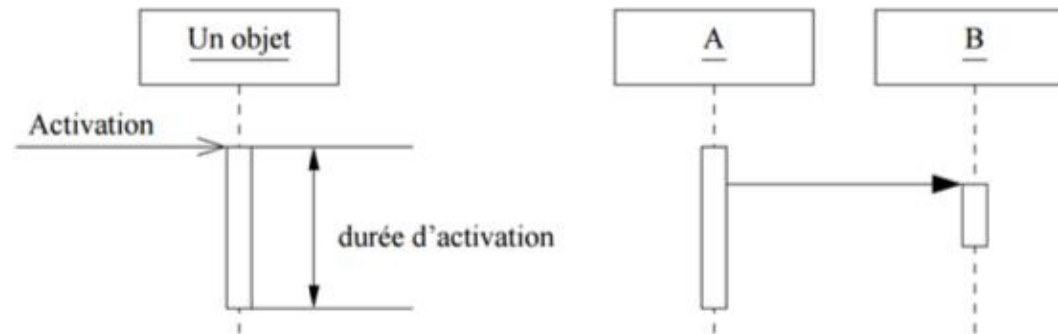
Un message définit une communication particulière entre des lignes de vie (objets ou acteurs). Plusieurs types de messages existent, dont les plus courants :

- l'envoi d'un signal.
- l'invocation d'une opération (appel de méthode).
- la création ou la destruction d'un objet.



LE POINT DE CONTRÔLE (BARRE D'ACTIVATION)

- La dimension verticale représente l'écoulement du temps. Une période d'activité correspond au temps pendant lequel un objet effectue une action directe ou indirecte. Elle est représentée par une bande verticale le long de la ligne de vie de l'objet. graphiquement elle est représentée en plaçant un rectangle au-dessus de la ligne de vie de l'objet.



LES PRINCIPAUX TYPES DE MESSAGE

- Un message synchrone bloque l'expéditeur jusqu'à la réponse du destinataire. Le flot de contrôle passe de l'émetteur au récepteur.

Typiquement : appel de méthode Si un objet A invoque une méthode d'un objet B, A reste bloqué tant que B n'a pas terminé.



On peut associer aux messages d'appel de méthode un message de retour (en pointillés) marquant la reprise du contrôle par l'objet émetteur du message synchrone.

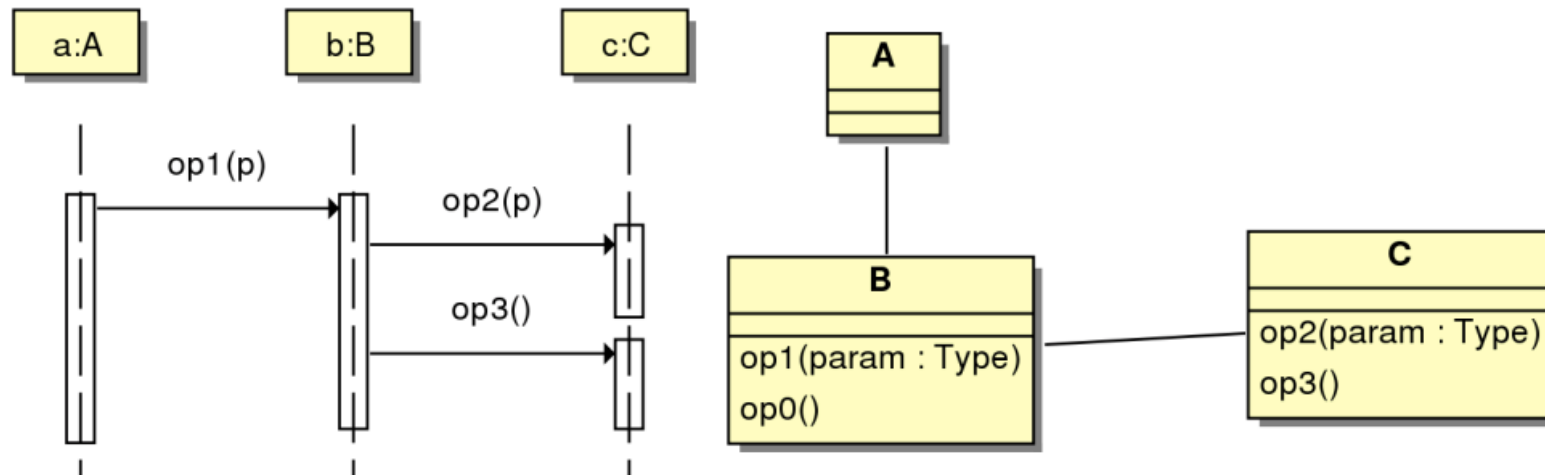
- Un message asynchrone n'est pas bloquant pour l'expéditeur. Le message envoyé peut être pris en compte par le récepteur à tout moment ou ignoré.

Typiquement : envoi de signal (voir stéréotype de classe « signal »)

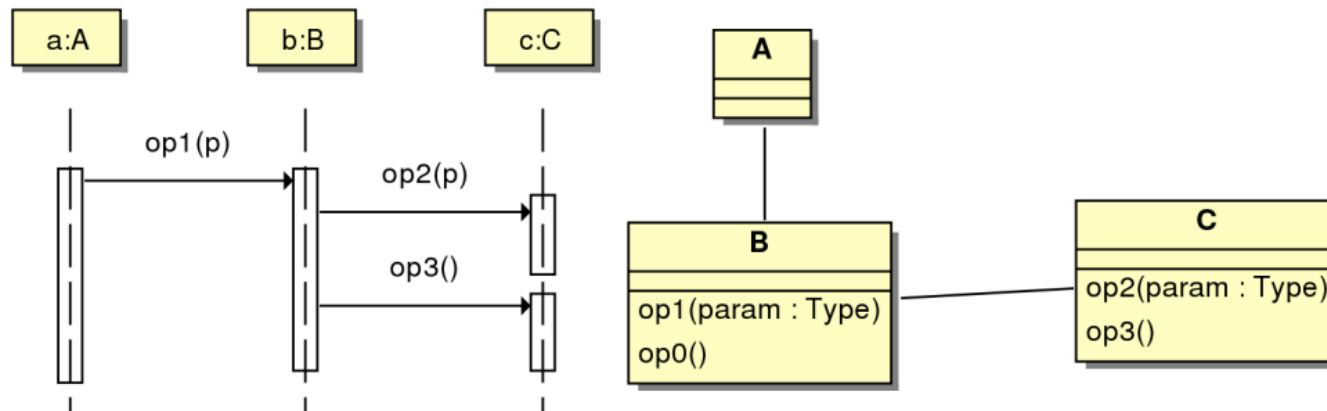


LES MESSAGES SYNCHRONES

- Les messages synchrones correspondent à des opérations dans le diagramme de classes.
- Envoyer un message et attendre la réponse pour poursuivre son activité revient à invoquer une méthode et attendre le retour pour poursuivre ses traitements.



L'IMPLÉMENTATION DES MESSAGES SYNCHRONES EN JAVA



```
class B {  
    C c;  
    op1(p:Type){  
        c.op2(p);  
        c.op3();  
    }  
}
```

```
class C {  
    op2(p:Type){  
        ...  
    }  
    op3(){  
        ...  
    }  
}
```

MESSAGES ASYNCHRONES

Dans le cas d'un message asynchrone, l'expéditeur n'attend pas la fin de l'activation de la méthode invoquée chez le destinataire. Un message asynchrone peut être :

- **Un appel de méthode : Fréquent dans un système multithreads (multitâche):**

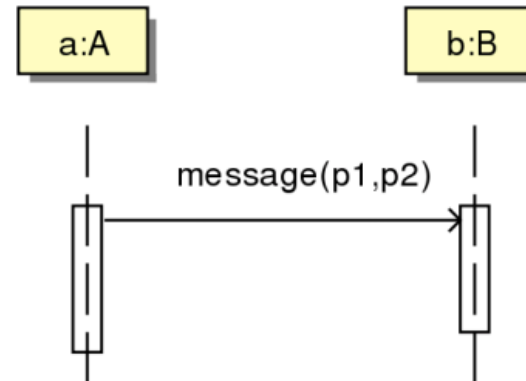
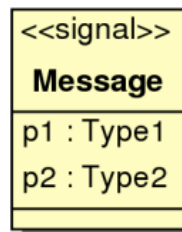
Ainsi, l'objet expéditeur n'étant pas bloqué pendant l'exécution de la méthode, continuer ainsi à envoyer d'autres messages.

MESSAGES ASYNCHRONES

Un message asynchrone peut être :

- **Un signal (cas le plus fréquent) :**

L'objet expéditeur transmet juste une information à l'objet destinataire. Souvent, ce sont les acteurs ou les périphériques qui envoient des signaux, typiquement utilisé dans la gestion événementielle d'une IHM graphique, comme la librairie QT, par exemple. Les signaux sont des objets dont la classe est stéréotypée «signal» et dont les attributs (porteurs d'information) correspondent aux paramètres du message.

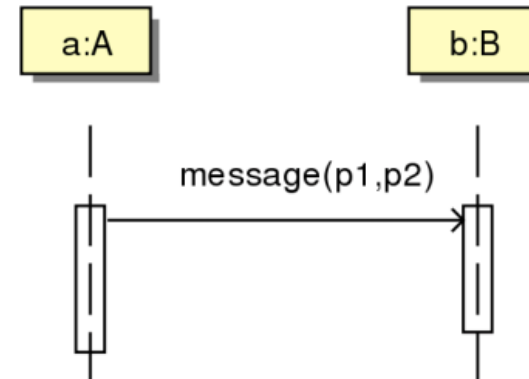
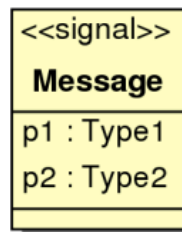


MESSAGES ASYNCHRONES

Un message asynchrone peut être :

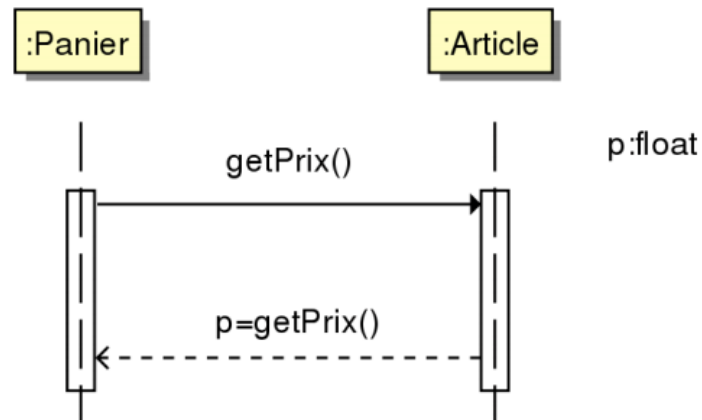
- **Un signal (cas le plus fréquent) :**

L'objet expéditeur transmet juste une information à l'objet destinataire. Souvent, ce sont les acteurs ou les périphériques qui envoient des signaux, typiquement utilisé dans la gestion événementielle d'une IHM graphique, comme la librairie QT, par exemple. Les signaux sont des objets dont la classe est stéréotypée «signal» et dont les attributs (porteurs d'information) correspondent aux paramètres du message.



MESSAGE DE RETOUR

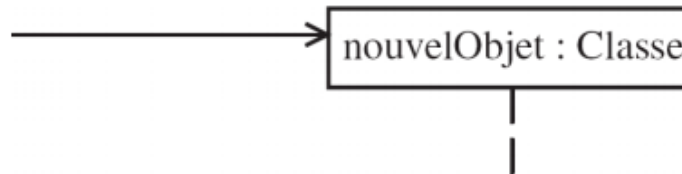
- Le récepteur d'un message synchrone rend la main à l'émetteur du message en lui envoyant un message de retour.
- Les messages de retour sont optionnels : la fin de la période d'activité marque également la fin de l'exécution d'une méthode. Ils sont utilisés pour spécifier le résultat de la méthode invoquée.



- Le retour des messages asynchrones s'effectue par l'envoi de nouveaux messages asynchrones.

LES MESSAGE DE CRÉATION ET DE DESTRUCTION DE LIGNES DE VIE

- La création d'un objet est matérialisée par une flèche qui pointe sur le sommet d'une ligne de vie. On peut aussi utiliser un message asynchrone ordinaire portant le nom «create».



- La destruction d'un objet est matérialisée par une croix qui marque la fin de la ligne de vie de l'objet.

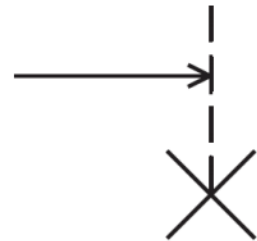
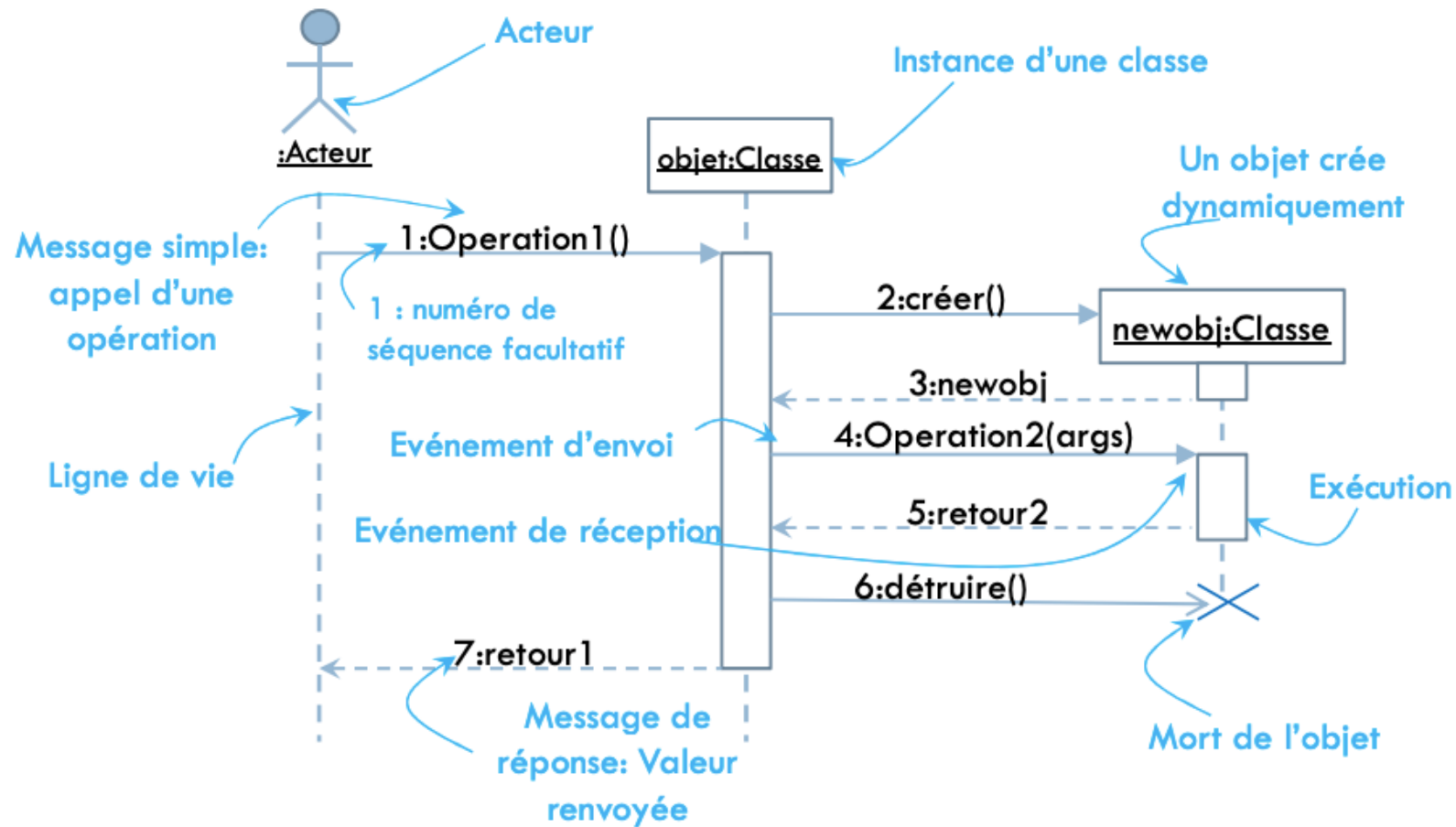


DIAGRAMME DE SEQUENCE



MESSAGES COMPLET, PERDU ET TROUVÉ

- Un message complet est tel que les évènements d'envoi et de réception sont connus. Il est représenté par une flèche partant d'une ligne de vie et arrivant à une autre ligne de vie.
- Un message perdu est tel que l'événement d'envoi est connu, mais pas l'événement de réception. La flèche part d'une ligne de vie mais arrive sur un cercle indépendant marquant la méconnaissance du destinataire. Exemple : broadcast.



- Un message trouvé est tel que l'événement de réception est connu, mais pas l'événement d'émission. Un message trouvé est un message pour lequel soit l'émetteur est inconnu, soit le message provient d'une source aléatoire.



SYNTAXE DES MESSAGES

- La syntaxe des messages est :
nomSignalOuOperation (paramètres)
- La syntaxe des arguments est la suivante :
nomParametre = valeurParametre
- Pour un argument modifiable :
nomParametre : valeurParametre
- Exemples :
 - appeler("Capitaine Hadock", 54214110)
 - afficher(x,y)
 - initialiser(x=100)
 - f(x:12)

MESSAGE RÉCURSIF

- appelé aussi message réflexif envoyé d'un objet vers lui-même. On peut montrer qu'un objet s'envoie un message à lui-même à l'aide d'une flèche circulaire.

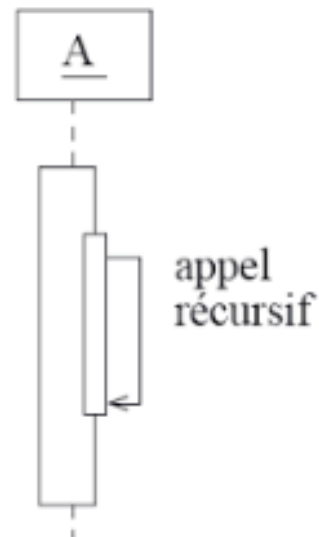


DIAGRAMME DE SÉQUENCE: EXEMPLE I

- Sur l'exemple suivant, un client demande un devis, un commercial recherche le prix de base du produit et les ristournes accordées et communique le devis au client.

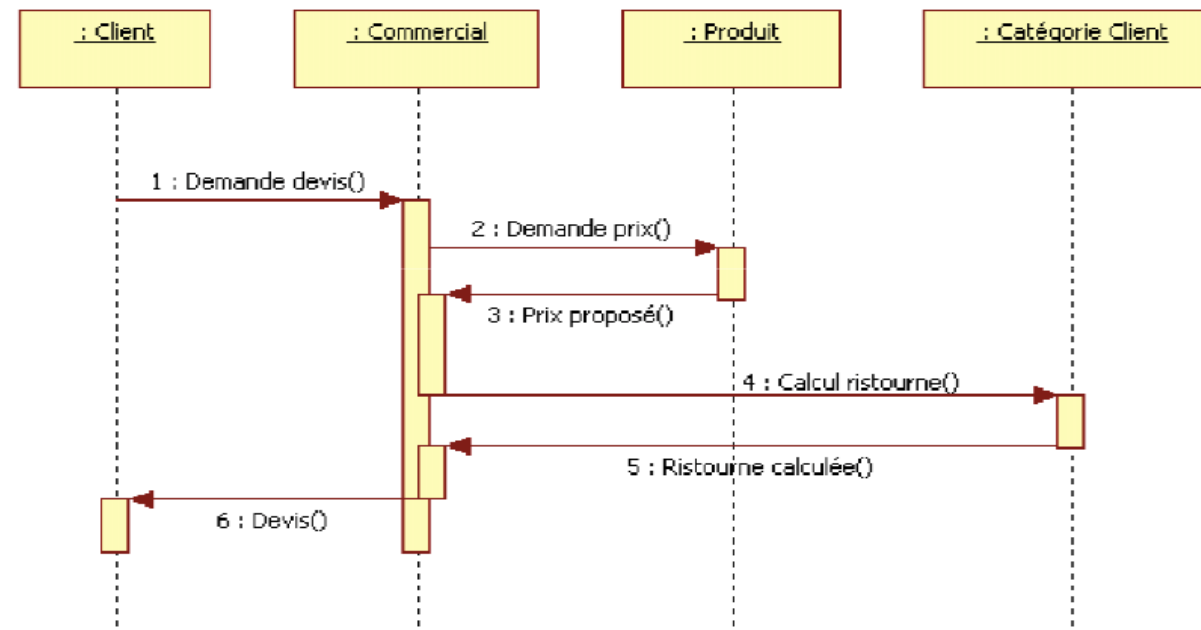


DIAGRAMME DE SÉQUENCE: EXEMPLE 2

Le DAB (Distributeur Automatique de Billet)

- Un DAB permet à tout détenteur de carte bancaire de retirer de l'argent.
- Si le détenteur de carte est un client de la banque propriétaire du DAB, il peut en plus consulter les soldes de ses comptes et effectuer des virements entre ces différents comptes.
- Les transactions sont sécurisées c'est-à-dire :
 - ☐ Le DAB consulte le Système d'Information de la banque (S.I. Banque) pour les opérations que désire effectuer un client de la banque (retraits, consultation soldes et virements).
 - ☐ Le DAB consulte le Système d'Autorisation Globale Carte Bancaire (Sys. Auto.) pour les retraits des porteurs de cartes non clients de la banque.
- Le DAB nécessite des opérations de maintenance tel que la recharge en billet, la récupération des cartes avalées, etc.

DIAGRAMME DE SÉQUENCE: EXEMPLE 2

Description textuelle du cas d'utilisation : « **Retirer de l'argent** »

Partie 1 : Description.

- Titre : Retirer de l'argent.
- Résumé : Ce cas d'utilisation permet aux possesseurs de carte bancaire de retirer de l'argent.
- Acteur principale : Un porteur de carte bancaire.
- Acteurs secondaires : Le Système d'Information de la banque et le Système d'Autorisation Globale Carte Bancaire.
- Date : 11/01/2013
- Responsable : E. REMY - Version : 1.0

DIAGRAMME DE SÉQUENCE: EXEMPLE 2

Description textuelle du cas d'utilisation : « Retirer de l'argent »

Partie 2 : Description des scénarios.

▪ Pré-conditions :

- Le DAB contient des billets.
- Les connexions avec le Système d'Autorisation et le Système d'information de la banque sont opérationnelles.

▪ Scénario nominale :

- 1) Le Porteur de carte introduit sa carte dans le DAB.
- 2) Le DAB vérifie que la carte introduite est bien une carte bancaire.
- 3) Le DAB demande le code de la carte au Porteur de carte.
- 4) Le Porteur de carte saisit son code.
- 5) Le DAB compare ce code avec celui qui est codé sur la carte.
- 6) Le DAB demande une autorisation au Système Globale d'autorisation.
- 7) Le Système d'Autorisation globale donne son accord et indique le crédit hebdomadaire.

▪ Scénario nominale (suite) :

- 8) Le DAB demande le montant désiré au Porteur de carte.
- 9) Le Porteur de carte saisit le montant.
- 10) Le DAB vérifie si le montant demandé est inférieur ou égale au crédit hebdomadaire.
- 11) Le DAB rend la carte et demande au Porteur de carte de la retirer.
- 12) Le Porteur de carte reprend sa carte.
- 13) Le DAB demande au Porteur de carte s'il désire un ticket.
- 14) Le Porteur de carte accepte le ticket.
- 15) Le DAB délivre le ticket et les billets.
- 16) Le Porteur de carte prend les billets et le ticket.

DIAGRAMME DE SÉQUENCE: EXEMPLE 2

Description textuelle du cas d'utilisation : « Retirer de l'argent »

▪ Scénario nominale (suite) :

- 8) Le DAB demande le montant désiré au Porteur de carte.
- 9) Le Porteur de carte saisit le montant.
- 10) Le DAB vérifie si le montant demandé est inférieur ou égale au crédit hebdomadaire.
- 11) Le DAB rend la carte et demande au Porteur de carte de la retirer.
- 12) Le Porteur de carte reprend sa carte.
- 13) Le DAB demande au Porteur de carte s'il désire un ticket.
- 14) Le Porteur de carte accepte le ticket.
- 15) Le DAB délivre le ticket et les billets.
- 16) Le Porteur de carte prend les billets et le ticket.

DIAGRAMME DE SÉQUENCE: EXEMPLE 2

Description textuelle du cas d'utilisation : « Retirer de l'argent »

▪ Scénarios alternatifs :

- Scénario alternatif SA1: Le code est erroné pour la première ou la deuxième fois.

SA1 commence au point 5 du scénario nominale.

Le DAB indique que le code est erroné.

Le DAB enregistre l'échec.

Le scénario reprend au point 3 du scénario nominal.

- Scénario alternatif SA2: Le montant demandé est trop élevé.

SA2 commence au point 10 du scénario nominale.

Le DAB affiche le montant max et demande au Porteur de carte de ressaisir un montant.

Le scénario reprend au point 9 du scénario nominal.

DIAGRAMME DE SÉQUENCE: EXEMPLE 2

Description textuelle du cas d'utilisation : « Retirer de l'argent »

▪ Scénarios alternatifs :

- Scénario alternatif SA3: Le ticket est refusé.

SA3 commence au point 13 du scénario nominale.

14) L'utilisateur refuse le ticket.

15) Le DAB délivre les billets.

16) L'utilisateur prend les billets.

- Scénario alternatif SA4: Le porteur de carte est client de la banque.

SA4 commence au point 7 du scénario nominale.

Le DAB demande une autorisation auprès du Système d'Information de la banque.

Le scénario reprend au point 9 du scénario nominal.

DIAGRAMME DE SÉQUENCE: EXEMPLE 2

Description textuelle du cas d'utilisation : « Retirer de l'argent »

■ Scénarios d'exception:

- Scénario d'exception SE1: Carte non valide.

SE1 commence au point 2 du scénario nominal.

Le DAB Indique que la carte n'est pas valide restitue la carte et met fin au cas.

- Scénario d'exception SE2: Le code est erroné pour la troisième fois.

SE2 commence au point 5 du scénario nominal.

Le DAB Indique que le code est erroné pour la troisième fois, confisque la carte et met fin au cas.

- Scénario d'exception SE3: Retrait non autorisé.

SE3 commence au point 6 du scénario nominal.

Le DAB Indique que tout retrait est impossible, restitue la carte et met fin au cas.

DIAGRAMME DE SÉQUENCE: EXEMPLE 2

Description textuelle du cas d'utilisation : « Retirer de l'argent »

▪ **Post-conditions:**

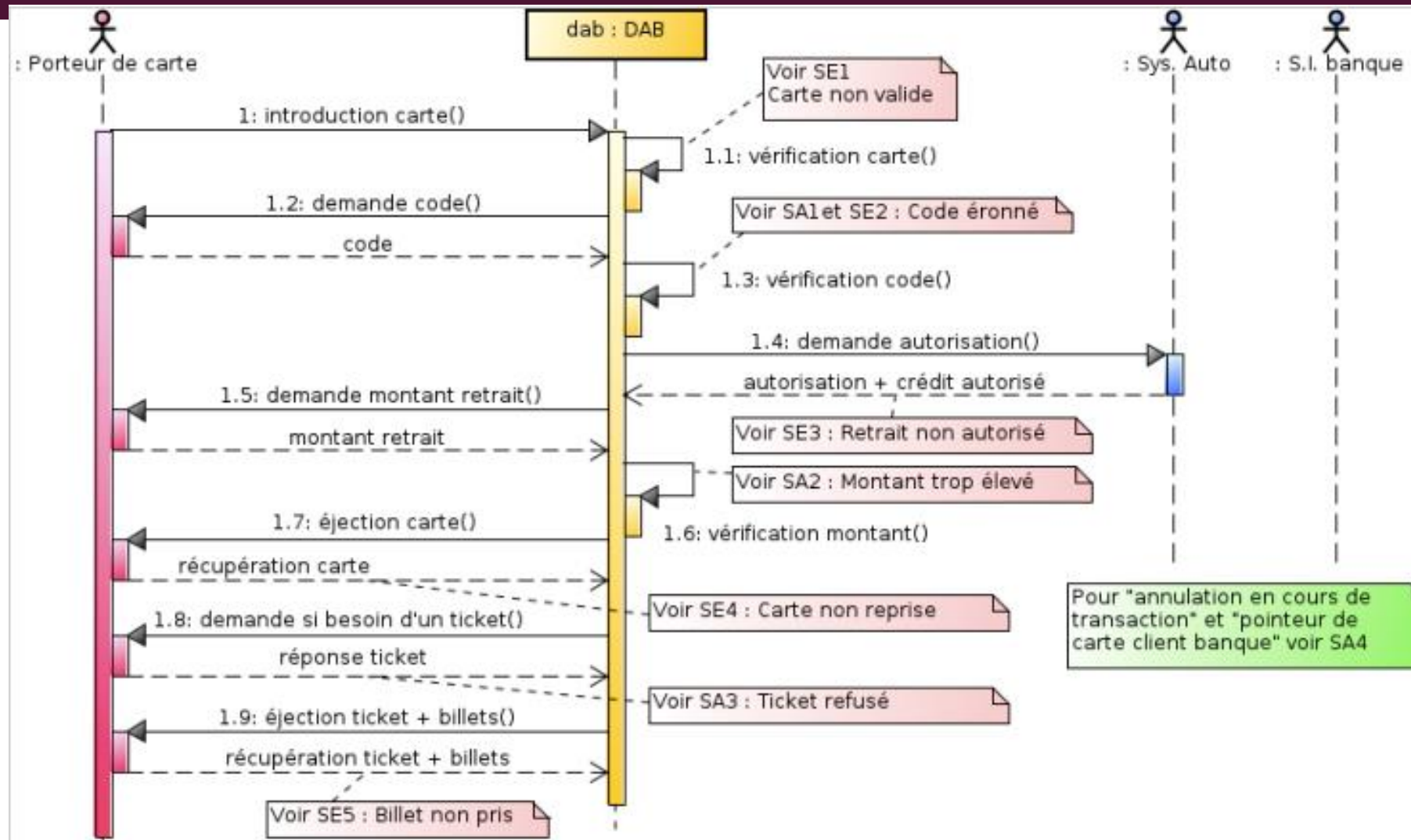
Les détails de la transaction doivent être enregistrés (montant, numéro carte, date...) aussi bien en cas de succès que d'échec.

▪ **Partie 3 : Exigences non fonctionnelles**

- La saisie du code confidentiel ne doit pas faire apparaître le code à l'écran.
- Le compte du client ne doit pas être débité tant que le billets n'ont pas été distribués.

La figure dans la diapositive suivante représente le diagramme de séquence système enrichi du cas d'utilisation « Retirer de l'argent »

DIAGRAMME DE SÉQUENCE: EXEMPLE 2



EXERCICE I:

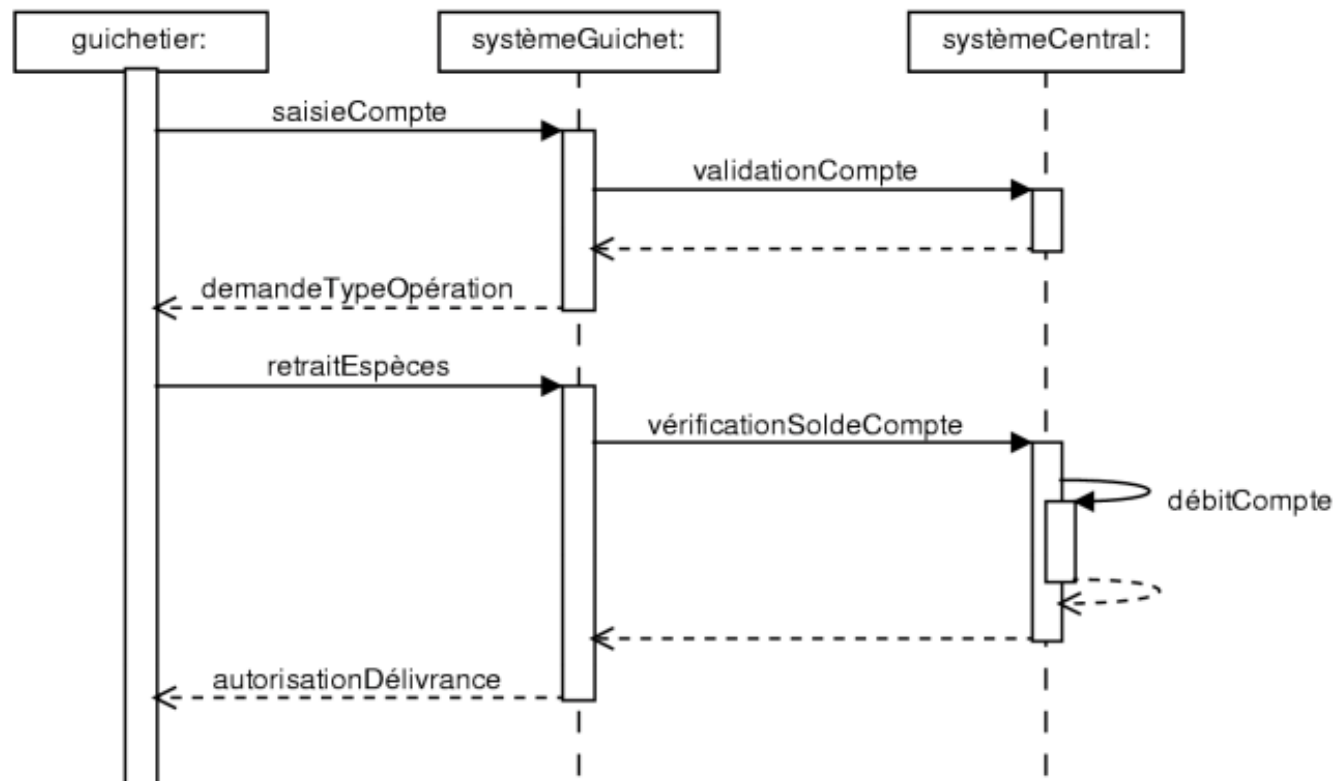
Description textuelle du cas d'utilisation : « Retirer de l'espèce »

- La rubrique « enchaînement nominal » du cas d'utilisation « retrait d'espèces » contient les éléments suivants :
- 1. Le guichetier saisit le numéro de compte du client ;
- 2. L'application valide le compte auprès du système central ;
- 3. Le guichetier demande un retrait de 100 euros ;
- 4. Le système « guichet » interroge le système central pour s'assurer que le compte est suffisamment approvisionné ;
- 5. Le système central effectue le débit du compte ;
- 6. En retour, le système notifie au guichetier qu'il peut délivrer le montant demandé.

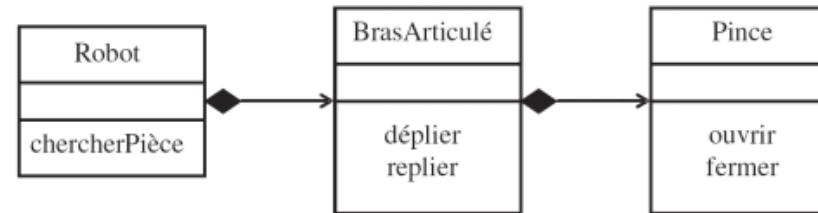
Question : Donner le diagramme de séquences associé à cette description textuelle.

SOLUTION

Diagramme de Séquence: « Retrait d'espèces »



EXERCICE 2: MODÉLISATION DE LA DYNAMIQUE DE L'INTERACTION ENTRE CLASSES



Le diagramme de classes ci-dessus décrit des robots composés de bras articulés, eux-mêmes composés de pinces. L'implémentation des classes en question est la suivante :

```
class Robot{
privée :
    BrasArticulé brasArticulé ;
publique :
    void chercherPièce() {
        brasArticulé.déplier() ;
        brasArticulé.replier() ;
    }
}
```

```
class BrasArticulé {
privée :
    Pince pince ;
publique :
    void déplier() {
        ...
        pince.fermer() ;
    }
    void replier() {
        pince.ouvrir() ;
    }
}
```

```
class Pince {
privée :
    ...
publique :
    void fermer() { ... }
    void ouvrir() { ... }
}

Début programme principal
    Robot robot ;
    robot.chercherPièce() ;
Fin programme principal
```

Question : Donnez un diagramme de séquence représentant l'échange de messages résultant de l'exécution du programme principal.

SOLUTION

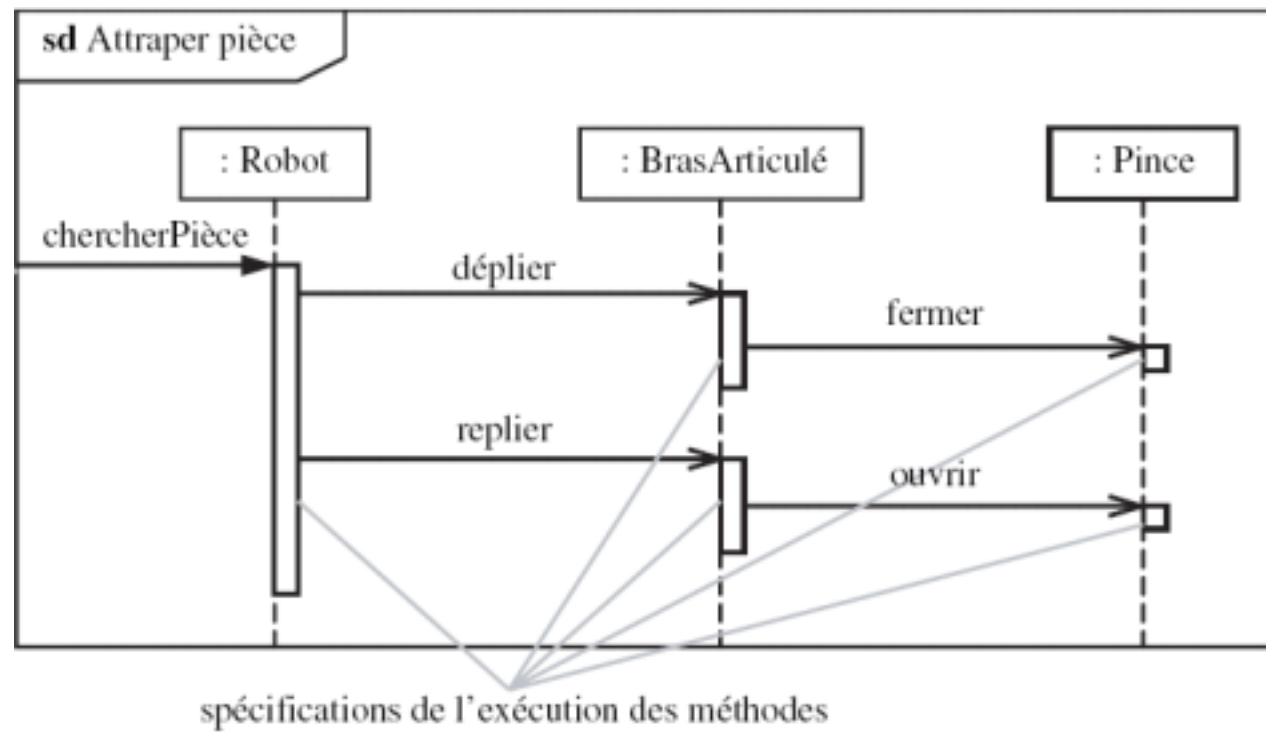


DIAGRAMME DE SÉQUENCE SYSTÈME

- **Diagramme de Séquence Système en boîte noire:**

Il est représenté en montrant l'acteur principal à gauche, puis un objet unique représentant le système en boîte noire, et, enfin, les éventuels acteurs secondaires sollicités durant le scénario à droite du système.

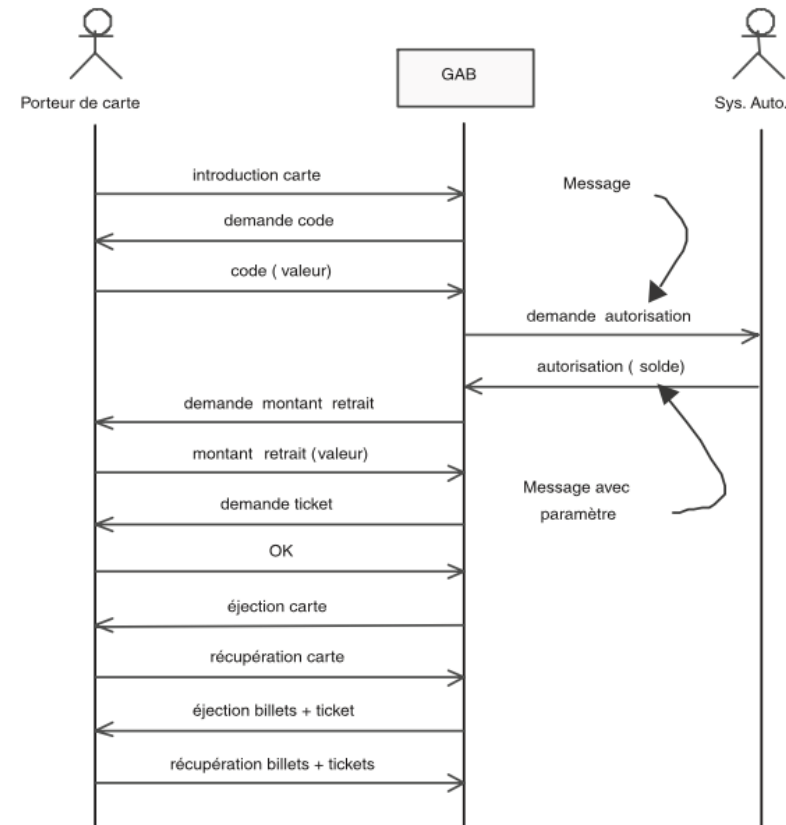
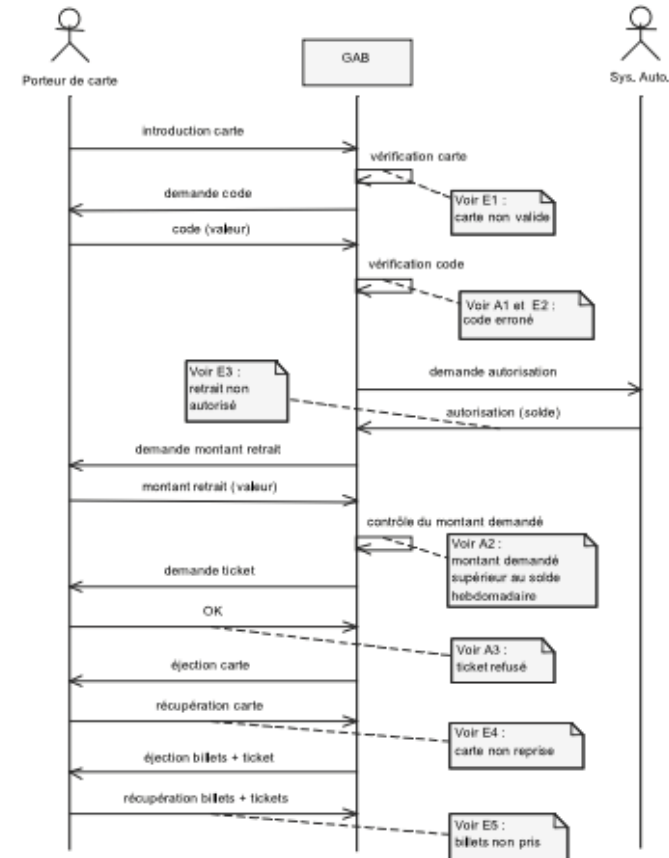


DIAGRAMME DE SÉQUENCE SYSTÈME

■ Diagramme de Séquence Système en boîte noire enrichi:

Une possibilité intéressante consiste à enrichir le diagramme de séquence système du scénario nominal pour faire apparaître également :

- les principales actions internes du système (au moyen de messages qu'il s'envoie à lui-même) ;
- les renvois aux enchainements alternatifs et d'erreur (au moyen de notes).

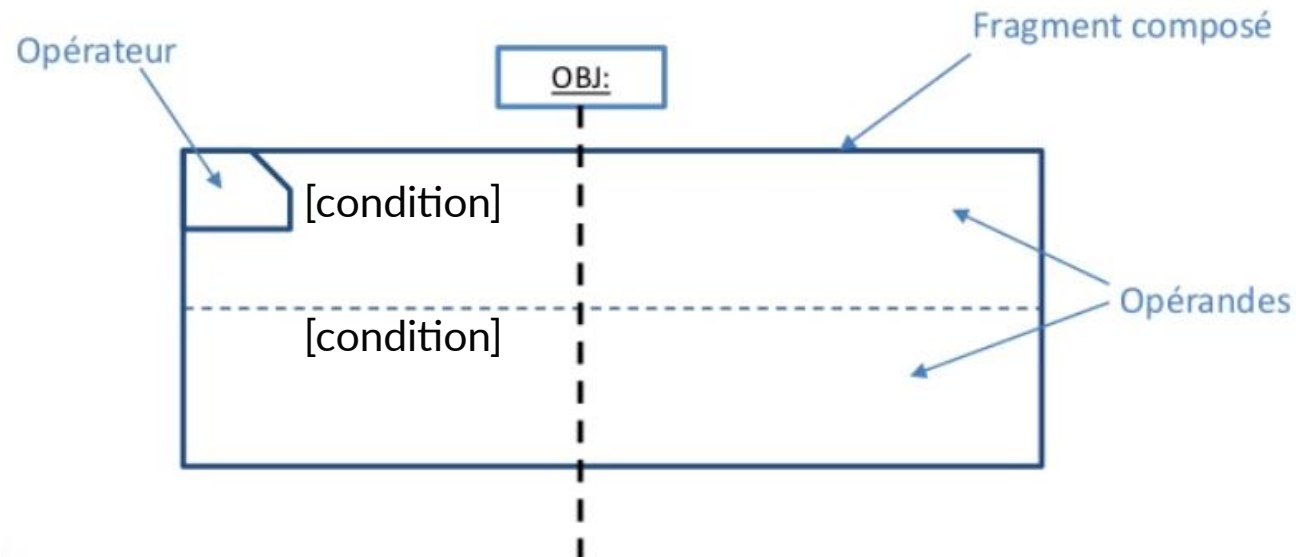




FRAGMENTS D'INTÉRACTIONS COMBINÉS

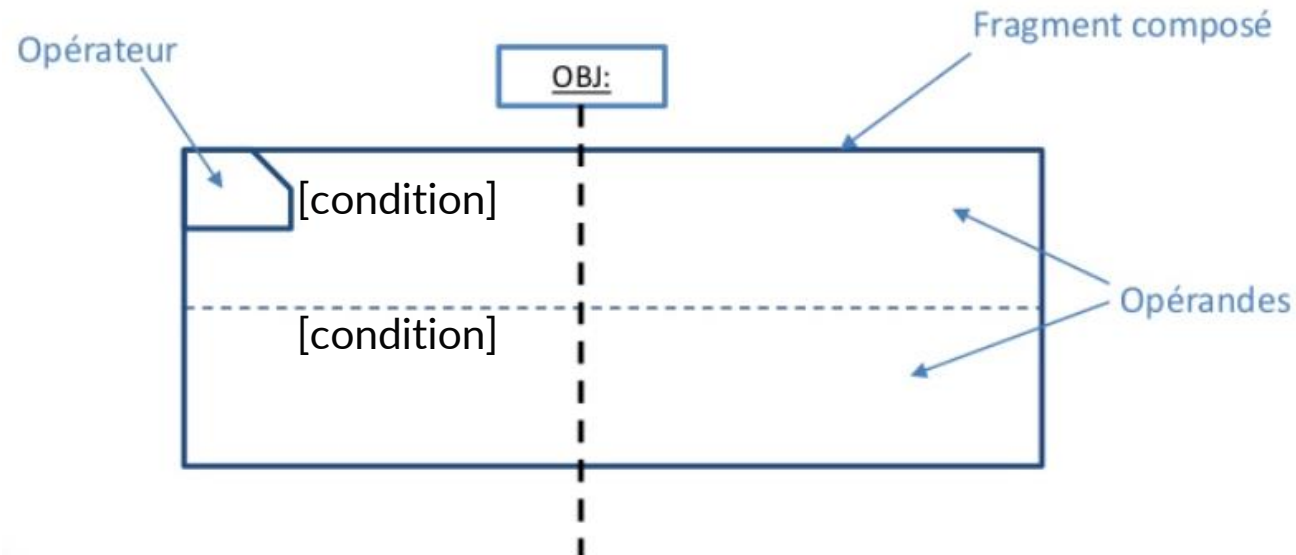
FRAGMENTS D'INTERACTIONS COMBINÉS

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.



FRAGMENTS D'INTERACTIONS COMBINÉS

- Les opérandes d'un opérateur d'interaction sont séparés par une ligne pointillée. Les conditions de choix des opérandes (éventuels) sont données par des expressions booléennes entre crochets ([]).



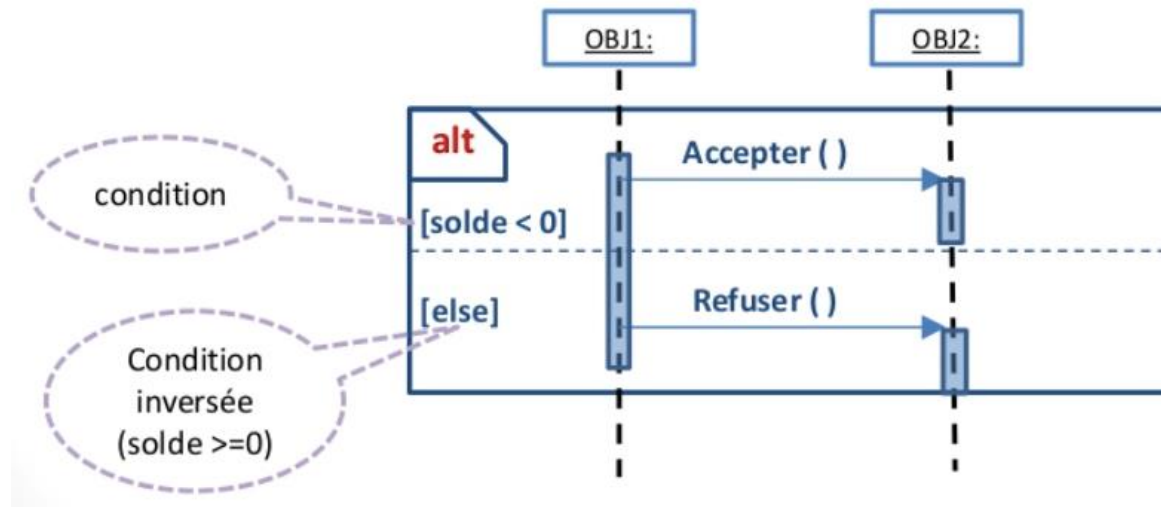
FRAGMENTS D'INTERACTIONS COMBINÉS

Il existe 13 opérateurs parmi lesquels on cite les opérateurs les plus utilisés:

- **alt** : différentes alternatives selon condition.
- **loop** : répète le fragment tant que condition.
- **par**: représente des interactions ayant lieu en parallèle
- **ref** : référence un autre diagramme de séquence.
- **opt** : fragment optionnel.
- **break**: Abandonner l'exécution du block si une condition est vérifiée.
- **critical**: représente une étape critique à réaliser.

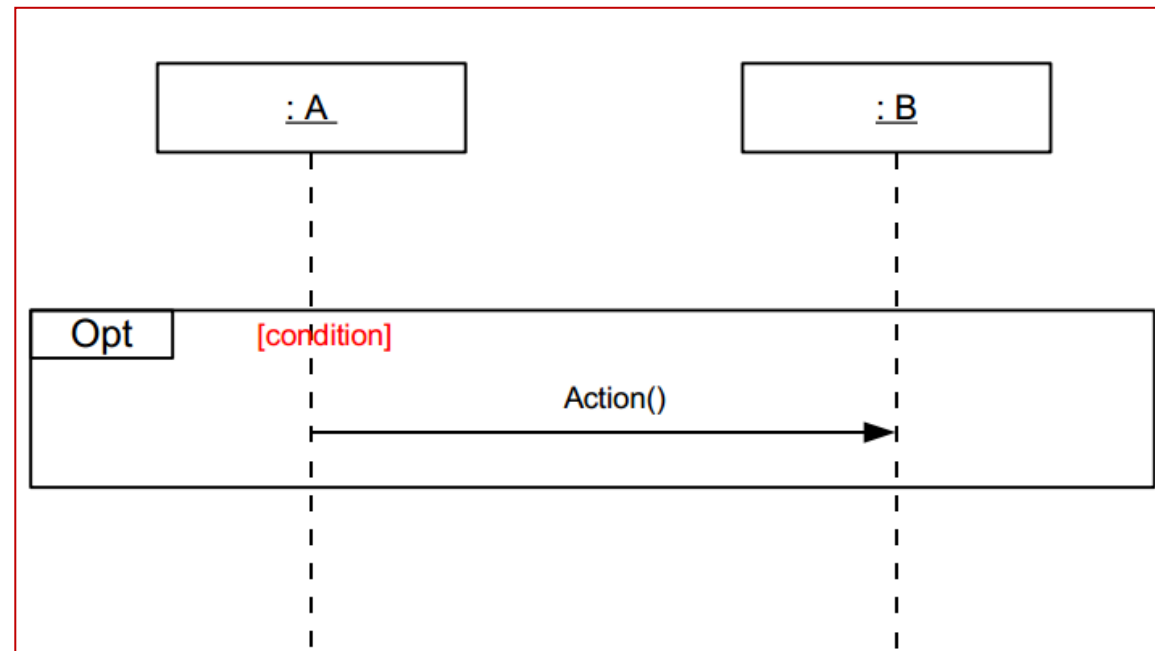
FRAGMENT D'INTERACTION AVEC OPÉRATEUR « ALT »

L'opérateur alternatif (alt) est un opérateur conditionnel possédant plusieurs opérandes séparés par des pointillés. C'est l'équivalent d'une exécution à choix multiples. Chaque opérande détient une condition de garde. Seul le sous-fragment dont la condition est vraie est exécuté. La condition else est exécutée si aucune autre condition n'est valide

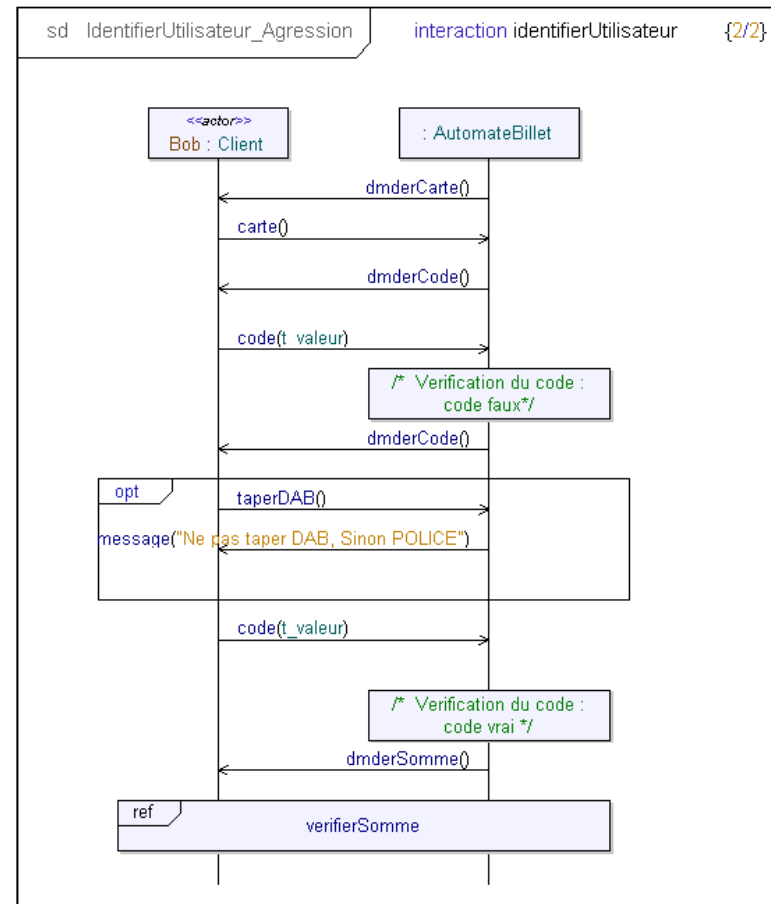


FRAGMENT D'INTERACTION AVEC OPÉRATEUR « OPT »

- Sous-séquence optionnelle exécutée si condition de garde est vraie. Elle est l'équivalent de **Si ...Alors**.



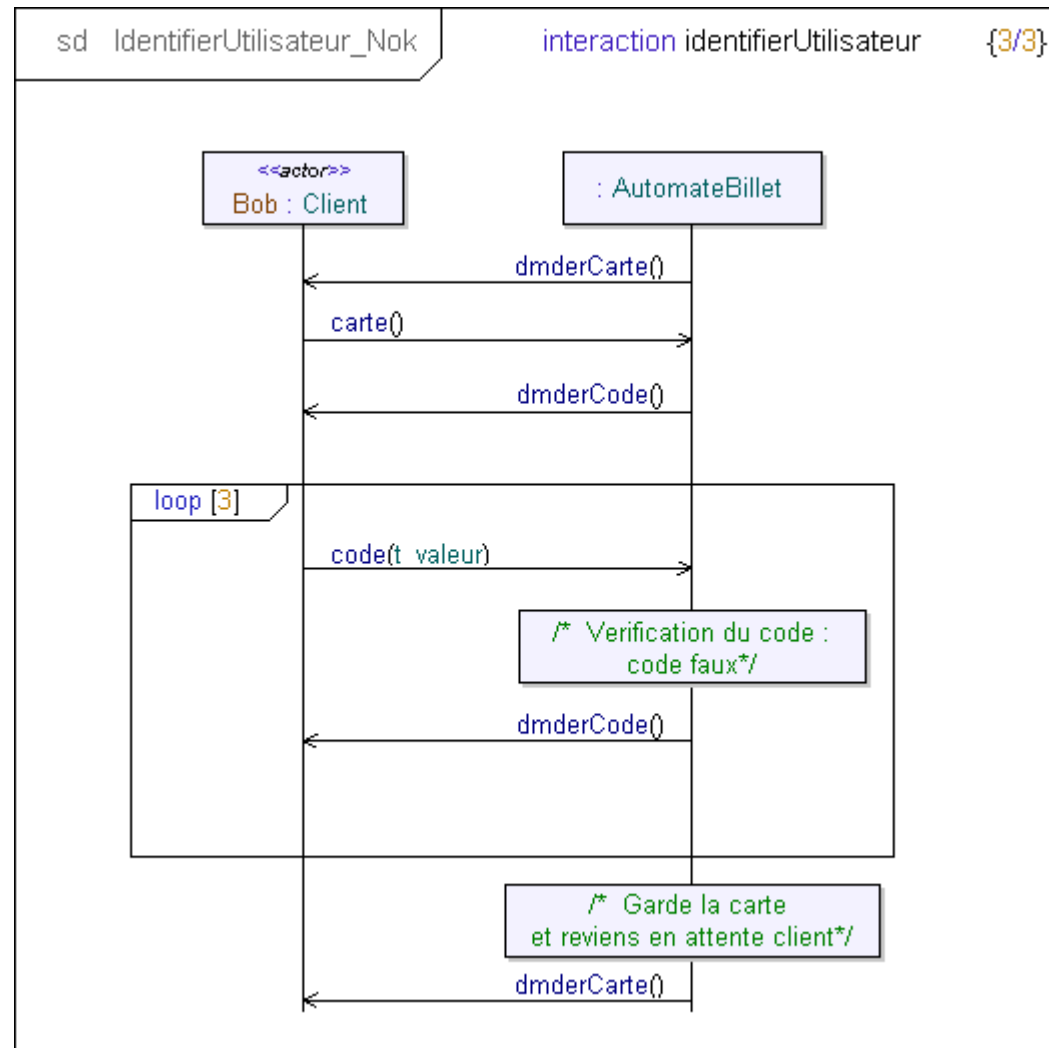
FRAGMENT D'INTERACTION AVEC OPÉRATEUR « OPT »



FRAGMENT D'INTERACTION AVEC OPÉRATEUR « LOOP »

Cet opérateur est utilisé pour décrire un ensemble d'interaction qui s'exécute en boucle. Un exemple de cet opérateur est représenté pour le CAPTCHA, une marque commerciale permettant de différencier de manière automatisée un utilisateur humain d'un ordinateur.

FRAGMENT D'INTERACTION AVEC OPÉRATEUR « LOOP »



FRAGMENT D'INTERACTION AVEC OPÉRATEUR « LOOP »

L'opérateur de boucle (loop) exécute une itérative dont la séquence qu'elle contient est exécutée tant que la garde qui lui est associée est vraie. - La garde s'écrit de la façon suivante :

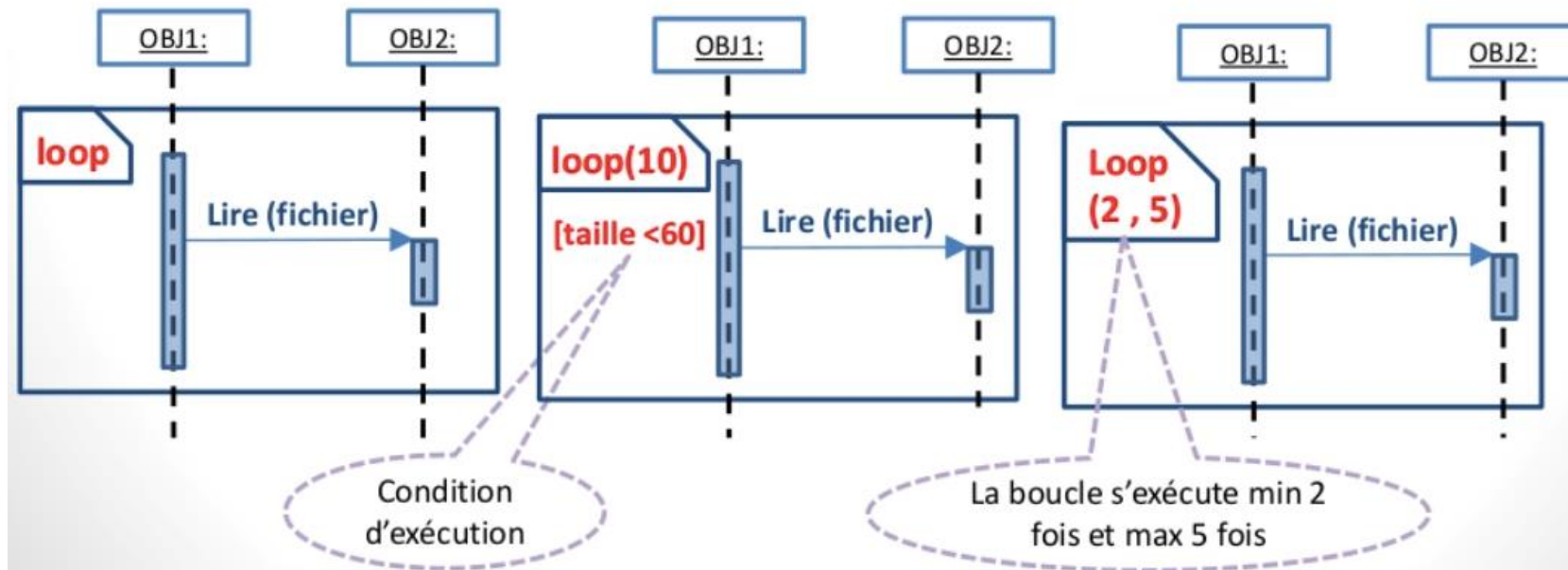
loop [min, max, condition] : Chaque paramètre (**min**, **max** et **condition**) est optionnel. Le contenu du cadre est exécuté **min** fois, puis continue à s'exécuter tant que la condition et que le nombre d'exécution de la boucle ne dépasse pas **max** fois.

▪ **Exemple de gardes :**

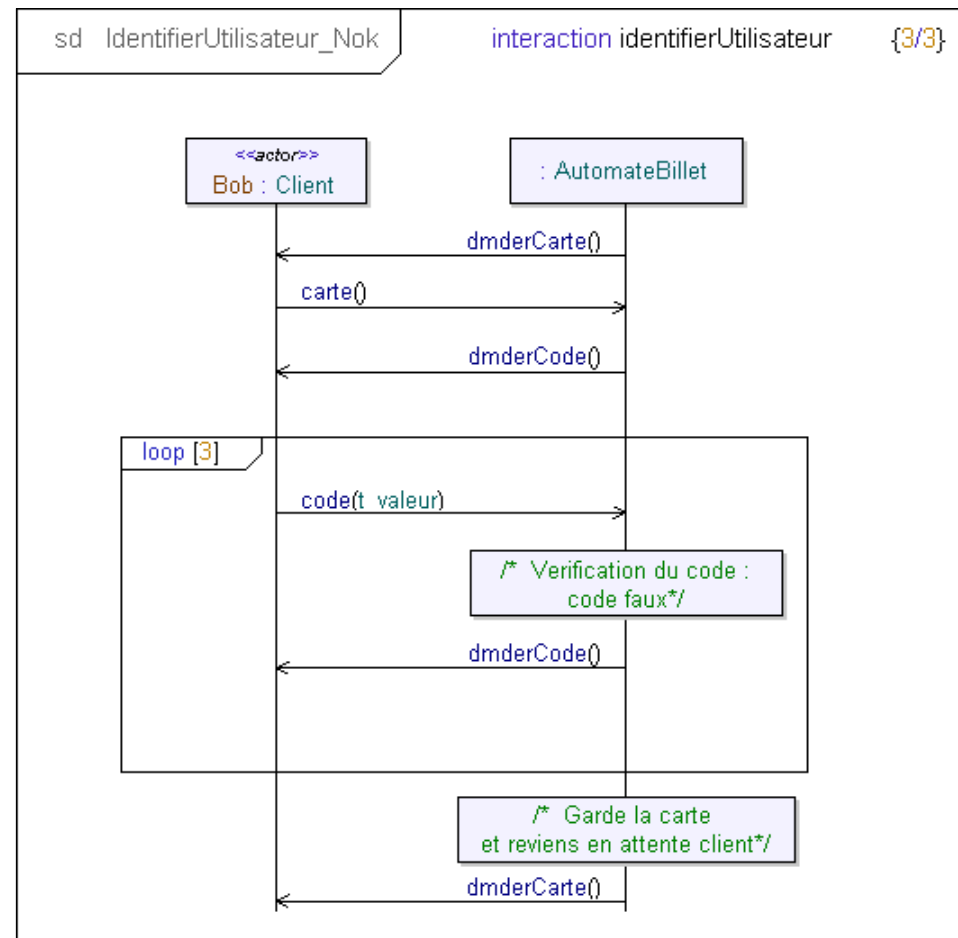
loop[3] → La séquence s'exécute 3 fois.

Loop[1, 3, code=faux] La séquence s'exécute 1 fois puis un maximum de 2 autres fois si code=faux.

FRAGMENT D'INTERACTION AVEC OPÉRATEUR « LOOP »

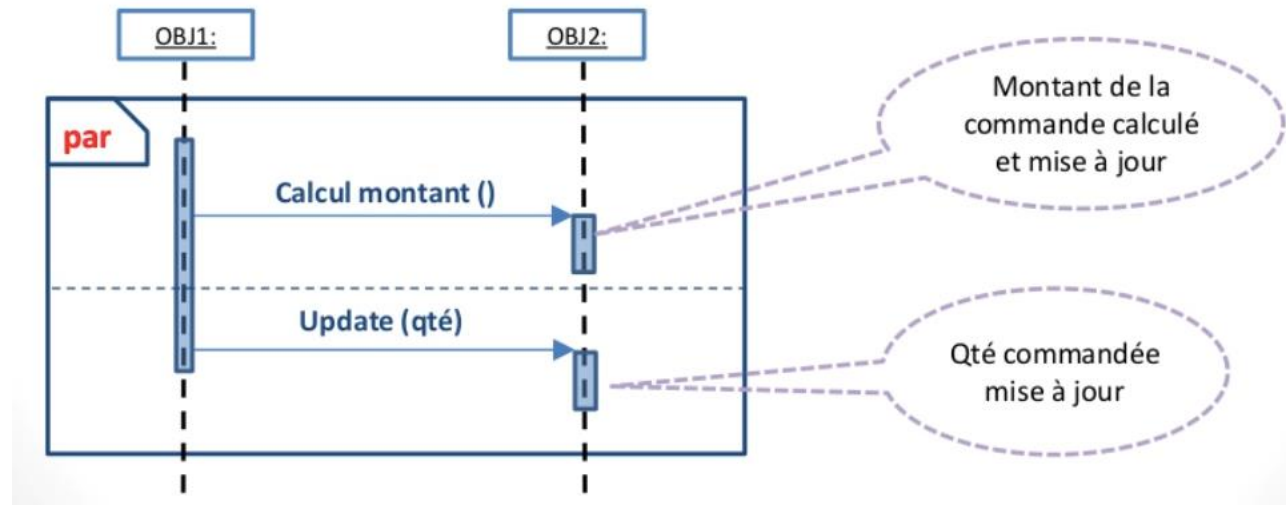


FRAGMENT D'INTERACTION AVEC OPÉRATEUR « LOOP »



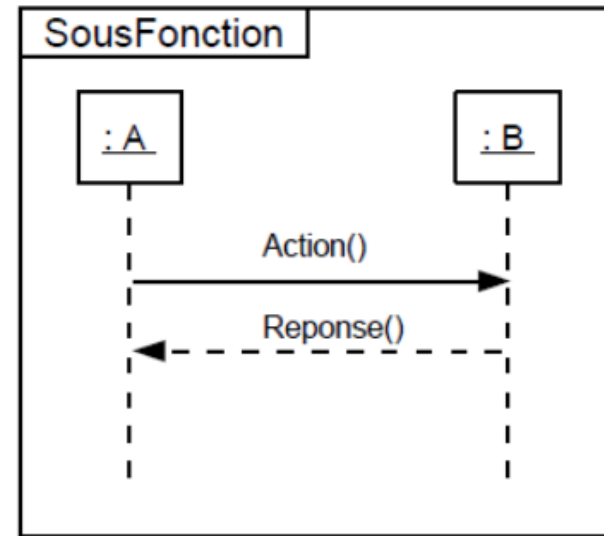
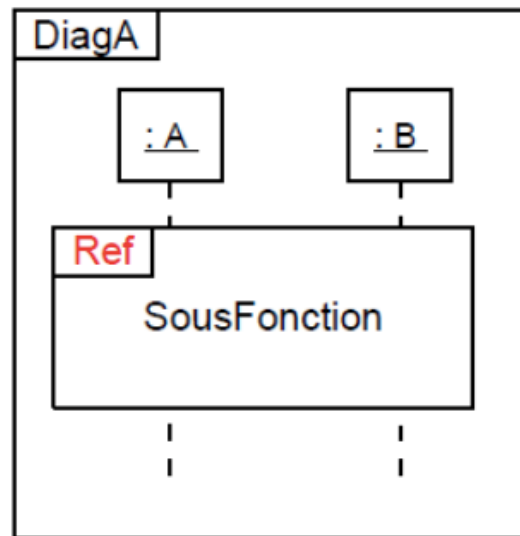
FRAGMENT D'INTERACTION AVEC OPÉRATEUR « PAR »

Il est utilisé pour représenter des interactions ayant lieu en parallèle.

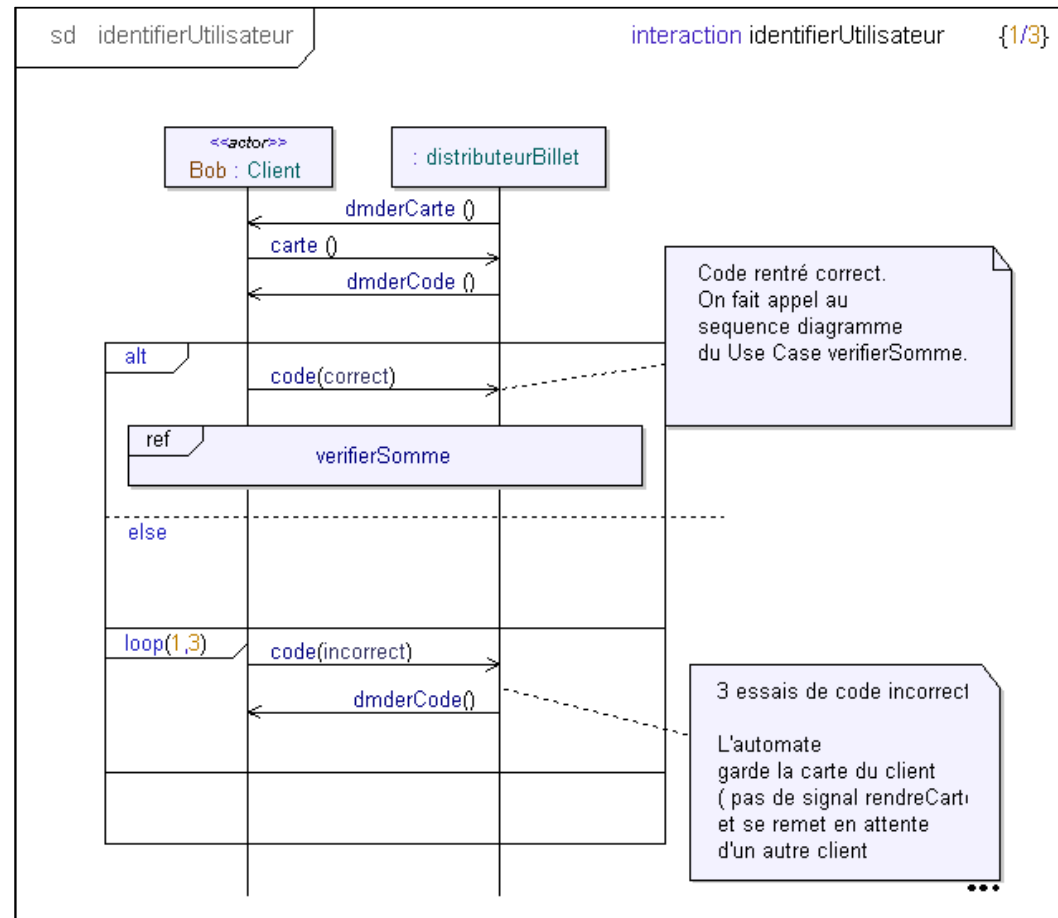


FRAGMENT D'INTERACTION AVEC OPÉRATEUR « REF »

- Un fragment ref permet d'inclure un autre diagramme de séquences. Il indique une référence vers un autre diagramme de séquence existant



FRAGMENT D'INTERACTION AVEC OPÉRATEUR « REF »



FRAGMENT D'INTERACTION AVEC OPÉRATEUR « REF »

- **Diagramme de Séquence Système en boîte noire enrichi:**

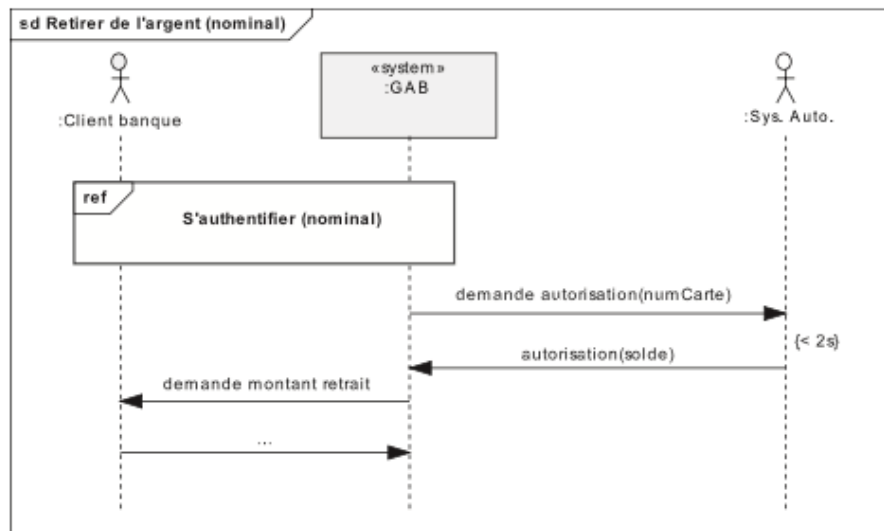


Figure 1: Nouveau diagramme de séquence système avec référence au cas inclus

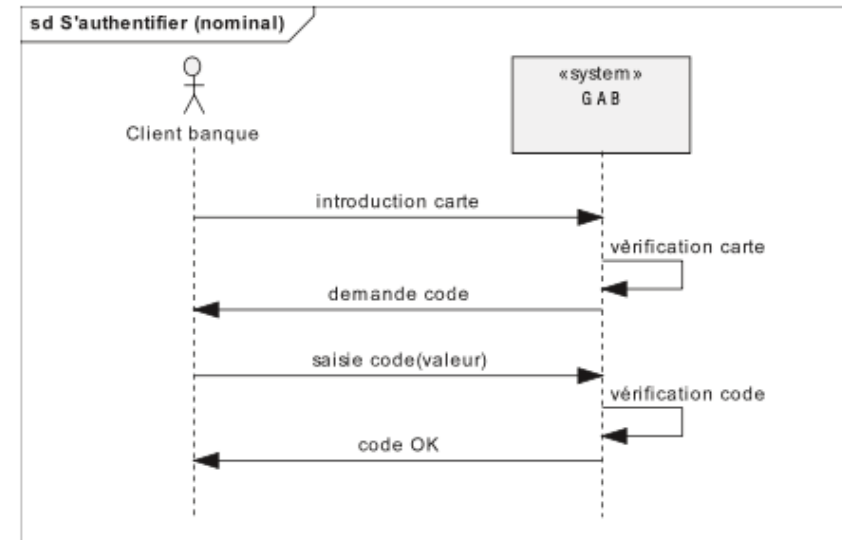
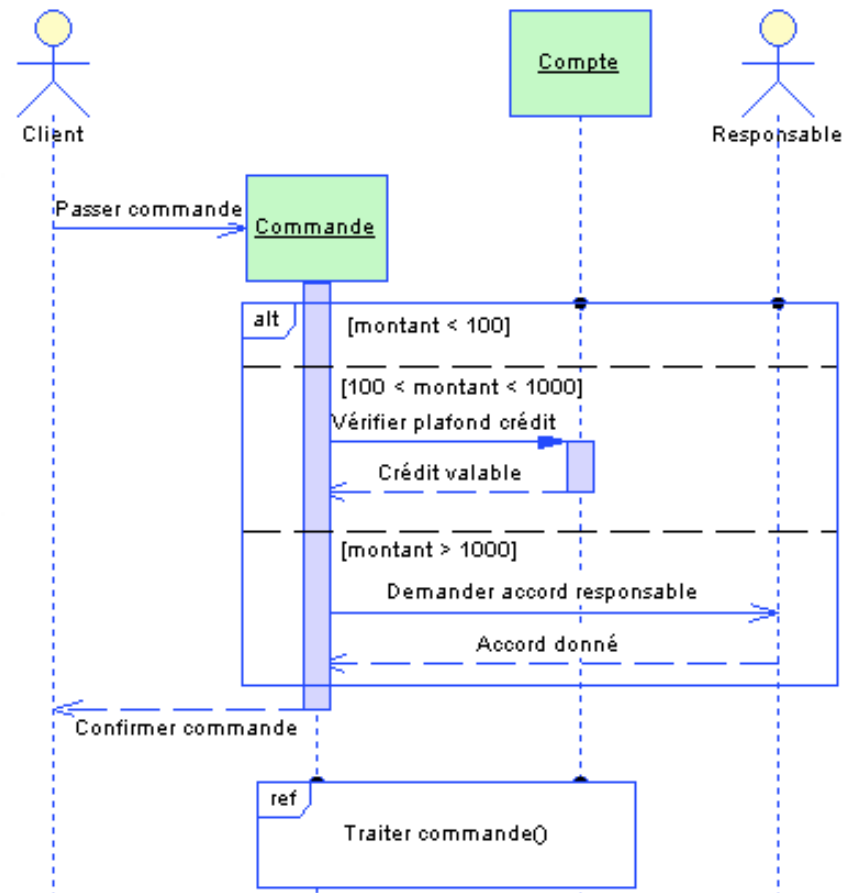


Figure 2: Diagramme de séquence système du fragment référencé

FRAGMENT D'INTERACTION AVEC OPÉRATEUR « REF »

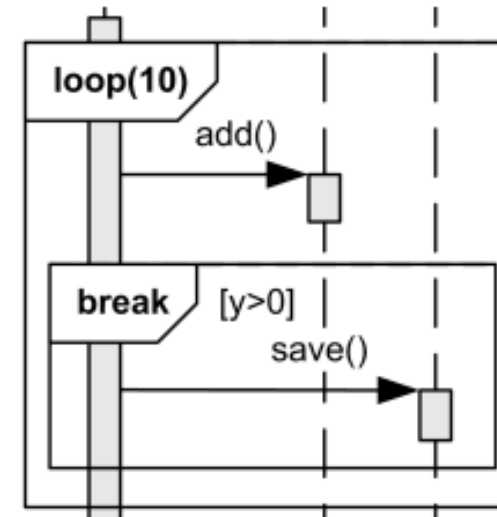
Exemple



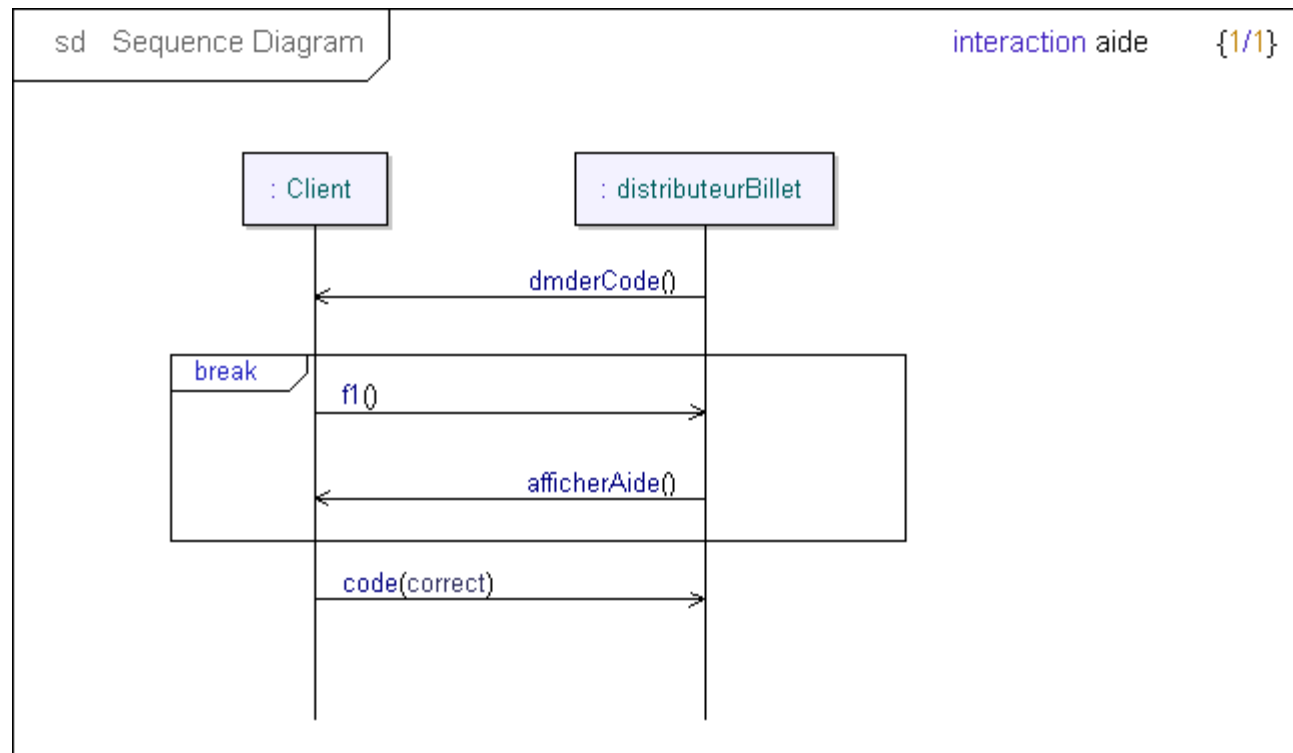
FRAGMENT D'INTERACTION AVEC OPÉRATEUR « BREAK »

Représente un scénario d'exception qui est exécuté à la place du reste du fragment d'interaction englobant. Un opérateur break avec une garde est choisi lorsque la garde est vraie. Dans ce cas, le reste du fragment d'interaction englobant directement est ignoré. Lorsque la garde de l'opérateur break est false, l'opérateur break est ignoré et le reste de l'interaction englobante le fragment se poursuit.

- Dans l'exemple, la clause break met fin à la clause loop lorsque $y > 0$.

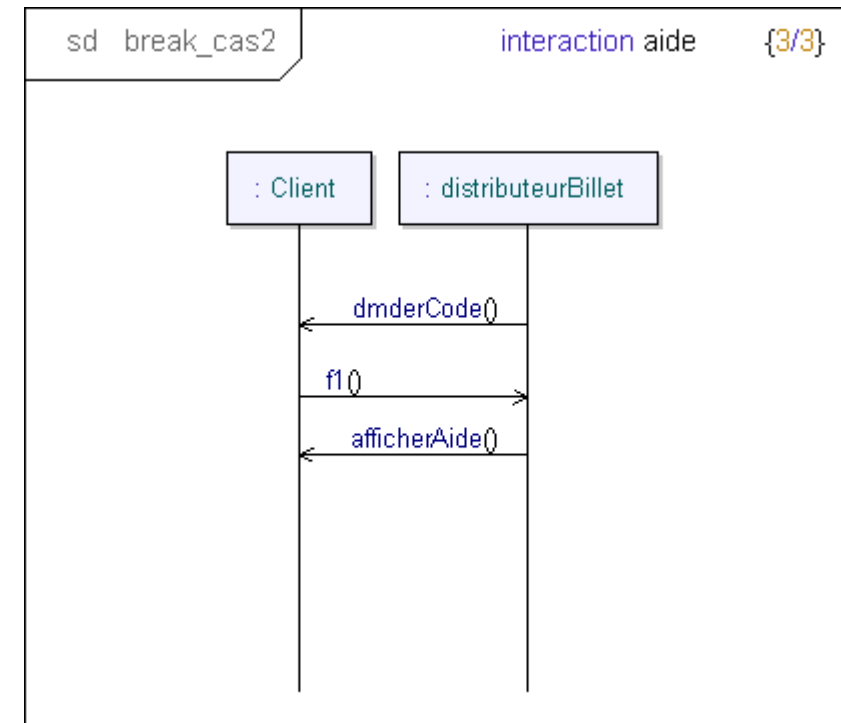
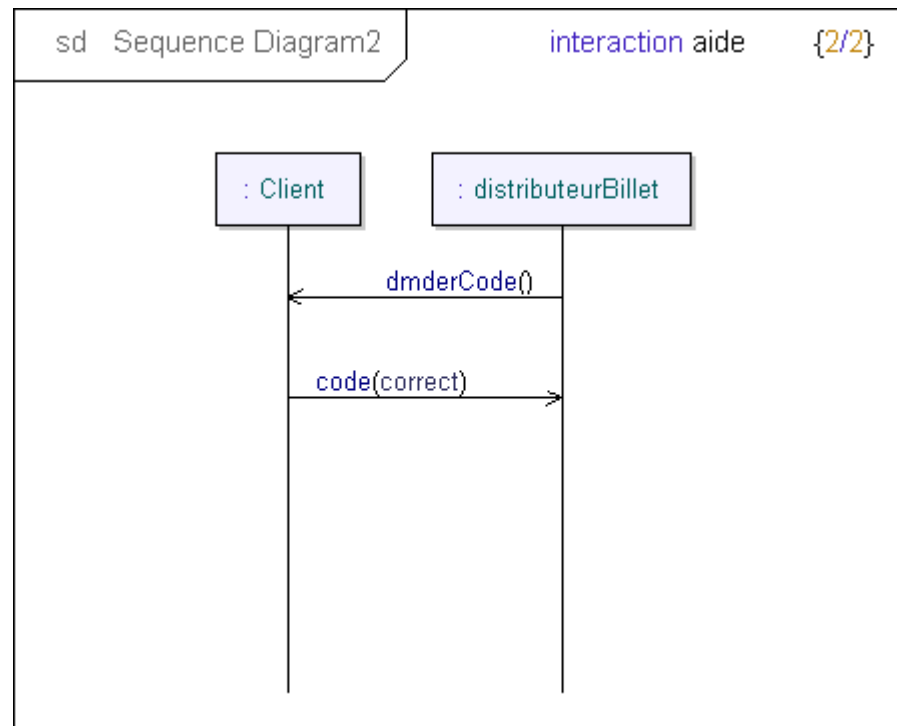


FRAGMENT D'INTERACTION AVEC OPÉRATEUR « BREAK »



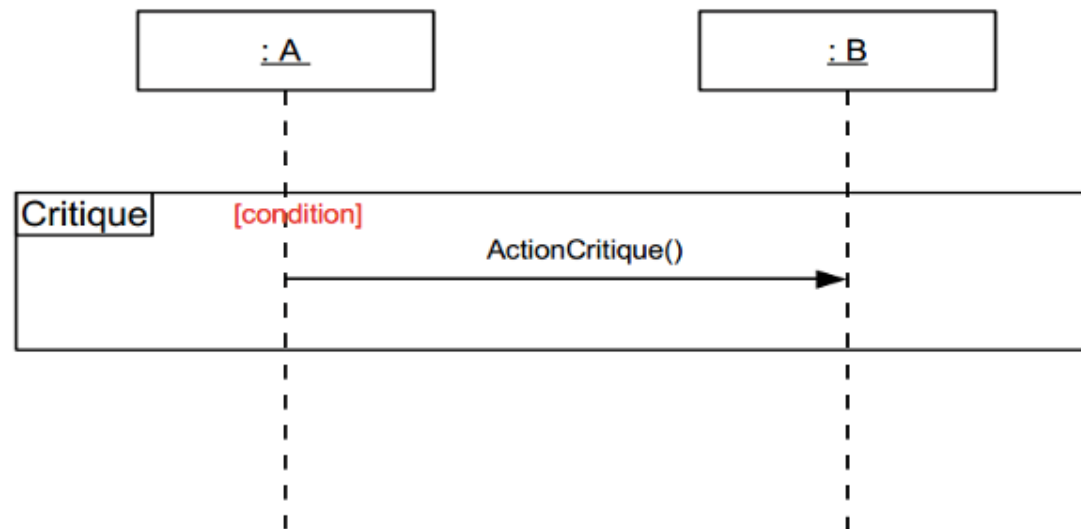
FRAGMENT D'INTERACTION AVEC OPÉRATEUR « BREAK »

L'équivalent de ce diagramme de séquence sans l'opérateur break correspond aux deux diagrammes de séquence ci-après :

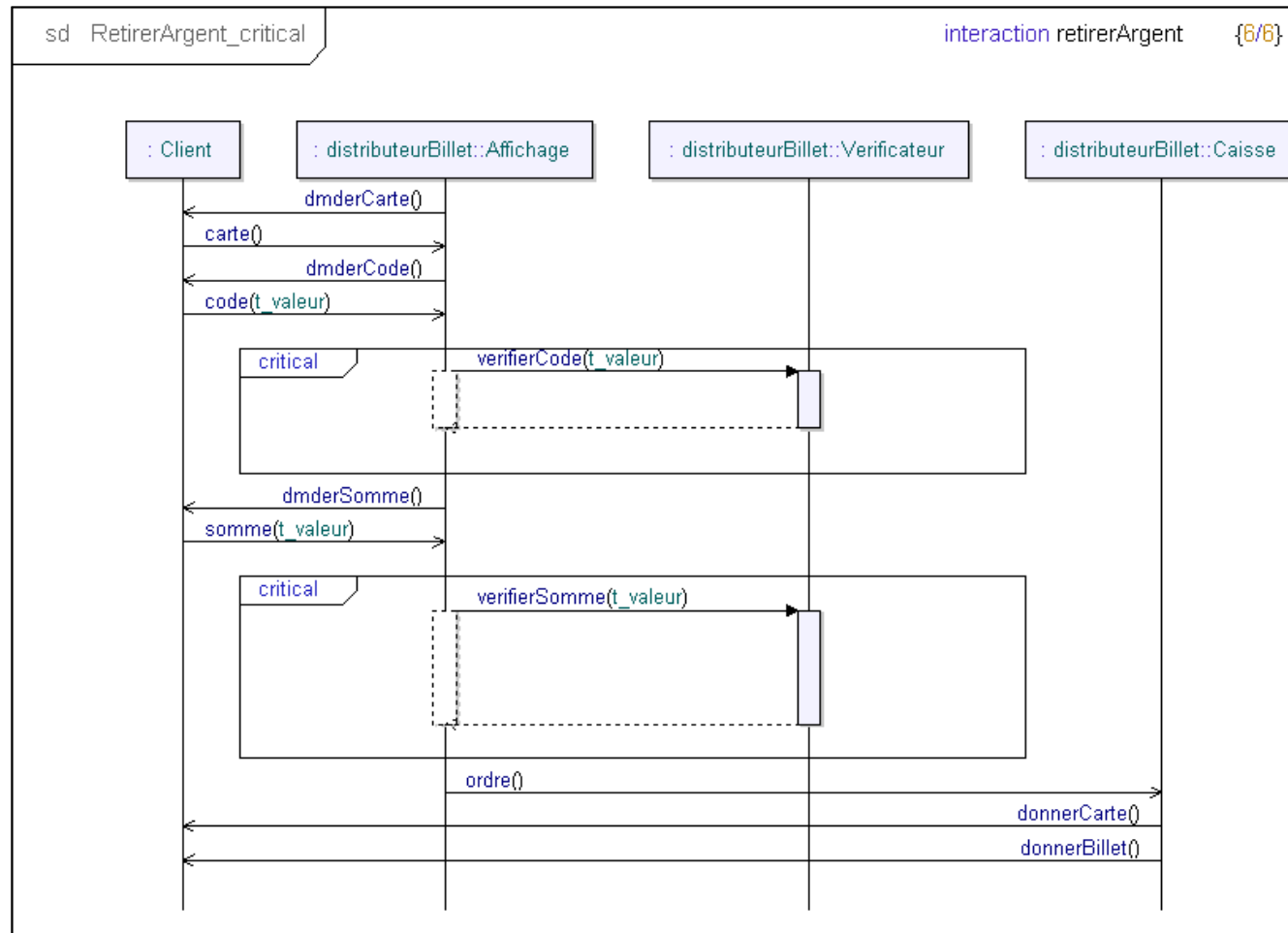


FRAGMENT D'INTERACTION AVEC OPÉRATEUR « CRITICAL »

L'opérateur d'interaction « critical » définit que le fragment combiné représente une région critique qui doit être exécutée sans interruption. Elle empêchera donc la réalisation des autres interactions jusqu'à ce qu'elle se termine .



FRAGMENT D'INTERACTION AVEC OPÉRATEUR « CRITICAL »



LES AUTRES OPÉRATEURS

- **"ignore"** Indique qu'il existe des messages qui ne sont pas présents dans le fragment combiné.
- **"consider"** Désigne les interactions à prendre en compte dans la séquence.
- **"assert"** (assertion) Indique que le fragment combiné est une assertion. Pour les fragments dont on connaît à l'avance les paramètres du message (exemple : après la saisie des 4 chiffres d'un code, la saisie suivante sera obligatoirement la touche « Entrée »).
- **"neg"** (negative) désigne un ensemble d'interactions invalides.
- **"seq"** (weak sequencing) notifie que les interactions qui s'opèrent entre des entités indépendantes n'ont pas d'ordre particulier.
- **"strict"** (strict sequencing) impose l'ordre décrit sur le diagramme.