

Université de Msila – Département  
d'Informatique

# Génie logiciel (GL)

(3ème année SI&ISIL)

Dr BOUNIF M-E

- 5. Le diagramme de séquence

Le diagramme de séquence décrit la dynamique du système. À moins de modéliser un très petit système, il est difficile de représenter toute la dynamique d'un système sur un seul diagramme. Aussi la dynamique globale sera représentée par un ensemble de diagrammes de séquence, chacun étant généralement lié à une sous-fonction du système.

Le diagramme de séquence décrit les interactions entre un groupe d'objets en montrant, de façon séquentielle, les envois de message qui interviennent entre les objets. Le diagramme peut également montrer les flux de données échangés lors des envois de message.

Pour interagir entre eux, les objets s'envoient des messages.

Lors de la réception d'un message, un objet devient actif et exécute la méthode de même nom. Un envoi de message est donc un appel de méthode.



Les envois de message sont représentés par des flèches horizontales reliant la ligne de vie de l'objet émetteur à la ligne de vie de l'objet destinataire (voir figure 5.2).  
Dans la figure 5.2, l'objet de gauche envoie un message à l'objet de droite. Ce message donne lieu à l'exécution de la méthode message de l'objet de droite, ce qui provoque son activation.

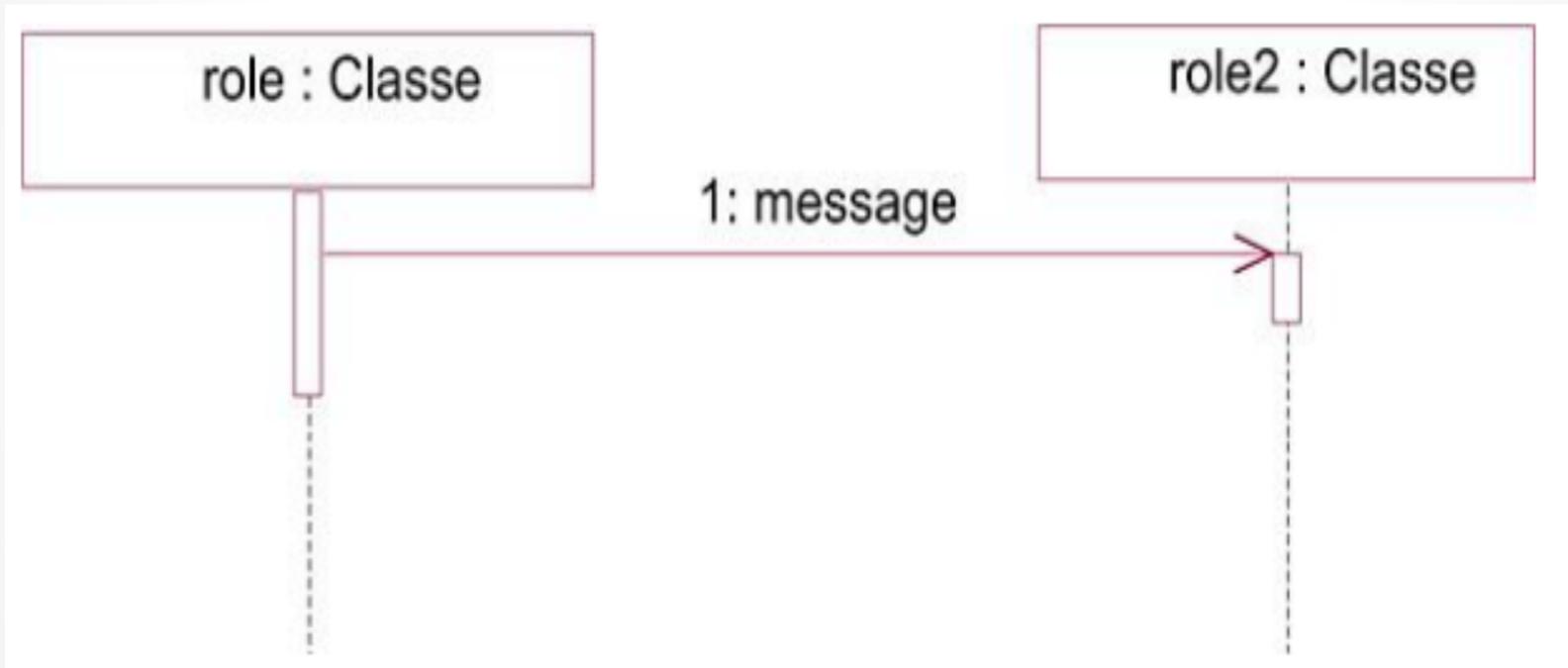


Figure 5.2 Envoi d'un message

## 5.2. La ligne de vie d'un objet

Comme il représente la dynamique du système, le diagramme de séquence fait entrer en action les instances des classes intervenant dans la réalisation de la sous-fonction qui lui est liée. À chaque instance est associée une ligne de vie qui montre ses actions et réactions, ainsi que les périodes pendant lesquelles elle est active, c'est à dire où elle exécute l'une de ses méthodes.

La représentation graphique de la ligne de vie est illustrée à la figure 5.1.

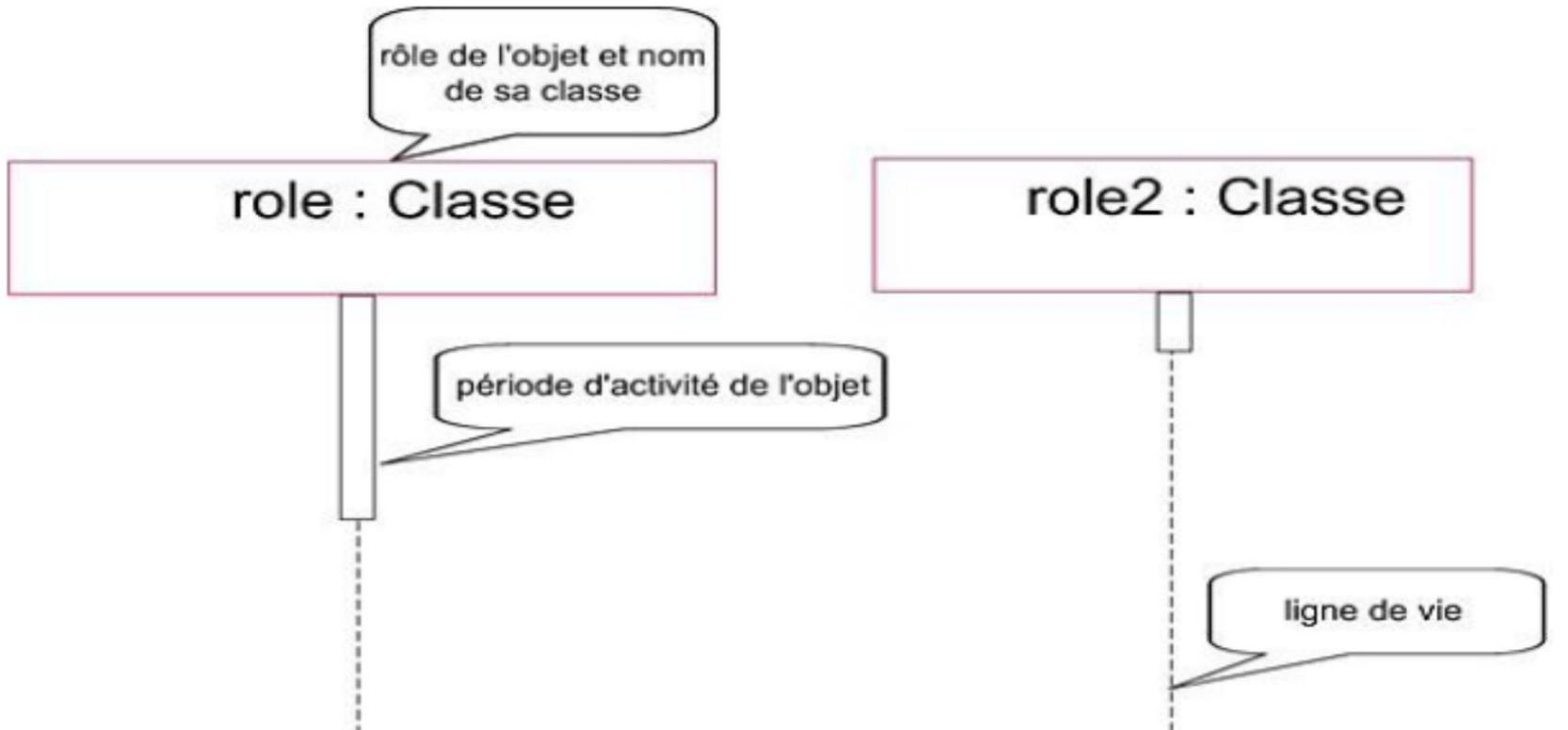


Figure 5.1 Lignes de vie

Les messages sont numérotés séquentiellement, à partir de un. Si un message est envoyé alors que le traitement du précédent n'est pas terminé, il est possible d'utiliser une numération composée (voir figure 5.3) où l'envoi du message 2 intervient pendant l'exécution du message 1.

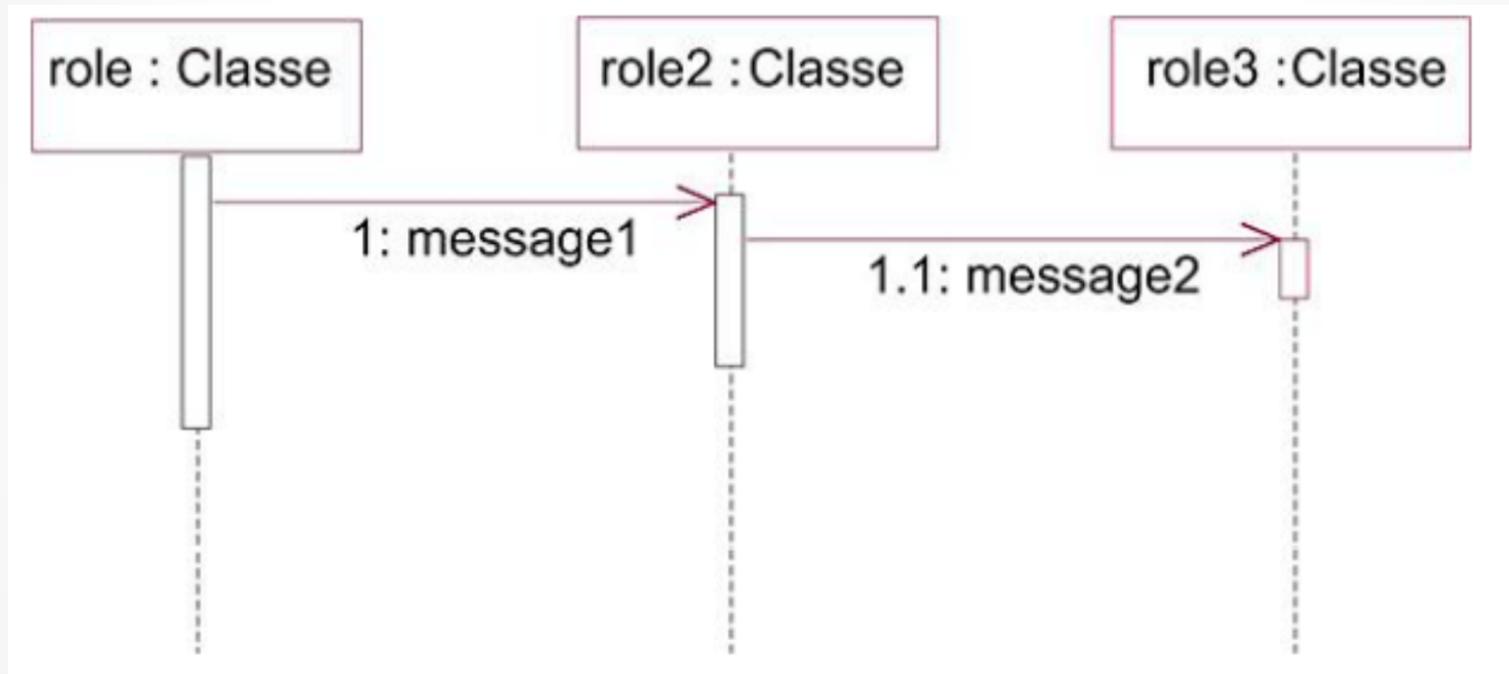


Figure 5.3 Numérotation des messages

La transmission d'information est également possible; elle est représentée par des paramètres transmis avec le message (voir figure 5.4).

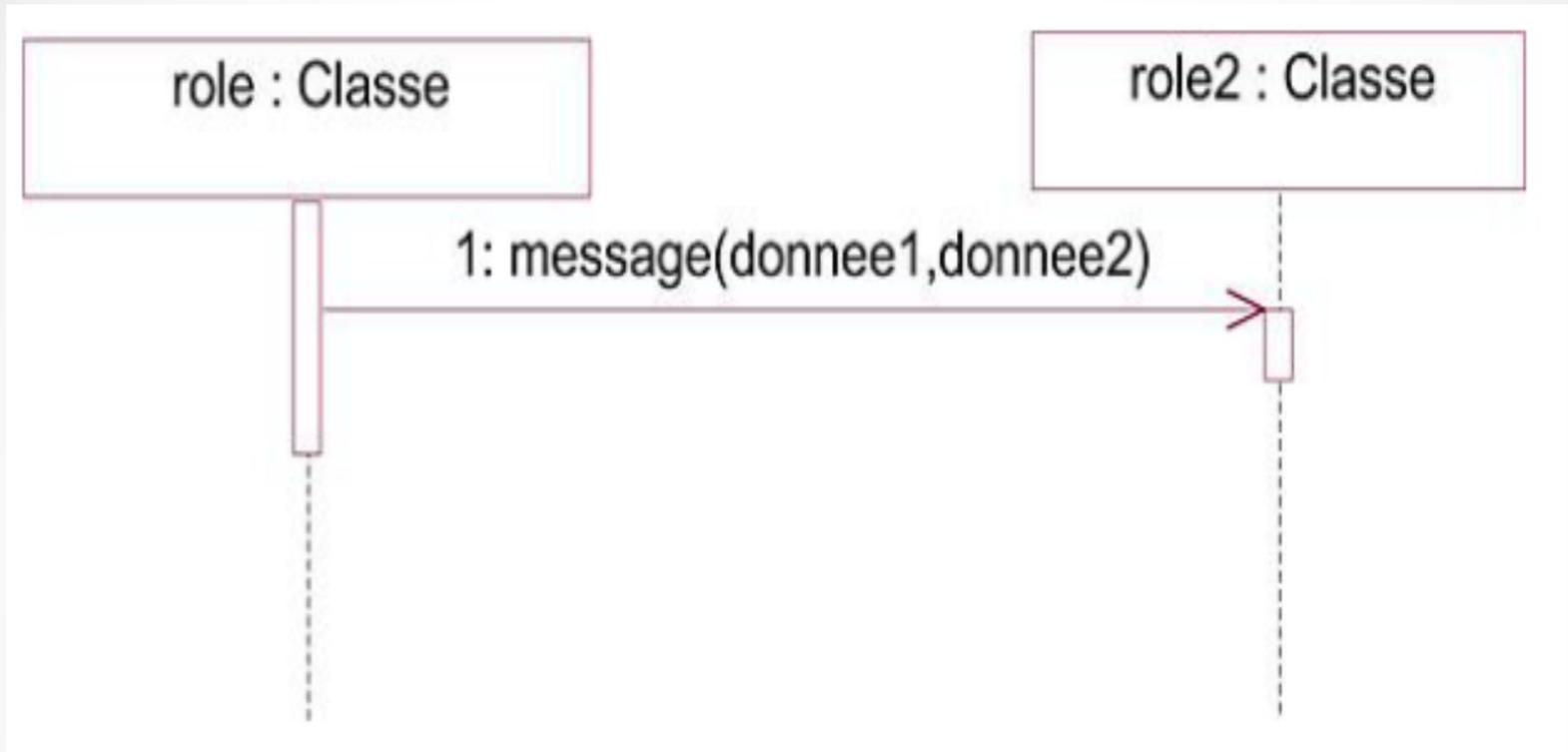


Figure 5.4 Transmission de données lors de l'envoi d'un message

Il existe différents types d'envois de message. La figure 5.5 en fournit une explication graphique.



Figure 5.5 Les différents types de messages

Le message synchrone est le plus fréquemment utilisé. Son emploi signifie que l'expéditeur du message attend que l'activation de la méthode invoquée chez le destinataire soit terminée avant de continuer son activité.

Dans le cas du message asynchrone, l'expéditeur n'attend pas la fin de l'activation de la méthode invoquée chez le destinataire. Ceci se produit lors de la modélisation d'un système où les objets peuvent fonctionner en parallèle (cas des systèmes multithread où les traitements sont effectués en parallèle).

Le message de retour de l'invocation d'une méthode n'est pas systématique, toutes les méthodes ne retournant pas un résultat. Un objet peut envoyer un message à lui-même. La représentation d'un tel message est illustrée à la figure 5.6.

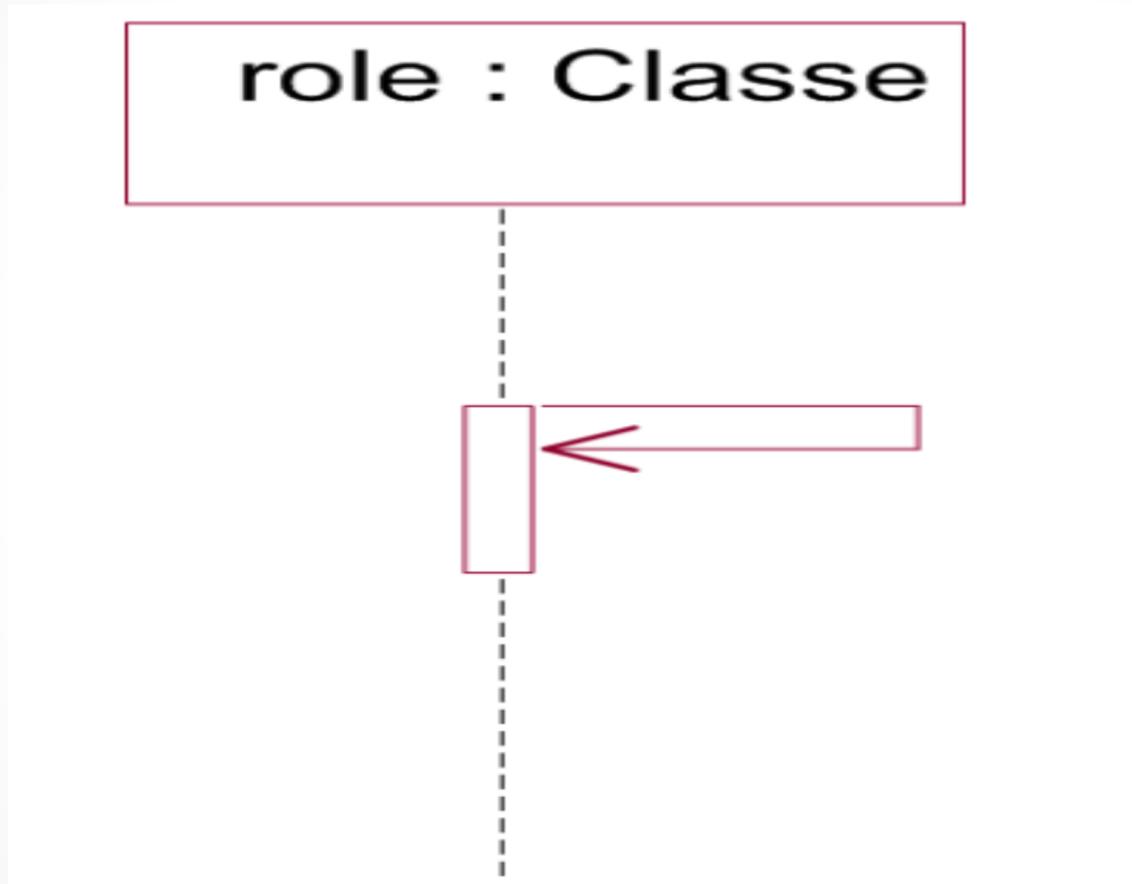


Figure 5.6 Envoi d'un message à soi-même

## La création et la destruction d'objets

Le diagramme de séquence décrivant la dynamique d'un système, celle-ci contient fréquemment des créations et des destructions d'objets.

La création d'objet est représentée par un message spécifique qui donne lieu au début de la ligne de vie du nouvel objet.

La destruction d'objet est un message envoyé à un objet existant et qui donne lieu à la fin de sa ligne de vie. Il est représenté par une croix.

Ces deux messages sont illustrés à la figure 5.7.

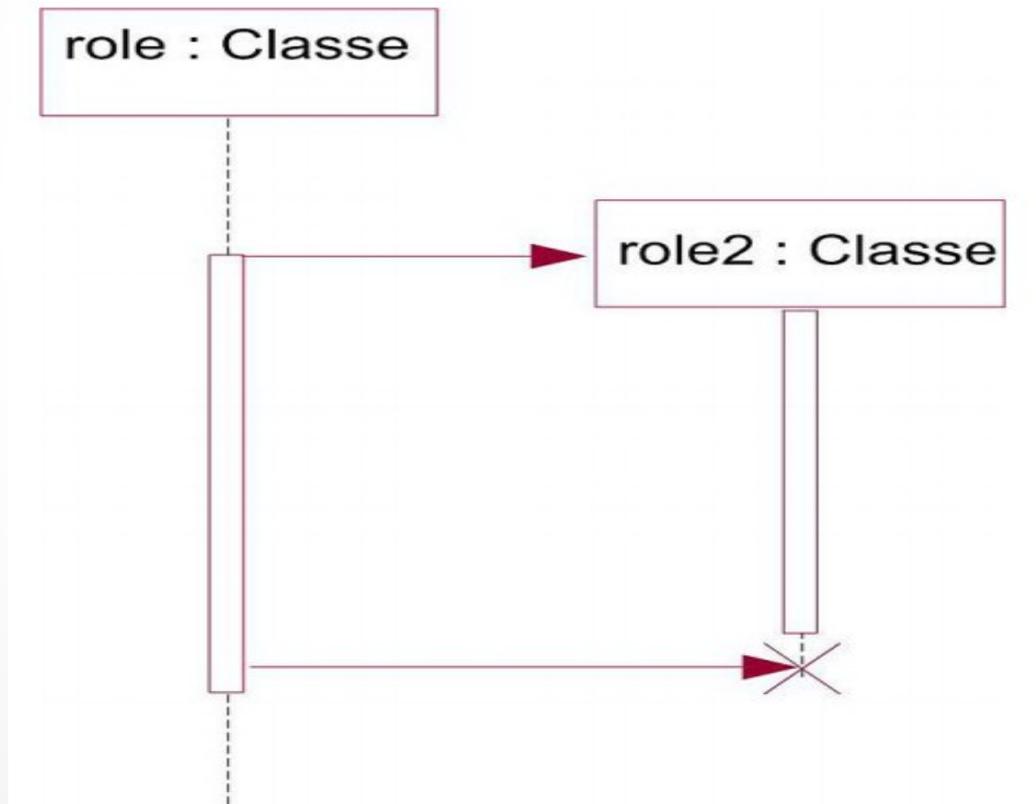


Figure 5.7 Messages de création et de destruction d'un objet

## **La description de la dynamique**

À partir des différents éléments introduits précédemment, il est maintenant possible de construire l'intégralité d'un diagramme de séquence et de décrire la dynamique d'un petit système ou une sous fonction d'un système plus important.

Exemple La figure 5.8 représente un scénario d'utilisation d'un DAB . Il n'y a aucune alternative; il s'agit donc bien d'un scénario.

Nous verrons par la suite comment introduire les alternatives et les boucles

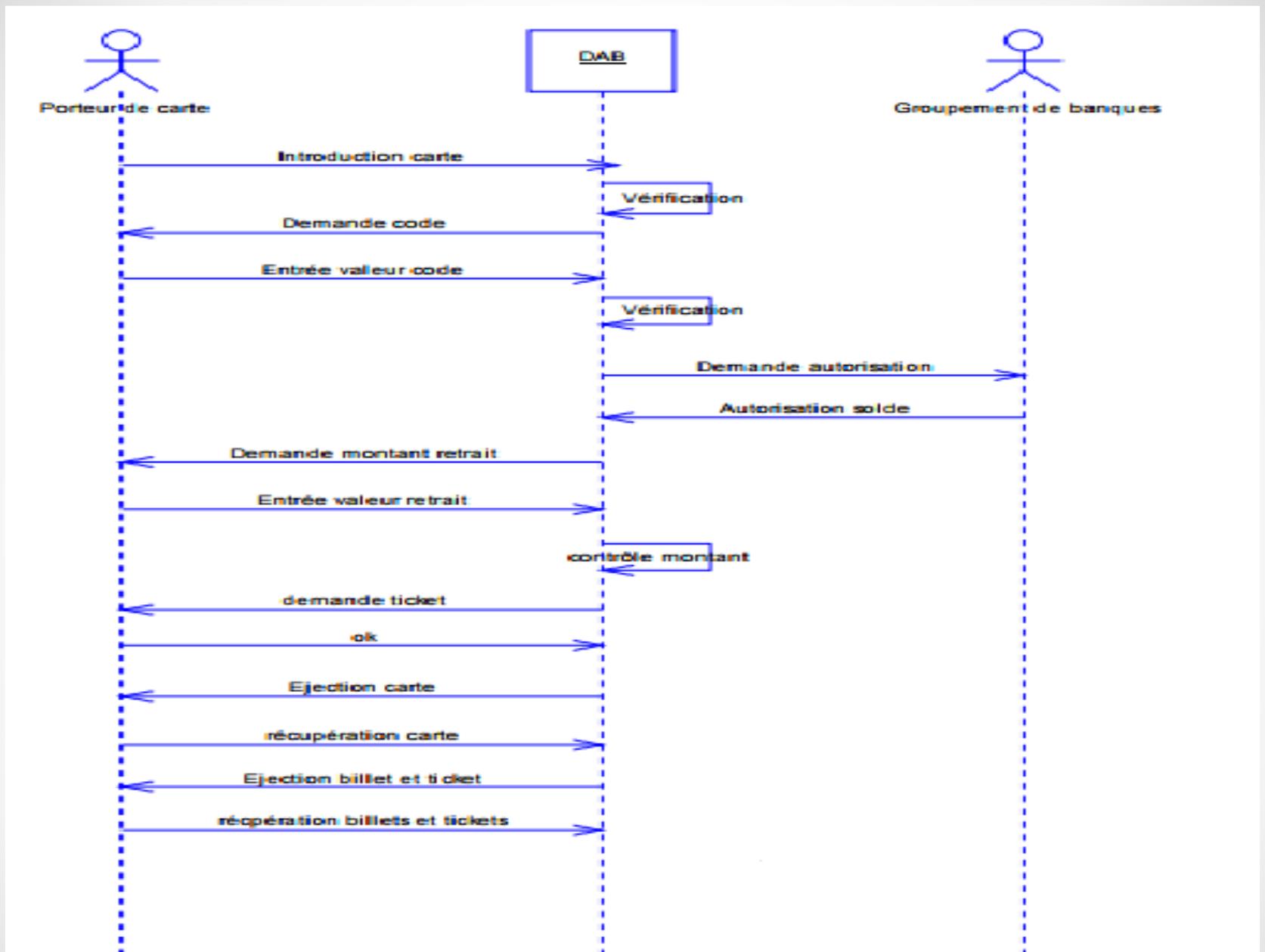


Figure 5.8 Exemple de diagramme de séquence : la représentation d'un scénario d'utilisation d'un DAB

## **Les cadres d'interaction (UML 2)**

Jusqu'ici, les constructions introduites pour l'écriture des diagrammes de séquence sont celles d'UML 1. Les diagrammes ainsi construits décrivent des scénarios. Par conséquent, pour représenter tous les scénarios de la dynamique d'un système, il convient d'écrire autant de diagrammes.

UML 2 généralise les diagrammes de séquence pour y introduire les cadres d'interaction. Cette extension importante offre le support des alternatives et des boucles et confère au diagramme de séquence le statut d'un véritable modèle des interactions.

### **La notion de cadre d'interaction**

Un cadre d'interaction est une partie du diagramme de séquence associé à une étiquette. Elle contient un opérateur qui en détermine la modalité d'exécution. Les principales modalités sont le branchement conditionnel et la boucle.

## L'alternative

L'alternative s'obtient en utilisant l'opérateur `opt` suivi d'une condition de test. Si la condition est vérifiée, le contenu du cadre est exécuté.



Figure 5.9 Cadre d'interaction d'alternative

Il existe un autre opérateur pour l'alternative. Nommé `alt`, il est suivi de plusieurs conditions de test puis du mot clé `else`. Le cadre est alors scindé en plusieurs parties dont le contenu n'est exécuté que si la condition associée est remplie. Le contenu de la dernière partie est associé au mot clé `else` (sinon). Il est exécuté uniquement si aucune des conditions précédentes n'est vérifiée.

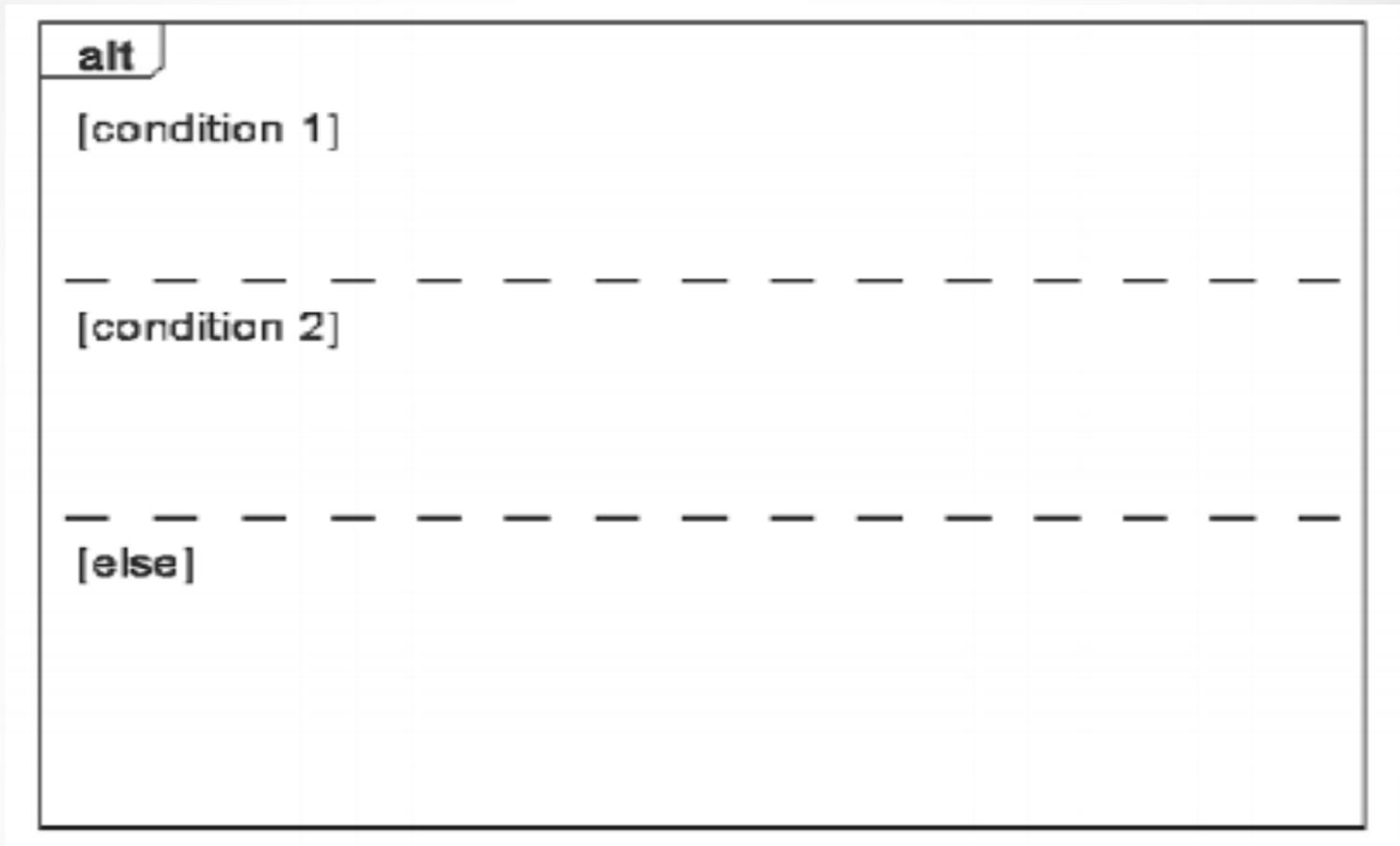


Figure 5.10 Cadre d'interaction basé sur l'opérateur alt

## La boucle

La boucle est réalisée par l'opérateur `loop` suivi des paramètres `min`, `max` et d'une condition de test.

Le contenu du cadre est exécuté `min` fois, puis tant que la condition de test est vérifiée et tant que le nombre maximal d'exécutions de la boucle ne dépasse pas `max`. Chaque paramètre est optionnel.

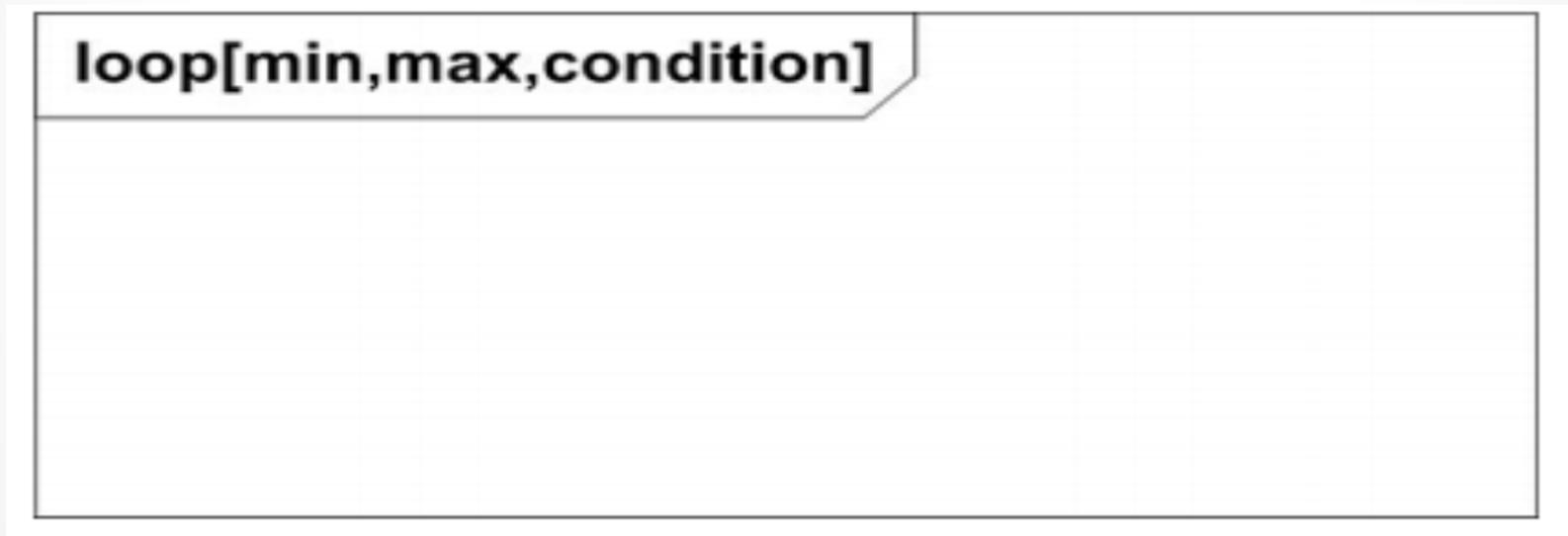


Figure 5.11 Cadre d'interaction de boucle

Exemple Pour faire une connexion, un client peut s'y prendre à plusieurs tentative.

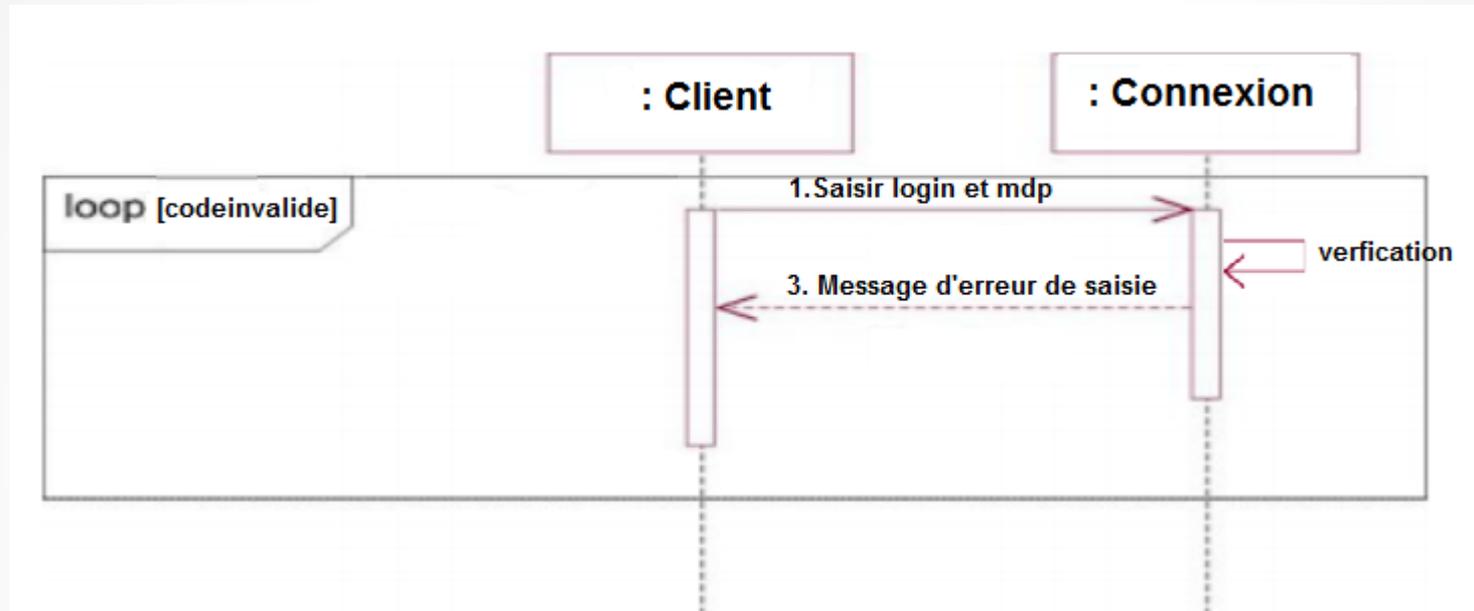


Figure 5.12 Exemple de boucle

## Utilisation des cadres d'interaction

À partir des différents éléments introduits jusqu'ici, il est maintenant possible de décrire de façon plus générale la dynamique du système.

Exemple La figure 5.13 représente le cas d'utilisation de retrait d'argents a partir d'un DAB.

À la différence de la figure 5.8, les alternatives et boucles introduites dans le cas d'utilisation sont prises en compte.

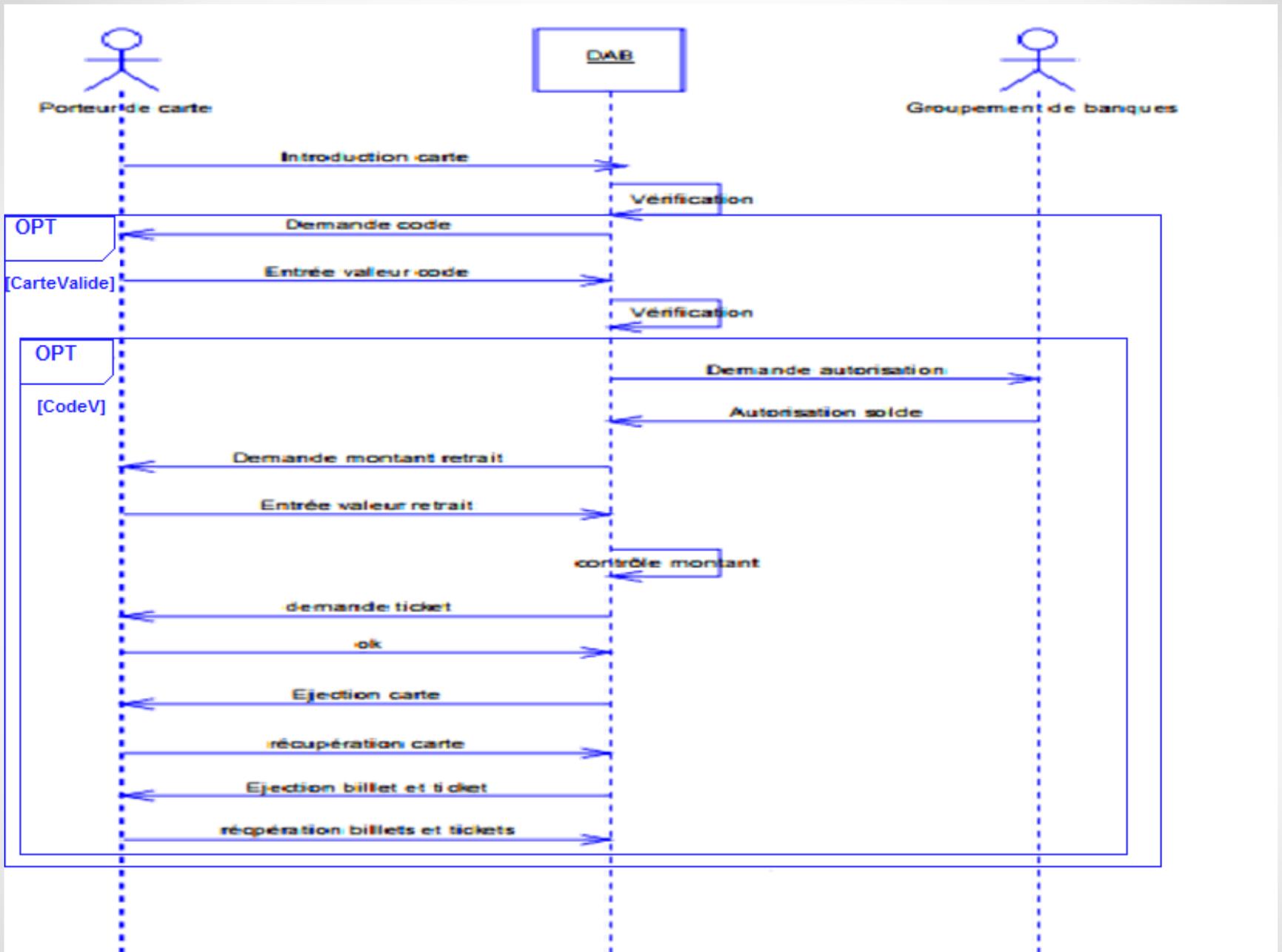


Figure 5.13 Exemple de diagramme de séquence : la représentation du cas d'utilisation Retrait

- **Conclusion**

Les diagrammes de séquence sont importants pour les raisons suivantes :

- illustrer et vérifier le comportement d'un ensemble d'objets (système ou sous-système) ;

- aider à la découverte des objets du système ;

- aider à la découverte des méthodes des objets.

Depuis l'introduction des cadres d'interaction, les diagrammes de séquence peuvent être utilisés pour décrire les cas d'utilisation.

Les diagrammes de séquence mettent en avant les aspects temporels et montrent les liaisons entre les classes.