

Modélisation objet avec UML : le diagramme de séquence

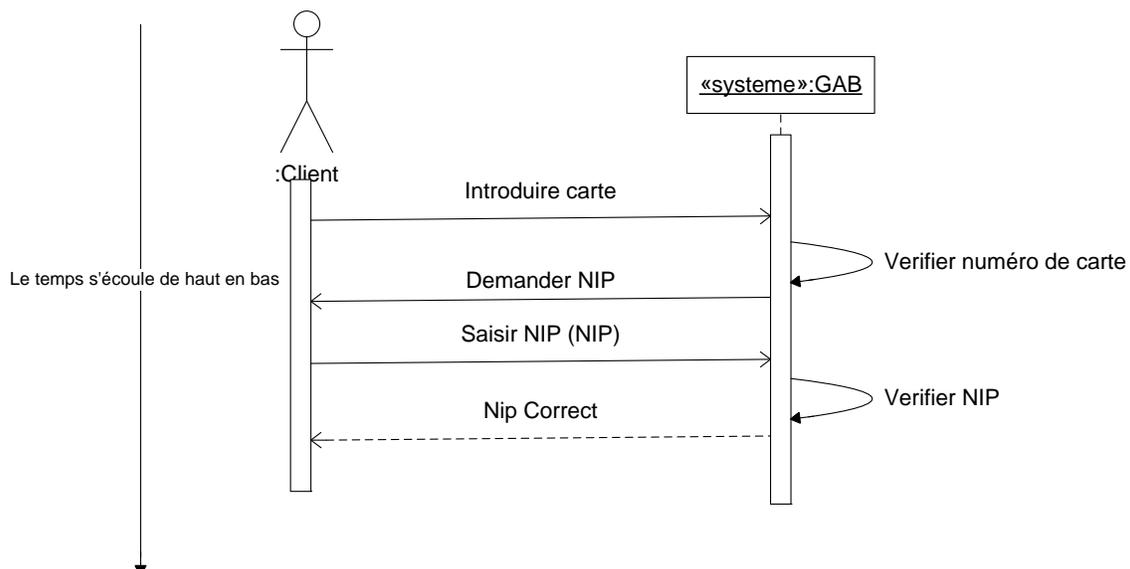
Définition et exemple

Un diagramme de séquence montre les interactions entre **les objets**, arrangés en séquence dans le temps. En particulier ils montrent les objets participants dans l'interaction par leur ligne de vie et les messages qu'ils s'échangent **ordonnés dans le temps**. Il ne montre pas les associations entre les objets.

L'ordre d'envoi des messages est déterminé par sa position sur l'axe vertical (l'axe du temps) du diagramme. Le temps s'écoule de haut en bas.

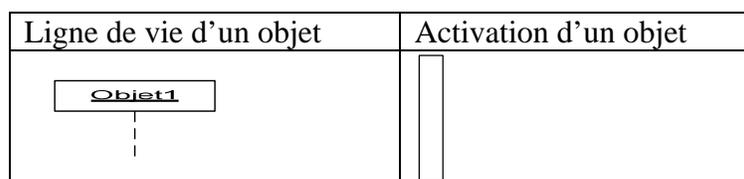
Les diagrammes de séquences permettent entre autre de faire une description graphique d'un cas d'utilisation

Exemple :



Sur un diagramme de séquence, il est possible de représenter de manière explicite les différentes périodes d'activité d'un objet au moyen d'une bande rectangulaire superposée à la ligne de vie de l'objet.

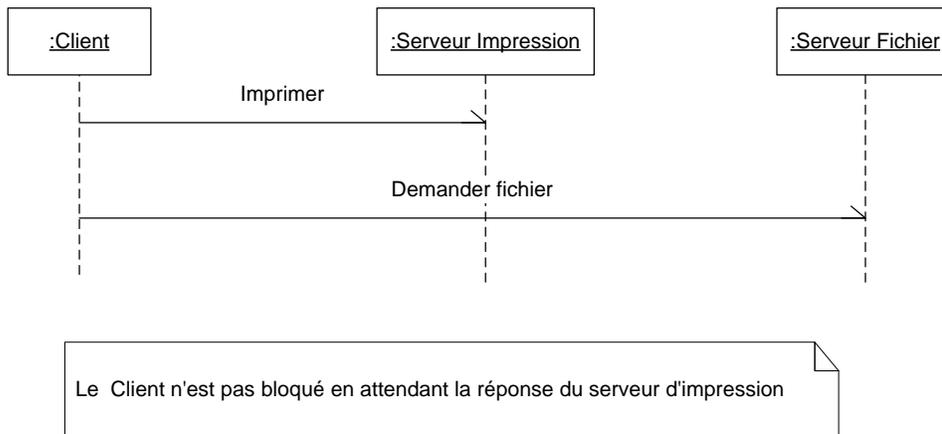
Un objet peut être activé plusieurs fois au cours de son existence. Un objet non actif n'est pas un objet détruit.



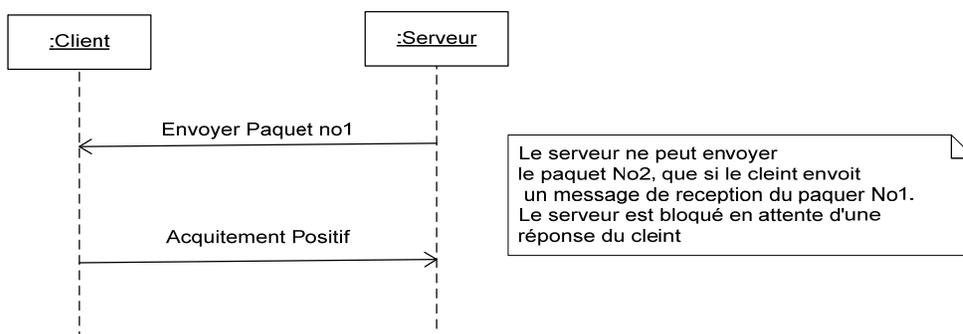
Type de Messages

Type de message	Symbole	Explication
Message simple	→	Message dont il n'y a aucune spécification particulière
Message asynchrone	→	L'émetteur n'est pas bloquée en attente d'une réponse de la part du récepteur.
Message minuté	→	L'émetteur du message est bloqué pour un temps t
Message synchrone	→	L'émetteur du message est bloqué en attente d'une réponse de la part du récepteur.
Message retour	→	

Exemple de message asynchrone



Exemple de message synchrone



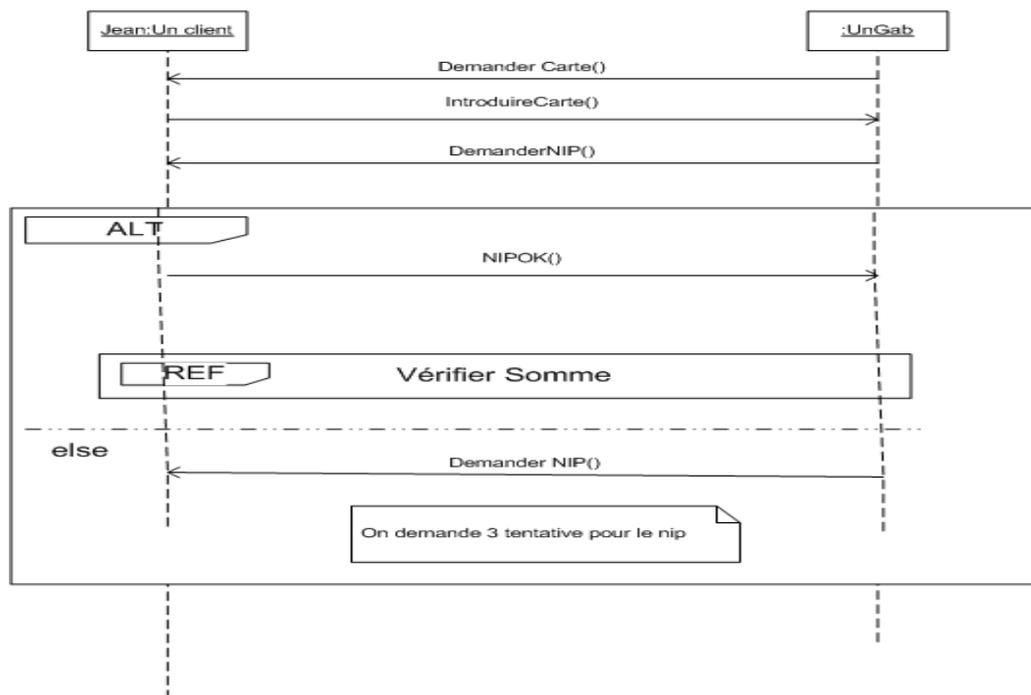
Les diagrammes de séquences selon UML 2.0

Les diagrammes de séquence tels que définis en UML1.x souffraient cependant d'un gros inconvénient. La quantité de diagrammes à réaliser pouvait atteindre un nombre conséquent dès lors que l'on souhaitait décrire avec un peu de détail les différentes branches comportementales d'une fonctionnalité. Le plus coûteux étant de remettre à jour ces diagrammes lors d'un changement au niveau des exigences ou bien du design. UML2 a donné plus de puissance à ses diagrammes de séquence en introduisant un certain nombre d'opérateurs:

1. Opérateur alt (alternative)

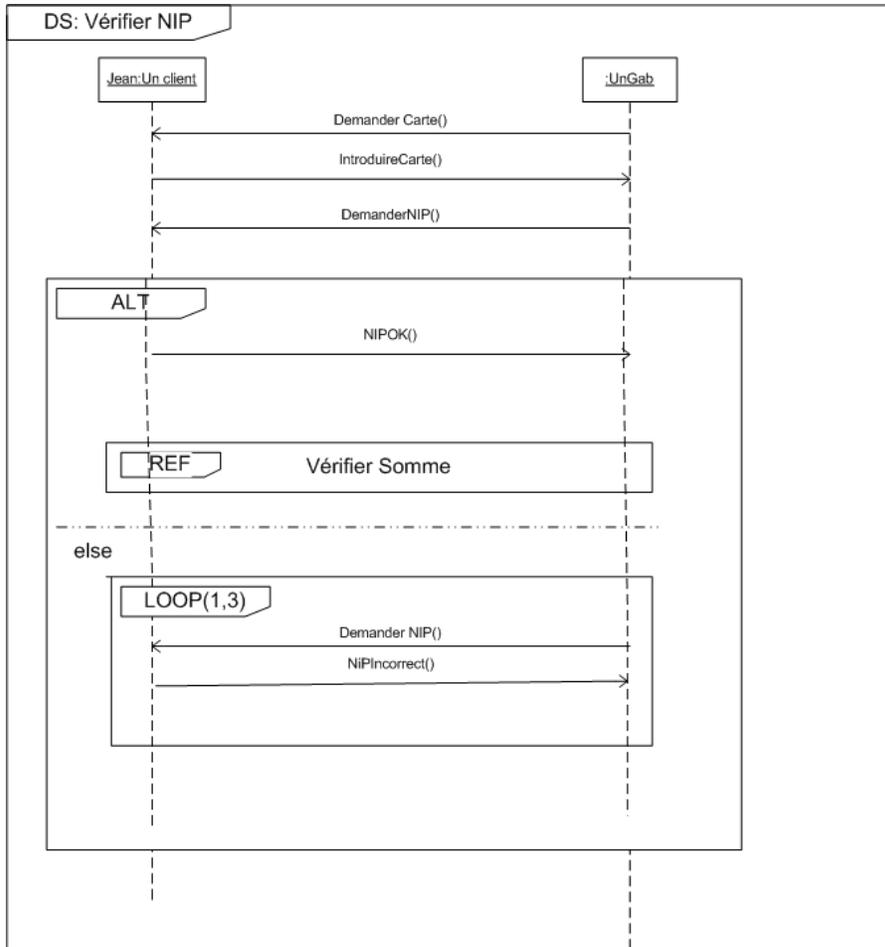
Désigne un choix, une alternative. Il représente deux comportements possibles : c'est en quelque sorte l'équivalent du **SI...ALORS...SINON** : donc, une seule des deux branches sera réalisée dans un scénario donné. La condition d'exécution d'une des deux branches (l'équivalent du SI) peut être explicite ou implicite.

L'utilisation de l'**opérateur else** permet d'indiquer que la branche est exécutée si la condition du alt est fausse.



2. Opérateur Loop

Cet opérateur est utilisé pour décrire un ensemble d'interaction qui s'exécute **en boucle**. En général, une contrainte appelée **garde** indique le nombre de répétitions (minimum et maximum) ou bien une condition booléenne à respecter.



3. Opérateur Opt (option)

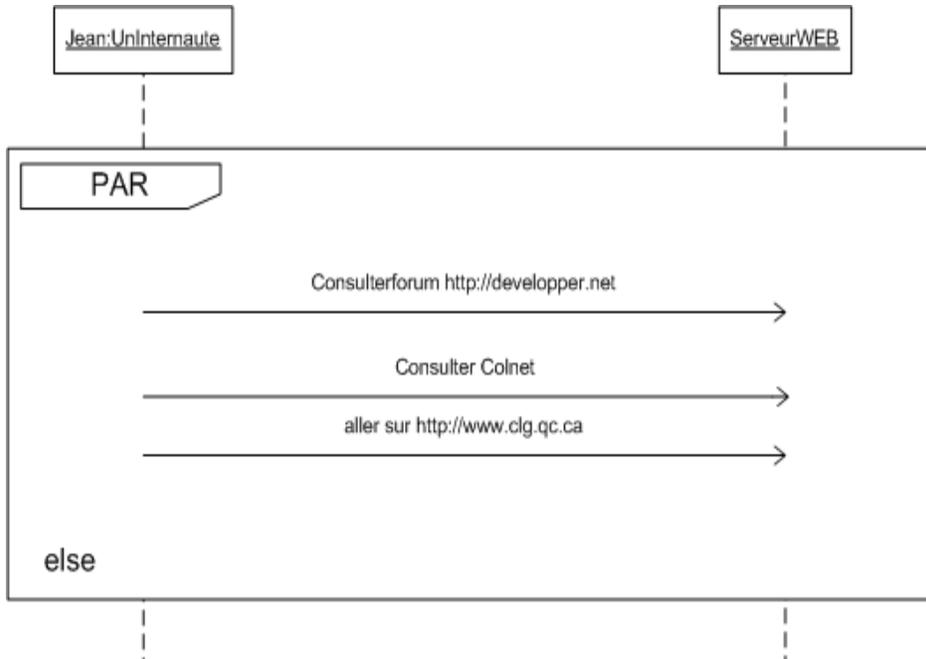
Désigne un fragment combiné optionnel comme son nom l'indique : c'est à dire qu'il représente un comportement qui peut se produire... ou pas. Un fragment optionnel est équivalent à un fragment "alt" qui ne posséderait pas d'opérande else (qui n'aurait qu'une seule branche). Un fragment optionnel est donc une sorte de **SI...ALORS**

4. Opérateur Break

Il est utilisé dans les fragments combinés qui représentent des scénarii d'**exception** en quelque sorte. 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

5. Opérateur Par (parallel)

Il est utilisé pour représenter des interactions ayant lieu en **parallèle**. Les interactions des différents opérandes (les deux branches de notre opérateur ci-dessous) peuvent donc se mélanger, s'intercaler, dans la mesure où l'ordre imposé dans chaque opérande est respecté.



6. 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.