

Opérateur alt

L'opérateur **alt** correspond à une instruction de test avec une ou plusieurs alternatives possibles. Il est aussi permis d'utiliser les clauses de type sinon.

Formalisme et exemple

L'opérateur alt se représente dans un fragment possédant au moins deux parties séparées par des pointillés. L'exemple donné (fig. 3.43) montre l'équivalent d'un test à deux conditions explicites (sans clause sinon).

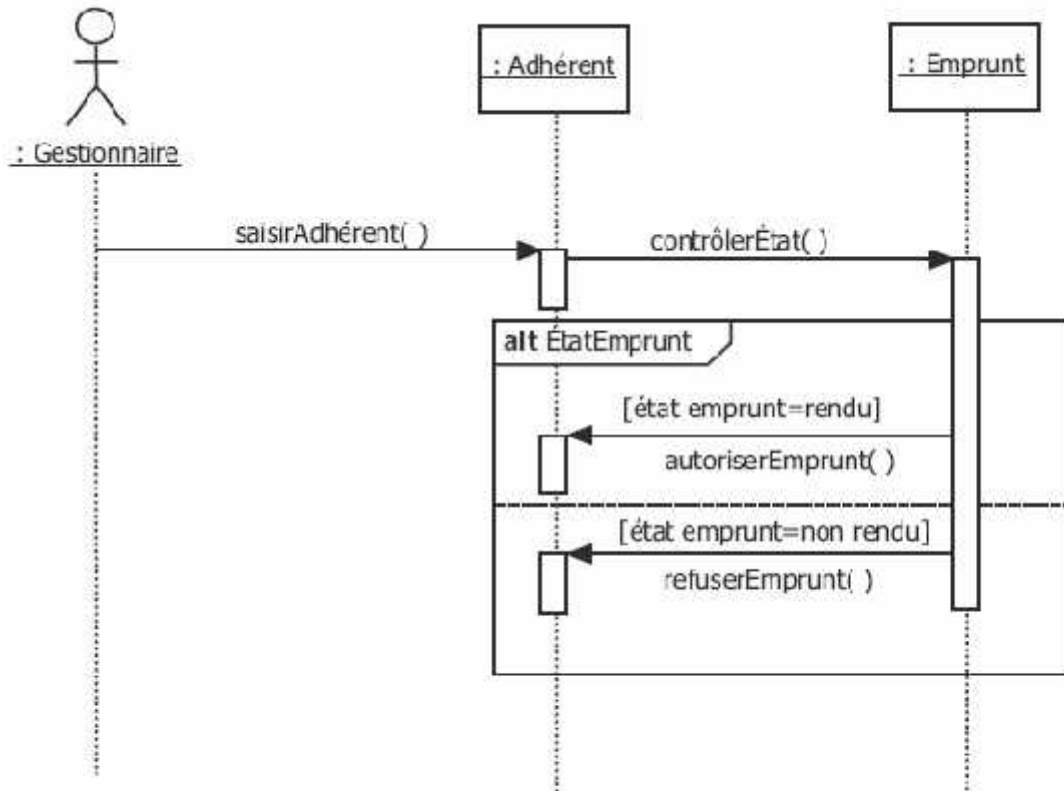


Figure 3.43 – Exemple de fragment d'interaction avec l'opérateur alt

Opérateur *opt*

L'opérateur **opt** (optional) correspond à une instruction de test sans alternative (sinon).

Formalisme et exemple

L'opérateur *opt* se représente dans un fragment possédant une seule partie (fig. 3.44).

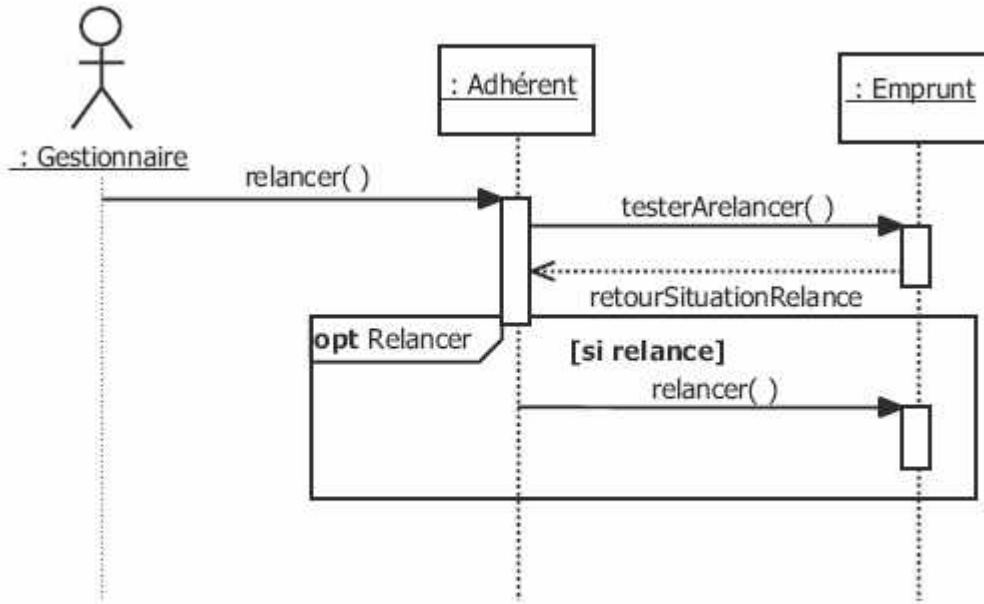


Figure 3.44 — Exemple de fragment d'interaction avec l'opérateur *opt*

Opérateur *loop*

L'opérateur **loop** correspond à une instruction de boucle qui permet d'exécuter une séquence d'interaction tant qu'une condition est satisfaite.

Il est possible aussi d'utiliser une condition portant sur un nombre minimum et maximum d'exécution de la boucle en écrivant : *loop* min, max. Dans ce cas, la boucle s'exécutera au minimum min fois et au maximum max fois. Il est aussi possible de combiner l'option min/max avec la condition associée à la boucle.

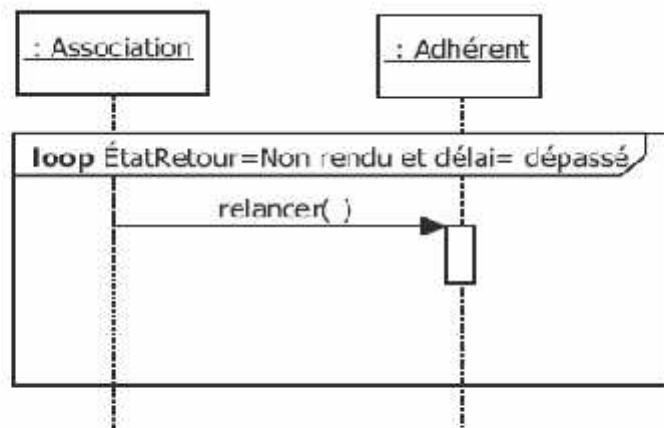


Figure 3.45 — Exemple de fragment d'interaction avec l'opérateur *loop*

Opérateur *par*

L'opérateur **par** (parallèle) permet de représenter deux séries d'interactions qui se déroulent en parallèle.

Formalisme et exemple

L'opérateur **par** se représente dans un fragment possédant deux parties séparées par une ligne en pointillé. C'est un opérateur qui est à notre avis plutôt utilisé dans l'informatique temps réel et c'est pour cela que nous ne donnerons qu'un exemple type (fig. 3.46).

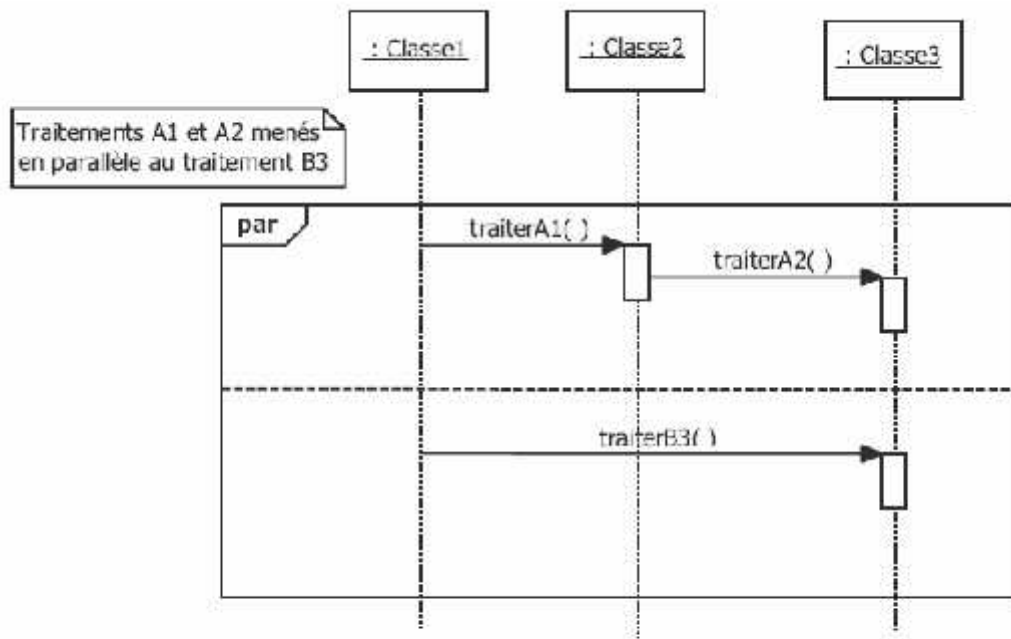


Figure 3.46 — Exemple de fragment d'interaction avec l'opérateur **par**

Opérateurs strict et weak sequencing

Les opérateurs **strict** et **weak** permettent de représenter une série d'interactions dont certaines s'opèrent sur des objets indépendants :

- L'opérateur strict est utilisé quand l'ordre d'exécution des opérations doit être strictement respecté.
- L'opérateur weak est utilisé quand l'ordre d'exécution des opérations n'a pas d'importance.

Formalisme et exemple

L'exemple présenté figure 3.47 montre que les opérations A1, A2, B1, B2 et A3 doivent être exécutées dans cet ordre puisqu'elles font partie du fragment d'interaction comportant l'opérateur strict.

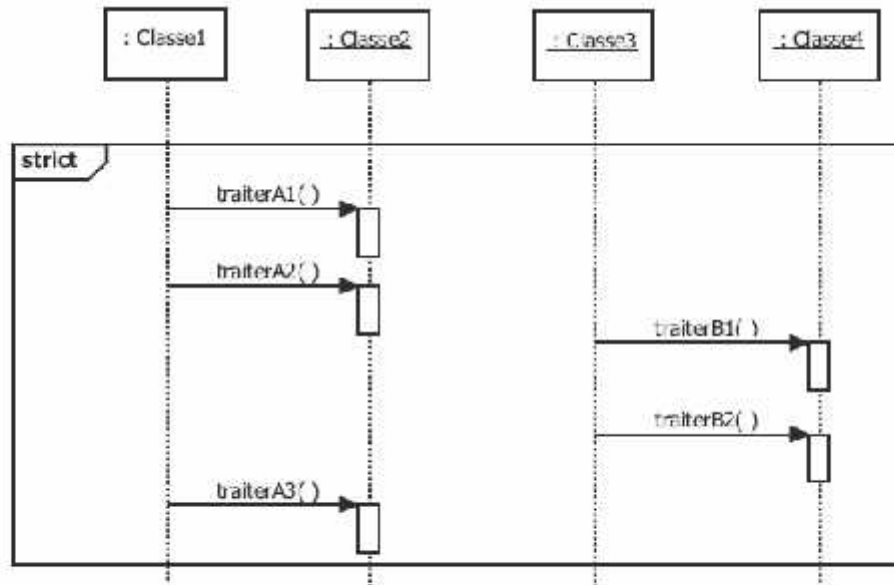


Figure 3.47 — Exemple de fragment d'interaction avec l'opérateur strict

Opérateur break

L'opérateur **break** permet de représenter une situation exceptionnelle correspondant à un scénario de rupture par rapport au scénario général. Le scénario de rupture s'exécute si la condition de garde est satisfaite.

Formalisme et exemple

L'exemple présenté figure 3.48 montre que les opérations annulerOp1(), annulerOp2() et afficherAide() ne seront exécutées que si la touche F1 est activée sinon le fragment est ignoré et la séquence de traitement passe directement de l'opération Op2() à l'opération Op3().

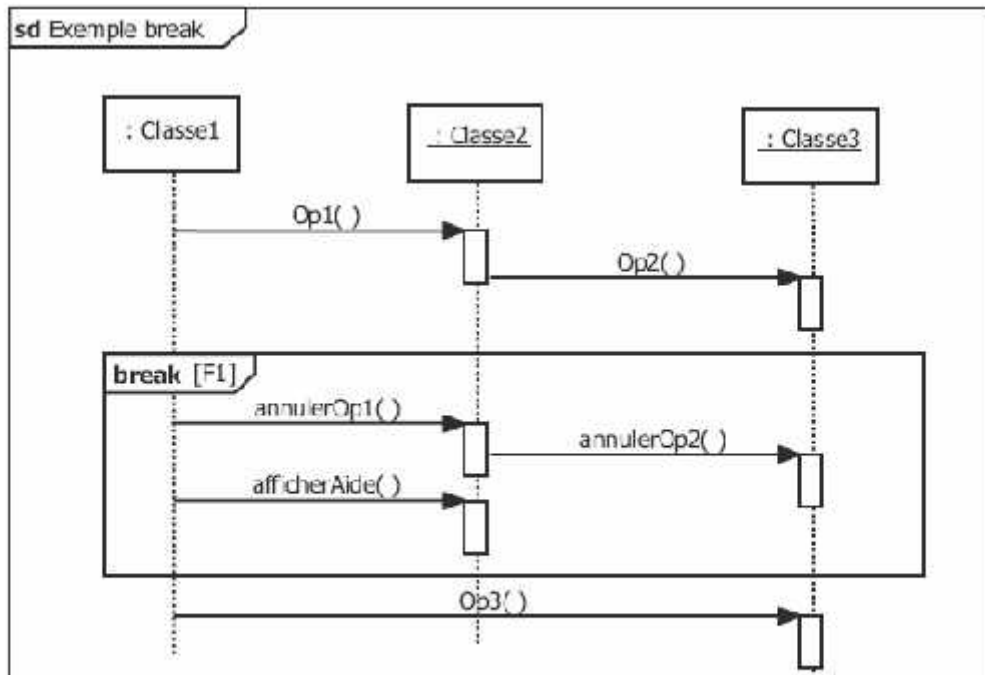


Figure 3.48 — Exemple de fragment d'interaction avec l'opérateur break

Opérateurs ignore et consider

Les opérateurs **ignore** et **consider** sont utilisés pour des fragments d'interactions dans lesquels on veut montrer que certains messages peuvent être soit absents sans avoir d'incidence sur le déroulement des interactions (ignore), soit obligatoirement présents (consider).

Formalisme et exemple

L'exemple présenté figure 3.49 montre que :

- dans le fragment **consider**, les messages Op1, Op2 et Op5 doivent être obligatoirement présents lors de l'exécution du fragment sinon le fragment n'est pas exécuté,
- dans le fragment **ignore**, les messages Op2 et Op3 peuvent être absents lors de l'exécution du fragment.

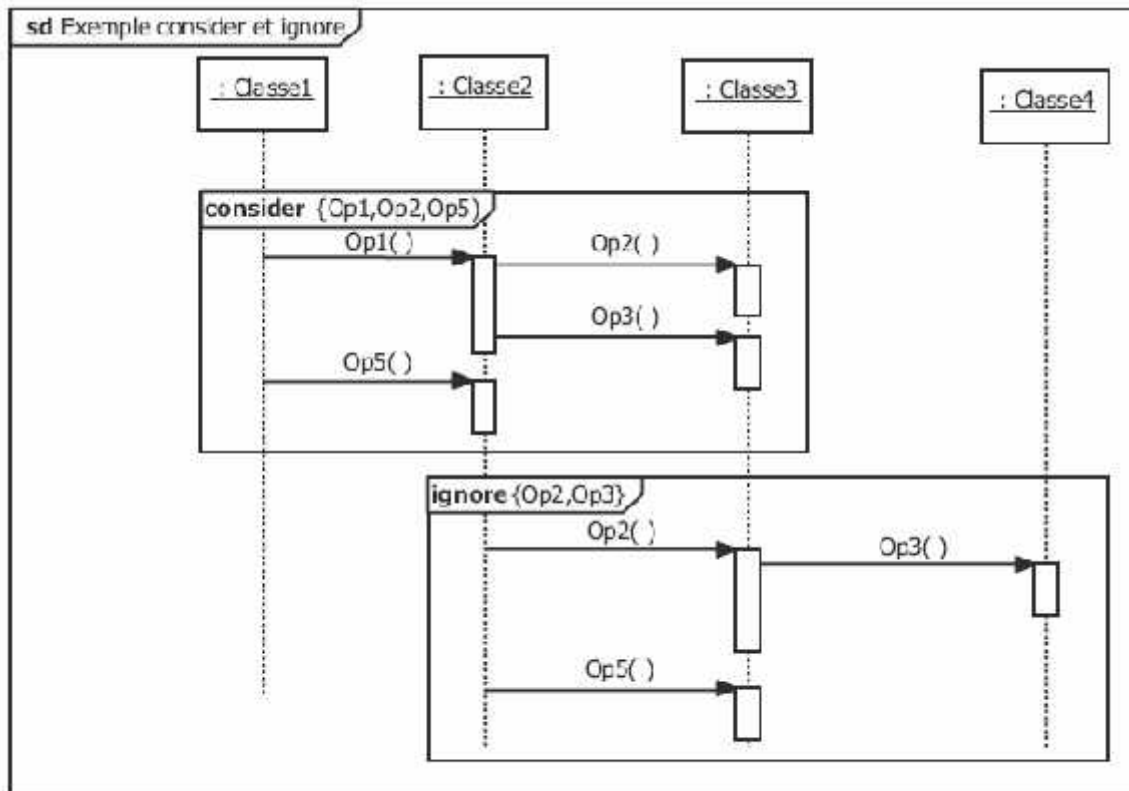


Figure 3.49 — Exemple de fragment d'interaction

Opérateur *critical*

L'opérateur **critical** permet d'indiquer qu'une séquence d'interactions ne peut être interrompue compte tenu du caractère critique des opérations traitées. On considère que le traitement des interactions comprises dans la séquence critique est atomique.

Formalisme et exemple

L'exemple présenté figure 3.50 montre que les opérations Op1(), Op2() et Op3() du fragment critical doivent s'exécuter sans interruption.

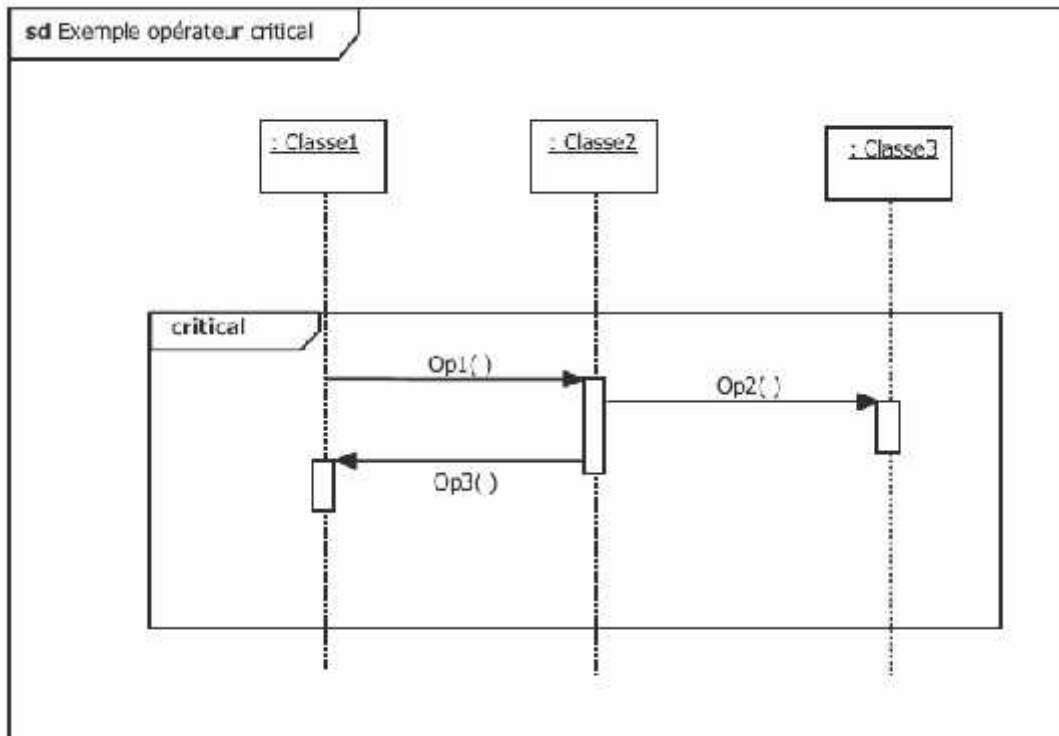


Figure 3.50 — Exemple de fragment d'interaction avec l'opérateur *critical*

Opérateur *negative*

L'opérateur **neg** (negative) permet d'indiquer qu'une séquence d'interactions est invalide.

Formalisme et exemple

L'exemple présenté figure 3.51 montre que les opérations Op1() et Op2() du fragment **neg** sont invalides. Une erreur sera déclenchée dans ce cas à l'exécution du fragment.

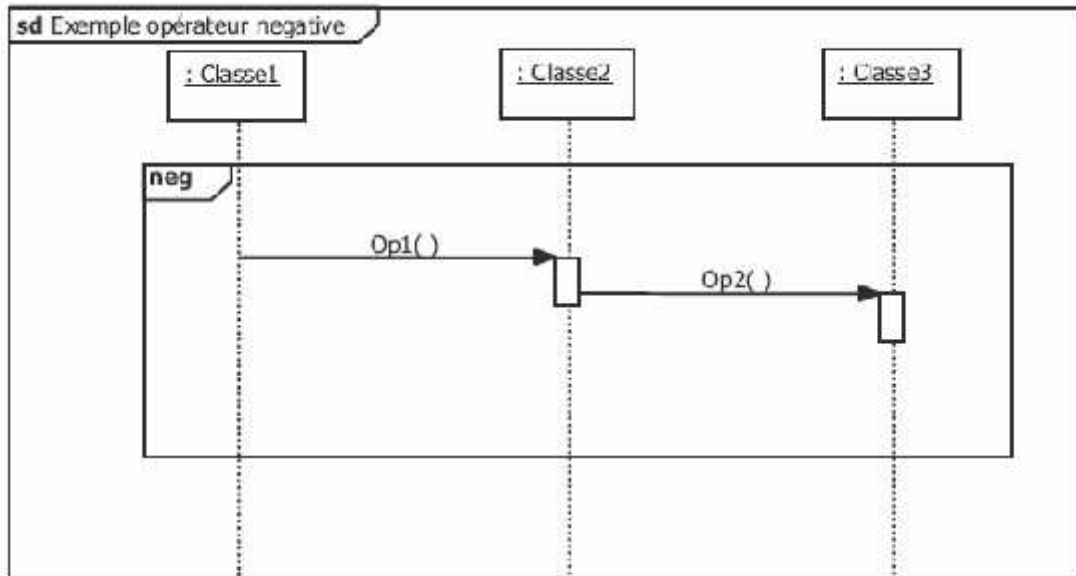


Figure 3.51 — Exemple de fragment d'interaction avec l'opérateur **neg**

Opérateur assertion

L'opérateur **assert** (assertion) permet d'indiquer qu'une séquence d'interactions est l'unique séquence possible en considérant les messages échangés dans le fragment. Toute autre configuration de message est invalide.

Formalisme et exemple

L'exemple présenté figure 3.52 montre que le fragment assert ne s'exécutera que si l'unique séquence de traitement Op1(), Op2() et Op3() se réalise en respectant l'ensemble des caractéristiques de ces opérations (paramètre d'entrée, type de résultat...). Toute autre situation sera considérée invalide.

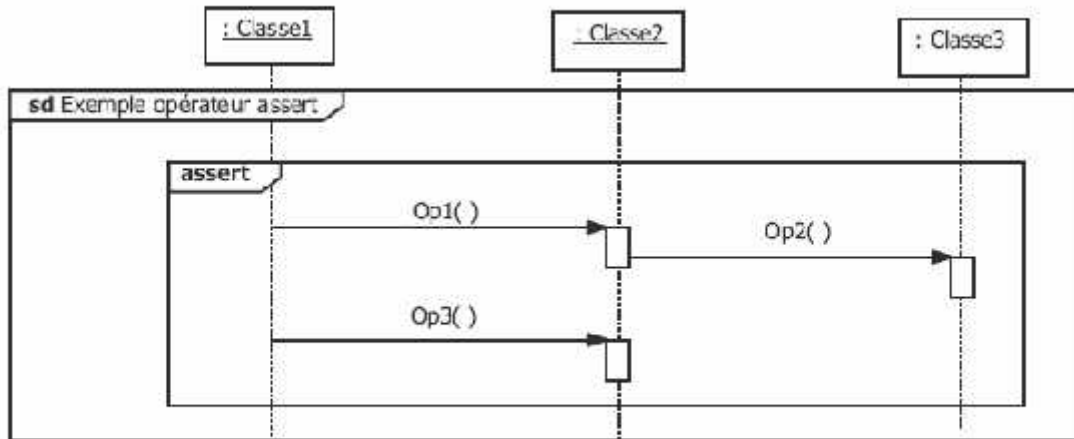


Figure 3.52 — Exemple de fragment d'interaction avec l'opérateur assert

Opérateur ref

L'opérateur **ref** permet d'appeler une séquence d'interactions décrite par ailleurs constituant ainsi une sorte de sous-diagramme de séquence.

Formalisme et exemple

L'exemple présenté figure 3.53 montre que l'on fait appel à un fragment « Contrôle des droits » qui est décrit par ailleurs.

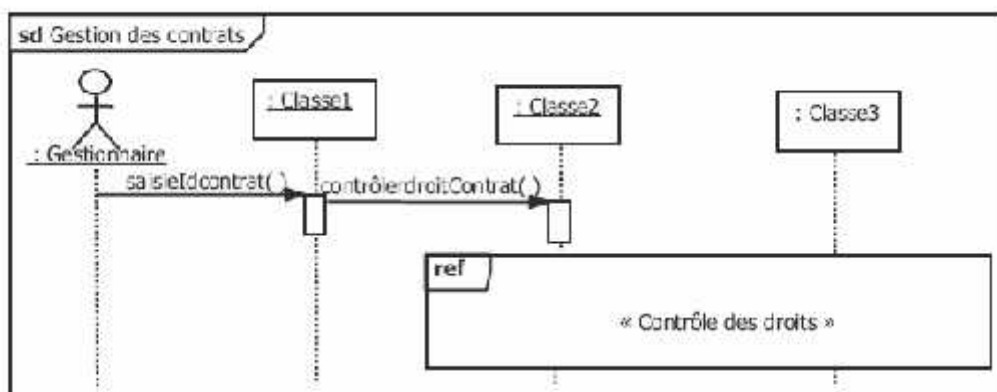


Figure 3.53 — Exemple de fragment d'interaction avec l'opérateur ref