Génie Logiciel

Diagrammes dynamiques de description de scénarios

Dit de Séquences

Scénario

- Un scénario est une *suite spécifique* d'événements survenant dans le système.
- Un scénario est une séquence spécifique d'actions et d'interactions entre les acteurs et le système.
  - C'est une histoire spécifique de l'utilisation d'un système.
- Par exemple, le scénario d'un achat réussi d'articles en liquide,

  ou

  le scénario d'un échec d'achat d'articles à cause d'un refus de paiement à crédit.

Scénario

Description d'un scénario :

- ***Scénario principal*** : description du chemin «*normal*» d'exécution du cas d'utilisation;

  Exemple : Retrait-distributeur pour un client existant et fiable,

- ***Scénarios secondaires*** : description de cas *alternatifs* (plusieurs choix), de cas *exceptionnels* ou de cas *d'erreur*.

  Exemple : Retrait-distributeur pour un client donnant un code erroné

- Il y a en UML 3 représentations possibles pour un scénario :
  - description textuelle en langage naturel
  - diagramme de séquence
  - diagramme de collaboration

Scénario - description textuelle en langage naturel

- Scénarios (principaux ou secondaires) peuvent être décrit informellement en utilisant une ‘mini-histoire’.

- C.a.d: nous avons une description narrative des actions et interactions des acteurs (ou objets) du système.

Scénario - description textuelle en langage naturel

Exemple:

Retrait-distributeur (*cas normal*) :

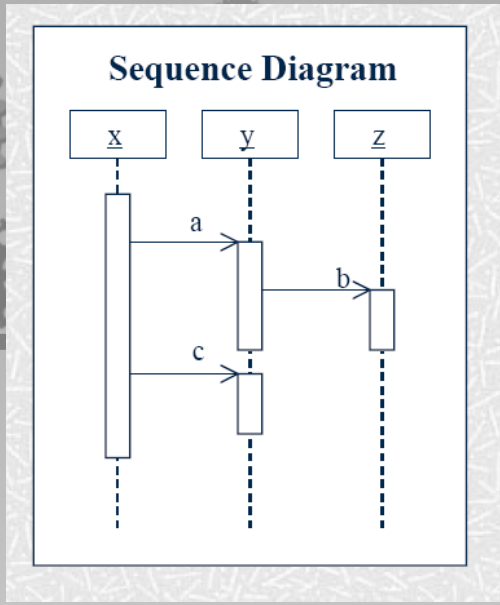
- Dupont insère sa carte,
- le distributeur D1 accepte la carte et lit le numéro du compte et de la banque,
- D1 demande le code de Dupont;
- Dupont entre son code '2341',
- D1 demande au consortium C1 de vérifier le numéro de compte et le code saisi,
- C1 signifie son acceptation à D1,
- D1 demande à Dupont le montant du retrait,
- Dupont indique le montant de 50 Euros ...

Scénario - description textuelle en langage naturel

Exemple: Retrait-distributeur (*cas d'erreur*) :

- Dupont insère sa carte,
- le distributeur D1 accepte la carte et lit le numéro du compte et de la banque,
- D1 demande le code de Dupont,
- Dupont entre le code '2341',
- D1 demande au consortium C1 de vérifier le numéro de compte et le code saisi,
- C1 signifie son refus à D1,
- D1 demande à Dupont demande à nouveau le code de Dupont,
- Dupont entre le code '2341',
- D1 demande à nouveau au consortium C1 de vérifier le numéro de compte et le code saisi,
- C1 signifie à nouveau son refus à D1,
- D1 signifie son refus de l'opération à Dupont, et conserve la carte,
- D1 met fin à son utilisation par Dupont...

- Il y a en UML 3 représentations possibles pour un scénario :
  - description textuelle en langage naturel
  - diagramme de séquence
  - diagramme de collaboration



Diagrammes de séquence

Diagrammes de séquence

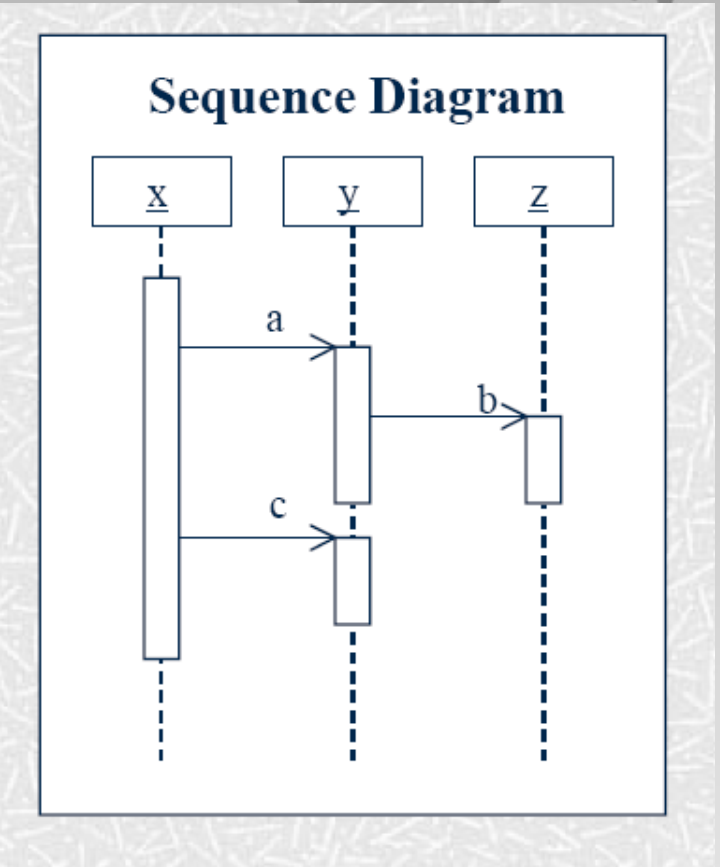
- Les diagrammes de séquences décrivent de manière plus formelle les interactions/collaborations entre les acteurs (ou objets) dans le système.
- Les diagrammes de séquences permettent de représenter des collaborations entre objets ou acteurs selon un point de vue **temporel**, on y met l'accent sur la **chronologie des envois de messages**.

Diagrammes de séquence

- Peuvent être utilisés à différents niveaux de détail et pour servir différents buts à plusieurs étapes du cycle de vie du développement.
- Typiquement utilisés pour représenter l'interaction détaillées en objets qui prend place dans un *cas d'utilisation* ou pour une *opération*.

Diagrammes de séquence

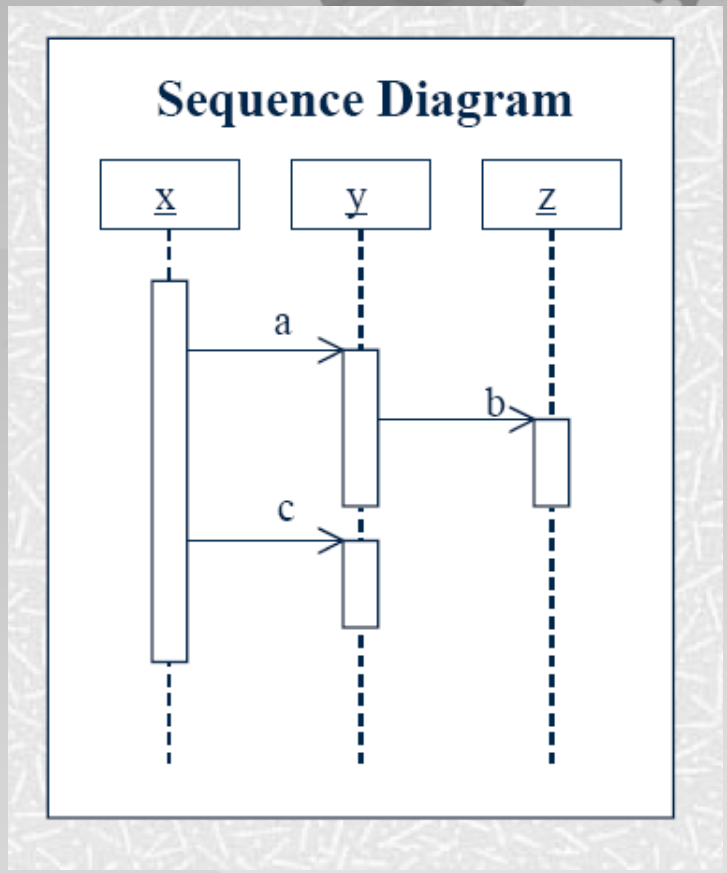
- La dimension verticale montre le temps.
- Les objets (ou sous-systèmes ou autre objets connectables) impliqués dans l'interaction apparaissent horizontalement sur la page et sont représentés par les lignes de vie («lifelines»).



Diagrammes de séquence

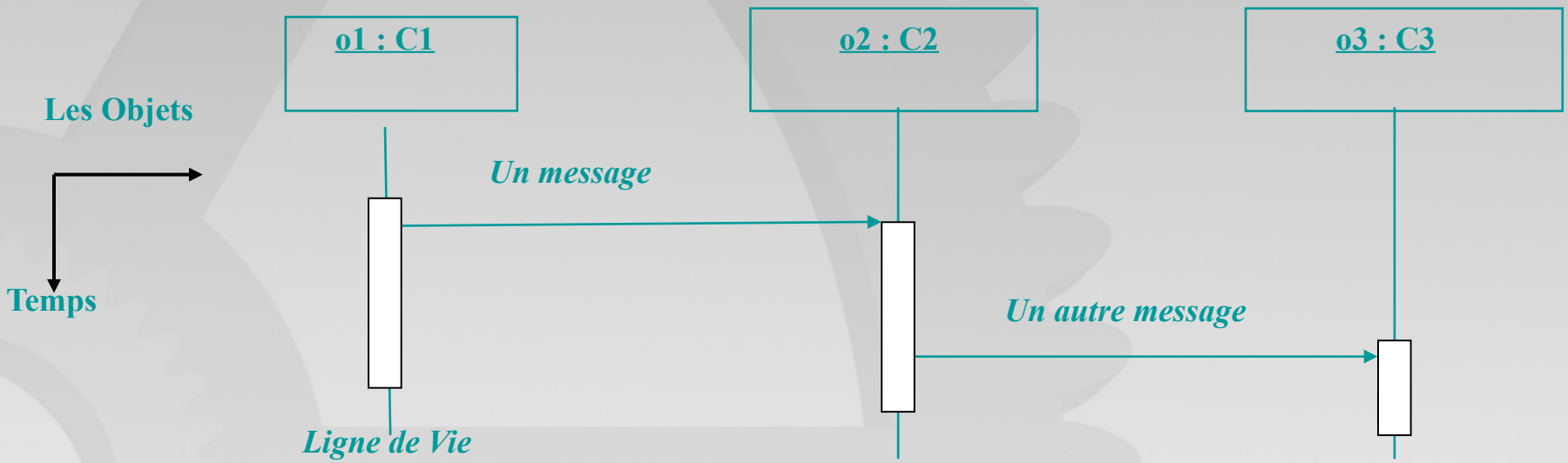
- Les messages sont signalés par une flèche horizontale pleine.

- L'exécution ou l'activation d'une opération est signalée par un rectangle sur la "lifeline" appropriée.



Diagrammes de Séquences (DS)

- Définissent les interactions entre Objets selon un point de vue temporel,
- Interaction = événement / envoi de message



Diagrammes de Séquences

DS de haut niveau de Retrait-distributeur :

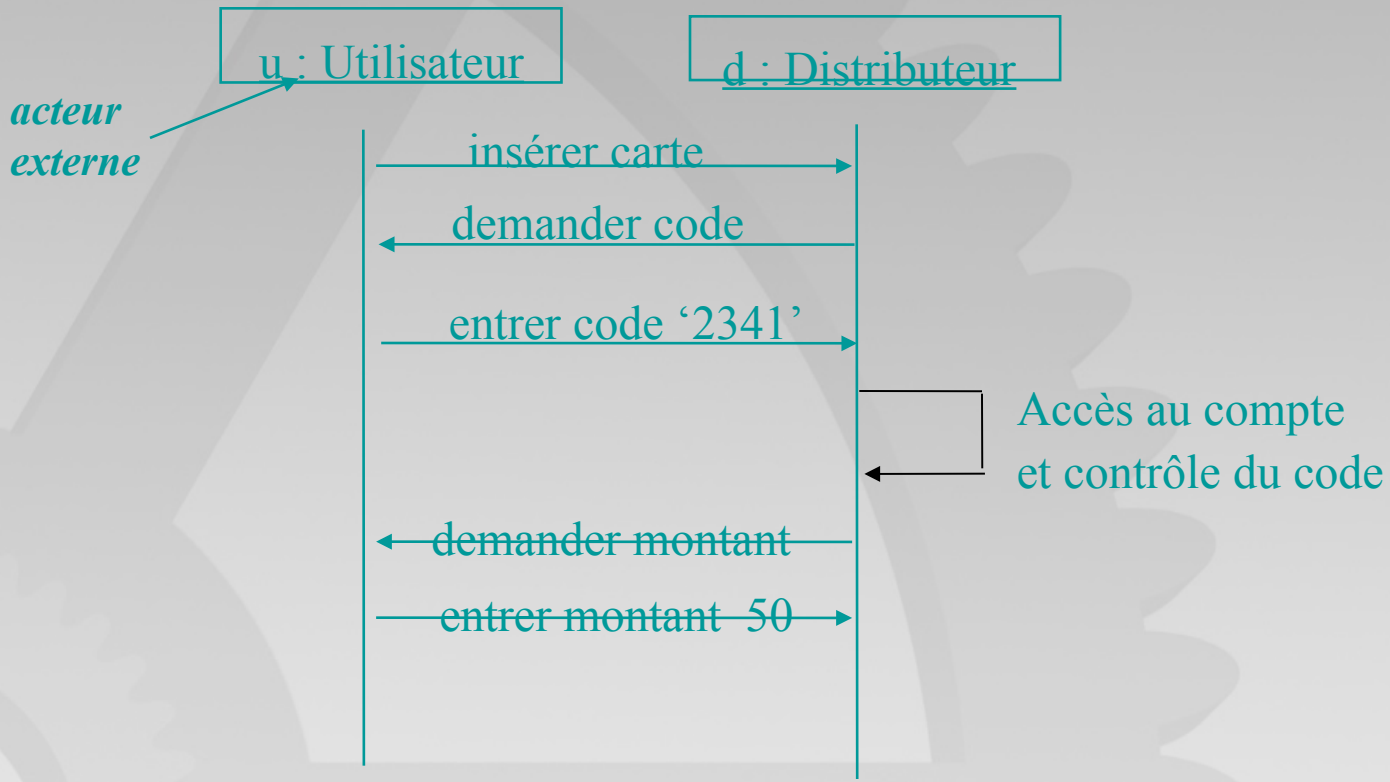
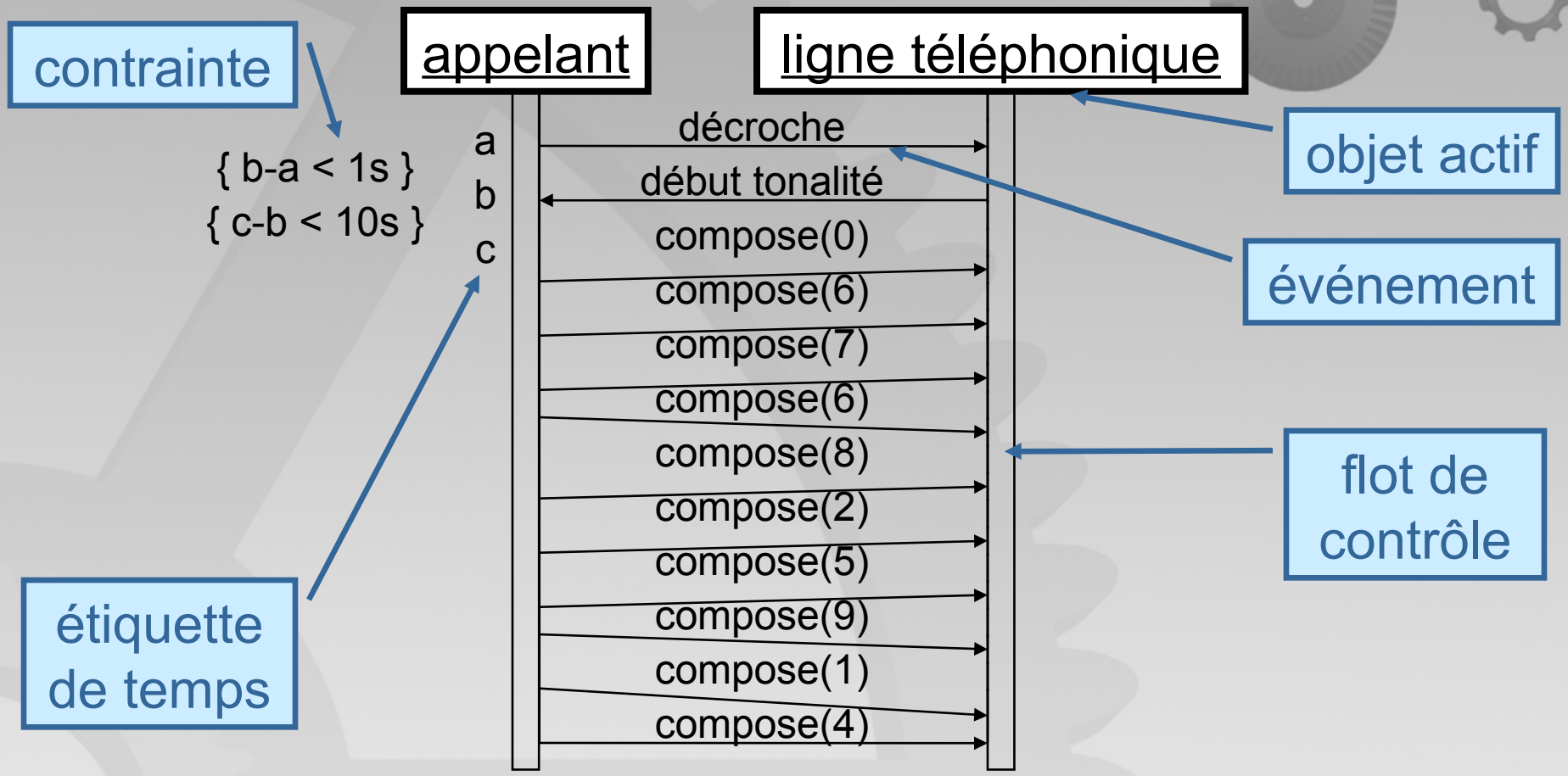
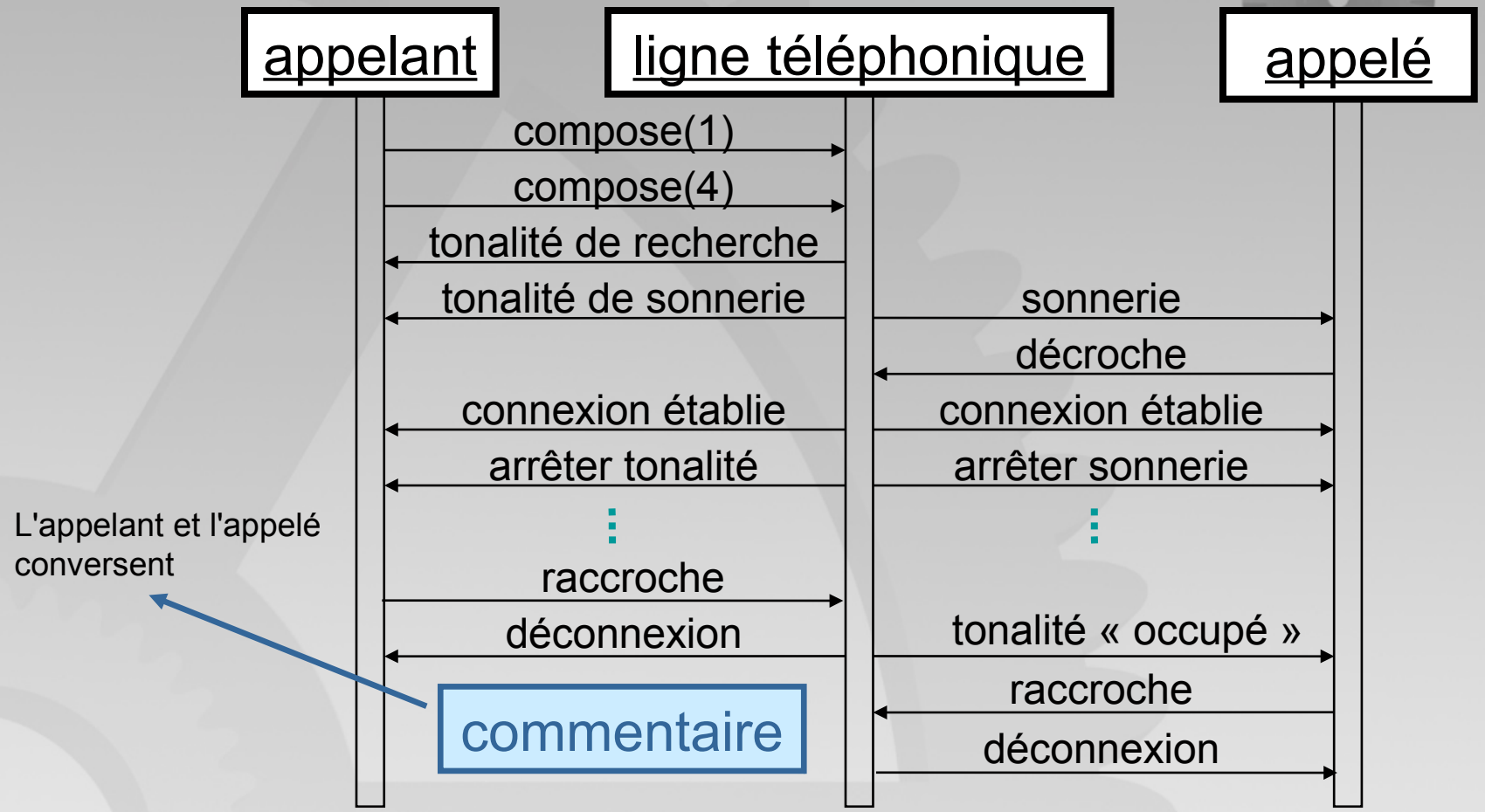


Diagramme de séquence



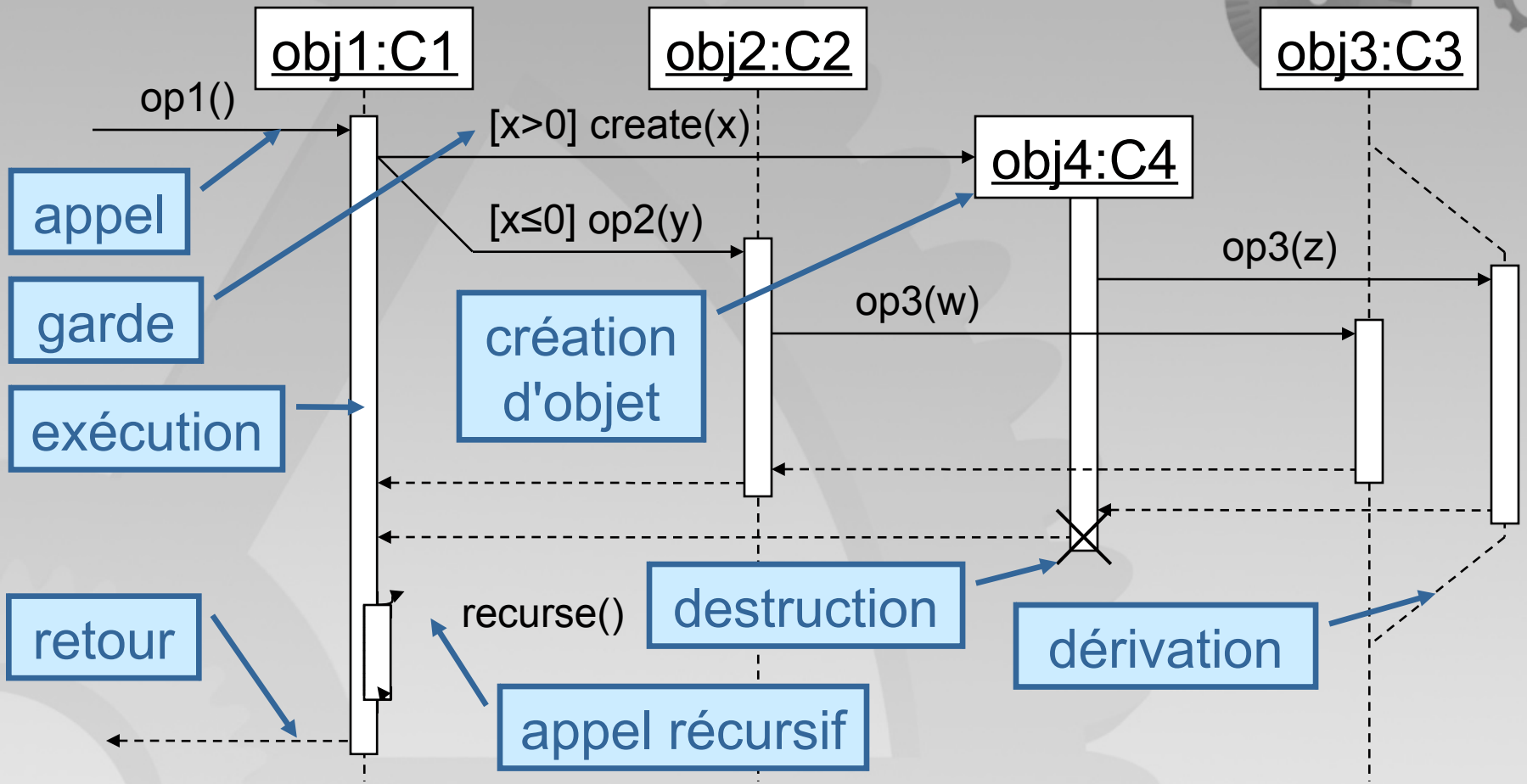
Le temps est censé s'écouler du haut vers le bas

Diagramme de séquence



Dans ce scénario, c'est l'appelant qui raccroche

Diagramme de séquence

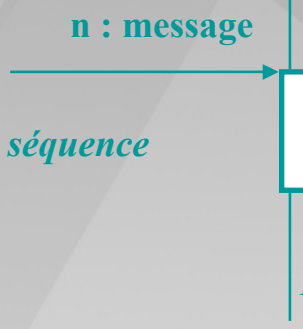


Un diagramme de séquence pour décrire une opération

Diagrammes de Séquences

Conventions graphiques :

Objet participant à l'interaction

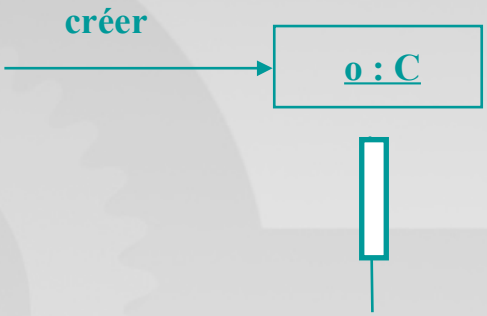


Activation : période de temps pendant laquelle l'objet exécute une action

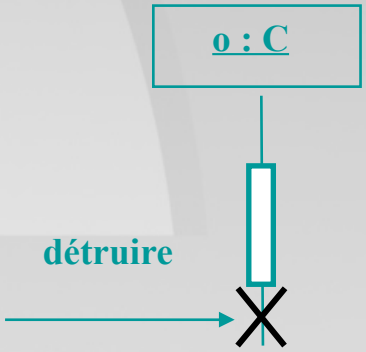
n : numéro de séquence facultatif

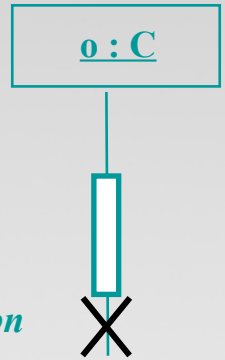
Ligne de vie

Pour démarrer sa ligne de vie



Pour arrêter sa ligne de vie

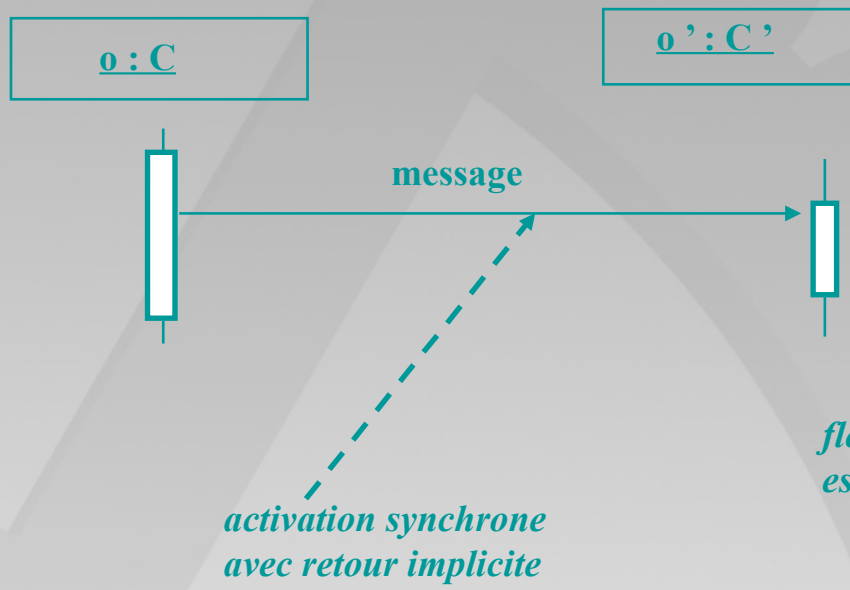




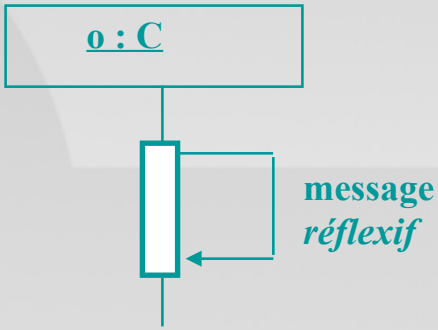
ou auto-destruction

Diagrammes de Séquences

Conventions graphiques (suite) :

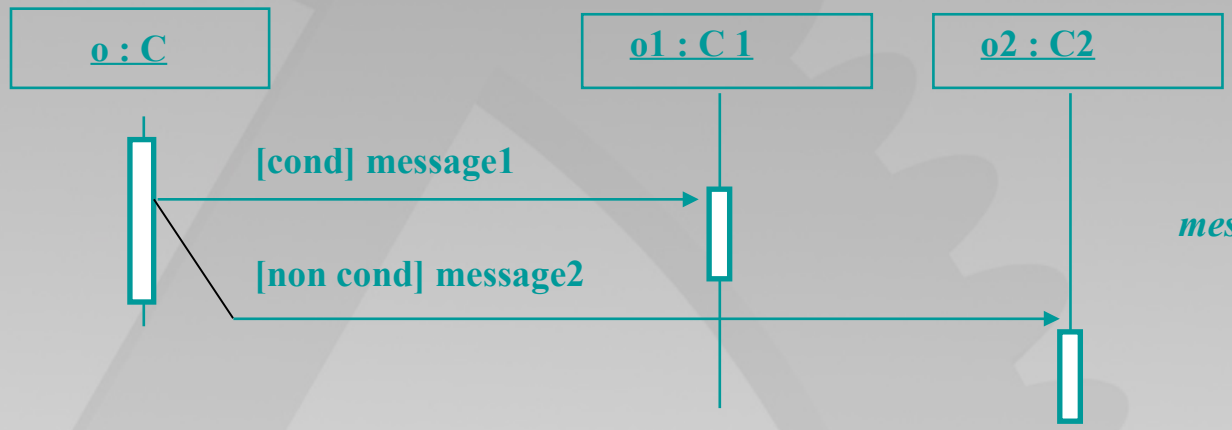


flèche horizontale : le temps de transmission est considéré comme nul

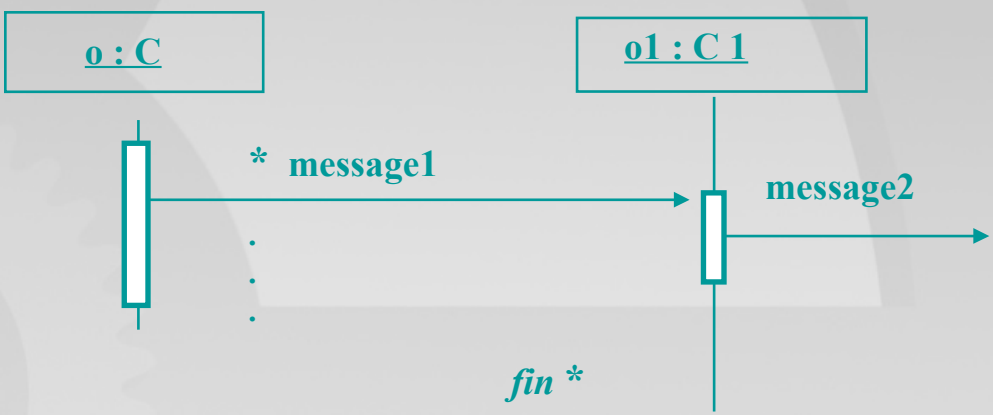


Diagrammes de Séquences

Conventions graphiques (suite) :



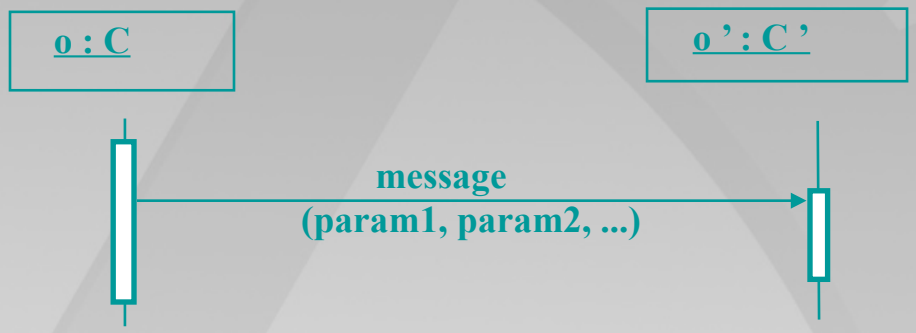
messages conditionnés



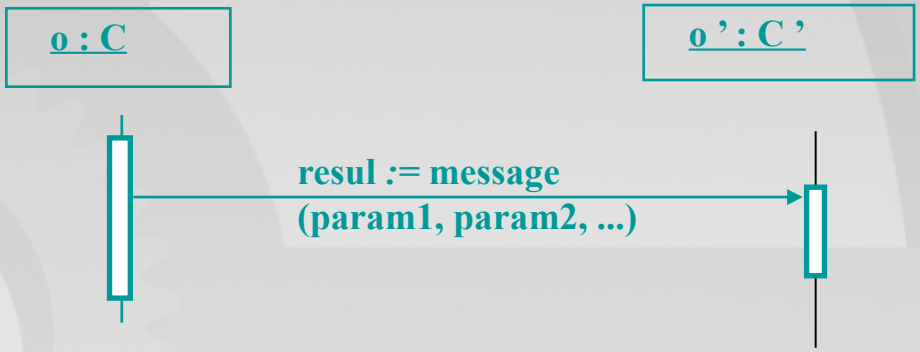
*Itération d'un message ou d'une séquence de messages (dans ce dernier cas, fin * délimite la séquence itérée)*

Diagrammes de Séquences

Conventions graphiques (suite) :



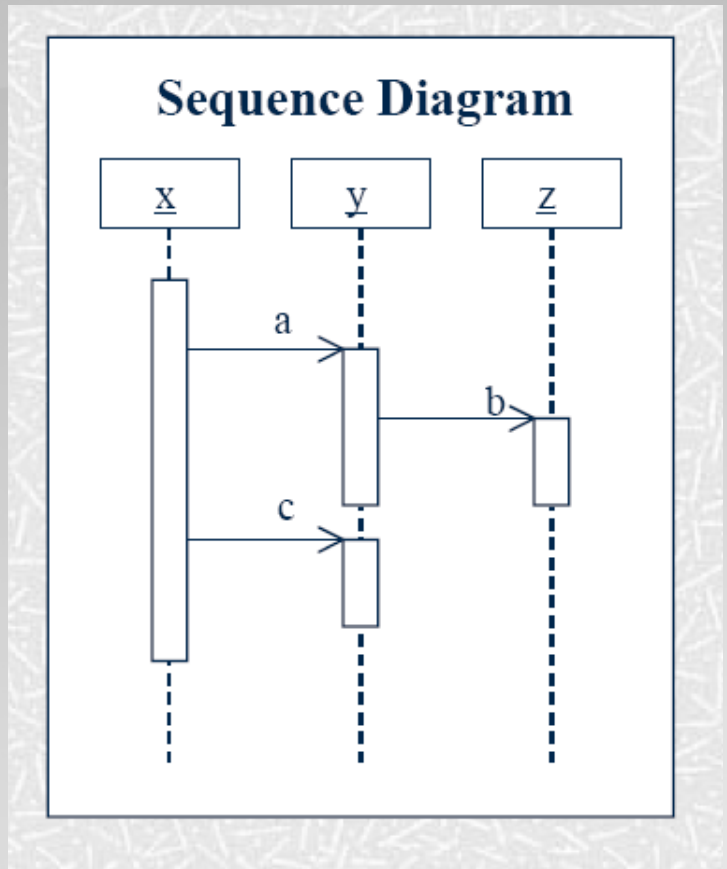
message avec paramètres



message avec paramètres et résultat de retour

Liens entre les messages et les opérations

- Le message qui est envoyé à un objet représente une opération/méthode que la classe de cet objet implémente.

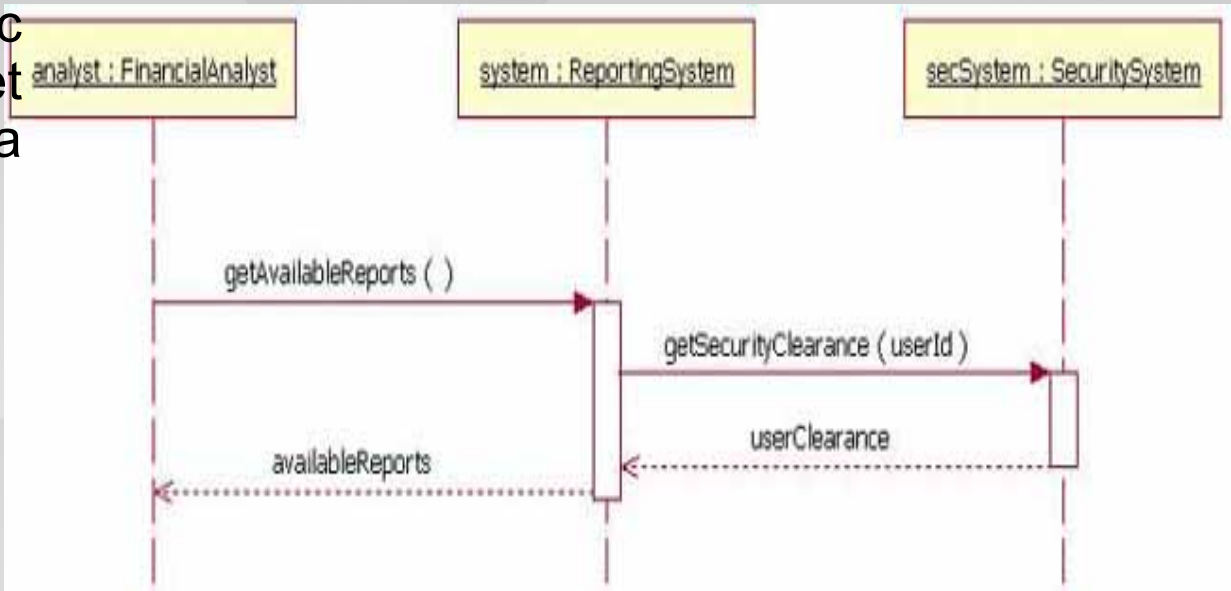


Liens entre les messages et les opérations : exemple

L'objet *'analyst'* fait un appel à l'objet '*system*' qui est une instance de la classe '*ReportingSystem*'.

L'objet '*analyst*' appelle la méthode '*getAvailableReports*' de l'objet '*system*'.

L'objet '*system*' appelle alors la méthode *'getSecurityClearance'* avec l'argument *'userId'* sur l'objet *'secSystem'*, qui est de la classe *'SecuritySystem'*.



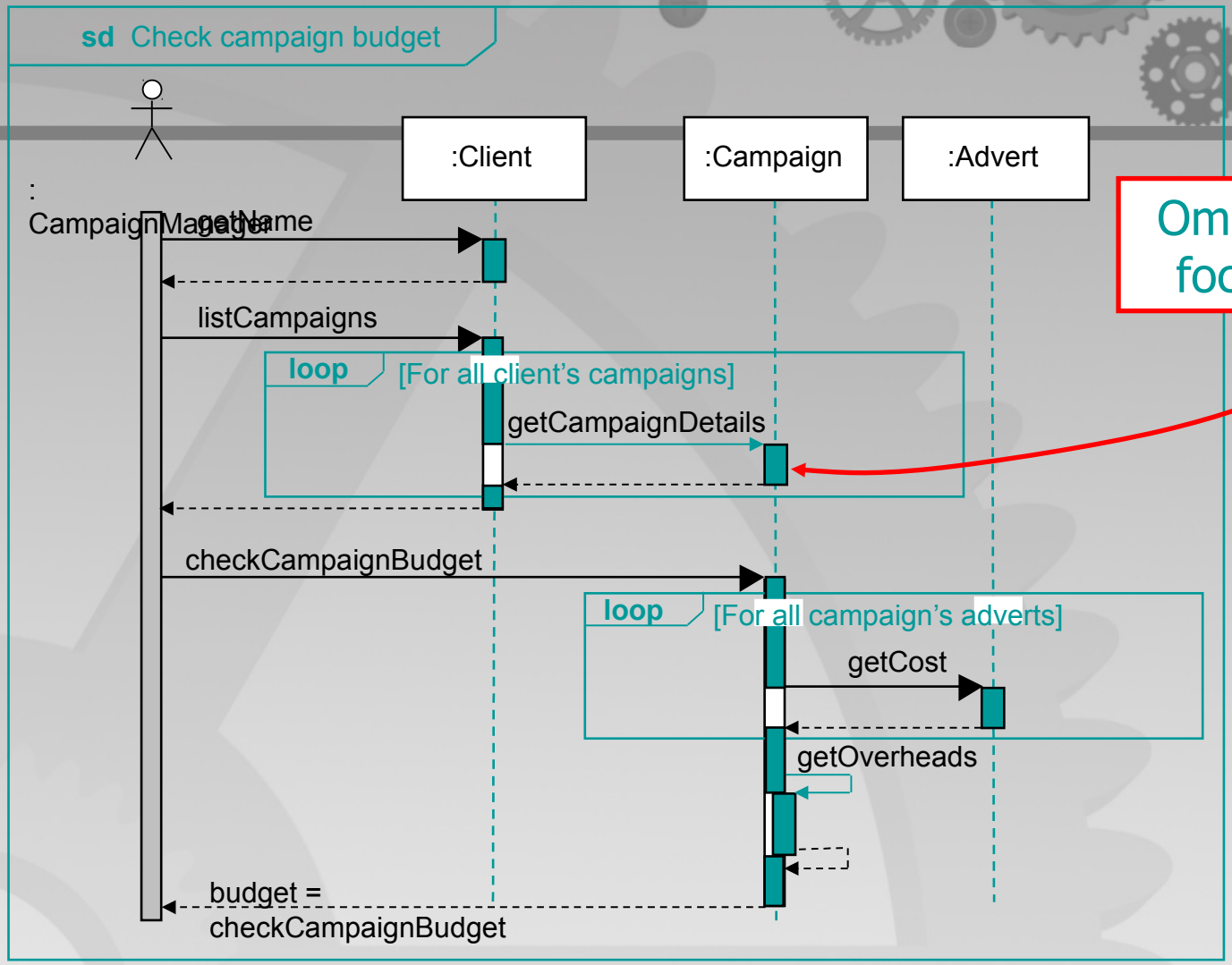
Focus du Contrôle

- Parfois il est nécessaire de modéliser des interactions plus complexes.
- Par exemple: plus de détails pour la synchronisation des messages
- Cela peut être obtenu par la notion de focus de contrôle

Focus du Contrôle

- Indique le temps d'une activation pendant lequel le traitement prend place dans cet objet.
- Les parties d'une activation qui ne sont pas dans le focus du contrôle représentent les périodes pendant lesquelles, par exemple, une opération attend la réponse d'un autre objet.
- Peut être signalé en ombrant les parties du rectangle d'activation qui correspondent au traitement actif par une opération.

Focus du Contrôle



Fragments combinés

- Un fragment combiné regroupe des ensembles de messages pour montrer un flot conditionnel.

- Nous utilisons des fragments combinés pour représenter :
  - Des alternatives,
  - Des options,
  - Des boucles

Fragments combinés

1. Alternatives

- Les alternatives montrent des choix mutuellement exclusifs entre deux ou plusieurs séquences de messages,

- Ils modélisent la logique classique : 'if…then…else'
  - Par exemple, **si** j'achète trois articles, **alors** j'obtiens 20% de réduction sur mes achats; **sinon** j'obtiens 10% de réduction.

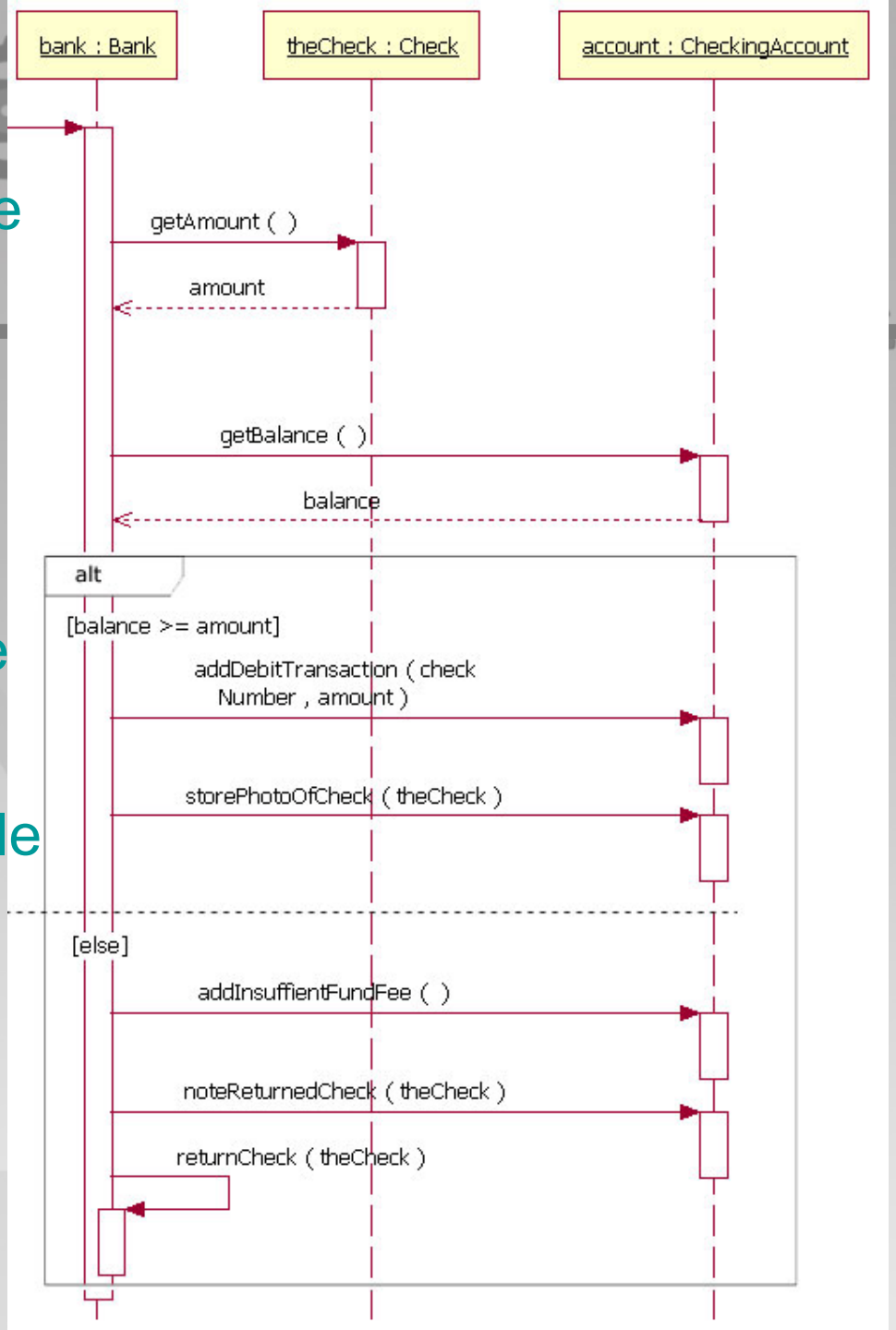
- Nous utilisons un cadre avec le mot "alt" au sommet.

- Le cadre est divisé en 'opérandes' séparés par une ligne en pointillés.

- Chaque opérande a une garde en haut d'une 'lifeline'.

- Si la garde d'un opérande égale « true », alors l'opérande est suivi.

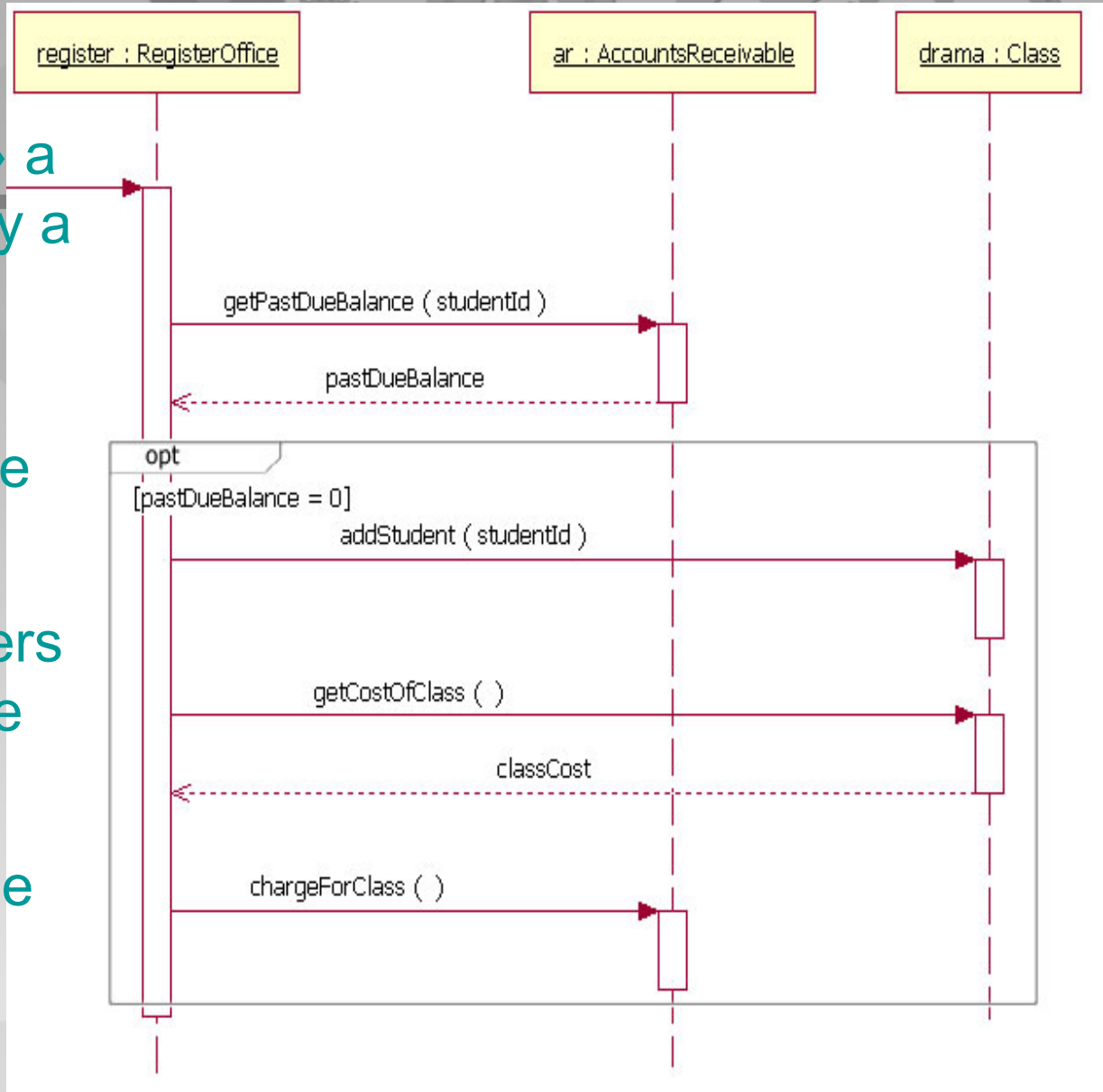
- Il peut y avoir autant de chemins alternatifs que nécessaire.



Fragments combinés

- 2. Options
- Utilisées pour modéliser une séquence qui, sous une certaine condition, se produira, sinon elle ne se produira pas.
- Une option est utilisée pour modéliser un simple "if then".

- Le fragment « option » a **un seul opérande** et il y a **jamais de garde « sinon »**.
- Nous utilisons un cadre avec "opt" au sommet.
- La garde est placée vers l'angle supérieur gauche en haut d'une 'lifeline'.
- Si la condition de garde n'est pas vraie, nous sautons juste ce cadre 'option'.



Fragments combinés

3. Boucles

- Utilisé pour modéliser une séquence qui est répétée.

• Utiliser un cadre avec "loop".

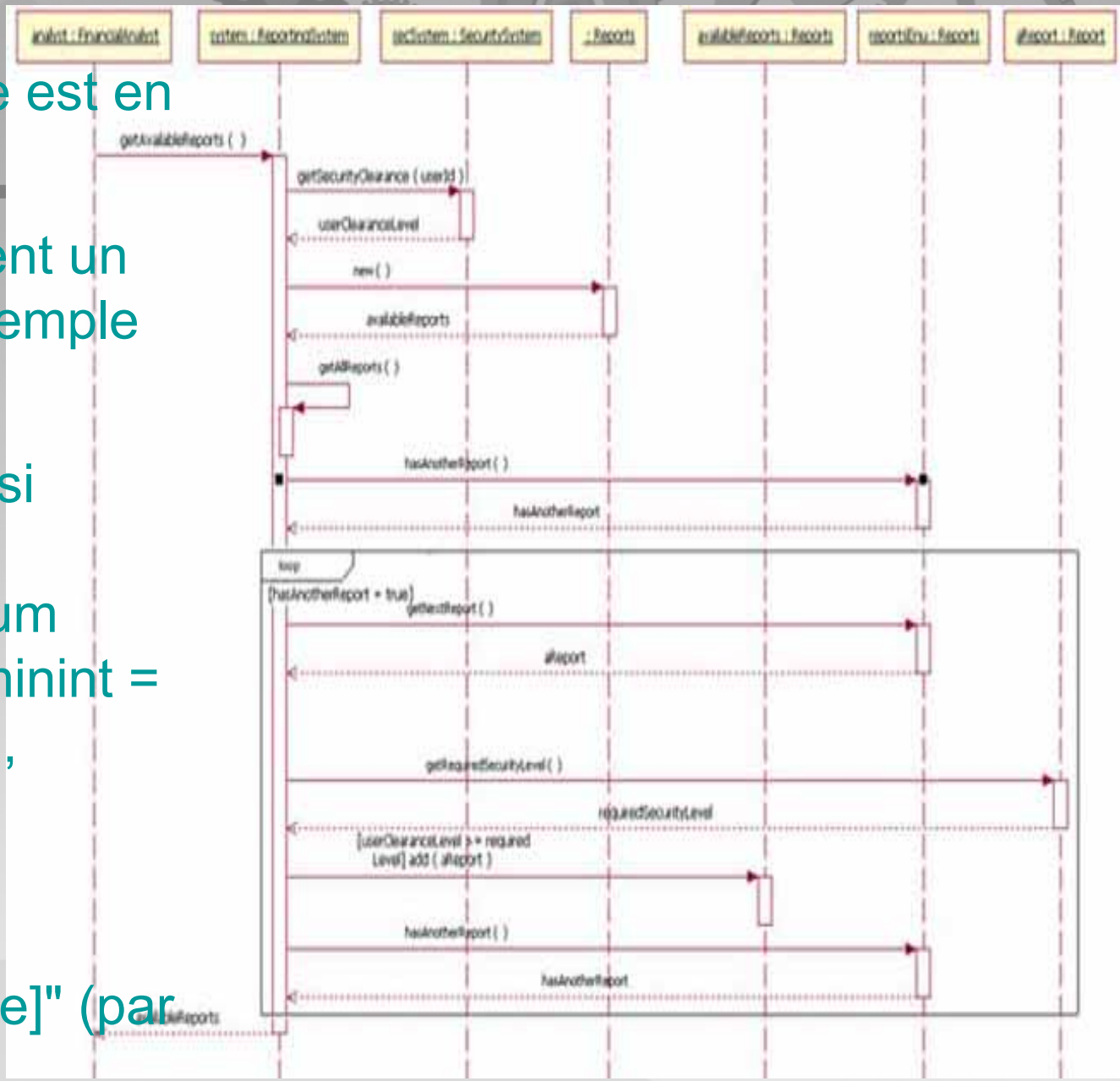
• La garde de boucle est en haut de la ‘lifeline’.

• La garde est souvent un test booléen (par exemple « if x is true do »)

• Nous pouvons aussi spécifier le nombre minimum ou maximum d’itérations avec : "minint = [le nombre]" (par ex., "minint = 1")

Et

•"maxint = [le nombre]" (par ex., "maxint = 5").



Fragments combinés

- D'autres fragments combinés existent aussi. Par exemple :
  - Reference (REF):

    Utilisé pour référencer un autre diagramme de séquence.
  - Break (BREAK):

    Communément utilisé pour modéliser la gestion des exceptions. Utilisé comme le mot-clé 'break' dans un langage de programmation
  - Parallel (PAR):

    montrent les activités conduites en parallèles.

Message de réponse

- Un *message de réponse* renvoie le contrôle à l'objet qui a généré le message qui a commencé l'activation.

- Les messages de réponses sont signalés par une flèche en pointillé, mais il est optionnel de les montrer puisqu'on peut supposer que le contrôle est rendu à l'objet d'origine à la fin de l'activation.

Guide pour les diagrammes de séquence

Guide pour les diagrammes de séquence

1. Décider à quel niveau vous modélisez l'interaction.

2. Identifier les éléments principaux impliqués dans l'interaction.

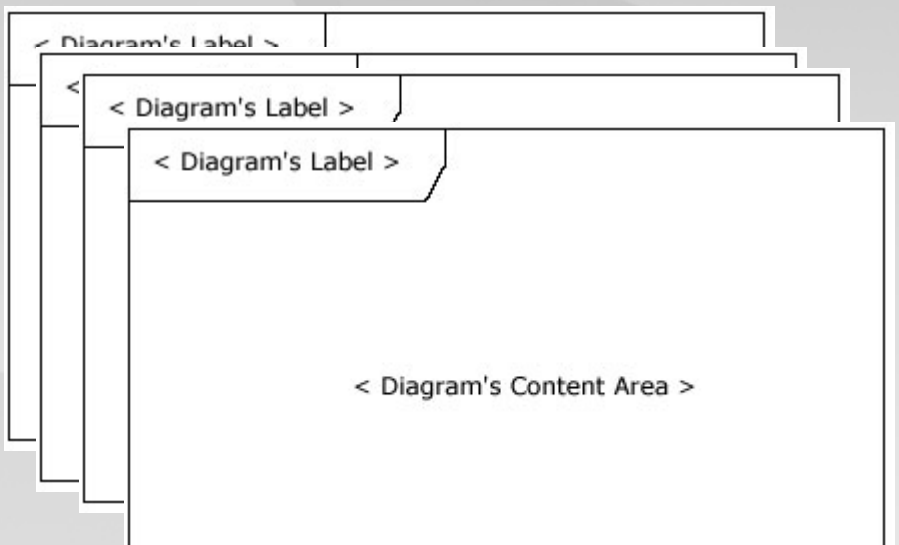
3. Considérer des scénarios alternatifs qui pourraient être nécessaires.

Guide pour les diagrammes de séquence

4. Dessiner la structure globale du diagramme.

5. Ajoutez l'interaction détaillée.

6. Vérifier la consistance avec les diagrammes de séquence liés et modifier si nécessaire.



7. Vérifier la consistance avec d'autres diagrammes UML.

Consistance de modèle

- L'allocation d'opérations à des objets doit être consistante avec le diagramme de classe et la signature de messages doit correspondre à la signature de l'opération.
  - Peut être forcé par des outils CASE.

- Chaque objet expéditeur doit avoir la référence d'objet pour l'objet de destination.
  - Soit une association existe entre les classes ou un autre objet passe la référence à l'expéditeur.
  - Ce sujet est crucial pour déterminer la conception d'associations.
  - Les chemins des messages doivent être analysés avec soin.

Consistance de modèle

- Toutes les formes de diagrammes de séquences utilisées doivent être consistantes.
- Les messages sur les diagrammes de séquence doivent être consistant avec la machine d'états pour les objets participants.
- Les changements implicites d'états dans les diagrammes de séquences doivent être consistant avec ceux modélisés explicitement dans la machine d'états.