

Diagrammes dynamiques en UML

diagrammes de séquence, de collaboration, d'état-transition,
d'activité

LIRMM / Université de Montpellier 2

22 novembre 2015

Introduction

Diagrammes dynamiques

Les machines à états

États et transitions
États initial et final
États composites
Pseudo-états

Les diagrammes d'activités

Les diagrammes de séquence

La ligne de vie

Les messages
Composition de fragments de diagrammes de séquence

- Diagrammes statiques (diagrammes d'instances et diagrammes de classe)
 - Structure d'un système
 - Signatures de méthodes
- Diagrammes dynamiques
 - Dynamique du système

Sommaire

Diagrammes dynamiques

Les machines à états

États et transitions

États initial et final

États composites

Pseudo-états

Les diagrammes d'activités

Les diagrammes de séquence

La ligne de vie

Les messages

Composition de fragments de diagrammes de séquence

1 Les machines à états

- États et transitions
- États initial et final
- États composites
- Pseudo-états

2 Les diagrammes d'activités

3 Les diagrammes de séquence

- La ligne de vie
- Les messages
- Composition de fragments de diagrammes de séquence

Sommaire

Diagrammes dynamiques

Les machines à états

États et transitions

États initial et final

États composites
Pseudo-états

Les diagrammes d'activités

Les diagrammes de séquence

La ligne de vie

Les messages
Composition de fragments de diagrammes de séquence

1 Les machines à états

- États et transitions
- États initial et final
- États composites
- Pseudo-états

2 Les diagrammes d'activités

3 Les diagrammes de séquence

- La ligne de vie
- Les messages
- Composition de fragments de diagrammes de séquence

Les machines à états

Diagrammes dynamiques

Les machines à états

États et transitions

États initial et final

États composites

Pseudo-états

Les diagrammes d'activités

Les diagrammes de séquence

La ligne de vie

Les messages

Composition de fragments de

diagrammes de séquence

Les machines à états, aussi appelés diagrammes d'état-transition, servent à modéliser la dynamique d'un sous-système, souvent d'une classe. Une machine à états décrivant le comportement d'une classe décrit en fait la dynamique de toutes ses instances à la réception ou à l'envoi de messages.

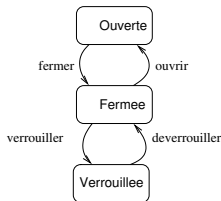


Figure: Diagramme d'état-transition très simple pour une porte

États et transitions

Diagrammes dynamiques

Les machines à états

États et transitions

États initial et final

États composites

Pseudo-états

Les diagrammes d'activités

Les diagrammes de séquence

La ligne de vie

Les messages

Composition de fragments de diagrammes de séquence

- Un état modélise une situation où un certain invariant (généralement implicite) est maintenu (la porte est fermée, un compte bancaire a un solde positif, ...)
- Transition : passage d'un état à un autre
 - Il peut y avoir plusieurs événements déclencheurs possibles, auquel cas on les liste tous (en les séparant par des virgules).
 - L'action peut être une affectation d'attribut, un appel de méthode, ...
 - Quand aucun événement déclencheur n'est spécifié, la transition est dite spontanée.

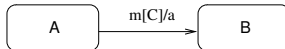


Figure: Une transition

États initial et final

Diagrammes dynamiques

Les machines à états

États et transitions

États initial et final

États composites Pseudo-états

Les diagrammes d'activités

Les diagrammes de séquence

La ligne de vie

Les messages

Composition de fragments de diagrammes de séquence

Un pseudo-état initial (noté graphiquement par un petit disque noir) représente un sommet qui est la source d'une seule transition vers l'état "par défaut" d'une machine à état ou d'un état composite. La transition initiale peut être munie d'une action.

L'état final matérialise le fait qu'une région (une machine à état ou une région d'état composite) est "terminée" (voir notation figure 3).



Figure: États initial et Final

États composites

Diagrammes dynamiques

Les machines à états

États et transitions

États initial et final

États composites
Pseudo-états

Les diagrammes d'activités

Les diagrammes de séquence

La ligne de vie

Les messages

Composition de fragments de diagrammes de séquence

Un état composite :

- soit contient une seule région
- soit est décomposé en 2 ou plusieurs régions orthogonales

Un état inclus dans une région d'un état composite est appelé un sous-état de cet état composite. C'est un sous-état direct quand il n'est pas contenu par un autre état, et sinon un sous-état indirect.

Exemple de machine à état avec état composite

Diagrammes dynamiques

Les machines à états

États et transitions

États initial et final

États composites

Pseudo-états

Les diagrammes d'activités

Les diagrammes de séquence

La ligne de vie

Les messages

Composition de fragments de diagrammes de séquence

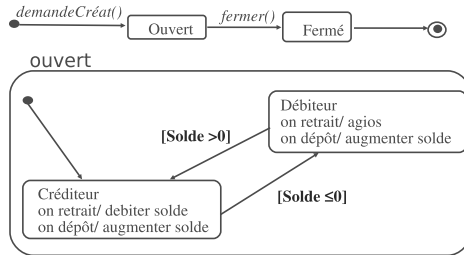


Figure: Exemple de machine à état avec état composite

Exemple avec état à régions orthogonales

Diagrammes dynamiques

Les machines à états

États et transitions

États initial et final

États composites

Pseudo-états

Les diagrammes d'activités

Les diagrammes de séquence

La ligne de vie

Les messages

Composition de fragments de diagrammes de séquence

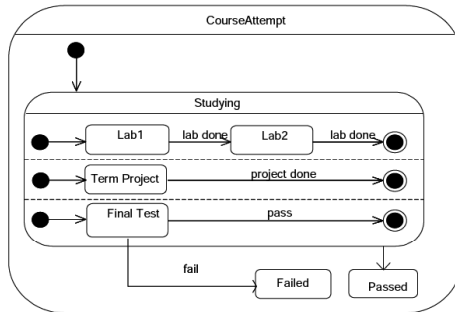


Figure: État composite orthogonal, extrait du document de spécification d'UML 2.0

Comportement d'entrée et de sortie, comportement dans un état

Diagrammes dynamiques

Les machines à états

États et transitions

États initial et final

États composites

Pseudo-états

Les diagrammes d'activités

Les diagrammes de séquence

La ligne de vie

Les messages

Composition de fragments de diagrammes de séquence

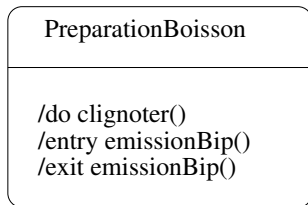


Figure: Actions d'entrée et de sortie des états, comportement dans les états

+ on evenement / action

États Historiques

Diagrammes dynamiques

Les machines à états

États et transitions

États initial et final

États composites

Pseudo-états

Les diagrammes d'activités

Les diagrammes de séquence

La ligne de vie

Les messages

Composition de fragments de diagrammes de séquence

Il existe des états dits “mémoire” qui permettent de rerentrer dans un état composite dans le même sous-état que quand on en est sorti. Il y a deux états mémoire : historique superficiel et historique profond.

Historique superficiel (Shallow history) (noté H). L'historique superficiel représente le sous-état actif le plus récent (mais pas les sous-états de ce sous-état).

Historique profond (Deep history) (noté H*). L'historique profond représente la configuration active la plus récente de l'état composite qui contient directement l'historique profond (c'est-à-dire la configuration active la dernière fois qu'on a quitté l'état composite).



Shallow history



DeepHistory

Autres pseudo-états

Diagrammes dynamiques

Les machines à états

États et transitions

États initial et final

États composites

Pseudo-états

Les diagrammes d'activités

Les diagrammes de séquence

La ligne de vie

Les messages

Composition de fragments de diagrammes de séquence

Il existe d'autres pseudo-états comme les jonctions, les choix ou les branchements, nous ne les détaillerons pas ici.

Question de cours

Diagrammes dynamiques

Les machines à états

États et transitions

États initial et final

États composites

Pseudo-états

Les diagrammes d'activités

Les diagrammes de séquence

La ligne de vie

Les messages

Composition de fragments de diagrammes de séquence

Nous étudions une montre très simple. Elle possède deux boutons : avance et mode. Le mode par défaut est le mode affichage. Quand on appuie une fois sur le bouton mode, la montre passe en mode de modification des heures. Chaque pression sur le bouton avance incrémente l'heure d'une unité. Quand on appuie une nouvelle fois sur le bouton mode, la montre passe en modification des minutes. Chaque pression sur le bouton avance incrémente les minutes d'une unité. Quand on appuie une nouvelle fois sur le bouton mode, la montre repasse en mode affichage.

→ **Représentez le diagramme d'états de la montre.**

Sommaire

Diagrammes dynamiques

Les machines à états

États et transitions
États initial et final
États composites
Pseudo-états

Les diagrammes d'activités

Les diagrammes de séquence
La ligne de vie
Les messages
Composition de fragments de diagrammes de séquence

- 1 Les machines à états
 - États et transitions
 - États initial et final
 - États composites
 - Pseudo-états
- 2 Les diagrammes d'activités
- 3 Les diagrammes de séquence
 - La ligne de vie
 - Les messages
 - Composition de fragments de diagrammes de séquence

Les diagrammes d'activité

Diagrammes dynamiques

Les machines à états

États et transitions

États initial et final

États composites
Pseudo-états

Les diagrammes d'activités

Les diagrammes de séquence

La ligne de vie

Les messages

Composition de fragments de diagrammes de séquence

Les diagrammes d'activités permettent de représenter des flots de contrôle et de données. Ils permettent donc par exemple de représenter le comportement d'une opération ou d'un cas d'utilisation. Les diagrammes d'activité sont des graphes, avec différents types de nœuds et d'arcs. Ils mettent en jeu principalement :

- des nœuds actions
- des nœuds de contrôle permettant de spécifier l'enchaînement des actions (synchronisation, branchement, ...)
- des nœuds d'objet permettant de représenter les objets créés ou utilisés au cours d'une activité
- des arcs de transition permettant de relier les nœuds.

Exemple

Diagrammes dynamiques

Les machines à états

États et transitions

États initial et final

États composites

Pseudo-états

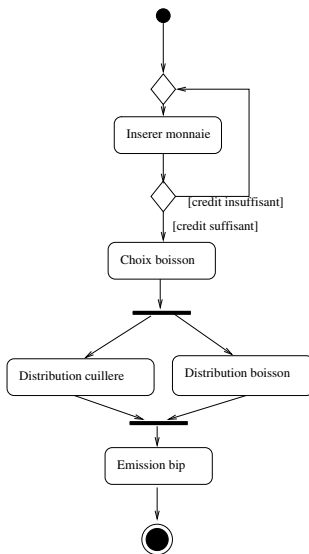
Les diagrammes d'activités

Les diagrammes de séquence

La ligne de vie

Les messages

Composition de fragments de diagrammes de séquence



Représentation de graphes de flot de contrôle

Diagrammes dynamiques

Les machines à états

États et transitions

États initial et final

États composites Pseudo-états




Les diagrammes d'activités

Les diagrammes de séquence

La ligne de vie

Les messages

Composition de fragments de diagrammes de séquence

- Nœud initial (Initial node) . Point d'entrée pour invoquer une activité. Un jeton de contrôle est placé au nœud initial quand l'activité commence.
- Nœud de fin d'activité (Activity final node) . Stoppe tous les flots dans une activité. Un jeton atteignant un nœud de fin d'activité fait avorter tous les flots en cours, l'activité est donc terminée et le jeton est détruit (ainsi que tous les jetons circulant dans l'activité).
- Nœud de fin de flot (Flow Final node) . Termine un flot. Le nœud de flot final détruit les jetons y entrant.

Représentation de graphes de flot de contrôle

Diagrammes dynamiques

Les machines à états

États et transitions

États initial et final

États composites

Pseudo-états



Les diagrammes d'activités

Les diagrammes de séquence

La ligne de vie

Les messages

Composition de fragments de diagrammes de séquence

- Nœud d'action (Action node) . Unité fondamentale de la fonctionnalité exécutable d'une activité. Une action s'exécute quand toutes les contraintes sur ses *flots de contrôle* entrants sont satisfaites (jonction implicite). L'exécution consomme les jetons de contrôle entrants puis présente un jeton sur chaque flot sortant (branchement implicite).
- Flot de contrôle (Control flow) . Passage des jetons. Les jetons offerts par le nœud source sont offerts au nœud destination.

Représentation de graphes de flot de contrôle

Diagrammes dynamiques

Les machines à états

États et transitions

États initial et final

États composites

Pseudo-états


Les diagrammes d'activités


Les diagrammes de séquence

La ligne de vie

Les messages

Composition de fragments de diagrammes de séquence

- Nœud de décision (Decision node)  . Choix parmi les flots sortants. Chaque jeton arrivant sur un nœud de décision ne peut traverser qu'un seul flot sortant. Les jetons ne sont pas dupliqués. Ce sont les gardes sur les flots sortants qui permettent le choix (les gardes doivent assurer le déterminisme du choix).

- Nœud de branchement (Fork node)  . Partage d'un flot en flots concurrents. Les jetons arrivant d'un branchement sont dupliqués sur les flots sortants.

Représentation de graphes de flot de contrôle


Diagrammes dynamiques


Les machines à états

États et transitions
États initial et final
États composites
Pseudo-états

Les diagrammes d'activités

Les diagrammes de séquence
La ligne de vie
Les messages
Composition de fragments de diagrammes de séquence

- Nœud de jonction (Join node)  . Synchronisation de plusieurs flots. Si un jeton de contrôle est offert sur chaque flot entrant, alors un jeton de contrôle est offert sur le flot sortant.

- Nœud de fusion (Merge node)  . Rassemblement de plusieurs flots. Tous les jetons offerts sur les flots entrants sont offerts sur le flot sortant sans synchronisation.
- Partition d'activité (Activity Partition). Identifie des actions ayant une caractéristique commune. Les partitions



n'affectent pas le flot des jetons.

Sommaire

Diagrammes dynamiques

Les machines à états

États et transitions
États initial et final
États composites
Pseudo-états

Les diagrammes d'activités

Les diagrammes de séquence

La ligne de vie
Les messages
Composition de fragments de diagrammes de séquence

- 1 Les machines à états
 - États et transitions
 - États initial et final
 - États composites
 - Pseudo-états
- 2 Les diagrammes d'activités
- 3 Les diagrammes de séquence
 - La ligne de vie
 - Les messages
 - Composition de fragments de diagrammes de séquence

Les diagrammes de séquence

Diagrammes dynamiques

Les machines à états

États et transitions
États initial et final
États composites
Pseudo-états

Les diagrammes d'activités

Les diagrammes de séquence

La ligne de vie
Les messages
Composition de fragments de diagrammes de séquence

- Les diagrammes de séquence permettent de représenter les interactions entre des instances particulières. Un diagramme met en jeu :
 - des instances, et éventuellement des acteurs,
 - des messages échangés par ces instances. Un message définit une communication entre instances. Ce peut être par exemple l'émission d'un signal, ou l'appel d'une opération.
- Le diagramme de séquence permet d'insister sur la chronologie des interactions : le temps s'écoule grosso modo du haut vers le bas.
- Les diagrammes de séquence ont été profondément modifiés lors du passage d'UML1.x à UML2.0, et à l'heure actuelle, peu de gens utilisent la nouvelle notation

Exemple

Diagrammes dynamiques

Les machines à états

États et transitions

États initial et final

États composites
Pseudo-états

Les diagrammes d'activités

Les diagrammes de séquence

La ligne de vie

Les messages
Composition de fragments de diagrammes de séquence

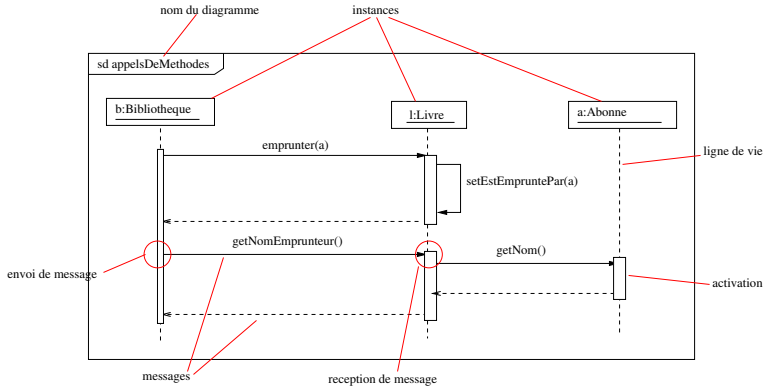


Figure: Premier exemple de diagramme de séquence

La ligne de vie

Diagrammes dynamiques

Les machines à états

États et transitions

États initial et final

États composites
Pseudo-états

Les diagrammes d'activités

Les diagrammes de séquence

La ligne de vie

Les messages
Composition de fragments de diagrammes de séquence

- À chaque instance est associée une ligne de vie, qui représente la vie de l'objet.
- Les événements survenant sur une ligne de vie (réception de message ou envoi de message) sont ordonnés chronologiquement.
- La ligne de vie est représentée par une ligne pointillée quand l'instance est inactive, et par une boîte blanche ou grisée quand l'instance est active.
- Quand une instance est détruite, on stoppe la ligne de vie par une croix.

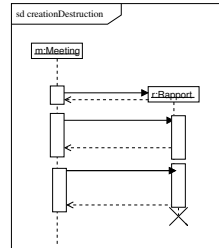


Figure: Ligne de vie

Les messages

Diagrammes dynamiques

Les machines à états

États et transitions
États initial et final
États composites
Pseudo-états

Les diagrammes d'activités

Les diagrammes de séquence

La ligne de vie

Les messages

Composition de fragments de diagrammes de séquence

- Les messages sont représentés par des lignes fléchées.
- À chaque extrémité de la ligne fléchée correspond un événement (réception ou envoi). Le sens de la flèche permet de déterminer dans quel sens va le message.
- Messages synchrones et asynchrones (voir Figure 11).

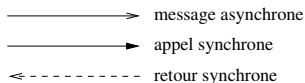


Figure: Messages

Exemple

Diagrammes dynamiques

Les machines à états

États et transitions

États initial et final

États composites
Pseudo-états

Les diagrammes d'activités

Les diagrammes de séquence

La ligne de vie

Les messages

Composition de fragments de diagrammes de séquence

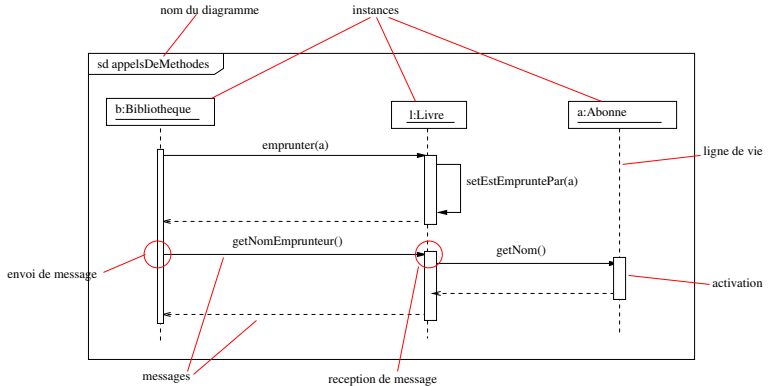


Figure: Premier exemple de diagramme de séquence

Syntaxe des noms de message

Diagrammes dynamiques

Les machines à états

États et transitions

États initial et final

États composites

Pseudo-états

Les diagrammes d'activités

Les diagrammes de séquence

La ligne de vie

Les messages

Composition de fragments de diagrammes de séquence

La syntaxe pour le nom d'un message est :

```
([attribut =] signal-ou-NomOperation ([liste-arguments]))[: valeur-retour]) | *
```

où la syntaxe pour un argument est :

```
([nomParam =] valeur-argument) | (attribut = nomParamOut [: valeurArgument]) | -
```

* signifie : n'importe quel type de message

- signifie : paramètre indéfini

Par exemple, on peut avoir les noms de message suivants :

- `getAge()`
- `getAge() :12`
- `age=getAge() :12`
- `setAge(age=15)`
- `setAge(-)`

Exemple d'appel de méthode

Diagrammes dynamiques

Les machines à états

États et transitions

États initial et final

États composites

Pseudo-états

Les diagrammes d'activités

Les diagrammes de séquence

La ligne de vie

Les messages

Composition de fragments de diagrammes de séquence

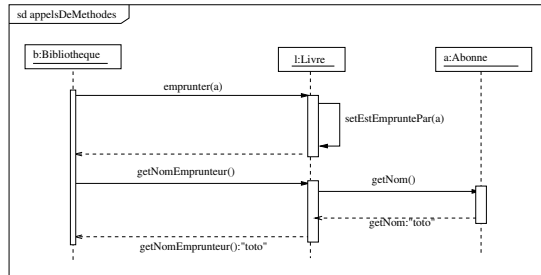


Figure: Appels de méthodes

Les diagrammes de séquence ne sont pas à concevoir indépendamment des autres diagrammes, comme par exemple le diagramme de classes.

Composition de fragments de diagrammes de séquence

Diagrammes dynamiques

- possible depuis la version 2.0 d'UML
- existence de plusieurs opérateurs de composition

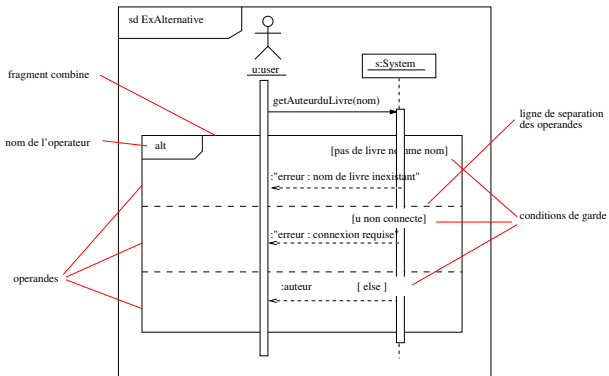


Figure: Opérateurs de composition et fragments combinés

Alternative et optionnalité

Diagrammes dynamiques

Les machines à états

États et transitions

États initial et final

États composites
Pseudo-états

Les diagrammes d'activités

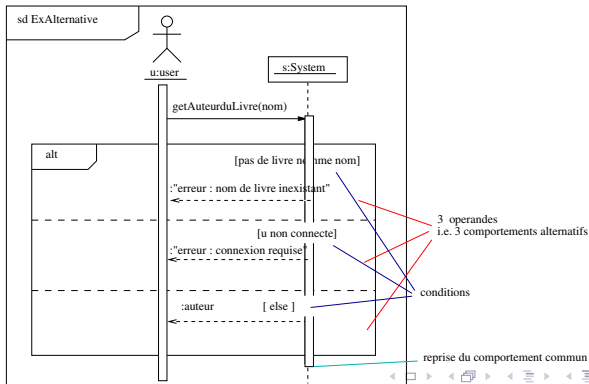
Les diagrammes de séquence

La ligne de vie

Les messages

Composition de fragments de diagrammes de séquence

- alternative (noté alt) permet de représenter le choix (exclusif) entre plusieurs comportements
- optionnalité (noté opt) permet de représenter un comportement qui n'a lieu que si une condition de garde est vraie



Composition parallèle

Diagrammes dynamiques

Les machines à états

États et transitions

États initial et final

États composites

Pseudo-états

Les diagrammes d'activités

Les diagrammes de séquence

La ligne de vie

Les messages

Composition de fragments de diagrammes de séquence

- L'opérateur de composition parallèle (noté par) permet de spécifier des comportements qui peuvent avoir lieu en parallèle les uns des autres. Cet opérateur est n-aire.
- Quand un comportement A est en parallèle avec un comportement B, l'ordre partiel des événements de A et de B est conservé.
- Raccourci syntaxique : *corégion*

Composition séquentielle faible

Diagrammes dynamiques

Les machines à états

États et transitions

États initial et final

États composites

Pseudo-états

Les diagrammes d'activités

Les diagrammes de séquence

La ligne de vie

Les messages

Composition de fragments de diagrammes de séquence

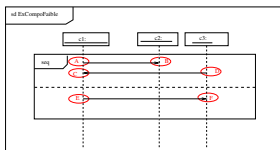


Figure: Composition séquentielle faible (seq)

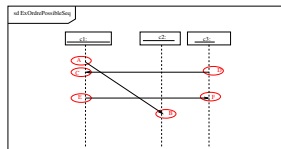


Figure: Un diagramme de séquence pouvant en résulter

- $A \prec B$
- $D \prec C$
- $E \prec F$
- $C \prec E$
- $D \prec F$

Composition séquentielle forte

Diagrammes dynamiques

Les machines à états

États et transitions

États initial et final

États composites

Pseudo-états

Les diagrammes d'activités

Les diagrammes de séquence

La ligne de vie

Les messages

Composition de fragments de diagrammes de séquence

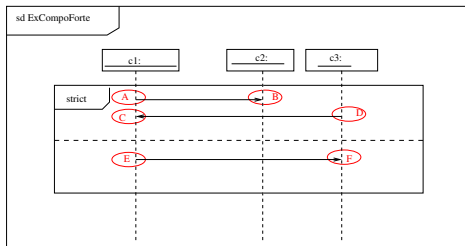


Figure: Composition séquentielle forte (strict)

- $A \prec B$
- $D \prec C$
- $E \prec F$
- $C \prec E$
- $D \prec F$
- $B \prec E$

Boucle

Diagrammes dynamiques

Les machines à états

États et transitions
États initial et final
États composites
Pseudo-états

Les diagrammes d'activités

Les diagrammes de séquence

La ligne de vie

Les messages
Composition de fragments de diagrammes de séquence

L'opérateur loop permet d'itérer des comportements. On doit pour cela spécifier :

- le nombre minimum `minInt` de tours de boucles,
- le nombre maximum `maxInt` de tours de boucle (* signifie infini),
- une condition de garde,
- une unique opérande représentant le comportement sur lequel on boucle.

Syntaxe de la boucle :

```
loop[ (minInt [ , maxInt ] ) ]
```

Par défaut, `minInt=0` et `maxInt=*..`

Exemple de boucle

Diagrammes dynamiques

Les machines à états

États et transitions

États initial et final

États composites
Pseudo-états

Les diagrammes d'activités

Les diagrammes de séquence

La ligne de vie

Les messages
Composition de fragments de diagrammes de séquence

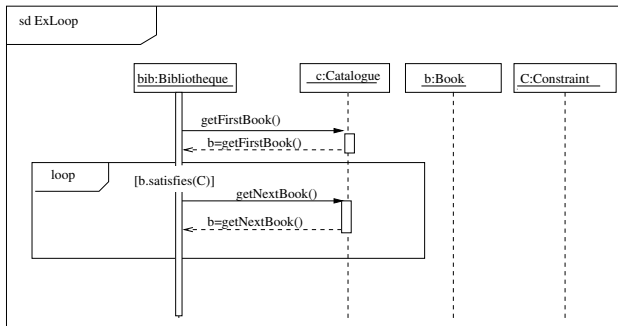


Figure: Boucles dans les diagrammes de séquence

Question de cours

Diagrammes dynamiques

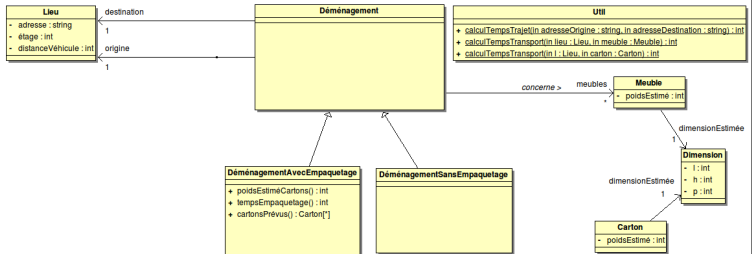
Les machines à états

États et transitions
États initial et final
États composites
Pseudo-états

Les diagrammes d'activités

Les diagrammes de séquence

La ligne de vie
Les messages
Composition de fragments de diagrammes de séquence



→ Une méthode qui calcule le temps de déménagement d'un déménagement avec empaquetage : temps de trajet + temps d'empaquetage + distance au véhicule de l'origine et de la destination.