

Conception des systèmes d'information

Chapitre 5 : diagrammes de séquence

Mariem MAHFOUDH

mariem.mahfoudh@gmail.com

2 LSI-ADBD, ISIMS, 2023-2024



Ce cours a été construit en se basant sur les références suivantes :

- ▶ Livre "UML 2 pratique pour la modélisation", Benoît Charroux, Aomar Osmani et Yann Thierry-Mieg
- ▶ Livre "UML 2 pour la pratique", Pascal Roques
- ▶ Livre " UML 2 pour les developpeurs", Xavier Blanc et Isabelle Mounier
- ▶ Livre "UML 2 de l'apprentissage à la pratique", Laurent Audibert
- ▶ Livre "Modélisation Objet avec UML", Pr. Pierre Alain Muller
- ▶ Cours " Conception des systèmes d'information", Pr. Faiez Gargouri
- ▶ Cours " Analyse, Conception Objet", Stephane Galland
- ▶ Cours " Cours Conception Orientée Objet", Dr. Faiza Ghazzi
- ▶ Cours " Langage UML", Emmanuel Remy

- 1 Introduction
- 2 Définition
- 3 Concepts des diagrammes de séquence
- 4 Conclusion
- 5 Exercice

Introduction

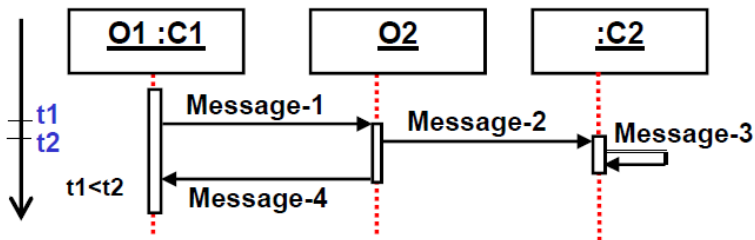
- ▶ Les **diagrammes de cas d'utilisation** modélisent à **quoi** sert le système, en organisant les interactions possibles avec les acteurs.
- ▶ Les **diagrammes de classes** permettent de spécifier la structure et les liens entre les objets dont le système est composé : ils spécifie **qui** sera à l'oeuvre dans le système pour réaliser les fonctionnalités décrites par les diagrammes de cas d'utilisation.
- ▶ Les **diagrammes de séquence** (DS) permettent de décrire **quand et comment** les éléments du système interagissent entre eux et avec les acteurs.
 - Les objets au coeur d'un système interagissent en s'échangeant des messages.
 - Les acteurs interagissent avec le système au moyen d'IHM (Interfaces Homme-Machine).

Diagramme de séquence

- ▶ Le diagramme de séquence fait partie des **diagrammes comportementaux (dynamique)** et plus précisément des **diagrammes d'interactions**.
- ▶ Il permet de représenter des échanges entre les différents objets et acteurs du système en fonction du temps.
- ▶ A moins que le système à modéliser soit extrêmement simple, nous ne pouvons pas modéliser la dynamique globale du système dans un seul diagramme.
 - Nous ferons donc appel à un ensemble de diagrammes de séquences chacun correspondant à **un cas d'utilisation**.

Concepts

- ▶ L'objet.
- ▶ La ligne de vie.
- ▶ Les messages.
- ▶ Les fragments d'interactions combinés.

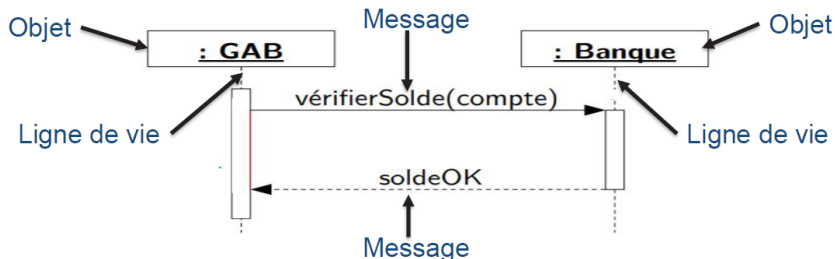


Représentation générale d'un diagramme de séquence

Concepts des diagrammes de séquence

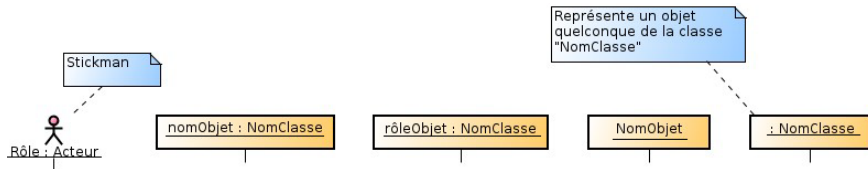
Concepts

- ▶ L'objet.
- ▶ La ligne de vie.
- ▶ Les messages.
- ▶ Les fragments d'interactions combinés.



L'objet

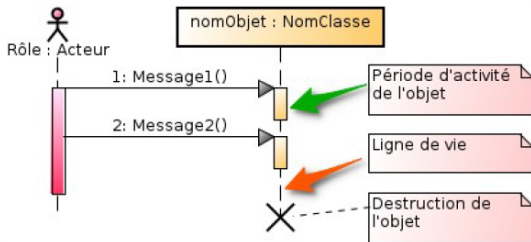
Les diagrammes de séquences représentant les échanges entre les objets mais aussi les échanges avec les acteurs (la représentation du stickman).



Notation des noms des objets

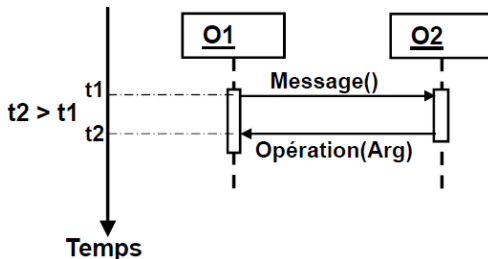
La ligne de vie

- ▶ A chaque objet est associé une ligne de vie (en trait pointillés à la verticale de l'objet) qui peut être considéré comme un axe temporel (le temps s'écoule du haut vers le bas).
- ▶ La ligne de vie indique les périodes d'activité de l'objet.
- ▶ Une période d'activité correspond au temps pendant lequel un objet effectue une action, soit directement, soit par l'intermédiaire d'un autre objet.
- ▶ Lorsque l'objet est détruit, la ligne de vie s'achève par un croix.



Les messages

- ▶ Un message définit une communication particulière entre des lignes de vie, plus précisément d'un objet vers un autre objet.
- ▶ La réception d'un message est considérée par l'objet récepteur comme un événement qu'il faut traiter (ou pas).
- ▶ La numérotation des messages est optionnelle. Elle est plutôt remplacée par l'ordre d'apparition des messages (ligne verticale).

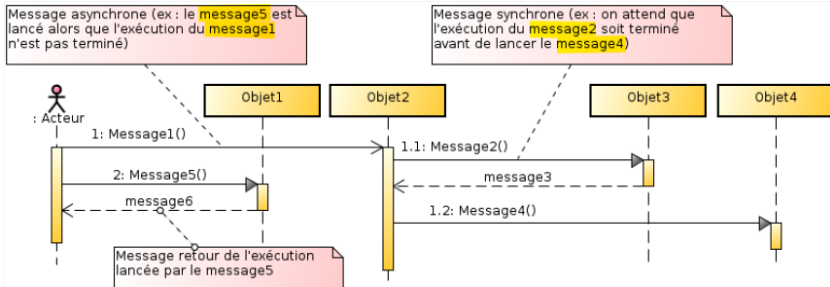


Concepts des diagrammes de séquence

Les messages

Plusieurs types de messages existent, les plus communs sont :

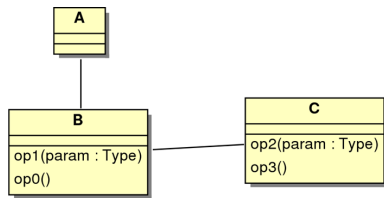
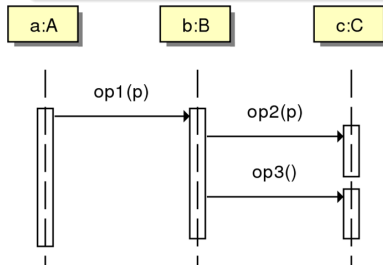
- ▶ messages synchrones ;
- ▶ messages asynchrones ;
- ▶ message de retour ;
- ▶ message de construction et destruction.



Concepts des diagrammes de séquence

Messages synchrones

- ▶ Les messages synchrones correspondent à **des opérations** dans le diagramme de classes.
- ▶ L'expéditeur du message reste bloqué pendant toute l'exécution de la méthode et attend donc la fin de celle-ci avant de pouvoir lancer un nouveau message.
- ▶ Graphiquement, les messages synchrones sont représentés par une flèche avec un **triangle plein** à son extrémité.



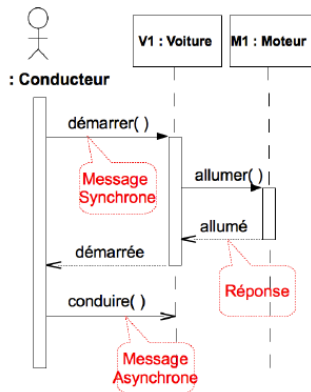
Concepts des diagrammes de séquence

Messages asynchrones

- ▶ L'expéditeur **n'attend pas** la fin de l'activation de la méthode invoquée chez le destinataire.
- ▶ Graphiquement, ils sont représentés par **une flèche simple**.

Un message asynchrone peut être :

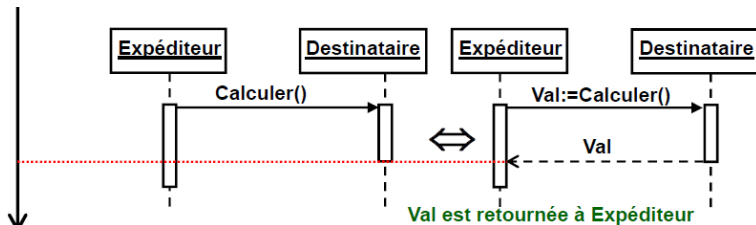
- ▶ Un appel de méthode : fréquent dans un système multi-threads (multi-tâche). L'objet expéditeur n'étant pas bloqué pendant l'exécution de la méthode, il peut continuer ainsi à envoyer d'autres messages.
- ▶ **Un signal** (cas le plus fréquent) : l'objet expéditeur transmet juste une information à l'objet destinataire.



Concepts des diagrammes de séquence

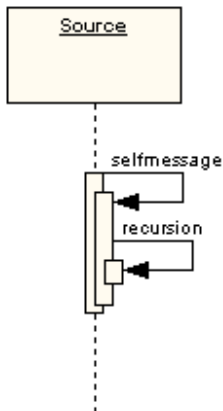
Messages de retour (réponses)

- ▶ Si une méthode qui a été activée (par un message) doit retourner des valeurs à la fin de son activation, cela se fait par un message retour.
- ▶ Le message de retour n'est donc pas un appel de méthode (il ne provoque donc pas l'activation d'un objet).
- ▶ Le message retour porte souvent le nom de l'élément retourné.
- ▶ Graphiquement, ils sont représentés par une **simple flèche en pointillés**.



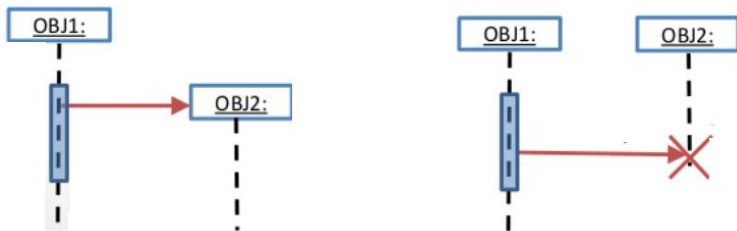
Messages réflexif

Un message d'auto (réflexif) peut représenter un appel récursif d'une opération, ou une méthode appelant une autre méthode appartenant au même objet.



Création et destruction d'objet

- ▶ Si un objet est créé au cours de l'exécution d'un scénario, celui-ci n'apparaît qu'au moment où il est créé.
- ▶ La création d'un objet est matérialisée par une flèche qui pointe sur le sommet d'une ligne de vie.
- ▶ La destruction d'un objet est matérialisée par **une croix (X)** qui marque la fin de la ligne de vie de l'objet.



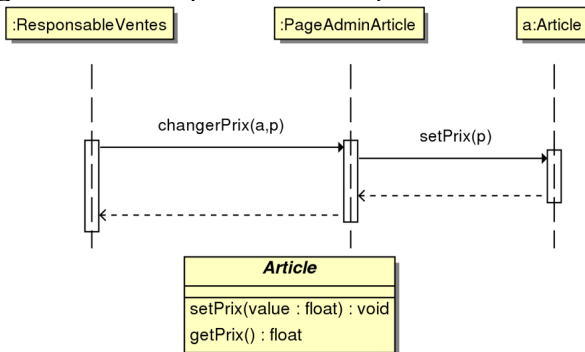
Concepts des diagrammes de séquence

Exemple

Cas d'utilisation :

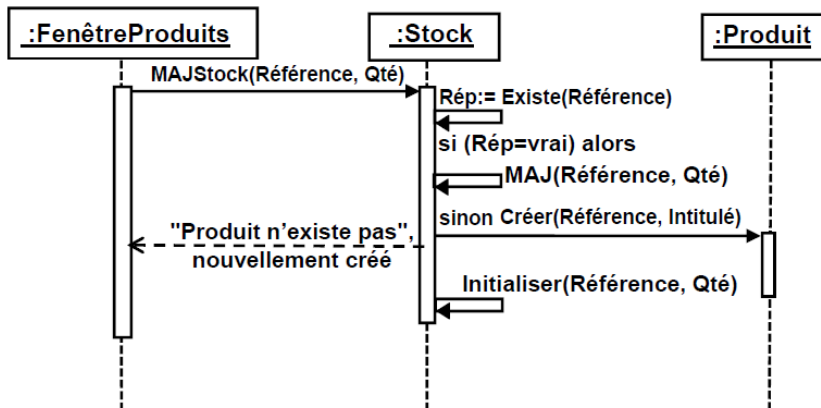


Diagramme de séquences correspondant :



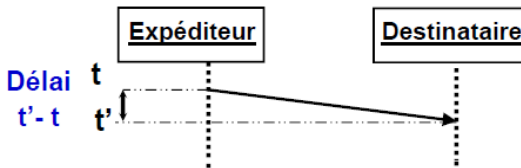
Exemple

Diagramme de séquence de CU " Mise à jour de stock".



Contraintes temporelles

- ▶ Des repères temporels avec des contraintes peuvent être placés le long de la ligne de vie.
- ▶ Un message avec un temps de propagation non négligeable peut être représenté par une flèche oblique ou en l'écrivant explicitement.

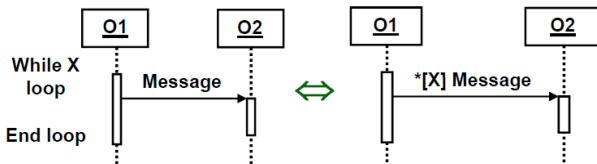


Concepts des diagrammes de séquence

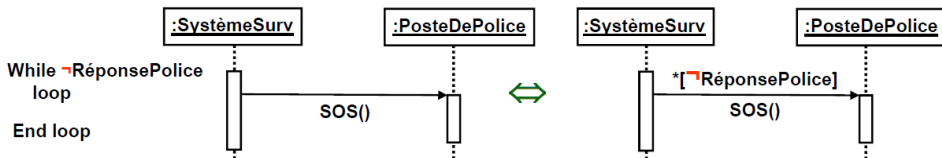
Structures de contrôle

Les diagrammes de séquence peuvent être complétés par des indications textuelles exprimées sous forme de texte libre ou pseudo-code.

Représentation de la boucle While



X = condition(s)



Concepts des diagrammes de séquence

Structures de contrôle

Branchements conditionnels

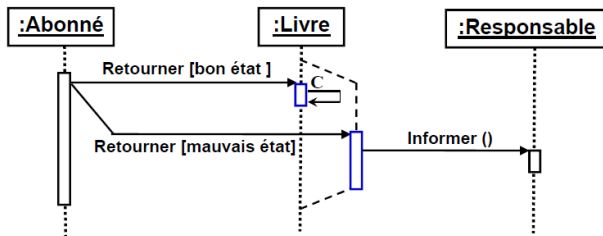
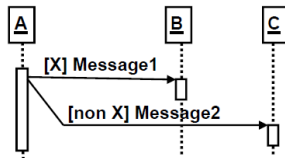
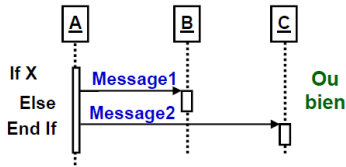
Si X = Vrai Alors

Envoyer(A, B, Message1)

Sinon

Envoyer(A, C, Message2)

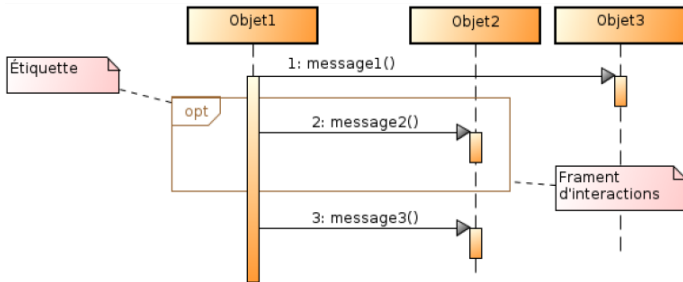
FINSI



C : Changer l'état du livre : disponible

Fragments d'interactions combinés

- ▶ Un fragment d'interactions est une partie du diagramme de séquence (délimitée par un rectangle) associée à une étiquette (dans le coin supérieur gauche).
- ▶ L'étiquette contient un **opérateur d'interaction** qui permet de décrire des modalités d'exécution des messages à l'intérieur du cadre.



Fragments d'interactions combinés

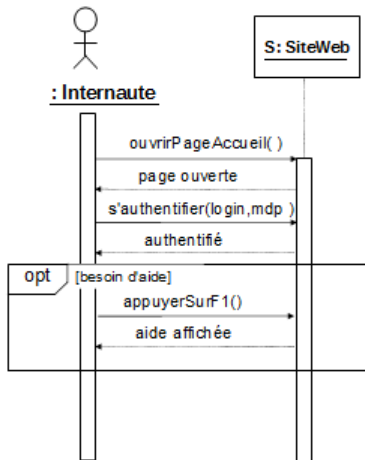
Exemples d'opérateurs :

- ▶ **alt** : opérateur conditionnel à plusieurs opérandes (équivalent d'un bloc if ... else if ... else en programmation) ;
- ▶ **opt** : opérateur conditionnel à une seule opérande (if sans else) ;
- ▶ **loop** : opérateur répétitif acceptant au max deux valeurs min et max ;
- ▶ **ref** : opérateur indiquant une référence vers un autre diagramme de séquence existant.
- ▶ **break** : si ce fragment est exécuté, le reste de la séquence est abandonné.

Concepts des diagrammes de séquence

Fragments d'interactions combinés (opt)

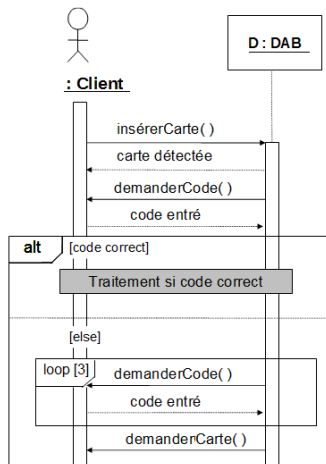
- Option (ou opt) représente un comportement qui peut se produire ou pas.



Concepts des diagrammes de séquence

Fragments d'interactions combinés (alt)

- ▶ Alternative (ou **alt**) : opérateur conditionnel
- ▶ Peut posséder **plusieurs opérandes**, chacune détient une condition de garde.
- ▶ Absence de condition de garde : condition vraie.
- ▶ La condition (garde) indique le nombre de répétitions (min et max) ou une condition booléenne à respecter.
- ▶ Condition **else** : vraie si aucune autre condition n'est vraie.
- ▶ "**Loop**" équivalent d'une boucle for et décrit des interactions qui s'exécutent en boucle.

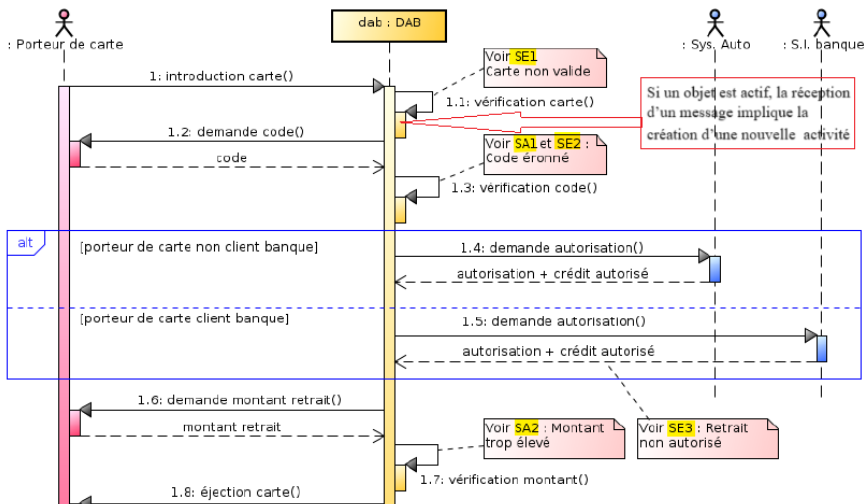


Le diagramme indique que lorsque l'utilisateur se trompe trois fois de code, la carte est gardée et le distributeur se remet en mode d'attente d'une carte.

Concepts des diagrammes de séquence

Exemple

Diagramme de séquence du scénario nominal et du scénario alternatif SA4 (porteur de carte client banque)



Concepts des diagrammes de séquence

Fragments d'interactions combinés (loop)

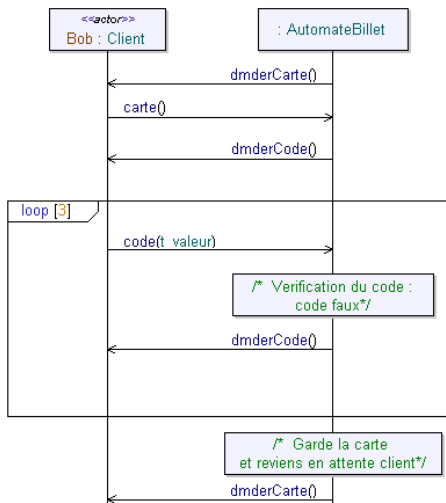
- **loop [min, max, condition]**: boucle un nombre maximum de fois tant que la condition est vraie.

Chaque paramètre (min, max et condition) est optionnel.

Ex.

loop[3] → La séquence s'exécute 3 fois.

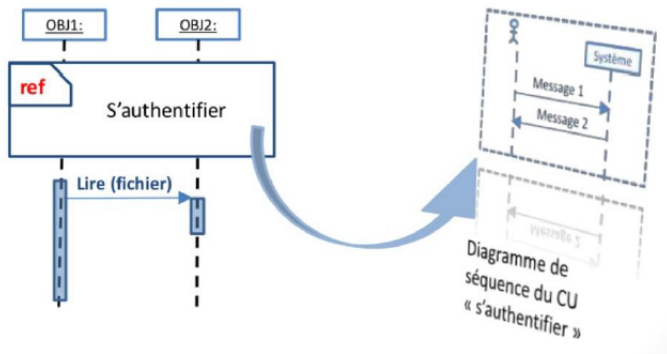
Loop[1, 3, code=faux] → La séquence s'exécute 1 fois puis un maximum de 2 autres fois si code=faux.



Concepts des diagrammes de séquence

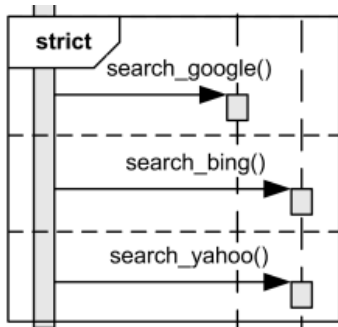
Fragments d'interactions combinés (ref)

- ▶ L'opérateur **ref** : sert comme une référence ou un raccourci vers un autre diagramme de séquence existant
- ▶ Le rôle de l'opérateur **ref** est de réutiliser une partie de diagramme de séquence commune.



Fragments d'interactions combinés (seq)

- ▶ **seq (strict)** : Il existe au moins deux fragments d'opérande.
- ▶ Les messages doivent se produire dans l'ordre des fragments.

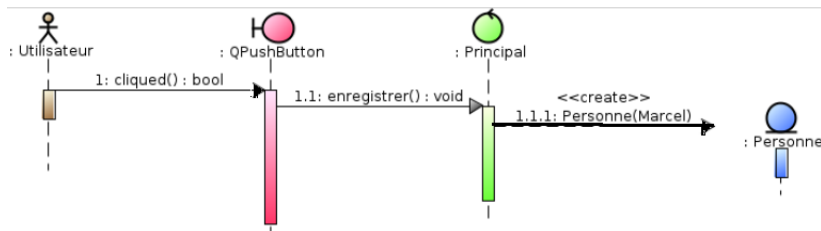
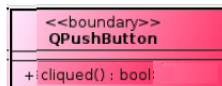


search_google() après search_bing() après search_yahoo()

Stéréotypes de Jacobson

- ▶ A l'intérieur d'un système, il existe très souvent des classes qui possèdent un rôle bien particulier, c'est le cas notamment :
 - pour les classes qui représentent des **composants de l'IHM** ;
 - pour les **classes de contrôle** ;
 - pour les classes qui implémentent la persistance des attributs (associées à une **base de données**).
- ▶ Trois stéréotypes sont distingués :
 - **"boundary"** : classes qui servent à modéliser les interactions entre le système et ses acteurs ;
 - **"control"** : classes utilisées pour représenter la coordination, l'enchaînement et le contrôle d'autres objets ;
 - **"entity"** : classes qui servent à modéliser des informations durables et souvent persistantes.

Stéréotypes de Jacobson



- ▶ Les diagrammes de CU expriment les besoins des utilisateurs d'une manière abstraite.
- ▶ Les diagrammes de séquence :
 - explicitent la dimension temporelle et permettent d'exprimer les contraintes temporelles dans les envois de messages.
 - sont un bon moyen pour représenter les interactions, organisées dans le temps, entre les objets ;
 - s'approchent davantage de l'implémentation avec l'utilisation des boucles, des structures de contrôles, etc. ;
 - permettent de détecter les erreurs qu'on ne voit d'habitude qu'au niveau de l'implémentation : interblocage d'appels;
 - conviennent bien aux applications critiques et temps réel.
- ▶ Comment faire pour représenter une partie du diagramme de séquence ou pour ajouter un bloc conditionnel ?
 - Utiliser les fragments combinés.

Système d'organisation d'une conférence internationale

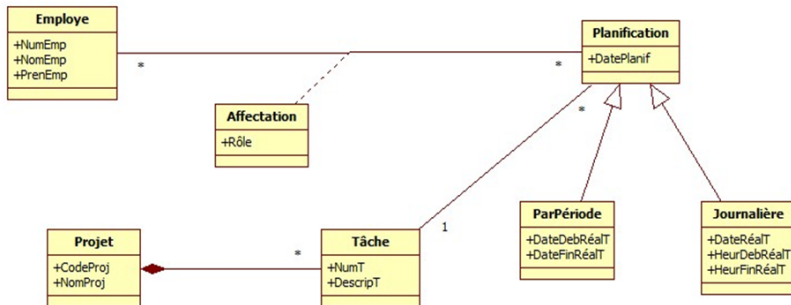
Considérons les règles de gestion suivantes d'un système d'organisation d'une conférence internationale.

- ▶ Le comité d'organisation adresse un appel à communication aux auteurs intéressés par les thèmes de la conférence.
- ▶ Un auteur soumet un papier décrivant ses travaux dans l'un des thèmes de la conférence.
- ▶ Un comité de programme est responsable de l'affectation des papiers reçus à des examinateurs spécialisés pour évaluation.
- ▶ Chaque examinateur évalue les papiers attribués.

Représenter le diagramme de séquence relatif à ces règles.

Exercice 2

Considérons le diagramme de classes suivant :



Élaborer le diagramme de séquence du scénario d'ajout d'une planification pour la réalisation d'une tâche d'un projet, par le responsable, selon la description suivante :

- ▶ Le système charge et affiche la liste des projets.
- ▶ Le responsable du projet sélectionne un projet.
- ▶ Le système charge et affiche la liste des tâches du projet choisi.
- ▶ Le responsable du projet sélectionne la tâche à planifier.
- ▶ Le responsable du projet choisit le type de la planification (par période ou journalière), saisit les informations relatives selon le type choisi (la date de planification aura par défaut la date système) et valide.
- ▶ Le système enregistre la planification et affiche le message "Ajout effectué avec succès".

Chapitre 6 : Diagrammes de composants et de déploiement