

Bases de la conception

« orientée objet »

DUT Informatique
Semestre 2

Mourad Ouziri
mourad.ouziri@parisdescartes.fr

IUT de Paris Descartes



Conception objet :

« Modélisation des interactions avec le
diagramme de séquence UML »

Objectif : savoir analyser la structure de l'application et identifier les objets participant à l'accomplissement d'une tâche et leurs interactions²

Introduction

- ☞ **Modèle fonctionnel – Diagramme de Cas d’Utilisation UML**
 - QUOI/QUELLES fonctions ?
 - Besoins fonctionnels du point de vue des utilisateurs (acteurs)
- ☞ **Modèle objet – Diagramme de Classes UML**
 - QUOI/QUELLES structure d’objets ?
 - Structure des objets (données) de l’application
- ☞ **Modèle d’interactions – **Diagramme de séquence UML****
 - COMMENT ?
 - Comment les objets collaborent pour réaliser les fonctions du DCU
 - Deux types de diagrammes d’interactions en UML : diagramme de séquence et diagramme de collaboration

Modélisation des interactions

Analyse des fonctions de l'application

☞ Rappel : application à objets = société d'objets en interaction

☞ Conception d'une fonction de l'application

- L'objet « Responsable » de la fonction (expert d'information)

- L'objet « Porte-parole » de la fonction

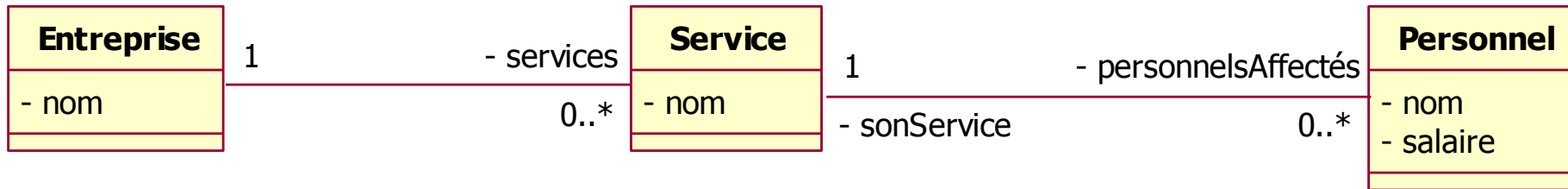
- Les objets « participants » à la séquence d'interactions

- Règle de conception : l'objet « Porte-parole » **doit déléguer le traitement à**

 - l'objet « Responsable » en passant par les objets « participants »

Modélisation des interactions

Analyse des fonctions de l'application

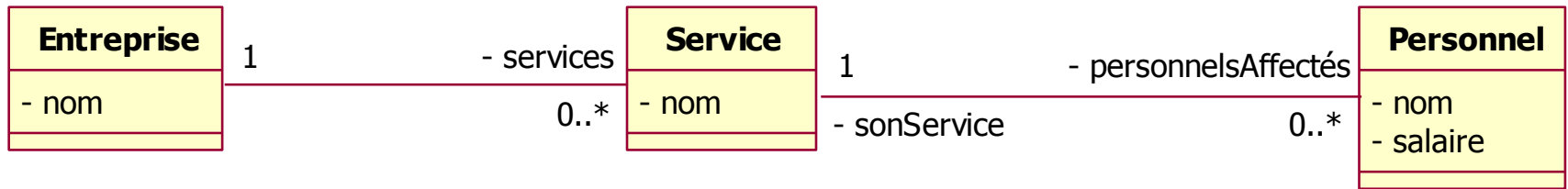


☞ **L'objet « responsable »** de la fonction ou « *Expert d'information* »

- Objet contenant les attributs/données nécessaires à la réalisation de la fonction
- Consulter le salaire d'un personnel : qui détient le salaire du personnel ?
- Embauche d'un personnel : qui permet d'enregistrer l'embauche ?
- Calcul du coût salarial de l'entreprise : qui détient l'ensemble des salaires ?

Modélisation des interactions

Analyse des fonctions de l'application



👉 L'objet « porte-parole » de la fonction

– L'objet détenu (dont on dispose d'une référence) et qui peut atteindre l'objet *Expert d'information*

– Dépend souvent des paramètres d'entrée de la fonction :

void embaucherPersonnel (String nom, double salaire)

*void embaucherPersonnel (String nom, double salaire, **String nomServiceAff**)*

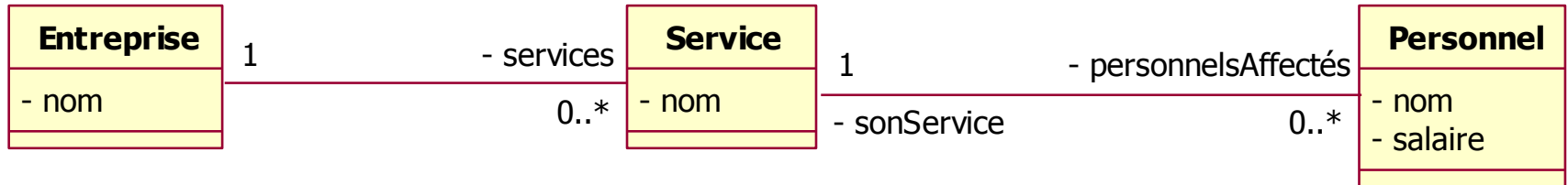
– Cas 1 : service non donné en paramètre => c'est forcément Service

– Cas 2 : nom du service donné en paramètre. Nécessité de trouver l'objet Service

=> Entreprise

Modélisation des interactions

Analyse des fonctions de l'application



☞ **Les objets « participants »** à la séquence d'interactions, deux rôles :

- Relayer la demande du « Porte-parole » au « Responsable » de la fonction
- Réaliser les traitements intermédiaires qui relèvent de leur responsabilité
- Exemple : salaire du personnel Dupond

Responsable : Personnel

Porte-parole : Entreprise

Participants : Service

Diagramme de séquence

Définition et syntaxe

☞ Définition

- Représente les interactions entre objets de manière chronologique
- Montre comment les objets échangent des messages pour réaliser un scénario (ou plusieurs) d'un CU
- Formalise les fiches descriptives de CU

☞ Syntaxe graphique

- En colonnes :
 - Objets (instances de classes) participant au scénario
 - Ligne de vie et périodes d'activité des objets
- En lignes : messages échangés par les objets dans l'ordre chronologique
- Axe du temps : axe vertical allant du haut vers le bas

Diagramme de séquence

Syntaxe graphique

Exemple

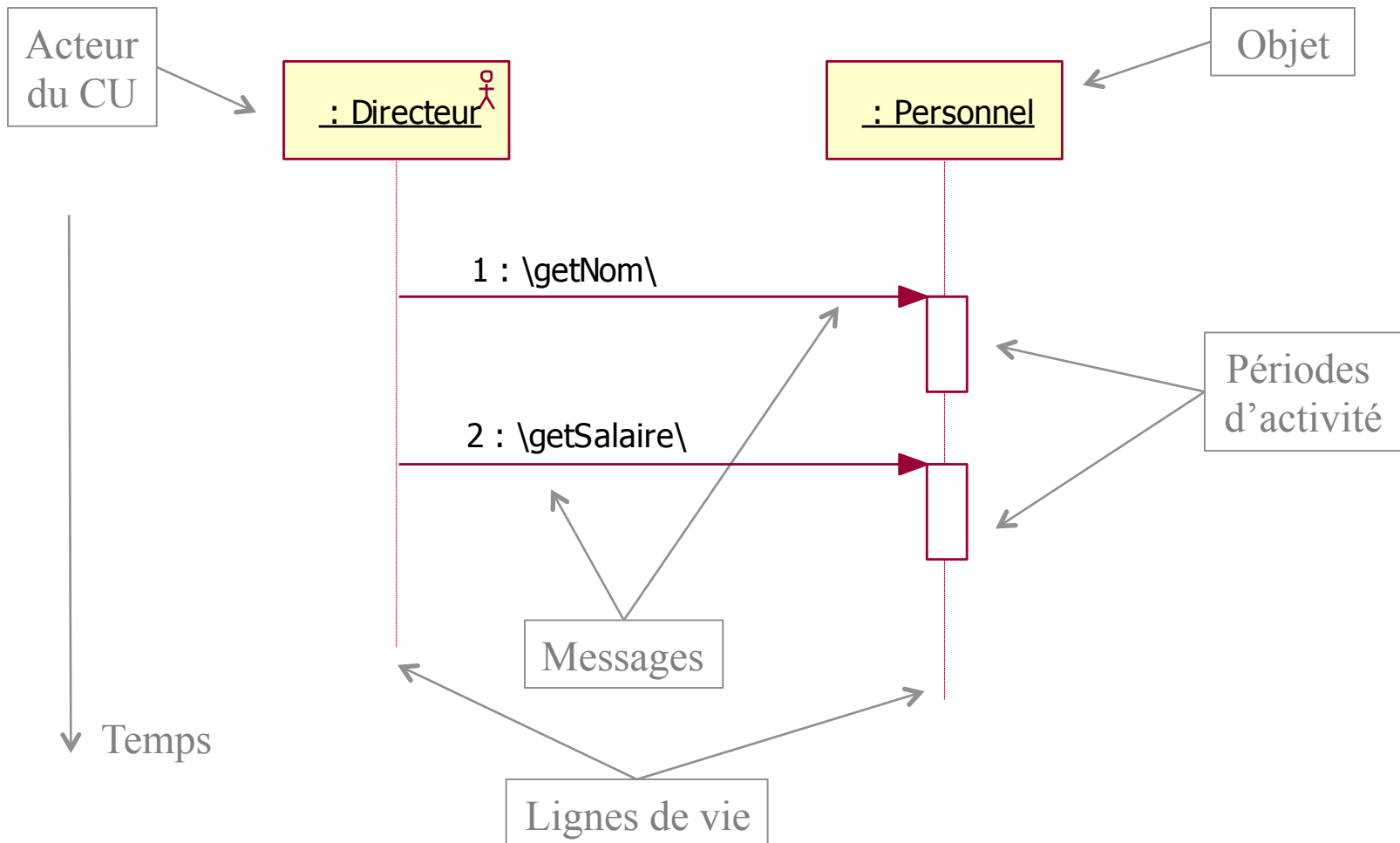


Diagramme de séquence

Eléments de modélisation

☞ Message

- Moyen unique de communication entre objets (encapsulation)
- Objet O1 envoie un message à Objet O2. O2 exécute le message et renvoie, le cas échéant, le résultat à O1
- Concrètement, un message correspond à l'appel d'une méthode d'un objet
- Lorsque le message comporte des informations, il convient de les indiquer en paramètre du message (exemple : *modifierSalaire (Marie, 3000)*)

Diagramme de séquence

Eléments de modélisation

☞ Ligne de vie

- Elle représente l'existence d'un objet pendant l'exécution du scénario.
- Si l'objet existe pendant toute cette période, sa ligne de vie couvre l'ensemble de la hauteur du diagramme.
- Si l'objet est créé lors de l'exécution du scénario, sa ligne de vie débute après le message de création.
- Si l'objet est détruit lors de l'exécution du scénario, sa ligne de vie se termine à l'arrivée du message de destruction.

☞ Période d'activité

- Elle montre la période pendant laquelle un objet accomplit une action

Diagramme de séquence

Exemple

👉 Exemple

- Consulter le nom du service d'affectation d'un personnel

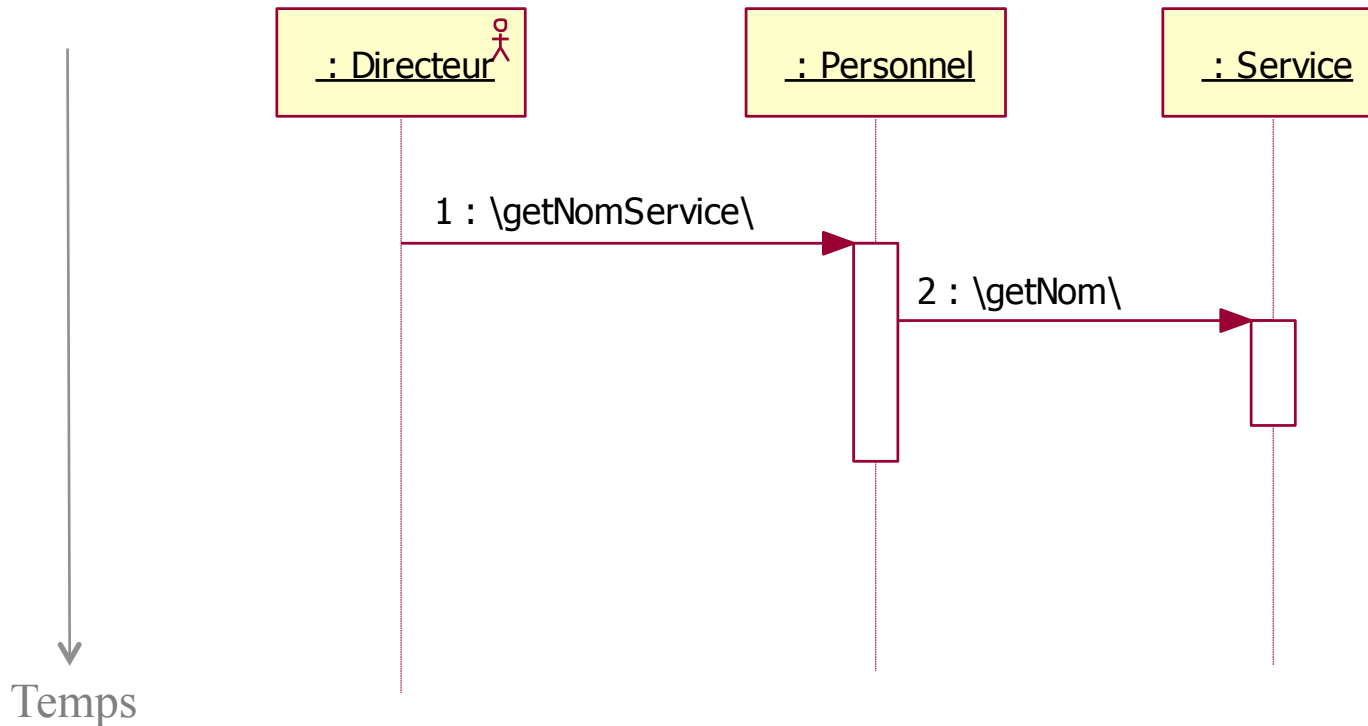
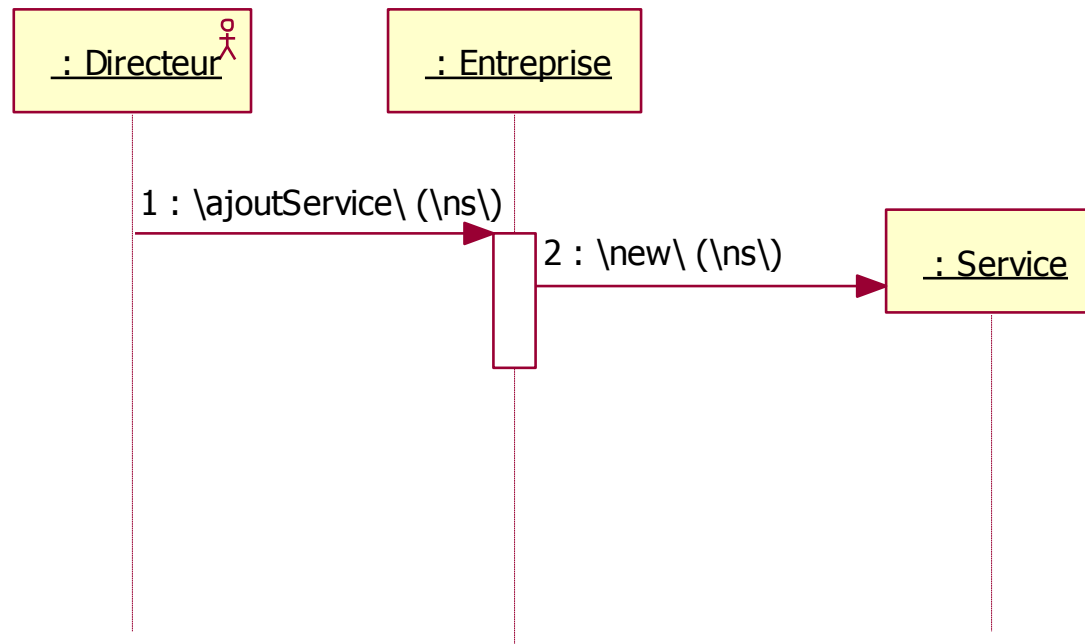


Diagramme de séquence

Eléments de modélisation

☞ Création d'objets

- Ajout d'un nouveau service



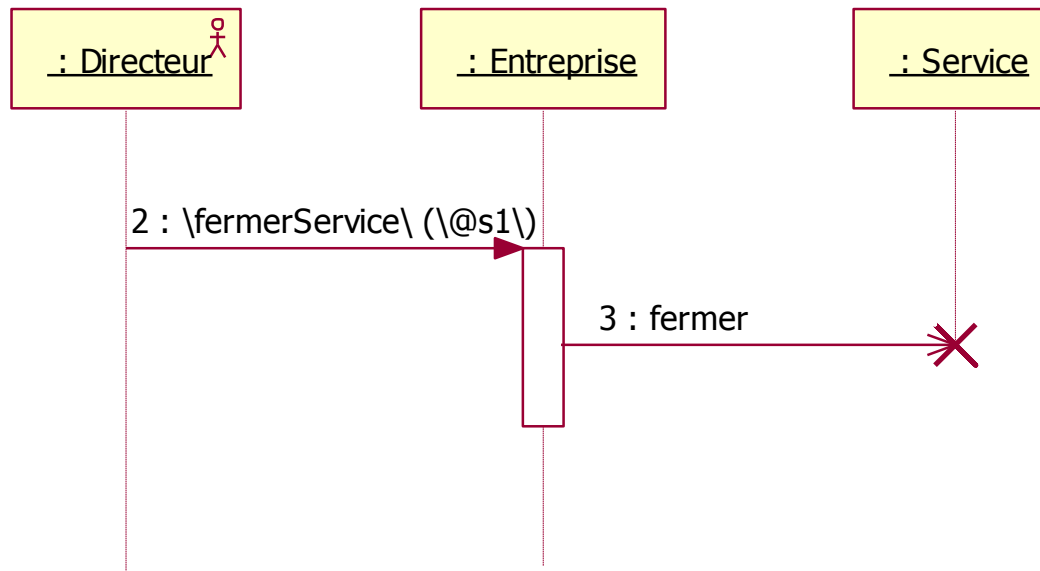
- Exercice : faire l'embauche d'un nouveau personnel à un service

Diagramme de séquence

Eléments de modélisation

☞ Destruction d'objets

- Fermeture d'un service et sa suppression de l'application



- Exercice : faire le départ d'un personnel et sa suppression de l'application

Diagramme de séquence

Conditions et gardes

☞ Messages conditionnés

- Augmentation de salaires : 10% si inférieur à 2000 euro et 5% sinon

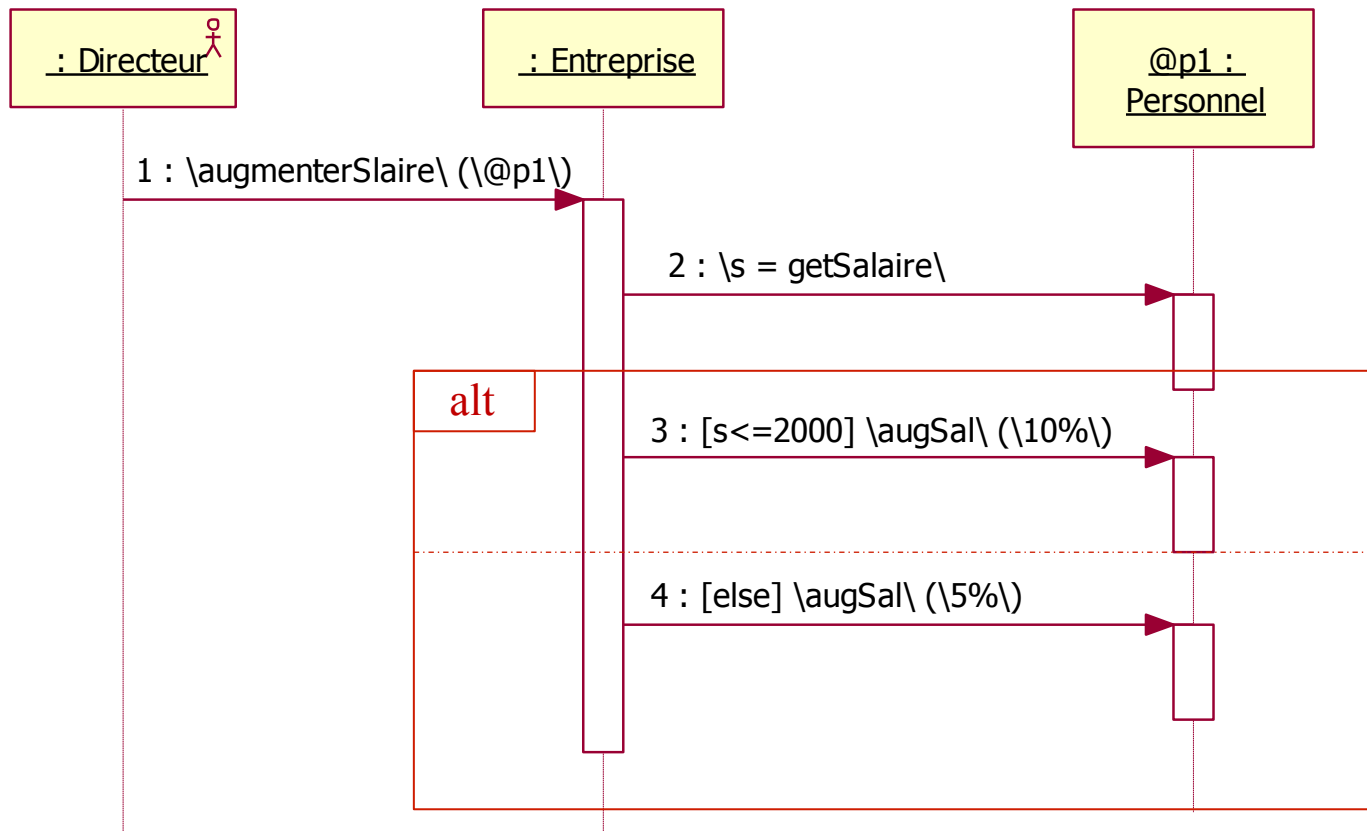


Diagramme de séquence

Itérations

☞ Messages itérés sur plusieurs objets

- Calculer le coût salarial du service *@s1* (somme des salaires de ses personnels)

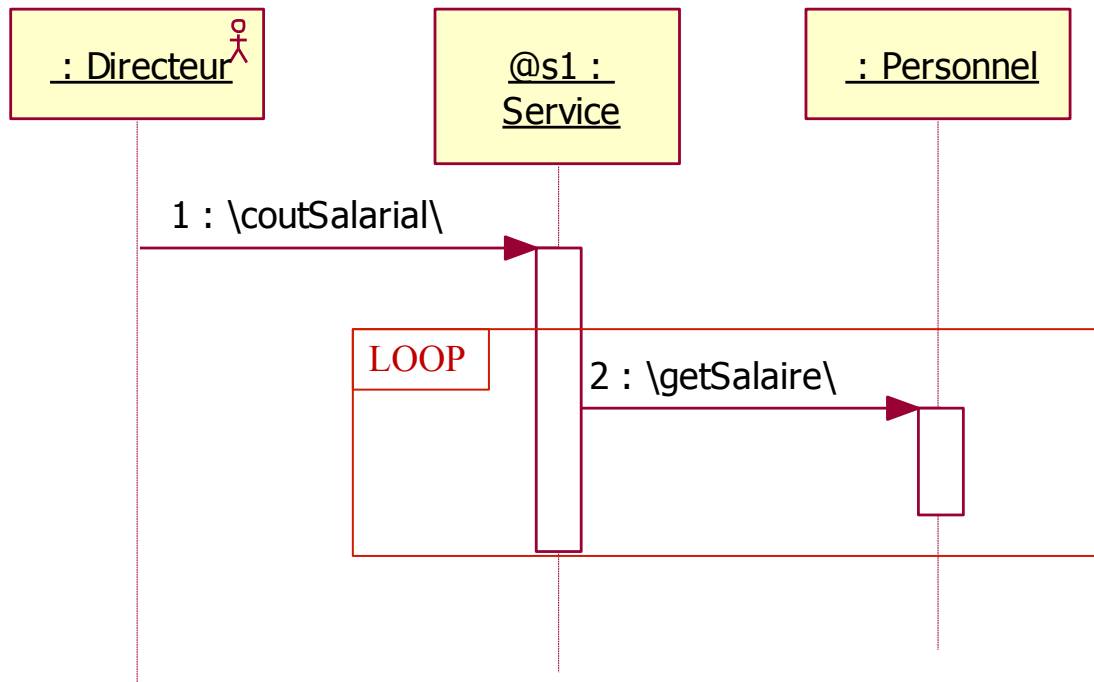


Diagramme de séquence

Point d'entrée d'une application objet

☞ Point d'entrée de l'application objet

- Objet unique (Singleton) représentant la totalité de l'application
- Il permet d'atteindre tous les objets de l'application
- Un diagramme de séquence complet commence souvent par cet objet

☞ Exemple

- Faire le diagramme de séquence permettant de consulter le salaire de « Dupont »
 - Cas 1 : on connaît l'objet *@p2* représentant *Dupont*
 - Cas 2 : on ne connaît que le nom du personnel, l'objet correspondant doit d'abord être trouvé en passant l'objet unique (point d'entrée) de l'application

Principe de délégation

☞ Flot de contrôle centralisé

- Un traitement est réalisé par un seul objet (une de ses méthodes)
- Non recommandée : maintenance difficile

☞ Délégation : flot de contrôle décentralisé

- Un traitement est réalisé par plusieurs objets différents
- Délégation du traitement effectif à l'objet « Responsable »
- Avantage : faible dépendance, bonne maintenabilité

Diagramme de séquence

Flots de contrôle

☞ Flot de contrôle centralisé

- Consulter le coût salarial de l'entreprise

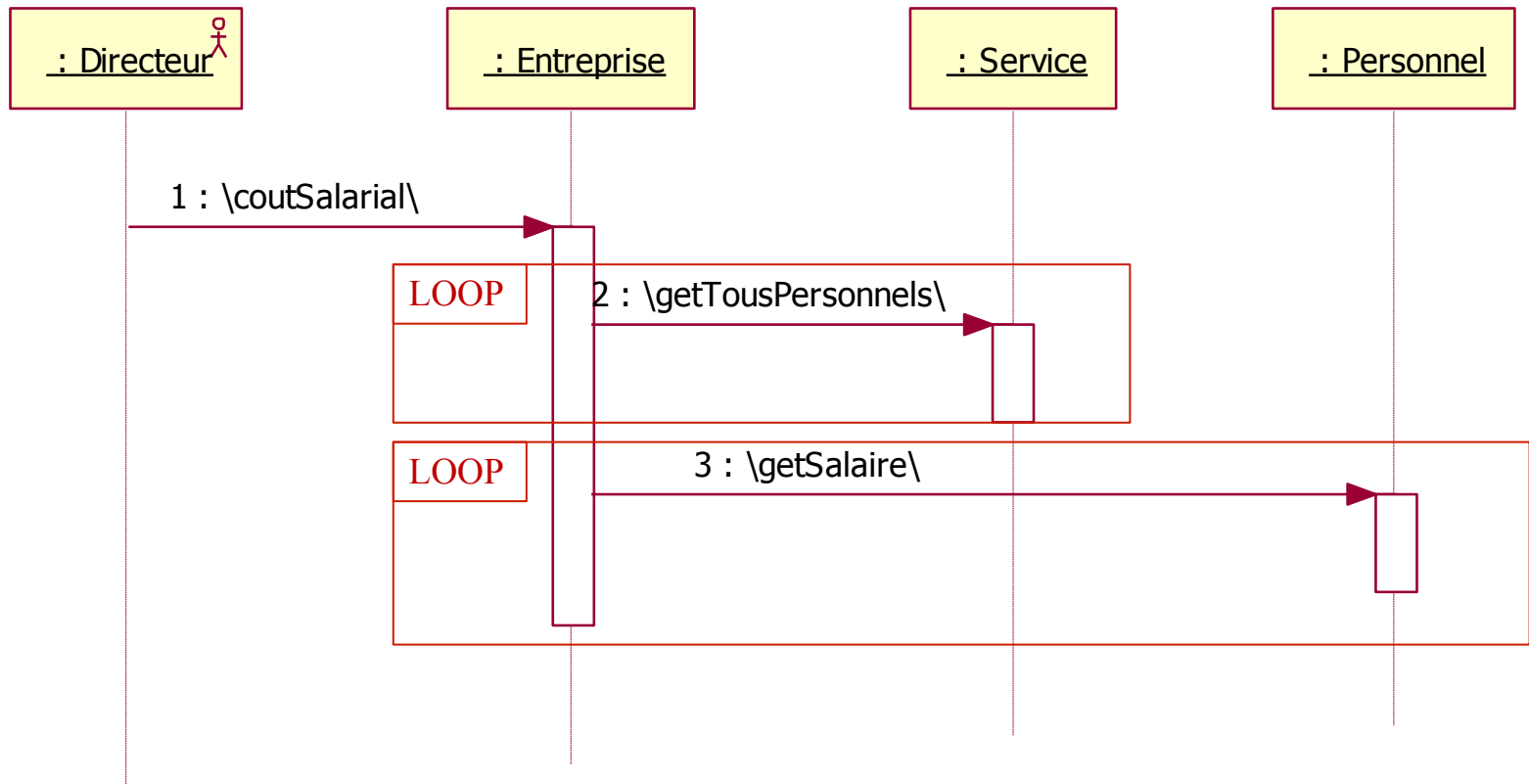


Diagramme de séquence

Flots de contrôle

☞ Flot de contrôle décentralisé : délégation

– L'entreprise Délègue le calcul à ses services

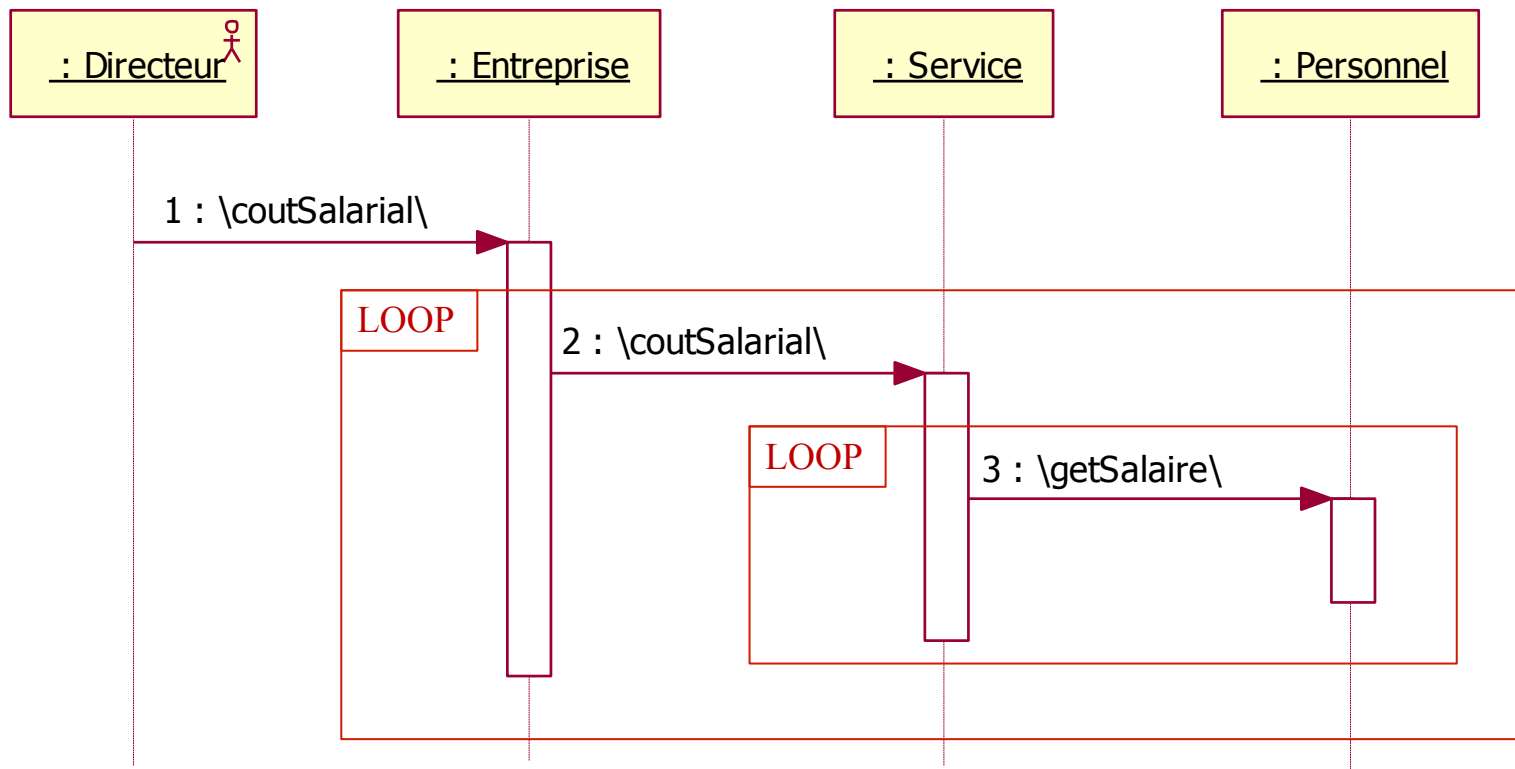


Diagramme de séquence

Types de message

☞ Synchrones

- Message bloquant son émetteur jusqu'à la fin de son exécution

message synchrone() →

☞ Asynchrone

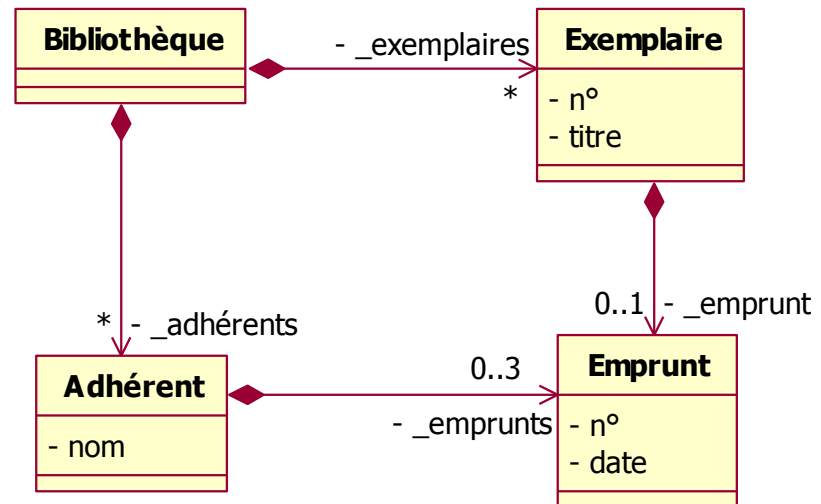
- L'émetteur poursuit son exécution après l'envoi du message
- Nécessité de message de retour si résultat requis

message asynchrone() ↘

Diagramme de séquence

Exemple de synthèse

Exemple : gestion simplifiée d'une bibliothèque



- Inscrire un nouvel adhérent (nom de l'adhérent)
- Vérifier la disponibilité d'un exemplaire (n° de l'exemplaire)
- Enregistrer un emprunt (nom de l'adhérent, n° de l'exemplaire)
- Enregistrer le retour d'un emprunt (n° de l'exemplaire)

Diagramme de séquence

Conclusion

☞ Quelques recommandations :

☞ Vérifier la cohérence des DS par rapport au diagramme de classes

☞ Concevoir par délégation de traitement (ou en flot de contrôle décentralisé)

☞ Utiliser la classe « Singleton » représentant l'objet global (et unique) de l'application (point d'entrée de l'application objet)

☞ Débuter les scénarios à partir de cet objet « Singleton »