



Conception d'Applications Hétérogènes Distribuées

Master 2 Traitement de l'Information

Lionel Médini

Septembre-novembre 2015

Plan du cours

- Outils de programmation avancés
 - Retour sur les patrons de conception
 - Inversion de contrôle (conteneurs d'objets)
 - Contexte et annuaires (communication dans un conteneur)
 - Réserve d'instances (gestion du cycle de vie)
- Systèmes d'information distribués
 - Appel de méthodes distantes (CORBA, RMI)
- Frameworks Java
 - Objets transactionnels distribués (EJB)
 - Composants déployables "à chaud" (OSGi)
- Introduction à l'urbanisation des SI

Rappel : le cycle de vie des objets

Plan

- > Rappels : cycle de vie des objets
- Object Pool
- Conclusion sur les patterns

- Création
 - Appel du constructeur (instanciation)
 - Chargement du bytecode à partir du fichier .class
 - Habituellement depuis le système de fichiers, dans le classpath
 - Allocation de la mémoire
 - Propriétés d'instance / statiques
 - Positionnement d'un marqueur de référence sur l'objet
 - Appel de la méthode `clone()`
 - Duplication de la zone mémoire liée à l'objet

Rappel : le cycle de vie des objets

Plan

- Rappels : cycle de vie des objets
- Object Pool
- Conclusion sur les patterns

- Utilisation
 - Modèle grossier
 - Tout appel de méthodes de l'objet
 - Modèle plus détaillé : différenciation
 - des méthodes métier
 - des traitements “utilitaires” (gestion du cycle de vie)
 - Injection de dépendances, chargement de données...
 - ➔ Standardisation et automatisation de l'appel de ces méthodes

Rappel : le cycle de vie des objets

Plan

- Rappels : cycle de vie des objets
- Object Pool
- Conclusion sur les patterns

- Destruction
 - Si le marqueur de références sur l'objet est à 0
 - Noter l'objet comme inutile : méthode `dispose()`
 - Passer le garbage collecting

Le pattern Object Pool

Plan

Rappels : cycle de vie des objets

> Object Pool

Conclusion sur les patterns

- Contexte
 - Plusieurs objets doivent utiliser des instances d'une même classe
 - Ces instances doivent être paramétrées pour être utilisées
 - Par exemple, liées à un support de persistance
- Problèmes
 - Diminuer le coût de création et de destruction d'instances
 - Obtenir des références sur les instances recherchées
 - Maîtriser le nombre d'instances en fonctionnement

Le pattern Object Pool

Plan

Rappels : cycle de vie des objets

➤ Object Pool

Conclusion sur les patterns

- Solution
 - Création d'une classe Manager
 - Singleton
 - Gère les références sur les instances pour les autres objets de l'application
 - Mettre les instances dans une réserve ("pool")
 - Évite la création/destruction
 - Permet de limiter le nombre d'instances en fonctionnement
 - Gérer le cycle de vie des instances à l'entrée et la sortie du pool
 - Chargement / sauvegarde des données

Le pattern Object Pool

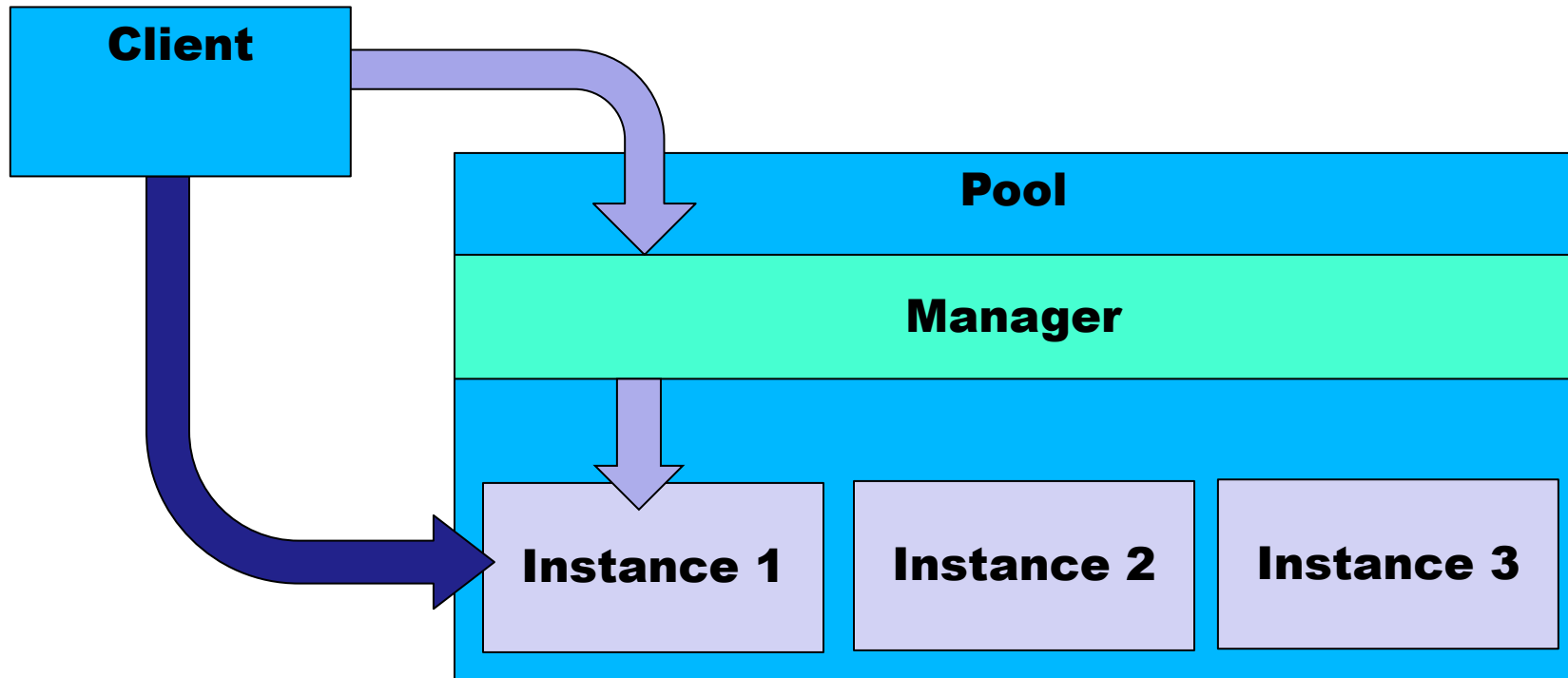
Plan

Rappels : cycle de vie des objets

➤ Object Pool

Conclusion sur les patterns

- Principe



Le pattern Object Pool

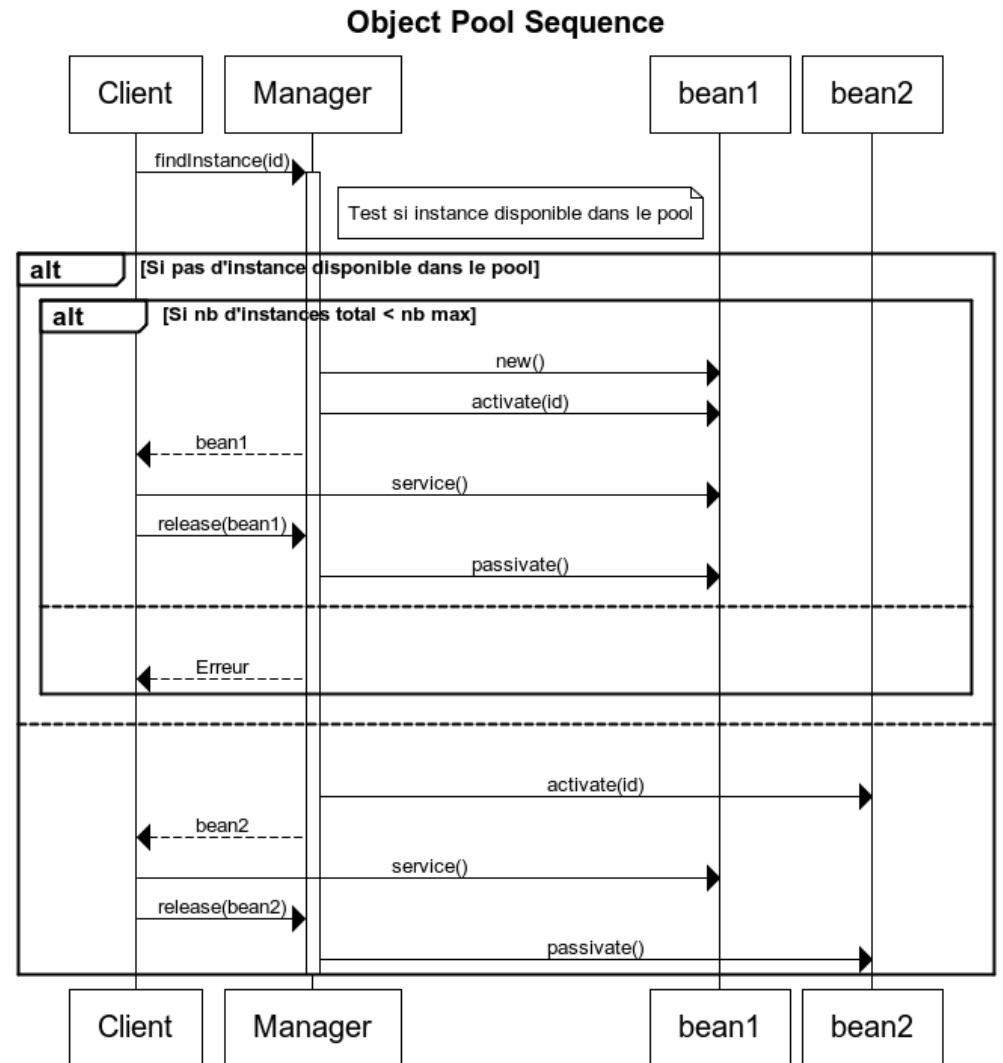
Plan

Rappels : cycle de vie des objets

➤ Object Pool

Conclusion sur les patterns

- Initialisation et service



Le pattern Object Pool

Plan

Rappels : cycle de vie des objets

➤ Object Pool

Conclusion sur les patterns

- Avantages
 - Facilite l'accès aux instances
 - Améliore les performances
 - Permet le contrôle du nombre d'instances par paramétrage
- Inconvénients
 - Peut être lourd à mettre en place pour un gain incertain
 - Dans les langages “modernes”, l'instanciation et le garbage collecting sont optimisés
 - Peut induire des cycles de vie complexes
 - Exemple : diagramme d'états d'un EJB entité

Conclusion

- Autres patterns à utiliser
 - MVC
 - La plupart des frameworks sont conçus pour le MVC
 - Design by contract
 - N'utilisez pas une classe d'implémentation si l'interface suffit
 - DAO
 - Traitements d'accès aux données regroupés dans des objets spécialisés
 - Observateur
 - Notification des changements d'états d'un objet à des observateurs (programmation événementielle)
 - Façade, singleton...

Conclusion

- N'hésitez pas à faire appel à
 - Différents paradigmes de conception
 - Objet : bonnes pratiques
 - Aspects : séparation des préoccupations
 - Événementiel : dynamicité
 - Distribué...
 - Des outils facilitant la programmation
 - Frameworks
 - Bibliothèques de composants
 - Outils de déploiement...

Références

Plan

Rappels : cycle de vie des objets

➤ Object Pool

Conclusion sur les patterns

- Object pool

http://en.wikipedia.org/wiki/Object_pool_pattern

<http://www.oodesign.com/object-pool-pattern.html>

<http://matthieu-brucher.developpez.com/tutoriels/conception/object-Pool/>