



Inversion of Web Service Invocation using Publish/Subscribe Push-Based Architecture

Projet développement avancé

Valentin Brémond
Gaylord Cherencey
Florian Massacret

Sommaire

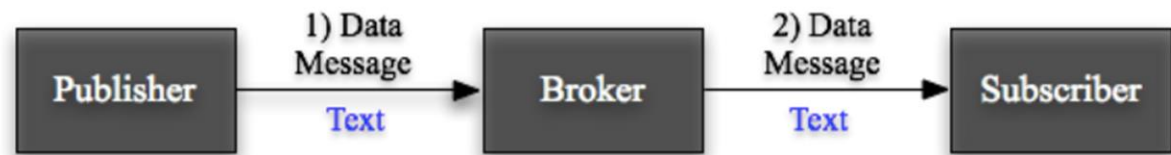
- Introduction
- Rappels
- Projet
 - Technologies utilisées
 - Architecture de l'application
- Démonstration
- Conclusion
- Références

Introduction

Développement d'un logiciel Publisher / Subscriber de type « push » en Java

Rappels

- Publisher / Subscriber de type « push »



- Avantages du « push » par rapport au « pull »
 - Engorgement
 - Nombre de requêtes
 - Temps réel

Technologies utilisées

Patron de conception Observateur

- Permet la réduction des dépendances
- Permet de mettre à jour un composant en temps réel (équivalent du « push » au niveau objet)
- Utilisation pour la gestion de l'interface graphique

Technologies utilisées

JAX-WS

- JAX-WS est l'acronyme de Java API for XML Web Services
- Développement Java de Web Services étendus en utilisant WSDL et SOAP
- Génération d'un document WSDL à partir d'une classe Java

Technologies utilisées

AspectJ

- Outil de Java permettant la programmation par aspects
- Utilisé pour logger des informations
 - Permet de capturer des événements sur les méthodes appelées

Architecture de l'application

Broker

- Passerelle entre le fournisseur et le client
- Stockage des différents types du fournisseur (cuisine, cinéma, ...)
- Stockage de la liste des différents clients
- Abonnement / désabonnement des clients
- Transfert des messages aux clients en fonction de leurs abonnements

Architecture de l'application

Fournisseur

- Se connecte à un broker
- Génère régulièrement des messages de types aléatoires

Architecture de l'application

Client

- Se connecte à un broker
- Récupère la liste des types disponibles
- S'abonne aux types souhaités
- Interface graphique

Architecture de l'application

Interfaces

► Interface Broker

- `boolean ajouterTypeInformation (String type);`
- `boolean envoyerInformation (Information info);`
- `String[] recupererTypesInformations ();`
- `boolean sAbonner (String[] listeTypesInformations);`
- `boolean seDesabonner ();`

► Interface Client

- `boolean envoyerInformation (Information info);`

Démonstration

Conclusion

- Performances
 - Temps réel
 - Division du nombre de requêtes par 2
- Réponse au problème de fiabilité
 - Problème déporté du fournisseur au broker
 - Plus facile de dédoubler un broker qu'un fournisseur

Références

- Inversion of Web Service Invocation using Publish/Subscribe Push-Based Architecture
Thanisa Numnonda
- Inversion of Web Service Invocation using Publish/Subscribe Push-Based Architecture
Rattakorn Punsuph, Thanisa Numnonda

Merci pour votre attention

Questions ?