IAWS - Web Services approche "contract first"

Franck Silvestre <franck.silvestre@irit.fr>

Table of Contents

01 - Pourquoi l'approche "Contract First" ?

Contract Last vs Contract first

- Contract Last

  Le code source est écrit en premier, le contrat au format WSDL est généré de manière automatique

- Contract First

  Le contrat au format WSDL est écrit en premier, le contrat est implémenté dans le langage cible après

Commentaires

Object/XML Impedance Mismatch (IM)

L'expression "... Impedance Mismatch" est souvent utilisée dans le contexte des problématiques de correspondance entre modèle objet et modèle relationnel. L'expression utilisée dans ce dernier cas est la suivante "Object/relational Impedance Mismatch". Cette expression synthétise les causes des difficultés rencontrées quand deux modèles doivent cohabiter, communiquer alors qu'ils reposent sur des fondements très différents.

Commentaires

IM 1 - XSD extensions

```
<simpleType name="CodeAeroport">
```

```
        <restriction base="string">
            <pattern value="[A-Z][A-Z][A-Z]"/>
        </restriction>
    </simpleType>
```

La notion de restriction n'est pas supportée dans les langages objets courant.

Commentaires

En Java la meilleure représentation possible (sans code ajouté manuellement) de ce type est le type String.

IM 2 - Types non "portables"

- Certains types existent dans un langage mais pas dans un autre

  La traduction automatique d'un langage à un autre via le XML généré peut conduire à des sémantiques très différentes.

- Le TreeMap est un type supporté en Java mais non supporté nativement dans d'autres langages comme C#

  La structure de donnée utilisée par C# pour récupérer les données a des fortes chances d'être une HashTable avec des propriétés différentes.

Commentaires

Différence entre HashTable et TreeMap : l'odre des clés n'est pas garanti dans un HashTable,...

IM 3 - Gestion des graphes cycliques

```
public class Mission {
    private String code;
    private List<Acteur> acteurs;

    // getters et setters ...
}

public class Acteur {
    private String nom;
    private Mission mission;

    // getters et setters ...
}
```

- La traduction du graphe cyclique en XML est difficilement automatisable
- Se centrer sur le XML à échanger est souvent la bonne approche

Commentaires

La création d'un fichier XML traduisant le graphe cyclique et conforme au WS-I basic profile [http://www.ws-i.org/Profiles/BasicProfile-1.1.html#SOAP_encodingStyle_Attribute] n'est pas triviale et donc difficile à automatiser...

Autres arguments

- Contract Last
  - Couplage fort WSDL / interface du langage source
  - changer de "stack" peut mener à un changement de WSDL généré
  - contrôle et maitrise des flux échangés difficiles

  Contract first
  - Approche centrée sur contrôle et la maitrise des flux échangés
  - Réutilisation possible des types XSD entre différents services

Commentaires

Le couplage fort entre le langage source et le WSDL entraîne souvent un défaut de stabilité du Web Service.

Le défaut de contrôle peut rendre difficile notamment la maitrise des performances.

La réutilisation possible des XSD s'inscrit dans une démarche large de conception des flux XML échangés pour les besoins du SI étendu. Placer les échanges normaliser sur un niveau d'exigence en terme de conception aussi haut que l'exigence de conception des autres composantes du SI.

02 - Mise en oeuvre avec Spring WS

Exercice

Le service de scolarité dispose d'une application Java permettant de récupérer les relevés de notes de tous les étudiants pour une année scolaire donnée, un niveau donné et un semestre donné. Vous êtes missionné(e) pour mettre en place un Web Service exposant le service de récupération de relevé de notes. L'application Java est Open Source disponible sur Github sous le projet ReleveNotes [https://github.com/FranckSilvestre/ReleveNotes-Spring-WS/tree/master/ReleveNotes-Spring-WS/ReleveNotes].

Objectif : mettre en place le Web Service en utilisant l'approche Contract First en s'appuyant sur le framework Spring WS.

- Proposez un format XML adapté pour le format de la requête et de la réponse du Web Services. Commencez à proposer des exemples de fichier XML puis définissez un schéma XSD définissant la structure des messages impliqués dans votre Web Service.
- En vous appuyant sur la documentation de Spring WS, récupérez et complétez le projet ReleveNotes-WS [https://github.com/FranckSilvestre/ReleveNotes-Spring-WS/tree/master/ReleveNotes-Spring-WS/ReleveNotes-WS] pour qu'il fonctionne. La génération dynamique du WSDL est l'approche retenue pour ce projet.

Commentaires