Université de Nice Sophia-Antipolis

RAPPORT DE PROJET TER

Système de diffusion d'informations en milieu scolaire

ENCADRANTS

Mireille Blay-Fornarino,Sébastien Mosser

REKIMA AHMED 2008-2009

REMERCIEMENTS

Je tiens à remercier les encadrants Mme Blay-Fornarino Mireille et Mr Mosser Sébastien pour l'aide et les conseils qu'ils m'ont apportés tout au long du TER.

Table des matières

1. Introduction:

Sujet du Ter:

Un grand écran a été placé à l'entrée de l'institut pour enfants handicapés sensoriels, Clément Ader. Une architecture à base des services Web a été définie pour permettre la diffusion d'informations sur cet écran:

-Annonces des pauses sonores et visuelles.
-Absence de professeurs.
-Journal télévisé en langue des signes.

Suite à différents problèmes techniques, le système doit être redéfini.
Un système similaire mais avec d'autres types de services (emplois du temps, horaires de bus, ..) a été mis en place à l'Ecole Polytechnique Universitaire Sophia Antipolis.
Le but du TER est d'adapter ce travail pour l'institut Clément Ader. Ce projet comprend donc à la fois une part développement avec les nouvelles technologies et une prise en compte effective du client. Il s'intègre dans un objectif plus large qui est le projet SEDUITE.

Plateforme:

La plateforme SEDUITE citée plus haut est une plateforme qui va permettre de diffuser toutes sortes d'informations, tout cela au sein d'établissements scolaires. Ces informations peuvent être de natures différentes:

-Localisation d'une salle, bureaux des professeurs…
-Vacances scolaires, jours de grèves…

Ces exemples non exhaustifs peuvent êtres complétés suivant les différents besoins des établissements scolaires.

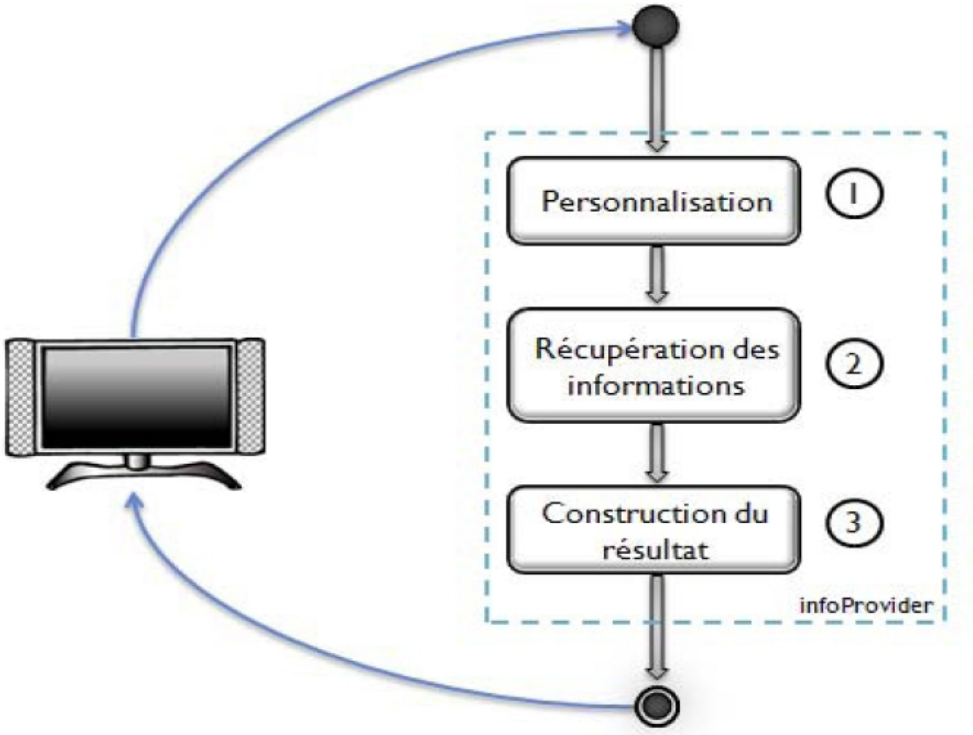
Description:

Ce système utilisé à l'Ecole Polytechnique Universitaire sur Sophia Antipolis diffuse des informations sur plusieurs écrans (donc plusieurs types d'informations destinées à plusieurs utilisateurs différents), tandis que dans l'objectif du TER pour l'Institut Clément Ader un seul écran est concerné(les utilisateurs vont êtres les élèves ainsi que les enseignants).

2.Analyse de l’existant:

A l’EPU de Sophia Antipolis le système a été mis en place par des étudiants en Master2 (voir rapport technique en annexe), dont l’objectif était de réaliser une implémentation concrète du projet de recherche SEDUITE et ainsi mettre en place une architecture orienté service qui permettrait de récupérer des informations provenant de sources différentes aussi bien interne (emploi du temps, absences de professeurs,…) qu’externe (site Web pour les horaires des bus, météo,…).

Orchestration InfoProvider:



(Figure provenant du rapport technique, figure2 page 9)

La personnalisation permet d’avoir plusieurs clients différents (plusieurs écrans…), de leur attribuer un profil et ainsi savoir à quels services il est abonné et ainsi lui renvoyer uniquement ces informations.

La récupération des informations comme son nom l’indique va récupérer des informations de natures et de sources différentes.

La construction du résultat consiste à rassembler les informations récupérées et les transmettre au client.

A partir de là le projet du TER consistait à définir de nouveaux services adaptés à l’Institut Clément Ader en se basant sur le travail déjà effectué pour l’EPU de Sophia Antipolis.

3.Qu'est-ce qu'un Service Web ?

Les services Web (ou en anglais Web services) représentent un mécanisme de communication entre applications distantes à travers le réseau internet indépendant de tout langage de programmation et de toute plate-forme d'exécution :

Pour cela ils utilisent le protocole HTTP comme moyen de transport. Ce qui veut dire que les communications s'effectuent sur un support universel, maitrisé et généralement non filtré par les pare-feux.

Ils emploient une syntaxe basée sur la notation XML pour décrire les appels de fonctions distantes et les données échangées.

Ils organisent les mécanismes d'appel et de réponse.

Grâce aux Services Web, les applications peuvent être vues comme un ensemble de services métiers, structurés et correctement décrits, dialoguant selon un standard international plutôt qu'un ensemble d'objets et de méthodes entremêlés.

Ce découpage permet de maintenir avec facilité l'application, ainsi que l'interopérabilité qui permet de modifier facilement un composant (un service) pour le remplacer par un autre, éventuellement développé par un tiers. Qui plus est, les services Web permettent de réduire la complexité d'une application car le développeur peut se focaliser sur un service, indépendamment du reste de l'application.

Couches
Le fonctionnement des services Web repose sur un modèle en couches, dont les trois couches fondamentales sont les suivantes :

Invocation, visant à décrire la structure des messages échangés par les applications.

SOAP (Simple Object Access Protocol), fonctionnant selon le modèle objet.

Quel que soit le standard utilisé, le principe de programmation est le même : l'appel de méthode distante est réalisé grâce à une bibliothèque cliente qui transmet la demande au fournisseur de service en la formatant en XML de manière transparente; au niveau du serveur une bibliothèque serveur décode la requête, le serveur fait ses traitement, puis répond grâce à cette même bibliothèque; la bibliothèque client décode enfin la réponse afin qu'elle puisse être utilisée par l'application client.

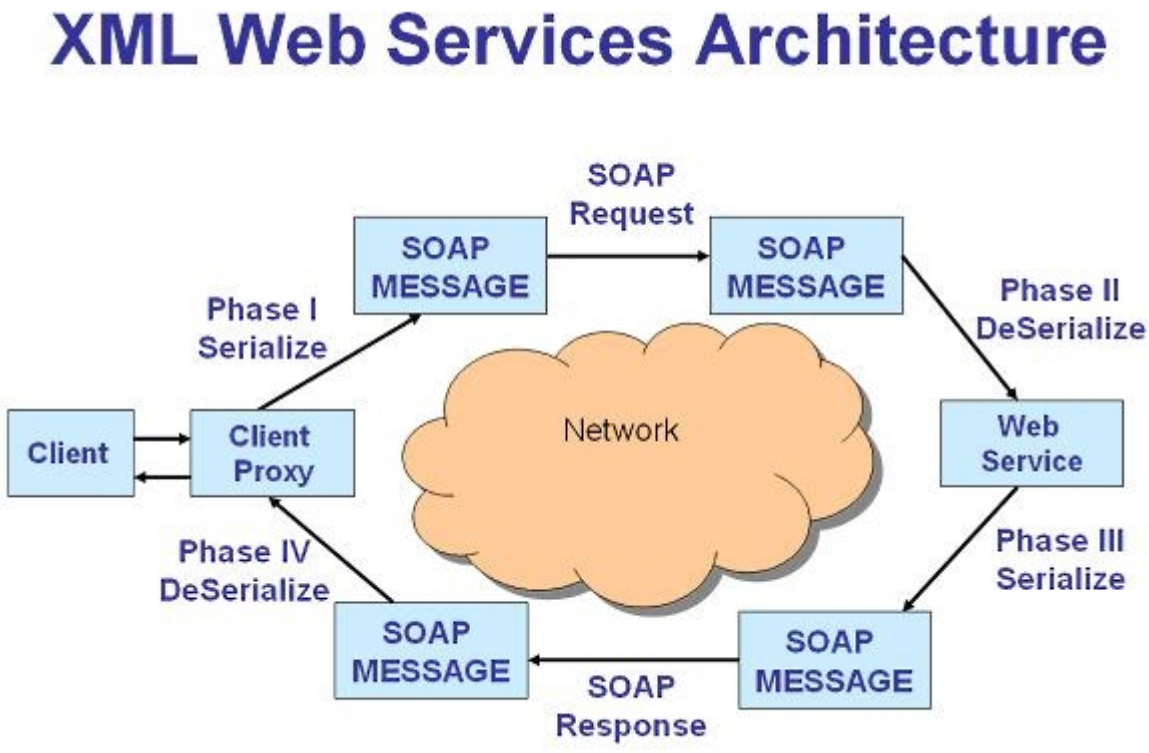
Découverte, pour permettre de rechercher et de localiser un service Web particulier dans un annuaire de services décrivant le nom de la société, l'objectif de chaque service, etc.

Le protocole standard le plus utilisé pour la découverte de services est **UDDI**.

Description, dont l'objectif est la description des interfaces (paramètres des fonctions, types de données) des services Web.

Le protocole standard le plus utilisé pour la description de services est **WSDL**.

Comment fonctionne un Web Service ?



4.Déroulement du Travail

Analyse:

Mon travail pour le TER a été d'analyser un document de travail qui m'a été fourni par les encadrant du TER.Ce document est un Compte rendu d'une réunion qui s'est déroulée à l'Institut Clément Ader (institut pour élèves à déficience visuelle et auditive) ainsi qu'un rapport technique rédigé par des étudiants en Master2(voir annexe), et mon objectif a été d'en déduire l'ensemble de sources d'informations à implanter pour spécialiser le système d'information Séduite en un système adapté à la déficience visuelle. Dans ce document plusieurs problèmes ont été abordés et les points importants que j'en ai extrait sont:

-L'annonces de pauses sonores
-L'affichage d'absence des professeurs
-L'affichage d'images ou de photos (Photos des sorties diffusées aux jeunes...)
-L'affichage d'alarmes
-L'affichage des vacances scolaires

Toutes ces informations seront à destination d'un seul et unique Écran visible par tous.

A partir de ça, j'ai établis une liste de services et d'orchestrations pour qu'ils puissent être déployer dans l'institut et ainsi répondre à ses besoins.

Conception:

Pour chaque service, une Interface (c.-à-d. opération et signature) a été définie, premièrement vous trouverez ci-dessous les interfaces déduites d'après mon analyse du document fournit:

Pour le service Alarmes:

Ce Web service comprend la méthode «public DayAlarm getDayAlarm()» qui en fonction du jour courant renvoie un objet de type «DayAlarm».
Pour connaitre le jour courant il utilise une méthode «getDay()» de la classe DayAlarm.

Une fois le jour récupéré, le service se connecte à une base de données pour récupérer le tuple correspondant dans la table Alarme:

ALARME

ID	DAY	DESC_ALARME
1	Lundi	Pause
2	Mardi	Pause

Par la suite les encadrant ont proposés d'ajouter au tuple un lien vers un fichier sonore, ainsi que le passage d'une date comme paramètre au service.

Pour le service Absences des professeurs:

Ce service comprend la méthode«Absences[] getAbsences(Date date)» qui en fonction de la date passée en paramètre renvoie un tableau d'Absences, une Absence étant une classe qui contient le nom de l'enseignant ainsi que la date de l'absence et la durée .

ABSENCES

ID	NOMP	DATE	DUREE
1	DURAND	12 AVRIL 2009	10
2	PIERRE	15 AVRIL 2009	5

Mon idée était de comparer l'intervalle entre les deux dates, à savoir la date passée en paramètre et la date d'absence du professeur. Mais cela rendais le service plus complexe, donc au final avoir la date de début et de fin revenait à la même chose.

Pour le service Vacances:

Ce service comprend la méthode «Vacances[] getDayVacances(String month)» qui prend un mois en paramètre et renvois les Vacances prévues dans ce mois.

Vacances étant une classe contenant une date, une description correspondant à ces vacances ainsi que la durée pour cette période de vacances.

VACANCES

ID	DATE	DESC	DUREE
1	12/12/2009	NOEL	15
2	13/04/2010	PAQUES	5

Pour le service Menu:

Ce service comprend la méthode «Menu getMenu(Date date)» qui prend en paramètre une date de et renvoie le menu correspondant. Un menu étant une classe qui comprend une date,ainsi qu'une liste d'entrées de plats ainsi que de desserts.

MENU

ID	DATE	ENTREE	PLAT	DESSERT
1	date	fromage	pizza	pommes
2	date	salade	épinards	glace

Pour le service Image:

Ce Web service a déjà été implémenté sur la plateforme SEDUITE

Pour orchestrer tout cela une Orchestration «AdlerOrchestration».

Fonctionnement:

Une fois l'écran démarré il demande des services, l'orchestration se chargera d'invoquer les Web
Services auxquels il est associé en fournissant les paramètres attendus par les interfaces.

Modifications:

Au final il a été décidé (manque de temps puis réduction d'effectif) de construire un Séduite comprenant seulement les services suivants, basé sur mon travail d'analyse des différents documents:

Pour les Services :

- Service gestion absence des professeurs
- Service gestion des images
- Service menu de la cantine

Pour l'orchestration:

- AdlerOrchestration

Les services d'absences et d'images existent déjà dans jSeduite, donc ils vont simplement être réutilisé.
Pour le menu de la cantine étant donné qu'il comprend un ensemble d'entrées, de plats, d'accompagnements et de desserts le modèle de donnée sera de type complexe.

La gestion des menus comprenant deux services on été définis ainsi:

-RestaurantMenu
-RestaurantAdmin

Le service RestaurantMenu propose une opération "getMenu()" qui prend en paramètre une date et renvoi le menu correspondant à la date passée en paramètre. Ce service possède aussi une autre opération "getTodaysMenu()" qui elle ne prend aucun paramètre et renvoie le menu correspondant au jour courant.

Le second service quand à lui implémente un ensemble d'opération permettant:

-L'ajout d’un menu
-La suppression d’un menu
-La modification d’un menu
-La lecture de menu

Pour cela a été ajouté un service d’authentification pour permettre d’utiliser ces services uniquement si on à les permissions.

Pour l’orchestration (non fait), ile but est de connecter les 3 services (images, absences, menu) pour en faire un "provider", comme dans le rapport technique cité plus haut.

Outils de développements:

Pour implémenter tous ces services Web Services l'IDE utilisé a été NetBeans car les outils nécessaires(Glassfish,Bpel...) y sont intégrés,le langage de programmation utilisé est Java. Tout ces choix ont été imposé par les encadrants.

Choix technologiques:

Comme le langage et l'IDE étaient imposés,le seul choix technologique que j'ai eu à faire était de savoir où et comment stocker les éléments d'un menu et comment les récupérer,le choix s'est tout naturellement porté sur une une base de données(ici Oracle car la seule disponible sur les machines de la fac).

Implémentation:

Interfaces des méthodes nécessaires aux services :

--public Boolean isRegistred(String name, String passe)

Cette méthode du service sert à vérifier si l'utilisateur qui s'identifie est bien enregistré dans la base de données comme utilisateur.
Renvoie un booléen suivant que l'utilisateur se trouve ou non dans la base.

--public void ajouterMenu(ArrayList<String> entrees, ArrayList<String> plats, ArrayList<String> accomp, ArrayList<String> desserts, Date date)

Cette méthode permet d'ajouter un menu dans la base de données. Elle prend en paramètre quatre listes correspondantes aux entrées, plats, accompagnements et desserts qui vont servir à constituer le menu.
Le choix de quatre listes comme paramètres au lieu de passer en paramètre un objet de type Menu vient du fait que lors des tests avec un client il y' avait un conflit type entre le type Menu de la méthode du service et celui du Client, problèmes dûs à la sérialization de types complexes.

--public void deleteMenu(Date date)

Cette méthode prend en paramètre une date et supprime dans la base de donnée le menu correspondant

--public Menu getMenu(Date date)

Cette méthode prend en paramètre une date et récupère dans la base de données le menu correspondant. Ici pas de problème de compatibilité de type, on peut récupérer le type Menu venant du service, les accesseurs de la classe Menu sont accessibles même ceux de types complexes hormis pour les modificateurs de types complexes, les accesseurs de types primitifs sont tout à fait accessibles.

--public void updateAddtoMenu(ArrayList<String> entree,ArrayList<String> plat, ArrayList<String> accomp, ArrayList<String> dessert, Date date)

Cette méthode permet de modifier un menu en lui rajoutant un ou plusieurs éléments, une liste peut être vide cela ne pose pas de problème lors d'ajout dans la base de donnée, le cas est traité. Le fait de passer un objet de type Menu revenait aux même problèmes rencontrés plus haut.

--public void updateDeletefromMenu(ArrayList<String> entree, ArrayList<String> plat, ArrayList<String> accomp, ArrayList<String> dessert, Date date)

Cette méthode permet de modifier un menu en lui supprimant un ou plusieurs éléments elle est calquée sur la même méthode se trouvant juste au-dessus.

-public Menu getMenu(Date date) *et* ***public Menu getTodaysMenu()***

Pour ces deux méthodes rien de spéciales la récupération des menus se faisant de manière normale en cherchant dans la base de donnée. Le type de retour étant un menu le client à accès aux accesseurs pour récupérer les différentes listes constituant le Menu.

Toutes ces méthodes ont été implémentés puis testées à partir d'un client(une simple application java et une servlet).

Schéma de la base

Représentation de la base de données avec les différentes tables:

Une table regroupant tous les desserts:

dessert_menu

ID	NAME
1	Mousse au chocolat
2	Petit pot de glace
3	Yaourts
4	Fruits
5	Tarte au Pommes

Une table regroupant tous les plats:

plat_menu

ID	NAME
1	Sauté de dinde forestier
2	Cordon bleu
3	Filet de poisson sauce basilic
4	Raviolis gratinés
5	Blanquette de poisson

Une table regroupant tous les accompagnements:

accomp_menu

ID	NAME
1	Edam
2	Salade verte
3	Frites
4	Riz créole
5	Petit-Suisse

Une table regroupant toutes les entrées:

entree_menu

ID	NAME
1	PIZZA
2	Carottes râpées
3	Concombre vinaigrette
4	Salade de tomates
5	maïs
6	thon

Une table Menu:

date_menu

ID	M_DATE
1	12/05/09
2	13/05/09
3	14/05/09
4	15/05/09

Une fois ces tables en place l'idée était d'avoir une table de relation qui ferait le lien entre la table Menu et les autres tables constituant un menu:

r_menu_dessert

ID	ID_ENTREE	ID_MENU
10	1	2
11	2	2
1	1	1
2	2	1

r_menu_entree

ID	ID_ENTREE	ID_MENU
10	2	2
11	3	2
1	1	1
2	2	1

etc…

Exemple d'utilisation:

A partir d'une date on récupère son ID, on récupère les ID_DESSERT correspondant à l'ID de la date ensuite on récupère les noms dans la table DESSERT correspondant à ces ID.

Après analyse par les encadrants le schéma précédent était un peu compliqué et ont proposés de simplifier le schéma en utilisant seulement deux tables, une table COURSE contenant tous les desserts, les plats, les entrées et les accompagnements chacun ayant un nom unique, et une autre table Menu comme suit:

COURSE

ID	KIND	NAME
1	starter	carottes râpées
2	Starter	tomates
3	Dessert	pommes
4	Side_dish	pâtes

MENU

DATE	COURSE
12/12/2009	1
13/12/2009	5
14/12/2009	8

Interface Php

Choix technologique:

Ici le langage PHP a été imposé,et comme il s'agissait de réaliser une interface accessible depuis le net,l'utilisation du HTML était naturelle.
Pour la mise en page le choix a été fait d'utiliser le CSS car il permet d'alléger le code de la page et d'avoir différent styles en toute simplicité,les feuilles de styles se trouvant dans des fichiers séparés et cela permet aussi de modifier l'apparence graphique du site en toute facilité.

Interface en PHP pour gérer les menus:

Par la suite il a fallu faire une interface en PHP afin de pouvoir gérer tout ce qui concerne la gestion des menus à savoir:

-Ajouter un Menu
-Supprimer un Menu
-Modifier un Menu

Elle permet aussi à l'administrateur d'ajouter à la base de données des entrées,plats,accompagnements et desserts qui ne se trouveraient pas dans la base de données.

Une interface de connexion permet d'identifier l'utilisateur en vérifiant que l'utilisateur est bien enregistré dans la base.

L'interface de modification se divise en trois parties(3 colonnes),la partie gauche concerne tout ce qui est en rapport avec l'ajout dans la base de données,la partie du milieu concerne uniquement la suppression d'un menu(le menu est effacé définitivement de la base) ou une partie d'un menu.
La partie se trouvant à droite permet de visualiser une menu en cliquant sur la date correspondante dans un Calendrier,ou bien en entrant dans le champ qui correspond la date désirée.

L'utilisateur de l'interface ne pourra pas rentrer n'importe quoi comme données,surtout au niveau des dates, il y' a une vérification du bon format ainsi que la vérification que la date entrée n'est pas absurde.
Tout ajout dans la base subit une vérification pour éviter tout problème de redondance.

Les liens du site se trouvent à l'annexe(deux premiers liens).

Difficultés rencontrées:

Le premier problème a été au niveau de la gestion du temps,le travail de familiarisation avec les différents outils de développements ainsi que la technologie utilisée qui devait se faire avant de passer à "temps plein" n'a débuté que lors du passage à "temps plein".Le second problème a été la réduction d'effectif(abandon d'un membre du groupe) .
Sinon au niveau de l'implémentation pour les Web services j'ai rencontré différents problèmes lors de la sérialization d'objets de types complexes passés en paramètres du coté serveur.
Pour la représentation dans la base de données,mon schéma était un peu complexe donc les requêtes étaient SQL étaient elles aussi un peu complexes.

Pour la conception de l'interface en PHP, il n'y a pas eu vraiment de grand difficultés rencontrées mise à par pour la connexion à la base de données Oracle la version de PHP sur les machines de la fac ne reconnaissait pas les fonctions de connexions pour oracle. Comme proposé par les encadrant je me suis rabattu sur un hébergeur gratuit disposant ainsi d'une base de donnée MySql.

5.Conclusion

Objectifs Atteints:

Au final,l'objectif initialement prévu du sujet de TER n'a pas était atteint au complet,j'ai implémenté les Web services demandés,je me suis créer un client pour tester les Web services pour voir s'ils fonctionnaient correctement en testant différents cas possibles,les menus étaient bien ajoutés dans la base de données,la modification d'un menu fonctionnait ainsi que la suppression. Globalement tout était fonctionnel lors du test avec le client.

L'orchestration qui devait se charger d'utiliser ces Web services n'a pas été faite(j'ai commencé une ébauche incomplète).

Pour l'interface en PHP tout est fonctionnel,les manipulations dans la base de données se font sans problèmes. L'interface empêche l'utilisateur d'entrer n'importe quoi,surtout au niveau des dates.

Améliorations:

Il reste à améliorer un l'interface et à gérer les problèmes de rafraichissement lorsqu'on ajoute des éléments à un menu afin que l'utilisateur n'est pas à rafraichir la page lui-même.
Les Web services sont à améliorer et l'orchestration(à peine commencée) est à compléter.

6.Annexes

http://ter.lescigales.org

http://ter.lescigales.org/formulaire.php

http://www.jseduite.org/doku/

http://www.i3s.unice.fr/%7Emh/RR/2009/RR-09.01-S.MOSSER.pdf

http://fr.wikipedia.org/wiki/Service_web

http://www.commentcamarche.net/contents/web-services/web-service.php3

http://www.commentcamarche.net/contents/web-services/soa-architecture-orientee-services.php3

http://www.netbeans.org/kb/60/websvc/jax-ws.html

http://www.netbeans.org/kb/60/soa/index.html

http://www.phpfrance.com

http://www.siteduzero.com/tutoriel-3-14668-un-site-dynamique-avec-php.html