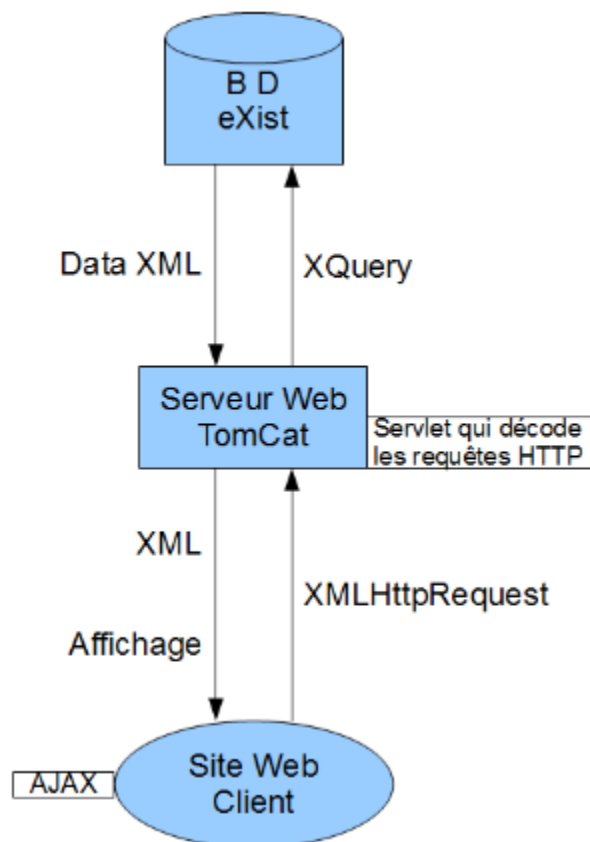


Projet XML

MyRawWeb

G. Cabaret F.Michaudon A.Kutner R.Fritz C.Fezai E.Rol

Architecture Web 3 tiers



Partie Client

API utilisées: maps + visualization + geocode + DOM

Google Maps

Google Maps est un service gratuit de carte géographique et de plan en ligne. Il met à notre disposition une API assez complète permettant d'afficher une carte puis d'y mettre des marqueurs pour géolocaliser des lieux, l'affichage des informations se faisant sous forme de bulles.

1: Création de la carte.

Plusieurs méthodes pour afficher la carte.

- En javascript
- En php
- En flash

Nous avons décidé d'utiliser Google maps en javascript :

Il faut lui ajouter un élément HTML qui sera utilisé comme conteneur pour la carte, par exemple une balise <div>, avec un identifiant explicite.

Pour pouvoir utiliser les services de Google Maps, on doit posséder une Clef d'API, qui permet à Google Maps de nous identifier. (possible sans clé mais qu'en local)

Une fois que l'on a obtenu une clef d'API, on doit inclure à notre page HTML le fichier JavaScript permettant d'interagir avec les services Ajax de Google.

Ensuite il suffit juste d'initialiser la map grâce à l'API, puis de la charger au démarrage de la page (load).

2: Ajout des marqueurs.

Il est très simple de créer un marqueur à partir de coordonnées (longitude et latitude). Cependant pour afficher les conférences on ne dispose pas de ces deux données.

Heureusement il existe une API de Geocodage (*Geocoding API*) qui permet de connaître les coordonnées d'une ville. On utilise donc celle-ci pour créer nos marqueurs (ville+pays car il peut y avoir le même nom de ville dans plusieurs pays: exemple Nancy...Fr Us).

On peut être aussi précis que l'on veut (nom d'un pays, le nom d'une ville ou encore une adresse complète).

3: Ajout des informations dans les bulles des marqueurs.

Le principe de base pour l'affichage d'une fenêtre d'informations est le suivant :

- Création d'un marqueur (vue précédemment),
- Création d'une fonction ouvrant une fenêtre d'informations sur un marqueur,

- le marqueur est créé avec un listener qui permet de pouvoir lancer des actions

JavaScript

Le client crée une requête (a)synchrone (Objet XMLHttpRequest: lit des données ou fichiers sur le serveur) qui sera envoyée au serveur. En retour il récupère des données au format XML (dans notre cas) qui pourront être parcouru à l'aide des méthodes de DOM (Document Object Model).

Pour googleMap: structure des données dans la base a permis de simplifier la structure du document XML

```
<markers>    --> noeud racine
    <marker TitreConference="AFIA Atelier Représentation" Annee="2008" Pays=" France"
        Ville="Montpellier"/>
    <marker TitreConference="CEMRACS Modelling"      --> noeuds contenant des
attributs
        Annee="2008" Pays=" France" Ville="Marseille"/>
```

...

Pour les statistiques: on envoie une requête afin de récupérer toutes les conférences d'un pays et leurs années

structure xml:

```
<ConferencePays total="594">    --> noeud racine
    <confYear>    --> child node
        <TitreConference>ACM IEEE International Conference</TitreConference>
        <year>    --> child node of <confYear>
            <Annee>
                2007
            </Annee>
            <Annee>    --> element node
                2008    --> text node
            </Annee>
            <Annee/>    --> element node
        </year>
    </confYear>
```

...

on parcourt ce document afin de stocker le nombre de conférences effectuées chaque année;
-> accès aux noeuds éléments <Annee> (certains sont vides il faut alors vérifier qu'ils ont un noeud texte)

->un tableau intermédiaire stock le nombre d'occurrences des années

[2006, 2007, 2008, 2009,2010] -> [0,1,1,0,0]

ces données permettent alors de créer un graphique de type camembert (Visualization API de Google) afin de voir l'évolution du nombre de conférences chaque année dans un pays.

Choix de conception:

- L'utilisation de la googleMap est limitée aux pays (on ne peut afficher les conférences que d'un pays et non sur toute la carte compte tenu de la taille des données stockées dans la base de données)
- L'utilisateur peut effectuer plusieurs recherches tout en gardant les marqueurs déjà affichés (leur accès dans la liste est cependant supprimé pour cause de mémoire) ou effacer la carte.
- Un des problèmes rencontrés est lorsque plusieurs conférences ont eu lieux dans la même ville, sur la map les marqueurs seront alors superposés et un seul sera accessible au click de la souris; un choix de conception a été alors de stocker les marqueurs dans une liste (situé a gauche de la map) afin de pouvoir accéder aux infos-bulles de tous les marqueurs (y compris les non visibles). Cependant Pb lors du stockage: si trop de données dépassement de la capacité...

Partie Serveur

Le serveur tomcat fait une requête en Xquery à la base de données et retourne le résultat sous forme de flux XML.

Requete Xquery

Afin de pouvoir récupérer les données de la base de données, nous avons utiliser le logiciel eXist.

Les différentes requêtes que nous avons réalisées nous permettent d'interroger la base de données.

Pour pouvoir extraire des informations et reconstruire de nouveaux documents ou fragments XML nous utilisons un langage de requête XQuery.

la syntaxe utilisée est la syntaxe "naturelle " dites FIWORE (for , let , where , order , by et return) qui joue un rôle similaire à celui du langage SQL.

Type d'utilisation de XQuery:

Pour extraire des informations du documents ou collections XML en fonction de différents paramètres nous utilisons des requêtes paramétrées. Avec l'obtention des paramètres de l' URL.

Les Requêtes:

1- google.xqy

La requête interroge la base de données XML Raweb en fonction du nombre de paramètres de l' URL pour obtenir certaines informations des Conférences , le titre, l'année, le pays et la ville.

Les paramètres pour utiliser l' URL sont pays et year.

on peut utiliser ces deux paramètres ensemble ou l'un des deux séparément.

s'il y a comme paramètres year=2006&pays=fr on aura comme résultats toutes les Conférences de 2006 en France.

Si on utilise seulement year=2009 on aura comme résultats toutes les Conférences de 2009 du monde entier.

ou seulement pays = us on aura comme résultats toutes les Conférences des USA sur

différentes années .

2-ProjectByYear.xqy

Retourne un document Xml de tous les projets d'une année donnée en paramètres URL.

3-ProjectByNameThem.xqy

Retourne une liste des projets qui travaillent sur ce thème sous format d'un document Xml.

4-ATitleOfTheme.xqy

Retourne la liste des titres de tous les Thèmes dans la base de données Raweb

5-stat.xqy

Retourne la liste des titres de conférences avec des années différentes dans la base de données Raweb pour faire les statistiques sur le nombre de conférences par rapport aux années dans chaque pays.

6-ConferenceByProject.xqy

Retourne toutes les conférences d'un projet.

Java EE servlet

Nous utilisons une servlet (classe Java qui permet de créer dynamiquement des données au sein d'un serveur HTTP) pour faire la communication entre le client et la base de données. Nous utilisons par défaut la méthode POST mais il est possible d'utiliser la méthode GET pour avoir une réponse (cela nous a permis de tester le résultat de la servlet sans la partie cliente de l'application).

La servlet construit la requête et l'envoie à la base de données à travers le protocole HTTP. Dans un deuxième temps elle récupère le résultat et l'envoie sur la sortie standard, cette sortie sera lue par une requête faite à l'aide de l'objet XMLHttpRequest d'ajax.

Java EE nous permet aussi d'utiliser des Java Server Page (jsp) qui nous aident à dynamiser nos pages html.