Projet Master IC - SIG3T : médiatisation “artisanale” de données géographiques

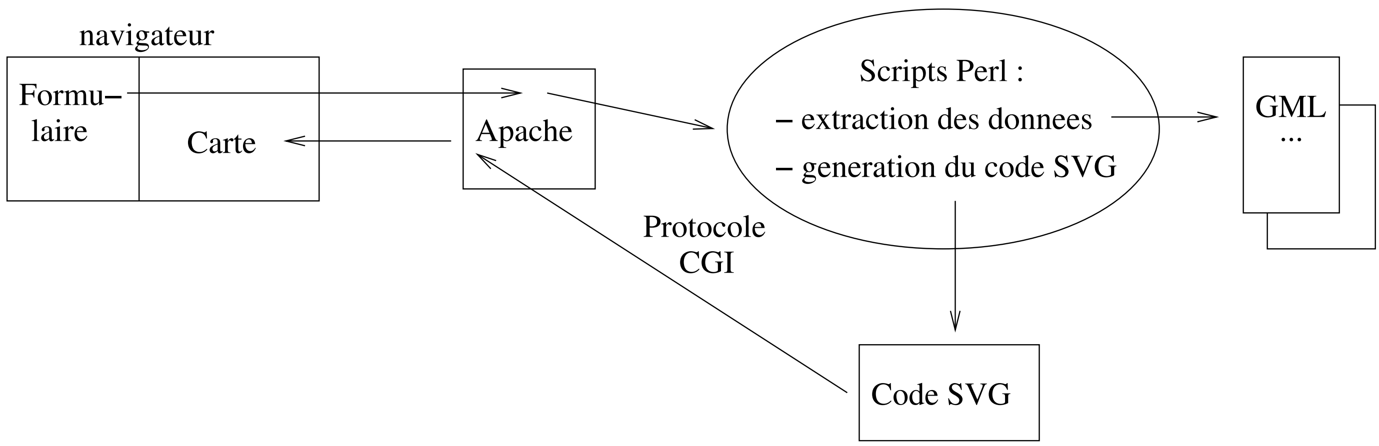
Pierre Pompidor

Architecture du projet

Le projet sera constitué de trois parties :

- **le “tier” données** : des données struturées sous forme d'un ou plusieurs fichiers textes, et définissant plusieurs couches d'informations géographiques superposables ;
- **le “tier” application** : un ou plusieurs script Perl paramétrés par l'utilisateur et
  - permettant d'extraire les données correspondant aux couches spécifiées par l'utilisateur ;
  - de générer un rendu graphique par la création de code **SVG (Scalable Vector Graphics)**.
- **le “tier” présentation** : un ou plusieurs script **HTML** permettant de créer le formulaire permettant d'invoquer le/les script(s) Perl. (Nous pouvons considérer que la génération du code SVG fait également partie de ce tier).

Architecture du projet SIG "artisanal"



Du côté du navigateur (client-side) :

Comment diviser le navigateur en deux parties (en deux frames)

grâce aux balises HTML suivantes placées dans un premier fichier HTML d'accueil :

- <**frameset**> : pour permettre un découpage horizontal ou vertical en sous-fenêtres ;
- <**frame**> : pour indiquer quelle sera la ressource (page HTML, code SVG...) à charger dans la sous-fenêtre.

Exemple :

```
<frameset cols="30%,*" border="2">
     <frame src="formulaire.html" name="gauche" />
     <frame src="carte.svg" name="droite" />
</frameset>
```

Comment émettre des paramètres à partir d'un formulaire :

grâce :

- à l'attribut **action** de la balise <**form**> qui permet de spécifier le script à appeler ;
- à l'attribut **name** de la balise <**input**> qui permet de nommer chaque paramètre ;
- et à la balise <**input type="submit"**> qui permet de créer un bouton de soumission.

Exemple :

```
<form action="SIG.pl">
      <input type="checkbox" name="delim">   Délimitations </input> <br/>
      <input type="checkbox" name="voies_f"> Voies ferrées </input> <br/>
      <input type="checkbox" name="routes">  Routes        </input> <br/>
      <br/>
      <input type="submit" value="Générer la carte" />
</form>
```

Du côté du serveur (server-side) :

Comment récupérer les paramètres dans un script Perl :

en allant les chercher dans la chaîne de caractères `$ENV{'QUERY_STRING'}` qui contient tous les paramètres séparés par des **&** : **nom_p1=valeur_p1&nom_p1=valeur_p1&...**.

Exemple de transformation de la chaîne de caractères en un tableau associatif **%p** plus facile à manipuler ;-) :

```
$parametres = $ENV{'QUERY_STRING'};
@parametres = split /&/, $parametres;
foreach $param (@parametres) {
    if ($param =~ /(.*)=(.*)/) {
        $p{$1} = $2;
    }
}
```

Premiers pas en SVG (Scalable Vector Graphics)

Exemples d'un petit dessin en SVG

Dessin de deux yeux dans une tête

```
<?xml version="1.0" ?>
<svg>

<circle cx="70" cy="95" r="50" style="stroke:black;fill=none" />
<circle cx="55" cy="80" r="5" style="stroke:black;fill=black" />
<circle cx="85" cy="80" r="5" style="stroke:black;fill=black" />

</svg>
```

Et si on rajoute des moustaches ...

```
<?xml version="1.0" ?>
<svg>

<circle cx="70" cy="95" r="50" style="stroke:black;fill=none" />
<circle cx="55" cy="80" r="5" style="stroke:black;fill=black" />
<circle cx="85" cy="80" r="5" style="stroke:black;fill=black" />

<g id="moustaches">
   <line x1="75" y1="95" x2="135" y2="85" style="stroke:black;" />
   <line x1="75" y1="95" x2="135" y2="105" style="stroke:black;" />
</g>

<use xlink:href="#moustaches" transform="scale(-1 1)
                              translate(-140 0)" />

</svg>
```

Et avec un sourire en forme de salière, on obtient un chat

```
<?xml version="1.0" ?>
<svg>

<circle cx="70" cy="95" r="50" style="stroke:black;fill=none"/>
<circle cx="55" cy="80" r="5" style="stroke:black;fill=black"/>
<circle cx="85" cy="80" r="5" style="stroke:black;fill=black"/>

<g id="moustaches">
   <line x1="75" y1="95" x2="135" y2="85" style="stroke:black;"/>
   <line x1="75" y1="95" x2="135" y2="105" style="stroke:black;"/>
</g>

<use xlink:href="#moustaches" transform="scale(-1 1)
                              translate(-140 0)" />

<polyline points="65 65, 75 65, 72 75, 68 75"
      style="stroke:black; fill:yellow" />

</svg>
```