

## Le langage SVG

### Objectif

L'objectif de ce TP est de présenter SVG, un langage de description d'images vectorielles utilisant XML. La génération automatique avec XSLT sera abordée.

Outils utilisés :

- Processeur XSLT : Xalan (<http://xml.apache.org/xalan-j/index.html>)
- Navigateur WEB (mozilla, firebird, IE) + plugin SVG Adobe (<http://www.adobe.com/svg/viewer/install/>)  
Il existe une version beta <http://www.adobe.com/svg/viewer/install/beta.html>
- Plugin de Corel [www.corel.com/svg/](http://www.corel.com/svg/)
- Lecteur de SVG Batik (<http://xml.apache.org/batik/>)
- Editeur de dessins vectoriel
  - sodipodi (<http://www.sodipodi.com/>).
  - inkscape (<http://www.inkscape.org/>)
- Visualiseur : Eyes of GNOME (eog)

### 1 Exemples

Des exemples de documents SVG se trouvent ici : `/usr/local/batik/batik-1.5/samples/`

ou là <http://www.adobe.com/svg/examples.html> (ne fonctionne bien que sous windows avec le plugin adobe). Vous pouvez aussi utiliser les logiciels de dessin inkscape ou sodipodi pour créer un document SVG.

Attention, pour les visualiser vous pouvez **sur la machine sis**, lancer mozilla (utilise le plugin adobe) ou firefox ou executer la commande `batik-squiggle`. Vous pouvez aussi installer sur votre machine le plugin adobe pour mozilla ou bien batik.

Des infos, sont données sur ce site : <http://svg.org/> et là <http://svgfr.org/> (tutoriel animés).

### 2 Constructions de graphiques à partir d'un document XML

Le document `tables.xml` présente des informations concernant des enchères. L'élément `users` contient une liste d'éléments qui décrivent un utilisateur. Chaque utilisateur possède un identifiant unique (`userid`). L'élément `items` contient la liste des objets en vente. Chaque objet possède un identifiant unique (`itemno`). L'élément `bids` contient la liste des offres faites par un utilisateur pour un objet à une certaine date.

1. Ecrire une feuille de style XSLT qui transforme le document XML source en un document SVG. Vous commencerez par présenter uniquement pour l'utilisateur U01 et sous la forme d'un histogramme, pour chaque ITEM (sur lequel il a fait une offre) le montant de cette offre. Vous y ajouterez ensuite le montant de l'offre la plus élevée. (cf. l'exemple `barChart.svg`)
2. Ecrire une autre feuille de style pour créer un graphe qui indique pour chaque objet le montant de toutes les offres qui ont été faites et le nom des utilisateurs correspondants.
3. Rendre ces documents dynamiques. Le survol de chaque barre devra indiquer au moins la date de l'offre et éventuellement la description de l'objet et la meilleure offre sur l'objet.
4. Ajouter ce qui vous semble utile pour permettre la visualisation des données.

### 3 Pour aller plus loin, une application complète

1. Créer un document XML et sa DTD qui représente les notes d'élèves dans cinq matières pour un nombre de contrôles variables (par élève et par matière). Les descriptions des élèves et de leurs notes par matière seront séparées (vous utiliserez des attributs ID et IDREF).
2. Ecrire les expressions XPath qui permettent d'extraire les informations importantes du document XML.
3. Ecrire une feuille de style XSLT qui présente ces données dans une page HTML.
4. Ecrire une feuille de style XSLT qui permette de transformer ce document XML en SVG pour le visualiser : courbes des notes, histogramme par matière, ...
5. Rendre les documents SVG dynamiques et animés, réaction à la souris pour afficher des informations supplémentaires (nom des élèves, ...) ou permettre un lien entre les vues.