

Berthet Laurent

Bugnard Julien

TP de XML

Compte rendu : Représentation des graphes conceptuels

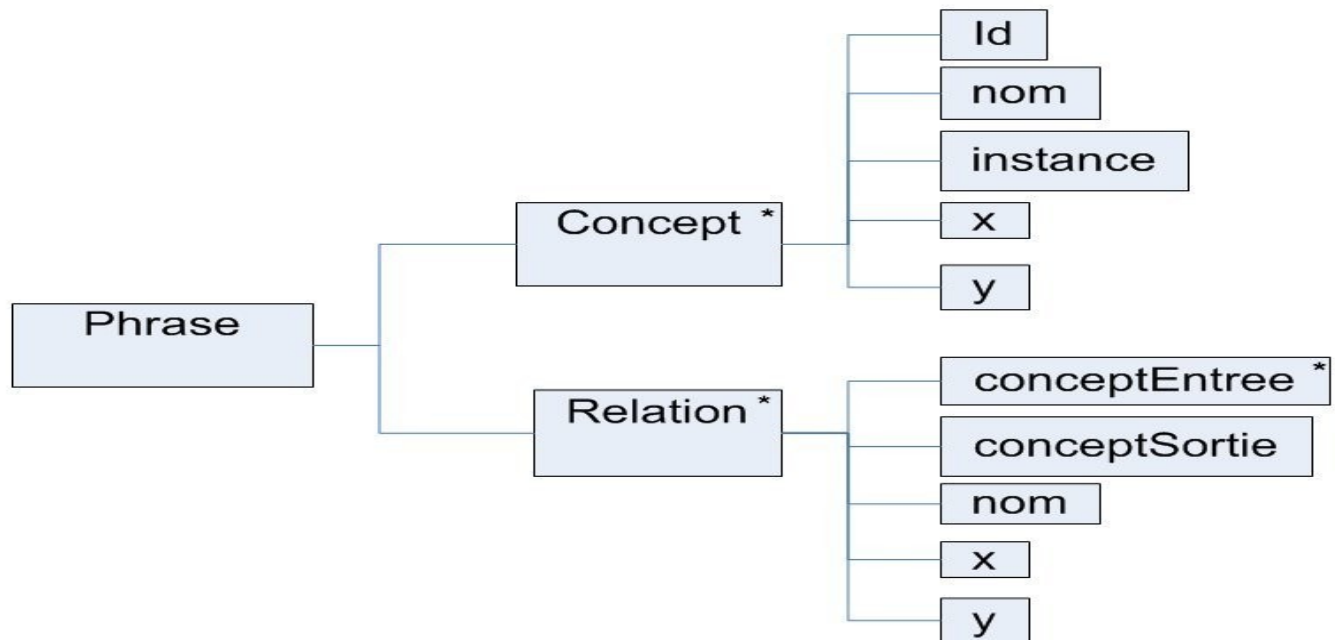
10 octobre 2007

Introduction

Le but de ce TP était de définir une DTD permettant de représenter des graphes conceptuels ainsi que deux codes XSLT permettant la représentation en Calcul des prédicats et d'afficher le graphe sous forme d'image.

Pour réaliser ceci, nous avons fait un schéma XML et une DTD.

Schéma XML



Le symbole " * " signifie que le noeud peut apparaître plusieurs fois dans le fichier XML.

Nous avons décidé de structurer notre fichier XML comme dans le schéma ci-dessous. Notre racine est la phrase et elle contient des éléments qui sont les concepts et les relations entre les concepts.

- Les éléments contenus dans les concepts:
 - id: Cet id permet d'identifier le concept
 - nomConcept : Nous donnons le nom du concept
 - instance : Cet élément n'est défini que si le concept est une instance
 - Afin de pouvoir représenter notre graphe conceptuel correctement sous forme d'image, nous ajoutons deux coordonnées qui permettent de placer le concept à un endroit précis. ces coordonnées se nomment x et y.

- Les éléments contenus dans les relations:
 - idConceptEntree : la relation peut avoir plusieurs concepts en entrée.

- idConceptsortie : la relation doit avoir seulement un concept
- nomRelation : Nous donnons un nom à la relation
- Comme pour les concept nous intégrons des coordonnées x et y afin de pouvoir représenter les relations correctement

Explications

. DTD

La DTD est la traduction du schéma XML :

L'élément Phrase est la racine du document.

Une phrase est composée de plusieurs concept et de relations.

Un concept est composé d'un nom, d'un ID, d'un type et de coordonnées X et Y.

Une relation est composée d'un nom, de 1 ou plusieurs idConceptEntree et d'un idConceptSortie ainsi que de deux coordonnées X et Y.

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT phrase (concept+, relation*)>
<!ELEMENT concept (id, nom, type, x, y)>
<!ELEMENT id (#PCDATA)>
<!ELEMENT nom (#PCDATA)>
<!ELEMENT type (#PCDATA)>
<!ELEMENT relation (idConceptEntree+, idConceptSortie, nom, x, y)>
<!ELEMENT idConceptEntree (#PCDATA)>
<!ELEMENT idConceptSortie (#PCDATA)>
<!ELEMENT nom (#PCDATA)>
<!ELEMENT x (#PCDATA)>
<!ELEMENT y (#PCDATA)>
```

. Création du fichier XSLT pour faire le calcul de prédicat :

Pour faire le calcul de prédicat, il suffit de parcourir tous les concepts, on affiche le nom du concept ainsi que son ID.

On parcourt une seconde fois les concepts pour savoir si le concept est une instance. Si c'est le cas on affiche inst('le nom de l'instance').

Ensuite on parcourt toutes les relations pour afficher le nom de la relation ainsi que les ID des concepts en entrée et en sortie.

. Création du fichier XSLT pour l'affichage du résultat en SVG :

Pour dessiner des objets, nous créons des "xsl:element" avec tous les attributs nécessaires pour afficher notre éléments comme nous le voulons.

Par exemple, pour dessiner un rectangle, nous créons un élément "rect" :

```

<xsl:element name="rect" >
  <xsl:attribute name="transform"> translate(100,100) </xsl:attribute> permet de faire une
translation des coordonnées x,y
  <xsl:attribute name="x"><xsl:value-of select="x/text()"></xsl:value-of></xsl:attribute>
cela nous permet de récupérer les coordonnées stockées dans le fichier XML, les coordonnées sont
l'angle en haut à gauche
  <xsl:attribute name="y"><xsl:value-of select="y/text()"></xsl:value-of></xsl:attribute>
  <xsl:attribute name="width">150 </xsl:attribute> on définit la hauteur et la largeur du
rectangle
  <xsl:attribute name="height">50 </xsl:attribute>
  <xsl:attribute name="style">fill:blue</xsl:attribute> on choisit la couleur
</xsl:element>

```

Tous les éléments ont la même forme que le rectangle. Toutes les explications sont dans le fichier XSLT.

La plus grande difficulté a été d'afficher les flèches car il a fallu parcourir un certain nombre de fois les concepts et les relations.

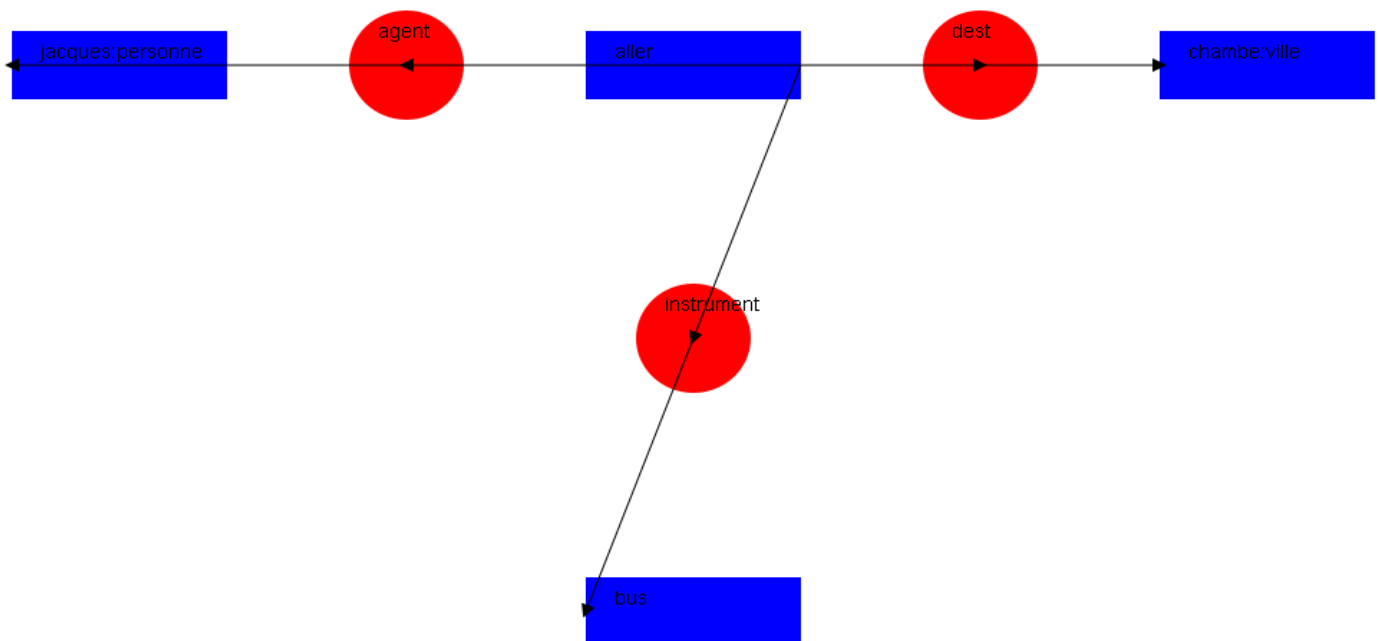
Conclusion

Pour conclure, ce TP nous a permis d'utiliser le format SVG pour afficher le contenu d'un fichier XML grâce à un fichier XSLT de transformation. Nous nous sommes rendus compte de la complexité de XSLT pour réaliser un affichage complexe.

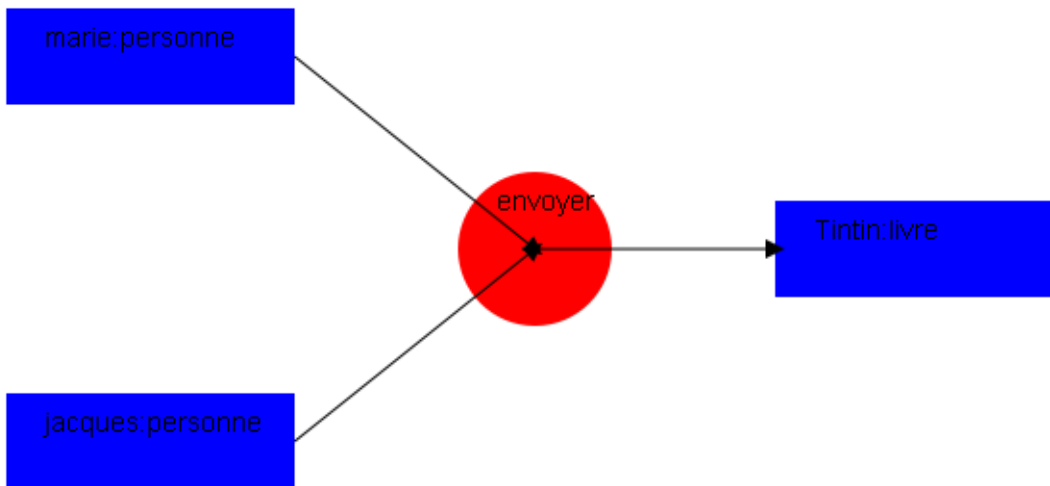
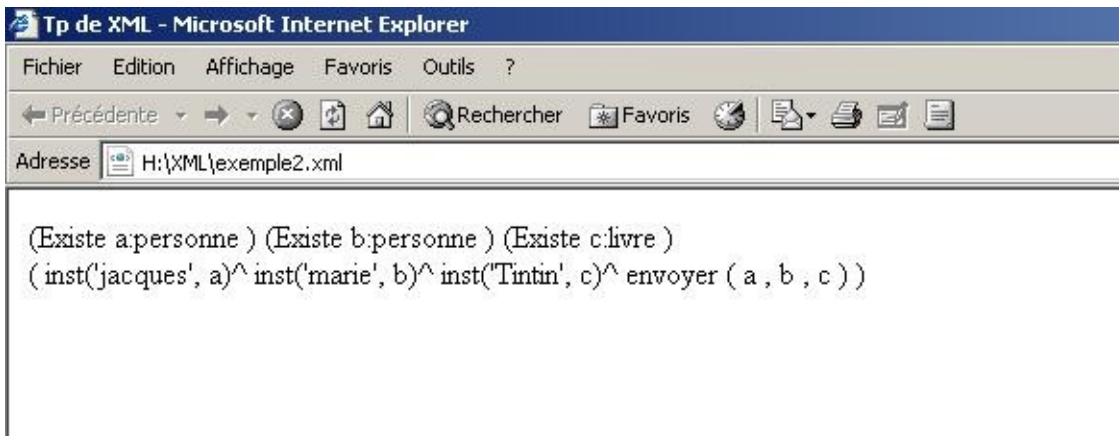
Exemples

Voici les résultats des exemples en XML que nous avons créés. La première image et la phrase sous forme de calcul des prédicats et la seconde est la représentation en graphe conceptuel.

Exemple1:



Exemple 2:



Exemple3:

