I - Introduction

Comme tout le monde le sait, la suite Office 2007 comporte plusieurs applications. Word, Excel, PowerPoint et Access pour ne citer que les plus utilisées.

Depuis la version 2003, la suite Office est à même de pouvoir prendre en charge le format XML qui a tendance à prendre de l'importance, à tel point que Microsoft a décidé d'introduire un nouveau format de fichier : Office Open XML. **Description de ce nouveau format libre et ouvert**

Pour cet article, nous allons utiliser la suite bureautique de Microsoft dans sa dernière version 2007 et un logiciel de traitement XML qui est Altova XMLSpy.

Ces logiciels sont téléchargeables dans leur version démo sur les sites des éditeurs respectifs.

Altova XMLSpy

Version d'essai Microsoft Office 2007

I-A - Le XML Extensible Markup Language

Le XML est le frère du HTML tous deux étant en effet issus du SGML. Là où le HTML a une approche de mise en forme de texte , le XML est lui orienté données. Son système de balise simple et définie par le créateur du fichier vise aussi bien à un parsage aisé par une application qu'à garder un aspect lisible pour l'utilisateur humain.

Le XML est rarement seul, il est souvent accompagné de deux autres types de fichier : le XSD et le XSL(T).

I-A-1 - Le XML

La structure d'un fichier XML est toujours la même, elle commence toujours par :

```
<?xml version="1.0" encoding="UTF-8"?>
```

ou encore

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

qui comporte la version du XML utilisée ainsi que le type d'encodage du document.

Il existe bien d'autres possibilités.

Vient ensuite la hiérarchie des balises avec les nodes. Pour imager un peu, je vous propose la visualisation d'un fichier XML assez simple pour lister des voitures.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSpy v2007 rel. 3 (http://www.altova.com) by Olivier Lebeau (Heureux-oli sur
 DVP) -->
<Voiture xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
  xsi:noNamespaceSchemaLocation="C:\LOCALD~1\Forum\voiture.xsd">
   <marque>Renault</marque>
   <modele>Scénic</modele>
   <serie>Mer du nord</serie>
   <cyl>1500</cyl>
   <carburant>Diesel</carburant>
   <km>30000</km>
   <datemiseencirculation>2006-02-01</datemiseencirculation>
  </Voiture>
```

Dans ce XML très simple, on remarquera que les balises commencent par < et se terminent par >. Pour chaque balise ouvrante, il faut une balise fermante.

Nous avons un node père Voiture et des nodes enfants marque; modèle; série; cyl; carburant; km et datemiseencirculation. Entre ces différentes balises, on trouve les valeurs.

I-A-2 - Le XSD "XML Shema Description"

Le XSD, est un fichier nécessaire si l'on veut valider les données du XML. Lié à un fichier XSD, le XML ne pourra contenir un autre type de données que celui prévu par le concepteur.

Ci-dessous un exemple de XSD.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSpy v2007 rel. 3 (http://www.altova.com) by Olivier Lebeau (Heureux-oli sur
 DVP) -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
 <xs:element name="Voiture">
  <xs:annotation>
   <xs:documentation>Comment describing your root element</xs:documentation>
  </xs:annotation>
  <xs:complexType>
   <xs:sequence>
    <xs:element name="marque" type="xs:string"/>
    <xs:element name="modele" type="xs:string"/>
    <xs:element name="serie" type="xs:string"/>
    <xs:element name="cyl">
     <xs:simpleType>
      <xs:restriction base="xs:string">
       <xs:maxLength value="4"/>
      </xs:restriction>
     </xs:simpleType>
    </xs:element>
    <xs:element name="carburant">
     <xs:simpleType>
      <xs:restriction base="xs:string"/>
     </xs:simpleType>
    </xs:element>
    <xs:element name="km" type="xs:int"/>
    <xs:element name="datemiseencirculation" type="xs:date"/>
   </xs:sequence>
  </xs:complexType>
 </xs:element>
</xs:schema>
```

Dans ce XSD, on définit les données pour chaque enfant. Par exemple, marque est de type texte (string); km est de type nombre entier (int).

Si vous essayez d'entrer une autre valeur que celle prévue dans les XSD, le fichier ne pourra pas être validé.

I-A-3 - Le XSL(T) " eXtensible Stylesheet Language "

Et troisième fichier important, le XSL. Ce fichier prend toute son importance dès que l'on veut afficher les données, ou encore modifier la structure ou les données.

Un dessin vaut mieux qu'une longue explication.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="2.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:fo="http://www.w3.org/1999/XSL/Format" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:fn="http://www.w3.org/2005/xpath-functions">
<xsl:template match="/">
<html>
 <head>
  <title></title>
 </head>
 <body>
<table border="1" cellspacing="0" cellpadding="3">
 <tbody>
  <tr bgcolor="#FFFF00">
   <td>Marque</td>
   <td>Modele</td>
   <td>Serie</td>
  </tr>
  <xsl:for-each select="Voiture">
  <tr>
   <td><xsl:value-of select="marque"/></td>
   <td><xsl:value-of select="modele"/></td>
   <td><xsl:value-of select="serie"/></td>
  </tr>
  </xsl:for-each>
 </tbody>
</table>
 </body>
</html>
</xsl:template>
</xsl:stylesheet>
```

Voilà un fichier XSL lié à nos voitures. Si vous ajoutez cette ligne en plus dans le fichier XML.

```
<?xml-stylesheet type="text/xsl" href="voiture.xsl"?>
```

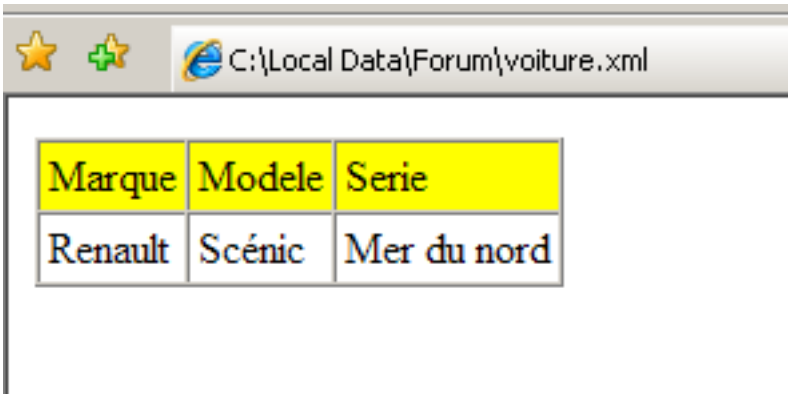
Vous obtiendrez ceci

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSpy v2007 rel. 3 (http://www.altova.com) by Olivier Lebeau (Heureux-oli sur
 DVP) -->
<?xml-stylesheet type="text/xsl" href="voiture.xsl"?>
<Voiture xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="C:\LOCALD~1\Forum\voiture.xsd">
 <marque>Renault</marque>
 <modele>Scénic</modele>
 <serie>Mer du nord</serie>
 <cyl>1500</cyl>
 <carburant>Diesel</carburant>
 <km>30000</km>
```

```
    <datemiseencirculation>2006-02-01</datemiseencirculation>
  </Voiture>
```

L'ajout de cette ligne dans notre fichier XML, va tout simplement faire référence à notre XSL.

Si vous ouvrez dans un navigateur le fichier XML, vous devriez obtenir un affichage comme ceci :



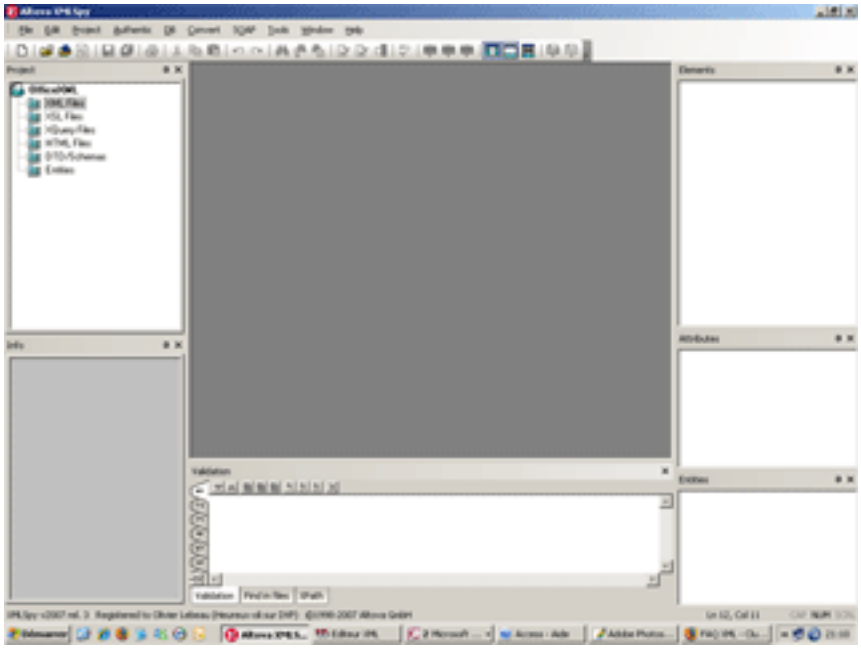
I-B - Comment arriver à ces fichiers XML

Je ne suis pas un expert en XML, il a donc fallu que je trouve quelques astuces et pour arriver à faire des fichiers cohérents, j'ai tâtonné un peu.

Il faut commencer par faire le XSD qui sera l'ossature du fichier.

XMLSpy est pour moi un allié précieux, cet outil me simplifie la vie.

On peut faire ces manipulations via le Bloc-notes ou avec n'importe quel autre éditeur de texte, mais si vous jetez un #il aux fichiers ci-dessus, vous verrez que ce n'est pas si simple.

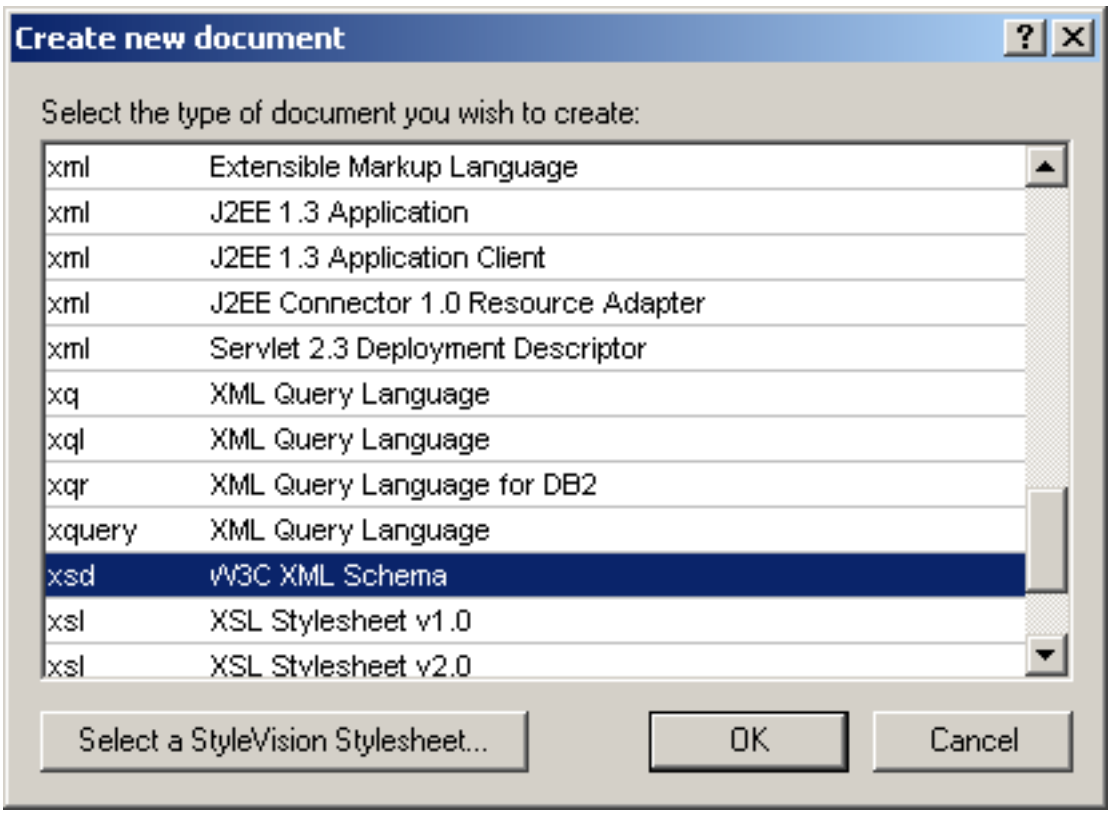


L'editeur XML

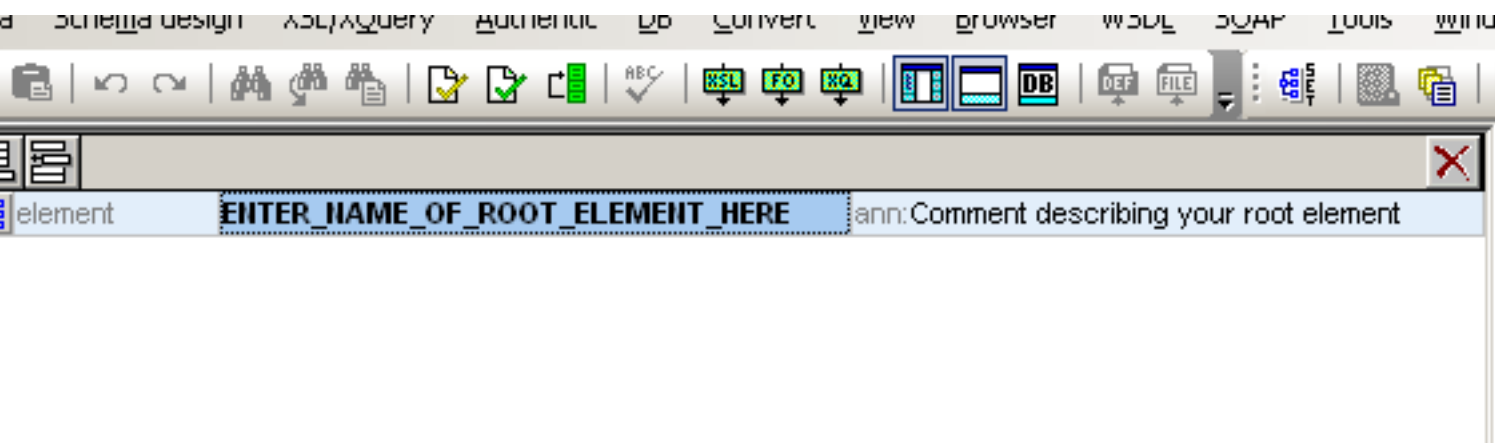
Une fois l'éditeur ouvert, nous allons créer un nouveau fichier XSD.

Comme pour tous les logiciels, File -> New pour arriver à une boîte de dialogue.

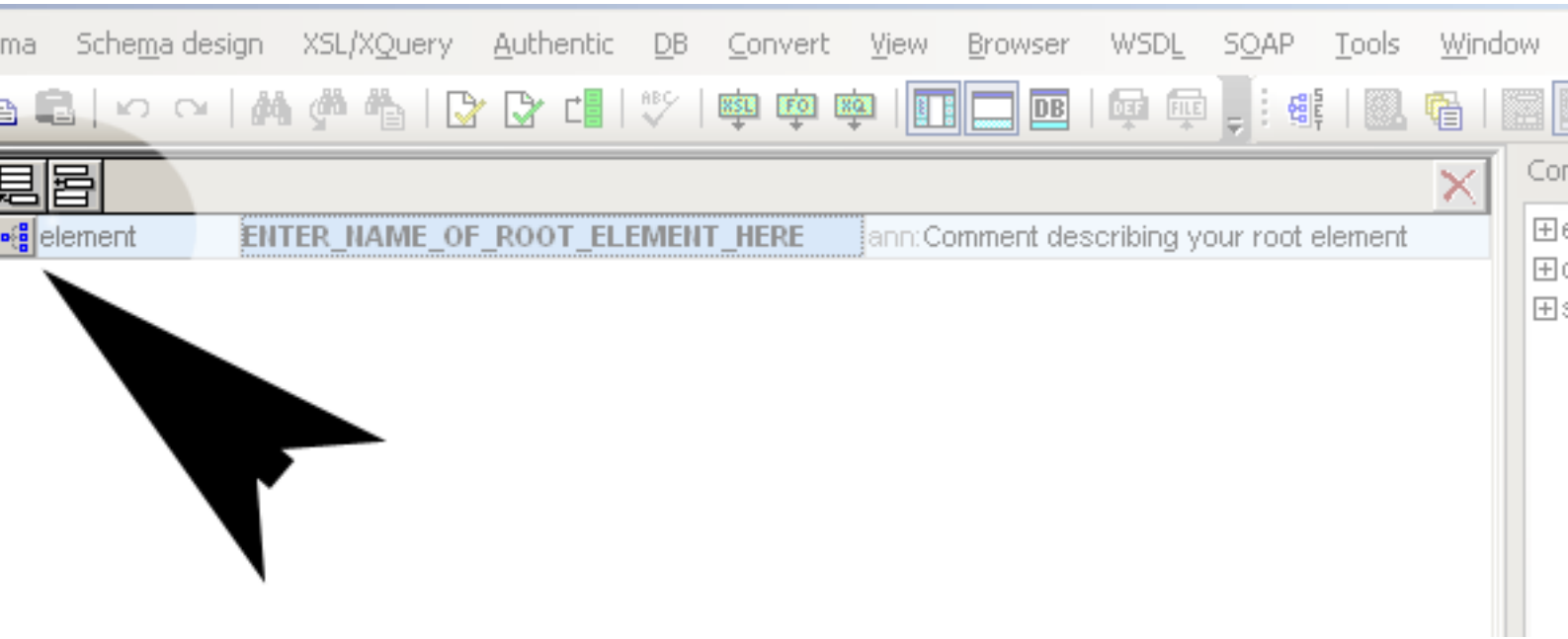
Le choix de fichier : XSD.



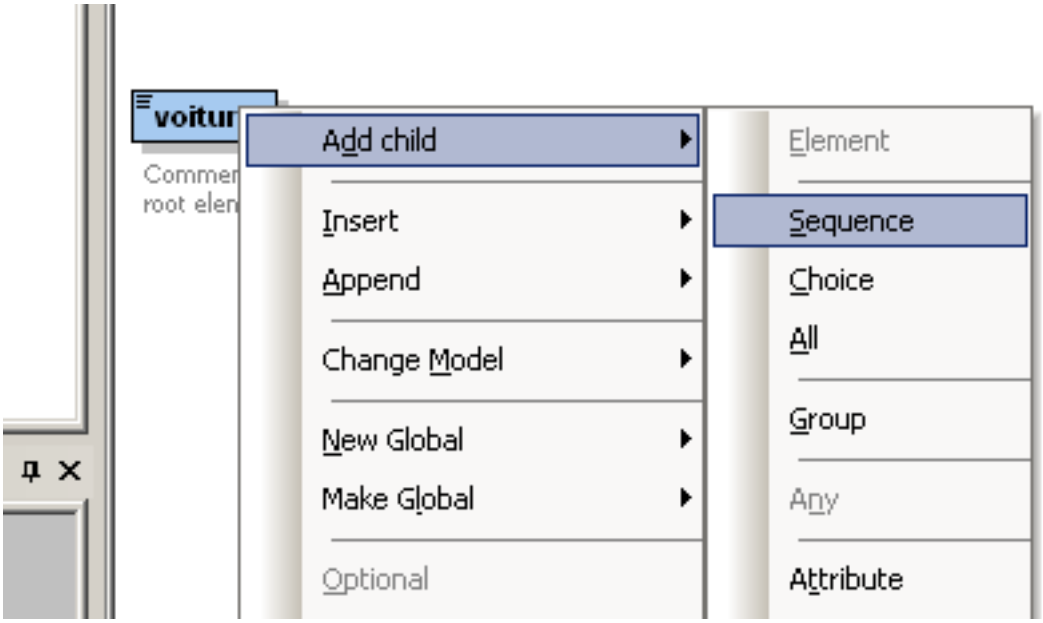
Voilà comment se présente la fenêtre de l'éditeur.



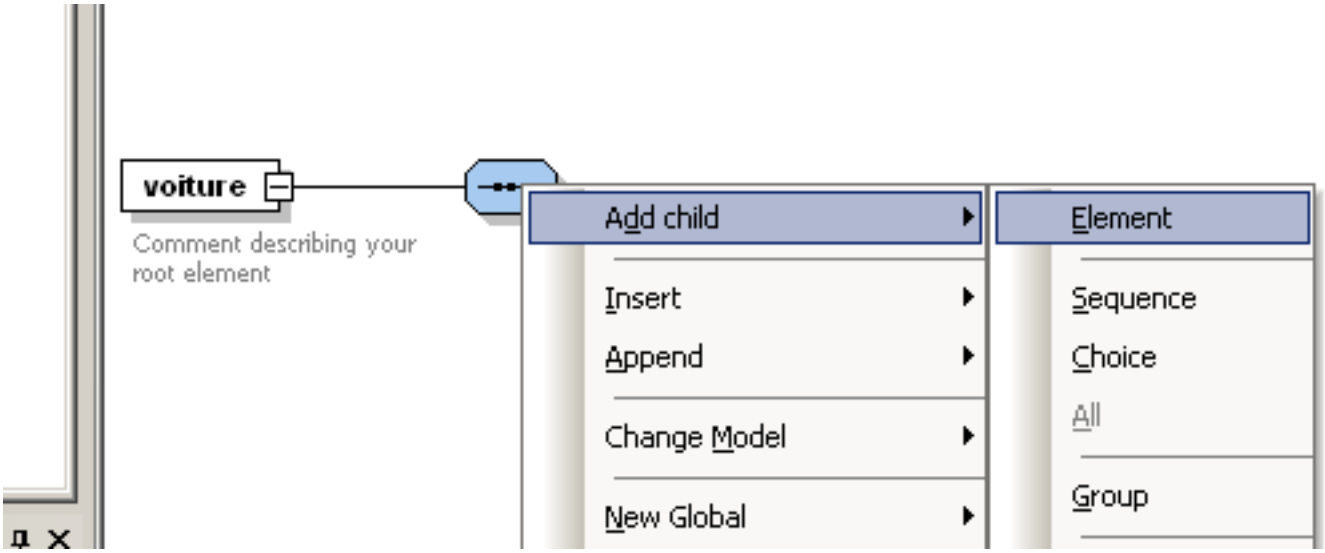
Pas très convivial, mais on va changer l'aspect.



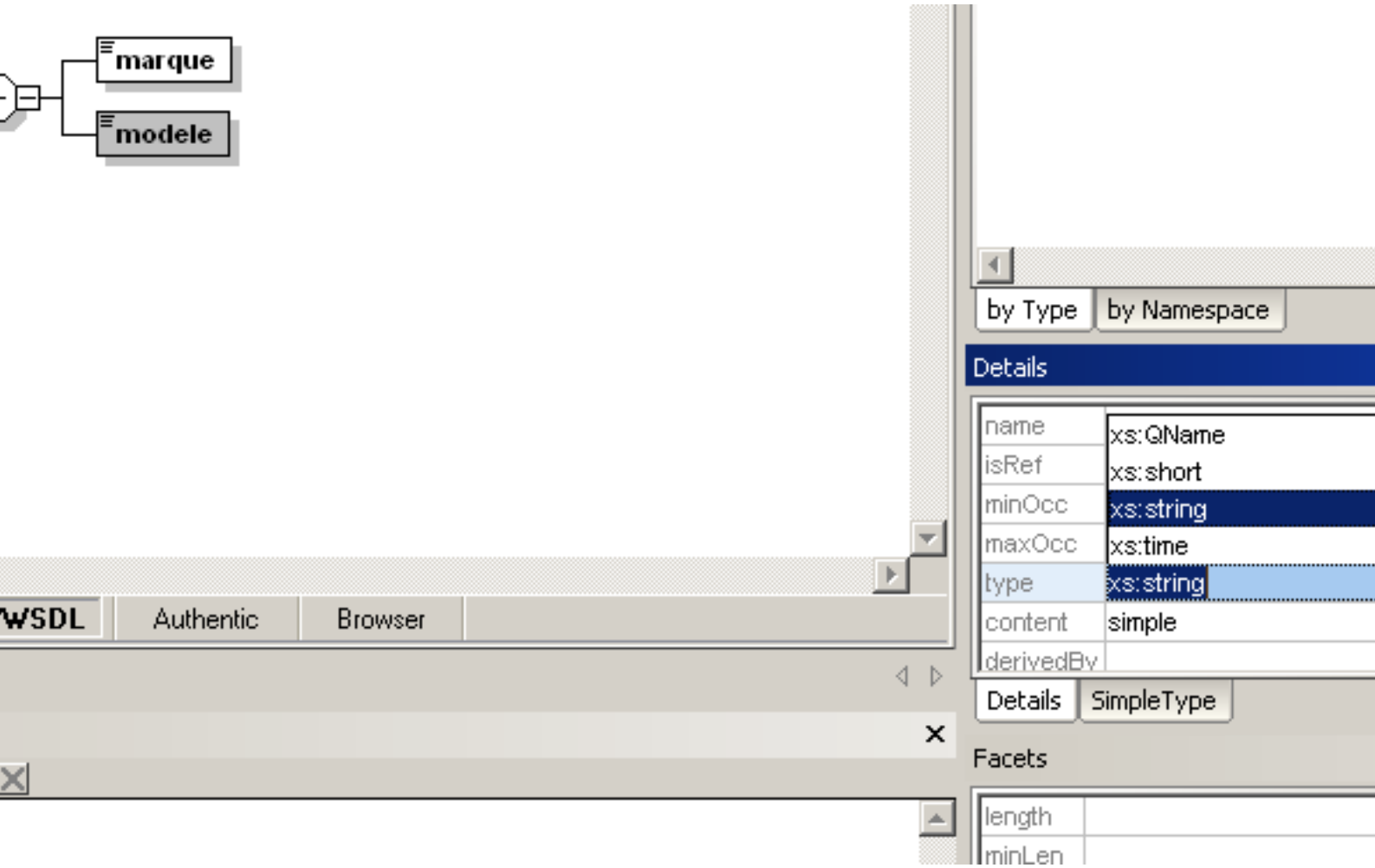
Petit clic pour basculer vers un mode graphique. On change le nom de l'élément et avec un clic droit, on va ajouter une séquence.



Sur le symbole de la séquence, on va recommencer l'opération clic droit et ajouter des éléments enfants.

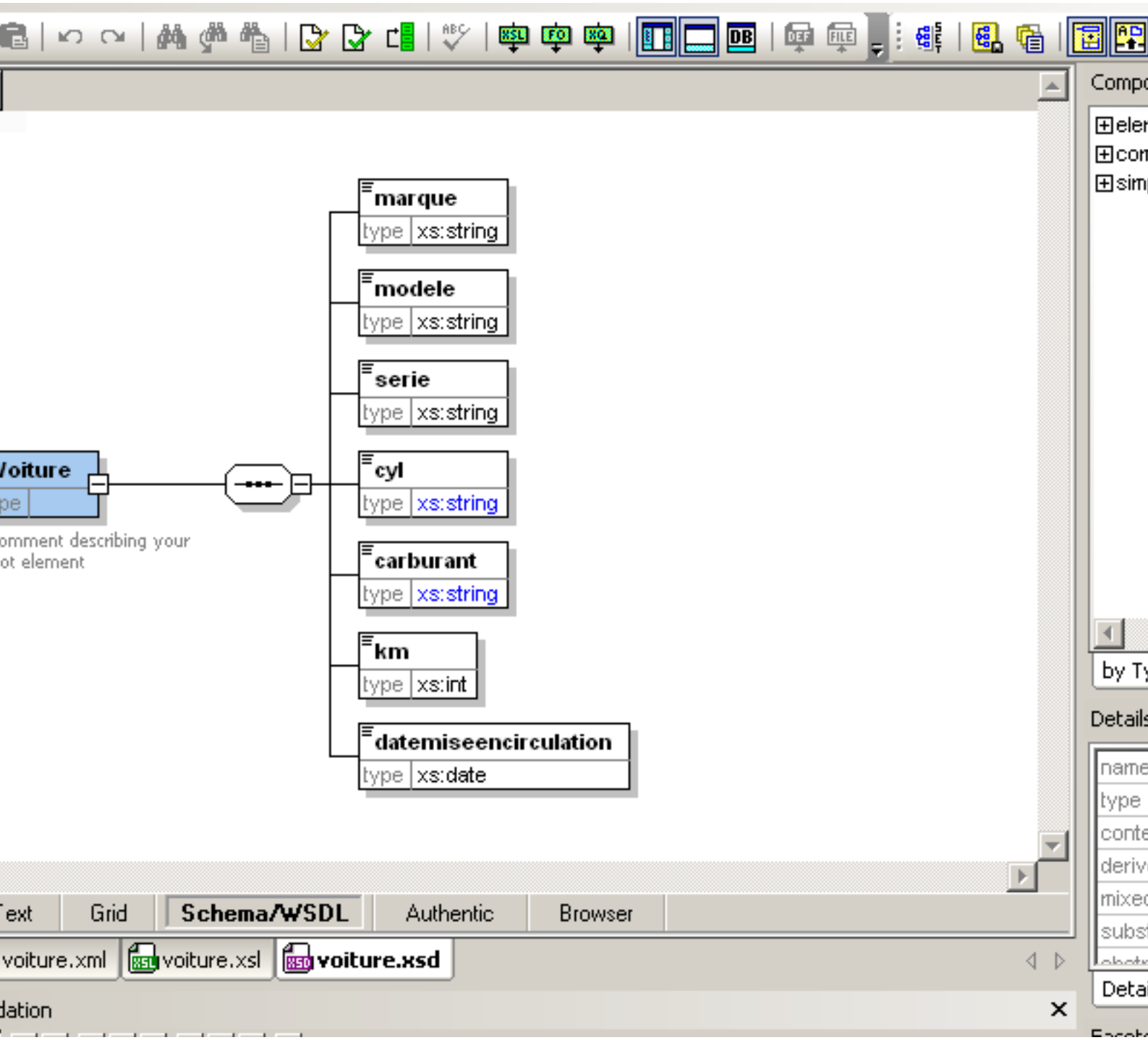


On donne un nom à cet enfant, on peut en ajouter plusieurs. Chose intéressante, on peut définir le type de données de l'élément. Pour le modèle, j'ai choisi le type **xs:string**.



Si on passe en mode texte, on se retrouve avec le fichier repris ci-dessus.

Voilà à quoi devrait ressembler votre fichier.



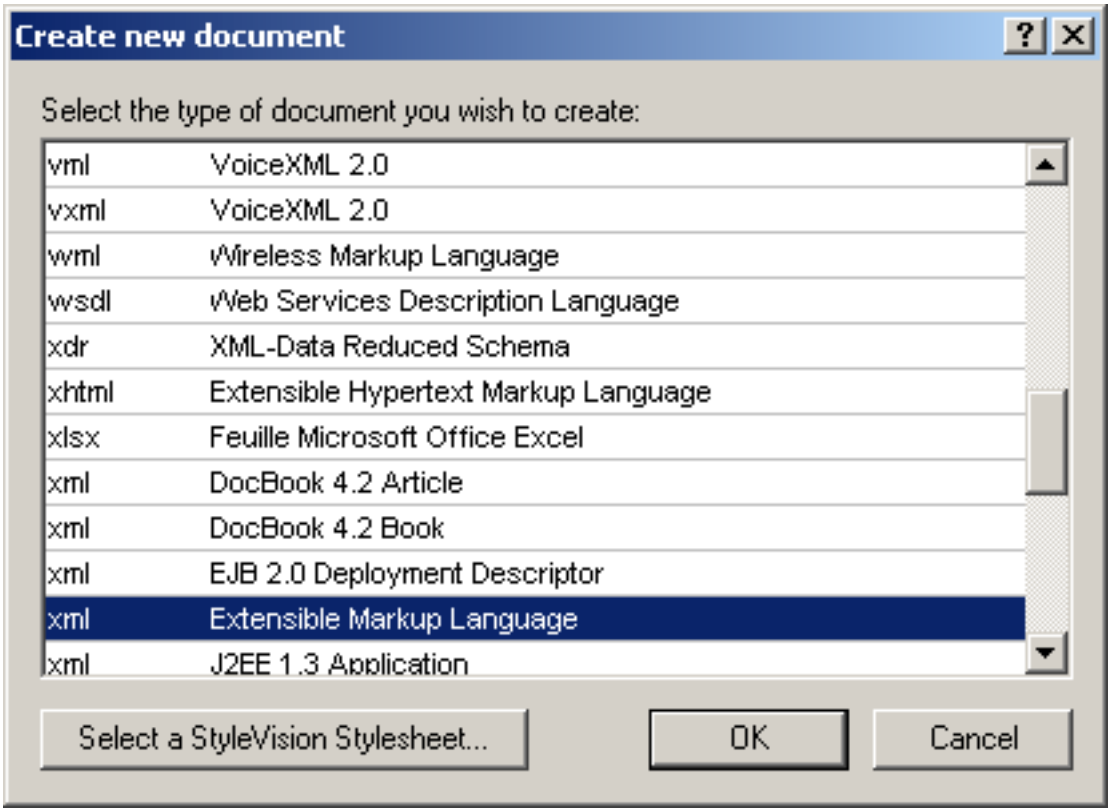
Nous allons utiliser ce fichier tout au long de l'article.

Une fois le fichier schéma terminé, il ne nous reste plus qu'à faire notre fichier XML.

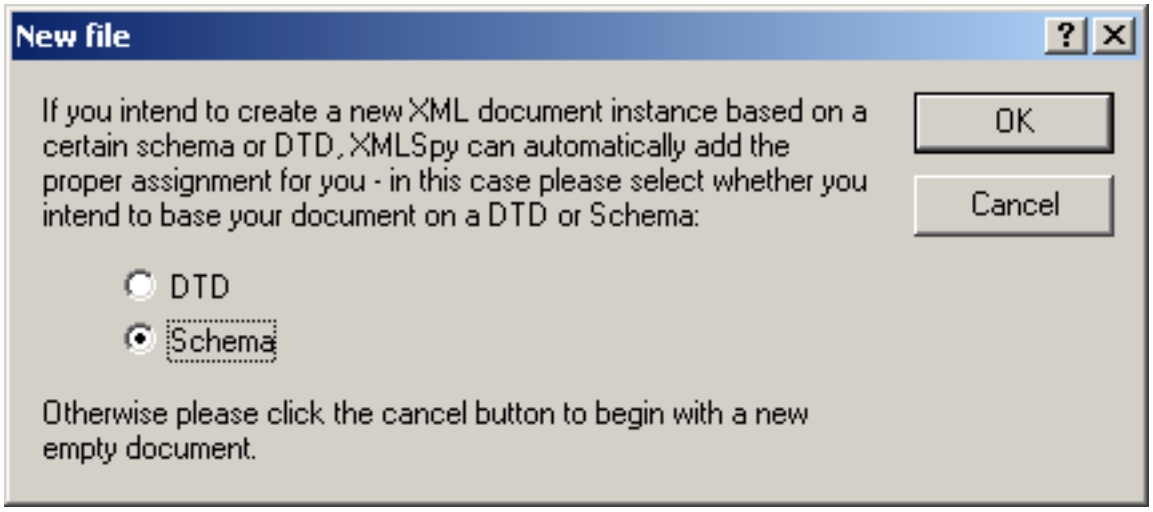
Ce fichier est encore plus facile à réaliser.

Toujours le même principe : File -> New.

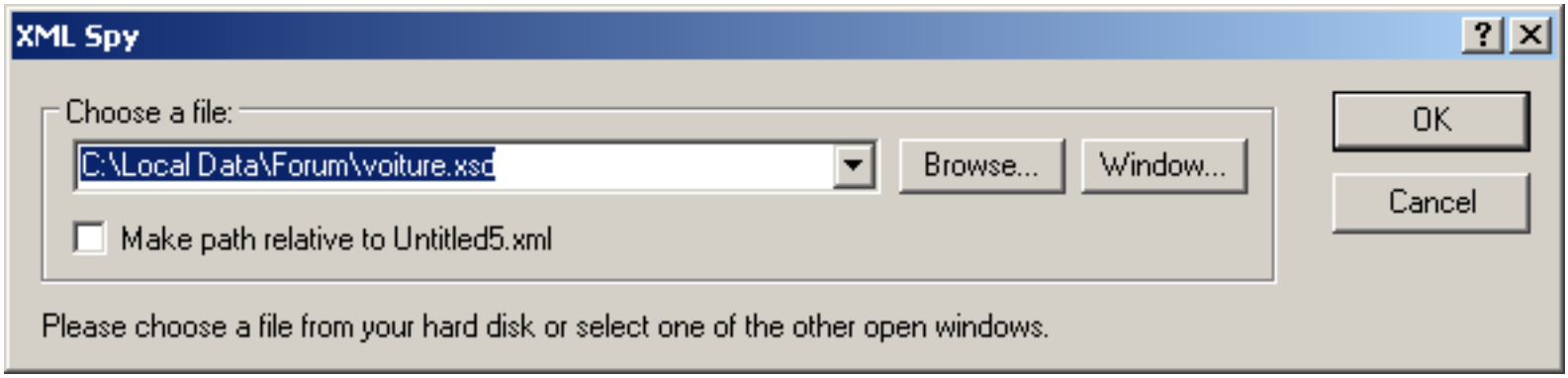
Notre boîte de dialogue et le choix pour un fichier XML



A nouveau un choix, comme nous venons de faire un fichier XSL, nous allons baser notre XML sur ce dernier.



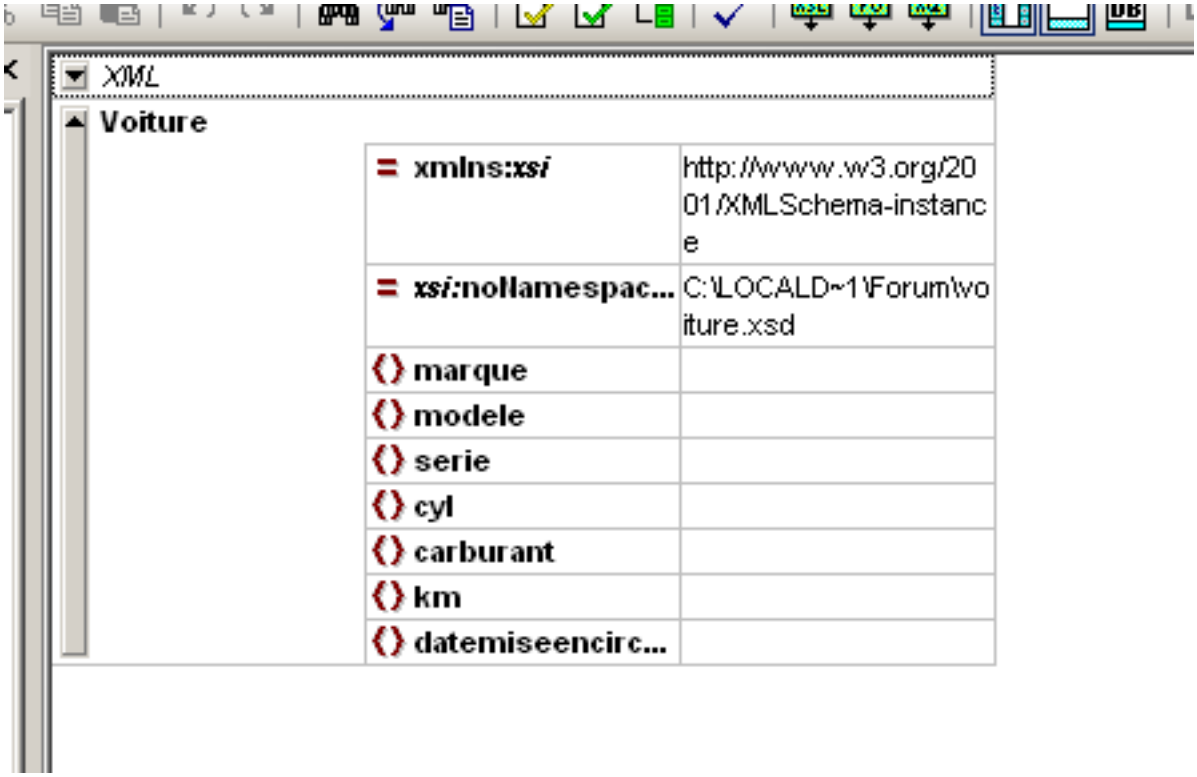
Et le choix du XSD qui va être utilisé.



Son résultat

```
1   <?xml version="1.0" encoding="UTF-8"?>
2   <Voiture xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="C:\LOCALD~1\Forum\voiture.xsd">
3       <marque/>
4       <modele/>
5       <serie/>
6       <cyl/>
7       <carburant/>
8       <km/>
9       <datemiseencirculation/>
10  </Voiture>
11
```

Si l'on demande l'affichage de la grille "Grid"



Finalement pour le XSL, il faut un module complémentaire que je n'ai pas, j'ai consulté la partie de DVP dévolue au XML et j'ai utilisé le Bloc-notes.

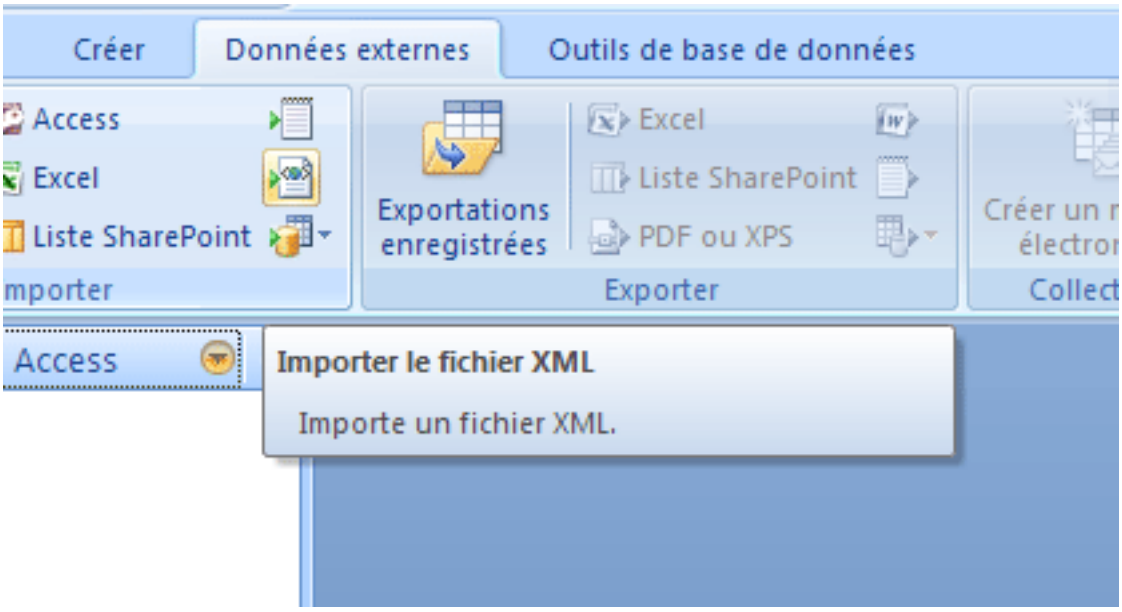
II - Access 2007 et le XML

J'ai utilisé Access 2007, mais il est possible d'utiliser tout ou en partie cet article pour la version 2003.

II-A - Utilisation simple

II-A-1 - Importation des données

Access permet l'utilisation du XML pour les données. Pour incorporer des données XML dans Access, il suffit d'utiliser la méthode d'importation classique et de choisir un fichier XML.

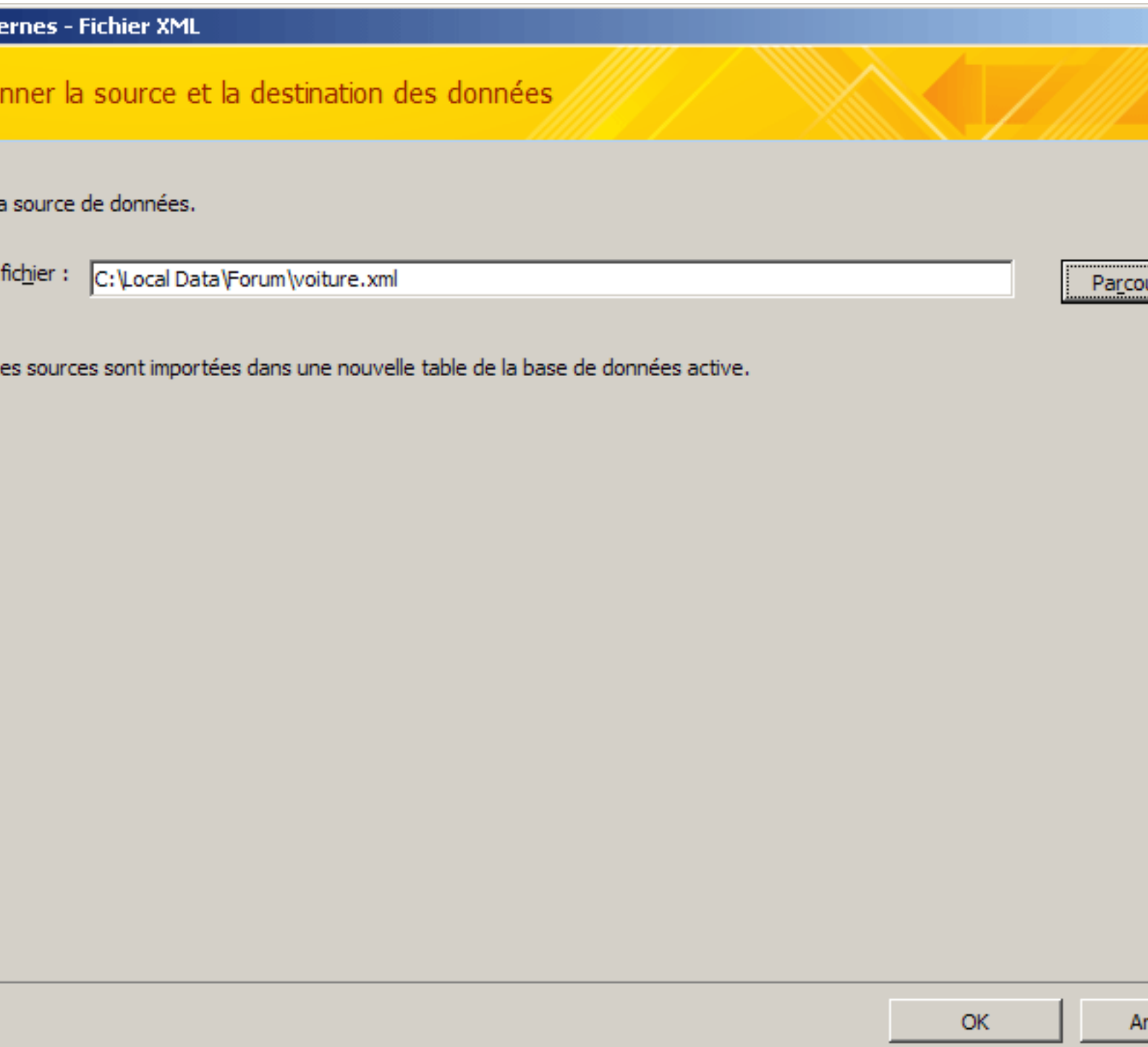


Importation d'un fichier XML

Lier un fichier XML n'est pas possible, Access ne propose que l'importation.

Nous allons donc importer le fichier dans Access. Cette importation ne comporte aucune difficulté, Access fait tout.

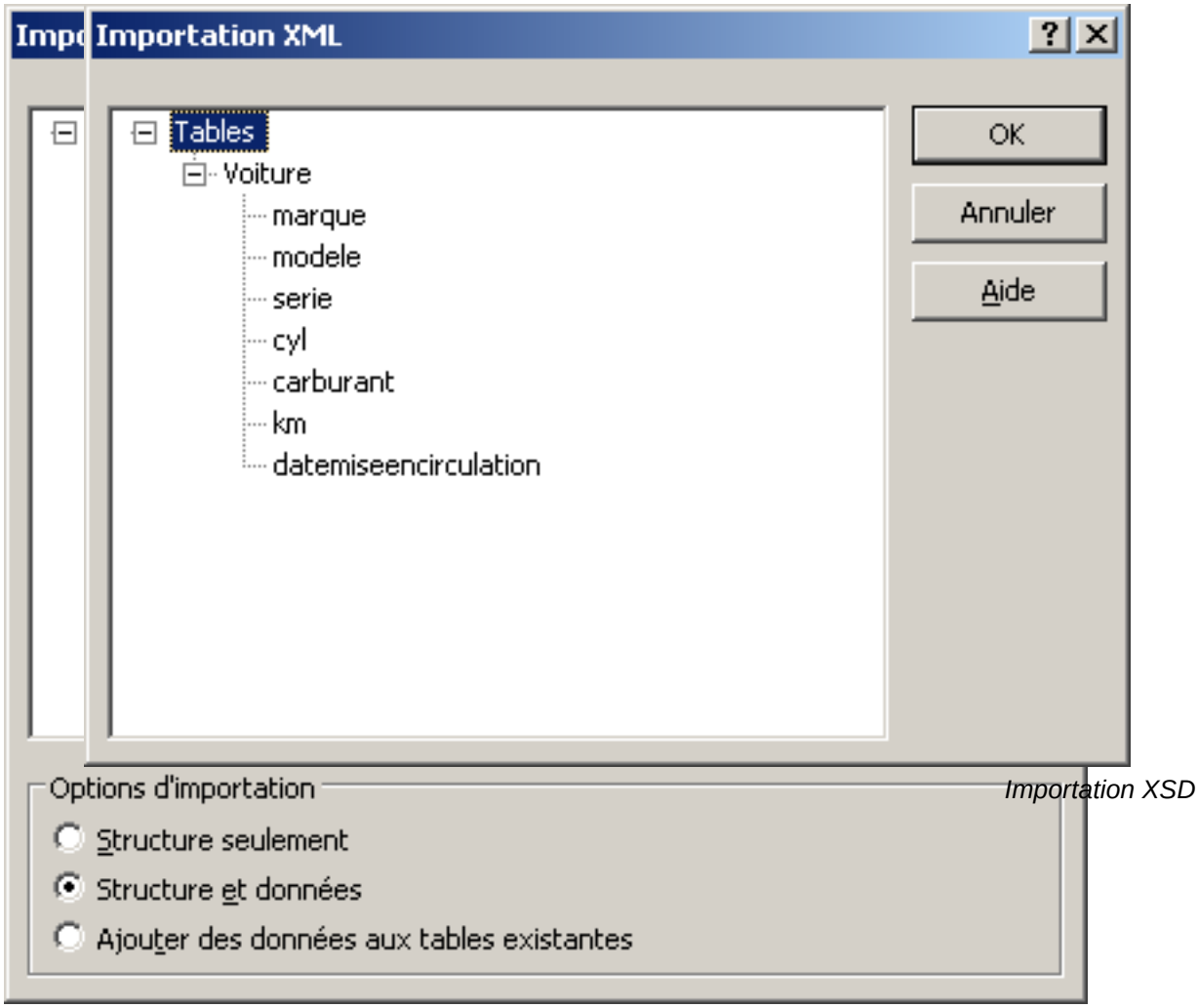
La première étape, la sélection du fichier.



Seconde étape, le choix du traitement à réserver au fichier.

- Importer la structure.
- Importer Structure et Données.
- Ajouter les données.

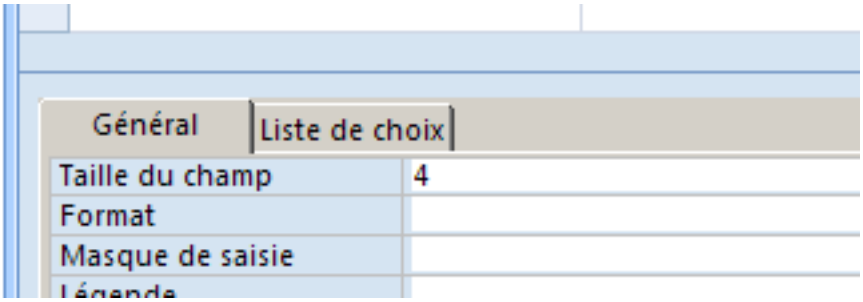
Dans le cas de la sélection du XSD, aucun choix n'est proposé, seule la structure est importée et heureusement, ce fichier ne contient qu'un schéma de document XML.



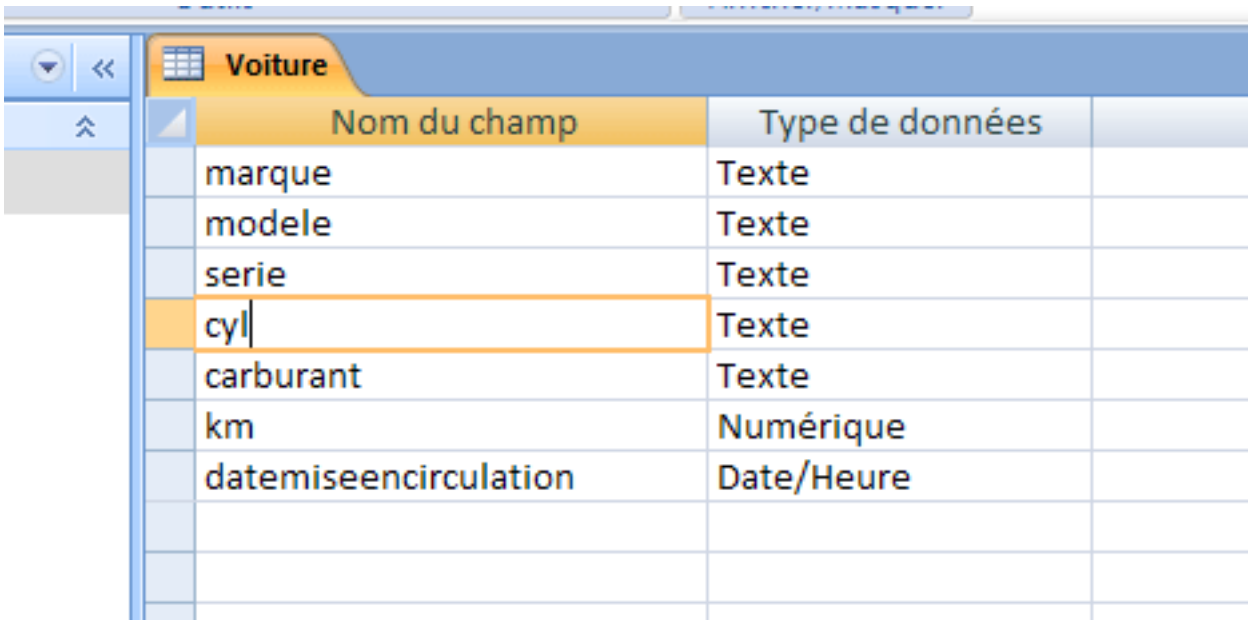
Importation XML

Importation XSD

Si votre schéma est correctement construit, Access donne aux champs les propriétés que vous avez définies aux éléments de votre XSD.



les propriétés des champs



les champs de la table

On se retrouve donc avec nos données XML dans une table Access.

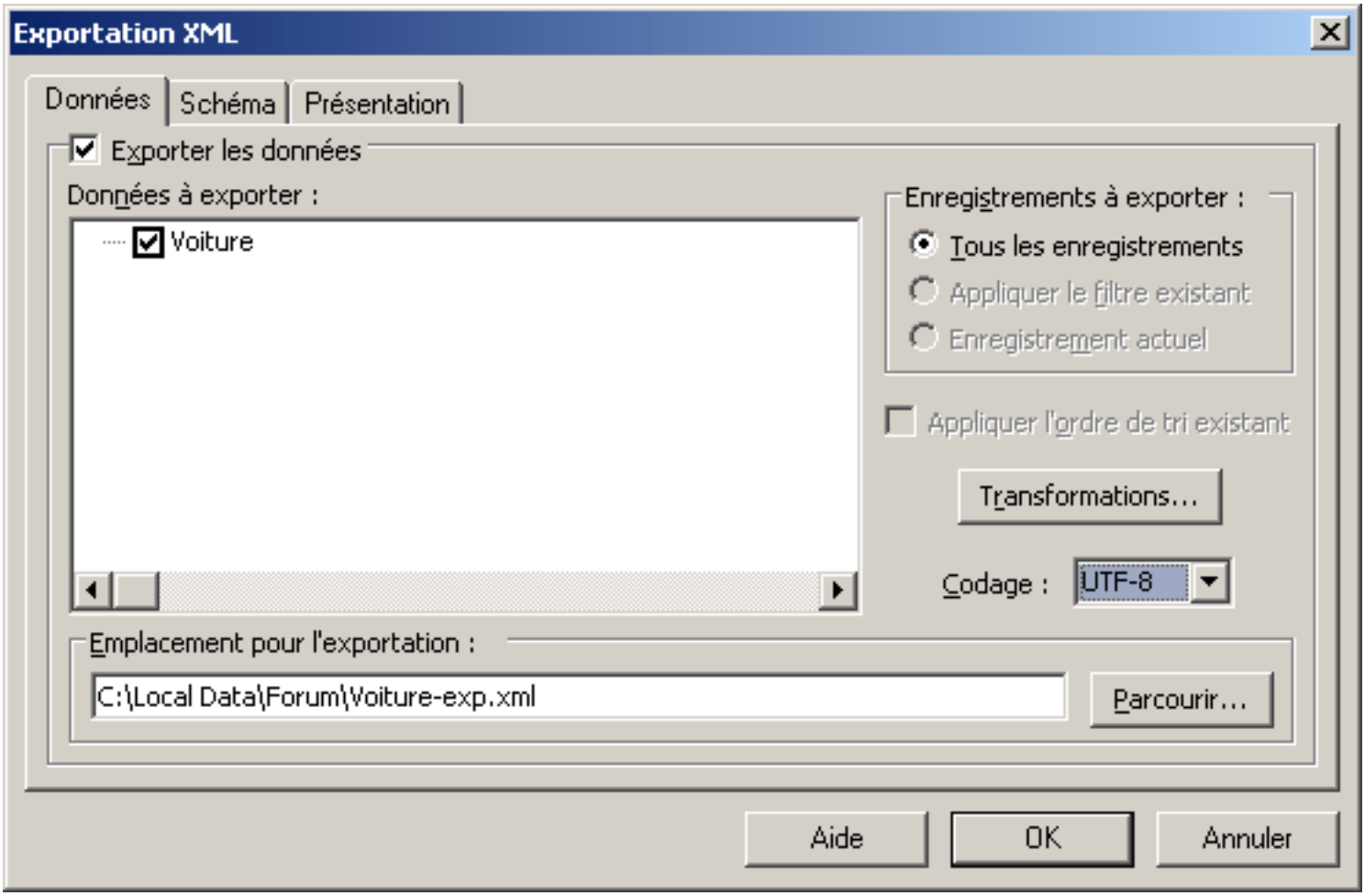
II-A-2 - Exportation des données.

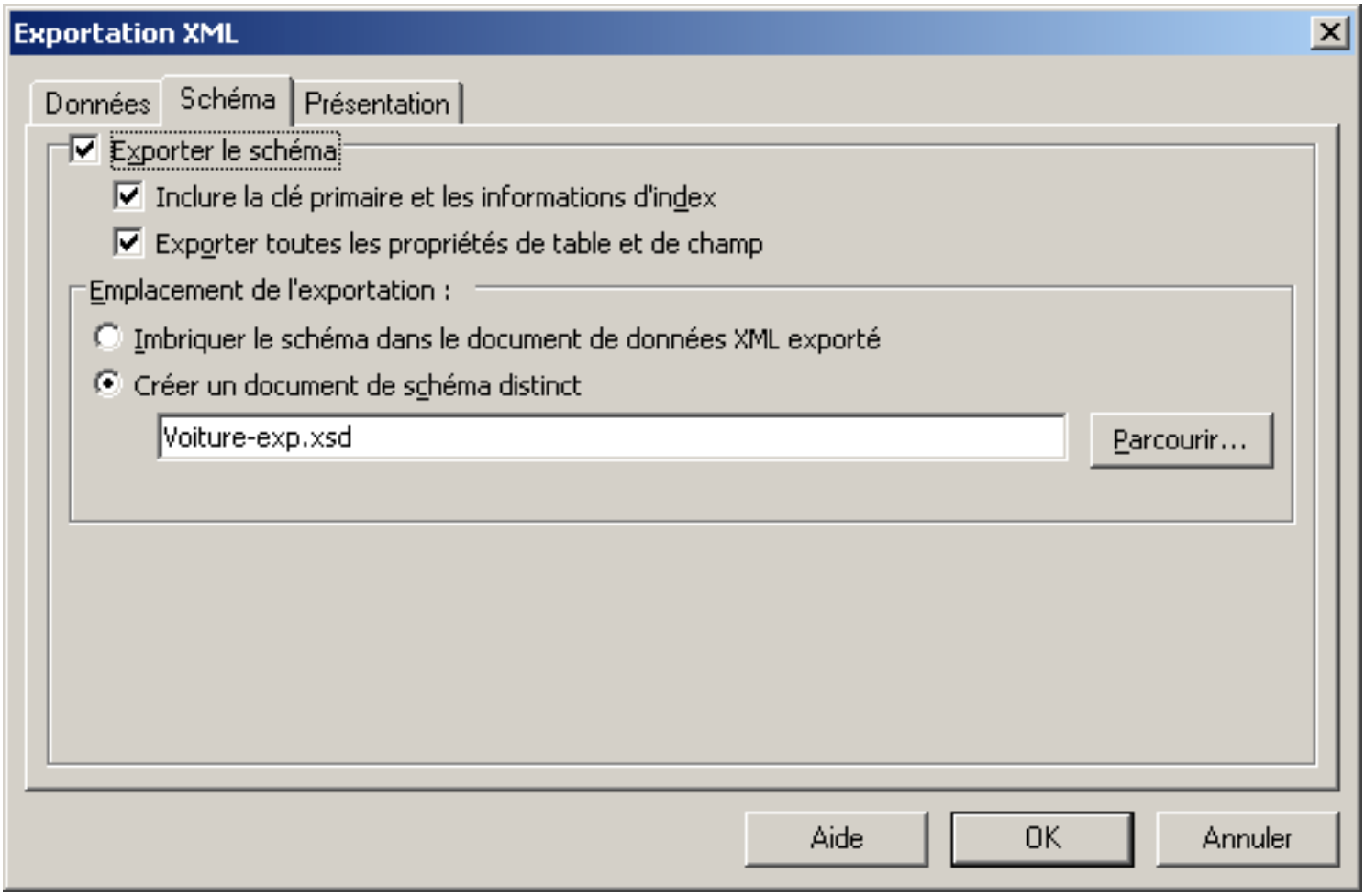
Nous allons maintenant aborder la façon dont Access va exporter nos données vers un fichier XML et comparer le fichier de départ avec le fichier obtenu.

Pour l'exportation, Clic droit sur la table et dans le menu contextuel, Exporter fichier XML.

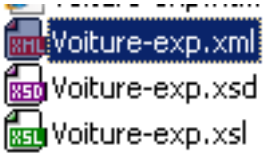
Pour éviter d'écraser le fichier de départ, on va ajouter "-exp" dans le nom du fichier, ce qui sera aussi plus commode pour faire la comparaison.

Une nouvelle boîte de dialogue apparait. Un bouton "Autres Options" permet des choix un peu plus pointus pour l'exportation.





On se retrouve avec trois fichiers :



Un fichier XML, un fichier XSD et le dernier un XSL.

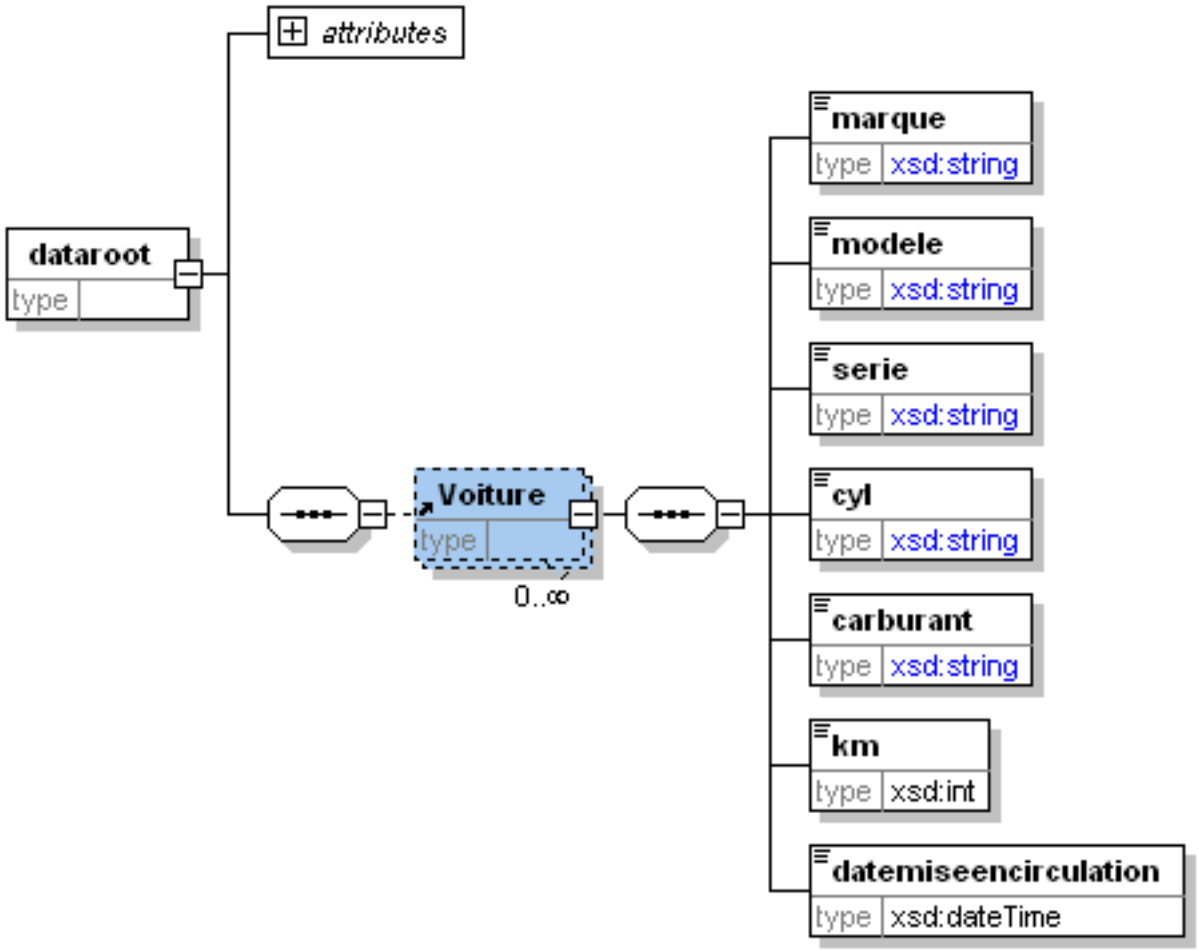
II-A-2-a - voiture-exp.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<dataroot xmlns:od="urn:schemas-microsoft-com:officedata"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="Voiture-exp.xsd" generated="2007-06-16T13:05:08">
<Voiture>
<marque>Renault</marque>
<modele>Scenic</modele>
<serie>Mer du Nord</serie>
<cyl>1500</cyl>
<carburant>Diesel</carburant>
<km>30000</km>
<datemiseencirculation>2006-02-14T00:00:00</datemiseencirculation>
</Voiture>
```

```
</dataroot>
```

Il n'est pas fondamentalement différent du fichier d'origine. On peut donc utiliser Access comme outil pour entrer des données dans un fichier XML.

II-A-2-b - voiture-exp.xsd



Le fichier obtenu n'est pas très différent du fichier initial. Dans le fichier initial, il n'y avait pas de taille pour les éléments texte, alors que maintenant, on se retrouve avec un taille limitée à 255 caractères.

II-A-2-c - voiture-exp.xsl

Erreur lors de la transformation XLST : Une fonction d'extension XPath inconnue a été appelée.

Pour le XSL, il n'en va pas de même. Ce fichier semble ne pas être correct.

II-A-3 - Comment procéder avec plusieurs tables ?

II-A-3-a - Export des données

Dans les paragraphes précédents, nous n'avons utilisé qu'une seule table. Or il est rare de trouver des bases de données ne comportant qu'une seule table.

Access permet de transférer vers un fichier XML plusieurs tables.

Pour transférer plusieurs tables dans un même fichier XML, Access dispose d'une méthode "CreateAdditionalData" qui permet de créer un objet "AdditionalData". Avec ces deux méthodes, nous allons pouvoir ajouter les tables liées à la table principale.

La procédure suivante permet de transférer dans une série de fichier XLM les données contenues dans plusieurs tables, pour autant qu'il existe une relation entre ces tables.

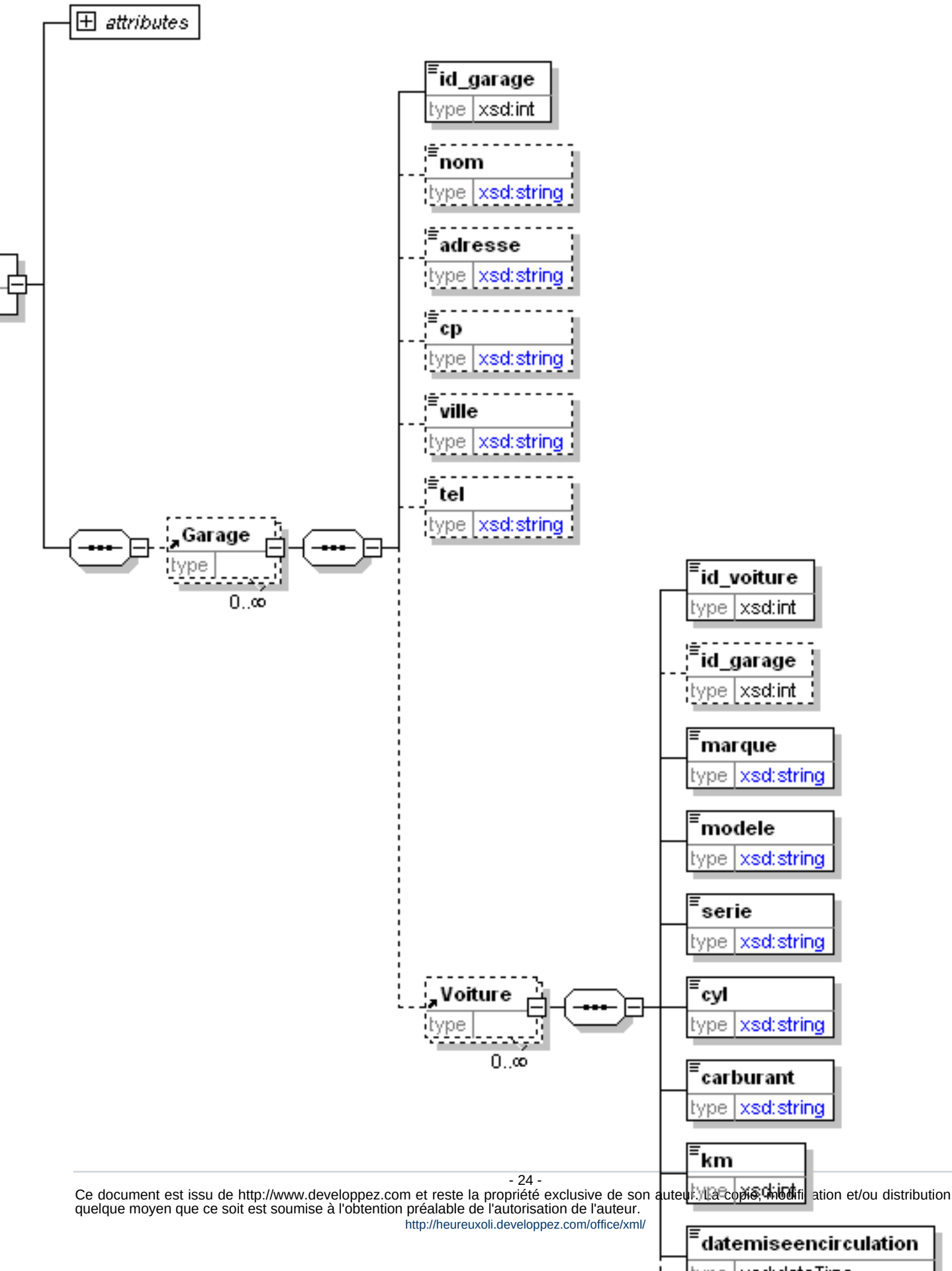
```
Public Sub ExportTablesXML()
Dim objAD As AdditionalData

Set objAD = Application.CreateAdditionalData

objAD.Add "Voiture"

Application.ExportXML acExportTable, "Garage", _
    "C:\temp\garage.xml", "c:\temp\garage.xsd", _
    "c:\temp\garage.xsl", AdditionalData:=objAD



End Sub
```

Le fait de ne mentionner que la table "Voiture" pour l'exportation n'est pas un problème, Access va aussi exporter la table garage.

Si on ajoute une table supplémentaire, la table "chauffeur" par exemple, lors de l'export, il faut l'ajouter avec la méthode Add. Le code de notre procédure devient donc :

```
Public Sub ExportTablesXML()
Dim objAD As AdditionalData

Set objAD = Application.CreateAdditionalData

With objAD
    .Add "Voiture"
    objAD("Voiture").Add "Chauffeur"
End With
Application.ExportXML acExportTable, "Garage", _
    "C:\temp\garage.xml", "c:\temp\garage.xsd", _
    "c:\temp\garage.xsl", AdditionalData:=objAD



End Sub
```

Pour vérifier le résultat, je vais utiliser XMLSpy et afficher le XSD obtenu.



Le résultat est conforme aux espérances, on retrouve notre structure hiérarchique.

Les données le sont tout autant :

Fichier XML

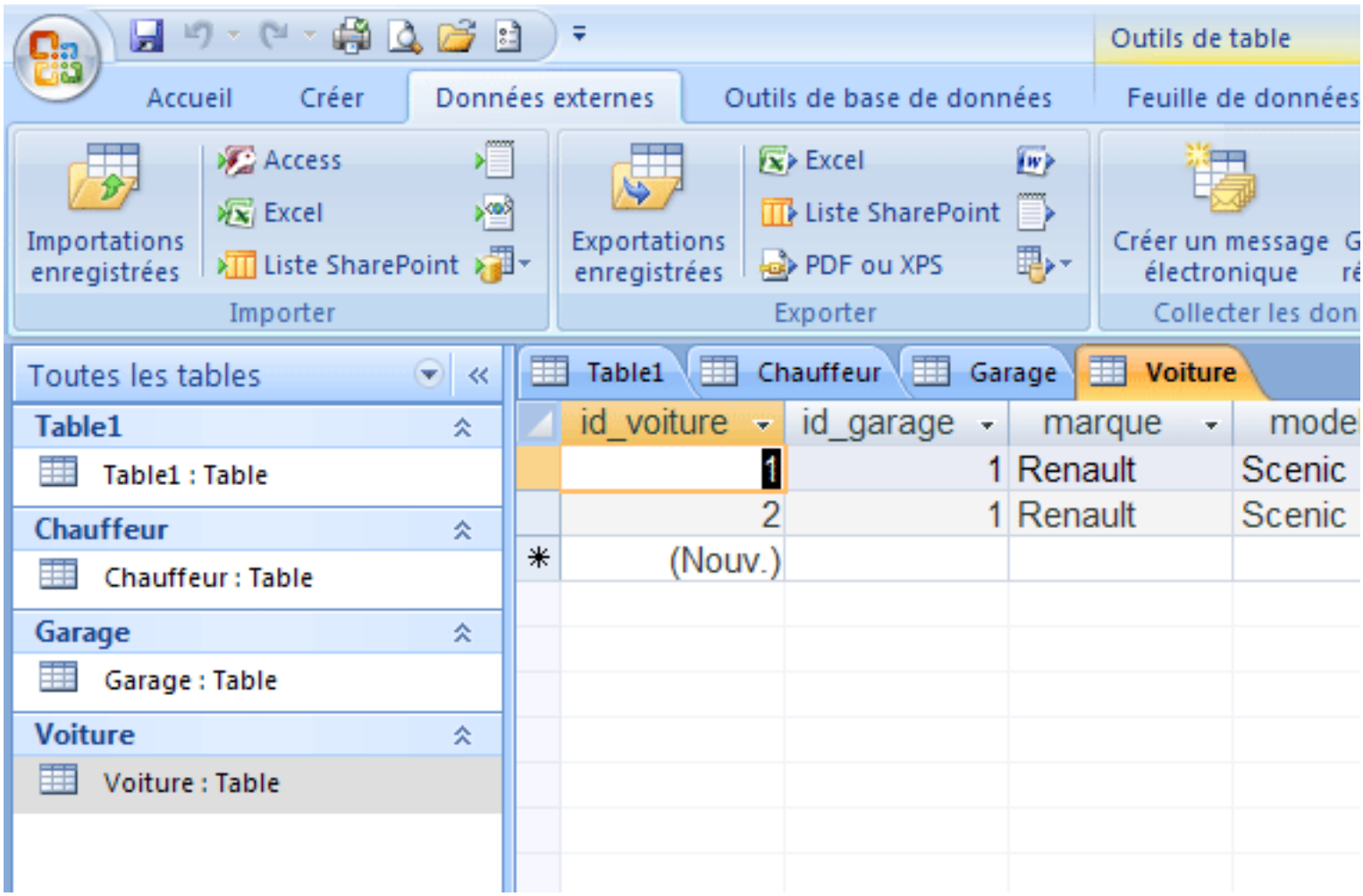
```
<?xml version="1.0" encoding="UTF-8"?>
<dataroot xmlns:od="urn:schemas-microsoft-com:officedata"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="garage.xsd" generated="2007-07-29T08:52:44">
<Garage>
<id_garage>1</id_garage>
<nom>fama</nom>
<adresse>rue du trieu</adresse>
<cp>7100</cp>
<ville>Manage</ville>
<tel>05/56.56.55</tel>
<Voiture>
<id_voiture>1</id_voiture>
<id_garage>1</id_garage>
<marque>Renault</marque>
<modele>Scenic</modele>
<serie>Mer du Nord</serie>
<cyl>1500</cyl>
<carburant>Diesel</carburant>
<km>30000</km>
<datemiseencirculation>2006-02-14T00:00:00</datemiseencirculation>
<Chauffeur>
<id_Chauffeur>1</id_Chauffeur>
<Id_Voiture>1</Id_Voiture>
<NomChauff>Marie</NomChauff>
</Chauffeur>
<Chauffeur>
<id_Chauffeur>2</id_Chauffeur>
<Id_Voiture>1</Id_Voiture>
<NomChauff>Olivier</NomChauff>
</Chauffeur>
</Voiture>
<Voiture>
<id_voiture>2</id_voiture>
<id_garage>1</id_garage>
<marque>Renault</marque>
<modele>Scenic</modele>
<serie>II</serie>
<cyl>1900</cyl>
<carburant>Diesel</carburant>
<km>130000</km>
<datemiseencirculation>1997-07-17T00:00:00</datemiseencirculation>
<Chauffeur>
<id_Chauffeur>3</id_Chauffeur>
<Id_Voiture>2</Id_Voiture>
<NomChauff>Claudine</NomChauff>
</Chauffeur>
</Voiture>
</Garage>
</dataroot>
```

Une fois de plus, le XSL obtenu n'est pas d'une grande utilité. Il contient des scripts qui empêchent son utilisation avec certains logiciels par mesure de sécurité.

II-A-3-b - Import des données

La méthode d'import des données est la même qu'au paragraphe II-A-1

Maintenant que nous avons réussi à exporter nos données dans un fichier XML, nous allons importer nos données dans une nouvelle base de données.



Les tables avec nos données ont bien été importées, seules les relations sont manquantes.

En comparant les données de la base de données obtenue et de la base de données de départ, rien ne manque.

III - Excel et le XML

Excel se révèle être un bon outil pour les fichiers XML simples. La méthode pour ajouter des données est on ne peut plus facile, on ajoute simplement des lignes dans un tableau.

III-A - Schéma XML ou XSD

Dans tous les cas de figure, il faut un schéma XML avant de commencer. Excel ne pouvant générer de fichiers XSD, une fois de plus, nous allons avoir recours à XMLSpy.

Pour faire dans un autre registre, nous allons faire un répertoire. Pour ce carnet d'adresses, nous allons avoir besoin d'au moins un nom, un prénom, une adresse, une localité, un code postal, un numéro de téléphone et pour terminer une date de naissance. Selon la terminologie Excel, le carnet d'adresse sera l'élément parent. Les autres données seront les éléments enfants et dans ces éléments enfants, le nom sera un élément requis.

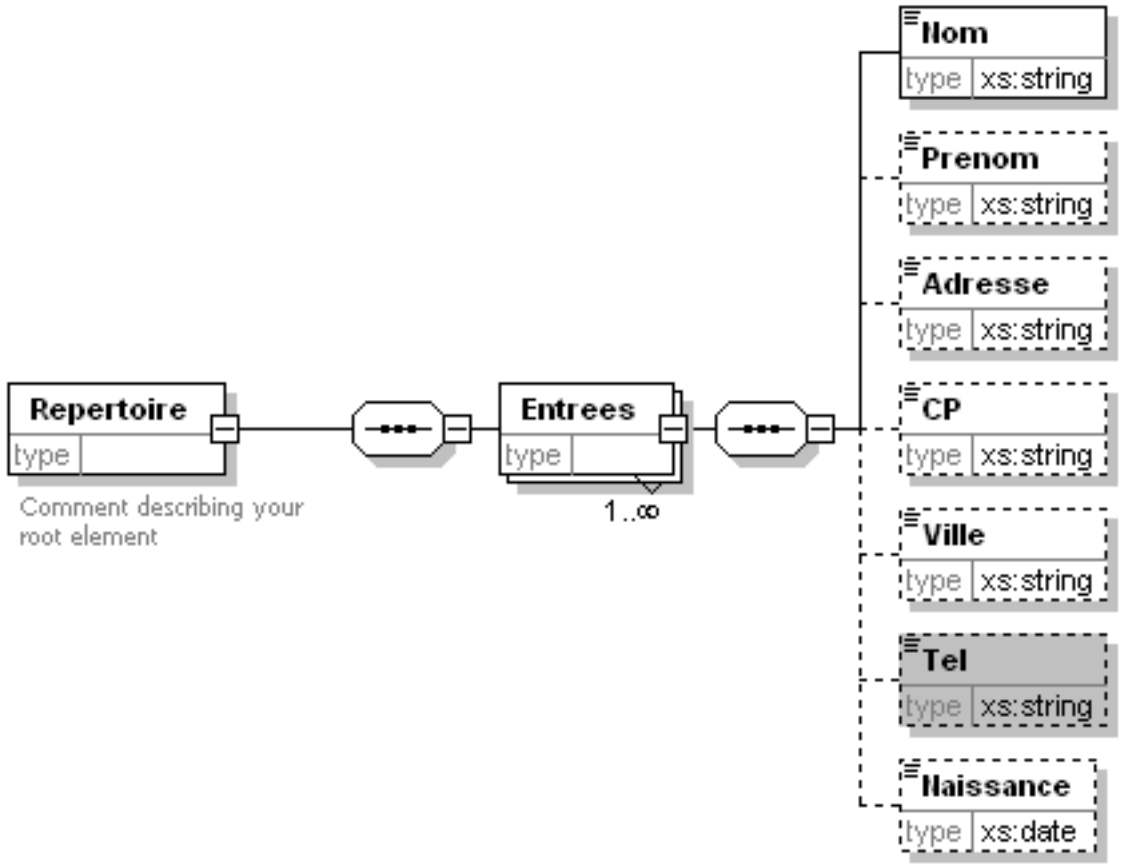
Le format du fichier se doit d'être un XSD.

Type d'élément	Icône
Élément parent	
Élément parent requis	
Élément parent répété	
Élément parent répété requis	
Élément enfant	
Élément enfant requis	
Élément enfant répété	
Élément enfant répété requis	
Attribut	
Attribut requis	
Contenu simple dans une structure complexe	
Contenu simple requis dans une structure complexe	

Terminologie et représentation

III-A-1 - Construction de notre schéma

Même procédé qu'au I-B Pour le résultat ci-dessous :

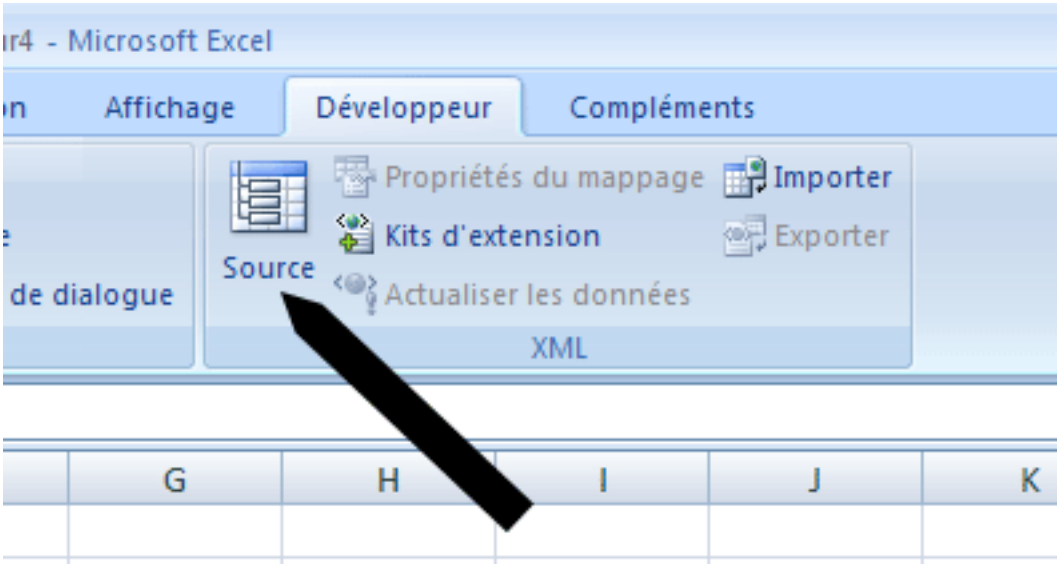


En texte voilà ce que donne notre schéma

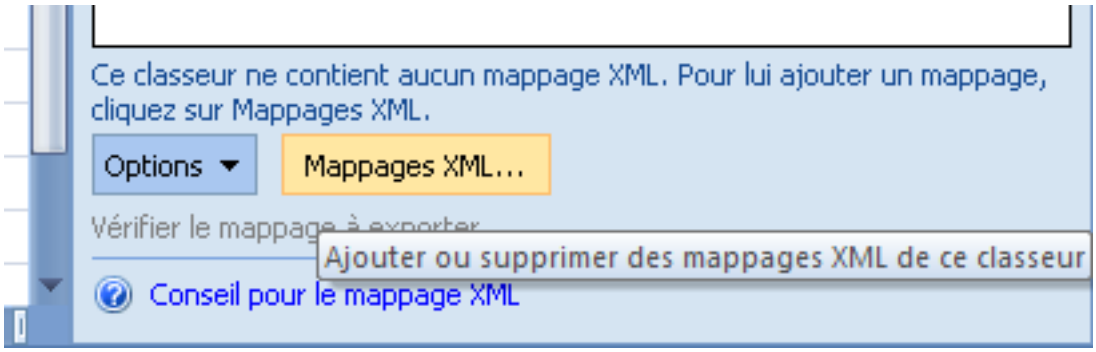
```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" attributeFormDefault="unqualified">
 <xs:element name="Repertoire">
  <xs:annotation>
   <xs:documentation>Comment</xs:documentation>
  </xs:annotation>
  <xs:complexType>
   <xs:sequence>
    <xs:element name="Entrees" maxOccurs="unbounded">
     <xs:complexType>
      <xs:sequence>
       <xs:element name="Nom" type="xs:string"/>
       <xs:element name="Prenom" type="xs:string" minOccurs="0"/>
       <xs:element name="Adresse" type="xs:string" minOccurs="0"/>
       <xs:element name="CP" type="xs:string" minOccurs="0"/>
       <xs:element name="Ville" type="xs:string" minOccurs="0"/>
       <xs:element name="Tel" type="xs:string" minOccurs="0"/>
       <xs:element name="Naissance" type="xs:date" minOccurs="0"/>
      </xs:sequence>
     </xs:complexType>
    </xs:element>
   </xs:sequence>
  </xs:complexType>
 </xs:element>
</xs:schema>
```

III-B - Le XML : enfin les données

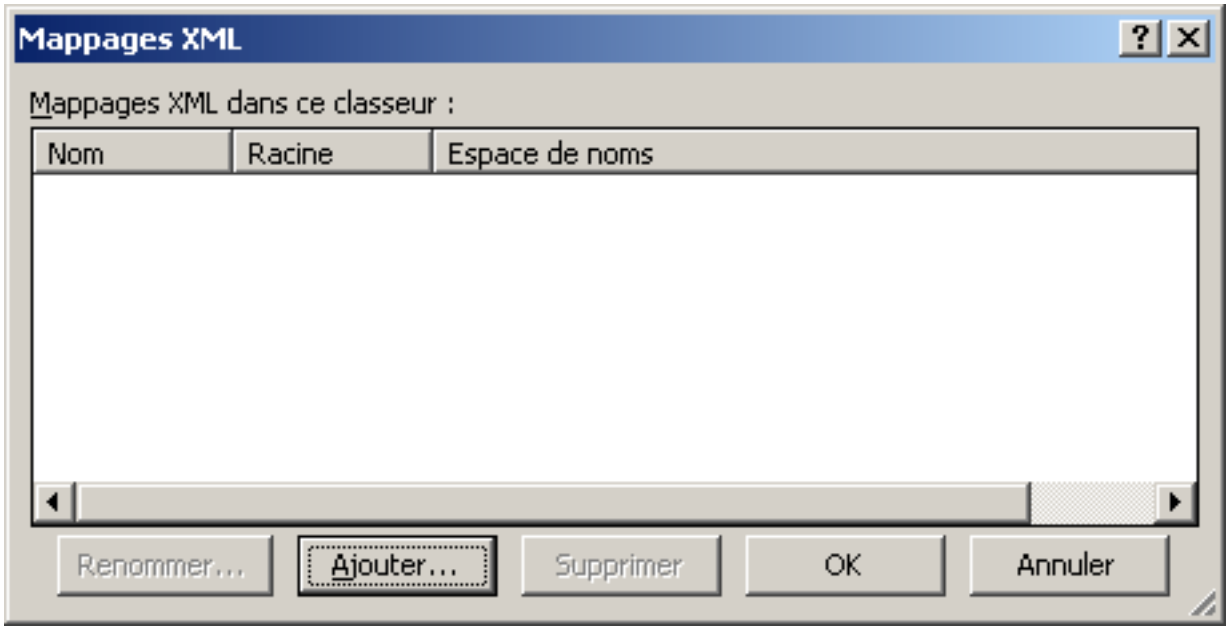
Avant toute chose, il faut créer un nouveau classeur vide. Une fois ce classeur créé, nous allons utiliser l'onglet développeur du ruban. Pour afficher l'onglet développeur, Bouton Office, Options Excel, Standard, Afficher l'onglet Développeur dans le ruban.



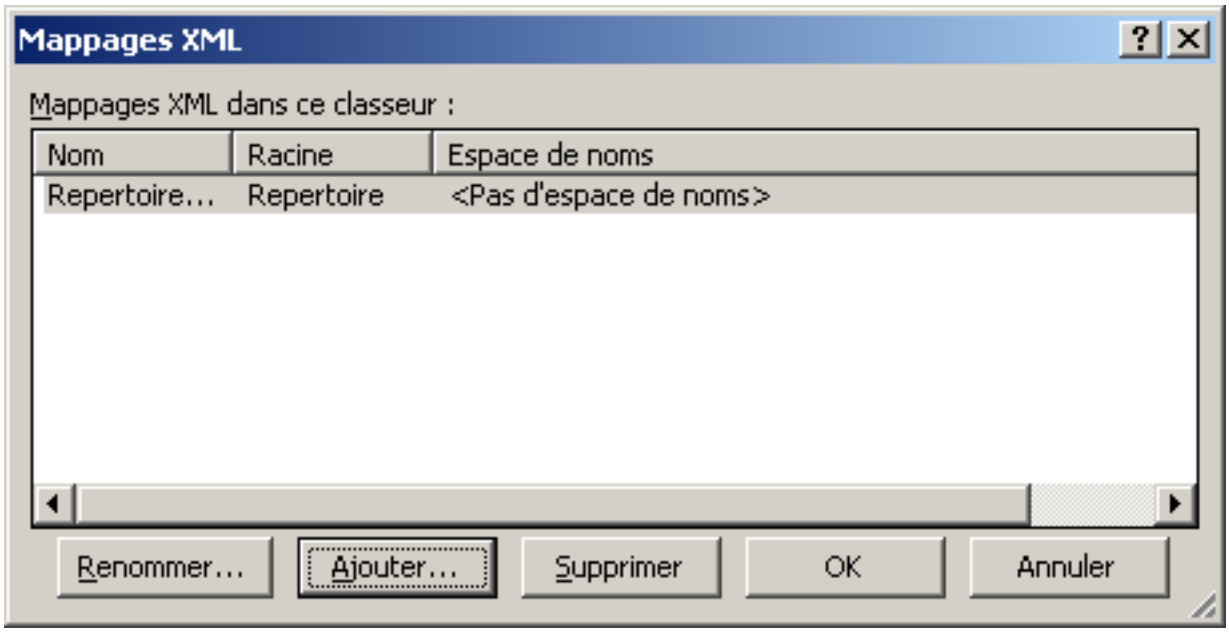
Dans cet onglet, cliquez sur le bouton source. Un volet apparaît. Ce volet sert à la gestion de l'XML dans Excel.



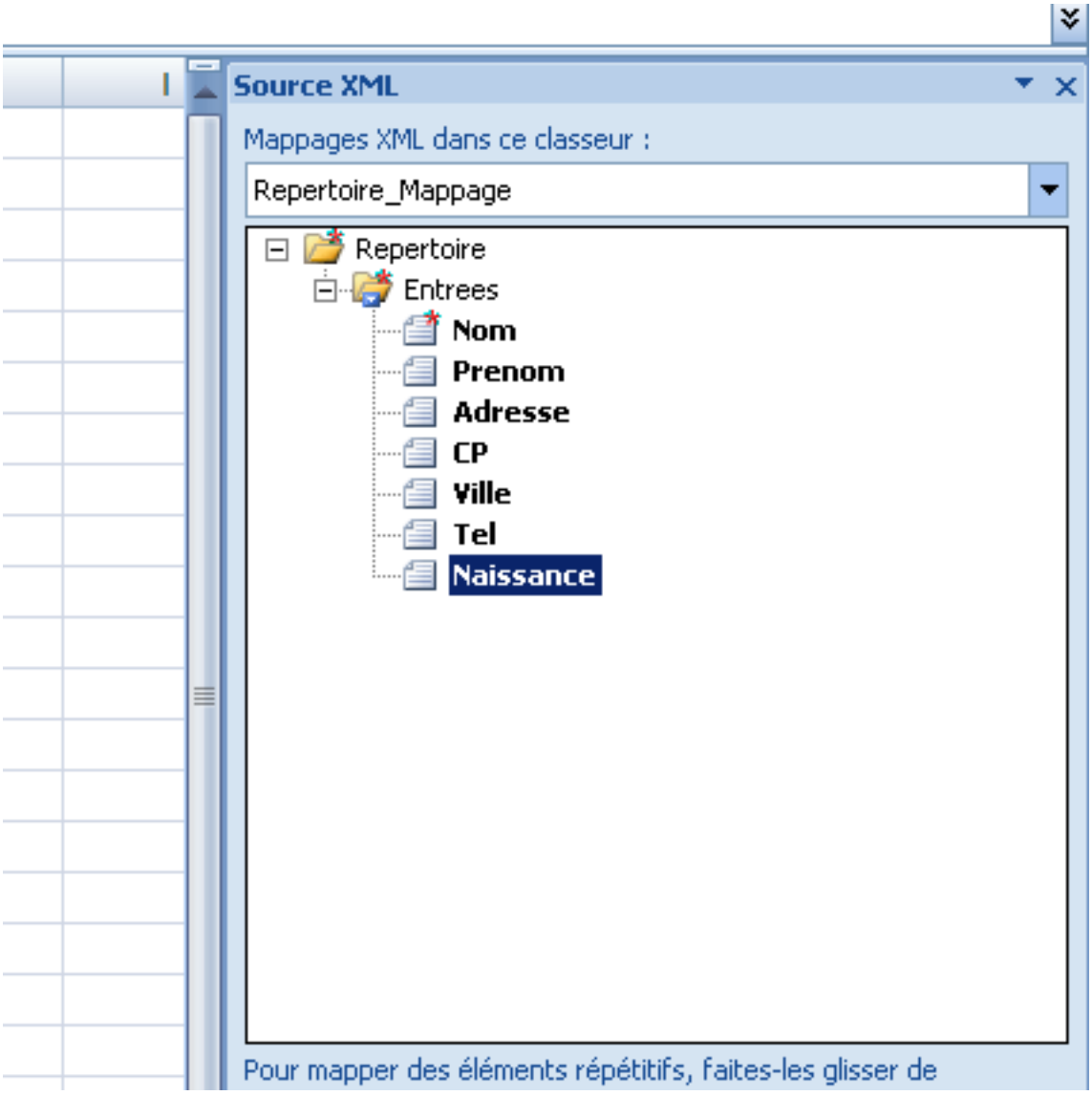
Nous allons ajouter un mappage au classeur.



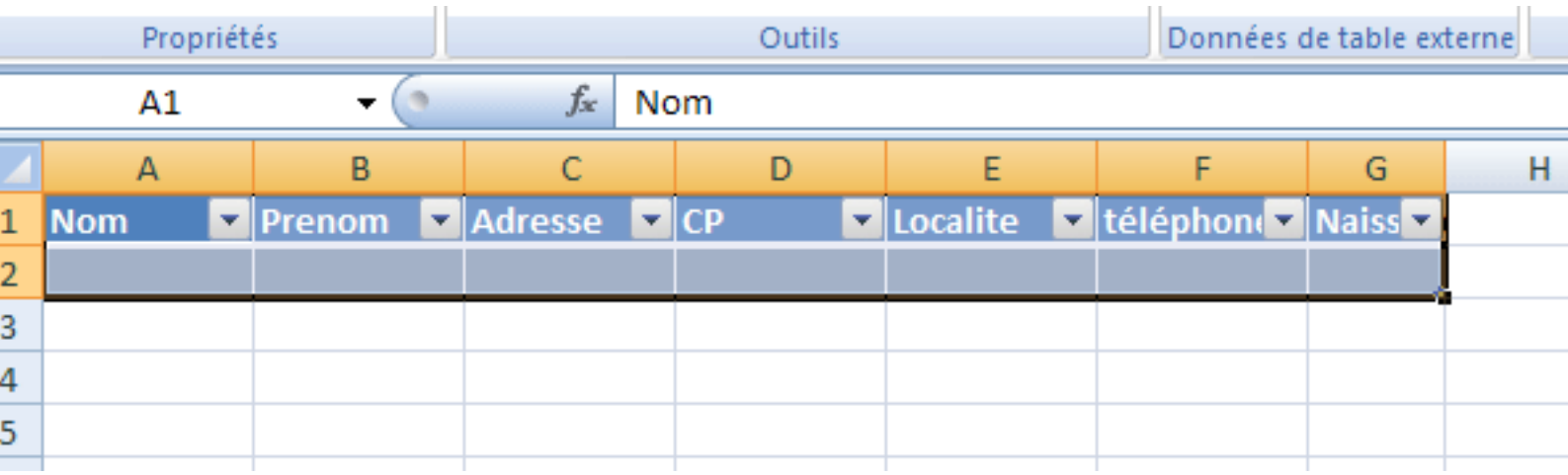
La boîte de dialogue habituelle s'ouvre pour le chemin du fichier XSD. Une fois le fichier XSD sélectionné :



Et le mappage dans le volet XML.



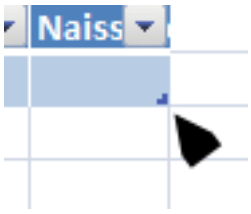
Pour utiliser le mappage dans notre classeur, un simple glisser déposer suffit. Et le résultat est immédiat.



Pour l'édition, difficile de faire plus simple. N'importe qui trouvera le remplissage d'un fichier XML très facile.

Pour ajouter des enregistrements à notre fichier XML, on peut opter pour plusieurs solutions.

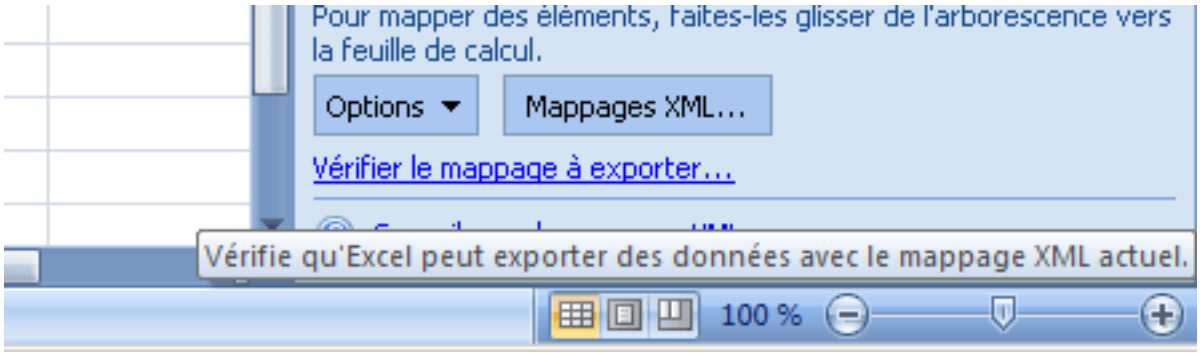
La première, simple est de se positionner sur la ligne suivante et de compléter les colonnes. La suivante est une méthode usuelle avec Excel, c'est de sélectionner le coin inférieur droit d'une cellule avec la souris et de glisser ce coin vers le bas.



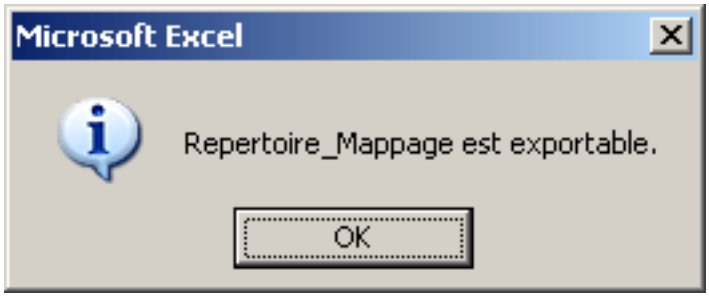
Excel permet l'utilisation pour un même classeur de plusieurs Mappages de données. Ces Mappages sont alors disponibles dans la liste déroulante du volet XML.

Outre le fait que les données peuvent être exportées en XML, la feuille de calcul peut également être sauvegardée comme n'importe quelle autre feuille de calcul Excel.

Avant l'exportation de nos données vers un fichier XML, Excel offre la possibilité de vérifier ces données.



Si les données sont conformes, une boîte de message vous avertira que vous pouvez exporter vos données.



Le fichier XML obtenu :

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Repertoire>
 <Entrees>
  <Nom>Lebeau</Nom>
  <Prenom>Olivier</Prenom>
  <CP>7141</CP>
  <Ville>Carnières</Ville>
```

```
    <Tel>064/44,44,44</Tel>
    <Naissance>1964-10-19</Naissance>
   </Entrees>
   <Entrees>
    <Nom>Marcelle</Nom>
    <Prenom>Chantal</Prenom>
    <Naissance>1964-05-23</Naissance>
   </Entrees>
  </Repertoire>
```

III-C - VBA et le XML

Il est possible en VBA d'intervenir de façon plus pointue sur le XML.

VBA permet certaines actions sur le XML.

Je vais simplement parcourir quelques possibilités offertes par Excel pour intervenir sur le XML.

Ajouter des mappages à la collection XmlMaps. On peut aussi ajouter des fichiers XSD présents sur le net et pas simplement sur la machine.

```
ActiveWorkbook.XmlMaps.Add ("c:\temp\garage.xsd")
```

Obtenir le nom du premier mappage dans le classeur ouvert.

```
ActiveWorkbook.XmlMaps.Item(1).Name
```

Si l'on veut la liste des mappages disponibles dans notre classeur, le code suivant va les lister et donner leur index.

```
Sub ListeMappagesXML()
Dim i As Integer
Dim j As Integer

i = ActiveWorkbook.XmlMaps.Count
j = 1
For j = 1 To i
    Debug.Print j; ActiveWorkbook.XmlMaps.Item(j).Name
Next j
End Sub
```

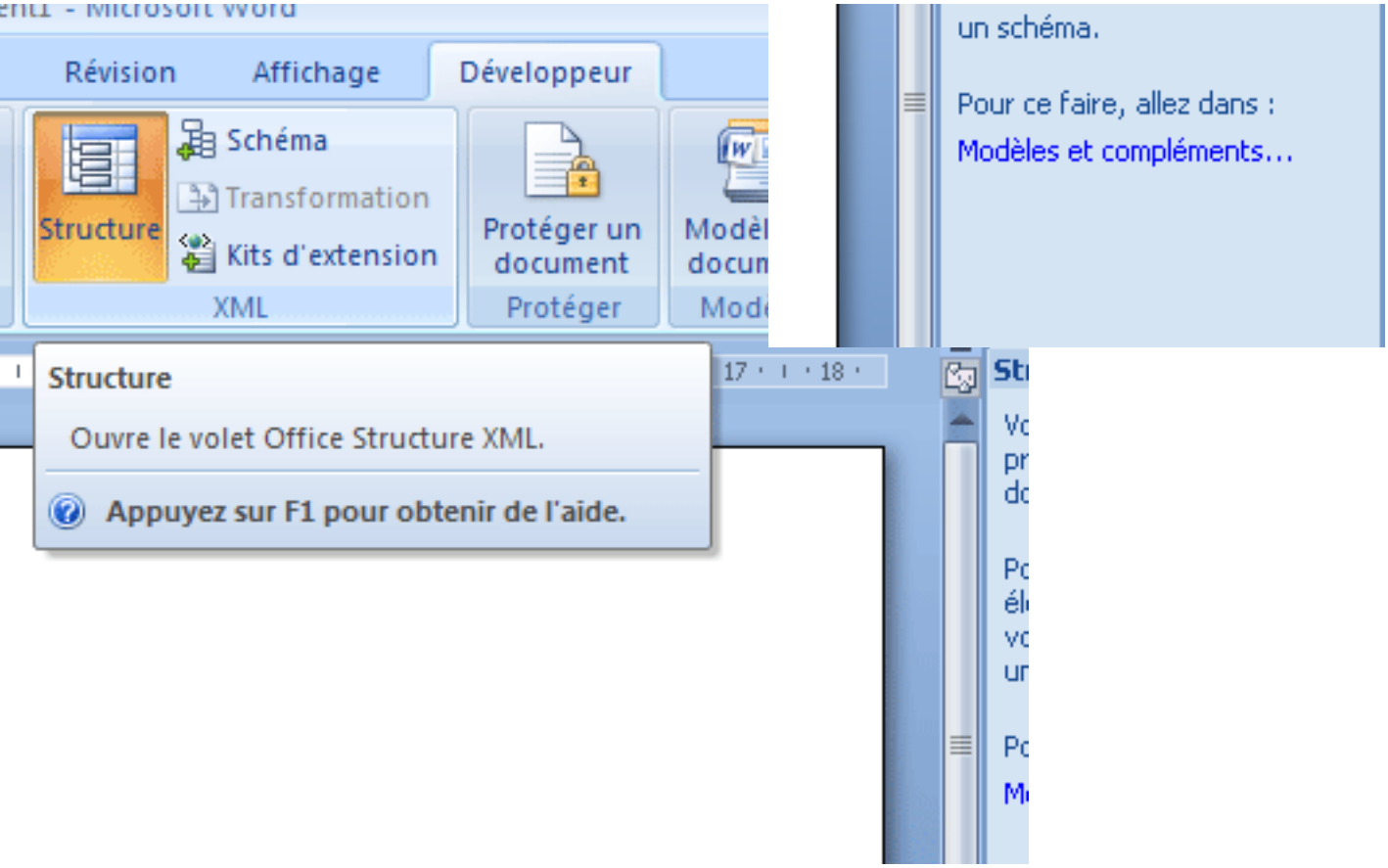
III-D - Le XSL : l'affichage des données.

Si vous appliquez dans Excel un fichier XSL aux données XML, vous aurez un affichage similaire à l'affichage dans un navigateur.

IV - Word et le XML

Word gère le XML d'une façon différente de Excel. Pour avoir accès aux options XML sous Word, il faut aller dans l'onglet développeur du ruban et on y retrouve la même icône que pour Excel.

Pour Word, nous allons utiliser les mêmes fichiers que pour Excel.



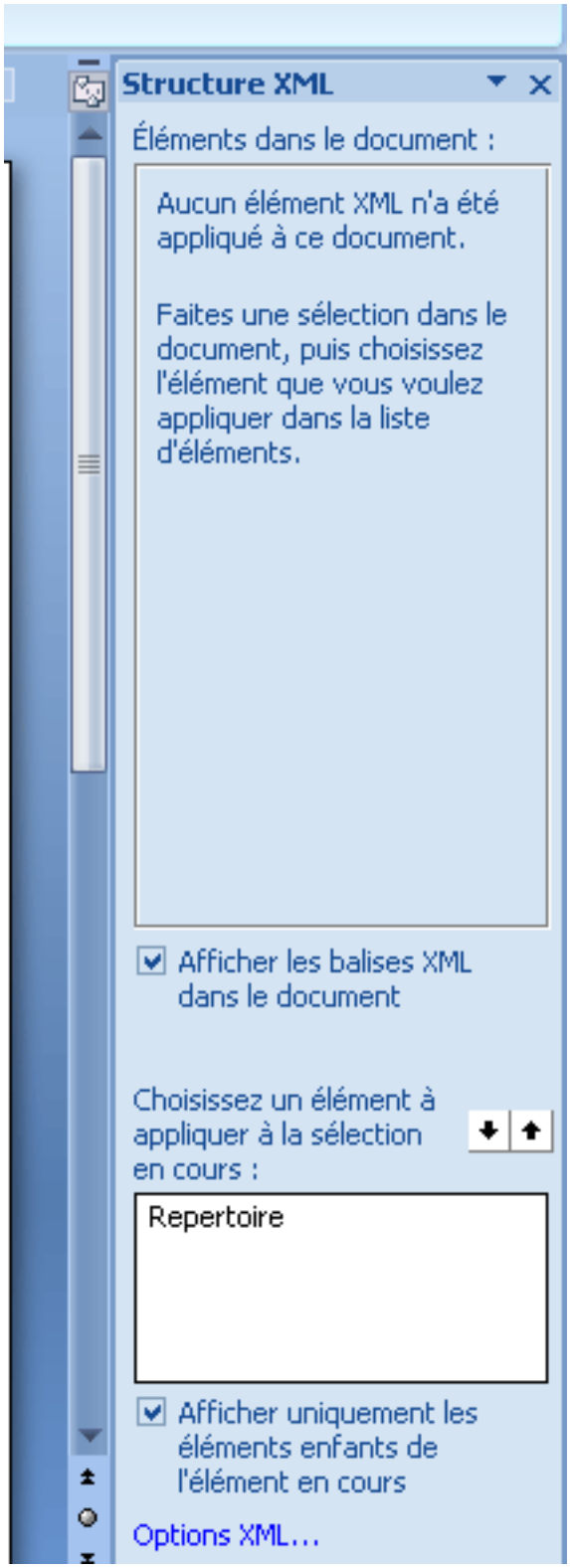
IV-A - Notre document XML

Nous allons utiliser le fichier de notre répertoire pour utiliser le XML sous Word.

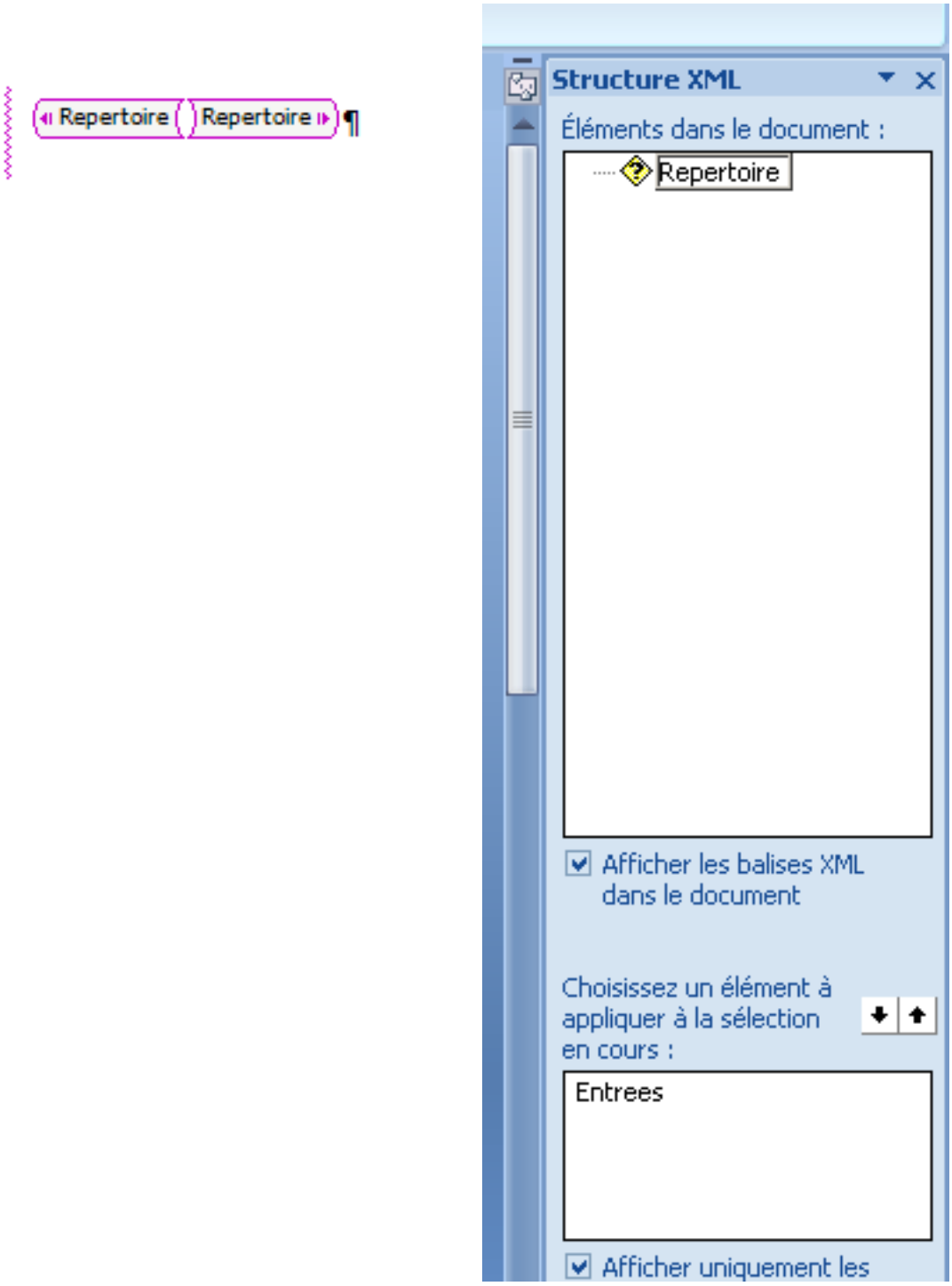
"Ajouter un schéma" nous permet d'ajouter un schéma XSD à notre document, ou plus précisément, à Word, car ce schéma sera disponible dans Word pour tous les documents.

Heureusement, on peut aussi les supprimer.

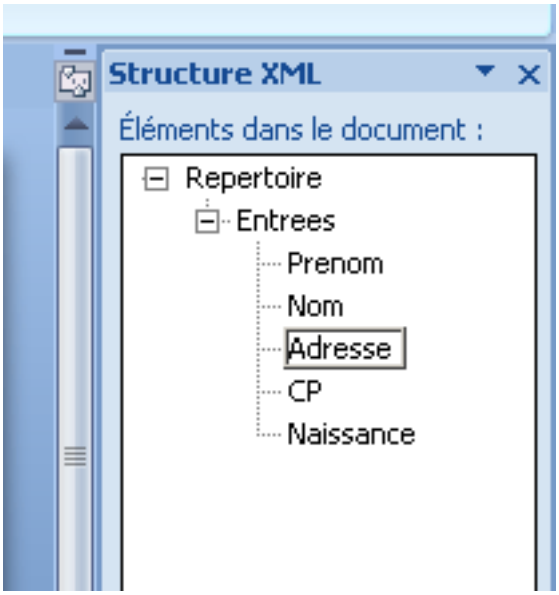
Une fois le schéma ajouté, le volet XML change d'aspect et nous permet d'utiliser notre schéma pour notre document.

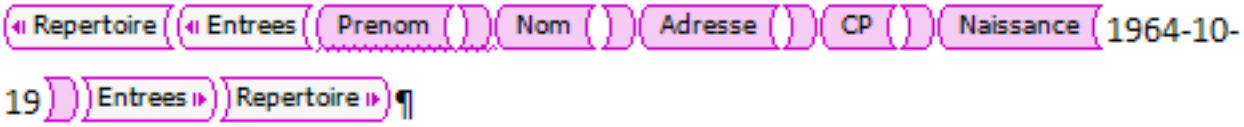


Un clic sur le "mot" répertoire et aussitôt le document change ainsi que le contenu du volet XML.



En positionnant le curseur correctement dans le document et en cliquant sur les éléments XML, on peut générer notre document.





C'est entre les représentations des balises que les données doivent être introduites. On peut répéter les éléments qui composent le document, pour générer une base de données.

Alors que pour Excel un bouton dans l'onglet du développeur permet d'exporter facilement les données, pour Word, il faut aller dans le menu Office et choisir enregistrer sous, Autre format et choisir l'option "Document XML Word 2003 *.XML" et cocher l'option "Enregistrer les données uniquement".

L'ouverture pour contrôle du document obtenu avec XMLSpy permet de découvrir un document conforme à nos attentes.

Nous venons donc de voir qu'il est possible de générer des documents xml avec Word 2007, mais bien moins agréable qu'avec Excel ou Access. Je pense que Word doit être utilisé en dernier recours.

Word possède pourtant une facilité qu'Excel ne peut exploiter, ce sont les fichiers XML complexes.

IV-B - Word VBA et XML

Comment récupérer le "NameSpace" de notre document.

```
Sub recupNameSpaces()
Dim objSchema As XMLNamespace
    For Each objSchema In Application.XMLNamespaces
      Debug.Print objSchema.URI
    Next
End Sub
```

Comment connaître le nombre de "NameSpaces" dans votre document :

```
Sub nbreNameSpaces()
Application.XMLNamespaces.Count
End Sub
```

Avec le code suivant, vous allez pouvoir vérifier lors de l'insertion d'un nouveau node, s'il est conforme au XSD. Si des données complémentaires doivent être introduites, une boîte de message vous avertira.

```
Private Sub Document_XMLAfterInsert(ByVal NewXMLNode As XMLNode, _
        ByVal InUndoRedo As Boolean)

    NewXMLNode.Validate

    If NewXMLNode.ValidationStatus <> wdXMLValidationStatusOK Then
        MsgBox NewXMLNode.ValidationErrorText
    End If
End Sub
```

V - Liens intéressants

XML sur DVP
http://khany.developpez.com/tutoriel/xml/

VI - Conclusion

Avec ces fonctionnalités, Microsoft ouvre le XML à sa célèbre suite Office et tente par là même de séduire les utilisateurs du XML.

Même si l'approche du XML est différente suivant le produit utilisé de la suite, tous sont à même d'utiliser le XML.

Probablement que l'Office Open XML n'est pas étranger à cette évolution.

VII - Remerciements

Merci à toutes les personnes qui m'ont aidé ou ont collaboré à cet article de près ou de loin. Sans elles, je n'y serais jamais parvenu.

Tara Lefave de chez Altova.

Toute l'équipe Office de DVP qui a bien voulu se muter en banc de test.

Toute l'équipe XML de DVP

Et finalement DVP pour le support qu'il nous offre et la qualité de ses services.