

Sommaire

INDEX DES ALGORITHMES.....	vii
INDEX DES FIGURES.....	viii
INTRODUCTION.....	1
PARTIE I : L'approche Xweb.....	3
1 Le modèle structurel.....	5
2 Le modèle médiatique.....	7
2.1 Les types d'unités médiatiques et leur description en XML.....	8
A Les Xobjets.....	8
B Les contextes navigationnels.....	9
B.1 Exemple de contexte simple : les fournisseurs de produits par nom.....	10
B.2 Exemple de contexte composé : les produits vendus par fournisseur.....	10
C Les index.....	11
D Les menus.....	11
E Les liens.....	11
F Les textes.....	12
G Les images.....	12
H Les unités pages.....	12
2.2 Les différentes étapes pour la construction de la base médiatique.....	13
3 Génération de la base médiatique.....	13
PARTIE II : Intégration de Services Web dans Xweb.....	16
I Les services web.....	17
1 Généralités.....	17
2 Etude du fonctionnement des services web.....	17
2.1 Fonctionnement général.....	17
2.2 Technologies utilisées.....	19
2.2.1 SOAP.....	19
2.2.2 WSDL.....	21
2.2.3 UDDI.....	23
3 Implémentation.....	24
3.1 JAX-RPC.....	24
3.2 JAXM.....	26
3.3 .NET.....	27
3.4 BEA Weblogic.....	27
3.5 Apache AXIS.....	28
II Intégration de services web dans Xweb.....	28
1 Unité médiatique de type service web.....	28
2 Extension de l'algorithme de génération d'U.M.....	29
3 Les différentes étapes de la procédure d'intégration de services web dans Xweb.....	30
PARTIE III : Application.....	32
1 Application à des données environnementales.....	33
2 Génération de la base médiatique.....	41
3 Accès à un service web.....	46
3.1 Description d'une U.M. de type service web.....	46

3.2 Déploiement automatique d'une classe java.....	48
3.3 Utilisation d'un service web par un client.....	49
<u>CONCLUSION</u>	<u>53</u>
<u>RÉFÉRENCES</u>	<u>54</u>
<u>ANNEXE</u>	<u>56</u>

Index des algorithmes

Algorithme 1 : Algorithme de génération d'unités médiatiques.....	15
Algorithme 2 : Algorithme de génération d'unités médiatiques et d'intégration de service web dans Xweb.....	30

Index des figures

Figure 1 : Diagramme du schéma entité-association de la base de données de la centrale d'achat Ndakarou khéweul.....	5
Figure 2 : Exemple de graphe XML représentant une partie du document XML centrale_achat.xml.....	7
Figure 3 : Architecture fonctionnelle de Xweb.....	8
Figure 4 : Schématisation d'un exemple de contexte navigationnel simple.....	10
Figure 5 : Schématisation d'un exemple de contexte navigationnel composé.....	11
Figure 6 : Page d'accueil du site web de la centrale d'achat Ndakarou Khéweul.....	12
Figure 7 : Architecture fonctionnelle des services web.....	18
Figure 8 : Structure principale d'un message défini par SOAP.....	19
Figure 9 : Schéma d'invocation d'un service web JAX-RPC.....	25
Figure 10 : Schématisation du système Xweb.....	29
Figure 11 : Page d'accueil du site de la DRDR.....	36
Figure 12 : Production de canne à sucre dans la région de Saint-Louis.....	39
Figure 13 : Proportion des espèces dominantes en pourcentage par département.....	40
Figure 14 : Proportion en pourcentage.....	41
Figure 15 : Interface de compilation de l'application.....	42
Figure 16 : Interface d'entrée d'une U.M. du modèle médiatique.....	42
Figure 17 : Interface d'entrée du catalogue de métadonnées.....	43
Figure 18 : Interface d'entrée d'une U.M. à générer dans la base médiatique.....	43
Figure 19 : Interface d'entrée d'un répertoire représentant une base médiatique.....	44
Figure 20 : Interface d'entrée de la source globale.....	44
Figure 21 : Interface d'information sur le résultat de la génération d'une U.M.....	45
Figure 22 : Interface de la fin de l'exécution de l'application.....	45
Figure 23 : Interface de consultation de la base médiatique.....	46
Figure 24 : L'annuaire de services web.....	47

Figure 25 : Description du service web CalculService.....	48
Figure 26 : Interface de consultation du proxy.....	49
Figure 27 : Invocation de la méthode « add » du service nommé CalculService.....	50
Figure 28 : Resultat de l'exécution de CalculService.....	51
Figure 29 : Interface de compilation d'un client de service web.....	52
Figure 30 : Interface d'exécution du client.....	52

INTRODUCTION :

Le développement rapide du WWW (World Wide Web) et le succès du langage HTML (HyperText Markup Language) ont permis la construction de milliers de sites web générant une quantité importante d'information accessible sur Internet.

Lors de la construction de la plupart de ces sites, l'approche la plus courante consiste à se focaliser beaucoup plus sur l'implantation d'une solution que sur le processus de développement. Ces sites web présentent un ensemble de pages HTML (i) statiques : le contenu ne varie que lorsque l'administrateur du serveur les modifie ou (ii) interactives et dynamiques : le contenu dépend soit des informations localisées sur le serveur (connexion avec une base de données par exemple), soit de paramètres donnés de façon transparente par le navigateur du client.

Quelques outils de développement ont permis d'apporter une aide appréciable dans la génération et la mise en œuvre rapide d'applications web, à l'aide des technologies ASP (Active Server Pages de Microsoft), JSP (Java Server Pages), PHP (Personal Home Page ou Hypertext Preprocessor), PL-SQL (Oracle-Web), etc... Ces technologies permettent d'extraire dynamiquement des informations à partir de diverses sources de données et de les inclure dans des modèles de pages HTML. Dans ces applications, les concepteurs ont souvent privilégié l'aspect présentation au détriment de la structuration des données. C'est lors de l'exploitation de ces sites que cette approche montre ses limites. Les problèmes posés sont souvent dûs à l'augmentation de la taille et de la complexité des sites, au besoin d'une interopérabilité avec d'autres applications, à la nécessité de modifications au cours du temps et au manque évident de possibilités d'interrogation des pages HTML.

La construction de sites web pour une entreprise ou de systèmes d'information basés sur le web contenant quelques centaines de pages interagissant avec différents programmes nécessite une méthodologie de conception adéquate, basée sur un modèle de données assez riche pour permettre la mise en œuvre d'un processus de recherche d'information spécifique liée au domaine d'intérêt de l'utilisateur [19].

Quelques méthodologies de conception ont été proposées pour des applications Web basées sur HTML. Mais les limites imposées par ce langage, notamment durant le processus de recherche d'information, et l'émergence de XML comme format de données a amené Lo et Hocine [19] à proposer l'utilisation de XML pour la construction de sites web importants.

Une des originalités de leur approche, nommée Xweb, réside dans le fait que le contenu du site est entièrement représenté en XML. Ce choix permet d'exploiter les énormes possibilités de représentation et d'interopérabilité offertes par ce langage. Cette approche permet d'une part d'effectuer une séparation nette et distincte entre le contenu du site (données) et la présentation durant le processus de conception du site et, d'autre part, d'exploiter les données du site après sa réalisation. Le contenu du site se trouve alors dans un entrepôt de documents XML appelé dataweb. Un dataweb est défini comme étant une collection de données (i) structurées comme c'est le cas de celles qui sont stockées dans les bases de données relationnelles ou objets, et/ou (ii) semi-structurées comme celles qui sont disponibles sur le web. Il permet généralement d'intégrer des données issues de sources hétérogènes.

La conception d'un dataweb est une tâche complexe nécessitant un effort d'analyse et de conceptualisation important. La méthode de conception et de développement comporte trois étapes qui correspondent à la mise en œuvre des trois modèles suivants :

- Le modèle structurel (ou conceptuel) qui permet d'identifier les différentes sources d'informations du dataweb et de les intégrer en XML. C'est le point de départ de la tâche de conception.
- Le modèle médiatique (ou navigationnel) qui décrit les unités médiatiques et leurs structures navigationnelles. Plusieurs modèles de navigation peuvent être construits à partir d'un modèle structurel.
- Le modèle de présentation qui décrit l'interface utilisateur où chaque unité médiatique est associée à une forme médiatique (i.e. une présentation qui spécifie la forme externe par laquelle les informations correspondantes sont présentées à l'utilisateur).

Cependant, l'approche Xweb permet une génération automatique de sites web basés sur XML mais n'offre pas l'accès à des applications.

Les services web reprennent la plupart des idées et des principes du web, et les appliquent à des interactions entre ordinateurs. Comme pour le World Wide Web, les services web communiquent via un ensemble de protocoles fondamentaux qui partagent une architecture commune. Ils ont été conçus pour être réalisés sur de nombreux systèmes développés et déployés de façon indépendante. Comme pour le World Wide Web, les protocoles des services web doivent beaucoup à l'héritage d'Internet, fondé sur le texte. Ils sont conçus pour respecter une structure en couches sans être dépendants de façon excessive de la pile des protocoles[22].

Mais les services web diffèrent du World Wide Web dans leur portée. HTTP et HTML ont été définis pour permettre une exploration, essentiellement en mode lecture, d'un contenu souvent statique ou pouvant être mis en cache. À l'opposé, les services web permettent des interactions fortement dynamiques entre programmes. De nombreux types de systèmes répartis peuvent être implémentés dans l'architecture de services web: par exemple, des fermes de serveurs de calculs, des systèmes mobiles connectés en réseau. Les très nombreux impératifs existant dans les interactions de programme à programme imposent à la pile des protocoles des services web d'être conçue pour un usage plus général que les premiers protocoles web. Cependant, comme pour le World Wide Web, les services web reposent sur un petit nombre de protocoles spécifiques [22].

L'objectif de ce stage de DEA est l'implémentation de l'approche Xweb dans un environnement Java, son application sur un cas réel et l'intégration de services web dans Xweb.

La première partie s'intéresse à la présentation du modèle de conception d'applications web dénommé Xweb proposé par Lo et Hocine[19]. L'étude des services web et leur intégration dans Xweb feront l'objet de la deuxième partie. Enfin la troisième et dernière partie s'intéresse à un cas pratique d'utilisation de Xweb.

PARTIE I: L' approche Xweb

Dans cette partie nous étudions un modèle de conception d'applications Web basées sur XML (Xweb) proposé par Lo et Hocine [19].

Plusieurs travaux ont été menés dans le but de proposer des méthodologies et des modèles de conception d'applications Web. On peut distinguer les méthodologies de conception d'applications hypermédia et les systèmes de gestion de sites Web dynamiques.

Les premiers travaux réalisés dans le domaine avaient pour but de résoudre les problèmes liés aux systèmes hypermédias. Il s'agissait de trouver des méthodologies adaptées à ce type d'applications qui ont des spécificités par rapport aux applications traditionnelles. Toutes ces méthodologies sont fondées sur la séparation entre l'analyse du domaine, la spécification de l'espace navigationnel et la conception de l'interface utilisateur.

Les solutions proposées dans ces travaux peuvent être adaptées au contexte du Web et ont inspiré la plupart des travaux menés pour la conception d'applications Web. En plus des problèmes liés au contexte hypermédia, d'autres problèmes liés à la spécificité du Web se posent au concepteur d'application Web : intégration de données de sources diverses, interopérabilité, nature dynamique du Web, nécessité de couplage avec des SGBD, etc. C'est dans ce cadre que plusieurs systèmes ont vu le jour. On peut citer Strudel[7], Araneus[11], WebML[4].

Tous ces systèmes (Strudel, Araneus, WebML) sont basés sur HTML et s'appuient sur la notion de médiateur. Un médiateur est un module logiciel qui exploite de la connaissance codée sur quelques ensembles et sous ensembles de données afin de créer une information pour une couche applicative de niveau supérieur.

Les systèmes de médiation traitent de l'intégration de données issues de sources hétérogènes. Ils permettent l'exploitation de plusieurs sources d'information à partir d'une vue intégrée de données.

Cependant, ils utilisent une approche virtuelle et ne sont pas très adaptés à certains types d'application comme les systèmes d'information environnementaux qui présentent la spécificité de ne pas nécessiter une mise à jour régulière de leurs données. En effet, la réalisation de ce type d'applications nécessite plus des solutions permettant la construction de bases d'information exploitables à travers le Web que de solutions de médiation.

Dans **l'approche virtuelle**, les données sont laissées dans leurs sources respectives sans être dupliquées et les requêtes sont décomposées et appliquées en temps réel sur les sources. Cette forme d'intégration ne facilite :

- ni la mise en forme des données intégrées pour leur diffusion sur le Web,
- ni l'ajout d'un module de recherche d'informations pertinentes.

En plus, la définition du schéma intégré et sa liaison avec les schémas locaux sont des difficultés liées à cette approche.

Par ailleurs, les avantages offerts par le langage XML comme format de représentation et d'échange de données sur le Web font de ce standard un outil incontournable pour la réalisation de systèmes d'information pour le Web. Il faut dire que cela offre de nombreux avantages[20]. D'abord, XML permet une représentation naturelle de données provenant de n'importe quelle source. Il offre la possibilité de développer des adaptateurs permettant de transformer toute donnée source ayant un format quelconque en XML. Ensuite, XML facilite l'interopérabilité entre les différents composants d'un système d'intégration en offrant une syntaxe de données standard.

Dans cette section nous nous focalisons sur la présentation des deux modèles principaux (structurel et médiatique), autour desquels s'articule la méthode de conception et de

développement d'applications Web. Cette méthode s'appuie sur les nouveaux standards XML (eXtensible Markup Language) [2] et sur la notion de dataweb, un dataweb étant une collection de données (i) structurées comme c'est le cas de celles qui sont stockées dans les bases de données relationnelles ou objets, ou (ii) semi-structurées comme celles qui sont disponibles sur le web [9].

1.Le modèle structurel

La démarche de conception du modèle structurel qu'ils proposent s'effectue en plusieurs étapes :

- Identifier les sources de données (structurées et semi-structurées) du dataweb,
- Utiliser le modèle entité-association (E-A) pour modéliser chaque source de données structurées, et XML pour chaque source de données semi-structurées [1],
- Traduire les modèles E-A obtenus à l'étape précédente en schémas relationnels,
- Obtenir pour chaque source, grâce à une technique de mapping (relationnel – XML) qui est détaillée dans [9], un schéma structurel représentant la structure logique de la base de données relationnelle en XML.
- Insérer dans une base de documents XML les données issues des bases de données relationnelles.

L'ensemble des schémas structurels issus des différentes sources de données structurées du dataweb, et les schémas des documents XML issus des sources de données semi-structurées, forment le modèle structurel du dataweb.

Pour illustrer simplement l'approche, nous avons utilisé Xweb pour réaliser le site web d'une centrale d'achat. Nous avons considéré deux sources de données :

- les informations sur les fournisseurs et les produits de la centrale d'achat Ndakarou khéweul, modélisées à l'aide du modèle E/A (données structurées) ;
- les informations relatives aux services web décrites dans un document texte, formalisées dans un document XML (données semi-structurées). Rappelons que les services web ne sont pas pris en compte dans l'approche proposée par Lo et Hocine.

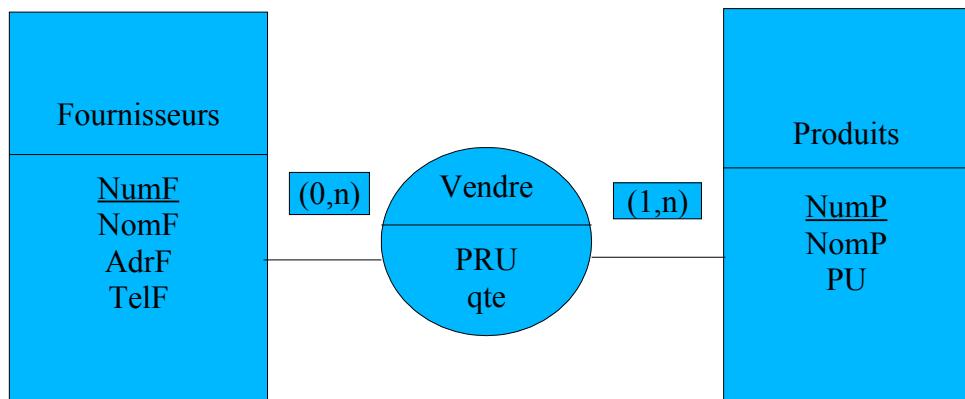


Figure 1 : Diagramme du schéma entité-association de la base de données de la centrale d'achat Ndakarou khéweul

Le modèle structurel du dataweb est donné ci-dessous :

```

<bd_centrale_achat>
  <fournisseurs>
    <fournisseur id="">
      <nom/>
      <adresse/>
      <telephone/>
    </fournisseur>
  </fournisseurs>
  <produits>
    <produit id="">
      <nom/>
      <poids_unitaire/>
    </produit>
  </produits>
  <ventes>
    <vente fournisseur="" produit="">
      <prix_unitaire/>
      <quantite/>
    </vente>
  </ventes>
  <services_web>...</services_web>
</bd_centrale_achat>

```

Le problème de la transformation (ou mapping) des données issues de bases de données relationnelles en documents XML a donné lieu à quelques outils dont DB2XML [15], PLSXML [12] et XML-DBMS [3]. Ces trois systèmes sont des packages Java et ont adopté la même approche : ils utilisent une technique de mapping très simple basée sur le DOM (Document Object Model) [18]. Le problème des clés étrangères n'est pas abordé, ce qui implique une redondance d'informations dans le document obtenu. La structure de ce document est une structure d'arbre et non de graphe orienté comme celle que nous avons proposée (Fig. 2).

XML autorise l'utilisation des types spéciaux d'attributs pour traduire la notion de graphe : un élément peut avoir un attribut de type ID dont la valeur fournit un identificateur unique qui peut être référencé par des attributs de type IDREF ou IDREFS à partir d'autres éléments. Notre modèle de données supporte les attributs IDREF(s) et permet donc de représenter les données XML par un graphe orienté et étiqueté : les attributs IDREF(s) implémentent les associations du modèle E/A.

Exemple de données XML :

Nous utilisons un extrait simplifié de la base de documents XML obtenue après l'opération de mapping :

```

<bd_centrale_achat>
  <fournisseurs>
    <fournisseur id="BL">
      <nom>SBMA</nom>
      <adresse>Lot N4 EST SODIDA
      DAKAR</adresse>
      <telephone>849-56-06</telephone>
    </fournisseur>
    <fournisseur id="SAT">
      <nom>SATREC</nom>
      <adresse>DAKAR-SENEGAL BP 22130</adresse>
      <telephone>828-14-33</telephone>
    </fournisseur>
    <fournisseur id="KIR">
      <nom>KIRENE</nom>
      <adresse>BP 7020 DAKAR-SENEGAL</adresse>
      <telephone>956-22-97</telephone>
    </fournisseur>
    <fournisseur id="FONT">
      <nom>FONTAINE</nom>
      <adresse>DAKAR-SENEGAL</adresse>
      <telephone>828-45-96</telephone>
    </fournisseur>
    <fournisseur id="COL">
      <nom>COLGATE PALMOLIVE</nom>
      <adresse>DAKAR-SENEGAL</adresse>
      <telephone>8825-15-16</telephone>
    </fournisseur>
    <fournisseur id="MED">
      <nom>MEDIANE</nom>
      <adresse>Dakar-SENEGAL</adresse>
      <telephone>825-18-26</telephone>
    </fournisseur>
    <fournisseur id="CRO">
      <nom>CROCO</nom>
      <adresse>RUFISQUE</adresse>
      <telephone>848-28-16</telephone>
    </fournisseur>
  </fournisseurs>
  <produits>

```

```

<produit id="EKYR">
  <nom>Eau Minerale KIRENE</nom>
  <poids_unitaire>330g</poids_unitaire>
</produit>
<produit id="EFONT">
  <nom>Eau Minerale FONTAINE</nom>
  <poids_unitaire>500g</poids_unitaire>
</produit>
<produit id="TCOL">
  <nom>Pate dentifrice COLGATE</nom>
  <poids_unitaire>120g</poids_unitaire>
</produit>
<produit id="TMED">
  <nom>Pate dentifrice MEDIANE</nom>
  <poids_unitaire>120g</poids_unitaire>
</produit>

```

.....

```

<produit id="SCRO">
  <nom>Savon CROCO</nom>
  <poids_unitaire>900g</poids_unitaire>
</produit>
<produit id="LSAT">
  <nom>Lait en Poudre (VITALAIT)</nom>
  <poids_unitaire>22.5g</poids_unitaire>
</produit>
<produit id="LBL">
  <nom>Lait en Poudre (BARALAIT)</nom>
  <poids_unitaire>22.5g</poids_unitaire>
</produit>
</produits>
...
</bd_centrale_achat>
```

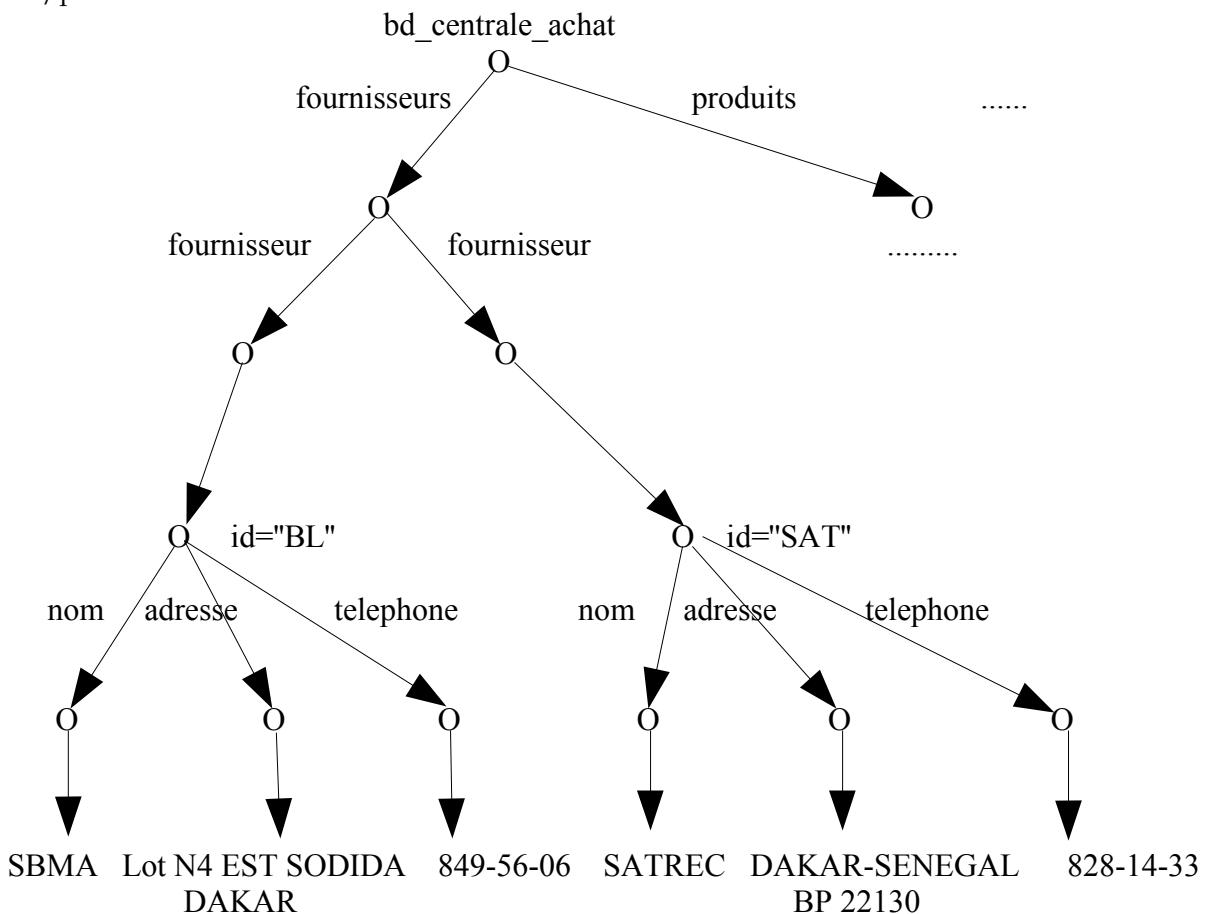


Figure2: Exemple de graphe XML représentant une partie du document XML ci-dessus

2.Le modèle médiatique

Le modèle médiatique est une vue du modèle structurel. Il décrit les unités médiatiques du dataweb et leur structure navigationnelle.

Une unité médiatique (U.M.) est une «unité informative qui a une certaine autonomie du point de vue d'un utilisateur (i.e. qui a un sens en soi et donc présente une idée ou un concept cohérent) et qui mérite d'être sollicitée dans plusieurs démarches de consultation » [5].

Dans le dataweb, une U.M. est représentée en XML et constitue le contenu d'une page Web. L'ensemble des unités médiatiques et leurs liens navigationnels (ie une **base médiatique**) est obtenu à partir d'un algorithme de génération automatique s'appuyant sur le modèle médiatique. Ce modèle, décrit à l'aide de balises XML spécifiques, est étudié dans cette section.

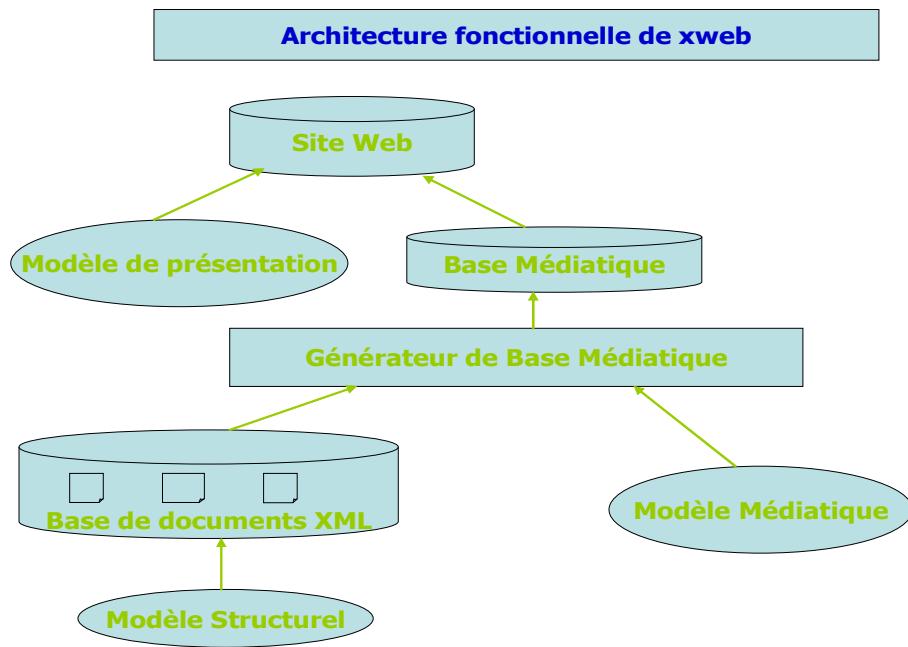


Figure3: Architecture fonctionnelle de Xweb

2.1Les types d'unités médiatiques et leur description en XML

Lo et Hocine[19] proposent huit types d'unités médiatiques : Xobjets, contextes navigationnels, index, menus, liens, textes, images et pages web.

A - Les Xobjets : un Xobjet est un extrait de la base de documents XML, obtenu à partir de l'application d'un filtre.

Un filtre peut être engendré automatiquement à partir d'une vue définie en XML. A titre d'exemple, voici une vue permettant d'obtenir des informations sur les produits des fournisseurs de la centrale d'achat Ndakarou khéweul :

```

<um id="u2" type="Xobjet">
    <vue_filtre id="f_produits_xobjet" type="produit" definition="bd_centrale_achat/produits/produit">
        <nom/>
        <poids_unitaire/>
    </vue_filtre>
</um>
    
```

Les éléments prédéfinis de l'objet vue_filtre sont : <vue_filtre>, <definition>. Ils permettent d'engendrer automatiquement une requête (appelée filtre) écrite en XSLT (eXtensible Style Language Transformation) [17], XSLT étant un langage permettant de transformer des documents XML en d'autres documents XML.

Nous donnons ci-dessous le code XSLT obtenu à partir de l'objet vue_filtre :

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
  <xsl:output method="xml" indent="yes"/>
  <xsl:template match="/">
    <Xobjet>
      <xsl:for-each select="bd_centrale_achat/produits/produit ">
        <produit>
          <nom><xsl:apply-templates select="nom"/></nom>
          <poids_unitaire><xsl:apply-templates select="poids_unitaire"/></poids_unitaire>
        </produit>
      </xsl:for-each>
    </Xobjet>
  </xsl:template>
</xsl:stylesheet >
```

L'application de ce filtre sur la base de documents XML donne le résultat suivant :

```
<xobjet>
  <produit>
    <nom>Eau Minerale KIRENE</nom>
    <poids_unitaire>330g</poids_unitaire>
  </produit>
  <produit>
    <nom>Eau Minerale FONTAINE</nom>
    <poids_unitaire>500g</poids_unitaire>
  </produit>
  <produit>
    <nom>Pate dentifrice COLGATE</nom>
    <poids_unitaire>120g</poids_unitaire>
  </produit>
  <produit>
    <nom>Pate dentifrice MEDIANE</nom>
  </produit>
  <poids_unitaire>120g</poids_unitaire>
</produit>
<produit>
  <nom>Savon CROCO</nom>
  <poids_unitaire>900g</poids_unitaire>
</produit>
<produit>
  <nom>Lait en Poudre (VITALAIT)</nom>
  <poids_unitaire>22.5g</poids_unitaire>
</produit>
<produit>
  <nom>Lait en Poudre (BARALAIT)</nom>
  <poids_unitaire>22.5g</poids_unitaire>
</produit>
</xobjet>
```

B - Les contextes navigationnels : un contexte navigationnel est un ensemble de Xobjets accessible à partir d'un index.

On distingue deux types de contexte navigationnels :

- Les *contextes simples* dont le filtre, permettant d'obtenir les xobjets, fait intervenir une seule entité du modèle E/A ;
- Les *contextes composés* dont les filtres font intervenir plusieurs entités (donc au moins une association) du modèle E/A.

B1 – Exemple de contexte simple : les fournisseurs de produits par nom.

Un tel contexte permet d'obtenir plusieurs Xobjets représentant chacun la page d'un fournisseur de produits et un index donnant accès à chacun de ces Xobjets. On le décrit comme suit :

```
<um id="fournisseurs" type="contexte_simple">
  <vue_filtre id="f_four_cont" type="fournisseur"
definition="bd_centrale_achat/fournisseurs/fournisseur">
  <info_index>
    <index>nom</index>
    <titre>fournisseurs</titre>
  </info_index>
  <nom />
  <adresse />
  <telephone />
</vue_filtre>
</um>
```

Les éléments prédéfinis de l'objet vue_filtre sont : <vue_filtre>, <definition>, <info_index> et <index>. Cette vue est ensuite traduite en un filtre XSLT dont l'application permet d'obtenir les divers Xobjets (ie les fournisseurs) et de construire l'index. On obtient alors le résultat suivant :

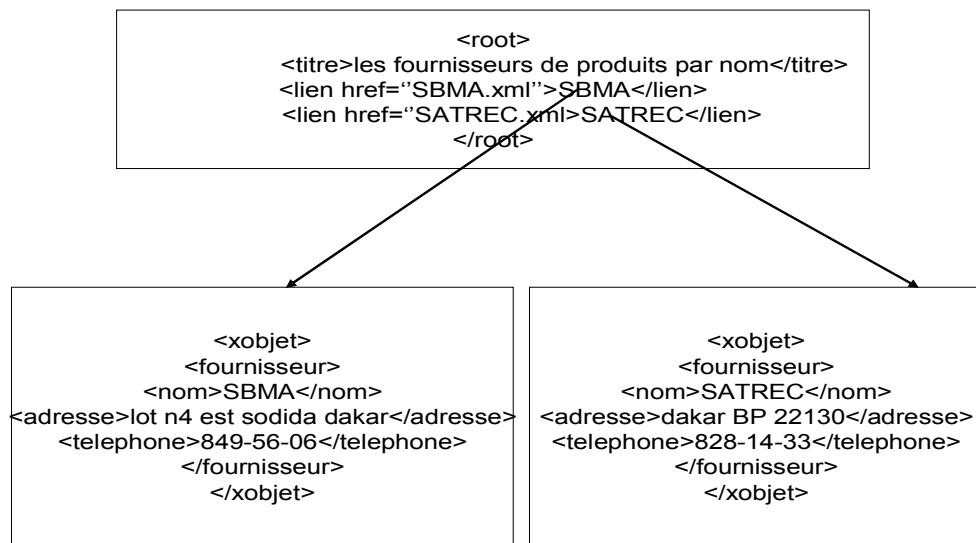


Figure4 : Schématisation d'un exemple de contexte navigationnel simple

B2 – Exemple de contexte composé : les produits vendus par fournisseur.

Contrairement aux contextes simples, les contextes composés font intervenir plusieurs entités et donc au moins une association du modèle E/A. Dans notre première approche, seules deux entités et une association sont prises en compte.

L'exemple fait intervenir l'association "Vendre" du modèle E/A (Fig.1) reliant les entités "Fournisseurs" et "Produits", donc les balises <fournisseurs> et <produits> du modèle structurel. Deux filtres sont alors utilisés :

- Un premier filtre, appliqué à la balise <fournisseurs>, permet d'obtenir la liste des

fournisseurs (id, nom, adresse, telephone). L'identifiant (id), qu'on peut avantageusement remplacer par l'attribut nom, détermine l'index principal.

- Un deuxième filtre, appliqué à la balise <produits> conditionnellement à chaque valeur de l'index principal (ie conditionnellement à chaque fournisseur), permet d'obtenir les produits vendus par fournisseur (les Xobjets représentés ci-dessous) et un index secondaire (les produits de ce fournisseur).

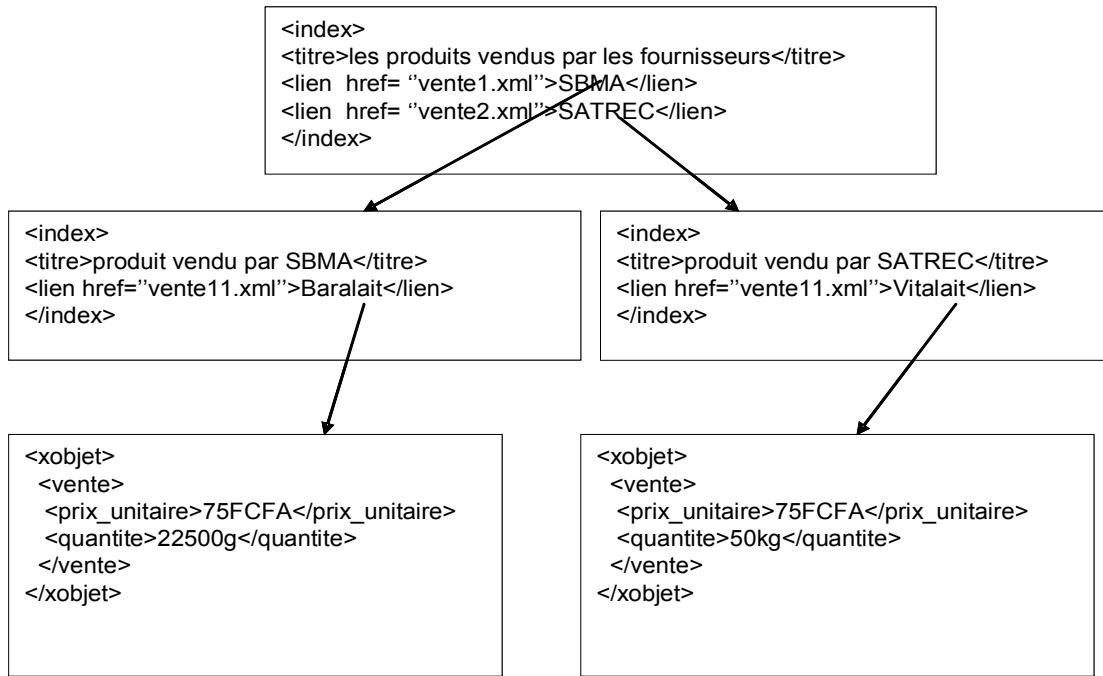


Figure5 : Schématisation d'un exemple de contexte navigationnel composé

C - Les index : ce sont des unités médiatiques composées de liens menant vers des Xobjets appartenant à une thématique donnée. Un index dépend d'un contexte navigationnel.

D - Les menus : un menu est un ensemble de liens (référentiels) vers d'autres unités médiatiques.

```

<um id="menu_gen" type="menu" >
    <lien id=".\\resultats\\produits.xml">Produits</lien>
    <lien id=".\\resultats\\fournisseurs.xml">Fournisseurs</lien>
    <lien id=".\\resultats\\uddi.xml">L'annuaire de services web</lien>
</um>
  
```

E - Les liens : ces U.M. permettent de décrire les liens hypertextes dans le dataweb. On distingue trois sortes de liens :

E1 - Les liens structurels : ils décrivent la structure navigationnelle des U.M. A partir d'une unité médiatique, un lien structurel permet d'aller vers les unités connexes dans la structure navigationnelle (unités suivante, précédente, page d'accueil, etc...).

E2 - Les liens contextuels : ils proviennent du modèle structurel et permettent de connecter des unités médiatiques ayant des liens sémantiques, par exemple les liens se trouvant dans les index.

E3 - Les liens référentiels : ils permettent d'aller à n'importe quelle U.M., et même à des URL extérieures.

Notons que seuls les liens référentiels sont décrits dans les U.M. ; les autres types de liens sont générés automatiquement à partir de la structure navigationnelle et du modèle structurel.

F - Les textes : une U.M. de type texte est un texte libre ajouté par le concepteur.

<texte>Ce travail est réalisé dans le LANI</texte>

G - Les images : une U.M. de type image permet d'insérer une image à partir d'un fichier.

<image>ndakarou_khéweul.gif</image>

H - Les unités pages : ce sont des unités englobantes, décrites en XML, qui seront associées à des feuilles de style pour générer les pages web du site.

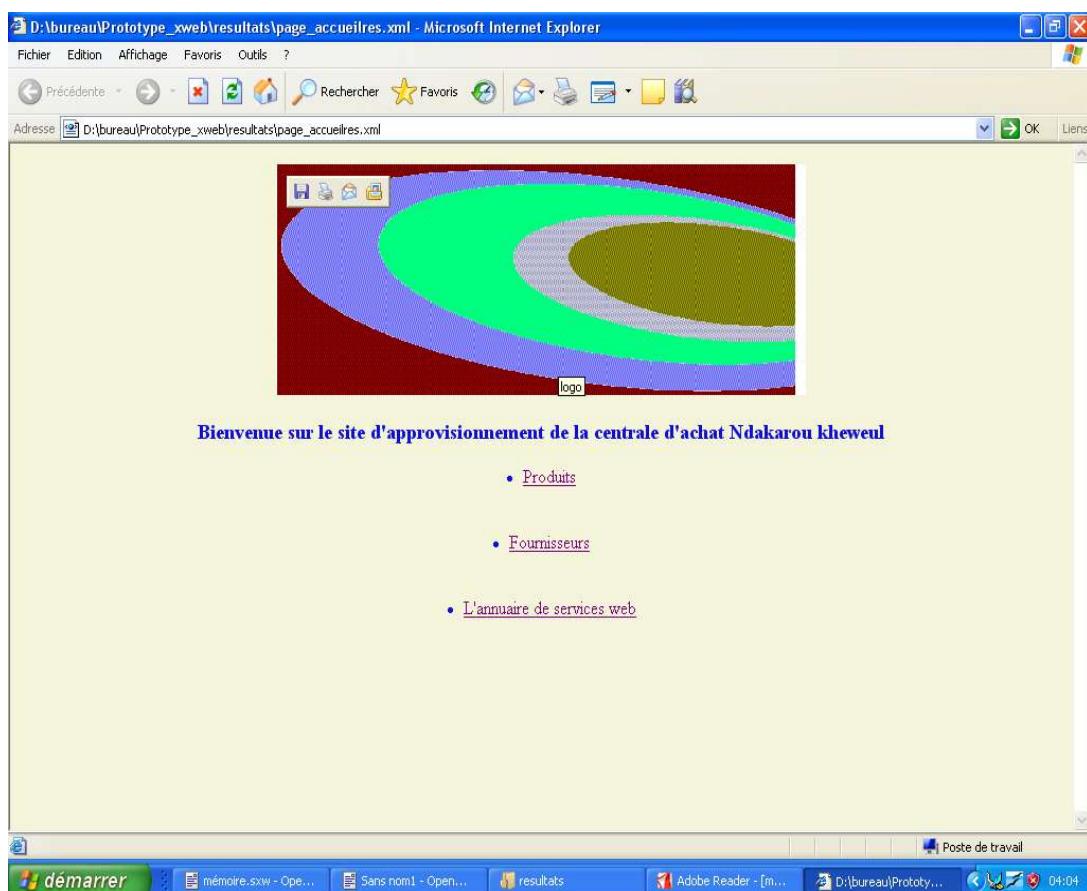


Figure 6 : Page d'accueil du site web de la centrale d'achat Ndakarou Khéweul

La figure 6 montre la page d'accueil de la centrale d'achat Ndakarou Khéweul. Cette page résulte de l'application d'une forme médiatique (ici une feuille de style) à l'unité médiatique suivante :

```

<racine>
  <image href=".\\donnees1\\ndakarou_kheweul.gif"/>
  <titre>Bienvenue sur le site d'approvisionnement de la centrale d'achat Ndakarou kheweul</titre>
  <lien href=".\\resultats\\produits.xml">Produits</lien>
  <lien href=".\\resultats\\fournisseurs.xml">Fournisseurs</lien>
  <lien href=".\\resultats\\uddi.xml">L'annuaire de services web</lien>
</racine>

```

Cette unité médiatique, de type "unité page", a été engendrée à partir de la description suivante :

```

<um id="page_accueil" presentation="menu.xsl">
  <image>ndakarou kheweul</image>
  <titre>Bienvenue sur le site d'approvisionnement de la centrale d'achat Ndakarou kheweul</titre>
  <lien id=".\\resultats\\produits.xml">Produits</lien>
  <lien id=".\\resultats\\fournisseurs.xml">Fournisseurs</lien>
  <lien id=".\\resultats\\uddi.xml">L'annuaire de services web</lien>
</um>

```

2.2 Les différentes étapes pour la construction de la base médiatique

Pour construire une base médiatique, Lo et Hocine proposent les étapes suivantes :

- a) Déterminer les "unités pages" (les nommer et décrire leurs fonctions) ;
- b) Déterminer les liens navigationnels entre unités pages ;
- c) Décrire formellement en XML les unités médiatiques à construire ;
- d) Engendrer la base médiatique à partir de la description des U.M. et de la base de documents XML

3.Génération de la base médiatique :

La génération de la base médiatique résulte d'un processus relativement complexe (notamment pour les unités médiatiques de type « contexte composé »).

L'algorithme de génération reçoit en entrée trois types d'informations :

- La base de documents XML (Fig 2).
- La description des unités médiatiques en XML (section 2.1), c'est à dire la description de chaque unité médiatique de type « unité page».
- Un catalogue de métadonnées permettant notamment d'établir une correspondance entre les noms logiques des ressources (images, URL de destination des liens hypertextes, filtres...) et leurs noms physiques : on peut par exemple préciser que l'U.M. correspondant au logo de la centrale d'achat, nommée « ndakarou kheweul » (voir paragraphe G, section 2.1), correspond au fichier « ndakarou_kheweul.gif ». Ce catalogue est lui aussi écrit en XML.

Le résultat de l'algorithme est une collection d'unités médiatiques, essentiellement des unités médiatiques de type « Xobjet » et « unités pages ». Pour construire les pages Web du site à partir de la base médiatique, il reste à associer une forme médiatique à chaque unité médiatique. Cela peut se faire en utilisant des feuilles de style (CSS ou XSL) ou même en manipulant directement les documents XML contenant les U.M. à l'aide d'un éditeur WYSIWYG.

Dans la suite de cette section, nous étudions l'algorithme de génération d'une page puisque l'algorithme général en découle directement. Cet algorithme s'appuie sur la structure arborescente de tout document XML [18], en l'occurrence sur la structure arborescente d'une unité médiatique de type "unité page".

Procédure Générer_Page (Racine : Noeud)

Début

Ouvrir en écriture le fichier (nommé F) dans lequel sera créé l'UM de type «unité page»

/* le nom physique est déterminé à partir de la balise id et du catalogue */

Pour chaque noeud fils (nommé nd_fils) de la racine de l'arbre Faire

 Traiter_Noeud (nd_fils, F)

Fin Pour

Fin

Procédure Traiter_Noeud (nd : Noeud, F : Fichier)

Debut

Selon le type de nd

Cas "Texte" :

 insérer le texte dans le fichier F

Cas "Image" :

 trouver le nom physique du fichier image à partir du catalogue

 l'insérer dans le fichier F

Cas "Lien" :

 trouver le nom physique de l'URL à partir du catalogue

 créer un lien référentiel vers cette URL dans le fichier F

Cas "Xobjet" :

 exécuter la vue

 insérer le résultat dans le fichier F

Cas "contexte simple" :

 exécuter la vue

 en déduire les Xobjets et un index

 insérer l'index dans le fichier F

 créer un fichier pour chaque Xobjet

Cas "contexte composé" : /* version simplifiée : seules 2 vues sont considérées ! */

 exécuter la 1^{ère} vue

 créer un index principal à partir de la 1^{ère} vue et l'insérer dans le fichier F

 Pour chaque élément de l'index principal

 exécuter la 2^{ième} vue conditionnellement à cet élément

 en déduire les Xobjets et un index secondaire

 créer un fichier pour l'index secondaire

 créer un fichier pour chaque Xobjet

 Fin Pour

Fin Selon

Fin

Algorithme 1 : Algorithme de génération d'unités médiatiques

PARTIE II:
Intégration de Services Web dans Xweb

Dans cette partie nous étudions le concept de services web et leur intégration dans Xweb (système de gestion de site web dynamique basé sur XML).

I. Les services web

1. Généralités

Le concept de services web propose simplement de faire communiquer des composants logiciels distants. Certains modèles (Corba, EJB) devaient, en théorie, apporter une solution fiable à cette problématique[31].

Mais deux difficultés se posent dans la pratique. D'une part, il est difficile de faire communiquer un objet d'un type avec un objet d'un autre type ; d'autre part, les coupe-feu des entreprises bloquent les protocoles de communication utilisés par ces objets, compliquant terriblement leur déploiement. Les services web permettent de résoudre ces problèmes.

Leur vocation est de permettre à des applications de dialoguer entre elles à distance via Internet, sans barrières techniques, c'est-à-dire sans même avoir besoin de savoir à priori comment elles ont été conçues et sur quelle plate-forme elles opèrent. Pour ce faire, les services web s'appuient sur des protocoles standardisant les modes d'invocation mutuels de composants applicatifs. Les standards sous-jacents doivent permettre notamment :

- La définition des protocoles de communication et des structures de messages échangés,
- La description et la publication des services pertinents dans un annuaire accessible à toute application susceptible de les invoquer,
- La transformation des formats de données s'ils sont différents entre l'émetteur et le récepteur du service,
- La description et la codification des objets métiers comme le produit, le fournisseur, le bon de commande, etc.

Le principe simple du service web permet de créer un modèle de type fournisseur/consommateur, où fournisseur et consommateur sont à même de se mettre en relation de manière souple, standardisée et dynamique.

2. Etude du fonctionnement des services web

2.1 Fonctionnement général [31]

Un service web est un composant logiciel qui interagit avec d'autres composants logiciels autonomes au moyen de protocoles universels.

Les services web permettent l'appel d'une méthode d'un objet distant en utilisant un protocole web pour le transport (HTTP en général) et XML pour formater les échanges.

Les services web fonctionnent sur le principe du client/serveur :

- Un client appelle le service web
- Le serveur traite la demande et renvoie le résultat au client
- Le client utilise le résultat

L'appel de méthodes distantes n'est pas une nouveauté mais la grande force des services web est d'utiliser des standards ouverts et reconnus : HTTP et XML. L'utilisation de ces standards permet d'écrire des services web dans plusieurs langages et de les utiliser sur des systèmes d'exploitation

différents.

Les services web utilisent des échanges de messages au format XML.

Il existe deux types de services web :

- synchrone : appel de méthodes (SOAP)
- asynchrone : échange de messages (SOAP, ebXML)

Les services web ne sont pas encore complètement matures à cause de la jeunesse des technologies utilisées pour les mettre en œuvre. Il reste encore de nombreux domaines à enrichir (sécurité, gestion des transactions, workflow, ...). Des technologies pour répondre à ces besoins sont en cours de développement.

Le schéma suivant montre le fonctionnement global des services web :

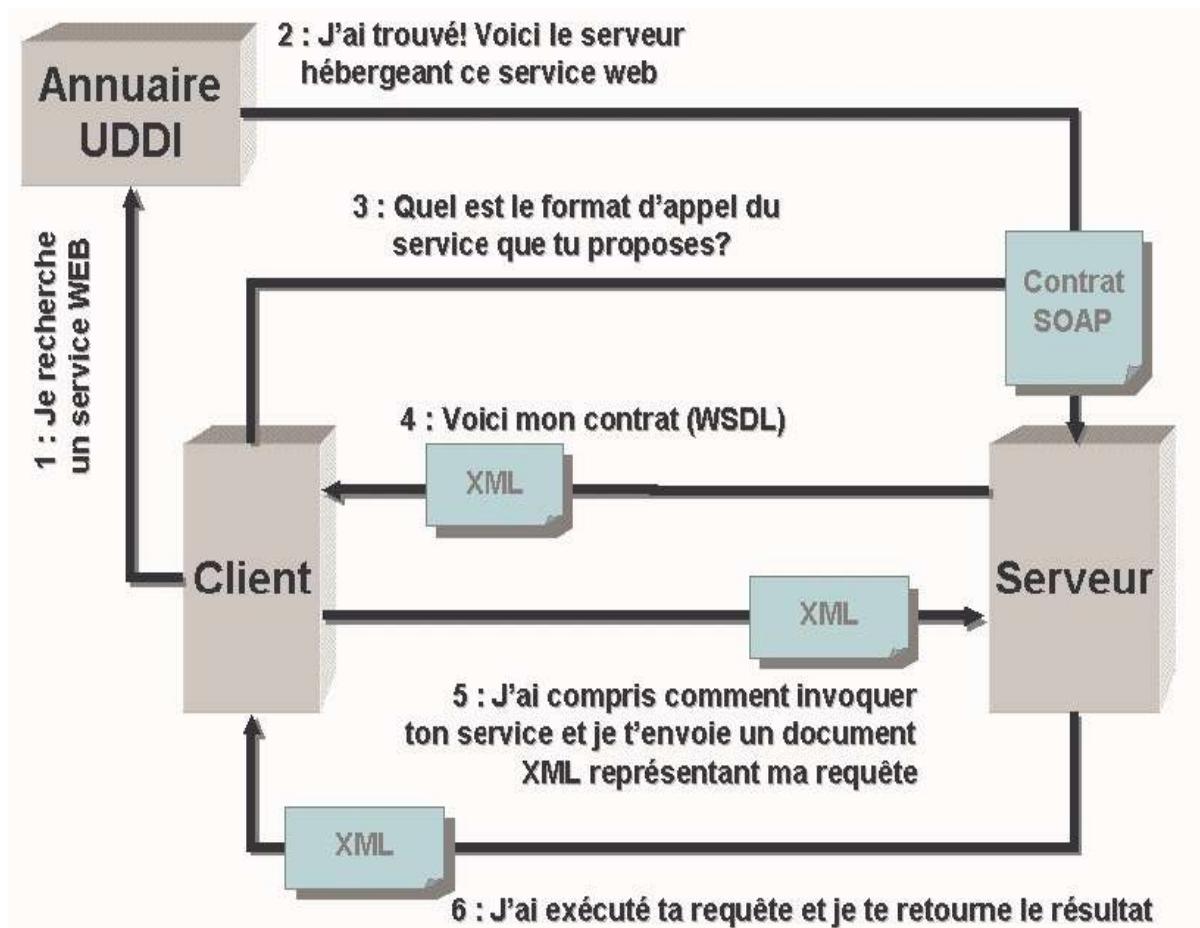


Figure7 : Architecture fonctionnelle des services web

Il est possible de détailler ces différentes étapes :

- Un service est créé en fonction d'un besoin lié à l'activité d'une entreprise à l'aide des outils et des langages appropriés
- Une fois créé, le service est publié dans un annuaire d'activités (UDDI) qui va contenir des informations complètes sur le service et son créateur (références de l'entreprise + références du service (adresse du fichier WSDL)).
- Le client fait une recherche dans un annuaire d'activités. Il récupère les références correspondant au service qu'il a choisi. Cette référence lui servira à utiliser le service

- Le service est invoqué au moyen de technologies standardisées, la transmission des données s'accomplit avec SOAP en combinant XML et HTTP.

Le client récupère le résultat voulu et le traite. L'application du client peut être développée dans n'importe quel langage, il faut juste qu'il utilise SOAP pour l'invocation du service

2.2 Technologies utilisées

Les services web utilisent trois technologies :

- SOAP (Simple Object Access Protocol) pour le service d'invocation : il permet l'échange de messages dans un format particulier
- WSDL (Web Services Description Language) pour le service de description : il permet de décrire les services web
- UDDI (Universal Description Discovery and Integration) pour le service de publication : il permet de référencer les services web

2.2.1 SOAP

SOAP signifie *Simple Object Access Protocol*. Ce protocole est celui utilisé pour la communication entre le client et le serveur qui héberge le service. Ce protocole est un peu spécial car il ne définit aucun protocole de transport. Les écrivains traitant de ce sujet aiment à dire que SOAP peut-être véhiculé par HTTP, SMTP ou même par pigeon voyageur [24].

SOAP est une norme de communication qui standardise l'échange de messages en utilisant un protocole de communication (le plus utilisé est HTTP) et XML pour formater les données [32].

SOAP se veut simple à utiliser et extensible.

SOAP peut être utilisé :

- pour l'appel de méthodes (SOAP RPC)
- pour l'échange de message (SOAP Messaging)

SOAP définit la structure principale du message, dite « enveloppe » qui contient deux parties :

- l'en-tête (Header) : facultatif
- le corps (Body) : obligatoire

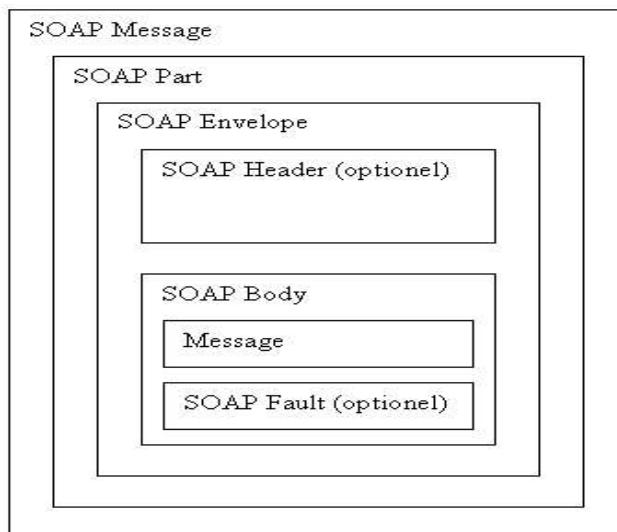


Figure 8: Structure principale d'un message défini par SOAP

Le corps est composé d'un ou plusieurs blocs. Un bloc contient des données ou un appel de méthode avec ses paramètres.

Un message SOAP peut aussi contenir des pièces jointes contenues chacune dans une partie optionnelle nommée AttachmentPart. Ces parties sont au même niveau que la partie SOAP Part.

SOAP définit aussi l'encodage pour les différents types de données qui est basé sur la technologie schéma XML du W3C. Les données peuvent être de type simple (chaine, entier, flottant, ...) ou de type composé.

Les types simples peuvent être

- un type de base : string, int, float, ...
- une énumération
- un tableau d'octets (array of byte)

Les types composés

- une structure (Struct)
- un tableau (Array)

La partie SOAP Fault permet d'indiquer qu'une erreur est survenue lors des traitements du service web. Cette partie peut être composée de 4 éléments :

- faultcode : indique le type de l'erreur (VersionMismatch en cas d'incompatibilité avec la version de SOAP utilisée, MustUnderstand en cas de problème dans le header du message, Client en cas de manque d'informations de la part du client, Server en cas de problème d'exécution des traitements par le serveur)
- faultstring : message décrivant l'erreur
- faultactor : URI de l'élément ayant déclenché l'erreur
- faultdetail : Détail de l'erreur non lié au Body du message

Pour l'appel de méthodes, plusieurs informations sont nécessaires :

- l'URI de l'objet à utiliser
- le nom de la méthode
- éventuellement le ou les paramètres

La sécurité est un sujet souvent abordé lors de conversation au sujet de SOAP. SOAP ne gère aucun mécanisme de sécurité. Cependant, rappelons que celui-ci n'est pas lié à un protocole de transport défini. Ainsi, il est possible d'augmenter le niveau de sécurité des messages en utilisant HTTPS en lieu et place d'HTTP. Le service se reposant sur le serveur, Web par exemple, pour les tâches de transport, ceci n'est pas forcément très compliqué à mettre en place.

Les trois exemples suivants représentent les messages SOAP envoyés, respectivement reçus par le client [33].

L'envoi :

Demande de calcul à un serveur

```
POST /path/foo.pl HTTP/1.1
Content-Type: text/xml
SOAPAction: interfaceURI#Add
Content-Length: nnnn
<soap:Envelope xmlns:soap='uri for soap'>
<soap:Body>
<Add xmlns='interfaceURI'>
<arg1>24</arg1>
<arg2>53.2</arg2>
</Add>
</soap:Body>
</soap:Envelope>
```

Et le retour s'il n'y a pas d'erreur:

```
200 OK
Content-Type: text/xml
Content-Length: nnnn
<soap:Envelope
  xmlns:soap='uri for soap'>
<soap:Body>
<AddResponse xmlns='interfaceURI'>
<sum>77.2</sum>
</AddResponse>
</soap:Body>
</soap:Envelope>
```

En vert se trouve la valeur souhaitée. On voit ici que, par rapport à d'autres solutions comme CORBA ou RMI, il y a une quantité non négligeable de données qui transitent pour une quantité utile relativement faible. Notons également l'espace de nom interfaceURI qui permet de s'assurer que le message soit bien celui attendu.

Et le retour s'il y a erreur:

```
<SOAP:Envelope
  xmlns:SOAP="urn:schemas-xmlsoap-org:soap.v1">
<SOAP:Body>
<SOAP:Fault>
<faultcode>200</faultcode>
<faultstring>
  SOAP Must Understand Error
</faultstring>
</SOAP:Fault>
<SOAP:Body>
</SOAP:Envelope>
```

2.2.2 WSDL

WSDL signifie *Web Services Description Language*. Il s'agit d'un langage, lui aussi basé sur XML, permettant de définir un service Web. Ce fichier contient le chemin d'accès au serveur, mais aussi les prototypes des méthodes qui servent de points d'entrées et la description des types complexes [24].

WSDL peut être vu comme un complément de SOAP car il facilite l'interopérabilité des Web Services. Tout comme IDL (Interface Definition Language) qui joue le rôle de descripteur de

services avec CORBA, WSDL est une syntaxe XML pour décrire les Web Services. Grâce à WSDL, les applications seront capables d'auto-configurer les échanges entre Web Services, tout en masquant la plupart des détails techniques de bas niveau [25].

Concrètement, les services sont définis dans un document WSDL au moyen de six éléments principaux :

- types : fournissent des définitions de types de données afin de décrire les messages échangés.
- message : sert à représenter une définition abstraite des données transmises ; un message est constitué de parties logiques, chacune étant associée à une définition de type.
- type de port (*portType*) : ensemble d'opérations abstraites. Chaque opération se réfère à un message entrant et à des messages sortants.

Une opération est composée d'un message de type input et d'un message de type output. Les deux types de message sont optionnels et leur ordre désigne le type de l'opération:

- One-way: input sans output
- Request-response: input suivi de output
- Solicit-response: output suivi de input
- Notification: output sans input

• rattachement ou liaison WSDL (*binding*) : ce qui spécifie les aspects concrets de protocole de communication et le format des données pour les opérations et messages définis par un type de port particulier.

• port : ce qui spécifie une adresse pour un rattachement, déterminant ainsi un seul noeud branché, ou point terminal (*endpoint*).

• service : ce qui sert à regrouper un ensemble de ports.

Dans [24], le client possède généralement des outils permettant de créer des proxy de manière tout aussi automatique. Une classe proxy est une sorte d'interface (pas au sens Java) permettant d'accéder à un service Web. Elle encapsule le code nécessaire à la communication distante, ce qui permet, lors de l'implémentation du client, d'accéder au service Web comme s'il s'agissait d'un objet local.

Notons que la classe proxy permet donc d'implémenter tout type de client imaginable application en ligne de commande, en mode fenêtré, applet, HTML dynamique, etc. Les trois premiers sont considérés comme des clients lourds car ils s'exécutent sur la machine de l'utilisateur, alors que le dernier est nommé un client léger car il s'exécute sur un serveur Web et ne nécessite donc aucune installation sur la machine du client.

Lors de la réalisation de projets plus complexes que de simples exemples, il apparaît assez vite que la génération automatique du schéma WSDL n'est pas suffisante. Il est souvent nécessaire de modifier le fichier WSDL ou la classe *proxy* générée. Pour le premier, une partie du travail peut souvent être réalisée dans le code, par exemple à l'aide de scripts (chez BEA) ou d'attributs (.NET).

Voici un exemple de code WSDL. Celui-ci définit un service Web possédant une méthode HelloWorld, sans paramètre et retournant une chaîne de caractères. Celui-ci a été simplifié de manière à ne permettre que les communications via SOAP.

```
<?xml version="1.0" encoding="utf-8" ?>
<definitions
    xmlns:ng="http://schemas.xmlsoap.org/wsdl/extension/"
    xmlns:tm="http://microsoft.com/wsdl/mime/textMatching"
    xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
    targetNamespace="http://eivd.ch/helloworld"
    xmlns="http://schemas.xmlsoap.org/wsdl/">
    <types>
```

```

<s:schema elementFormDefault="qualified"
targetNamespace="http://eivd.ch/helloworld">
  <s:element name="HelloWorld">
    <s:complexType />
  </s:element>
  <s:element name="HelloWorldResponse">
    <s:complexType>
      <s:sequence>
        <s:element minOccurs="0" maxOccurs="1"
          name="HelloWorldResult" type="s:string" />
      </s:sequence>
    </s:complexType>
  </s:element>
  <s:element name="string" nillable="true" type="s:string" />
</s:schema>
</types>
<message name="HelloWorldSoapIn">
  <part name="parameters" element="s0:HelloWorld" />
</message>
<message name="HelloWorldSoapOut">
  <part name="parameters" element="s0:HelloWorldResponse" />
</message>
<portType name="HelloSoap">
  <operation name="HelloWorld">
    <documentation>Renvoie un
HelloWorld</documentation>
    <input message="s0:HelloWorldSoapIn" />
    <output message="s0:HelloWorldSoapOut" />
  </operation>
</portType>
<binding name="HelloSoap" type="s0:HelloSoap">
  <soap:binding
    transport="http://schemas.xmlsoap.org/soap/http"
    style="document" />
  <operation name="HelloWorld">
    <soap:operation soapAction=
      "http://eivd.ch/helloworld/HelloWorld"
    style="document" />
    <input>
      <soap:body use="literal" />
    </input>
    <output>
      <soap:body use="literal" />
    </output>
  </operation>
</binding>
<service name="Hello">
  <documentation>Premier service
Web</documentation>
  <port name="HelloSoap" binding="s0:HelloSoap">
    <soap:address location=
      "http://ap0501>HelloWorld/HelloWorld.asmx" />
  </port>
  <port name="HelloHttpGet"
    binding="s0:HelloHttpGet">
    <http:address location=
      "http://ap0501>HelloWorld/HelloWorld.asmx" />
  </port>
</service>
</definitions>

```

2.2.3UDDI

UDDI (Universal Description, Discovery and Integration) est un protocole et un ensemble de services pour utiliser un annuaire afin de stocker les informations concernant les services web et de permettre à un client de les retrouver. Il existe plusieurs serveurs, tous compatibles avec le protocole UDDI qui permettent d'automatiser la recherche de services. Nous noterons par exemple uddi.ibm.com et uddi.microsoft.com. Ces serveurs sont atteignables via un navigateur Web standard.

L'interface Web des serveurs UDDI permet généralement d'inscrire un nouveau service Web ou de retrouver des services existants. Il est parfois possible de les atteindre avec des services Web pour, par exemple, automatiser la recherche ou de permettre aux clients d'être plus indépendants de la localisation physique du service [24].

L'annuaire [22] inclut des informations sur les fournisseurs de services, les services hébergés et les protocoles mis en oeuvre par ces services. L'annuaire propose également des mécanismes pour ajouter des métadonnées à toute information enregistrée.

Cette approche par annuaire est intéressante lorsque les informations sont stockées dans les emplacements bien connus. Une fois que l'annuaire est localisé, il reste à l'interroger pour obtenir les informations souhaitées. L'emplacement de l'annuaire UDDI est généralement défini par l'administrateur du système.

Les fournisseurs de services web présentent diverses options pour déployer les registres UDDI. Les scénarios de déploiement se répartissent en trois grandes catégories: déploiement public, inter-entreprise et intra-entreprise. Pour les déploiements publics, un groupe d'entreprises regroupant notamment Microsoft, IBM et SAP, soutiennent UBR (UDDI Business Registry)[23]. UBR est un registre public repliqué dans de nombreuses entreprises. Il sert à la fois de ressources pour les services web présents sur Internet et de test pour les développeurs de services web. Bien que l'implémentation publique d'UDDI ait reçu le plus d'attention jusqu'à présent, les concepteurs utilisent le plus souvent l'approche inter- et intra-entreprise.

Dans ces deux scénarios de déploiement, un registre privé est déployé par une entreprise; un contrôle plus étroit sur les types d'informations enregistrées est alors possible. Ces registres privés peuvent être dédiés à une seule entreprise ou à des groupes de partenaires commerciaux.

Les annuaires UDDI contiennent des informations détaillées concernant les services web et leurs emplacements. Une entrée d'annuaire UDDI se compose de trois parties principales: le fournisseur du service, les services web offerts et les liens vers les implémentations. Chacune de ces parties donne progressivement davantage d'informations sur le service web.

L'information la plus générale décrit le fournisseur du service. Elle n'est pas destinée à un logiciel mais à un développeur ou à un installateur qui peut avoir besoin de contacter directement quelqu'un responsable du service. Cette information inclut des noms, des adresses, des contacts et d'autres détails administratifs.

La liste des services web disponibles est stockée dans une entrée de type fournisseur de service. Les services peuvent être organisés en fonction de leur utilisation prévue: ils peuvent être groupés en domaines d'application, par régions géographiques, ou tout autre schéma approprié. Une information de service stockée dans un registre UDDI inclut simplement une description d'un service et un pointeur vers les implémentations de service web qu'il contient. Il est aussi possible d'enregistrer des liens vers des services hébergés par d'autres fournisseurs. Ces liens se nomment « projection de service ».

La dernière partie d'une entrée de fournisseur de service UDDI est la liaison vers une implémentation. La liaison associe l'entrée de service web à l'URI exact identifiant l'emplacement du service. Cette liaison spécifie aussi le protocole à utiliser pour accéder au service. Enfin, elle contient des références aux protocoles exacts qui sont mis en œuvre.

Ces détails sont suffisants pour qu'un développeur puisse écrire une application qui fasse appel au service web. La définition détaillée du protocole est fournie par une entité UDDI nommée Type Model (ou tModel). Dans de nombreux cas, le tModel référence un fichier WSDL qui décrit l'interface SOAP du service web mais les entités tModel sont suffisamment flexibles qu'elles peuvent servir à décrire pratiquement n'importe quel type de ressource.

3Implémentation

3.1JAX-RPC

JAX-RPC [26] (Java API for XML based RPC) est une API permettant l'appel de méthodes distantes et la réception de leur réponse en utilisant SOAP 1.1 et HTTP 1.1. Elle facilite la communication via Internet en permettant à des paramètres au format XML d'être passés à des services distants et en permettant à des valeurs au format XML d'être renvoyées.

JAX-RPC inclut un modèle extensible de mapping de types de données (mécanisme de

correspondance entre types de données) associé à des mécanismes de sérialisation/désérialisation entre objets Java et types de données XML-Schema. D'autre part, JAX-RPC définit des fonctionnalités de mapping à double sens entre Java (classes interfaces, méthodes), et descriptions XML/WSDL d'interfaces de services web.

Le grand avantage de cette API est de masquer un grand nombre de détails de l'utilisation de SOAP et ainsi de la rendre facile à utiliser.

Le schéma ci-dessous est une vue simplifiée de ce qui se passe lors de l'appel d'un service web JAX-RPC appelé Calculator par un client.

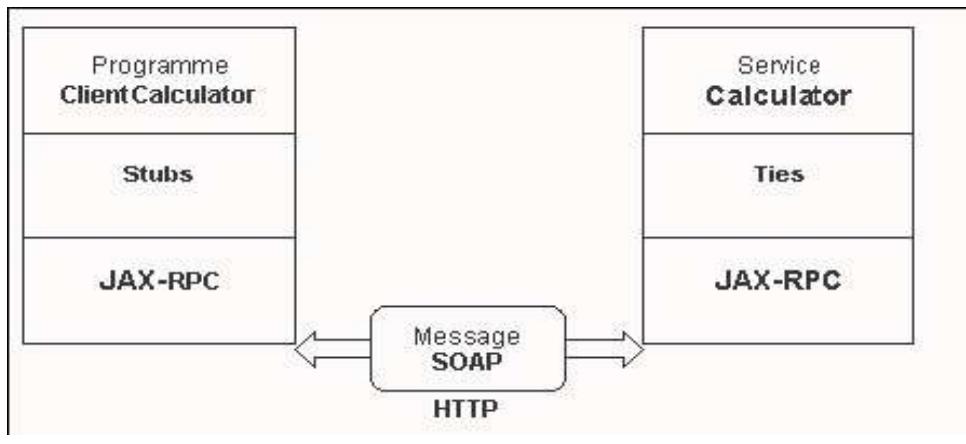


Figure 9: Schéma d'invocation d'un service web JAX-RPC

Voici une description plus détaillée de ce qui se passe :

- Pour appeler une procédure distante, le programme CalculatorClient invoque une méthode d'un stub, un objet local qui représente le service
- Le stub invoque des méthodes de JAX-RPC
- JAX-RPC convertit les appels à distance en un message SOAP et le transmet comme une requête HTTP
- Le serveur reçoit la requête, JAX-RPC extrait le message SOAP et le transforme en un appel de méthode
- JAX-RPC invoque la méthode de l'objet tie
- L'objet tie invoque la méthode présente dans l'implémentation du service Calculator
- Le serveur convertit la réponse en un message SOAP et le transmet au client comme une requête HTTP
- Chez le client, JAX-RPC extrait le message SOAP de la réponse et le transforme en réponse pour le programme CalculatorClient

Lors de la programmation du service ou du client, le programmeur ne s'occupe que des couches supérieures : les implémentations.

L'implémentation du service se passe en plusieurs phases :

- Ecriture des classes JAVA représentant le service (interface + implémentation)
- Utilisation de wsdeploy pour générer toutes les classes nécessaires au service et le fichier WSDL.

- Déploiement sur le serveur
 - L'implémentation du client se passe en plusieurs phases :
- Ecriture du client JAVA
- Utilisation de wscompile pour générer les classes nécessaires au client à partir du fichier WSDL décrivant le service
- Exécution du client
 - Le client du service web peut être codé de différentes façons en fonction du type de stub utilisé :
 - Stub statique : le stub est créé avant l'exécution par wscompile.
 - Stub dynamique : le stub est créé pendant l'exécution. Avant de le créer, le client récupère les informations concernant le service dans le fichier WSDL.
- DII (Dynamic Invocation Interface) : le client peut appeler une procédure distante même si le nom de la procédure ou du service ne sont pas connus avant l'exécution.
 - Ce dernier type est le plus flexible, il peut être utilisé dans le cas d'une application recherchant les services dans un annuaire, configure et exécute les appels aux méthodes distantes.

3.2JAXM

JAXM [27] (Java API for XML Messaging) permet de créer, d'envoyer et de recevoir des documents XML. Le grand avantage de cette API est qu'elle permet de faire abstraction de la syntaxe SOAP, un peu plus laborieuse et compliquée. Les services web basés sur JAXM utilisent les échanges de documents contrairement à ceux basés sur JAX-RPC qui font appel à des méthodes. Les services web basés sur JAXM peuvent être utilisés dans les cas suivants :

- Quand les interactions entre client et service sont asynchrones. JAXM propose un support pour les échanges synchrones et asynchrones de documents XML
- Quand client et service veulent échanger des données à l'intérieur de documents XML utilisant des schémas XML bien définis au lieu d'appeler des méthodes

L'API JAXM se conforme à la spécification SOAP 1.1 ainsi qu'à la spécification SOAP with Attachments. Cette API se compose de deux packages principaux :

- javax.xml.soap : le package défini dans la spécification SOAP with Attachments API for JAVA (SAAJ) et qui constitue le package de base de l'échange de messages SOAP, contenant ainsi l'API destinée à créer et à remplir un message SOAP. Ce package contient toutes les API nécessaires pour envoyer des messages requêtes/réponses (communication synchrone).
- javax.xml.messaging : le package défini dans la spécification JAXM 1.1. Ce package contient l'API nécessaire afin d'utiliser un fournisseur de messages (messaging provider), et ainsi d'envoyer des messages one-way (communication asynchrone). Le second package est dépendant du premier, l'inverse n'étant pas vrai. Ainsi, un client envoyant des messages requêtes/réponse peut tout à fait se contenter de l'API javax.xml.soap.

Le service JAXM traite les messages JAXM envoyés par le client JAXM. Pour développer le service, il faut créer une servlet héritant de javax.xml.messaging.JAXMServlet.

Elle doit implémenter soit l'interface OneWayListener soit l'interface ReqRespListener selon que la communication entre le client et le service sera asynchrone ou synchrone.

Le client JAXM échange des messages avec le service JAXM. Il peut interagir soit directement avec le service soit à travers un provider selon le type de communication choisi.

Un client qui utilise un provider peut participer à des interactions synchrones ou asynchrones alors qu'un client n'utilisant pas de provider peut seulement participer à des interactions synchrones.

Un provider de messages est un service qui permet la transmission et le routage des messages.

L'utilisation d'un provider permet d'avoir une séparation entre le client et le service. Elle permet aussi d'avoir accès à des services supplémentaires tels que la sécurité ou la gestion de protocoles (eb-XML par exemple).

3.3.NET

Le *framework* .NET [29] est très récent. Son développement a été parallèle à celui des services Web. Il contient déjà toutes les librairies et tous les outils nécessaires pour la réalisation de services Web. Ainsi, les espaces de nom System.Web.Services ou System.Web.Protocols.Soap contiennent les objets de base pour la sérialisation ou l'implémentation des services, de la sérialisation, etc.

Tout service Web réalisé en .NET doit hériter de l'objet System.Web.Services.WebService.

Le code permet de spécifier des attributs pour les méthodes, les classes, etc. Ces attributs contiennent des informations pour les méthodes de sérialisation. Ainsi, un attribut *WebMethod* d'une méthode définit celle-ci comme méthode web, qui peut alors être appelée par un client. Si une méthode n'est pas précédée par un attribut *WebMethod*, elle ne sera pas disponible pour le client. L'attribut *WebMethod* permet de modifier l'espace de nom XML de la méthode Web, son nom, s'il diffère de celui de la méthode de l'objet ou encore le fait que la méthode utilise les options de sessions du serveur Web.

L'attribut *WebService* quant à lui n'est pas obligatoire. Il permet principalement de modifier l'espace de nom du service Web. Si celui-ci n'est pas redéfini pour les méthodes ou les types, ceux-ci récupèrent celui du service.

Un autre avantage du *framework* .NET est Visual Studio. Certes, l'acquisition de celui-ci est onéreuse, mais le temps de développement est fortement diminué. De même, des outils gratuits tel Asp.NET Web Matrix ou SharpDevelop permettent également d'accélérer le développement de services Web.

On le voit, le développement de service Web est bien intégré dans la technologie .NET. Ceci devrait amener de nombreux développeurs à se rapprocher de celle-ci.

3.4BEA Weblogic

BEA [30] a développé un ensemble d'outils et de librairies pour le développement de service Web à l'aide du langage Java. Le principal problème de cette solution est son prix. En effet, contrairement aux deux autres solutions proposées dans ce document, celle-ci est payante ; Il est cependant possible d'obtenir une version d'évaluation d'une durée de nonante jours.

L'outil BEA Weblogic workshop s'occupe de tout pour nous. Le résultat est que nous développons des services Web qui fonctionnent, mais le développeur moyen ne sait pas par quelle magie cela fonctionne.

Le Workshop de BEA permet de développer rapidement des services Web servant d'interface avec des EJB ou d'autres objets déjà existants. Leur développement en soi n'est donc pas très complexe. Cependant, il est nécessaire de garder à l'esprit que, même si le temps de développement d'un service Web est fortement diminué, il reste très important de respecter certaines règles.

Notons encore que l'outil de BEA est livré avec un serveur d'application faisant office de serveur Web. Il est possible d'utiliser ce serveur pour faire fonctionner des EJB ou *Enterprise Java Beans*.

Il n'est pas nécessaire d'utiliser ceux-ci pour réaliser des services Web ; ils permettent cependant, à condition d'être utilisés correctement, de gérer les transactions, ce qui n'est pas possible si nous utilisons uniquement les services Web. Dans ce cas, les services Web sont une sorte d'interface d'accès aux EJB.

3.5Apache Axis

Apache Axis [28] est une implémentation open-source sur la plate-forme Java des spécifications SOAP version 1.1 et SOAP Messages with Attachments (messages SOAP avec pièces jointes), et inclut certaines fonctionnalités de la nouvelle version de la norme, SOAP 1.2. Axis est développé par la fondation Apache.

Apache Axis est à la fois un environnement d'hébergement de services web, et un toolkit complet de développement pour la création de services ainsi que l'accès client à des services tiers. Ce toolkit comporte des fonctionnalités avancées de génération de code automatique basées sur l'analyse de descriptions WSDL de services web et le mapping automatique à double sens entre classes Java et descriptions WSDL de services.

Axis est un projet qui est issu du projet SOAP du projet Apache. Pour plus de renseignements, le site xml.apache.org contient toutes les informations nécessaires.

Le projet Apache fournit un serveur Web efficace (très connu dans le monde Linux) ; cependant, bon nombre d'autres projets gravitent autour de celui-ci. Le projet AXIS en est un exemple. Pour le moment encore en version Beta 3, il s'agit en fait de la version 3 de Apache SOAP. Cette API est gratuite mais nécessite un serveur d'application pour le support des EJB.

La librairie permet cependant les manipulations habituelles, telles que le développement de service ou de client. Le principal problème est que la version actuelle est une Beta. Ceci implique qu'il faut, comme bien souvent dans ce cas, se contenter d'une version pas toujours très stable, mais surtout plus complexe à mettre en oeuvre. Enfin, notons que la librairie d'Apache ne fournit pas d'outils de développement aussi efficace que le Weblogic Workshop de BEA.

Axis est un package qui fournit :

- Un environnement pouvant soit fonctionner comme un serveur SOAP indépendant soit comme un plug-in de moteurs de servlet (en particulier TOMCAT)
- Une API pour développer des services web SOAP RPC ou à base de messages SOAP
- Le support de différentes couches de transport : HTTP, FTP...
- La sérialisation/désérialisation automatique d'objets Java dans des messages SOAP
 - Des outils pour créer automatiquement les WSDL correspondant à des classes Java ou inversement pour créer les classes Java sur la base d'un WSDL (classe proxy en quelque sorte, qui fait le lien entre l'application Java cliente et le service distant)
 - Des outils pour déployer, tester et monitorer des web-services

II. Intégration de services web dans Xweb

Dans cette section nous étudions l'intégration de services web dans xweb. Cette intégration doit permettre la conception de sites web permettant l'accès à des services web.

1. Unité médiatique de type service web

Nous rappelons qu'une unité médiatique (U.M.) est une « unité informative qui a une certaine autonomie du point de vue d'un utilisateur (i.e qui a un sens en soi et donc présente une idée ou un concept cohérent) et qui mérite d'être sollicitée dans plusieurs démarches de consultation »[5].

Il existait huit types d'unités médiatiques dans xweb et nous en avons ajouté une de plus de type service web.

Une unité médiatique de type service web décrit un service afin de le rendre accessible à partir d'une autre U.M. . Elle fournit le nom, le fournisseur, les fonctionnalités et l'adresse WSDL du service.

Voici un exemple d'unité médiatique de type service web qui permet de décrire le service nommé Calcul1Service.

```
<um id="sw" type="service_web">
<nom>CalculService</nom>
<fournisseur>LANI</fournisseur>
<fonctionnalites>
  <fonction>somme</fonction>
  <fonction>produit</fonction>
</fonctionnalites>
<adresse_wsdl>http://localhost:8080/axis/calcul.jws?wsdl</adresse_wsdl>
</um>
```

2.Extension de l'algorithme de génération d'U.M.

Xweb permet de générer un (ou plusieurs) site(s) web à partir des données XML d'une base de documents XML.

Xweb se compose de trois couches: Données, Application et Présentation.

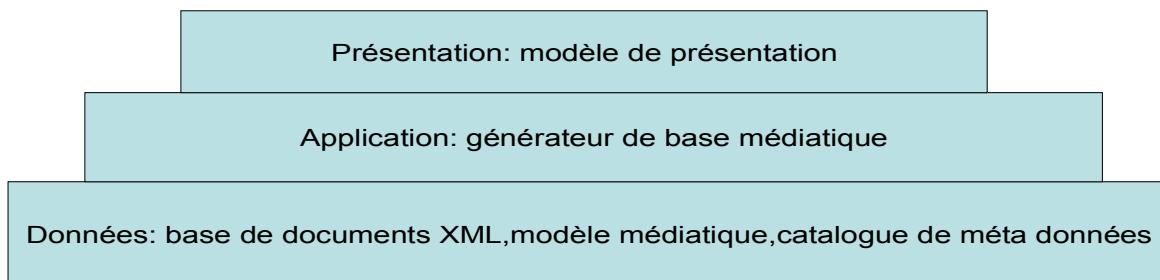


Figure 10: Schématisation du système Xweb

La couche Données offre à l'administrateur du système Xweb la possibilité de définir et de mettre à jour un modèle médiatique ou un catalogue de métadonnées.

La couche Application permet de générer automatiquement une base médiatique (matérialisant le contenu d'un site web) à partir d'un modèle médiatique. Elle utilise l'algorithme de génération d'unités médiatiques.

La couche Présentation permet d'associer à chaque document XML d'une base médiatique une forme médiatique (feuille de style XSL ou CSS) définie par l'administrateur afin de permettre une visualisation conviviale du côté client.

Nous avons étendu l'algorithme de génération d'unités médiatiques en lui permettant de pouvoir traiter les unités médiatiques de type service web.

Procédure Générer_Page (Racine : Noeud)

Début

Ouvrir en écriture le fichier (nommé F) dans lequel sera créé l'UM de type «unité page»

/* le nom physique est déterminé à partir de la balise id et du catalogue */

Pour chaque noeud fils (nommé nd_fils) de la racine de l'arbre Faire

Traiter_Noeud (nd_fils, F)

Fin Pour

Fin

Procédure Traiter_Noeud (nd : Noeud, F : Fichier)

Debut

Selon le type de nd

Cas "Texte" :

insérer le texte dans le fichier F

Cas "Image" :

trouver le nom physique du fichier image à partir du catalogue

l'insérer dans le fichier F

Cas "Lien" :

trouver le nom physique de l'URL à partir du catalogue

créer un lien référentiel vers cette URL dans le fichier F

Cas "Xobjet" :

exécuter la vue

insérer le résultat dans le fichier F

Cas "contexte simple" :

exécuter la vue

en déduire les Xobjets et un index

insérer l'index dans le fichier F

créer un fichier pour chaque Xobjet

Cas "contexte composé" : /* version simplifiée : seules 2 vues sont considérées ! */

exécuter la 1ère vue

créer un index principal à partir de la 1ère vue et l'insérer dans le fichier F

Pour chaque élément de l'index principal

exécuter la 2ième vue conditionnellement à cet élément

en déduire les Xobjets et un index secondaire

créer un fichier pour l'index secondaire

créer un fichier pour chaque Xobjet

Fin Pour

Cas "service web" :

/* un fichier vide nommé uddi.xml ayant un seul et unique noeud (<racine/>)

sera créé au préalable par l'administrateur du système Xweb */

insérer le nom du service web dans le fichier uddi.xml

insérer et le nom et les autres informations relatives au service dans le fichier F

Fin Selon

Fin

Algorithme 2 : Algorithme de génération d'unités médiatiques et d'intégration de services web dans Xweb**3.Les différentes étapes de la procédure d'intégration de services web dans Xweb**

a) Créer en tant que administrateur du système Xweb un fichier XML vide nommé uddi.xml (i.e ce document XML ne comporte qu'un seul noeud qui est le noeud racine).

b) Décrire formellement en XML les unités médiatiques de type « service web » à intégrer dans le système Xweb.

L'algorithme de génération d'une page étant maintenant en mesure de traiter des unités médiatiques de type « service web » s'opère en recevant en entrée une U.M. de ce type comme suit :

- il ouvre d'abord en écriture le fichier uddi.xml et y insére le nom du service web émanant de l'U.M. reçue en entrée (i.e dans ce cas la liste des services web disponibles sera stockée dans une entrée de type nom de service).
- il génère ensuite une U.M. de type «unité page» à partir de la description de l'unité médiatique reçue comme paramètre d'entrée.

PARTIE III:
Application

Nous montrons dans cette partie les différents aspects d'utilisation de Xweb en l'appliquant d'abord à la construction d'une application web pour la Direction Régionale du Développement Rural (DRDR) de Saint-Louis.

1. Application à des données environnementales

Nous avons utilisé Xweb pour réaliser un site web à partir de données fournies par la DRDR de Saint-Louis. Les données reçues étaient sous format Excel et portaient sur l'agriculture et l'élevage dans la région de Saint-Louis. Les données Excel ont été d'abord transformées en XML grâce à l'utilisation d'un wrapper développé en Java [21].

Cela a permis d'obtenir un dataweb dont le contenu est fourni en annexe.

La base de documents XML dénommée source globale comprend le fichier « source_globale.xml » qui est un agrégat de documents XML issus des données Excel de la DRDR.

Voici un extrait du document « source_globale.xml » (le document complet est fourni en annexe).

source_globale.xml

```

<source_globale>
  <prix_au_producteur_an_2000>
    <produits>
      <produit>
        <nom>Riz paddy</nom>
        <prix>105FCFA/Kg</prix>
      </produit>
      ...
      <traitements>
        <traitement>
          <region>Dagana</region>
          <superficie_prospectee>15000
          ha</superficie_prospectee>
          <superficie_infestee>12200
          ha</superficie_infestee>
          <superficie_traitee>350 ha</superficie_traitee>
        </traitement>
        <traitement>
          <region>Podor</region>
          <superficie_prospectee>16200
          ha</superficie_prospectee>
          <superficie_infestee>8700 ha</superficie_infestee>
          <superficie_traitee>350 ha</superficie_traitee>
        </traitement>
      </traitements>
    </produits>
  </source_globale>
  <region>Saint-Louis</region>
  <superficie_prospectee>12500
  ha</superficie_prospectee>
  <superficie_infestee>9070 ha</superficie_infestee>
  <superficie_traitee>632 ha</superficie_traitee>
  </traitement>
  </traitements>
  ...
  <campagne>
    <annee>2001/2002</annee>
    <sup_totales>8224,9 ha</sup_totales>
    <sup_sous_canne>7440,3 ha</sup_sous_canne>
    <sup_recoltees>7295 ha</sup_recoltees>
    <production_cannes>840109
    T</production_cannes>
    <rendt>115,2 T/ha</rendt>
    <qte_estimee_melasse>33604,4
    T</qte_estimee_melasse>
    </campagne>
    </campagnes>
  </source_globale>
  <nom_physique>..\donnees\ndakarou_kheweul.gif</nom_physique>
  </image>
  <lien id="lprod">
```

Le catalogue de métadonnées

```

<?xml version="1.0" standalone="yes" ?>
<!DOCTYPE catalogue (View Source for full doctype...)>
<catalogue>
  <image id="imnk">
    <nom_logique>ndakarou kheweul</nom_logique>
    <nom_physique>..\donnees\ndakarou_kheweul.gif</nom_physique>
    </image>
    <lien id="lprod">
```

```

<nom_logique>prix au producteur pour l'annee
2000</nom_logique>                                <nom_logique>statistiques d'exportation du cheptel dans
                                                    la region de Saint-Louis</nom_logique>

<nom_physique>..\resultats\produits.xml</nom_physique> <nom_physique>..\resultats\stat_exports.xml</nom_
e>                                                 physique>
                                                    </lien>
\_<lien id="lexport"> <nom_logique>statistiques d'exportation des
semences</nom_logique>                                <nom_logique>production de cannes a sucre dans la
                                                    region de Saint-Louis</nom_logique>

<nom_physique>..\resultats\exports.xml</nom_physique> <nom_physique>..\resultats\campagnes.xml</nom_physique>
</nom_physique> </lien>
\_<lien id="levolut"> <nom_logique>evolution des fourchettes de prix des
rizieres</nom_logique>                                <nom_logique>Retour a la page
                                                    d'accueil</nom_logique>

<nom_physique>..\resultats\evolutions.xml</nom_physique> <nom_physique>..\resultats\page_accueires.xml</nom_p
ue> </lien>
\_<lien id="ltrait"> <nom_logique>bilan des traitements des CLV dans la
region de Saint-Louis en 2002-2003</nom_logique> <nom_physique>..\resultats\uddi.xml</nom_physique>
                                                    </lien>
\_<lien id="lpropoport"> <nom_logique>proportion des especes dominantes en
pourcentage par departement</nom_logique> <nom_physique>..\resultats\client.html</nom_physique>
                                                    </lien>
</nom_physique> <nom_logique>Exemple d'un client utilisant la fonction
                                                    somme du service Calcul</nom_logique>
\_<lien id="lstateexport"> </catalogue>

```

Un modèle et une base médiatique

Description du modèle médiatique:

```

<modele_mediatique>
<baseXML>..\donnees\source_globale.xml</baseXML>
<repertoire_resultat>..\resultats\</repertoire_resultat>
<fichier>..\donnees\page_accueil.xml</fichier>
<fichier>..\donnees\um_xob_produits.xml</fichier>
<fichier>..\donnees\um_xob_exports.xml</fichier>
<fichier>..\donnees\um_xob_evolutions.xml</fichier>
<fichier>..\donnees\um_xob_stat_exports.xml</fichier>
<fichier>..\donnees\um_xob_campagnes.xml</fichier>
<fichier>..\donnees\um_xob_traitements.xml</fichier>
<fichier>..\donnees\um_cont_proportions.xml</fichier>
<fichier>..\donnees\um_service_web_lani.xml</fichier>
<fichier>..\donnees\um_service_web_khadim.xml</fichier>
</modele_mediatique>

```

Description de l'UM de la page d'accueil et résultat obtenu :**Définition de l'UM**

```
<um id="page_accueil" presentation=".\\resultats\\menu.xsl">
<image>ndakarou kheweul</image>
<titre>Bienvenue sur le site de la DRDR</titre>
<lien id=".\\resultats\\produits.xml">prix au producteur pour l'annee 2000</lien>
<lien id=".\\resultats\\exportations.xml">statistiques d'exportation des semences</lien>
<lien id=".\\resultats\\evolutions.xml">evolution des fourchettes de prix des rizieres</lien>
<lien id=".\\resultats\\traitements.xml">bilan des traitements des CLV dans la region de Saint-Louis en
2002-2003</lien>
<lien id=".\\resultats\\proportions.xml">proportion des especes dominantes en pourcentage par
departement</lien>
<lien id=".\\resultats\\stat_exports.xml">statistiques d'exportation du cheptel dans la region de Saint-
Louis</lien>
<lien id=".\\resultats\\campagnes.xml">production de cannes a sucre dans la region de Saint-Louis</lien>
<lien id=".\\resultats\\uddi.xml">L'annuaire de services web</lien>
</um>
```

UM générée :

```
<?xmlstylesheet type="text/xsl" href=".\\resultats\\menu.xsl" ?>
<racine>
<image href=".\\donnees\\ndakarou_kheweul.gif" />
<titre>Bienvenue sur le site de la DRDR</titre>
<lien href=".\\resultats\\produits.xml">prix au producteur pour l'annee 2000</lien>
<lien href=".\\resultats\\exportations.xml">statistiques d'exportation des semences</lien>
<lien href=".\\resultats\\evolutions.xml">evolution des fourchettes de prix des rizieres</lien>
<lien href=".\\resultats\\traitements.xml">bilan des traitements des CLV dans la region de Saint-Louis en
2002-2003</lien>
<lien href=".\\resultats\\proportions.xml">proportion des especes dominantes en pourcentage par
departement</lien>
<lien href=".\\resultats\\stat_exports.xml">statistiques d'exportation du cheptel dans la region de
Saint-Louis</lien>
<lien href=".\\resultats\\campagnes.xml">production de cannes a sucre dans la region de Saint-
Louis</lien>
<lien href=".\\resultats\\uddi.xml">L'annuaire de services web</lien>
</racine>
```

Page web résultat :



Figure 11 : Page d'accueil du site de la DRDR

Description de l'UM sur la production de canne à sucre dans la région de Saint-Louis

Définition de l'UM :

```

<um id="campagnes" presentation=".\\resultats\\campagnes.xsl">
  <titre>production de cannes a sucre dans la region de Saint-Louis</titre>
  <um id="u5" type="xobjet">
    <vue_filtre id="campagnes_xobjet" type="campagne"
      definition="source_globale/production_de_cannes_a_sucre_dans_la_region_de_saint_louis_en_tonnes/ca
      mpagnes/campagne">
      <annee />
      <sup_totales />
      <sup_sous_canne />
      <sup_recoltees />
      <production_cannes />
      <rendt />
      <qte_estimee_melasse />
    </vue_filtre>
  </um>
  <lien id=".\\resultats\\page_accueilres.xml">Retour a la page d'accueil</lien>
</um>

```

Code XSLT généré pour la vue du Xobjet

```

<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
  <xsl:output method="xml" encoding="ISO-8859-1" />
  <xsl:template match="/">
    <xobjet>

```

```

<xsl:for-each
select="source_globale/production_de_cannes_a_sucre_dans_la_region_de_saint_louis_en_tonnes/campagnes/campagne">
  <campagne>
    <annee>
      <xsl:apply-templates select="annee" />
    </annee>
    <sup_totales>
      <xsl:apply-templates select="sup_totales" />
    </sup_totales>
    <sup_sous_canne>
      <xsl:apply-templates select="sup_sous_canne" />
    </sup_sous_canne>
    <sup_recoltees>
      <xsl:apply-templates select="sup_recoltees" />
    </sup_recoltees>
    <production_cannes>
      <xsl:apply-templates select="production_cannes" />
    </production_cannes>
  <rendt>
    <xsl:apply-templates select="rendt" />
  </rendt>
  <qte_estimee_melasse>
    <xsl:apply-templates select="qte_estimee_melasse" />
  </qte_estimee_melasse>
</campagne>
</xsl:for-each>
</xobjet>
</xsl:template>
</xsl:stylesheet>

```

l'UM générée:

```

<?xmlstylesheet type="text/xsl" href=".\\resultats\\campagnes.xsl" ?>
<racine>
  <titre>production de cannes a sucre dans la region de Saint-Louis</titre>
  <xobjet>
    <campagne>
      <annee>1992/1993</annee>
      <sup_totales>8124,8 ha</sup_totales>
      <sup_sous_canne>7383,4 ha</sup_sous_canne>
      <sup_recoltees>7351 ha</sup_recoltees>
      <production_cannes>796783 T</production_cannes>
      <rendt>108,4 T/ha</rendt>
      <qte_estimee_melasse>31871,3 T</qte_estimee_melasse>
    </campagne>
    <campagne>
      <annee>1993/1994</annee>
      <sup_totales>8111,4 ha</sup_totales>
      <sup_sous_canne>7490,2 ha</sup_sous_canne>
      <sup_recoltees>7375 ha</sup_recoltees>
      <production_cannes>839250 T</production_cannes>
      <rendt>113,8 T/ha</rendt>
      <qte_estimee_melasse>33570,0 T</qte_estimee_melasse>
    </campagne>
    <campagne>
      <annee>1994/1995</annee>
      <sup_totales>8099,4 ha</sup_totales>
      <sup_sous_canne>7531,1 ha</sup_sous_canne>
      <sup_recoltees>7054 ha</sup_recoltees>
    </campagne>
  </racine>

```

```
<production_cannes>901971 T</production_cannes>
<rendt>127,9 T/ha</rendt>
<qte_estimee_melasse>36078,8 T</qte_estimee_melasse>
</campagne>
<campagne>
<annee>1995/1996</annee>
<sup_totales>8230,3 ha</sup_totales>
<sup_sous_canne>7465,9 ha</sup_sous_canne>
<sup_recoltees>6542 ha</sup_recoltees>
<production_cannes>826361 T</production_cannes>
<rendt>126,3 T/ha</rendt>
<qte_estimee_melasse>33054,4 T</qte_estimee_melasse>
</campagne>
<campagne>
<annee>1996/1997</annee>
<sup_totales>8228,6 ha</sup_totales>
<sup_sous_canne>7547,1 ha</sup_sous_canne>
<sup_recoltees>7355 ha</sup_recoltees>
<production_cannes>750221 T</production_cannes>
<rendt>102,0 T/ha</rendt>
<qte_estimee_melasse>30008,8 T</qte_estimee_melasse>
</campagne>
<campagne>
<annee>1997/1998</annee>
<sup_totales>8219,8 ha</sup_totales>
<sup_sous_canne>7546,1 ha</sup_sous_canne>
<sup_recoltees>7413 ha</sup_recoltees>
<production_cannes>810614 T</production_cannes>
<rendt>109,4 T/ha</rendt>
<qte_estimee_melasse>32424,6 T</qte_estimee_melasse>
</campagne>
<campagne>
<annee>1998/1999</annee>
<sup_totales>8205,5 ha</sup_totales>
<sup_sous_canne>7406,9 ha</sup_sous_canne>
<sup_recoltees>7244 ha</sup_recoltees>
<production_cannes>836917 T</production_cannes>
<rendt>115,5 T/ha</rendt>
<qte_estimee_melasse>33476,7 T</qte_estimee_melasse>
</campagne>
<campagne>
<annee>1999/2000</annee>
<sup_totales>8204,0 ha</sup_totales>
<sup_sous_canne>7417,9 ha</sup_sous_canne>
<sup_recoltees>7252 ha</sup_recoltees>
<production_cannes>889363 T</production_cannes>
<rendt>122,6 T/ha</rendt>
<qte_estimee_melasse>35574,5 T</qte_estimee_melasse>
</campagne>
<campagne>
<annee>2000/2001</annee>
<sup_totales>8202,3 ha</sup_totales>
<sup_sous_canne>7415,7 ha</sup_sous_canne>
<sup_recoltees>7272 ha</sup_recoltees>
<production_cannes>850409 T</production_cannes>
<rendt>116,9 T/ha</rendt>
<qte_estimee_melasse>34016,4 T</qte_estimee_melasse>
</campagne>
<campagne>
<annee>2001/2002</annee>
<sup_totales>8224,9 ha</sup_totales>
<sup_sous_canne>7440,3 ha</sup_sous_canne>
<sup_recoltees>7295 ha</sup_recoltees>
```

```

<production_cannes>840109 T</production_cannes>
<rendt>115,2 T/ha</rendt>
<qte_estimee_melasse>33604,4 T</qte_estimee_melasse>
</campagne>
</xobjet>
<lien href=".\\resultats\\page_accueilres.xml">Retour a la page d'accueil</lien>
</racine>

```

Page Web résultat :

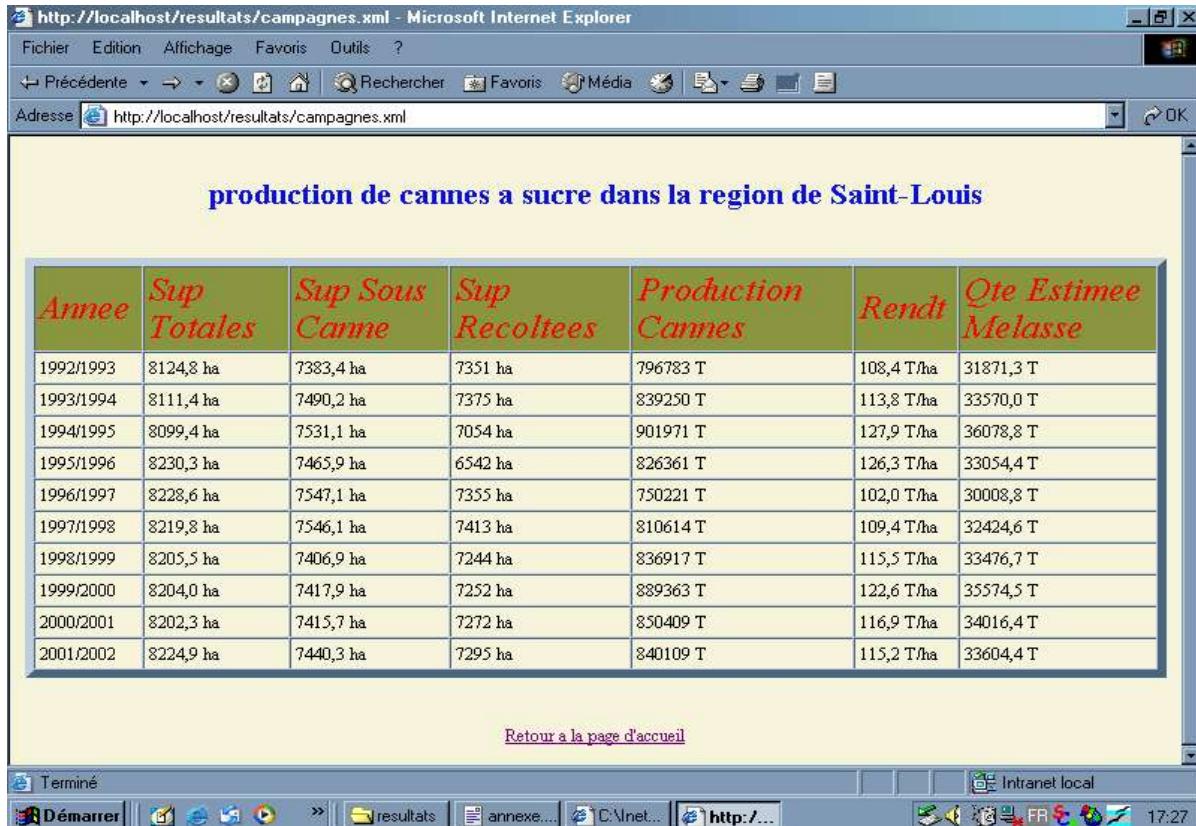


Figure 12 : Production de canne à sucre dans la région de Saint-Louis

Description de l'UM sur la proportion des espèces dominantes par département

Définition de l'UM :

```

<um id="proportion_especes" presentation=".\\resultats\\proportions.xls">
  <titre>proportion des especies dominantes en pourcentage par departement</titre>
  <um id="espece" type="contexte_simple">
    <vue_filtre id="espece_cont" type="proportion" definition="source_globale/proportion_des_especies_dominantes_en_pourcentage/proportions/proportion" presentation=".\\resultats\\proportion.xls">
      <info_index>
        <index>region</index>
        <titre>proportion des especies</titre>
      </info_index>
      <aphis_craccivora />
      <amsacta_moloneyi />
      <pacnoda />
      <oedaleus_senegalensis />
      <borers />
    </vue_filtre>
  </um>
</um>

```

Code XSLT généré pour le contexte navigationnel simple

```

<xsl:stylesheet
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
version="1.0">
<xsl:output method="xml" encoding="ISO-8859-1" />
<xsl:template match="/">
<xobjets>
<xsl:for-each
select="source_globale/proportion_des_especes_dominant
es_en_pourcentage/proportions/proportion">
<proportion>
<region>
<xsl:apply-templates select="region" />
</region>
<aphis_craccivora>
<xsl:apply-templates select="aphis_craccivora" />
</aphis_craccivora>
<amsacta_moloneyi>
<xsl:apply-templates select="amsacta_moloneyi" />
</amsacta_moloneyi>
<pacnoda>
<xsl:apply-templates select="pacnoda" />
</pacnoda>
<oedaleus_senegalensis>
<xsl:apply-templates select="oedaleus_senegalensis" />
</oedaleus_senegalensis>
<borers>
<xsl:apply-templates select="borers" />
</borers>
</proportion>
</xsl:for-each>
</xobjets>
</xsl:template>
</xsl:stylesheet>

```

Index généré :

```

<?xml-stylesheet type="text/xsl"
href=".\\resultats\\proportions.xsl" ?>
<racine>
<titre>proportion des especes dominantes en pourcentage
par departement</titre>
<index>
<region>Dagana</region>
<region>Podor</region>
<region>Saint-Louis</region>
<region>Matam</region>
</index>
</racine>

```

Index résultat (page contenant des liens qui mènent vers plusieurs Xobjets) :

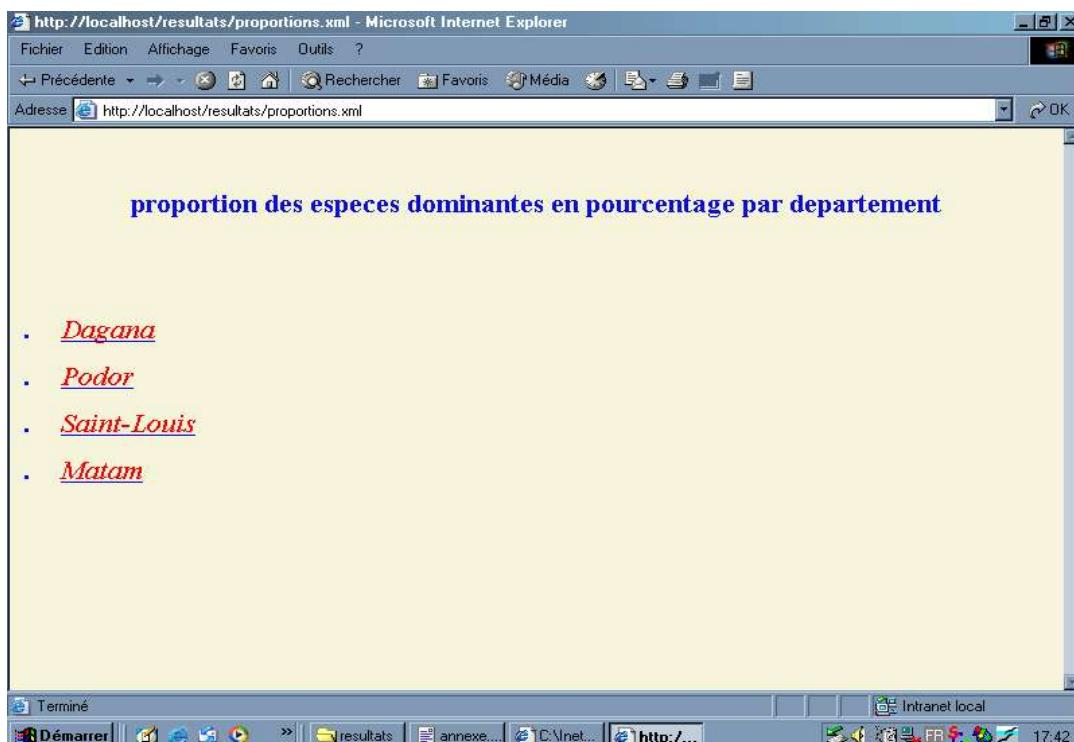


Figure 13 : Proportion des espèces dominantes en pourcentage par département

Un des Xobjets générés pour le contexte navigationnel et la page Web correspondante :

l'UM générée :

```
<?xmlstylesheet type="text/xsl" href=".\\resultats\\proportion.xsl" ?>
<proportion>
<region>Dagana</region>
<aphis_craccivora>50</aphis_craccivora>
<amsacta_moloneyi>30</amsacta_moloneyi>
<pacnoda>5</pacnoda>
<oedaleus_senegalensis>10</oedaleus_senegalensis>
<borders>5</borders>
</proportion>
```

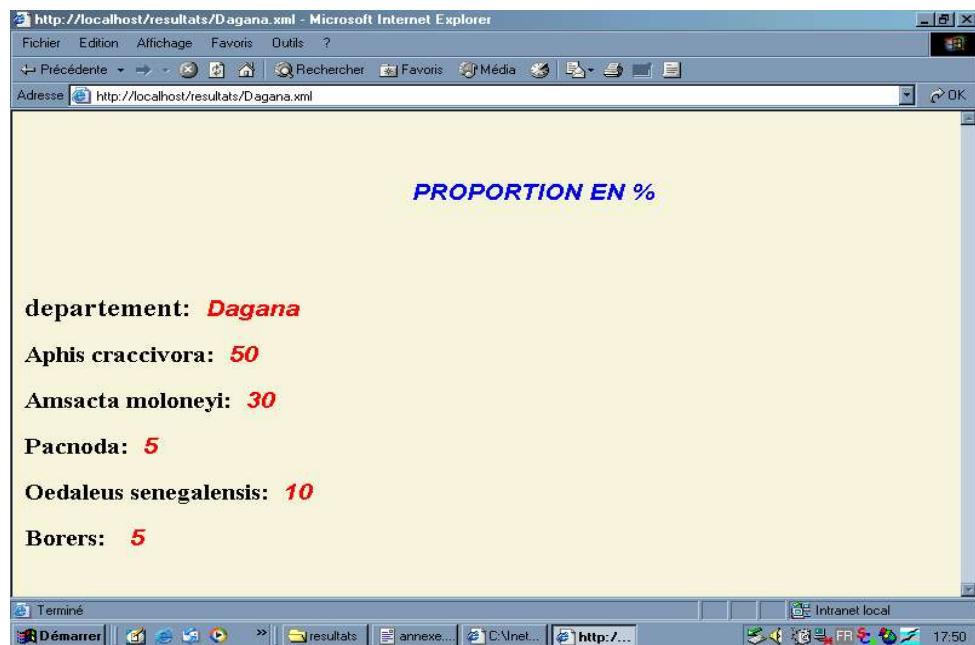


Figure 14 : Proportion en pourcentage

2.Génération de la base médiatique

Nous étudions dans cette partie : la génération de base(s) médiatique(s) par le biais de l'application nommée GenererPage.java.

L'application en recevant comme paramètres d'entrées: une unité médiatique de type page ou xobjet en général, un catalogue de métadonnées et une base de documents XML (source de données) permettra de générer une ou plusieurs base(s) médiatique(s).

Remarque:

Il faudra télécharger le fichier xalan-j_2_7_0.zip à l'URL <http://xml.apache.org> et de le dézipper dans un répertoire du système; en n'oubliant point d'ajouter le JAVA_HOME dans le PATH et les fichiers .jar du répertoire ainsi obtenu dans la variable définissant le CLASSPATH.

La classe GénérerPage.java est développée avec la dernière version de java c'est-à-dire le JDK1.5; donc il est nécessaire de l'installer pour pouvoir exécuter l'application.

Avant d'exécuter l'application, il faudra d'abord compiler la classe GenererPage.java en tapant sur l'invite de commandes l'expression suivante à partir du répertoire contenant l'application :

```
>javac GenererPage.java
```



Figure 15 : Interface de compilation de l'application

Si la compilation se passe bien, alors un byte code nommé GenererPage.class serait généré. Maintenant pour exécuter ce byte code il suffit seulement de taper sur l'invite de commandes l'expression:

```
>java GenererPage
```

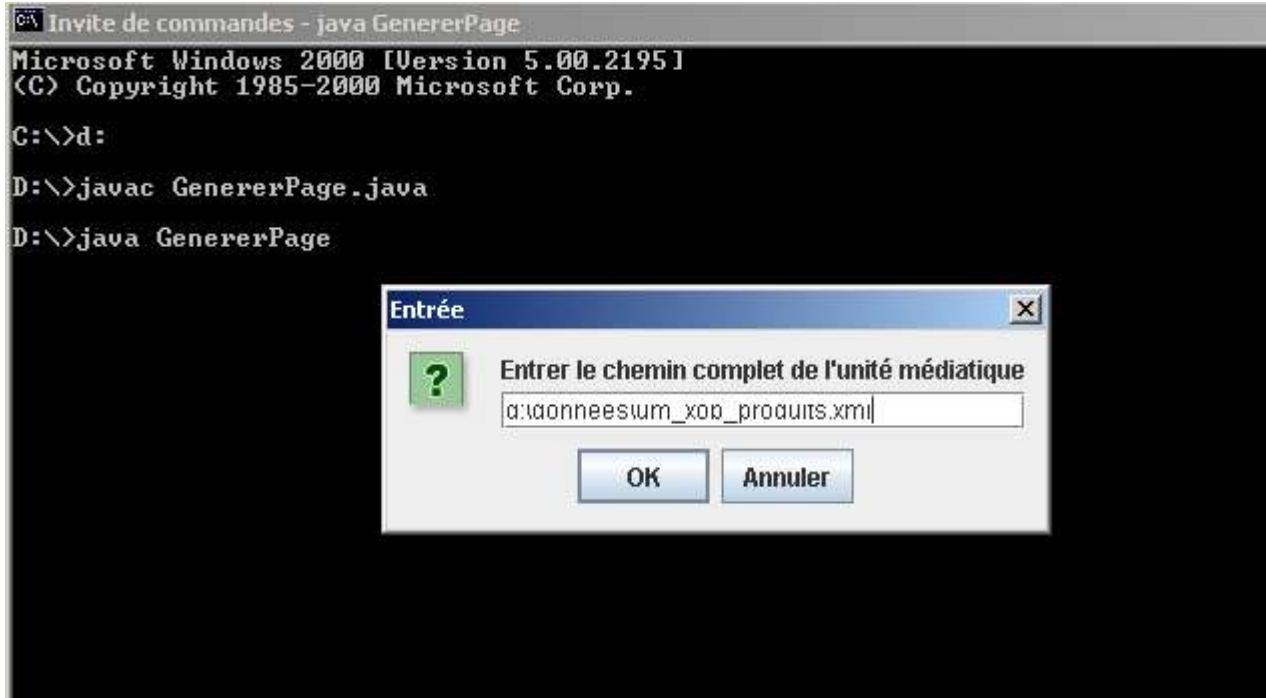


Figure 16 : Interface d'entrée d'une UM du modèle médiatique

L'application nous demande d'entrer le chemin complet de l'unité médiatique définie dans le modèle médiatique et de cliquer sur OK pour continuer ou bien de cliquer sur Annuler pour abandonner.

Si nous donnons le chemin complet de l'UM et cliquons sur OK , alors nous obtiendrions la fenêtre suivante:

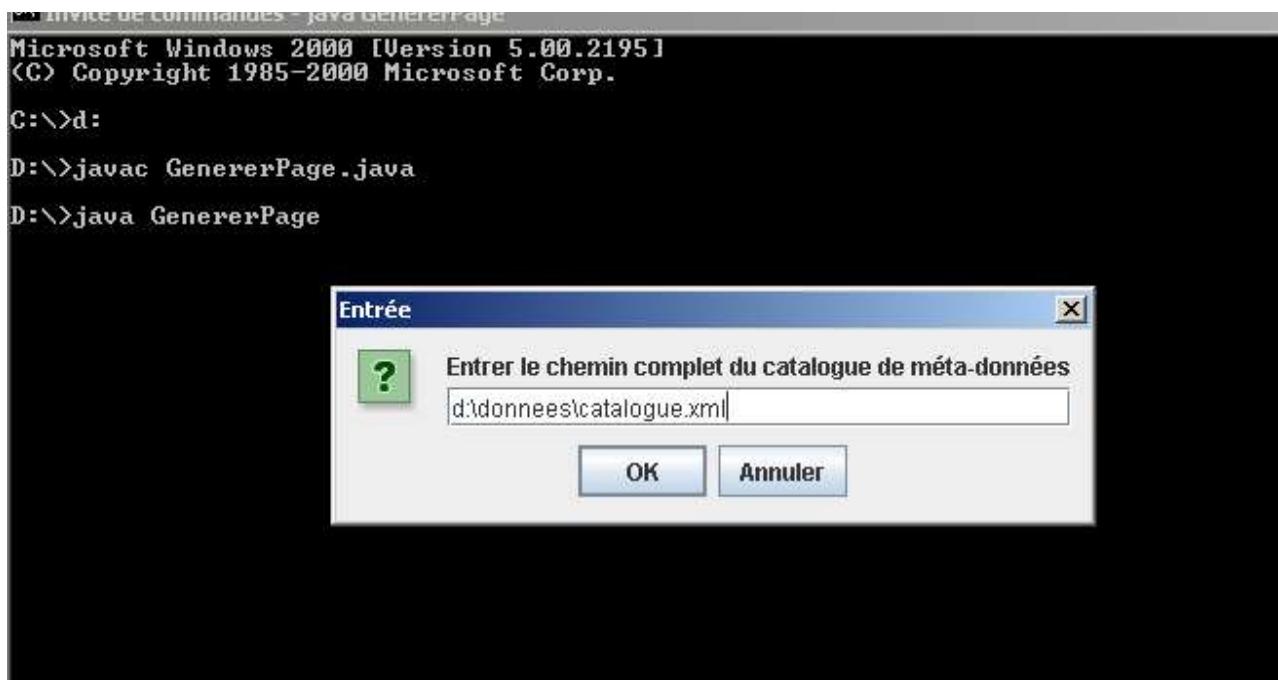


Figure 17 : Interface d'entrée du catalogue de métadonnées

En donnant le chemin complet du catalogue de métadonnées et en cliquant sur Ok, on obtient :

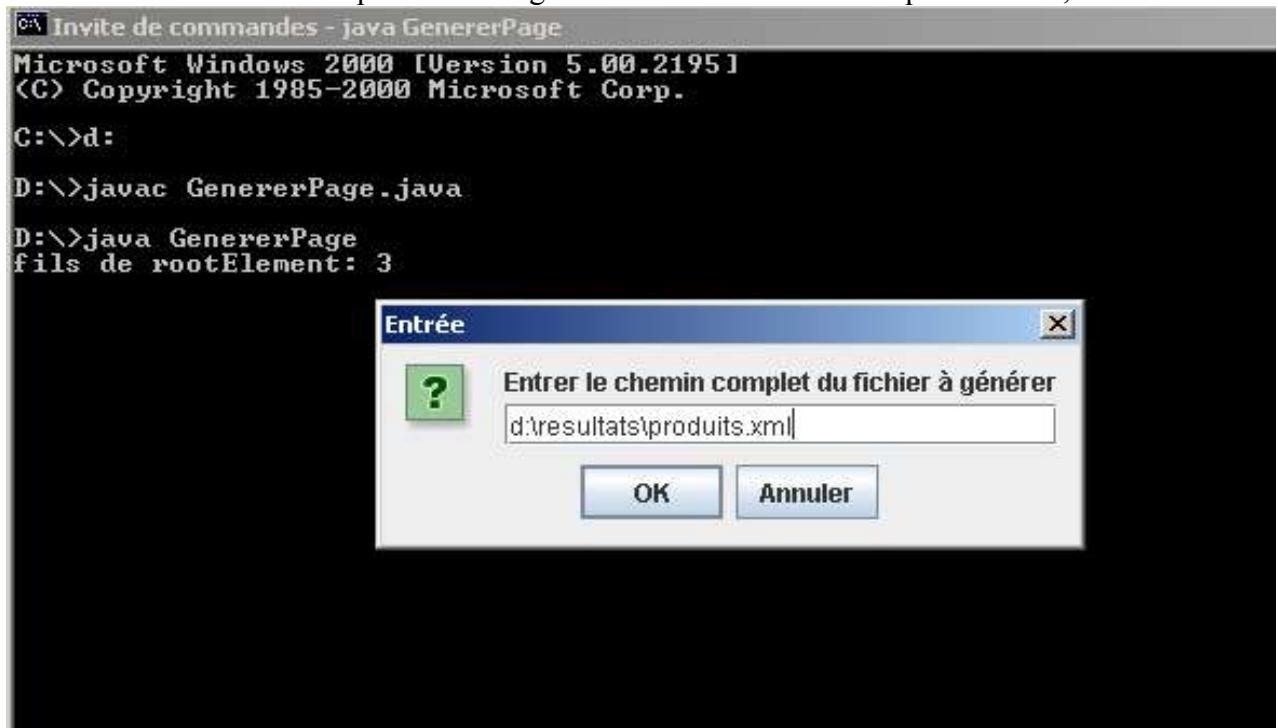


Figure 18 : Interface d'entrée d'une UM à générer dans une base médiatique

En donnant le nom du fichier à générer et en cliquant sur OK, on obtient :

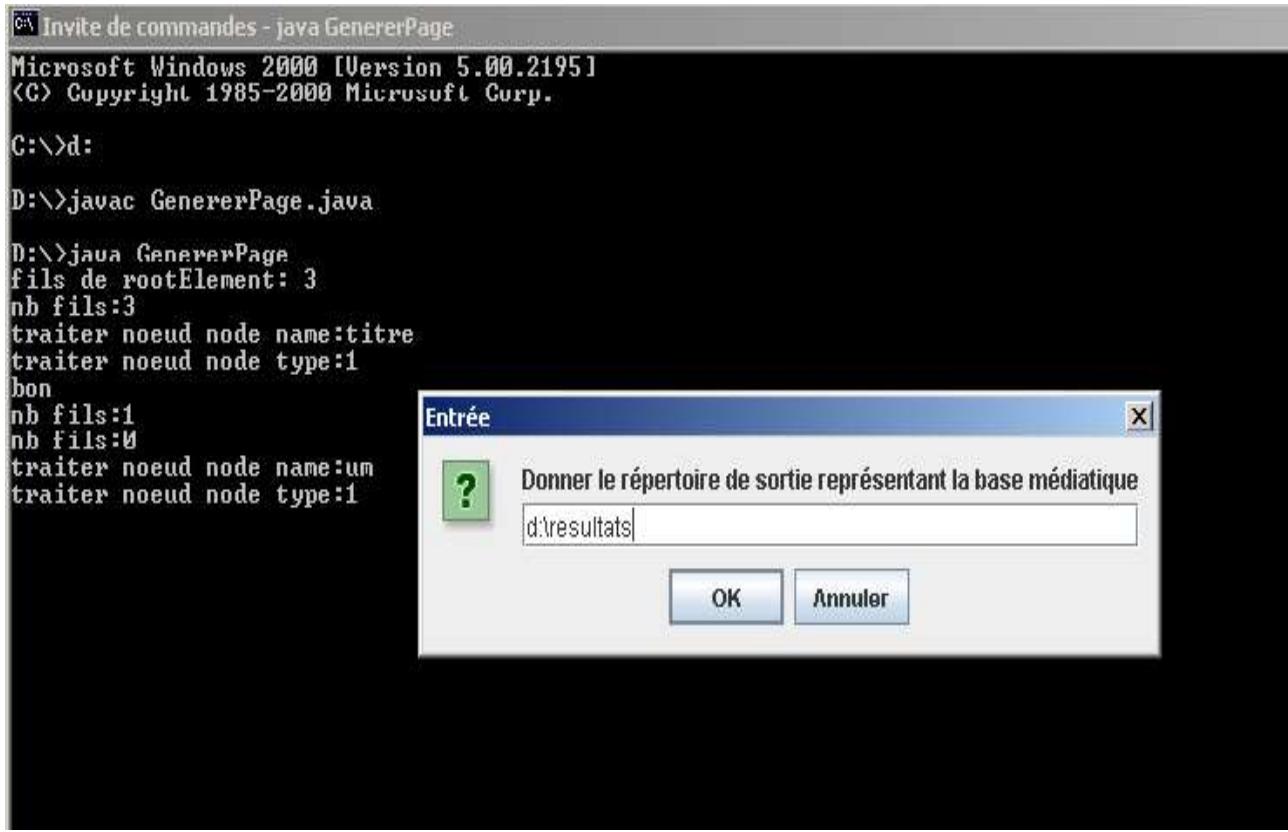


Figure 19 : Interface d'entrée d'un répertoire représentant une base médiatique

En donnant le répertoire de sortie (base médiatique) et cliquant sur OK, on obtient :

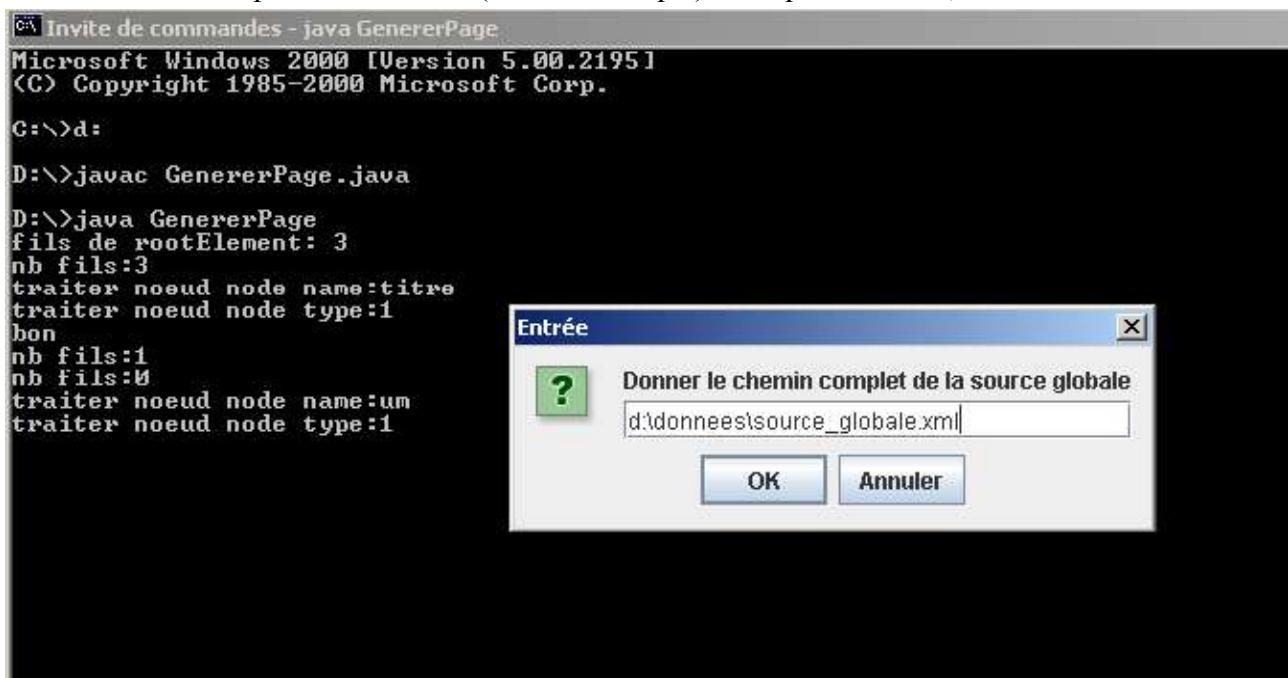
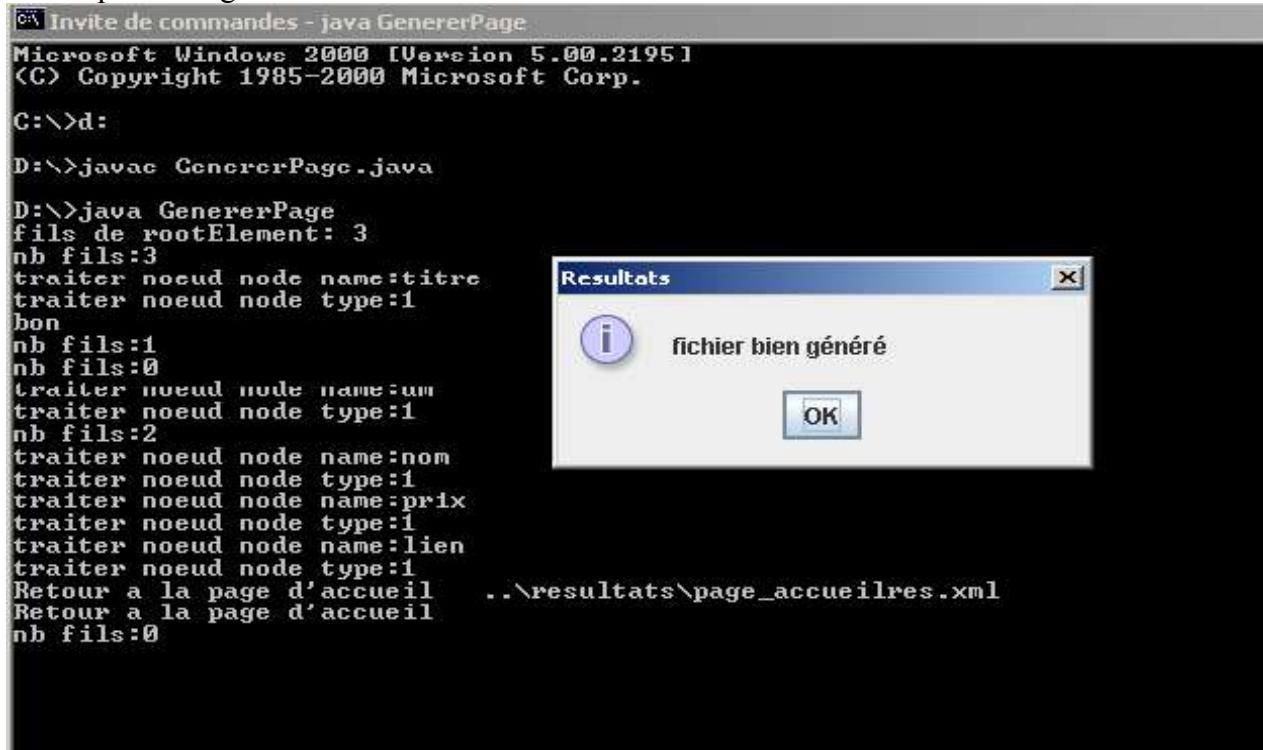


Figure 20 : Interface d'entrée de la source globale

Si nous donnons le chemin complet de la source globale et cliquons sur OK, alors une base médiatique serait générée et nous obtiendrions l'information suivante :



```

C:\ Invite de commandes - java GenererPage
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

C:>>d:
D:>>javac GenererPage.java

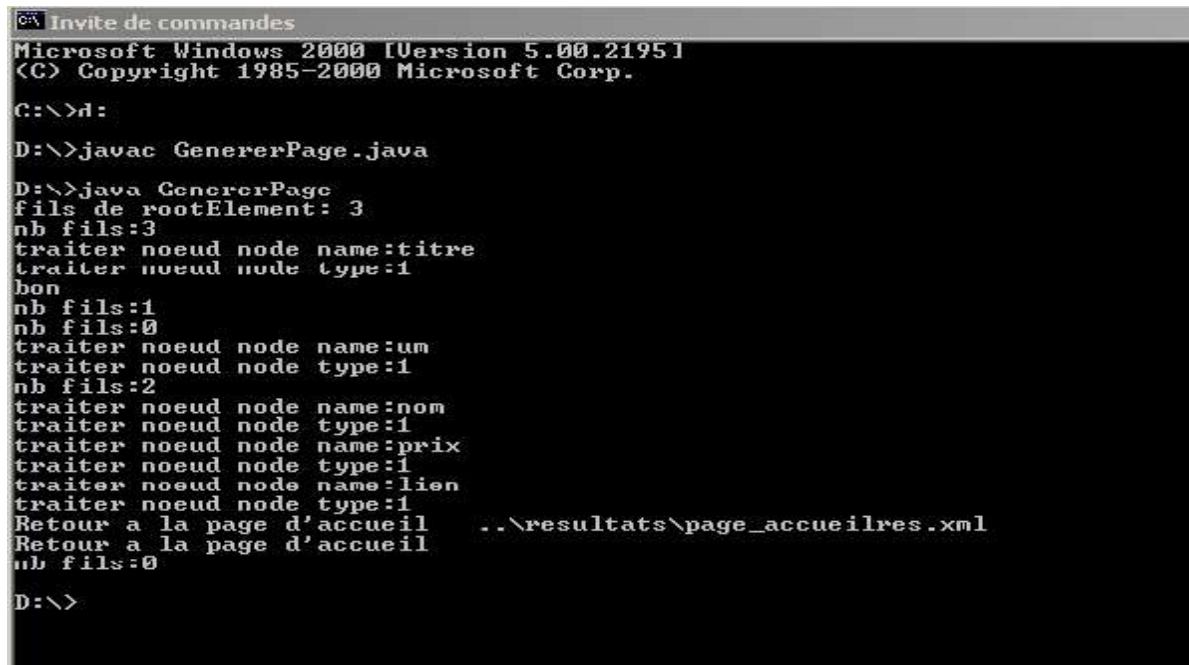
D:>>java GenererPage
fils de rootElement: 3
nb fils:3
traiter noeud node name:titre
traiter noeud node type:1
bon
nb fils:1
nb fils:0
traiter noeud node name:um
traiter noeud node type:1
nb fils:2
traiter noeud node name:nom
traiter noeud node type:1
traiter noeud node name:prix
traiter noeud node type:1
traiter noeud node name:lien
traiter noeud node type:1
Retour a la page d'accueil  ..\resultats\page_accueilres.xml
Retour a la page d'accueil
nb fils:0

```

A small 'Resultats' dialog box is overlaid on the command prompt window, containing the message "fichier bien généré" (file generated successfully) with an information icon and an "OK" button.

Figur 21 : Interface d'information sur le resultat de la génération d'une UM

En cliquant sur OK on met fin à l'exécution de l'application et on obtient enfin la fenêtre suivante



```

C:\ Invite de commandes
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

C:>>d:
D:>>javac GenererPage.java

D:>>java GenererPage
fils de rootElement: 3
nb fils:3
traiter noeud node name:titre
traiter noeud node type:1
bon
nb fils:1
nb fils:0
traiter noeud node name:um
traiter noeud node type:1
nb fils:2
traiter noeud node name:nom
traiter noeud node type:1
traiter noeud node name:prix
traiter noeud node type:1
traiter noeud node name:lien
traiter noeud node type:1
Retour a la page d'accueil  ..\resultats\page_accueilres.xml
Retour a la page d'accueil
nb fils:0

```

Figure 22 : Interface de la fin de l'exécution de l'application

La base médiatique ([D:\resultats](#)) contient les fichiers produits.xml et vue_xobjet.xsl qui sont générés par l'application GenererPage.java

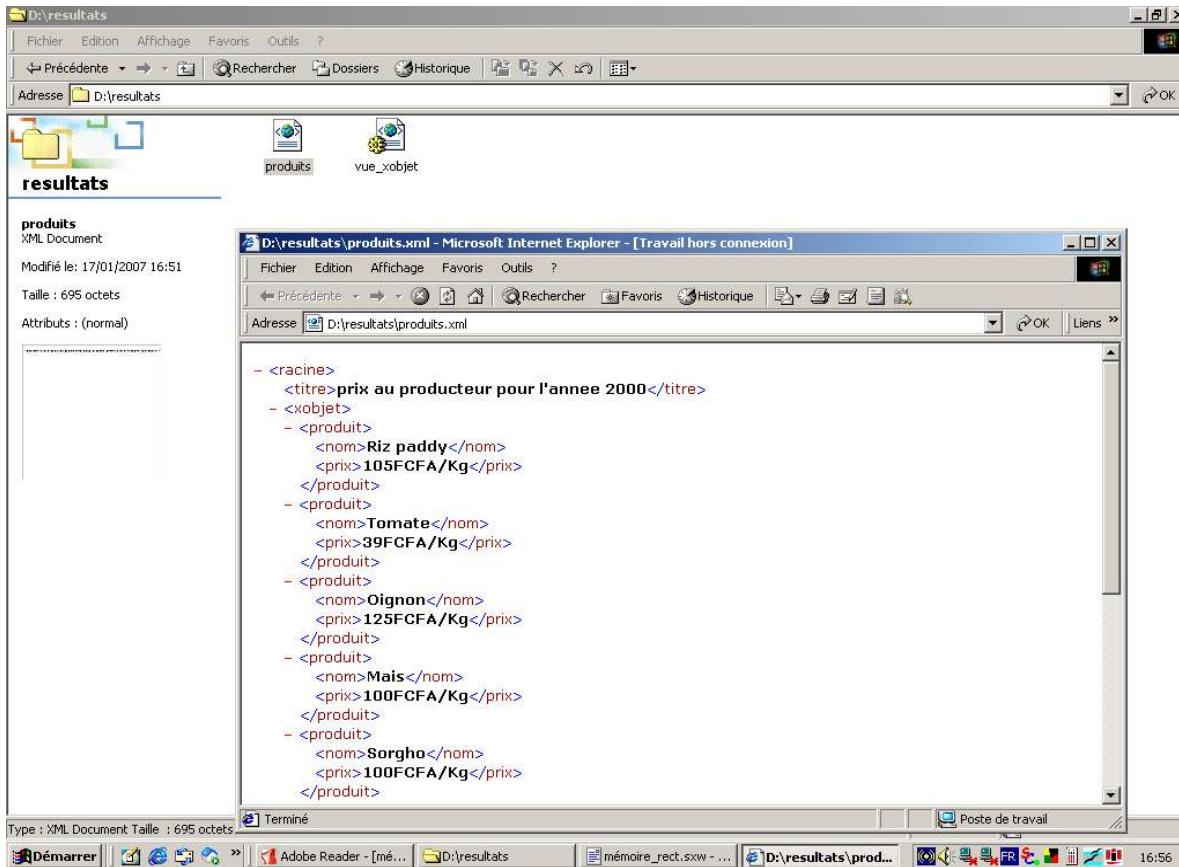


Figure 23 : Interface de consultation de la base médiatique

3. Accès à un service web

Nous montrons ici comment utiliser Xweb pour permettre l'accès à un service web.

3.1 Description d'une UM de type Service Web

Définition de l' UM :

```

<um id="sw" presentation=".\\resultats\\service_web_lani.xsl">
  <um id="calcul" type="service_web">
    <nom>CalculService</nom>
    <fournisseur>LANI</fournisseur>
    <fonctionnalites>
      <fonction>somme</fonction>
      <fonction>produit</fonction>
    </fonctionnalites>
    <adresse_wsdl>http://localhost:8080/axis/calcul.jws?wsdl</adresse_wsdl>
  </um>
  <lien id=".\\resultats\\client.html">Exemple d'un client utilisant la fonction somme du service Calcul</lien>
</um>

```

Index résultat et la page Web correspondante :

```
<?xmlstylesheet type="text/xsl" href=".\\resultats\\uddi.xsl" ?>
```

```
_<racine>
<titre>Liste des services web</titre>
<nom>CalculService</nom>
```

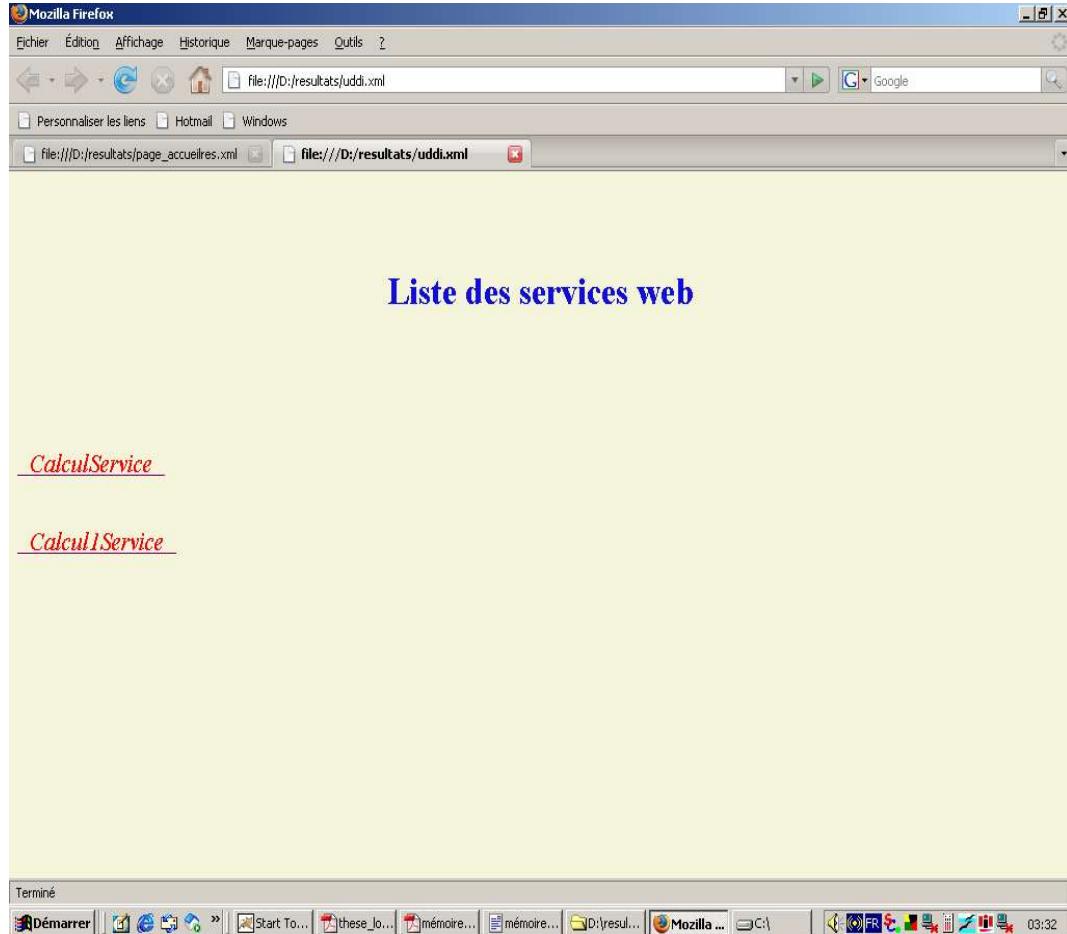


Figure 24 : L'annuaire de services web

l'UM générée et la page Web correspondante :

```
<?xmlstylesheet type="text/xsl" href=".\\resultats\\service_web_lani.xsl"?>
<racine>
<nom>CalculService</nom>
<fournisseur>LANI</fournisseur>
<fonctionnalites>
<fonction>somme</fonction>
<fonction>produit</fonction>
</fonctionnalites>
<adresse_wsdl>http://localhost:8080/axis/calcul.jws?wsdl</adresse_wsdl>
<lien href=".\\resultats\\client.html">Exemple d'un client utilisant la fonction somme du service Calcul</lien>
</racine>
```

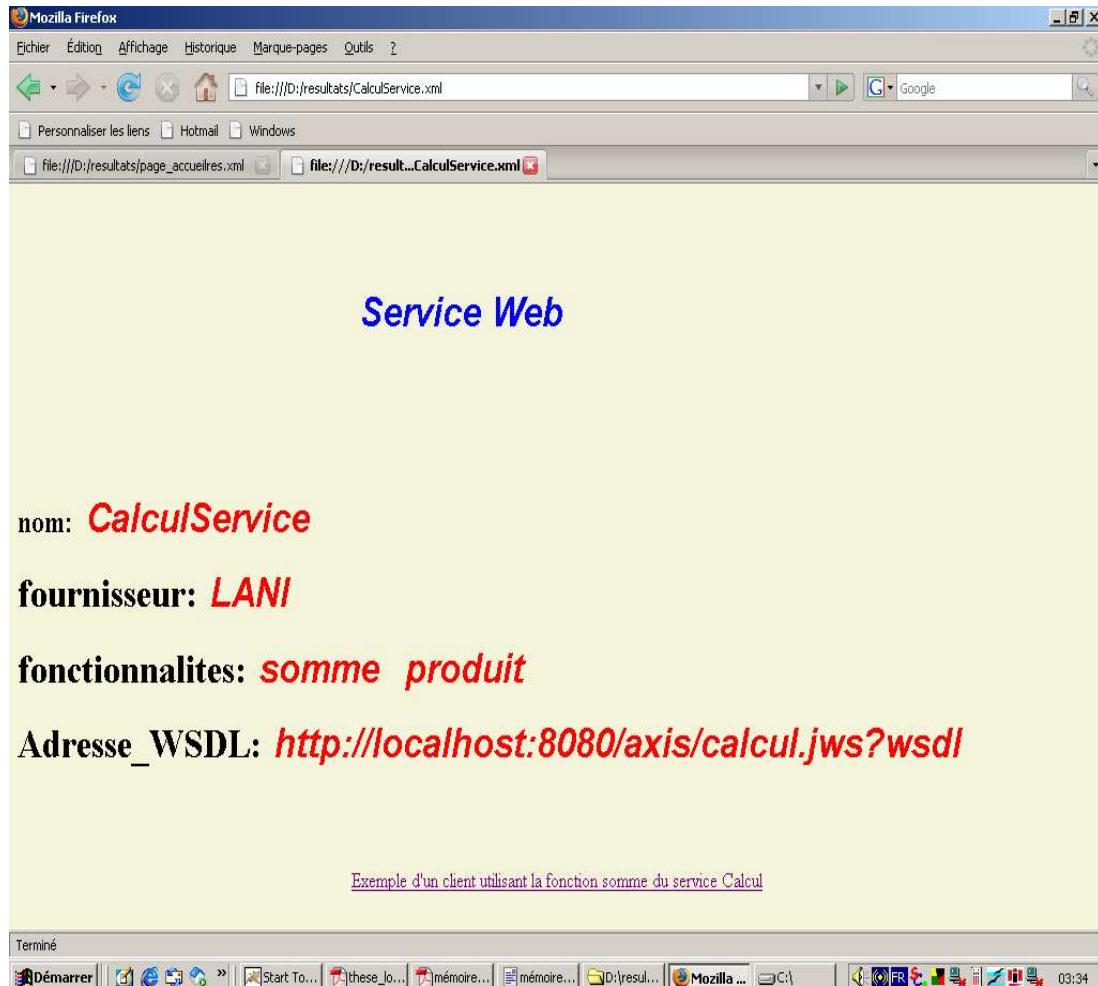


Figure 25 : Description du service web CalculService

Un click sur le lien « Exemple d'un client utilisant la fonction somme du service Calcul » permet au client d'invoquer la méthode « int add(int ,int)» du service web nommé CalculService.

3.2.Déploiement automatique d'une classe java

Axis propose deux méthodes pour développer et déployer un service web :

- le déploiement automatique d'une classe java
- l'utilisation d'un fichier WSDD.

La première méthode permet de déployer facilement et automatiquement une classe java en tant que service web. Il suffit simplement d'écrire la classe, de remplacer l'extension .java en .jws (java web service) et de copier le fichier dans le répertoire axis.

Considérons la classe Java suivante représentant le service web nommé CalculService :

```
public class Calcul {
    public int add(int a, int b){
        return a+b;
    }
}
```

```

        public int mult(int a, int b) {
            return a*b;
        }
    }
}

```

On peut vérifier que le service web correspondant est bien déployé en tapant l'URL :
<http://localhost:8080/axis/Calcul.jws>

3.3 Utilisation d'un service web par un client

Pour faciliter l'utilisation d'un service web, Axis propose l'outil WSDL2Java qui génère automatiquement à partir d'un document WSDL des classes qui encapsulent l'appel à un service web. Grâce à ces classes, l'appel d'un service web par un client ne nécessite que quelques lignes de code.

L'utilisation de l'outil WSDL2Java nécessite une url vers le document WSDL qui décrit le service web. Il génère à partir de ce fichier plusieurs classes dans le package localhost. Ces classes sont utilisées dans le client pour appeler le service web.

C:\Program Files\Apache Group\Tomcat 4.1\webapps\axis\WEB-INF\classes>java
org.apache.axis.wsdl.WSDL2Java <http://localhost:8080/axis/Calcul.jws?wsdl>

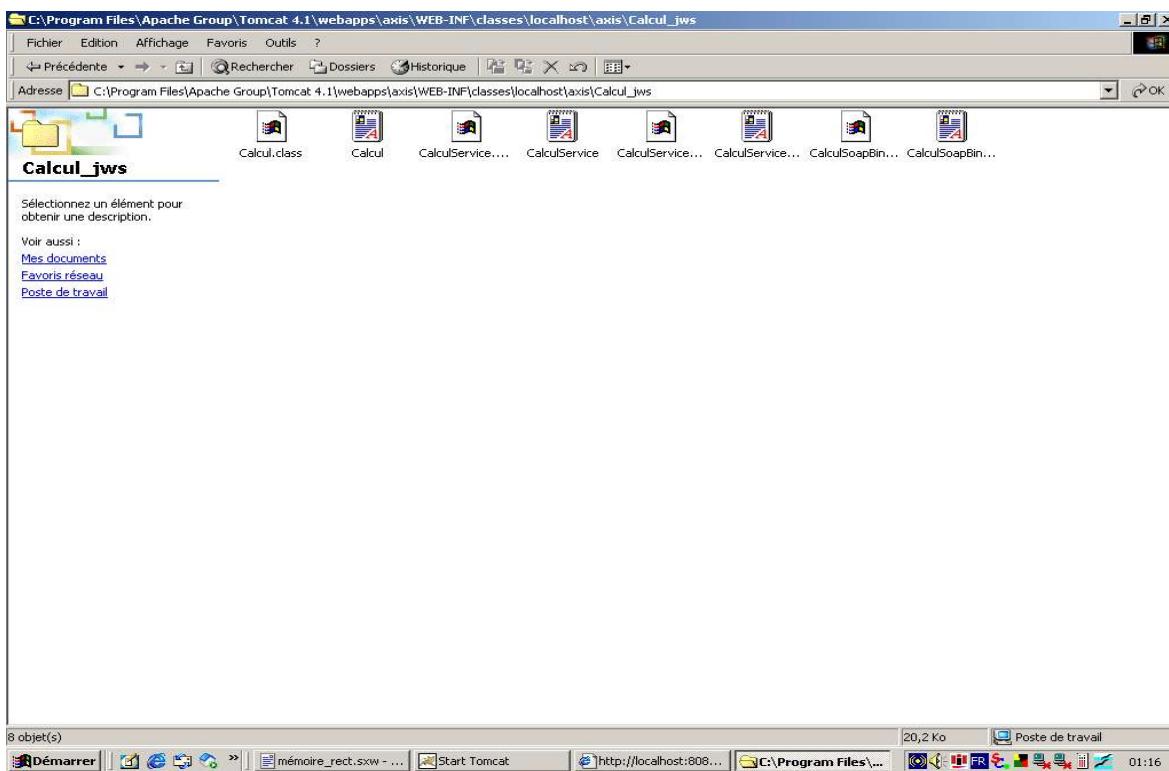


Figure 26 : Interface de consultation du proxy

On utilise les classes générées pour appeler le service web.

Le code suivant nommé « CalculClient.jsp » représente le client invoquant la méthode « int add(int , int) » du service « CalculService » :

```

<html>
<head>
<title>service web Calcul</title>
</head>
<body bgcolor="#f5f5dc">
<br>
<h1><center><span style="font-family:Arial;color:blue;margin-left:10px;font-style:italic">Resultat</span></center></h1>
<br>
<br>
<%@ page language="java"
import="localhost.axis.Calcul_jws.*;java.lang.*"%>
<%
try {
    CalculService locator=new CalculServiceLocator();
    Calcul monservice=locator.getCalcul();
    String input0=request.getParameter("valeur de a");
    String input1=request.getParameter("valeur de b");
    int a=Integer.parseInt(input0);
    int b=Integer.parseInt(input1);
    int s=monservice.add(a,b);
    %>
    <h1><span style="font-style:italic;font-family:Arial;font-size:12pt;background-color:teal;color:red;padding:12px">la somme vaut :
    <span> <%=s%> </span>
    <%
}
catch(Exception e) { }
%>
</body>
</html>

```

Invocation dynamique du service web nommé CalculService à partir de l'U.M. Générée de type « service web »

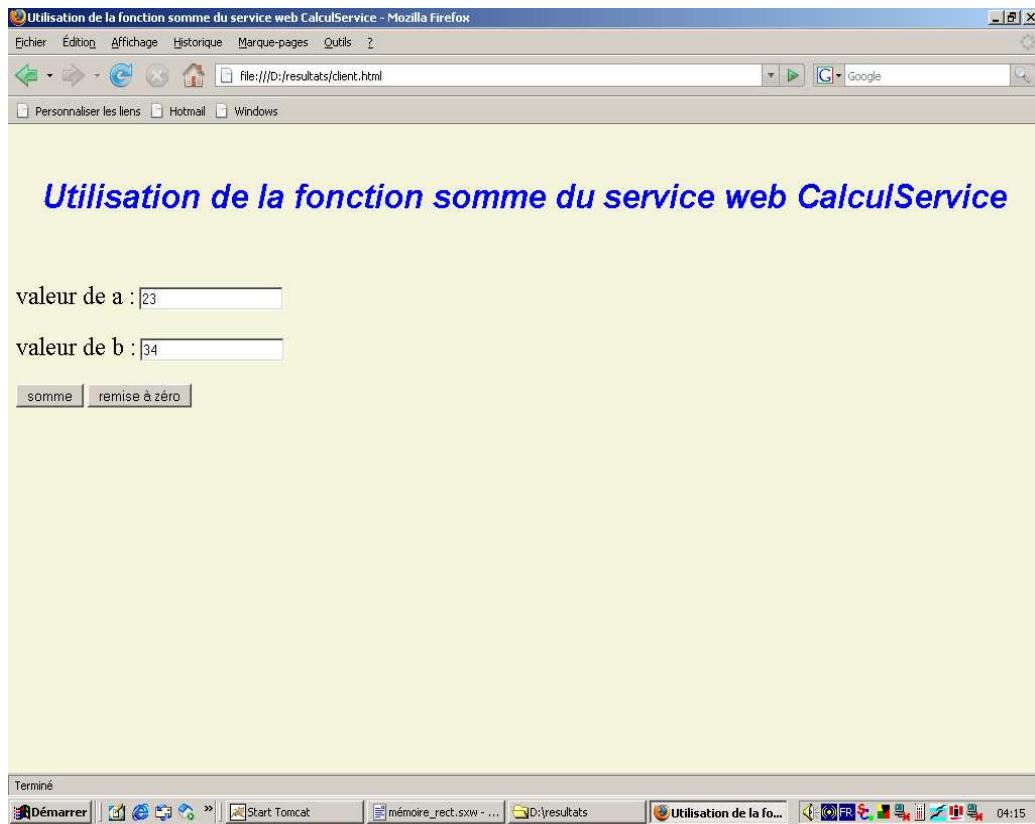


Figure 27 : Invocation de la méthode « add » du service nommé CalculService

Lorsque le client clique sur le bouton « somme », il obtient le résultat attendu

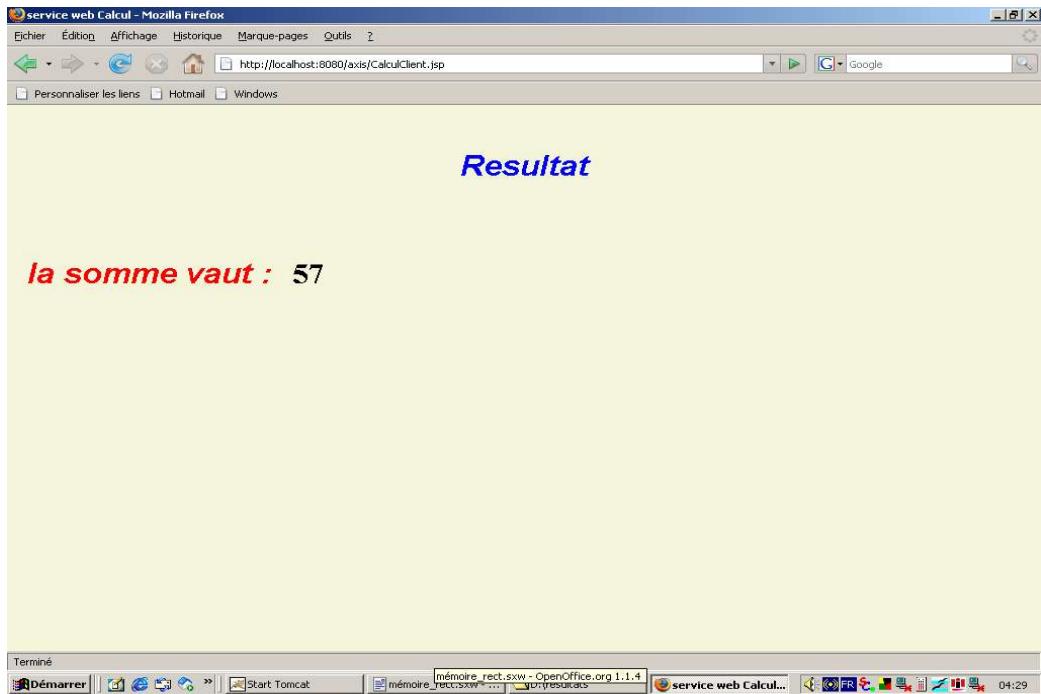


Figure 28 : Résultat de l'exécution de CalculService

Voici un exemple d'implémentation d'un client ayant consulté le site de la DRDR que nous avons réalisé à partir de Xweb étendu.

```
import localhost.axis.Calcul_jws.*;  
  
public class CalculClient {  
  
    public static void main(String args[]) throws Exception{  
  
        CalculService locator = new CalculServiceLocator();  
  
        Calcul monService = locator.getCalcul();  
  
        int s = monService.add(2,4);  
  
        System.out.println("resultat du service web :" +s);  
    }  
  
}
```

Compiler ...

```
C:\Program Files\Apache Group\Tomcat 4.1\webapps\axis\WEB-INF\classes>javac  
CalculClient.java
```

```
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

C:\>cd C:\Program Files\Apache Group\Tomcat 4.1\webapps\axis\WEB-INF\classes

C:\Program Files\Apache Group\Tomcat 4.1\webapps\axis\WEB-INF\classes>javac CalculClient.java

C:\Program Files\Apache Group\Tomcat 4.1\webapps\axis\WEB-INF\classes>
```

Figure 29 : Interface de compilation d'un client de services web

Exécuter ...

```
C:\Program Files\Apache Group\Tomcat 4.1\webapps\axis\WEB-INF\classes>java
CalculClient

resultat du service web :6
```

```
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

C:\>cd C:\Program Files\Apache Group\Toncat 4.1\webapps\axis\WEB-INF\classes

C:\Program Files\Apache Group\Toncat 4.1\webapps\axis\WEB-INF\classes>javac CalculClient.java

C:\Program Files\Apache Group\Toncat 4.1\webapps\axis\WEB-INF\classes>java CalculClient
resultat du service web :6

C:\Program Files\Apache Group\Toncat 4.1\webapps\axis\WEB-INF\classes>
```

Figure 30 : Interface d'exécution du client

CONCLUSION :

Nous avons présenté le modèle de conception d'applications web dénommé Xweb proposé par Lo et Hocine[19].

Cette approche fondée sur les trois modèles : structurel, médiatique et de présentation permet d'effectuer une séparation nette et distincte entre le contenu du site (données) et la présentation durant le processus de conception du site.

Elle se distingue notamment des autres systèmes :

- d'une part par l'utilisation du standard XML pour représenter les données du site, aussi bien durant sa conception (dans une base de documents XML issu du modèle structurel) que durant son exploitation (dans une base médiatique issue du modèle médiatique),
- d'autre part par les possibilités qu'elle offre pour intégrer par la suite un système de recherche d'informations pertinentes.

Il est à remarquer que pour les systèmes Strudel[7], Araneus[11] et WebML[4], les données du site Web sont représentées en HTML à la fin du processus de conception ; ce qui ne facilite ni la recherche d'informations, ni l'interopérabilité.

Nous avons brièvement étudié les services web et avons proposé une solution permettant de les intégrer dans Xweb en nous appuyant sur une unité médiatique nouvellement créée de type service web.

Nous avons aussi utilisé l'approche Xweb pour réaliser une application web pour la Direction Régionale du Développement Rural de Saint-Louis. Cela a permis de générer un site web avec des données XML préalablement obtenues à partir de fichiers Excel.

Ce stage nous a permis d'approfondir nos connaissances sur les services web, les langages comme Java et XML. Il nous a également permis d'acquérir des connaissances sur les systèmes de diffusion de données sur le Web.

Perspectives

Nos travaux porteront sur :

- a) l'intégration de la vidéo en temps réel dans le modèle Xweb
- b) l'intégration de la voix dans Xweb à l'aide du langage VoiceXML

Nous envisageons d'étendre le modèle Xweb de façon à ce qu'il soit en mesure de traiter de la voix et de la vidéo en temps réel.

Références

Xweb :

- [1] S. Abiteboul, D. Quass, J. McHugh, J. Widom and J. Wiener. The Lorel query language for semi-structured data. *Journal of Digital Libraries*, 1(1): 68-88, April 1997.
- [2] T. Bray, J. Paoli & C. Sperberg-MacQueen: Extensible Markup Language (XML) 1.0, W3C Recommandation disponible at <http://www.w3.org/TR/1998/REC-xml-19980210>.
- [3] R. Bourret, C. Bornhövd, A.P. Buchmann : A Generic Load/Extract Utility for Data Transfer between XML Documents and Relational Databases, Technical Report DVS99-1, Darmstadt University of Technology. December 1999.
- [4] S. Ceri, P. Fraternali & A. Bongio : Web Modeling Language (WebML) : a modeling language for designing Web sites, *WWW Conference*, Amsterdam, May 2000.
- [5] R. Deschamps : Bases de connaissances généralisées : une approche fondée sur un modèle hypertexte expert. *Thèse de Doctorat de l'Université Paul Sabatier*, Toulouse, 1995.
- [6] V. Ellisalde & K. Rousseau-Salet : Modélisation objet et implantation en Java d'une Base Médiatique, *Grand Projet*, DESS IMOI, Mars 2000, Université de Pau.
- [7] M. Fernandez, D. Florescu, A. Levy and D. Suciu: A query language and Processor for a Web-Site Management System; *In SIGMOD Record*, 26(3), September 1997.
- [8] F. Garzotto , L. Mainetti L. & P. Paolini : HDM : A model-based approach to Hypertext application design. *ACM Transactions of Information Systems*, 11(1), 1-26, 1993.
- [9] A. Hocine, M. Lo : Modeling and information retrieval on XML-based dataweb, in *LNCS vol. 1909, Proceedings of First Biennial on Advanced in Information Systems*, Izmir, Turquie, 25-27 octobre 2000.
- [10] T. Isakowitz, E. Stohr & P. Balasubramanian : A methodology for the design of structured hypermedia applications. *Communications of the ACM*, (8)38, 34-44, 1995.
- [11] G. Mecca, P. Atzeni, P. Merialdo, A; Masci, and G. Sindoni: From Databases to Web-Base: The Araneus experience, *Sigmod 98*, 1998.
- [12] S. Muench : PLSXML Utilities and demos. Oracle Technical Whitepaper , March 1999.
- [13] Sun : Les servlets Java, <http://java.sun.com/products/servlet/>
- [14] D. Schwabe and G. Rossi : OOHDM : The object-Oriented Hypermedia Design Model, *Communication of ACM*, August 1995.
- [15] V. Turau, Making legacy data accessible for XML applications, Technical Report, FH Wiesbaden, University of Applied Sciences, 1999.
- [16] W3C : <http://www.w3.org/TR/2000/CR-xsl-20001121/>
- [17] W3C : XSLT, <http://www.w3.org/TR/xslt.html>, Version 1.0, Recommendation 16 November 1999
- [18] W3C: DOM, <http://www.w3.org/TR/1998/REC-DOM-Level-1-19981001>
- [19] M. Lo, A. Hocine, P. Raffinat : *Un modèle de conception d'applications Web basées sur XML*, (version améliorée), Revue d'Information Scientifique et Technique (RIST), vol.11, N° 01, pp. 115-132, 2001.

[20] V. Christophides, S. Cluet, J. Simeon : *On wrapping query languages and efficient XML integration*. Proceedings of ACM SIGMOD Conference on Management of Data, 2000.

[21] K. Niang, C. Ahmadou Kassé: Développement d'un wrapper Excel-XML en Java, Juillet 2006

Services Web :

[22] L. Felipe Cabrera, C. Kurt, D. Box : Introduction à l'architecture de services Web et ses spécifications WS-*, Version 2.0, Octobre 2004

[23] UBR, UDDI : www.uddi.org

[24] C. Tallichet : Services Web XML, Décembre 2002

[25] D. Levy : Coordination de Web Services: Langage de description et plate-formes d'exécution, Septembre 2002

[26] JAXRPC: <http://java.sun.com/xml/jaxrpc>

[27] JAXM: <http://java.sun.com/xml/jaxm>

[28] AXIS: <http://ws.apache.org/axis>

[29] .NET : <msdn.microsoft.com/net>

[30] Bea : www.bea.com

[31] Web Services: <http://fr.sun.com/solutions/webservices/>

[32] J. Michel DOUDOUX : Développons en Java, Version 0.80.2, O7 Novembre 2004

[33] J. Fierstone : Les services Web, SAR5-Novembre 2002

ANNEXE

source_globale.xml

```
<source_globale>
  <prix_au_producteur_an_2000>
    <produits>
      <produit>
        <nom>Riz paddy</nom>
        <prix>105FCFA/Kg</prix>
      </produit>
      <produit>
        <nom>Tomate</nom>
        <prix>39FCFA/Kg</prix>
      </produit>
      <produit>
        <nom>Oignon</nom>
        <prix>125FCFA/Kg</prix>
      </produit>
      <produit>
        <nom>Mais</nom>
        <prix>100FCFA/Kg</prix>
      </produit>
      <produit>
        <nom>Sorgho</nom>
        <prix>100FCFA/Kg</prix>
      </produit>
      <produit>
        <nom>Coton</nom>
        <prix>170FCFA/Kg</prix>
      </produit>
      <produit>
        <nom>Arachide</nom>
        <prix>150FCFA/Kg</prix>
      </produit>
      <produit>
        <nom>Patate</nom>
        <prix>200FCFA/Kg</prix>
      </produit>
      <produit>
        <nom>Gombo</nom>
        <prix>350FCFA/Kg</prix>
      </produit>
    </produits>
  </prix_au_producteur_an_2000>
  <statistiques_exportation_semences_certifiees_riz_UNIS_zone_nord>
    <exportations>
      <exportation>
        <annee>1998</annee>
        <quantite_exportee>207,52T</quantite_exportee>
        <destination>PNUD/Mauritanie</destination>
        <valeur_estimee_devise_fcfa>51880000F.cfa</valeur_estimee_devise_fcfa>
      </exportation>
      <exportation>
        <annee>1999</annee>
```

```
<quantite_exportee>453T</quantite_exportee>
<destination>FAO/Guinee Bissau</destination>
<valeur_estimee_devise_fcfa>113250000F.cfa</valeur_estimee_devise_fcfa>
</exportation>
- <exportation>
  <annee>2001</annee>
  <quantite_exportee>4T</quantite_exportee>
  <destination>Mauritanie</destination>
  <valeur_estimee_devise_fcfa>1000000F.cfa</valeur_estimee_devise_fcfa>
</exportation>
- <exportation>
  <annee>2002</annee>
  <quantite_exportee>240T</quantite_exportee>
  <destination>Mauritanie</destination>
  <valeur_estimee_devise_fcfa>60000000F.cfa</valeur_estimee_devise_fcfa>
</exportation>
</exportations>
</statistiques_exportation_semences_certifiees_riz_UNIS_zone_nord>
<evolution_des_fourchettes_de_prix_des_riziers>
- <evolutions>
- <evolution>
  <annee>1994/95</annee>
  <collecte>31000T</collecte>
  <fourchette_prix_paddy>97,6-115</fourchette_prix_paddy>
  <moyenne>100</moyenne>
  <fourchette_riz_blancl>162-190F.cfa/kg</fourchette_riz_blancl>
  <moyenne>175</moyenne>
</evolution>
- <evolution>
  <annee>1995/96</annee>
  <collecte>23804T</collecte>
  <fourchette_prix_paddy>102-125</fourchette_prix_paddy>
  <moyenne>115</moyenne>
  <fourchette_riz_blancl>165-210F.cfa/kg</fourchette_riz_blancl>
  <moyenne>195</moyenne>
</evolution>
- <evolution>
  <annee>1996/97</annee>
  <collecte>17090T</collecte>
  <fourchette_prix_paddy>95-105</fourchette_prix_paddy>
  <moyenne>100</moyenne>
  <fourchette_riz_blancl>160-200F.cfa/kg</fourchette_riz_blancl>
  <moyenne>175</moyenne>
</evolution>
- <evolution>
  <annee>1997/98</annee>
  <collecte>22538T</collecte>
  <fourchette_prix_paddy>90-105,5</fourchette_prix_paddy>
  <moyenne>100</moyenne>
  <fourchette_riz_blancl>150-190F.cfa/kg</fourchette_riz_blancl>
  <moyenne>185</moyenne>
</evolution>
- <evolution>
  <annee>1998/99</annee>
  <collecte>24625T</collecte>
  <fourchette_prix_paddy>90-105</fourchette_prix_paddy>
  <moyenne>100</moyenne>
  <fourchette_riz_blancl>160-190F.cfa/kg</fourchette_riz_blancl>
  <moyenne>175</moyenne>
</evolution>
- <evolution>
  <annee>1999/00</annee>
  <collecte>21118T</collecte>
```

```
<fourchette_prix_paddy>95-105</fourchette_prix_paddy>
<moyenne>100</moyenne>
<fourchette_riz_blancl>165-180F.cfa/kg</fourchette_riz_blancl>
<moyenne>170</moyenne>
</evolution>
</evolutions>
</evolution_des_fourchettes_de_prix_des_riziers>
<bilan_des_traitements_par_les_CLV_dans_la_region_de_saint-louis_en_2002-2003>
<traitements>
<traitement>
<region>Dagana</region>
<superficie_prospectee>15000 ha</superficie_prospectee>
<superficie_infestee>12200 ha</superficie_infestee>
<superficie_traitee>350 ha</superficie_traitee>
</traitement>
<traitement>
<region>Podor</region>
<superficie_prospectee>16200 ha</superficie_prospectee>
<superficie_infestee>8700 ha</superficie_infestee>
<superficie_traitee>350 ha</superficie_traitee>
</traitement>
<traitement>
<region>Saint-Louis</region>
<superficie_prospectee>12500 ha</superficie_prospectee>
<superficie_infestee>9070 ha</superficie_infestee>
<superficie_traitee>632 ha</superficie_traitee>
</traitement>
</traitements>
</bilan_des_traitements_par_les_CLV_dans_la_region_de_saint-louis_en_2002-2003>
<proportion_des_especes_dominantes_en_pourcentage>
<proportions>
<proportion>
<region>Dagana</region>
<aphis_craccivora>50</aphis_craccivora>
<amsacta_moloneyi>30</amsacta_moloneyi>
<pacnoda>5</pacnoda>
<oedaleus_senegalensis>10</oedaleus_senegalensis>
<borders>5</borders>
</proportion>
<proportion>
<region>Podor</region>
<aphis_craccivora>50</aphis_craccivora>
<amsacta_moloneyi>30</amsacta_moloneyi>
<pacnoda>5</pacnoda>
<oedaleus_senegalensis>10</oedaleus_senegalensis>
<borders>5</borders>
</proportion>
<proportion>
<region>Saint-Louis</region>
<aphis_craccivora>50</aphis_craccivora>
<amsacta_moloneyi>30</amsacta_moloneyi>
<pacnoda>5</pacnoda>
<oedaleus_senegalensis>10</oedaleus_senegalensis>
<borders>5</borders>
</proportion>
<proportion>
<region>Matam</region>
<aphis_craccivora>50</aphis_craccivora>
<amsacta_moloneyi>30</amsacta_moloneyi>
<pacnoda>5</pacnoda>
<oedaleus_senegalensis>10</oedaleus_senegalensis>
<borders>5</borders>
</proportion>
```

```
</proportions>
</proportion_des_especes_dominantes_en_pourcentage>
<statistiques_exportation_du_cheptel_dans_region_de_saint_louis>
  <exportations>
    <exportation>
      <annee>1997</annee>
      <bovins>1</bovins>
      <ovins>151</ovins>
      <caprins>3</caprins>
      <equins />
      <asins>1896</asins>
      <camelins>8</camelins>
    </exportation>
    <exportation>
      <annee>1998</annee>
      <bovins>1</bovins>
      <ovins>151</ovins>
      <caprins>3</caprins>
      <equins />
      <asins>1896</asins>
      <camelins>8</camelins>
    </exportation>
    <exportation>
      <annee>1999</annee>
      <bovins />
      <ovins />
      <caprins />
      <equins>10</equins>
      <asins />
      <camelins>170</camelins>
    </exportation>
    <exportation>
      <annee>2000</annee>
      <bovins>13108</bovins>
      <ovins>31891</ovins>
      <caprins>2865</caprins>
      <equins>6</equins>
      <asins>144</asins>
      <camelins />
    </exportation>
    <exportation>
      <annee>2001</annee>
      <bovins>13108</bovins>
      <ovins>31891</ovins>
      <caprins>2865</caprins>
      <equins>6</equins>
      <asins>144</asins>
      <camelins />
    </exportation>
    <exportation>
      <annee>2002</annee>
      <bovins />
      <ovins>368</ovins>
      <caprins>547</caprins>
      <equins />
      <asins />
      <camelins />
    </exportation>
  </exportations>
</statistiques_exportation_du_cheptel_dans_region_de_saint_louis>
<production_de_cannes_a_sucre_dans_la_region_de_saint_louis_en_tonnes>
  <campagnes>
    <campagne>
```

```
<annee>1992/1993</annee>
<sup_totales>8124,8 ha</sup_totales>
<sup_sous_canne>7383,4 ha</sup_sous_canne>
<sup_recoltees>7351 ha</sup_recoltees>
<production_cannes>796783 T</production_cannes>
<rendt>108,4 T/ha</rendt>
<qte_estimee_melasse>31871,3 T</qte_estimee_melasse>
</campagne>
<campagne>
<annee>1993/1994</annee>
<sup_totales>8111,4 ha</sup_totales>
<sup_sous_canne>7490,2 ha</sup_sous_canne>
<sup_recoltees>7375 ha</sup_recoltees>
<production_cannes>839250 T</production_cannes>
<rendt>113,8 T/ha</rendt>
<qte_estimee_melasse>33570,0 T</qte_estimee_melasse>
</campagne>
<campagne>
<annee>1994/1995</annee>
<sup_totales>8099,4 ha</sup_totales>
<sup_sous_canne>7531,1 ha</sup_sous_canne>
<sup_recoltees>7054 ha</sup_recoltees>
<production_cannes>901971 T</production_cannes>
<rendt>127,9 T/ha</rendt>
<qte_estimee_melasse>36078,8 T</qte_estimee_melasse>
</campagne>
<campagne>
<annee>1995/1996</annee>
<sup_totales>8230,3 ha</sup_totales>
<sup_sous_canne>7465,9 ha</sup_sous_canne>
<sup_recoltees>6542 ha</sup_recoltees>
<production_cannes>826361 T</production_cannes>
<rendt>126,3 T/ha</rendt>
<qte_estimee_melasse>33054,4 T</qte_estimee_melasse>
</campagne>
<campagne>
<annee>1996/1997</annee>
<sup_totales>8228,6 ha</sup_totales>
<sup_sous_canne>7547,1 ha</sup_sous_canne>
<sup_recoltees>7355 ha</sup_recoltees>
<production_cannes>750221 T</production_cannes>
<rendt>102,0 T/ha</rendt>
<qte_estimee_melasse>30008,8 T</qte_estimee_melasse>
</campagne>
<campagne>
<annee>1997/1998</annee>
<sup_totales>8219,8 ha</sup_totales>
<sup_sous_canne>7546,1 ha</sup_sous_canne>
<sup_recoltees>7413 ha</sup_recoltees>
<production_cannes>810614 T</production_cannes>
<rendt>109,4 T/ha</rendt>
<qte_estimee_melasse>32424,6 T</qte_estimee_melasse>
</campagne>
<campagne>
<annee>1998/1999</annee>
<sup_totales>8205,5 ha</sup_totales>
<sup_sous_canne>7406,9 ha</sup_sous_canne>
<sup_recoltees>7244 ha</sup_recoltees>
<production_cannes>836917 T</production_cannes>
<rendt>115,5 T/ha</rendt>
<qte_estimee_melasse>33476,7 T</qte_estimee_melasse>
</campagne>
<campagne>
```

```
<annee>1999/2000</annee>
<sup_totales>8204,0 ha</sup_totales>
<sup_sous_canne>7417,9 ha</sup_sous_canne>
<sup_recoltees>7252 ha</sup_recoltees>
<production_cannes>889363 T</production_cannes>
<rendt>122,6 T/ha</rendt>
<qte_estimee_melasse>35574,5 T</qte_estimee_melasse>
</campagne>
<campagne>
<annee>2000/2001</annee>
<sup_totales>8202,3 ha</sup_totales>
<sup_sous_canne>7415,7 ha</sup_sous_canne>
<sup_recoltees>7272 ha</sup_recoltees>
<production_cannes>850409 T</production_cannes>
<rendt>116,9 T/ha</rendt>
<qte_estimee_melasse>34016,4 T</qte_estimee_melasse>
</campagne>
<campagne>
<annee>2001/2002</annee>
<sup_totales>8224,9 ha</sup_totales>
<sup_sous_canne>7440,3 ha</sup_sous_canne>
<sup_recoltees>7295 ha</sup_recoltees>
<production_cannes>840109 T</production_cannes>
<rendt>115,2 T/ha</rendt>
<qte_estimee_melasse>33604,4 T</qte_estimee_melasse>
</campagne>
</campagnes>
</production_de_cannes_a_sucre_dans_la_region_de_saint_louis_en_tonnes>
</source_globale>
```