

Un modèle de conception d'applications Web basées sur XML

Moussa Lo^{1,2}, Amrane Hocine², Patrick Raffinat²

¹Laboratoire LANI - Université Gaston
Berger B.P. 234 – Saint-Louis (Sénégal)

lom@ugb.sn, moussa.lo@univ-pau.fr

²Laboratoire d'Informatique - Université de
Pau B.P. 1155 - 64013 Pau Cedex (France)

{Amrane.Hocine, [Patrick.Raffinat](mailto:Patrick.Raffinat@univ-pau.fr)}@univ-pau.fr

1. 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 dus 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.

Dans ce papier nous nous focalisons sur la présentation des deux modèles principaux (structurel et médiatique), autour desquels s'articule notre 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].

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 nous a amenés tout naturellement à utiliser XML pour la construction de sites web importants.

Une des originalités de notre approche réside dans le fait que le contenu du dataweb 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.

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). Dans l'implémentation de notre prototype, la forme médiatique est obtenue par l'utilisation de feuilles de style (CSS ou XSL [16]) qui sont associées à chaque unité médiatique.

Dans le prototype en cours de développement [6], nous utilisons un éditeur XML pour décrire le modèle médiatique et un programme Java pour générer les unités médiatiques.

Après cette introduction, la section 2 aborde la technique d'intégration des sources de données hétérogènes du dataweb à travers le modèle structurel. Le modèle médiatique et le langage de description des unités médiatiques sont présentés dans la section 3. Dans la section 4 nous proposons un algorithme de génération de la base médiatique. La section 5 résume brièvement quelques travaux similaires au notre avant de conclure sur les apports de notre approche. Nous

étayerons nos propos à l'aide d'un exemple : la conception du site Web de notre groupe de recherche "SI-Web". Une implémentation de cette méthode est en cours de développement sur un serveur Java [13].

2. Le modèle structurel

La démarche de conception du modèle structurel que nous proposons 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) que nous avons 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.

Dans notre exemple, nous avons identifié deux sources de données :

- les informations sur les membres et les publications (dans des conférences) du groupe, modélisées à l'aide du modèle E/A (données structurées) ;
- les thèmes de recherche décrits dans un document texte, formalisés dans un document XML (données semi-structurées).

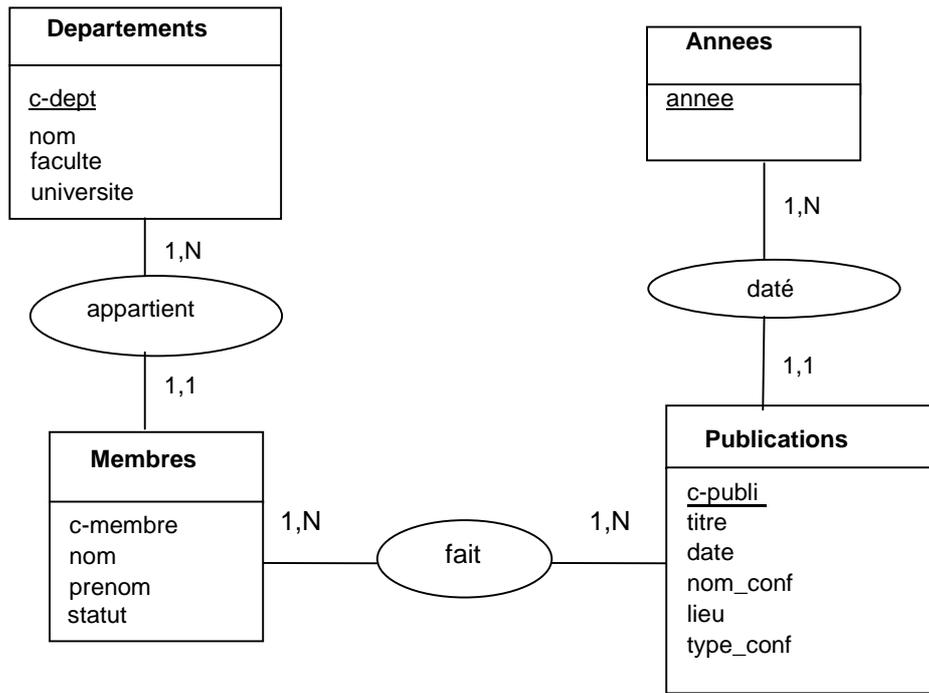


Figure 1 : Diagramme du schéma entité-association de la base de donnée du groupe SI-Web

Le modèle structurel du dataweb est donné ci-dessous :

```
<groupe_si_web>
<depts>
  <dept id="">
    <nom/>
    <faculte/>

    <universite/>
  </dept>
</depts>
<membres>
  <membre id=""
dept="">
    <nom/>
    <prenom/>
    <statut/>
  </membre>
</membres>
```

```
<annees>
  <annee id="" />
</annees>
<publis_conf>
  <publi id=""
membre="" annee="">
    <titre/>
    <nom_conf/>
    <date/>
    <lieu/>

  <type_conf />
</publi>
</publis_conf>
<themes> ...
</themes>
</groupe_si_web>
```

Le problème de la transformation (ou mapping) des données issues de bases de données relationnelles en documents XML est abordé depuis peu de temps et 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.

Nous rappelons ci-dessous l'algorithme général [9] permettant de construire la base de documents XML à partir du schéma relationnel et des données de la base :

- ◆ Création d'un élément racine (<ROOT_ELEMENT>),
- ◆ Chaque relation est transformée en élément parent,
- ◆ Pour chacun des éléments enfants : les différents attributs contenus dans la relation constituent des éléments attributs de ces éléments,
- ◆ La clé primaire devient un attribut de type ID. Cet attribut est spécifié dans la balise de l'élément,
- ◆ Les clés étrangères d'une relation sont transformées en attribut de type IDREF. Sa valeur

apparaît dans la balise principale de l'élément.

- ◆ Les clés apparaissant dans les relations provenant d'associations (n-n) sont transformées en valeurs d'attributs de type IDREFS.

Exemple de données XML :

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

```

<groupe_si_web>
<depts>
  <dept id="info_uppa">
    <nom>Dept
Informatique </nom>
    <faculte>Fac
sciences </faculte>
  </dept>
  <dept id="info_ugb">
    <nom>Section
Informatique </nom>
    <faculte>UFR MAI
</faculte>
  </dept>
  <dept id="stid">
    <nom>Dept STID
</nom>
    <faculte>IUT
</faculte>
  </dept>
</depts>
<membres>
  <membre id="ra">
dept="stid">
    <nom>Raffinat</nom>

    <prenom>Patrick</prenom>
>
    <statut>Maitre de
conferences</statut>
  </membre>
  <membre id="ha">
dept="info_uppa">
    <nom>Hocine</nom>

    <prenom>Amrane</prenom>
    <statut>Maitre de
conferences</statut>
  </membre>
  <membre id="ss">

```

```

dept="info_uppa">
  <nom>Smadhi </nom>

  <prenom>Salaha</prenom>

  <statut>ATER</statut>
</membre>
  <membre id="lm">
dept="info_ugb">
  <nom>Lo</nom>

  <prenom>Moussa</prenom>
>

  <statut>Doctorant</statut>
</membre>
</membres>
...
<publis_conf>
  <publi id="p1">
membre="halms"
annee="2000">
  <titre>Information
retrieval using a base of
concepts and xml</titre>
  <nom_conf>ISK06</nom_co
nf>
  <date> juillet
2000</date>
  <lieu>Toronto</lieu>
</publi>
  <publi id="p2">
membre="lmha"
annee="2000">
  <titre>Modelisation
de dataweb : une approche
basee sur xml</titre>
  <nom_conf>CARI 2000</nom
_conf>
  <date> octobre

```

```
2000</date>
  <lieu>Antananarivo</lieu>
u>
  </publi>
  <publi id="p3"
  membre="halim"
  annee="2000">
    <titre>Modeling and
  information retrieval on
  xml-based dataweb</titre>

  <nom_conf>ADVIS2000</no
  m_conf>
    <date> octobre
  2000</date>
    <lieu>Izmir</lieu>
  </publi>
<publis_conf>
...
<groupe_si_web>
```

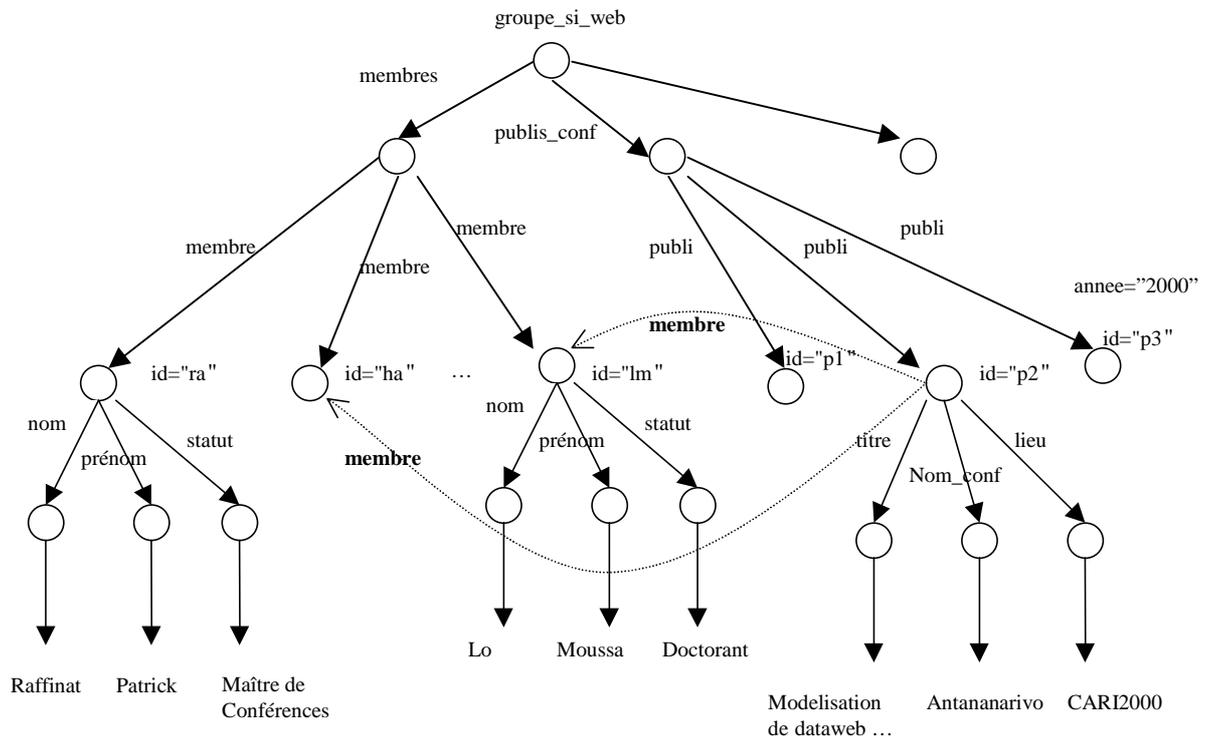


Figure 2 : Exemple de graphe XML représentant une partie du document XML ci-dessus

3. 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.

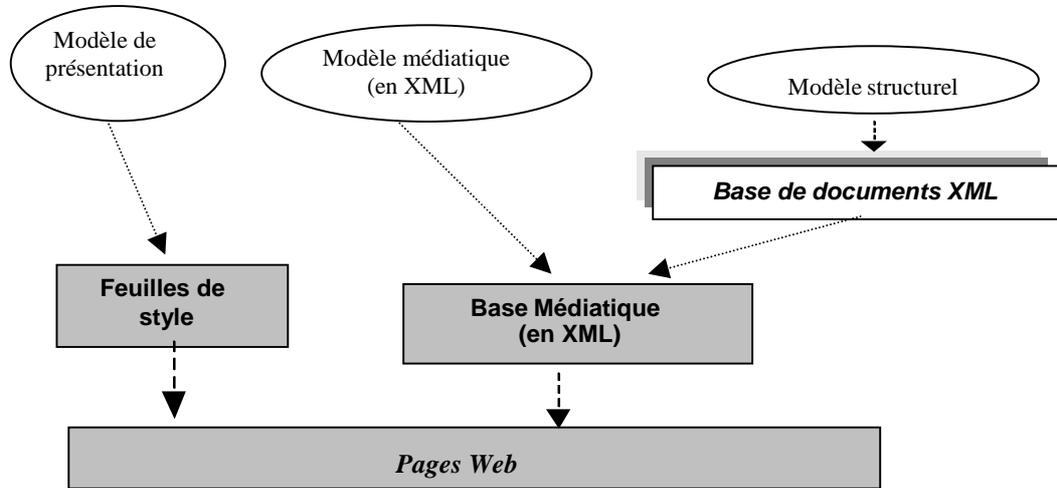


Figure 3 : Architecture fonctionnelle d'un dataweb

3.1. Les types d'unités médiatiques et leur description en XML

Nous proposons 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 les publications du groupe SI-Web dans les congrès internationaux :

```
<um id="u3" type="Xobjet">
```

```

<vue_filtre id="congre_int">
  <publi definition="groupe_si_web/publis_conf/publi">
    <annee definition="annee"/>
    <titre/>
    <nom_conf/>
    <date/>
    <lieu/>
  </publi>
</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 Transformations) [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="groupe_si_web/publis_conf/publi">
        <publi>
          <annee> <xsl:apply-templates
select="@annee"/> </annee>
          <titre> <xsl:apply-templates
select="titre"/></titre>
          <nom_conf> <xsl:apply-templates
select="nom_conf"/></nom_conf>
          <date> <xsl:apply-templates
select="date"/></date>
          <lieu> <xsl:apply-templates
select="lieu"/></lieu>
        </publi>
      </xsl:for-each>
    </Xobjet>
  </xsl:template>

```

</xsl:stylesheet>

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

```
<Xobjet>
  <publi>
    <titre>Information retrieval using
a base of concepts and ml</titre>
    <nom_conf>ISK06</nom_conf>
    <date>Juillet 2000</date>
    <lieu>Toronto, Canada</lieu>
  </publi>
  <publi>
    <titre>Modelisation de dataweb: une
approche basee sur xml</titre>
    <nom_conf>CARI 2000</nom_conf>
    <date>Octobre 2000</date>
    <lieu>Antananarivo,
Madagascar</lieu>
  </publi>
  <publi>
    <titre>Modeling and information
retrieval on xml-based dataweb</titre>
    <nom_conf>ADVIS2000</nom_conf>
    <date>Octobre 2000</date>
    <lieu>Izmir, Turquie</lieu>
  </publi>
</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 :

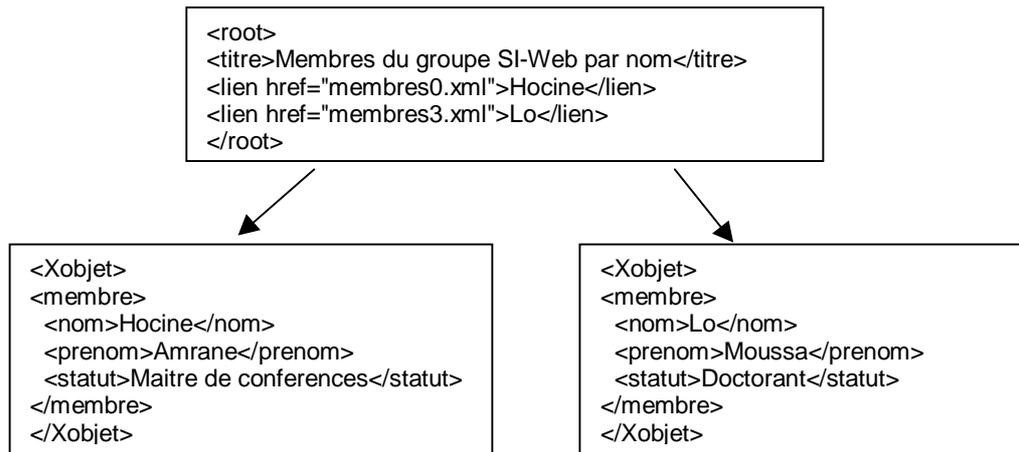
- Les *contextes simples* dont le filtre, permettant d'obtenir les xobjets, fait intervenir une seule entité du modèle E/A ;
- Les *contextes composés* dont les filtres font intervenir plusieurs entités (donc au moins une association) du modèle E/A.

B1 – Exemple de contexte simple : les membres du groupe SI-Web par nom.

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

```
<um id="membres" type="contexte_simple">
  <vue_filtre id="f_membres"
  definition="groupe_si_web/membres/membre">
    <info_index>
      <index>nom</index>
      <titre>Membres du Groupe
      SI_Web</titre>
    </info_index>
    <nom/>
    <prenom/>
    <statut/>
  </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 membres) et de construire l'index. On obtient alors le résultat suivant :



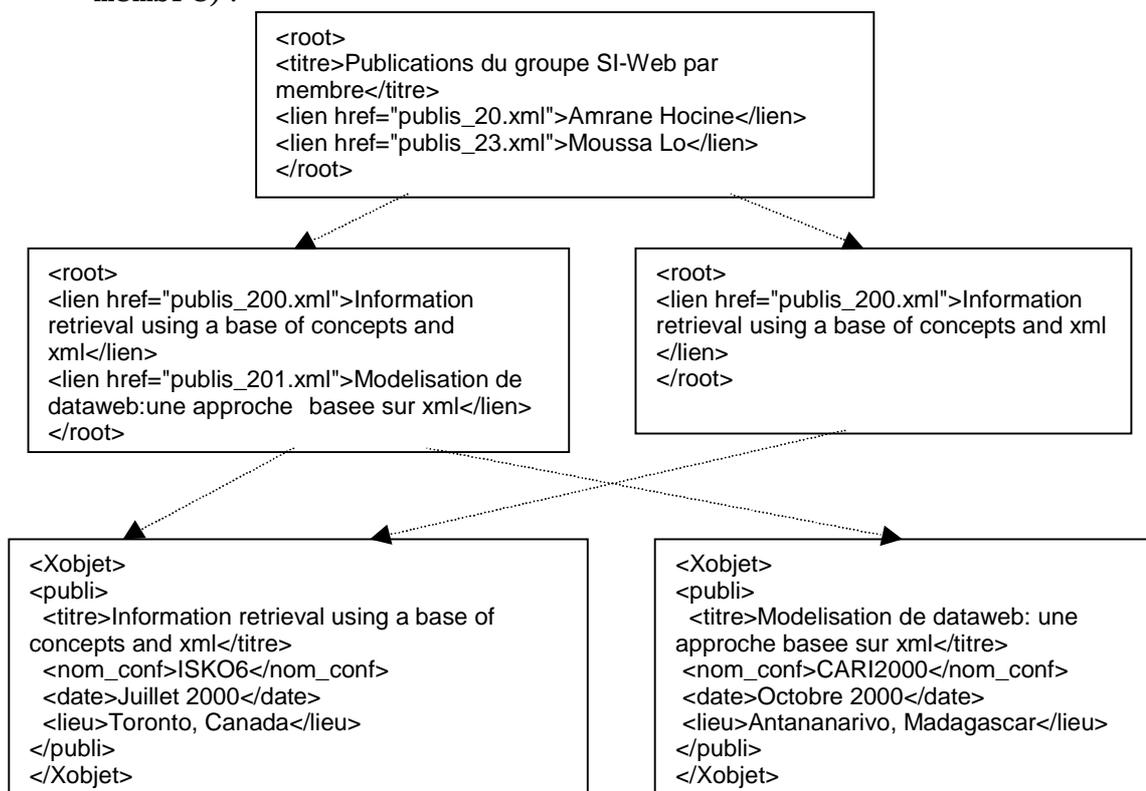
B2 – Exemple de contexte composé : les publications par membre.

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 "*fait*" du modèle E/A (Fig.1) reliant les entités "*membres*" et "*publications*", donc les balises `<membres>` et `<publis_conf>` du modèle structural. Deux filtres sont alors utilisés :

- Un premier filtre, appliqué à la balise `<membres>`, permet d'obtenir la liste des membres (id, nom, prénom) et une référence à leur département (dept). L'identifiant (id), qu'on peut avantageusement remplacer par le couple (nom, prénom), détermine l'index principal.
- Un deuxième filtre, appliqué à la balise `<publis_conf>` conditionnellement à chaque valeur de l'index principal (ie conditionnellement à chaque membre), permet d'obtenir les publications

par membre (les Xobjets représentés ci-dessous) et un index secondaire (les publications de ce membre).



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="themes">Themes de recherche</lien>
    <lien id="membres">Membres</lien>
    <lien id="publis">Publications</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 groupe</texte>
```

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

```
<image id="uppa" />
```

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.

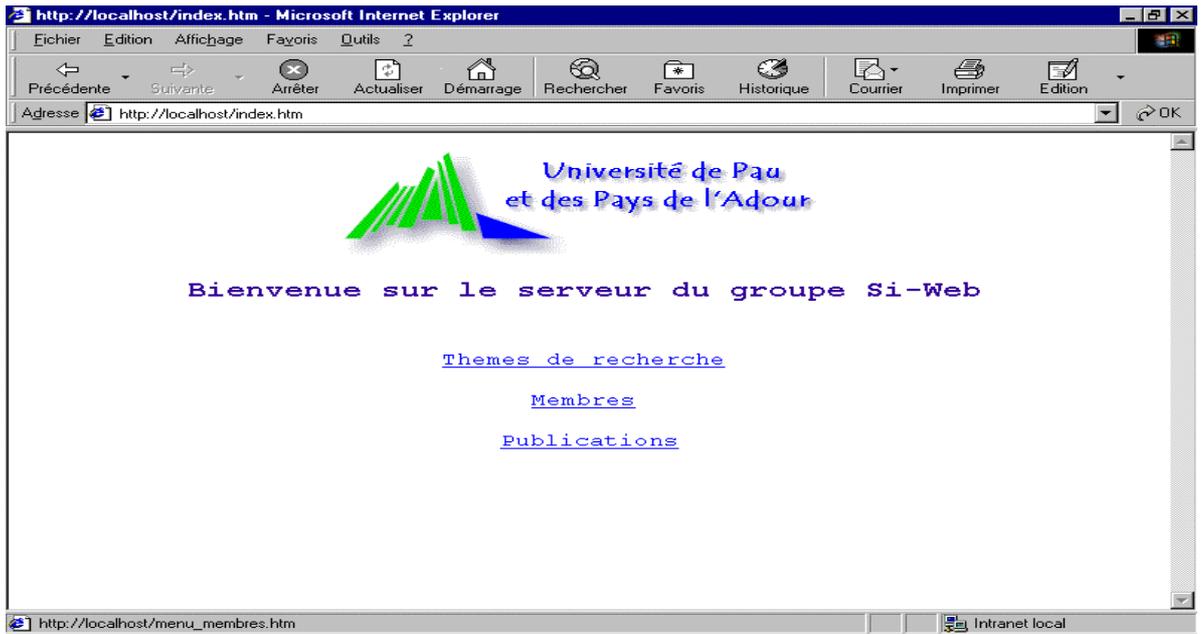


Figure 4 : Page d'accueil du site web du groupe si-web

La figure 4 montre la page d'accueil du groupe SI-Web. Cette page résulte de l'application d'une forme médiatique (ici une feuille de style) à l'unité médiatique suivante :

```
<root>
  <image>logouppa.gif</image>
  <titre>Bienvenue sur le serveur du groupe Si-
  Web</titre>
  <lien href="themes.xml">Themes de
  recherche</lien>
  <lien href="menu_membres.xml">Membres</lien>
  <lien href="publis.xml">Publications</lien>
</root>
```

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

```
<um id="menu_general" type="page_web">
  <image id="uppa"/>
  <titre>Bienvenue sur le serveur du groupe Si-
  Web</titre>
  <um id="menu_gen" type="menu" >
```

```
<lien id="themes">Themes de recherche</lien>
<lien id="membres">Membres</lien>
<lien id="publis">Publications</lien>
</um>
</um>
```

3.2. Les différentes étapes pour la construction de la base médiatique

Pour construire une base médiatique, nous proposons 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.

4. 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 ») : il est donc impossible de le présenter dans cet article sans être très réducteur.

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 3.1), c'est à dire la description de chaque unité médiatique de type « unité page».
- Un catalogue 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 l'université, nommée

« uppa » (voir paragraphe G, section 3.1), correspond au fichier « uppa.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 [17], en l'occurrence sur la structure arborescente d'une unité médiatique de type "unité page".

Procédure Générer_Page (Racine : Nœud)

Début

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

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

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

Traiter_Nœud (nd_fils, F)

Fin Pour

Fin

Procédure Traiter_Noœud (nd : Nœud, 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 le filtre
  insérer le résultat dans le fichier F
Cas "contexte simple" :
  exécuter le filtre
  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 ! */
  exécuter le premier filtre
  créer un index principal à partir du
  premier filtre et l'insérer dans le fichier F
  Pour chaque élément de l'index principal
  exécuter le deuxième filtre
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

```

Dans notre prototype réalisé en Java, il a fallu introduire des classes spécifiques pour les unités médiatiques les plus sophistiquées (contexte composé notamment) afin de réduire sensiblement la complexité du programme. Le programme comporte quelques centaines de lignes de code (sans compter bien sûr les paquetages relatifs au DOM et à XSLT), dont un peu moins de 200 lignes pour les contextes composés.

5. Sur quelques travaux similaires

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 hypermédiés. 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. Elles utilisent des techniques de modélisation basées d'une part sur le modèle entité-association (HDM [8], RMM [10]) et, d'autre part, sur le modèle objet (OOHDM [14]).

Les solutions proposées dans ces travaux peuvent être adaptées au contexte du Web et ont inspiré la plupart des travaux menés pour la conception de sites 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 dont une de leur caractéristique commune est le langage utilisé : HTML.

Strudel [7] est un système qui adapte les concepts classiques des SGBD au processus de construction des sites Web. Il distingue trois niveaux de données dans un site Web : l'information disponible dans le site et qui provient éventuellement de sources hétérogènes, la structure hypertexte (vue intégrée) et le site lui-même (présentation graphique).

A chaque niveau, les données sont modélisées par un

graphe ; celles du premier niveau sont stockées dans une base de données fondée sur le modèle de données de Strudel ou dans des sources externes (pages HTML, BD relationnelles ou objets, ...).

Le modèle de données de Strudel est un modèle de graphe. Une base de données est représentée par un graphe de données contenant des objets connectés par des arcs orientés et étiquetés par des attributs. Les objets sont eux-mêmes des nœuds avec un identificateur (Oid), ou des valeur atomiques (entiers, chaînes de caractères, fichiers, ...).

Strudel fournit un langage de requêtes nommé StruQL (Site Transformation Und Query Langage) utilisé aussi bien pour l'intégration de données provenant de sources externes dans un graphe, que pour la transformation d'un graphe (informations à publier) en un ou plusieurs autres graphes (vues).

Araneus [11] introduit la notion de Web-Base qui est définie comme une collection de données de nature hétérogène (structurées et semi-structurées). Araneus est alors un système de gestion de Web-Base, c'est-à-dire un système offrant les fonctionnalités de gestion de bases de données et de sites Web.

L'originalité de Araneus réside dans la définition d'un modèle de données (ADM) pour les documents Web et hypertextes, de plusieurs langages pour l'interrogation, la création et la mise à jour de sites Web, et des méthodes et techniques de conception et d'interrogation et d'implémentation de sites Web.

Dans le modèle de données ADM, un site Web est une collection de schémas de pages connectées par des liens. Les documents d'un site sont d'abord classés en différents types de documents. Un schéma de pages est ensuite construit manuellement pour chaque type de documents. Dans le système, les objets ADM sont manipulés grâce à un gestionnaire d'objets qui repose sur un SGBD.

WebML [4] est un langage de description pour concevoir des sites Web. Il est accompagné d'une méthodologie de

conception articulée autour de quatre modèles :

- Un modèle structurel qui décrit le contenu du site à l'aide du modèle entité-association,
- Un modèle hypertexte composé de deux sous modèles : un modèle de composition et un modèle de navigation. Le modèle de composition spécifie les pages du site. Le modèle de navigation décrit la structure hypertexte des unités et pages décrits dans le modèle de composition. Pour cela, le concepteur dispose de deux types de liens : contextuels (issus du schéma de données de l'application) et non-contextuels (libres).
- Un modèle de présentation qui permet d'associer un style de présentation à chaque page.
- Un modèle de personnalisation permettant d'effectuer une personnalisation du site selon des groupes d'utilisateurs.

Après cette phase de conception, les unités et pages sont transformées en documents HTML (ou WML). Tous les concepts du langage WebML sont associés à une notation graphique et à une syntaxe textuelle XML. Un atelier de conception Web nommé ToriiSoft implémente WebML et la méthodologie associée.

Remerciements : Nous tenons à remercier Melles V. Ellisalde et K. Rousseu-Salet, étudiantes du DESS IMOI, qui ont participé à l'amélioration du module de génération de la base médiatique (programmé en Java).

6. Conclusion

Nous avons proposé une approche de modélisation et de conception d'applications web basée sur la prise en compte des aspects logiques et sémantiques des données pour la représentation des documents web, et une meilleure structuration de ces documents par l'utilisation du formalisme XML.

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 l'ensemble de ces systèmes, 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.

7. Références bibliographiques

- [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 Recommendation 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. Rousseu-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. Meri aldo, 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 : <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 : <http://www.w3.org/TR/xslt.html>, Version 1.0, Recommendation 16 November 1999
- [18] W3C: <http://www.w3.org/TR/1998/REC-DOM-Level-1-19981001>