A Designing Model of XML-Dataweb

Moussa Lo [1, 2], Amrane Hocine[2] , Patrick Raffinat[2]

{first-name.last-name}@univ-pau.fr

[1]Laboratoire LANI - Université Gaston Berger
Saint-Louis (Sénégal)

[2] Laboratoire d'Informatique
Université de Pau (France)

Abstract

We propose a designing method of Web application based on the new standard XML and on the dataweb notion. The content of the dataweb is obtained by an integration technique of heterogeneous data sources through a model called structural. The dataweb content is diffused through the Web according to a method leaned on a media model. The method consists to define at first the media model that is a view of the structural model and generate next the media base. A presentation model describes the user interface where each media object is associated with a presentation. In this paper, we are interesting in the structural and media models.

Key words: Web application, dataweb, designing method, XML

1 Introduction

The Web is more and more used as a platform for complex information systems. Some development tools have allowed to bring an appreciable assistance in the generation and the quick realization of Web applications, with the ASP (Microsoft Active Server Pages), JSP (Java Server Pages), PHP (Personal Home Page or Hypertext Preprocessor), PL-SQL (Oracle-Web) technologies, etc. These technologies allow extracting dynamically information from different data sources and to include them in HTML (Hypertext Markup Language) pages patterns. In these applications, the designers have often privileged the presentation aspect to the detriment of the data structuration. This approach shows its limits during exploitation of these sites. The problems are often caused by the complexity of the sites, the need to have interoperability with other applications, the necessity of modification and the evident lack of interrogation possibility of HTML pages.

Most generally, these systems can be considered as important information's base integrating many heterogeneous data sources. The problem of these systems are the need of an adequate designing methodology, based on a quite rich data model to allow the realization of a specific information retrieval process linked to the domain user. The limits caused by the use of the HTML language during the information retrieval process and the emergence of the XML (eXtensible Markup Language) formalism, impose the use of XML for the construction of important Web sites: firm Web sites, information systems based on the Web, etc.

We propose a designing method of Web applications based on the new standard XML and on the dataweb concept. In the MEDX[1] (Modeling and Exploration of Dataweb based on XML) project, a dataweb is defined as a unified and integrated view of a collection of data: (i) structured such as the ones stored in relational or object databases; and, (ii) semi-structured [1] as the data of Web.

The dataweb integrates all the data of the information system in a global source; the use of XML allows having a uniform vision of those heterogeneous data.

[1] This work is done in the context of the MEDX project, a research project of the SI-Web group (Computer Science Laboratory of Pau University).

The XML properties to be published on the Web and to be stored in databases allow managing and diffusing data contained in the dataweb.

The designing of a dataweb is a complex task, which requires an significant effort of analysis and design. The designing method of dataweb we have developed is based on the three following models:

- The structural (or conceptual) model which allows to identify various sources of information of dataweb and integrate them with XML. That is the beginning of the designing task.
- The media (or navigational) model which describes the media objects and their browsing structures. Many media models can be built from a structural model. In the prototype we currently develop [7], we use an XML editor to describe the media model and a Java [15] program to generate the media units.
- The presentation model that describes the user interface where each media unit is associated with a media form (i.e. a presentation which specifies how the corresponding information is presented to user). In the implementation of our prototype, the media form is obtained by using stylesheets (CSS or XSL [18]) which are associated to each media unit.

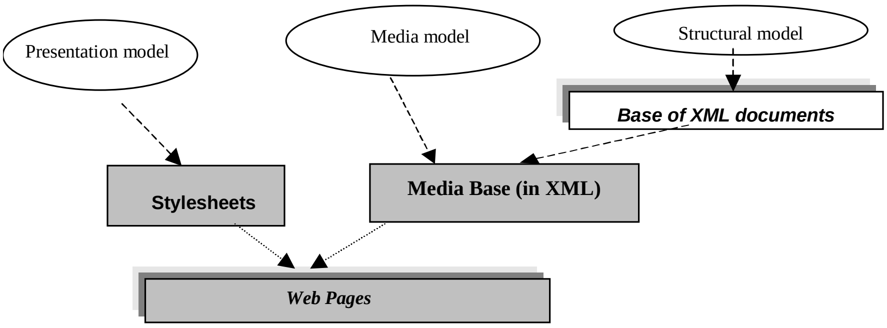


Fig. 1: Functional architecture of the dataweb

After this introduction, the section 2 takes up the integration technique for the heterogeneous data sources of the dataweb, through the structural model. The media model and the description language of the media units, particularly the Xobjects and the navigational contexts, are presented in the section 3. In the section 4, we present the media base which implements the media model. The related work with our approach is described in section 5. We will prop our talk with an example: the designing of the Web site of our research group ("the SI-Web group").

2 The structural model

The first step of our designing methodology is to integrate the various data sources of the dataweb. The problem of information integration is addressed by mediation systems [18]. There are two main research ways concerning mediation based systems: the first approach relies on a common data model (generally a semi-structured one) to represent data coming from heterogeneous sources and mediate queries expressed in a common query language; the second approach is domain-model based which relies on a domain model in the mediator level expressed in some database or knowledge formalism [5].

We propose to integrate the dataweb data sources through a structural model obtained by the next designing approach:

a) Identification of the data sources (structured and semi-structured) of the dataweb.

b) Use of the Entity-Relationship (E-R) formalism to model each structured data source, and of XML to formalize each semi-structured data source.

c) Translation of the E-R models obtained in the previous step in relational schemas.

d) The mapping technique (relational – XML) we have detailed in [11] allows, for each source, a structural schema which represents the logic structure of the relational database in XML. The problem of transformation of data coming from relational databases to XML documents is only approached since a short time and has given some tools like DB2XML [17], PLSXML [14] and XML-DBMS [3]. These three systems are Java packages and are adopted the same approach: they use a very simple mapping technique based on the DOM (Document Object Model) [21]. The foreign key's problem is not approached; that implies a redundancy of information in the obtained document. The structure of this document is a tree structure and not an oriented graph as thus we propose.

e) Insertion in a base of XML documents of the data coming from the relational databases.

The set of structural schemas coming from various structured data sources of the dataweb, and the schemas of XML documents coming from the semi-structured data, form the structural model of the dataweb.

In our example, we have identified two data sources:

- the information concerning the members and their publications (in conferences) of the research group are modeled with the E-R model (structured data).
- the research themes described in a text document are formalized in an XML document (semi-structured data).

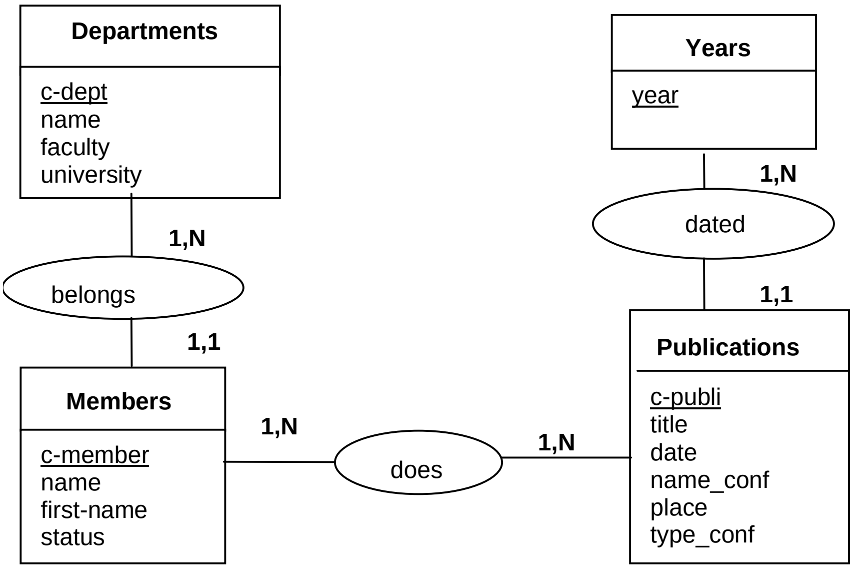


Fig. 2: E-R model schema of the database of the SI-Web group

The dataweb structural model and a simplified extract of the base of XML documents obtained after the mapping operation are given below:

```
    <si_web_group>
    <depts>
        <dept id="">
            <name/>
            <faculty/>
            <university/>
        </dept>
    </depts>
    <members>
        <member id="" dept="">
            <name/>
            <first-name/>
            <status/>
        </member>
    </members>
    <years>
        <year id=""/>
    </years>
    <publis_conf>
    <publi id="" member=""    year="">
            <title/>
            <name_conf/>
            <date/>
            <place/>
            <type_conf/>
        </publi>
    </publis_conf>
    <themes> ... </themes>
</si_web_group>
```

```
<si_web_group>
<depts>
<dept id="info_uppa">
    <name>Dept Info. </name>
    <faculty>Fac sciences </faculty>
    <university>UPPA</university>
</dept>
…
</depts>
<members>
<member id="ha" dept="info_uppa">
    <name>Hocine</name>
    <first-name>Amrane</first-name>
    <status>Maitre de conf.</status>
</member>
…
</members>
<publis_conf>
<publi id="p1" member="ha lm ss"
year="2000">
<title>Information retrieval …</title>
   <name_conf>ISKO6</name_conf>
    <date> juillet 2000</date>
    <place>Toronto</place>
</publi>
…
</publis_conf>
…
</si_web_group>
```

Fig. 3: The structural model (on the left) and the base of XML documents (on the right)

3 The media model

We use a declarative way to diffuse the content of the dataweb, through a Web site. That is to define the structure of the Web site as a view over the existing data, i.e. the base of XML documents. This base provides an uniform view over the underlying data sources.

Building a Web site using a declarative representation of the site structure presents significant advantages [9], particularly:

- It is easy to create multiples versions of a site; this property is very important for offering different views depending on class of users.
- It facilitates the evolution of the Web site's structure.

To use such method, we use a media model at the conceptual level; a media model is a view of the structural model. It describes the media units of the dataweb and their navigational structure.

A media unit (M.U.) is an " information unit which has a certain autonomy in a user's point of view (i.e. which has its own sense and so presents a coherent idea or a concept) and which merits to be solicited in many consultation steps" [6].

In the dataweb, a M.U. is described in XML and constitutes the content of a Web page. The set of media units and their navigational links (i.e. a media base) is obtained from an algorithm of automatic generation based on the media model. This model, described by specific XML tags is presented in this section.

We propose eight types of media units: Xobjects, navigational contexts, index, menus, links, texts, images and web pages.

The Xobjects and the navigational contexts provide views over the base of XML documents; they are described more precisely in the next subsections.

A media unit with text type is free text added by the designer.

The image type permits to insert a picture from a file.

The links allow describing the hypertext links in the dataweb. One distinguishes three sort of link:

- the structural links: they describe the navigational structure of the M.U.. From a media unit, a structural link permits to go to the closest related units in the navigational structure (previous structure, next structure, home page, etc.).
- the context links: they come from the structural model and permit to connect media units having a semantic link, for example the index links.
- the referential links*:* they permit to go to any M.U., and to outside URL.

The page units allow composing many M.U. into global M.U. described in XML, which will be associated to stylesheets to generate the pages of the web site.

A menu is a set of (referential) links to other media units.

3.1 The Xobjects

An Xobject is an extract of the base of XML documents, obtained from the application of a filter (or query). The definition of an Xobject requires the description of a view over the base of XML documents.

Syntactically, an Xobject is described by using the predefined elements <view_filter> and <definition>. They allow to generate automatically a query (named filter) written in XSLT (Extensible Style Language Transformations) [20]; a language allowing to transform XML documents into another XML documents.

As example, here is a view, which allows to obtain the SI-Web group publications in international congress:

```
<um id="u3" type="Xobject">
    <view_filter id="congre_int">
     <publi definition="si_web_group/publis_conf/publi">
        <year definition="year">
        <title/>
        <name_conf/>
        <place/>
     </publi>
    </view_filter>
</um>
```

The application of the filter coming from this view on the base of XML documents gives the next result:

```
<Xobject>
        <publi>
          <title>Information retrieval using a base of concepts and ml</title>
          <name_conf>ISKO6</name_conf>
          <date>Juillet 2000</date>
          <place>Toronto, Canada</place>
        </publi>
        …
</Xobject>
```

3.2 The navigational contexts

A navigational context is a set of Xobjects concerning a given thematic, they are accessible from an index. It is obtained, like an Xobject, by the application of a filter on the base of XML documents.

The definition of a navigational context comprises the description of a view over the base of XML documents and the description of an index. The view allows to obtain a set of Xobjects accessible from the index.

Syntactically, a navigational context is described by using the predefined elements: <view_filter>, <definition>, <info_index> and <index>.

We distinguish two types of navigational context:

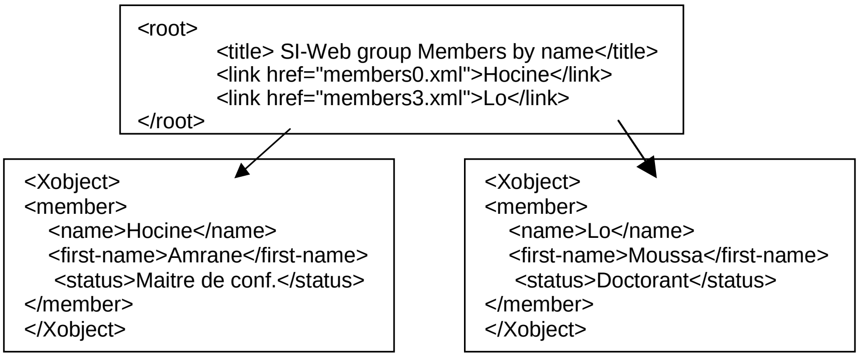
- *The simple contexts* which the filter, allowing to obtain the Xobjects, makes intervene an unique entity of the E-R model.
- *The composed contexts* which the filter make intervene many entities (then at least an association) of the E-R model.

3.2.1 Example of simple context: the SI-Web group's members by name

Such context permits to obtain many Xobjects which represent respectively the page of a group member and an index allowing access to each of these Xobjects. It is described as below:

```
 <um id="members" type="simple_context">
     <view_filter id="f_members">
      <member definition="si_web_group/member">
          <info_index>
              <index> name </index>
              <title> SI_Web Group Members </title>
          </info_index>
          <name/>
                  <first_name/>
                  <status/>
               </member>
       </view_filter>
</um>
```

This view is then translated into an XSLT filter which the application allows to obtain the concerned Xobjects (i.e. the members) and to construct the index. We obtain the next result:



```
<root>
          <title> SI-Web group Members by name</title>
          <link href="members0.xml">Hocine</link>
          <link href="members3.xml">Lo</link>
</root>
```

```
<Xobject>
<member>
   <name>Hocine</name>
   <first-name>Amrane</first-name>
    <status>Maitre de conf.</status>
</member>
</Xobject>
```

```
<Xobject>
<member>
   <name>Lo</name>
   <first-name>Moussa</first-name>
    <status>Doctorant</status>
</member>
</Xobject>
```

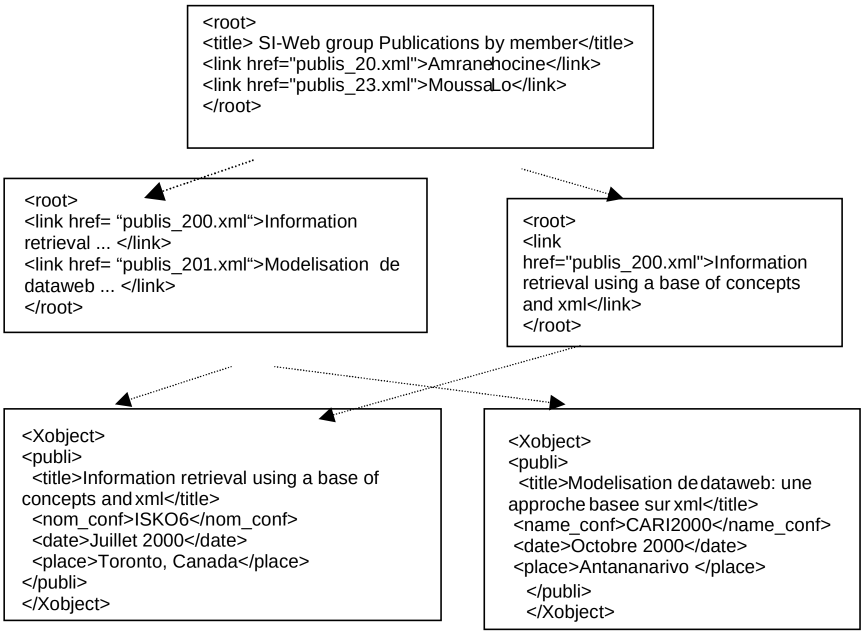
Fig. 4: Example of simple navigational context

3.2.2 Example of composed context: the publications by member

Contrary to the simple contexts, the composed contexts make intervene many entities and then at least an association of the E-R model. In our first approach, only two entities and an association are taken into account.

The example makes intervene the association "*does*" of the E-R model (fig. 2) linking the entities "*members*" and "*publications*", then the tags <members> and <publis_conf> of the structural model. Two filters are then used:

- A first filter, applied to the tag <members>, allows to obtain the list of the members (id, name, first-name) and a reference to their department (dept). The identifiant (id), we can replace by the couple (name, first name), determines the principal index.
- A second filter applied to the tag <publis_conf> conditionally to each value of the principal index (i.e. conditionally to each member), allows to obtain the publications by member (the Xobjects represented below) and an secondary index (the publications of this member).



```
<root>
<title> SI-Web group Publications by member</title>
<link href="publis_20.xml">Amranehocine</link>
<link href="publis_23.xml">MoussaLo</link>
</root>
```

```
<root>
<link href= "publis_200.xml">Information
retrieval ... </link>
<link href= "publis_201.xml">Modelisation  de
dataweb ... </link>
</root>
```

```
<root>
<link
href="publis_200.xml">Information
retrieval using a base of concepts
and xml</link>
</root>
```

```
<Xobject>
<publi>
  <title>Information retrieval using a base of
concepts and xml</title>
  <nom_conf>ISKO6</nom_conf>
  <date>Juillet 2000</date>
  <place>Toronto, Canada</place>
</publi>
</Xobject>
```

```
<Xobject>
<publi>
  <title>Modelisation de dataweb: une
approche basee sur xml</title>
 <name_conf>CARI2000</name_conf>
 <date>Octobre 2000</date>
 <place>Antananarivo </place>
   </publi>
   </Xobject>
```

Fig. 5: Example of composed navigational context

4 The media base

The media base implements the media model. To build a media base, we propose the following steps:

a) Determine the "page units";
b) Determine the navigational units between page units;
c) Describe formally in XML the media units to build;
d) Generate the media base from the M.U. description and the base of XML documents.

The generation of the media base results from a relatively complex process (particularly for the media units with "composed context" type): it is then impossible to present it without simplification.

The generation algorithm has as entry three types of information:

- the base of XML documents (Fig. 3)
- the XML description of the media units (section 3)
- a catalog allowing to establish a correspondence between the resources (images, destination URL of the hypertext links, filters, …) logic names and their physic names: we can for example precise that the M.U. corresponding to the university logo, named "uppa", corresponds to the file "uppa.gif". This catalog is also descriebd in XML.

The result of this algorithm is a collection of media units, essentially with "Xobjects" and "page units" types. To build the web pages from the media base, it remains to associate a media form to each media unit, by using stylesheets (CSS or XSL) for example.

Next, we present the algorithm for generation of a page; seeing that the general algorithm comes from it directly. This algorithm leans on the tree structure of any XML document [21], in this case the tree structure of a media unit with "page unit" type.

```
  Procedure Generate_Page ( Root : Node )
Begin
    Open in write mode the file (named F) in which the M.U with "page unit" type
will be created
           / * the physic name is determined by the tag id and the catalog */
    For each child node (named nd_child) of the root of the tree Do
        Treat_Node (nd_child, F)
     End For
End

Procedure Treat_Node ( nd : Node, F : File )
Begin
    Case type_MU Of
        "Text" : insert the text in the file F
        "Image" :   find the image physic file name from the catalog
                    insert the image in the file F
        "Link" : find the URL physic name from the catalog
                  create a referential link to this URL in the file F
        "Xobject" : execute the filter
                    insert the result in the file F
        "simple context " : execute the filter
                            infer  the Xobjects and an index
                             insert the index in the file F
      create a file for each Xobject
        "composed context" : /* simplified version */
              execute the first filter
            create a principal index from the first filter and insert it in the file F
            For each element of the principal index
                    execute the second filter conditionally to this element
                    infer  the Xobjects and a secondary index
                    create a file for the secondary index
                    create a file for each Xobject
            End For
      End Case
End
```

Fig. 6: Generation algorithm of the media base

5 Related work and conclusion

Many works are done to propose methodologies and models of conception of Web application. We can distinguish the conception methodologies of hypermedia applications and the management systems of Web dynamic sites.

The goal of the first works in the domain had to resolve the hypermedia systems's problems. That was to find methodology adapted to this type of applications, which are specific in relation to traditional ones. All these methodologies are founded on the separation between the domain analysis, the specification of the navigational space and the conception of the user interface. They use modeling techniques based on the one hand, on the Entity-Relashionship model (HDM [10], RMM [12]); and on the other hand, on the object model (OOHDM [16]). The solutions proposed in those works could however be applied to the Web context and inspired many of the works done for the conception of Web applications.

In addition to the problems linked to the hypermedia context, other problems linked to the Web specificity arise to Web application designer: integration of various data sources, interoperability, dynamic nature of the Web, need to couple with DBMS (Database Management Systems), etc.

In this context, many systems are developed; however, they are all build on HTML.

STRUDEL [8] is a system, which adapts the classic DBMS concepts to the process of building Web sites. It distinguishes three levels of data in a Web site: the available information in the site which eventually comes from heterogeneous sources, the hypertext structure (integrated view) and the site itself (graphical presentation). At any level, the data are modeled by a graph; those of the first level are stocked in a database founded on the STRUDEL data model or in extern sources (HTML pages, relational or object databases). STRUDEL provides a query language named STRUQL (Site Transformation Und Query Langage) used for the interrogation of data and graph transformation.

Araneus [13] introduces the Web-Base notion defined as a collection of heterogeneous data (structured and semi-structured). Araneus is then a system of Web-Base management, i.e. a system providing the functionality of DBMS and Web sites. The Araneus originality is the definition of a data model named ADM (Araneus Data Model) for the Web and hypertext documents, many languages for the interrogation, the creation and the updating of Web sites methods and techniques of designing, interrogation and implementation of Web sites.

WebML [4] is a descriptive language for designing Web sites. It is accompanied by a designing methodology based on four models. A structural model describes the site content with the Entity-Relationship model. The composition model specifies the pages of the site. The navigational model describes the hypertext structure of units and pages described in the composition model. A presentation model allows associating a presentation style to each page and a personalization model allows the personalization of the site according group users. After this designing step, the units and pages are transformed into HTML documents. All the concepts of the WebML language are associated to a graphic notation and to a textual XML syntax. A designer tool suite named ToriiSoft implements WebML and the designer methodology associated.

Our approach is different from these systems

- in the one hand, by the using of XML to represent the data of the site, during its designing stage (in a base of XML documents coming from the structural model), and also during its exploitation (in a media base coming from the media model).

- the enormous possibility it allows to integrate after a relevant information retrieval system. In fact, we adapt easily to the dataweb context our works concerning information retrieval by semantic content in a base of XML documents [11].

Our future works will particularly concern the problems linked up the dataweb update.

6 References

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. Sperbeg-MacQueen: Extensible Markup Language (XML) 1.0, W3C Recommandation, 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. V. Christophides: Community Webs (C-Webs): Technological Assessment and System Architecture, Research Report, INRIA, September 2000.
6. R. Deschamps: Bases de connaissances généralisées : une approche fondée sur un modèle hypertexte expert. Ph D of Toulouse University, France, 1995.
7. V. Ellisalde & K. Rousseu-Salet : Modélisation objet et implantation en Java d'une Base Médiatique, *Grand Projet*, DESS IMOI, Mars 2001, Université de Pau.
8. 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.
9. D. Florescu, A. Levy & A. Mendelzon: Database Techniques for the Word-Wide Web: A Survey, ACMSIGMOD record 17(3), September 1998.
10. 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.
11. A. Hocine, M. Lo : Modeling and information retrieval on XML-based dataweb, *Proceedings of First Biennal on Advanced in Information Systems*, Izmir, Turquie, LNCS vol. 1909 pp. 398-408, October 25-27, 2000.
12. T. Isakowitz, E. Stohr & P. Balasubramanian : A methodology for the design of structured hypermedia applications. *Communications of the ACM*, (8)38, 34-44, 1995.
13. G. Mecca, P. Atzeni, P. Merialdo, A; Masci, and G. Sindoni: >From Databases to Web-Base: The Araneus experience, *Sigmod 98*, 1998.
14. S. Muench : PLSXML Utilities and demos. Oracle Technical Whitepaper, March 1999.
15. Sun : http://java.sun.com/
16. D. Schwabe and G. Rossi : OOHDM : The object-Oriented Hypermedia Design Model, *Communication of ACM*, August 1995.
17. V. Turau, Making legacy data accessible for XML applications, Technical Report, FH Wiesbaden, University of Applied Sciences, 1999.
18. G. Wiederhold: Mediation in information systems, ACM Computing Surveys, 27(2), pp 265-267, June 1995.
19. W3C : http://www.w3.org/TR/2000/CR-xsl-20001121/
20. W3C : http://www.w3.org/TR/xslt.html, Version 1.0, Recommendation 16, November 1999.
21. W3C: http ://www.w3.org/TR/1998/REC-DOM-Level-1-19981001/