EIHA?!? : deploying Web and WAP services using XML technology

Chiara Biancheri, Jean-Christophe Pazzaglia, Gavino Paddeu
CRS4 Center for Advanced Studies, Research and Development in Sardinia
{chiara, jc, gavino} @crs4.it

Abstract

The exponential growth of resources on the web, and the wide deployment of devices for multimodal access to the Internet, lead to new problems in information management. In this context, and as part of the European project Vision, we have built an interactive telematic handbook of the culture and the territory of Sardinia. A team of cultural experts browsed the web to get a large collection of Internet resources.

The system built for the management of this data uses emerging Internet technologies such as the XML language suite and its applications. The result obtained is a multimodal service, called EIHA?!? *, available through PCs and mobile phones.*

Keywords Metadata, DTD, XML, XSL, WML, WAP, DBMS, Search Engine.

Introduction

The Internet phenomenon has exploded through the construction of web sites and personal pages for every kind of activity. Such new users bring various requirements and expectations, and have evidenced the fundamental importance of two distinguished and complementary instruments – search engines and the multimodality.

In this context, the main objective of project Vision [1] is the construction of an interactive telematic handbook of the culture and the territory of the Sardinia, available on the Internet trough a dedicated web site.

The first step in the project involves the collection of data already in existance, regarding every aspect of the culture and the economy of the Sardinia island. The web sites collected need one unique format for classification, consultation and search. We reached these requirements through the addition of meta-data, using which we were able to describe in a complete way several aspects of every resource. The XML technology offers therefore a useful instrument for the index creation and for the processing of the contents.

The article is organized as follow: in section 1 we describe the various features implemented by the system; in section 2 we present the general architecture; in section 3 we give the description of the types of data through an analysis of the DTD used by the system; in section 3.2 we introduce the query mechanism to the database; in section 4 we introduce shortly the language WML and its use inside of our system, and finally in section 5 we describe how the geographic localisation of the resources has been implemented in the system.

1 Functionality description

The main objective of the EIHA?!? project is the development of a web application in an XML environment, for the indexing, classification and integration of web sites about Sardinia. Therefore, a collection and categorization of pre-existing resources about Sardinia has been made using a document type definition (DTD), outlined in section 3, and undertaken in collaboration with cultural and historical experts. This collection, based on one DTD and containing meta-data on the archived resources, is available under two different and complementary aspects. The data is ordered in a hierarchical index, and also like an on-line catalog for intelligent search. This work, which has been carried out over many months, enabled the offering of a collection of more than 1700 resources describing the various aspects of Sardinian culture.

[1] The project Vision is financed by the European Social Found in the ADAPT - Bis program (Building the Information Society) and is realized at the CRS4

The functionality provided by the customer interface has been implemented to support various types of consultation through multiple channels. Our classification allows three types of browsing:

- navigation in the hierarchical index (Yahoo! like),
- contextual search engine based on the definition given by the experts (for example *Gavino* as a place name: “San Gavino” or as person: “Gavino Paddeu”),
- navigation on the Sardinia geographic map.

All these features are also offerend by well-known portals, but only few offer the same interface for different access devices. The consultation of the web site must therefore be available from various browser using HTML (see figure 9) or XML [2], and in particular using the dialect dedicated to the new generation of mobile phones.

2 Architecture overview

The overall architecture of the EIHA?!? system is presented in figure 4. In this architecture the XML language, conforming to our DTDs (see section 3), is used as a middleware language to carry relevant information from the database. The XSL stylesheets are used to adapt the presentation of the information to the target device. During this stage, the *transformation* capabilities of XSL [3, 7] are mainly used to filter and re-organize this information according to the characteristics of the device and the different hypertext languages. In the current version of our system, the effective presentation is dedicated to a Cascaded Style Sheet (CSS) compatible stylesheet for HTML supporting browsers and for Wireless Markup Language (WML) itself, while browsing with a WAP terminal (mainly a mobile phone).

The implementation is based upon current standards for dynamic web development in Java. A specialized servlet is running on top of the Tomcat server [11]. The roles of this servlet are multiple:

- It recognizes the user's device type and automatically delivers the appropriate markup language (mainly HTML and WML).
- It is responsible for storing session dependent information and manages a cache for time-consuming data elaborations (such as the elaboration of a map in SVG format).
- It manages the connection to the relational database using a JDBC connection, and an XML template based form.
- It elaborates the translation between the *advanced query* HTML form and the relational database.

We will now describe more precisely the set of DTD handled by the system, the template based system used to query the database, and also the WML language itself.

3 Data modelling and querying

3.1 The main DTDs

The first important activity for the development of our XML application has been the definition of the DTDs for the types of document used in our system. The analysis and the writing of DTDs allowed us to identify, and to describe the various components and their relations.

We can identify a selection of DTDs associated with the types of document that we need for the description of a hierarchical index:

- a DTD for the description of the hierarchy of the categories in which web sites are classified (figure 1: DTD `category`),
- a DTD for the description of a single resource, containing a URL, a set of additional information about the author, the language, the keywords and in particular an evaluation of the content, and geographic information (figure 2: DTD `resource`),
- a DTD for the description of the localities, as used by our GIS system (figure 3: DTD `map`).

These main DTDs have been built using a set of atomic DTDs, that are playing the role of basic abstract data type in our system. (As an example, one of this basic DTD is describing a format for a `date` while another is used to describe a `URL`).

The DTD `category` relating to the topic hierarchy enables us to describe a classic tree. Each node of this tree has a title, an abstract as well as a shortcut to its main topic, and a list of shortcut to related topics. Finally, a sub-list of categories is embedded in each category allowing the recursive description of

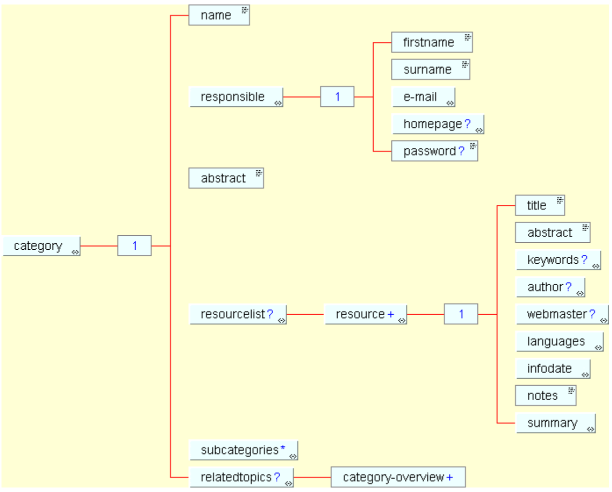

Figure 1: DTD `category`

the whole tree. The main difference between a *sub-category* and a *related category* resides in the recursive structure of a sub-category, compared to a related category being referenced as a link.

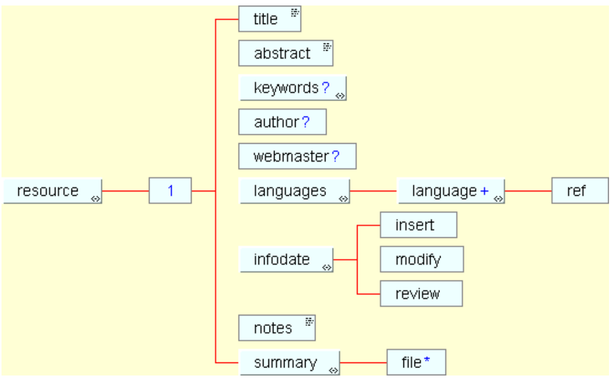

Figure 2: DTD `resource`

Our DTD for a `resource` can be seen as an extension of the Dublin Core (DC). In our vocabulary, a synthetic view describes the resource in a similar manner to the Dublin Core (i.e. title,abstract, keyword ...). We decided to identify two figures, the author of the content and the owner of the web site. In order to describe multi-language sites, we preferred to re-group the set of URLs for each language with the same description. Finally, we introduced a possibility to give an idea of the size of the site based on MIME types. This structure called *summary* enables us to store the number of documents available for each MIME type. Finally, each resource contains a link to its geographical position.

The `map` DTD is used to describe a classic geographic map. We are using a description based on

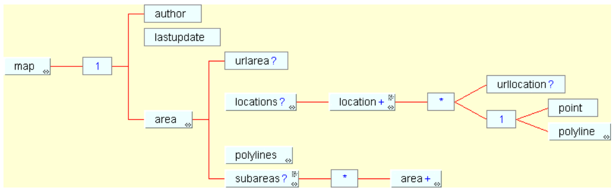

Figure 3: DTD `map`

the administrative sub-division of the territory (state, region, province, town). Every area is described using a set of poly-lines constructed with geo-referenced points (latitude and longitude). An area can also contains different places of particular interests (mountains, lakes, a city, for instance). All the elements hold a small description and a reference to a URL.

3.2 Template driven DB querying

Two major reasons dictated the choice of storing our data using a relational database instead of native XML. Firstly, the performance of such a system will enable us to store a larger set of information. Secondly, helper applications running either in batch or background mode could operate on the same dataset without having to deal with locking issues. However, in our use of multimodal information delivery, we were convinced of the advantages of XML technologies. The relevance of the XML framework is particularly convincing for transforming and re-ordering the information.

In order to achieve the extraction of information from a database to XML, two kinds of solutions [6] were considered :

1. a model driven mapping: the structure of the XML produced is tightly connected to the underlying storage model (in our case relational tables),

2. a template driven mapping: the data is smoothly inserted into an XML file and the underlying storage structure is totally hidden.

Both solutions could either be natively implemented in our servlet framework thanks to an ad-hoc Java program or thanks to a more sophisticated scheme using a configuration file. We thought that a template driven solution using XML configuration file was best suited to our needs. The PXSLServlet [12] appeared to be a good candidate for filling both requirements.

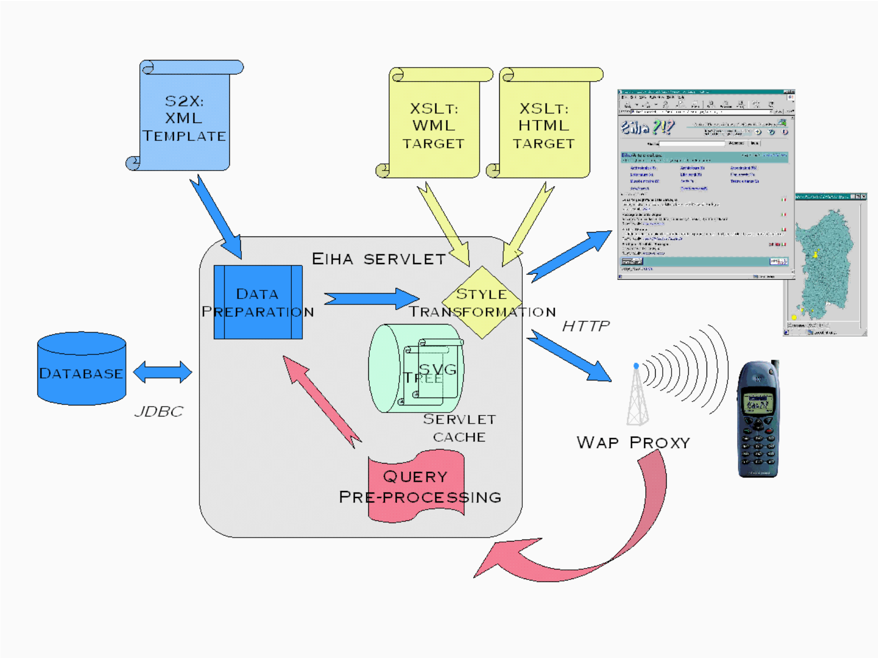

Figure 4: EIHA?!? Architecture

```
- <S2X source="CATEGORY" fields="CATEGORY_ID,CATEGORY_NAME,ABSTRACT,RESPONSIBLE_ID" condition="WHERE
    CATEGORY_ID=$/s2x.catid">
  - <S2X source="@../" fields="CATEGORY_ID,CATEGORY_NAME,ABSTRACT,RESPONSIBLE_ID">
    - <category>
      - <fathercategory>
        - <S2X source="SUBCATEGORY_NAME_VIEW" fields="CAT_ID,CAT_NAME,SCAT_NAME,SCAT_ID,NBRES"
            condition="WHERE SCAT_ID=$/s2x.catid">
          - <S2X source="@../" fields="SCAT_NAME,SCAT_ID,NBRES">
            - <category-overview>
                <id-category>$CAT_ID</id-category>
                <name>$CAT_NAME</name>
                <number>$NBRES</number>
              </category-overview>
            </S2X>
          </S2X>
        </fathercategory>
        <name>$CATEGORY_NAME</name>
      - <S2X source="PERSON" fields="PERSON_ID,FIRSTNAME,SURNAME,E_MAIL_USERNAME,E_MAIL_DOMAIN"
          condition="WHERE PERSON_ID=$../RESPONSIBLE_ID">
        - <S2X source="@../" fields="PERSON_ID,FIRSTNAME,SURNAME,E_MAIL_USERNAME,E_MAIL_DOMAIN">
          - <responsible>
            + <person>
            </responsible>
          </S2X>
        </S2X>
        <abstract>$ABSTRACT</abstract>
      + <resourcelist>
      + <subcategories id="AAA">
      + <relatedtopics>
      </category>
    </S2X>
  </S2X>
```

Figure 5: An S2X file

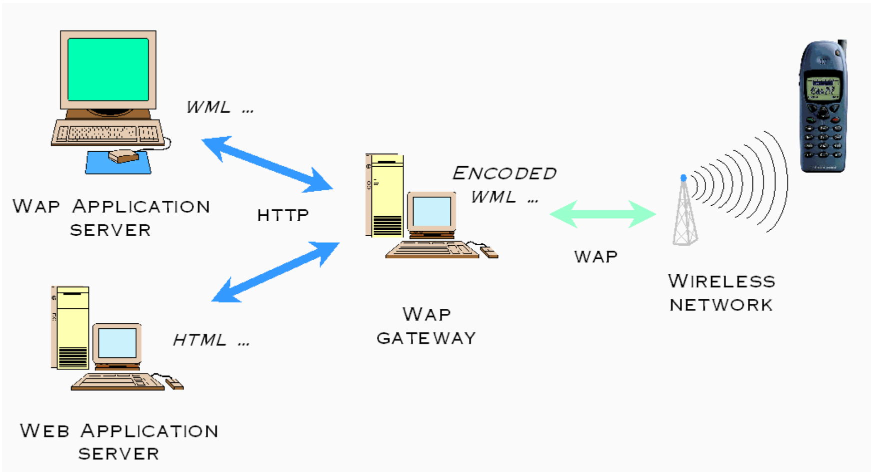

Figure 6: Wap Architecture

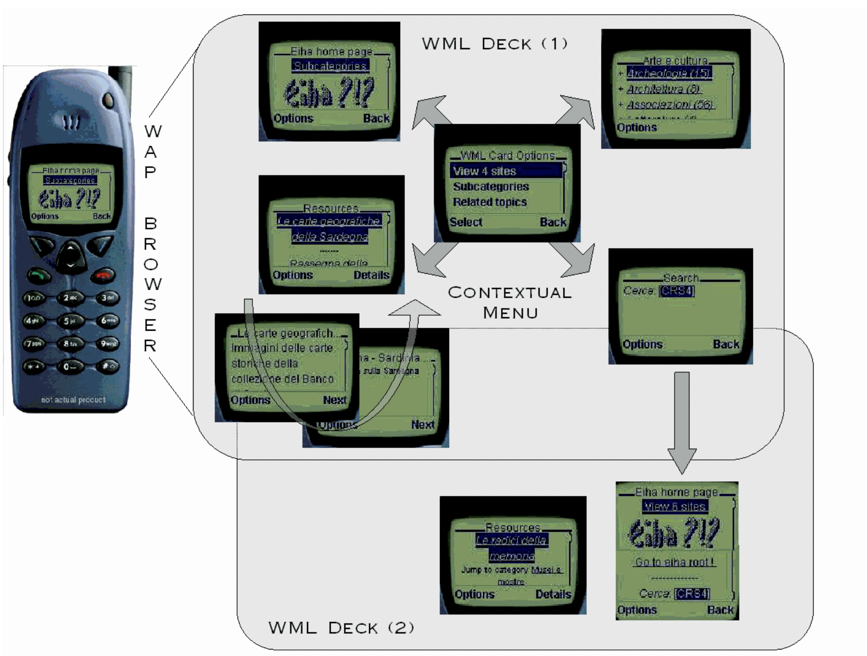

Figure 7: A synthetic view of Eiha?!? WML

The main advantage of the PXSLServlet is to use a template based mapping in XML. A special kind of node, the `S2X` tags, enabled us to specify database queries, while a simple syntax allowed us to use dataset results in the XML template 5. This format was used to complete the template in an intuitive and declarative way, without having to bother with specific JDBC programmation. Finally, in order to customize the queries, we added the possibility of including variables taken from a URL [2].

4 WAP e WML

Wireless devices, from mobile phones to PDAs, suffer various limitations tied to their batteries, autonomy and dimensions. These constraints suggest smaller CPUs, low memory, small and often black and white displays, and simplified input devices for data entry. At the same time, mobile networks are not as performant as traditional networks, and often provide low bandwidth and higher latencies.

4.1 The Wireless Application Protocol

As an anwser to these problems, an industrial consortium comprising of the main players in wireless communication, have proposed the WAP platform [9]. The wireless application protocol, uses existing technologies but also introduces new concepts, giving an environment that supports access to distributed services troughs mobile phones or PDAs, and often overriding the limitations coming from the mobile networks. The WAP architecture, as shown in figure 6, could be seen as an extension of the web framework dedicated to mobile devices.

In order to enable access to services from mobiles, a server is dedicated to translating HTTP based services to WAP-based clients. This server called a *WAP gateway* is responsible for translating a text-based markup language (like HTML) to an encoded version of WAP documents. The advantages of this encoded version are twofold: (1) it considerably reduces the amount of bytes needed for data transfer, and (2) it is assumed to produce error free documents. These two advantages enable lead to lightweight browsers, so called *micro-browsers*, which utilise low bandwidth protocols such as GSM.

[2] this capabilty was later added in release 0.3

Obviously, the standard web based services will not be sufficient with an automatic translation scheme, and therefore another family of services have been introduced, which take into account mobile device limitations. One such service is the native WAP hypertext language, the Wireless Markup Language (WML).

4.2 The WAP languages

In order to quickly introduced WAP languages, we will draw a parallel with web languages. The WAP alternative of HTML is called the Wireless Markup Language [10]. WML is an XML application [2] developed in order to distribute contents and services to mobile phones and more generally on devices with the following features: limited dimensions display with low resolution, limited input capacity, limited processing capacity and low bandwidth.

Since WML has been tailored to be an hypertext language, a lot of basic concepts are common with HTML (such as the concept of a URL and an Anchor). However, even if the capabilities of WML are a subset of HTML capabilities, we have to highlight that WML is not a subset of the HTML language.

One of the more important differences between these languages is the basic transfer unit. While we can roughly draw an equivalence between the HTML content (the body) and the displayed page, WML proposes the notion of a *deck* as a basic transfer unit, while the unit of display is a *card*. A deck re-groups a set of cards identified by their respective card IDs, and a shared template. The template enables the sharing of a common structure for the set of cards available in the deck. The presence of several cards in a deck enables a user to browse a set of cards without having to connect each time to the network – hence avoiding latency.

A scripting language, called WMLScript, is also supported. It shares the syntax of JavaScript but only offers a subset of the operators and functionalities. WML offers also a timer feature which enables various triggering actions (such as card switching and scrolling) based on timed event.

The advantages of using our XML framework was particularly efficient in developing the WAP based version of EIHA?!? . Two main characteristics must be considered while developing a WAP service:

1. Contrary to web browsers, the wap gateway only accepts compliant WML files in order to compile

them.

2. As shown in figure 7, the WAP version requires the developer to elaborate in quite a complex manner the XML data (such as multiple cards and contextual menus).

Since our production process is using an XSLT engine, these characteristics were addressed quite easily. By construction, XSLT is already producing a well formed XML, consequently the produced WML is DTD compliant and even the first version of EIHA?!? was practically error free. Although several characteristics of XSLT could be considered annoying (especially the lack of support for dynamic transformation), it is really easy to modify the result tree based on the structural nature of the input tree. We will recommend a similar approach to easily develop WAP based services.

5 Geographical Module

While an early prototype of our system was using a Java applet to display the resource location in a geographic map, the latest version relies on the SVG plugin developed by Adobe [5]. The SVG format [4] is a current working draft of the W3C. SVG is an XML dialect which relies on a specific DTD and which supports a DOM [1] based interaction. This format offers a means to distribute and display vector based drawings. Interactions with the browser makes use of the DOM event model and enables interaction to be controlled with JavaScript.

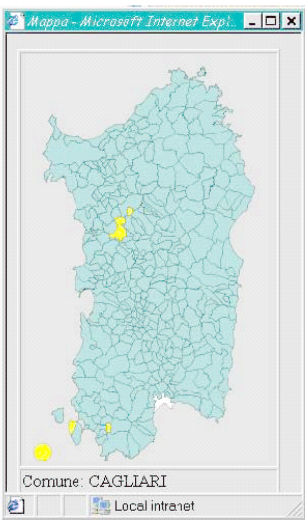

Figure 8: A set of comunes spotted on the EIHA?!? map

The paths of comunes have been extracted from a GIS system [8], and stored in our database. As shown in figure 4, the SVG file is automatically generated and cached by the servlet.

Since all web resources were geo-referenced during the classification stage, we could make a bidirectional link between the map and the description of resources. This link was able to show/hide resources on the web pages from the map, and reciprocally to spot the resources on the map (as illustrated in figure 8).

Conclusion

The realisation of the EIHA?!? site using this framework gave us a lot of interesting feedback on different aspects of XML related technologies. We summarise the lessons learned in this section.

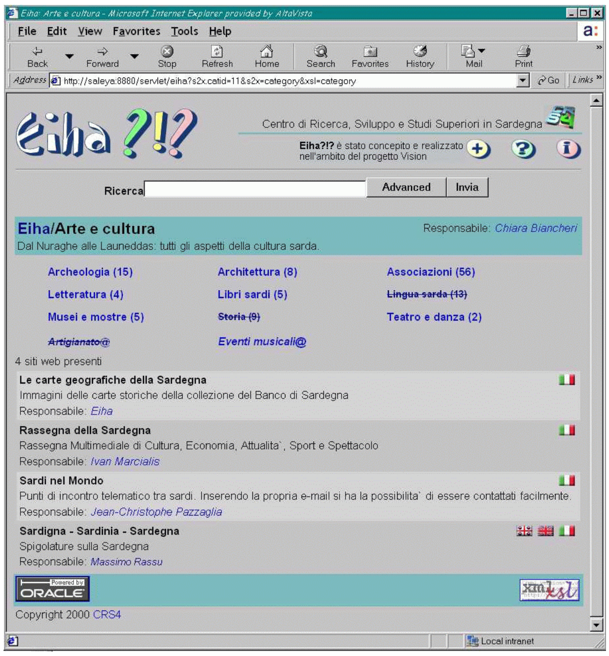

Figure 9: The standard version of EIHA?!?

Early on in the project, we were commited to using XML as a middleware language. In order to develop our architecture, we spent a substantial amount of time to define our set of DTDs. This work gave us the opportunity to clearly define the syntax between our modules. However, we found that the inclusion mechanism of DTDs provides poor support for abstraction and re-use.

It is quite easy to introduce a relational database into an XML based system. Template driven querying tools are powerfull and versatile for extracting information. It remains that two seperate data models, DTDs and database schema, have to be maintained. Since we were using XML to represent knowledge and not a document, we believe that the database schema could be semi-automatically extracted from the set of DTDs. However, annotations of DTDs are required in order to fully automate this mechanism. These annotations would enable us to tune the data types used, and to exploit the database constraints mechanism. Large projects must definitively follow this avenue in order to minimize the data structure maintenance effort.

One of the claimed benefits of XML is to expose the information together with meta-information or structure. Obviously, this feature is not directly available in our current architecture which delivers preprocessed information. However, a backdoor can facilitate access to an XML view of the information. We wanted to highlight that this possibility is avalaible not only for *standard* data, but also for geographic information. We believe that the SVG format is for geographic data, in a similar way to a formatting object being for textual data.

References

[1] Document object model (dom) level 1.0. http://www.w3.org/TR/REC-DOM-Level-1/, October 1998. W3C Recomandation.

[2] Extensible markup language (xml) 1.0. http://www.w3.org/TR/REC-xml, February 1998.

[3] Extensible stylesheet language (xsl) 1.0. http://www.w3.org/TR/xsl/, February 2000.

[4] Scalable vector graphics 1.0 specification. http://www.w3.org/TR/SVG/, March 2000. W3C Working Draft.

[5] Adobe. Svg technology preview web site. http://www.adobe.com/svg/viewer/install/, January 2000.

[6] Boutter, R. Xml and databases. Tech. rep., Technical University of Darmstadt, 2000.

[7] Clark, J. Xsl transformation (xslt) 1.0. http://www.w3.org/TR/xslt/, November 1999.

[8] ESRI. Gis and mapping software. http://www.esri.com, January 2000.

[9] Martin, B., and Hjelm, J. Wap forum - w3c cooperation white paper. http://www.w3.org/TR/NOTE-WAP, October 1998.

[10] Nokia. *WML Reference, version 1.1*. P.O. Box 226, FIN-00045 NOKIA GROUP, September 1999.

[11] PROJECT, J. Tomcat, is a world-class implementation of the java servlet 2.2 and javaserver pages 1.1 specifications. http://jakarta.apache.org, March 2000.

[12] Tchistopolskii, P. A. Pxslservlet. http://www.pault.com/Pxsl/, March 2000.