Abstract

This paper proposes a framework for the development of XML-based digital libraries. The goal of this work is to exploit the advantages that derive from the new XML standard in the field of digital libraries. The proposed framework is presented through a digital library architecture of University data. The diverse nature of such material dictates that any attempt to organize it should be based on standards. Based on the proposed framework, a prototype implementation is under development in our University, from which many useful conclusions have been extracted and presented throughout this paper.

1. Introduction

It is common knowledge that access to information resources today is much easier than it has ever been before. In the past, the only option for knowledge acquisition was the library, which is defined as "a catalogued repository of mass-produced physical objects (books, journals etc) ... It is local and generalized and is supported as a line-item in an agency, institutional or corporate budget" [26]. Today, things are much different. Besides the conventional libraries, another source of accumulated knowledge has become available. The enormous amount of information that exists on the Web has transformed it to a universal public information repository. The Web has brought such an impact to knowledge sharing that has forced traditional information providers like libraries and journals to publish their content to the Web. However, the fact that the Web is growing at a phenomenal rate has made it very difficult, if not impossible, to manipulate. One of the main reasons for this inconvenience is the fact that it is based on a markup language (HTML) that fails to accurately describe its content and structure. Therefore, modern Web applications like digital libraries aiming at bringing order to the chaotic Web should be based on new technologies like the eXtensible Markup Language (XML). One could argue though that the fact that XML is a descendant of SGML predetermines a short and unsuccessful future for this technology at the Web. SGML has been around for more than 15 years and nonetheless never managed to dominate the Web. This was primarily due to the complexity of the language and the resulting overhead required using it. Having these thoughts in mind, XML did away much of the complexity of its ancestor and kept only the parts needed for use in Web publishing [27].

Apart from the above general rule that applies to almost all Web applications, there are a number of specific requirements that should be taken under consideration for the development of a Web-based digital library. Thus, a digital library's storage system must be capable of handling a large amount of data in a variety of formats and accessing this data as quickly as possible [6]. Interaction with its users should be easy but powerful and information retrieval should be accurate. For this purpose, the employment of metadata [7,8] facilitates a solution that has been tested through time in a number of digital library applications. A digital library is not a static
application and it cannot survive in time without proper maintenance and regular updating. Scalability is also a key issue in the design of a digital library and should be adequately supported. The underlying architecture should be separate from the stored content. This is a fundamental rule that enables easier administration and maintenance [9].

In this paper, a framework is provided that facilitates the development of Web-based digital libraries. It is based on the XML standard and its corresponding recommendations. The framework is presented through a digital library architecture of University data (teaching material, thesis, etc). One of the goals of this work is to identify the benefits that derive from the employment of XML to the field of digital libraries. More specifically, XML allows the Web client to present different views of the same data to different users [4] since it separates content from presentation. Interoperability between different components of a digital library can be easily established since XML defines a common format for computer-to-computer interaction [5]. In the implementation of a deriving architecture, XML data structures that are both human readable and computer understandable reduce complexity. Finally, the fact that XML supports structuring complex data in a hierarchy, facilitates quick transactions on digital library material.

The rest of this paper is organized as follows: Section 2 refers to similar XML-based digital library projects that address the issues that have already been mentioned. Section 3 covers the proposed framework, which is presented through a digital library architecture of University data. In this section, the basic modules that compose the three-tier architecture are defined and explained. Separate subsections are devoted to the building block of the digital library, the document. Finally, section 4 summarizes the proposed framework and outlines the deriving conclusions from the employment of XML in a digital library environment.

2. Related work

The importance of digital library technologies has already been recognized from many major organizations and governments all over the world [28]. Since 1994 where the first Digital Library Initiative (DLI) was launched in the US, several projects are devoted in designing functional digital library applications. This section focuses on related digital library projects based on XML.

Aloisio et al. [5] propose a CGI-based, three-tier architecture of remote sensing archives based on XML. As compared to similar CGI-based architectures, the authors of this work dictate that XML-based services can be used in the context of previous requests, this way facilitating richer human-computer interaction. On the other hand, CGI scripts are stateless resulting in cumbersome, stand-alone requests.

The architecture proposed by Gupta et al. [10] is based on data objects and links. Data objects constitute the underlying repository and may be related to other data objects via typed links. The authors claim that due to its navigational nature, their “object-oriented” approach is suitable for modeling scientific information.

The SOMLib approach [11] provides access to distributed repositories based on a neural network model. Standard Web-based interfaces are incorporated and “computational intelligent tools” provide high-level functionality including scalability, user-profiling and integrated searching.

THETIS [12] is a Web-based, distributed environment consisting of one or more underlying repositories. Each server node contains a search-engine and a retrieval engine module. In this architecture, distributed queries against repositories whose objects are described by different metadata sets, are supported in the intersection of the sets.

A more "up to date" approach is followed in FEDORA [29]. Although this application is based on CORBA rather than XML, it is worth mentioning since it claims to deal with many of the issues previously mentioned in the field of digital libraries. Based on the well-known dienst protocol [29]. FEDORA proposes a general digital object model and repository architecture for encapsulating and securing multimedia content. It supports aggregation of distributed, heterogeneous types of data along with external rights management schemes. Extensibility of content types is provided as well as access to multiple views of the same object’s data.

Finally, the Networked digital library of Theses and Dissertations (NDLTD) [1,2] in the US is a inter-University digital library based on the SGML-XML and Z39.50 standards. In this work, an argument in favor of XML is presented, claiming that it offers a great deal of flexibility compared to other alternatives such as storing document descriptions in a relational database.
3. The proposed framework

The proposed framework supports web-based, three-tier architectures based on XML. It is going to be presented through a deriving digital library architecture of University data. A prototype implementation of this architecture is under development at the University of Piraeus. With trivial modifications, it can be applied to organize data collections of many kinds.

The conceptual design of the digital library is based on the organizational structure of our University. Thus, the back-end consists of a number of nodes that correspond to the departments within the University. Each department is planned to maintain a repository with department-specific documents. Although documents rarely have anything in common as we shift from department to department, the user is presented with a homogeneous and highly organized repository structure. This is due to the fact that XML facilitates management of documents with diverse structure/content.

The middleware consists of three modules that support the core functionality of the digital library. These modules are namely:

- the XML Descriptor module
- the Collection Constructor module and
- the Search-Engine module

supports interaction with end-users and provides extended functionality as compared to similar search engine interfaces. The Search Engine module is responsible for locating documents that satisfy users' requests. It is an XML-oriented program that accepts queries following the XQL specifications [18] and returns results in the form of XML trees [3].

A cache database is also maintained to improve the overall performance of the digital library. Another component that resides in the middleware is the schema repository. It stores the necessary schemas that are used from the authors as templates and definition files for their documents.

The client layer is actually an XML-capable Web browser enhanced with the ability to render XML files (e.g. IE 5). A general description of the proposed framework is presented in figure 1, providing knowledge for the various components of the digital library.

3.1 The XML Descriptor module

The XML Descriptor module facilitates Resource Discovery for the digital library users. As stated in Iannella et al. [13], "Resource Discovery is the term commonly used to refer to the exercise

![Diagram of digital library architecture](image)

The XML Descriptor module contains the various schemas that define the structure of each document in the back-end. According to the XML specification, each document can follow certain rules (e.g., constraints) referring to its structure and content. The Collection Constructor module accepts documents from individual repositories and merges them to a unified result-list document. This module of locating, accessing, retrieving and managing relevant resources for a user". In this context, each instance of the above module describes the contents of its underlying repository to the module that made the request. A similar approach is followed in NDLTD [1,2], where the Searchable Database Markup Language (SearchDB-ML) is introduced.
The structure of the XML Descriptor is defined in its referring schema (in XML terms known as a Document-Type-Definition .dtd file), which is named XML_Descriptor.dtd. This schema migrates to all nodes of the digital library in the form of a file named XML_Descriptor.xml, as shown in figure 2. A subset of the prototype schema (for simplicity reasons) is presented in figure 3a along with its deriving .xml file (figure 3b).

The elements of the XML Descriptor module are actually metadata [7,8]. A similar set of metadata is successfully being employed to describe documents that belong to another collection of University material [14] in Greece. The need for employment of metadata in a digital library with such a diverse content as the one described in this paper is becoming widely recognised. The big advantage of metadata is that they make large portions of information available both for human and computer understanding [4].

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE XML_Descriptor SYSTEM "XML_Descriptor.dtd">
<XML_Descriptor> <!--The root Element-->
  <University name = "UniPi"> <!--The University in which the node of the digital library belongs--> 
    <Department name = "Informatics"> <!--The Department it refers to--> 
      <Lecture> Compilers </Lecture> <!--The Lectures that are taught at the department-->
      <Lecture> Computer Architecture </Lecture>
      <ThematicArea> Computers </ThematicArea>
    </Department>
  </University>
</XML_Descriptor>
```

Figure 3b: A prototype XML_Descriptor.xml

The employment of XML technology provides the infrastructure for a simple and straightforward way of locating repositories that contain useful information. Indeed, through the use of adequate modules, the user of the digital library has the ability of deciding whether he/she will select repositories according to:

- the title of the department,
- the abstracted thematic area or
- the lectures that he/she are interested in.

This is accomplished through the utilisation of the content-specific tags Department, Lecture and ThematicArea from the appropriate modules of the digital library (i.e. search-engine module).
3.2 The Collection Constructor module

Upon initialisation of interaction with a user, the aforementioned XML Descriptor is invoked at each repository and is requested to transmit its corresponding XML-tree to the Collection Constructor module that acts as a "broker". The collection Constructor module merges all individual XML-trees to a unified XML-tree according to the XML-Data specification [16]. This pull model [15] that is applied for the construction of the resulting XML-tree is preferred to a push model because it guarantees the consistency of the underlying document collection. If some nodes of the digital library are unavailable at the time of a transaction, they simply won't be fetched. On the other hand, depending on the population of the individual nodes, the pull model could prove to be rather slow. However, there is clear evidence that software managing XML data efficiently will soon be produced.

After the initialisation, the user is requested to decide how to classify the collection. There are three ways available (supported by the XML Descriptor module): through a list of the lectures that are being taught, through a list of the individual departments and through a list of the predefined thematic areas. By selecting one or more individual items of a classification (e.g. the departments Sociology and Psychology from the "department" classification), the user actually narrows the search to the nodes that are associated with the above departments. This way, the searching process is faster and with more accurate results. Alternatively, the user can browse through the documents that reside in the selected nodes of the digital library. The ability to have multiple views of the same data derives from the fact that every XML document corresponds to a XML-tree that can be manipulated in many ways using appropriate XSL stylesheets [17].

Another feature of the proposed digital library architecture is scalability. When further requirements for the digital library are witnessed, the XML Descriptor schema can easily be updated with a common text editor while preserving compatibility with repositories in existing nodes.

3.3 The Search Engine module

After focusing in a certain subset of the entire document collection, a user submits his/hers search criteria to the Search Engine module. The retrieved search criteria will be translated to a query that is formatted according to the XML Query specifications [18]. The resulting query will be addressed to every local node that has been selected to participate in the search during the previously described first stage of the searching/browsing process. The documents that satisfy the given query will return a subset of their corresponding XML-tree containing some "meta" information about the document (i.e. category, author, title, creation date) and, most importantly, the context in which the query term was found in the document. For example, in the case where a user enters "Java" as a search term, a selected document reference in the search-result list about a users manual of a program written in Java, would contain a phrase stating that Java is the implementation language of the program:

- **Category**: program, **author**: John Smith, **title**: e-bookstore, **creation date**: 13/03/2000, **implementation language**: Java.

In another case of a technical report about Web server technologies where Java is met in the section of dominating technologies, the corresponding document reference would be:

- **Category**: technical report, **author**: John Scholes, **title**: Web Server technologies, **creation date**: 19/12/1999, **section**: dominating technologies-Java.

All document references of documents that satisfy the search criteria are merged to a composite XML-tree by the "merger component". This component accepts various XML-trees from the nodes of the back-end and composes a global one at the middleware that corresponds to the requested search-result list. The deriving tree is delivered to the client. Using XML capable software at the client side, the XML-tree is transformed to an XML file that is presented to the user. Each link of this file can be activated to lead (using XPath technology [19]) to its associated resource(s) that reside at the Back-end. This scenario is illustrated in figure 4. There is no need to expand the searching process beyond XML files since every file is associated directly or indirectly with an XML file.

The programming language that is potentially best suited to manage XML data and consequently implement the functionality that will be provided by the various modules of the digital library, is Java. Its platform-independent nature as well as its specialisation to Web applications, renders Java as the most appropriate language for such a task.


3.4 Back-end structure

A Web server is maintained at every node of the digital library. It registers the node to the middleware through the previously described XML Descriptor module. Considering the digital library as three broad classes of elements (data, metadata and processes) [20], each instance of every class should be referenced and identified consistently, through permanent names and identifiers [21]. The fact that every node of the overall collection is wrapped around a dedicated Web server provides the necessary namespace that settles for the uniqueness of all documents in the collection.

Furthermore, conflicts between the component names (element names, attribute names, user-defined entities etc) of the various XML files, are prohibited through the employment of XML namespaces [22]. Each XML file is associated with a namespace that includes the definition of the various XML component names that it discusses. Of course, the employment of a namespace schema requires not only the development of such schemas from experienced people within the academic community, but also their use from the authors of the digital library.

As it will be shown later, the proposed digital library framework supports various formats. Nevertheless, the fact that every file in the back-end is associated with a XML file, dictates that the modules residing at the middleware need to know the location of only these XML files. A common structure should therefore be followed at every node of the digital library for proper and transparent information retrieval. In the prototype implementation, we have chosen to follow a repository infrastructure similar to the one that is introduced by the dienst protocol [29]. Although it is based on a rather slow flat file system organisation, it remains simple to use and independent of commercial and therefore expensive database management systems.

![Diagram](image)

Figure 4: Explaining the Search Engine module

3.5 Document structure

Each document in the collection may have multiple files of various formats that constitute it. Since these files are created in formats that don't describe their content, an additional meta-file (i.e. metafile.xml) accompanies each such file. Its elements are essentially metadata that provide a description to the content of the "foreign" (pdf, ppt, doc, etc) file they refer to. Using XPath technology [19], the various sub-components of the document are referenced from links that belong to a single file that is named “composite.xml”. The overall document structure is presented in figure 5. Starting from the composite.xml, the user is able to navigate through the rest of the associated files and annotate, evaluate and make comments on them without having to modify the original read-only files.

The employment of XPath technology, is boosting the digital library's interaction with its users. Some of the features supported by this technology are links that lead to multiple destinations, bi-directional linking, attached roles, descriptive titles to the various ends a link attaches and so on. The only drawback of such an approach is the lack of dominating tools on the market that will automate the procedure of creating and
managing such interactive documents. However, it should be taken under consideration that although XML is a new and consequently not stable technology, it is already adopted from virtually all major companies that produce Web products.

The structure illustrated in figure 5 is common for every document in the collection and constructed and stored in the Schema repository, each one corresponding and serving as a template to a specific type of document. So far, the following document classes have been identified:

- Teaching material (presentations, case studies)
- Technical reports
- Theses (MSc, PhD)
- Final year projects

Figure 5: The Document components

3.6 Document types

As mentioned previously, the proposed digital library framework is suitable for storing any type of resources that can be referenced from a URL. Nevertheless, in order to exploit the advantages that derive from XML technology, authors are encouraged to create their documents in XML. For this purpose, a set of schemas (Document-Type-Definition – .dtd files) are followed at each node of the digital library.

Specifically, XML files are parsed with adequate parsers (e.g. XML4J [23]) and produce tree-like structures, where every node of the tree corresponds to an element of the parsed file. Using databasespecific XML software (e.g. Excelon [24]), every XML file of the document collection is stored in its tree form for faster access. The search engine module needs to search just the XML files, since information about others is encapsulated in the aforementioned XML metafiles. These “other” files can be stored anywhere in the node, as long as they are visible to the Web server and accessible through a URL. According to the proposed architecture, every node of the digital library will have its content stored in a local database, as described.

4. Conclusions

We have presented a framework for the development of XML-based digital libraries through a distributed digital library architecture specifically designed for University data. Throughout this work, we came to the conclusion
that the XML standard supports functionality that is common to the field of digital libraries. Consequently, certain key issues for the development of a digital library like scalability and namespaces are already solved. Additionally, the employment of XML enables manipulation of metadata in an easy and unified way and provides the infrastructure for the various components of the digital library to exchange data in a common format. The fact that XML distinguishes between data and presentation enabled us to design a system, where users of the digital library can have multiple views of the same data according to their specific needs. Another feature of the proposed architecture is the fact that searching through the collection is performed in two stages: In the first stage, users focus on the various nodes of the digital library that are more relevant to their needs. In the second stage, only the selected nodes participate in the search, thus improving the overall performance of the search procedure.

However, claims that any one technology has solved all of the issues posed in the design and implementation of digital libraries fail to address the entire problem [20]. Access control policies and copyright issues are not mentioned in this paper. We should also keep in mind that a digital library is not only a matter of solving technical issues. Although the implementation phase of the proposed framework is still under development, we have already realised that the success of such a project depends highly on the involvement of communities that will benefit from its use. In our case, students, researchers and professors should be encouraged to provide the necessary schemas that will be used from the rest of the University community to create intellectual work, not just digital data.

References

17. XSL Transformations (XSLT) Version 1.0, Nov. 1999, URL: http://www.w3.org/TR/xslt
18. XML Query Requirements, Jan. 2000, URL: http://www.w3.org/TR/xmlquery-req
19. XPath Language (XPath) Version 1.0, Nov. 1999, URL: http://www.w3.org/TR/xpath
22. XML namespaces, Jan. 1999, URL: http://www.w3.org/XML/9901names
23. XML Parser for Java: another alphaWorks technology, URL: www.alphaworks.ibm.com/tech/xml4j