Design and Exploitation of a Markup Strategy in a Digital Library

Alejandro Bia
abia@dlsi.ua.es
Miguel de Cervantes digital library

Rafael C. Carrasco
carrasco@dlsi.ua.es
Dept.Lenguajes y Sist.Informáticos

Universidad de Alicante, E-03071 Alicante, Spain

"This is a fascinating period in the history of libraries and publishing. For the first time, it is possible to build large-scale services where collections of information are stored in digital formats and retrieved over the networks."[Arm00]

Abstract This paper reports ongoing research and development activities being done at the Miguel de Cervantes digital library [1] in the field of text markup and derived applications, like automatic transformation of documents to different formats, and complex searches performed upon small textual objects (those defined by the markup scheme).

It also includes a brief survey of works done on Named Entity Recognition that can be applied to automatic markup. Finally there are some comments on the research lines we intend to follow concerning information retrieval and filtering from structurally marked up texts.

1 Introduction

The huge development of information technology has motivated the appearance of a new type of libraries, called digital libraries, whose essential difference with respect to traditional libraries is the way information is supported and issued to readers. Harter [Har97] concludes that, although the term digital library is rarely defined and it has been applied to an extraordinary range of applications, the common factor is digitization. This "digital coherence" allows all the objects in a digital library – sounds, images, texts – to be treated in a similar way. The difference in essence may be digital support, but the real differences, the ones that make digital libraries worth, are in functional aspects: things that can be done with digital libraries in an outstanding different way.

The use of computer means to store the texts, in addition to reducing the physical storage space and to facilitate their distribution, allows fast queries to be done, whose cost would be extraordinary if performed manually on paper books. On the other hand, to put the information on a network like Internet, makes it accessible to everybody worldwide. Arms mentions these and other reasons to justify the emergence and proliferation of digital libraries [Arm00], as for instance economical reasons, permanent availability of the information on the Web, formats that offer more possibilities than the traditional printed text (e.g. hypertext), as well as ease of storage and preservation of the documents in digital format.

These outstanding capabilities should not be limited to sorting and retrieving whole documents. We expect DLs to be capable of retrieving meaningful document fragments, what we call *textual objects*, like for instance paragraphs, poems,

[1] http://cervantesvirtual.com/

names, dates or headers, and to process them in a more interesting way, to allow the generation of literary concordances, statistical word use analysis, and complex searches.

The complexity of these goals and the huge quantity of information contained in these libraries demand new techniques for document classification, processing and retrieval.

On one hand, it is obvious that the mere accumulation of texts leads to a limited-use library. On the other hand, traditional databases, useful for cataloging purposes and for catalogue searches, are not adequate to store the contents of the texts (i.e. the *textual objects*: structural parts of texts). Traditional databases are best suited for fixed length data fields, and not variable length textual objects, structured in hierarchical way [2].

In this picture, *structural marked-up text* occupies an intermediate place between databases and free text. This type of structured information consists of tagged texts, this is, texts that include special *marks* that provide information about its features (also called metadata) while determining hierarchical inclusion.

Marks generally indicate the nature of the object (e.g. in poetry a verse line, a stanza, a poem, in drama a character or a speech line, in narrative a chapter, paragraph or sentence), and also specific attributes like the language in which a textual object is written, or the rendering. This information is useful at the time of applying processes to text: in this way, programs will be able to take decisions being based on this metadata (for example, to perform correct hyphenation based on the language attribute, or to generate an index of first lines of poems).

It is also possible to include catalog information within documents in the form of a document header, as suggested by the `TEI` [3] scheme [SMB94, Bur95, PSM]. We also call this kind of metadata, catalog metadata, to distinguish it from markup.

We can discuss whether this catalog metadata should be part of the document, be in a separate database, or both, in which case we have a duplication problem were we have to determine which is the main source and which the copy, and provide strict control to avoid inconsistencies. We can see strong advantages for both alternatives.

2 State of the technology in brief

The success of Internet and the `HTML` markup language —that allows a limited type of structured texts— has given a notable impetus to the studies related to text markup. Among them, we can mention the following:

- Definition of standard markup methods [Der99].
- Study of theoretical characteristics [BKW98, Aho97] of these methods and the design of procedures to aid markup, definition and transformation of document models [AMN97, Mur97a, Mur97b], etc.
- Use of structural markup for interesting tasks like information retrieval or extraction [LR98, MGM99].

The most standard methods of markup are those compliant with the `SGML` international standard (*Standard Generalized Markup Language*, ISO 8879). In particular, the `XML` (*Extensible Markup Language*, a standard of the W3C (*World Wide Web Consortium*) compatible with `SGML` is a promising standard designed

[2] Object Oriented Databases may be a viable alternative in this respect

[3] Text Encoding Initiative

specifically for the information exchange through Internet. For digital libraries are of special interest the recommendations of the TEI (*Text Encoding Initiative*, that defines an instance of the previous markup standards (i.e., a set of agreed SGML/XML tags) for the structuring of texts. The original TEI scheme was SGML compliant, and is now being adapted to XML (there is a version of the TEI-lite [BSM95], a subset of the TEI, already in XML).

3 The Miguel de Cervantes digital library

Our library covers many different areas, from a "library of voices" up to academic thesis. However, the vast majority of our present digital books are public domain hispanic classics, from the 12^{th} century up to these days, including narrative, theater, poetry, history and other subjects. Many professionals and technicians take part in the development of our digital books: librarians, scanner operators, correctors, markup specialists and computer technicians.

So we can distinguish two kinds of digital production processes, according to the volume and diversity of the contents:

3.0.1 Specialized portals

This are web portals dedicated to special subjects like rare books, given authors, literature of a given country, library of voices (recorded readings). They are varied in content and appearance. They contain multimedia material (audio, video) and a lot of graphic design. In these ones, content is created and changed periodically, but the amount of information is limited in size. These type of varied publication, requires a lot of craftsmanship, incompatible with massive production.

3.0.2 Traditional books

This are the hispanic classics mentioned above. A huge number of books with almost no multimedia material (except for some books that may contain static images). Their appearance is expected to be uniform, and resemble the format of traditional printed books. Once the edition process is finished, their contents are never changed. These books, in spite of requiring a lot of care at the level of correction and editing, can be processed in a massive uniform way at the level of rendering.

The methods described below explain the kind of automation we already implemented or expect to implement for the processing of the latter.

Diagram 1 (workflow diagram) describes the whole production process of this kind of digital books.

4 Project Steps

Within the Miguel de Cervantes digital library we have an interdisciplinary team, composed of philologists [4] and computer scientists, in charge of markup techniques and exploitation of massive textual resources. This markup project within the whole DL project, can be divided in three main aspects: markup, transformation, and exploitation.

- Markup
  - Choice/definition of a markup scheme
  - Implementation of the resulting manual markup procedure

[4] Manuel Sanchez and Elena Pellus

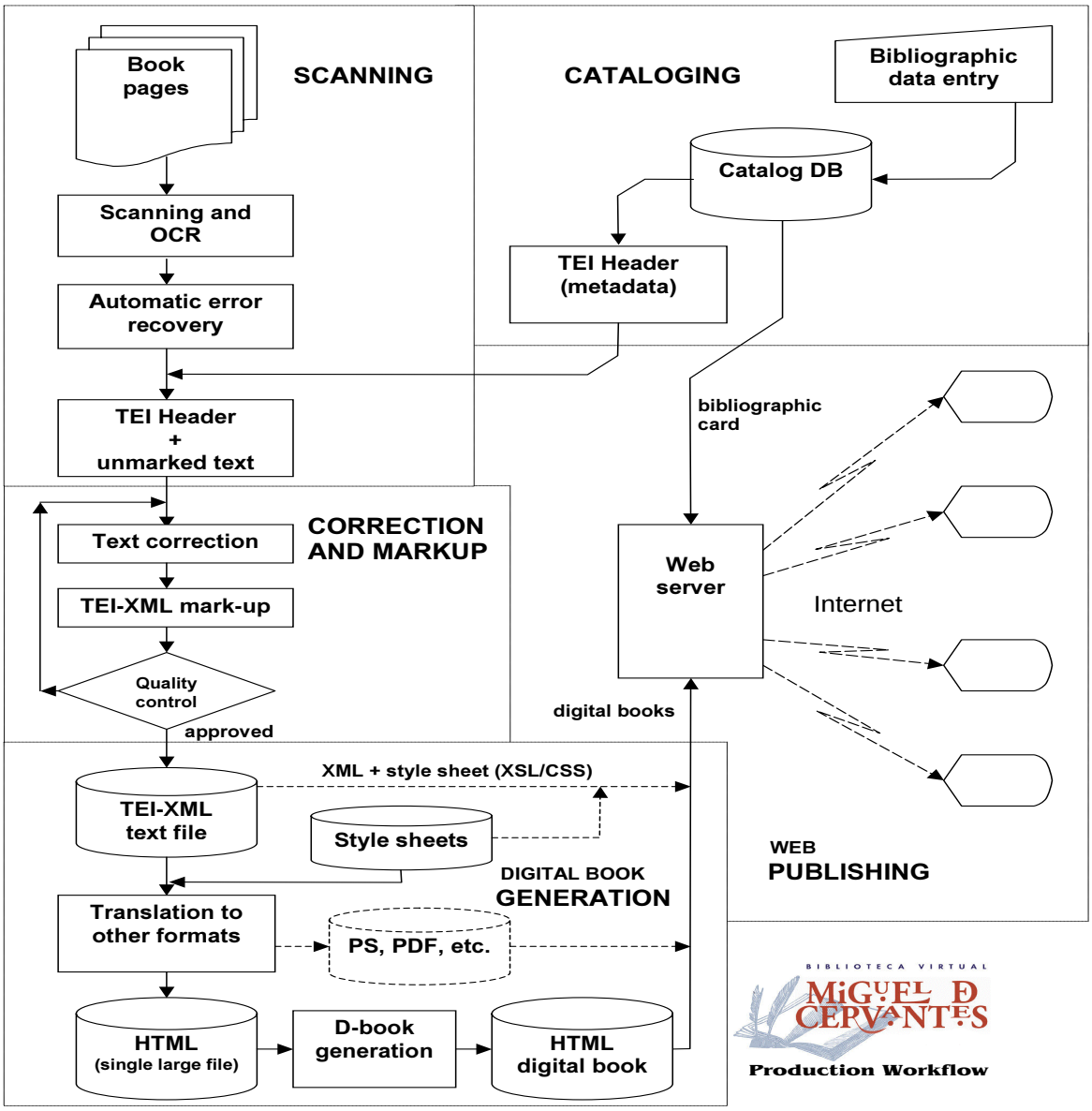

Figure 1: **The production workflow of the Miguel de Cervantes digital library**

  - Design of semiautomatic markup tools (computer assisted markup)
- Transformation
  - Design of automatic transformations to other text formats
- Exploitation
  - Definition of exploitation goals (complex search, concordances, generation of special indexes)
  - Design and implementation of exploitation tools

4.1 Markup

A principle of good practice in information technology is to avoid duplication of data, and this includes texts. So we needed to choose a format for our source documents, and all other formats should be automatically generated from this unique source. Not to follow this *unique-source* principle would result in sets of

conflicting different versions of the same document. Not to do it automatically would multiply the effort by the number of different formats to be maintained.

We decided that the format to use should comply certain requirements:

- expected to last and be easily transformed to future formats
- not be a proprietary format
- focus on structure rather than appearance of text
- be widely used to facilitate document interchangeability
- easily convertible to most commonly used formats
- directly supported by web-browsers

An important aspect of this preliminary decision concerned document preservation. The format chosen for the source documents should be expected to last as long as possible, and be easily converted to newer formats to come. This excluded proprietary formats, which are difficult to handle, change according to corporate decisions of the manufacturers, and are not easily converted to other formats when this facility is not supplied by the manufacturer.

The format should focus on the structure of the documents instead of its appearance. Appearance should be handled apart, from outside the document, in a way that changes could be applied at will over the whole collection of documents without excessive effort.

It is desirable that the format chosen should be as widely used as possible, if not an accepted standard, to facilitate document distribution and use, as well as application of standard processing tools.

The easily-adaptable-to-our-needs condition made us discard rigid formats like `HTML`, in favor of richer, extensible markup languages like `SGML` and `XML`.

An additional desirable but not mandatory feature would be that the format could be directly supported by web-browsers.

Requirement	**HTML**	**SGML**	**XML**
expected to last	Y	Y	Y
not a proprietary format	Y	Y	Y
focus on structure	N	Y	Y
be widely used	Y	N	E
easily convertible	Y	Y	Y
directly supported by browsers	Y	N	E
extensible	N	Y	Y
ease of use	Y	N	Y

Table 1: **How `HTML`, `SGML` and `XML` comply our DL requirements (Y=yes, N=no, E=expected to be)**

From the comparison among `HTML`, `SGML`, and `XML` (see table 1), we conclude that `XML` has the greater number of advantages. Steven DeRose compares this three markup languages in detail in his article `XML` *and the* `TEI` [Der99].

After choosing `XML` we had two options:

- to define our own set of tags
- to use an existent set of tags

The first option meant a lot of planning, and the result would do any good to document interchangeability. The second meant finding a markup scheme adequate to our requirements. After a little survey we've found two candidates: DocBook [Cov] and the TEI [PSM].

The kind of texts we need to markup are mainly literary and historic classical books, covering prose, verse, drama, and sometimes dictionaries. This meant we needed a very complete markup scheme. We decided in favor of the TEI scheme, as it seamed more adequate for this type of documents, and it's been successfully used in many important digitization projects [TEI00]. We have chosen to work with TEI-XML, the XML version of the TEI, for the advantages of XML described above. There is only a subset of the TEI scheme converted to XML at Oxford University, called the TEI-lite (teixlite.dtd), so we had to start with this DTD, and then make some modifications to it, while we wait for a complete TEI-XML DTD to be issued.

We had to normalize the values used for element attributes, by replacing CDATA declarations (free entry values) by a list of allowed values. This was done to prevent possible human mistakes, and to force the use of certain values needed for further processing. We also disabled certain TEI elements in our version of the TEI DTD, to simplify the markup task, to avoid mistakes, and to force following the decisions we made concerning markup options (e.g. to force the use of numbered divisions (<divN>) instead of the unnumbered alternative ( )).

We also defined ways to identify textual objects as poems and letters, which are not explicitly included in the TEI scheme, by using specific attribute values of our choice.

In the end, our markup scheme is a more restrictive subset of the TEI. Nevertheless the resulting documents are all TEI compliant.

4.2 Markup automation

We want to automate the markup process as much as possible.

Up to now, we've built simple parsing programs to convert documents from the two major commercial text editors to TEI-XML, making use of the style information to locate headings, deduct division borders, add TEI paragraph marks, as well as <text>, <front> and <body> marks. [5]

The result is an almost valid XML document, whose markup must be corrected and completed. The markup effort is reduced substantially in this way, since the most common tags (like <p>) are automatically added. But this solves only the problem of texts coming from this two commercial text editors. Output, in this case, looks like this:

```
<?xml version="1.0"?>
<?xml-stylesheet href="teixlite.xsl"type="text/xsl"?>
<!DOCTYPE TEI.2 SYSTEM "teixlite.dtd">
<TEI.2 lang="es">
   <teiHeader>
...
...
...
   </teiHeader>
   <text>
     <front>
       <titlePage>
         <docTitle><titlePart>El Abencerraje</titlePart></docTitle>
         <docAuthor>Antonio de Villegas</docAuthor>
     </front>
     <body>
       <div>
         <head type="main">El Abencerraje</head>
         <p>De Antonio de Villegas.</p>
```

[5] The <teiHeader> is generated from the catalog database.

```
        <p>Dirigido a la Majestad Real del Rey Don Felipe, nuestro se&#241;or. A&#241;o de 1565.</p>
        <p>Este es un vivo retrato de virtud, liberalidad, esfuerzo, gentileza y lealtad, compuesto de Rodrigo de
        Narv&#225;ez, y el Abencerraje, y Jarifa, su padre, y el rey de Granada, del cual, aunque los dos formaron y
        dibujaron todo el cuerpo, los dem&#225;s no dejaron de ilustrar la tabla y dar algunos rasgu&#241;os en ella.
        Y, como el precioso diamante engastado en oro, o en plata, o en plomo, siempre tiene su justo y cierto valor por los
        quilates de su oriente, as&#237; la virtud, en cualquier da&#241;ado sujeto que asiente, resplandece y muestra sus
        accidentes; bien que la esencia y efecto de ella es como el grano que, cayendo en la buena tierra, se acrecienta,
        y en la mala se perdi&#243;.</p>
...
...
...
      </div>
      <closer>
        Impreso en la noble villa de Medina del Campo, por Francisco del Canto.
        A&#241;o de 1565.
      </closer>
    </body>
  </text>
</TEI.2>
```

Something similar can be done with plain text, but with much more limited results, since there is no "style" information to infer headings or division limits. Only paragraphs, and the addition of the `<TEI.2>`, `<text>`, `<front>` and `<body>` marks. Anyway, a process like this saves time, since it adds the most common `<p>` markers, and generates an almost-always valid (or valid with a few corrections) XML document to start with. [6]

Something else that is done at this stage is converting the accented and special letters, of which the Spanish language has many, from ANSI, ASCII (or any other code) to the corresponding entities, e.g:

```
<p>Los di&#225;logos en estilo directo se han marcado con la
etiqueta Q, manteniendo las comillas. No se han indicado los
l&#237;mites de las p&#225;ginas del original.</p>
```

From this point on, what's left is to apply Natural Language Processing techniques to detect and mark textual objects. This is one of the objectives of our research efforts.

If we compare the above automatically-marked-by-simple-methods XML text, to the full marked result (below), we see the number of marks added is small, but the complexity of the decisions to add them is high.

```
    <body>
      <div>
        <head type="main">El Abencerraje</head>
        <byline>De <name type="persona">Antonio de Villegas.</name></byline>
        <p>Dirigido a la Majestad Real del Rey<name type="persona"> Don Felipe</name>, nuestro se&#241;or.
        A&#241;o de <date>1565</date>.</p>
        <p>Este es un vivo retrato de virtud, liberalidad, esfuerzo, gentileza y lealtad, compuesto de
        <name type="persona">Rodrigo de Narv&#225;ez</name>, y el Abencerraje, y <name type="persona">Jarifa</name>,
        su padre, y el rey de <name type="place">Granada</name>, del cual, aunque los dos formaron y dibujaron
        todo el cuerpo, los dem&#225;s no dejaron de ilustrar la tabla y dar algunos rasgu&#241;os en ella. Y, como
        el precioso diamante engastado en oro, o en plata, o en plomo, siempre tiene su justo y cierto valor por
        los quilates de su oriente, as&#237; la virtud, en cualquier da&#241;ado sujeto que asiente, resplandece
        y muestra sus accidentes; bien que la esencia y efecto de ella es como el grano que, cayendo en la buena
        tierra, se acrecienta, y en la mala se perdi&#243;.</p>
...
...
...
      </div>
      <closer>
        Impreso en la noble villa de <name type="place">Medina del Campo</name>, por
        <name type="persona">Francisco del Canto</name>.
        <date>A&#241;o de 1565.</date>
      </closer>
    </body>
```

4.3 Application of NLP techniques

The new marks that appear in the previous example are: `<name>`, `<date>`, `<byline>` and `<closer>`, as in the following examples:

[6] By valid we don't mean correctly and fully marked-up documents, but documents that pass the XML editor parser validation, according to a given DTD (Document Type Definition) file.

```
<name type="person">Don Felipe</name>
<name type="person">Rodrigo de Narv&#225;ez</name>
<name type="place">Granada</name>
<date>1565</date>
```

We believe some textual objects, like `<name>` and `<date>` can be detected and marked automatically. Within the NLP field, Named Entity Recognition has been studied thoroughly, and there are many approaches to tackle the problem. None of them grants an 100% recall and 100% precision [7], which makes them unsuitable for fully-automated markup, but the results obtained are good enough (around 90%) to encourage the development of computer-assisted markup tools.

4.3.1 Detecting and marking numbers, dates and names

One approach for the markup of *named entities* can be seen in [MGM99, MMG99]. They developed a set of tools for Named Entity recognition for the 7^{th} MUC [8] competition that took place in April 1998.

One of their programs `ltstop`, applies a maximum entropy model pre-trained on a corpus to solve the ambiguity between abbreviation and sentence-finals. In this way they can detect full stops in a reliable way. This technique could be applied to detect and tag sentences. In our case, the tool should be trained with a Spanish corpus.

Another tool, a transducer called `fsgmatch` makes use of different resource grammars to detect time and numerical expressions.

For entity names, this approach is not good enough, so they use contextual information to detect names, combining symbolic transduction with probabilistic partial matching in their MUC implementation.

They report a combined precision-recall score of 93.39%, the highest on that MUC competition, where scores went as low as 69.67%. This gives us an idea of the state of the art in Named Entity recognition, and reinforce our belief that tools can be made to assist markup, though full reliable markup automation is not yet possible.

McDonald [McD96] also agrees on the use of both *internal* and *external* evidence to recognize proper names. As he states: "Internal evidence is taken from within the sequences of words that comprise the name", while "by contrast, external evidence is provided by the context in which a name appears".

Internal evidence methods rely on incorporation names (e.g. Ltd., Inc., etc.), heuristics and large gazetteers. Both McDonald and Mikheev et al, agree that external evidence (i.e. contextual information) is necessary for high accuracy performance.

McDonald distinguishes three steps to analyze an instance of a proper name:

1. "delimit the sequence of words that make up the name" (detection)
2. "classify the resulting constituent based on the kind of individual it names" (categorization)
3. "record the name and the individual it denotes in the discourse model as our interpretation of the constituent's meaning"

[7] The *Recall* measure is the **number of correct tags detected over the number of correct tags** (that should have been detected), while *Precision* is the **number of correct tags detected over the number of tags detected**

[8] Message Understanding Conference

For automatic markup we only need the first two steps: *detection* of the name, and then *categorization* (we need to know whether the name belongs to a person, place, company, etc.). Step three, *recording*, is needed to build a semantic model useful for disambiguation and association of equivalent names (like "Sony" and "the company").

Space does not permit the description of other works on Named Entity Recognition that are worth considering: [MM96, PLYM96, WGW96, PV00].

4.4 Transformation

A final step prior to publication is the transformation of the *source* `TEI-XML` *documents* to the most commonly used printing or display formats: `HTML`, `PS`, `PDF` are the most common nowadays. Arms describes this process in a simplified way in chapter 9, pg. 163-166 of his book [Arm00]. Our version of the transformation process is shown in fig. 2.

Figure 2: **Generation of an `HTML` digital book, `PS` and `PDF` files, from an `XML` file using `XSLT`**

On one hand (left side of fig. 2), we use `XSL` transformations to convert the `TEI-XML` documents to a plain `HTML` single file (with special formatting commands embedded), which is then splitted into structural parts (e.g. chapters), while a hypertextual table of contents, indexes, and navigation buttons are added. This is done with a parser we developed called `MakeBook`, which generates the digital books you can see in our website [CV99]. Apart from the `HTML` with embedded commands, a set of templates is taken as a source, to give a specific rendering to the generated `HTML` files that conform the digital book. This is

done in this way to give a uniform appearance to all the books, as well as to allow changes in page headers, colors, backgrounds and navigation buttons to be applied easily and uniformly. As shown in the diagram, changes in the templates imply further batch processing of all the source single `HTML` files to have the digital books changed. This was modified recently, so that the rendering of the digital books is now applied on-the-fly, that is at the time of serving the files through Internet. The `MakeBook` process still divides the `HTML` in structural parts and generates the TOC and eventual indexes, making all the hyperlinks between the parts, but page headers, footers, logos and top/bottom navigation buttons (i.e. the elements contained in the templates) are now added on-the- fly. This facilitates the implementation of general rendering changes to the whole library. For an example of how similar digital books can be given a different rendering in different collection see both the books inside Cervantes Virtual [CV99] and Joan Lluis Vives [JLV99] digital libraries. Both are portals of the same library (same server, same production tools, same production team), one with books in Spanish and the other in Catalan language).

On the other hand (right side of fig. 2), we also use `XSL` transformations to generate printable books in Postscript (not yet available on the Web), and from this ones we obtain `PDF` files (good for display as for printing as well), by means of the Acrobat Distiller [9] tool.

For transformations we use James Clark's `XT`, an implementation of `XSLT` [10]. There is another implementation of an `XSL` transformation parser from IBM Alphaworks.

4.5 Exploitation

This is the area where we have more things yet to be done. Currently, users can only retrieve entire files, based on catalog searches.

We want to allow complex searches on smaller textual objects (as defined by our markup scheme). For instance: show all the paragraphs that include the person name Cervantes, show all the verse lines that contain the place name Granada, or all headings with the word "cartas" (letters).

To allow this kind of searches, a consistent markup scheme is needed for the whole collection. Then, adequate indexing and filtering techniques for search and retrieval of textual objects. This is a field where research can be done to develop efficient methods for this costly complex tasks.

Apart from this we intend to implement tools for what linguists call "concordances": appearances of a given word in context, used to show uses of the given word by a given author or within a given literary work.

An interesting tool for concordances and searches is TACT [Bra], developed by John Bradley at the University of Ontario.

4.5.1 Search techniques for structurally marked up texts

Though there are some interesting algorithms that can be applied to searches [NS98, KM95], existing implementations like `SGREP` [JK96] are too general and too complex for persons with scarce data processing knowledge (as is the case of most of the potential users of digital libraries).

Furthermore, they do not apparently take advantage of the structure of the document in a thoroughly efficient way in order to accelerate the search. The

[9] Copyright: Adobe Systems Incorporated

[10] Extended stylesheet language transformations

construction of adequate interfaces and the use of finite state machines or of search indexes are some of the opened tasks in this field.

We try also to obtain more accurate results in searches, compared to those currently obtained with traditional Web indexers, taking advantage for this also of structural markup [CH99].

References

[Aho97] H. Ahonen. Disambiguation of SGML content models. *Lecture Notes in Computer Science*, 1293:27, 1997.

[AMN97] H. Ahonen, H. Mannila, and E. Nikunen. Generating grammars for SGML tagged texts lacking DTD. *Mathematical and Computer Modelling*, 26(1):1–13, 1997.

[Arm00] William Arms. *Digital Libraries.* MIT Press, Cambridge, Massachusetts, 2000.

[BKW98] Anne Brüggemann-Klein and Derick Wood. One-unambiguous regular languages. *Information and Computation*, 142(2):182–206, 1 May 1998.

[BP96] Branimir Boguraev and James Pustejovsky, editors. *Corpus Processing for Lexical Acquisition.* Language, Speech and Communication. Massachusetts Institute of Thechnology, Cambridge, Massachusetts, 1996.

[Bra] John Bradley. TACT: Text Analysis Computing Tools. http://www.chass.utoronto.ca/cch/tact.html. Universtiy of Ontario, Canada.

[BSM95] Lou Burnard and C. M. Sperberg-McQueen. Tei lite: An introduction to text encoding for interchange (document no: Tei u 5). http://www.uic.edu/orgs/tei/intros/teiu5.html, 6 1995.

[Bur95] Lou Burnard. Text encoding for information interchange: An introduction to the text encoding initiative. http://www-tei.uic.edu/orgs/tei/info/teij31/index.html, 7 1995.

[CH99] Alex Ceponkus and Faraz Hoodbhoy. *Applied XML: a toolkit for programmers.* John Wiley and Sons, Inc., USA, 1999.

[Cov] Robin Cover. DocBook, homepage (within OASIS). http://www.oasis-open.org/docbook/.

[CV99] Biblioteca virtual miguel de cervantes saavedra. http://cervantesvirtual.com, 1999.

[Der99] S. Derose. XML and the TEI. In *Text Encoding Initiative: Anniversary conference*, volume 33(1) of *Computers and the Humanities*, pages 11–30, Dordrecht, The Netherlands, 1999. Kluwer Academic Publishers.

[Har97] Stephen Harter. Scholarly communication and the digital library: Problems and issues. *JoDI (Journal of Digital information)*, 1(1), April 1997.

[JK96] Jani Jaakkola and Pekka Kilpeläinen. Using SGREP for querying structured text files. Technical Report Report C-1996-83, Department of Computer Science, University of Helsinki, November 1996. 11 pages.

[JLV99] Biblioteca Virtual Joan Lluis Vives. http://lluisvives.com/, 1999.

[KM95] Pekka Kilpeläinen and Heikki Mannila. Ordered and unordered tree inclusion. *SIAM Journal on Computing*, 24(2):340–356, April 1995.

[LR98] M. Lalmas and I. Ruthven. Representing and retrieving structured documents using the dempster-shafer theory of evidence: Modelling and evaluation. *Journal of Documentation*, 54(5):529–565, December 1998.

[McD96] David McDonald. Internal and External Evidence in the Identification and Semantic Categorization of Proper Names. In Boguraev and Pustejovsky [BP96], chapter 2.

[MGM99] Andrei Mikheev, Claire Grover, and Marc Moens. XML tools and architecture for Named Entity recognition. *Markup Languages: Theory & Practice*, 1(3), Summer 1999.

[MM96] Inderjeet Mani and T.R. MacMillan. Identifying Unknown Proper Names in Newswire Text. In Boguraev and Pustejovsky [BP96], chapter 3.

[MMG99] Andrei Mikheev, Marc Moens, and Claire Grover. Named Entity Recognition without Gazetteers. In *Proceedings of EACL'99*, pages 1–8, 1999.

[Mur97a] M. Murata. DTD transformation by patterns and contextual conditions. In *SGML/XML'97*, 1997.

[Mur97b] M. Murata. Transformation of documents and schemas by patterns and contextual conditions. *Lecture Notes in Computer Science*, 1293:153–??, 1997. este lo he pedido.

[NS98] Andreas Neumann and Helmut Seidl. Locating matches of tree patterns in forests. In *18th FSTTCS*, volume 1530 of *Lecture Notes in Computer Science (LNCS)*, pages 134–145, 1998.

[PLYM96] Woojin Paik, Elizabeth Liddy, Edmund Yu, and Mary McKenna. Categorizing and Standardizing Proper Nouns for Efficient Information Retrieval. In Boguraev and Pustejovsky [BP96], chapter 4.

[PSM] Wendy Plotkin and C.M. Sperberg-McQueen. Text Encoding Initiative, homepage (tei@uic.edu). http://www.uic.edu/orgs/tei/index.html.

[PV00] Maria Teresa Pazienza and Michele Vindigni. Identification and Classification of Italian Complex Proper Names. In *ACIDCA'2000, International Conference on Artificial and Computational Intelligence for Decision, Control and Automation in Engineering and Industrial Applications*, pages 146–151, Monastir, Tunisia, 3 2000.

[SMB94] C. M. Sperberg-McQueen and Lou Burnard, editors. *Guidelines for Electronic Text Encoding and Interchange (Text Encoding Initiative P3), Revised Reprint, Oxford, May 1999.* TEI P3 Text Encoding Initiative, Chicago - Oxford, May 1994.

[TEI00] Tei a 14: Tei application page. http://quirk.oucs.ox.ac.uk/TEI/Applications/, June 2000. Last updated: 8th June 2000. Copyright TEI 2000.

[WGW96] Takahiro Wakao, Robert Gaizauskas, and Yorick Wilks. Evaluation of an Algorithm for the Recognition and Classification of Proper Names. In *Proceedings of the* 16^{th} *International Conference on Computational Linguistics (COLING96)*, pages 418–423, Copenhagen, 1996.