A reusable hypermedia design for DL manuscripts, based on XML, XSLT and Java.

Alejandro Bia
Miguel de Cervantes Digital Library
University of Alicante, Spain
E-mail: abia@dlsi.ua.es
Phone: 34-965909567

Manuel Sánchez
Miguel de Cervantes Digital Library
University of Alicante, Spain
E-mail: manuel.sanchez@cervantesvirtual.com
Phone: 34-965909567

ABSTRACT
The purpose of this article is to describe our approach to the massive production of facsimile-type hypertext books that contain digital images of manuscripts and old printings to be published on the Internet as one of our DL services [1]. The goal of this project is to offer an easy-to-use interface that allows customizable views of facsimile images of book pages in several sizes and formats with transcriptions that are offered in three forms: Madison style, normal, and modernized. We explain the hypertext design, and the time-saving production methodology we use.

KEYWORDS: digital libraries, digital library development, hypermedia

INTRODUCTION
Differing from small web-sites, Digital Libraries [1] usually produce and publish a big number of digital books. Standardization in format and functionality is a must, so uniform, efficient, and low-cost methods for production of digital texts with images is of critical importance.

After successfully solving the problem of massive production of digital books in hypertext format (basically text with few images), we now face the problem of massive production of facsimile-type digital books (all pages displayed as images, plus in some cases text transcriptions).

The need for this type of digital books is based on the need to display manuscripts or ancient rare books not available to the general public. This would allow distant researchers and book lovers to take a look at books they would otherwise never get to see.

To start with, we did some bibliographic research on the first writings concerning hypertext and found some inspiration on several Hipertext classics like the articles by Vannevar Bush and Theodore Nelson [7, 21, 20, 26, 25].

From a high level or conceptual point of view, we applied ideas and design methods borrowed from the discipline of software engineering [13, 30, 5, 28, 27, 17] to design a semi-automated production process for developing facsimile versions of ancient books in a high scale, with both a low human-time cost and high quality.

From a low level or implementation level, we followed DeMarco's ideas on work-teams [9] by combining the productive cooperation and expertise of graphic designers, linguists, markup specialists and software technicians to develop the different parts of this manuscript production toolkit.

TEXT PLUS IMAGES PLUS HYPERLINKS
Why use a computer? Geoff Barnbrook poses this question in chapter 1 of his book Language and Computers [2]. He answers it justifying the importance of digital text and computer tools for linguistic research. Facsimile images of book pages published by Internet provide distant researchers with the possibility of analyzing ancient or rare books, which may otherwise never reach their hands, like manuscripts or *editiones principes* which are confined and not available for public use due to preservation reasons. The combination of both, facsimile images of pages and their text transcriptions provide the best of both worlds, i.e. the possibility of viewing the page in its original rendering, plus the benefits of text which can be searched (let's consider concordances as a special kind of search), cut, pasted, and processed in many ways.

ADVANTAGES OF CUSTOMIZING VIEWS
Fulfilling requirements most often implies complex trade-offs among quality, speed and size [14]. Text is light in terms of transmission time, whereas images are heavier. High resolution and big images usually imply a long wait. Sometimes the need may be to see a good quality image of a manuscript, but some others a text transcription may be enough. There is a wide range of possible combinations of different-quality images and transcriptions that can suit different requirements. We want to give users the chance to decide what will be displayed on their browsers as well as the time they are going to

[1] http://cervantesvirtual.com/

wait. Therefore, we allow users to construct the view that best suits their needs, by combining different image sizes and different transcriptions at will. The possible image types are thumbnail image, small image, big image, and three partial horizontal strips of the page. The available transcription types offered are normal, modernized and Madison transcription [22].

So we have two aspects to control:

- A compromise between transmission time and image quality.
- Optional transcriptions of several types.

QUALITY ASSURANCE

In a complex hypermedia design like this, quality assurance plays an important role. Each book may have hundreds of pages, each containing 6 different images and up to three different transcriptions. A quality assurance method [4] had to be developed to ensure that all the elements are in the correct place, the images are of good quality and the texts are free of mistakes. We decided to apply a check-list methodology to systematically perform a fast but accurate verification of these features [24].

DESIGN METHODOLOGY

The design method followed was simple.

First, we developed a prototype [2] that made evident the high effort necessary to develop a hypertext construction of this nature.

Secondly, we designed the production workflow described below.

Third, we made a specific document type definition (DTD) based on the XML-TEI DTD we currently use to markup our digital books.

Finally, we built the XSL transformation stylesheet necessary to convert XML into HTML. We also used another parser (MakeBook [3]) developed to process the rest of our digital books.

MANUSCRIPT PRODUCTION WORKFLOW

The production workflow for this type of books is shown by means of a diagram (see figure 4) that clarifies the production steps.

We have two parallel independent process: text processing and image processing.

Image processing begins with book pages being scanned or digitally photographed (the later only in cases where the age and value of the book prevent it from the rough handling of scanning). A single high resolution TIFF image is obtained from each page. OCR is not usually applicable to works selected for facsimile publication, since they are usually manuscripts or ancient printings, which are both interesting targets for image analysis and also very hard to convert to text by means of optical character recognition. Anyway, in cases where OCR can be applied with reasonable efficiency we do so to avoid the typing stage. After scanning, images are processed automatically in batch mode. Resizing, cropping and format conversion filters are applied to obtain 6 images: thumbnail, small (see figure 7), big (see figure 8) and three 1/3rd horizontal frames (see figure 6). Batch image processing saves a considerable amount of time.

Test processing begins by typing the text from the manuscript or ancient book. In rare cases where OCR can be applied, text correction replaces typing. In this way we obtain the Normal transcription. From this one we can obtain two additional transcriptions: one by adding some special marks according to the Madison standard [22], and the other by modernizing the spelling of ancient words. These variations of the normal transcription are not always done, so further processing must take into account the possible absence of some of these transcriptions or even the absence of all.

All the elements, images and transcriptions, are put together in a single XML-TEI file (a file coded according to the rules of the eXtensible Markup Language [6] and using the elements of the Text Encoding Initiative [31] markup scheme). Many books [23, 15, 18, 16, 12] and articles [10, 11] have appeared recently showing the advantages of using XML markup.

A quality control stage assures that every image and transcription texts is present and in its proper place.

An XSLT transformation stage generates, from the XML file, a single HTML file for the whole book. Another parser called MakeBook then splits the single HTML file into many files (one per page), generates an index (HTML table of contents with links to each page file), and adds the headers and footers of every page file with their corresponding navigation buttons. The resulting ensemble of web pages has a star-ring topology: a ring of web-pages bidirectionally linked (one for each original book page), plus an index page (TOC) at the center of a star that is bidirectionally linked to every page (see figure 1).

MARKUP CONSIDERATIONS

The markup scheme used to develop these books is graphically described in figures 2 and 3.

The XML files have the standard structure of a TEI document, with a *front*, *body* and *back* sections.

Inside these, we use *div1* to mark book pages, containing both *figures* and *div2*. We use *div2* with adequate values of the attribute *type* (*madison*, *normal* and *modern*) to mark the different types of transcription. Transcription texts are divided in three parts using *div3* tags that correspond to each one of the horizontal-strip images of the page (*top*, *middle*

[2] The prototype was built by Francisco Pérez using manuscripts from Calderón de la Barca's "El Divino Cazador" (The Divine Hunter)

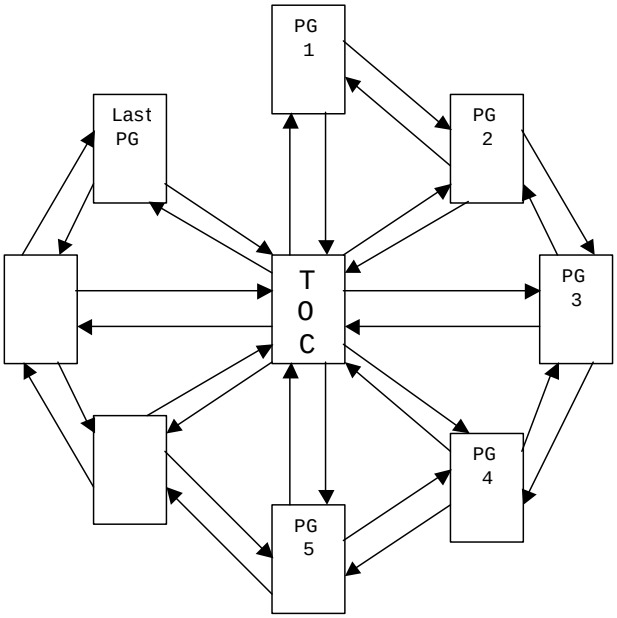

Figure 1: Star-Ring topology of the links of a digital book

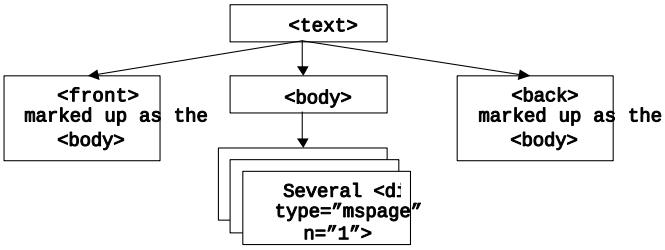

Figure 2: Markup structure of a manuscript file

Figure 3: Markup structure of a manuscript file

and *bottom*). An example of a view produced with a 1/3 strip-image can be seen in figure 6.

Figures correspond to the six types mentioned: *thumbnail*, *small*, *large*, *top*, *middle* and *bottom*.

JAVASCRIPT PROGRAMMING

A few Java functions were used to implement the changes of transcription style and the selection of images. A complete image can be selected for display, or a complete text instead, in which case the three possible transcription options apply. Users can select a horizontal-strip image, one third in height, by clicking on the thumbnail image that acts as a map, and then change the transcription type that is shown below at will.

ALTERNATIVE DESIGNS

We currently produce three different designs of facsimile pages that can be seen through Internet, as in these examples:

- The book "El divino cazador" by Calderón de la Barca: [3] This is the most complete and complex design, providing many sizes of images and different types of transcriptions.
- Another design can be seen in our portal for the Biblioteca Nacional de Chile: [4] This is a simple-to-navigate design.
- The last design with a different style is used in a portal about Charles the Fifth in our History section: [5]

[3] http://www.cervantesvirtual.com/bib_autor/Calderon/manuscrito/p1.html

[4] http://www.cervantesvirtual.com/servlet/SirveObras/bnc/48857617679851363142414 2/

[5] http://www.cervantesvirtual.com/historia/CarlosV/8_5_transcripcion_manuscrito.shtml

This different renderings are produced from texts with the same type of markup, but using different transformation stylesheets to produce the corresponding HTML pages.

REUSE OF THE DESIGN

In [8] the authors argue that "many word processing systems distract authors from their tasks of research and composition, toward concern with typographic and other tasks". Emphasis on WYSIWYG, while helpful for display, has ignored a more fundamental concern: representing document structure. The use of a markup language like XML, and a markup scheme like TEI allows our linguists to focus the efforts on developing contents, while format is added later by means of an XSLT stylesheet [19] and a postprocessing parser. In this way, the HTML [29] design of the manuscript pages is reused several times. Furthermore, we also obtain a uniform design for all the manuscripts of our DL. Maintenance tasks are also simplified, since making a change in the design of manuscript pages is done by changing the XSL stylesheet, and then the XML originals are automatically transformed into HTML in batch mode. As Steve DeRose says [8] "descriptive markup allows authors to focus on authorship."

CONCLUSIONS

- We reduced the task of developing these facsimile-oriented digital books to a minimum by using XML to markup the basic structure of the transcription texts and their corresponding images, leaving all formatting to be automatically applied by the XSL stylesheet.
- We reduced the task of image processing to just scanning the pages, and doing all further resizing and cropping automatically in batch mode.
- Changes in format can be easily performed by just modifying the XSL stylesheet. In this way, changes can be automatically propagated to all facsimile-type books by just reprocessing the XML source files with the modified stylesheet. This gives the whole collection of manuscript books a uniform look. Different collections of manuscripts with different rendering styles can be created by just having more than one style sheet.

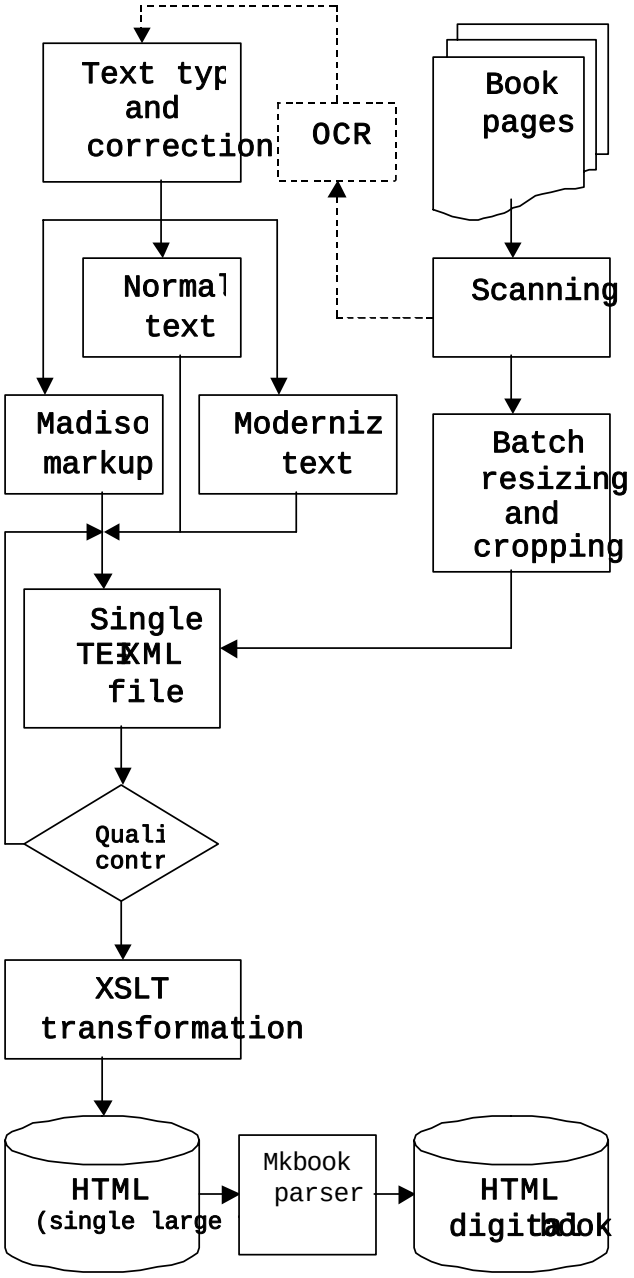

Figure 4: Workflow diagram of the manuscript production process

REFERENCES

1. William Arms. *Digital Libraries*. MIT Press, Cambridge, Massachusetts, 2000.
2. Geoff Barnbrook. *Language and Computers*. Edinburgh Textbooks in Empirical Linguistics. Edinburgh University Press, 22 George Square, Edinburgh EH8 9LF, 1996.
3. Alejandro Bia. MakeBook, a Yacc-Lex parser that processes HTML files to generate digital books. Technical report, Miguel de Cervantes DL, University of Alicante, July 1999.
4. Barry W. Boehm, Brown, Kaspar, Lipod, Macleod, and Merrit. *Characteristics of Software Quality*. TRW Series of Software Technology. Amsterdam, 1978.

Figure 5: Prototype, madison-transcription view of the manuscript

Figure 6: Prototype, 1/3rd-frame plus transcription view of the manuscript

Figure 7: Prototype, small-image view of the manuscript

Figure 8: Prototype, full view of the manuscript

5. Jorge Boria. *Ingeniería de Software*. Kapelusz, 1987.

6. Tim Bray, Jean Paoli, and C. M. Sperberg-McQueen. *Extensible Markup Language (XML) 1.0 W3C Recommendation*. World Wide Web Consortium, 10th February 1998.

7. Vannevar Bush. As we may think. *Atlantic Monthly*, 176:101–108, 7 1945.

8. J.H. Coombs, Alain H. Renear, and Steve J. DeRose. Markup systems and the future of scholarly text processing. *Communications ACM*, 30/11:933–947, 1987. Cf. CACM 31/7 (July 1988) 810-811.

9. Tom DeMarco and Timothy Lister. *Peopleware, Productive Projects and Teams*. Dorset House Publishing, 1987.

10. Steve Derose. XML and the TEI. In E. Mylonas and A. Renear, editors, *Text Encoding Initiative: Anniversary conference; 10th — November 1997, Providence, RI*, volume 33 of *Computers and the Humanities 1999; /2*, pages 11–30, Norwell, MA, USA, and Dordrecht, The Netherlands, 1999. Kluwer Academic Publishers Group.

11. Steve Derose and David Durand. XML linking. In *Hypertext00 workshop notes*, pages 11–55, San Antonio, Texas, USA, 2000.

12. Anamary Ehlen, editor. *HTML complete*. SYBEX, 1151 Marina Village Parkway, Alameda, CA 94501, USA, 2nd edition, 2000.

13. Richard E. Fairley. *Software Engineering*. McGraw Hill, 1985.

14. Donald Gause and Gerald Weinberg. *Exploring Requirements: Quality Before Design*.

15. Elliote Rusty Harold. *XML Bible*. IDG Books Worldwide, Inc, 999 E.Hillsdale Blvd., Suite 400, Foster City, CA 94404, USA, 1999.

16. Reaz Hoque. *XML for Real Programmers*. Morgan Kaufmann, 340 Pine Street, Sixth Floor, San Francisco, CA 94104-3205, USA, 1st edition, 2000.

17. Watts S. Humphrey. *Managing the Software Process*. Addison Wesley, 1990.

18. David Hunter, Curt Cagle, Dave Gibbons, Nikola Ozu, Jon Pinnock, and Paul Spencer. *Beginning XML*. Programmer to Programmer. Wrox Press, 1102 Warwick Road, Acocks Green, Birmingham, B27 6BH, UK, 1st edition, 2000.

19. Michael Kay. *XSLT Programmer's Reference*. Wrox Press, 1102 Warwick Road, Acocks Green, Birmingham, B27 6BH, UK, 1st edition, 2000.

20. F. Lancaster. *Toward paperless information systems*. Academic Press, New York, 1978.

21. J. Licklider. *Libraries of the future*. MIT Press, Cambridge, Massachusets, 1965.

22. David Mackenzie. *A Manual of Manuscript Transcription for the Dictionary of the old Spanish Language*. Madison, 4th edition, 1986. Victoria A. Burrus ed.

23. Michael Morrison. *XML Al descubierto*. Prentice Hall, Madrid, 2000. (translated from XML Unleashed, 2000, Sams, 0-672-31514-9).

24. Glenford J. Myers. *The Art of Software Testing*. John Wiley & Sons, 1979.

25. Theodore Nelson. The hypertext. In *Proceedings of the World Documentation Federation*, 1965.

26. Theodore Nelson. *Computer Lib.* Chicago, 1974.

27. Roger S. Pressman. *Making Software Engineering Happen*. Prentice Hall, 1988.

28. Roger S. Pressman. *Software Engineering, a Practitioners Approach*. McGraw Hill, 2nd. edition, 1988.

29. Dave Raggett, Arnaud Le-Hors, and Ian Jacobs. *HTML 4.0 Specification W3C Recommendation*. World Wide Web Consortium, 24th April 1998.

30. Ian Sommerville. *Software Engineering*. Addison Wesley, 2nd. edition, 1985.

31. 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.