OCR Alternatives for Electronic Publishing of Digitised Documents

Stefan Pletschacher

Chemnitz University of Technology, Institute for Print and Media Technology
Reichenhainer Straße 70, D-09126 Chemnitz, Germany
stefan.pletschacher@mb.tu-chemnitz.de

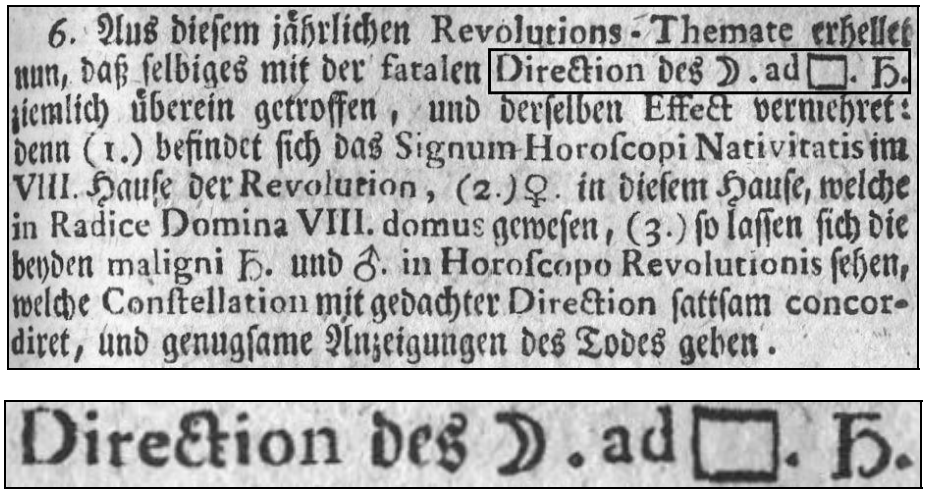
Abstract

This paper describes a general approach on how digitised documents may be automatically prepared for being stored and processed on various digital platforms. The focus is on documents that are not suitable for optical character recognition (OCR) methods but provide regular structures in the form of text-like blocks. By extracting a document immanent alphabet, preserving the graphical representations by means of vectorisation and based on these steps encoding the original document, it is possible to gather benefits of encoded text without the effort and the possible mistakes that arise from recognition methods. The use of the Extensible Markup Language (XML) for structural descriptions and Scalable Vector Graphics (SVG) for graphical representations enables a seamless integration into style sheet based output workflows for producing system specific layouts.

1 Introduction

In the field of digital preservation and delivery of printed resources it has to be dealt with huge amounts of material and therefore methods are needed for automated processing of data. When it comes to republishing digitised documents, especially for various output channels, there are several strategies.
A basic method is the reproduction of scanned documents by means of making copies using a bitmap facsimile of the original. For this purpose, not much processing or intelligent document analysis is necessary, but consequently the possible output formats and devices are limited. While this method is handy for simply storing documents or serving printers, it is not suitable for delivering content to new and emerging output devices like PDAs or mobile phones. Without further processing it is hardly possible to scale a document image to fit to a small screen for convenient reading.
A more sophisticated way of handling documents for efficient storage and redisplay is to conduct a document image analysis (DIA) involving several processing steps (O'Gorman & Kasturi, 1997), (Liang et. al., 1997). The main idea is to identify different parts of a document and to treat each in an appropriate way. Images are mostly left unchanged, whereas textual parts are processed further by means of OCR engines. Once the textual contents have been recognised, this approach enables individual layout modifications and even manipulation of the content. However, the results of OCR engines vary depending on quality and type of the scanned material. While modern machine writings are mainly covered by current engines, there are still considerable problems with rare fonts as recognition engines are not trained for them. Furthermore, OCR can only provide a certain recognition accuracy and in most cases it is necessary to review the results manually.
In order to avoid the effort of OCR and to reduce the risk of mistakes during the recognition process, there are approaches trying to locate text elements without recognising them. Usually, these systems cut words or characters in the form of bitmap images out of the document and thus create a structural description of the original document that also contains references to the text images. This allows the application of basic layout modifications like changing the word wrap for different display formats.
Our motivation in this field is the automated preparation, transformation and electronic provision of huge amounts of digitised documents, for instance whole books or historical sources (Ullrich et al., 2003), (Kreulich, 2003). We follow the latter of the above mentioned approaches to facilitate the processing of material that is not suitable for OCR but possesses a regular text-like structure (Figure 1). This often occurs in documents which mix different languages or fonts, as well as in historical resources with unknown characters. Our aim is to integrate the idea of a universal method in automated production workflows and to enable further transformations using more efficient representation formats. In order to obtain more benefits in comparison to other methods there are some requirements that a system implementation should provide. One issue is the compression of scanned images for efficient storage and network transfer of documents. Furthermore, any restrictions of the available character set for processing, which is inherent in OCR engines, are to be avoided. Additionally, it is desirable to achieve as much automation as possible, i.e. no manual interventions like reviewing of recognition results are needed. This can only be realised if the system does not introduce errors or falsifications into the digital representation of the original during processing. The system should offer functions for modifying the structure and layout of the content like changing the line break for adaptation to a smaller page or display size. Therefore new output channels for displaying the content of digitised documents will be

opened by applying transformations to the storage format. Finally, the system should use standards for easy exchange of data and better reusability.


6. Aus dieſem jährlichen Revolutions-Themate erhellet nun, daß ſelbiges mit der fatalen Direction des ☽ . ad □ . ♄. ziemlich überein getroffen, und derſelben Effect vermehret: denn (1.) befindet ſich das Signum Horoſcopi Nativitatis im VIII. Hauſe der Revolution, (2.) ♀. in dieſem Hauſe, welche in Radice Domina VIII. domus geweſen, (3.) ſo laſſen ſich die beyden maligni ♄. und ♂. in Horoſcopo Revolutionis ſehen, welche Conſtellation mit gedachter Direction ſattſam concordiret, und genugſame Anzeigungen des Todes geben.

Direction des ☽ . ad □ . ♄.

Figure 1. Example for a text-like block not suitable for conventional OCR (Nicolai, 1737).

2 An Alternative System Approach

When accessing the content of digitised documents via different media types and devices, various problems have to be dealt with in the production of output formats. There are several systems especially developed for this purpose (Altamura, Esposito, & Malerba, 2001), (Barrett & Barzee, 1998), (Liang et. al., 1996). Some systems offer numerous functionalities for recognition and modification of documents but most of them rely on OCR as a core component. The results of these systems are directly related to the possibility of recognising textual content. For a broad spectrum of books and documents recently printed, they achieve good recognition results and enable functions for search, modification and output. However, most of them fail when confronted with historical books or documents containing various languages, fonts or unknown characters.

Other approaches that do not rely on OCR are mainly based on the idea to extract prototypes or templates of the document immanent characters or words (Kopec & Lomelin, 1996), (Breuel et al., 2002). Thus it is possible to handle unknown characters since a document specific alphabet is generated during the analysis process. All different characters are to be found, and those characters of a certain similarity are mapped to their representative for subsequent compression. This is often referred to as Pattern Matching and Substitution (PMS) (Luczak & Szpakowski, 1995), (Atallah, Genin & Szpakowski, 1995). Textual parts of a document will be encoded by putting references to these prototypes. This concept is related to the idea of token-based or symbolic compression (Howard, 1997), (Kia, 1997). Most of these systems use only bitmap graphics for the graphical representation of the particular words or symbols.

If the layout is to be modified in order to be put out on different platforms, it is reasonable to create representations that can be easily manipulated. As the atomic elements of text-like blocks mostly possess regular geometric properties, reducing them to their shapes is an alternative which leads to vectorisation of glyphs. There are numerous methods for the vectorisation of drawings and images (Tombre et al., 2000). If preceding classification ensures that only glyph images are attempted to be vectorised, it becomes possible to obtain good results as the vectorisation methods can be especially adjusted to this task. Once the graphical representations have been obtained it is necessary to encode the document in an appropriate format. For the description of documents and their structures, XML offers great potential by using special Document Type Definitions (DTD) respectively XML Schema Definitions (XSD) as well as using SVG for the direct integration of graphical representations (Hitz, Robadey, & Ingold, 1999), (XML, 2005), (SVG, 2005). Furthermore, the XML format enables powerful output modifications by means of Extensible Stylesheet Language (XSL) and the corresponding transformations. The goal is not to replace existing DIA systems, but to use their functionalities – like segmentation, block classification, structure and layout recognition as well as character recognition – as a basis and expand them for a wider spectrum of documents by considering unknown characters and symbols.

3 System and Workflow Design

The following section describes the main steps necessary for providing the previously mentioned functionalities. For an overview of document image analysis methods see (Baird, Bunke & Yamamoto 1992), (Bunke & Wang 1997) or (Kasturi, O'Gorman & Govindaraju 2000). The envisaged system is divided into several components

which are responsible for the following tasks: image capture, pre-processing, segmentation and classification of image parts, recognition of textual blocks, extraction of the document specific character set, character vectorisation and representation in SVG, document encoding and description in XML, modifications of the layout and the production of new output formats.
Starting point for these considerations is an already pre-processed (at least deskewed and binarised) document image. The basis for further processing is the segmentation into different parts. On the top level, a region-based segmentation is needed that supplies the main blocks of the document. Depending on the media type, which has to be determined by means of classification methods, a decision has to be made on how to treat each block. The textual parts are of interest for further processing while images are kept as they are. To gather the complete structural information as well as to obtain the atomic elements, the segmentation has to be continued with text lines and then words resulting in single characters. (Figure 2a) illustrates the basic segmentation and classification. During this process the structural information along with the geometric attributes and positions have to be collected and stored in an adequate format like XML.

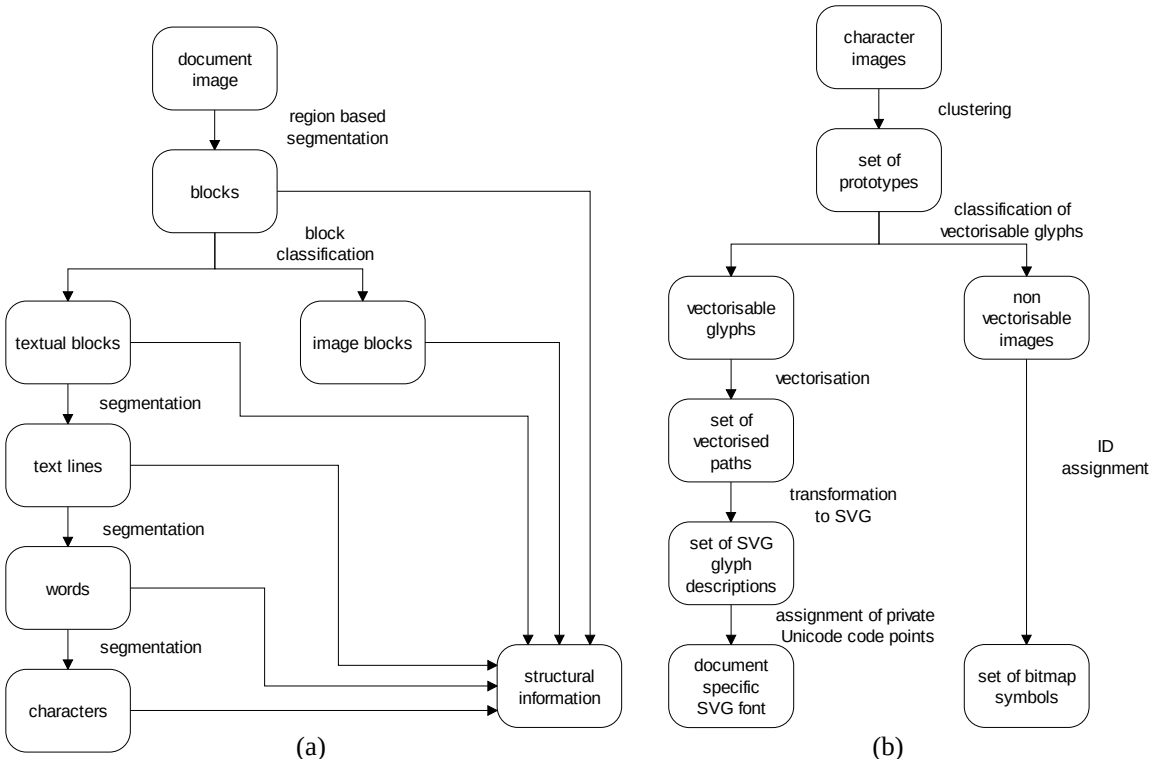


Figure 2. Sequence for document segmentation and block classification (a) as well as prototype extraction for a document specific alphabet and transformation to SVG (b)

After segmentation it is desirable to find the equivalence classes within the set of all characters in the document. This is achieved by means of clustering methods based on pattern matching or feature based similarity comparison (Figure 2b). The result is a set of prototype images which stand for their particular class of characters. Unlike OCR, this system does not assign characters to instances of a predefined alphabet. Therefore it is possible to avoid recognition errors when it is necessary to distinguishing between two similar characters. The proposed system creates a new character in the document specific alphabet if the similarity measure of the current character compared to already existing ones is too low. The threshold for this similarity can be relatively high to reliably distinguish between different characters. Thus it happens that two semantically equal characters can receive two separate entries in the alphabet. There is a trade-off between the complexity of the extracted alphabet and the accuracy of the classification.
After the extraction of the alphabet, it is reasonable to perform a second classification of the media type since textual blocks, especially in historical documents, may contain graphical parts like initials that are not suitable for the subsequent vectorisation. The images which are supposed to be glyphs are vectorised to preserve their shape for the later encoding. As XML is the primary format, SVG is used for the expression of vectorised paths. The appropriate SVG elements for shapes of characters are font definitions consisting of glyph descriptions for the whole alphabet. To make the graphical representation of a glyph accessible within text elements it is necessary to assign a code point to it. Unicode offers so-called private use areas that facilitate the coexistence of the document specific alphabet next to existing code ranges (Unicode, 2005). Therefore it is possible to combine OCR recognised text and document specific encoded text in a single XML file.

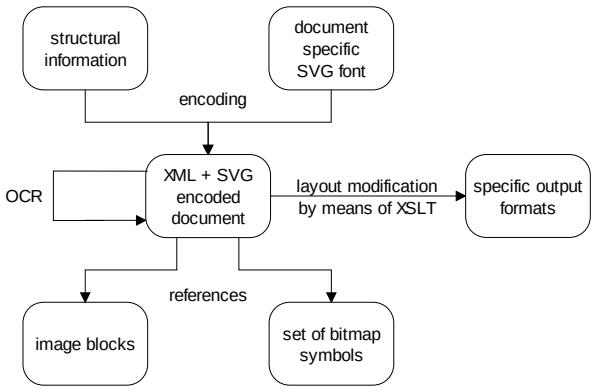


Figure 3. Encoding of the resulting XML document and possibilities for further processing.

The encoding of the final XML file is conducted by using the structural information collected during the segmentation process. The XML file represents the particular blocks with their positions as well as the detailed structure of textual blocks in form of text lines and words. Words are expressed in SVG <text> elements by a number of Unicode code points which refer to the document specific alphabet. This alphabet together with its graphical description is included in the head of the file using an SVG <defs> part. Images that can not be further processed are included by means of a reference to the file in combination with the original position.
Based on XML, SVG and Unicode standards are used that allow efficient storage and easy exchange. Once the document description is available in XML it is possible to produce various output formats by applying transformations. Additionally, the preservation of the original character shapes as well as the assignment of private Unicode code points still permits executing a conventional OCR afterwards. This OCR engine has to exchange private code points with known codes for recognised characters. For display or print, the stored shapes can still be used. The mixture of recognised and unrecognised text leads to partially searchable documents.

4 Results and Discussion

Several functions of DIA systems for segmentation and classification were used for proof of concept. Vectorisation has been performed using different commercial tools. Additionally, some transformation tasks and the encoding were done manually as the system is work in progress. (Figure 4) shows an SVG instance which is an intermediate result of the above described workflow.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.0//EN"
        "http://www.w3.org/TR/2001/REC-SVG-20010904/DTD/svg10.dtd">
<svg xmlns="http://www.w3.org/2000/svg" xml:space="preserve"
        width="500" height="500" viewBox="0 0 500 500">
    <defs>
        <font id="FontID" horiz-adv-x="300">
            <font-face font-family="DIA"/>
            <missing-glyph horiz-adv-x="300">
                <path d="M0 0 v40 h40 v-40z"/>
            </missing-glyph>

            <!-- Private Use Area in BMP -->
            <glyph id="gl01" unicode="&#xE000;" horiz-adv-x="200">
                <path d="M19 6c-4,60 -6,120 -8,180 60,10 60,10 88,6 0,-4 0,-8 0,-14 -16,0 -32, ...
            </glyph>

            <!-- Private Use Area-A, Supplementary -->
            <glyph id="gl02" unicode="&#xF0000;" horiz-adv-x="80">
                <path d="M23 9c-4,36 -6,72 -8,106 10,2 22,2 34,2 0,-34 0,-68 0,-102 -10,-2 -18, ...
            </glyph>
            ...
        </font>
    </defs>
    <text x="10" y="100">Known fonts may be combined with the document immanent font</text>
    <text x="100" y="200" font-family="DIA">&#xE000;&#xF0000; ... </text>
    <image id="source_image" xlink:href="source.jpg" x="98" y="170" width="142" height="14"/>
</svg>
```

Figure 4. SVG source code example for a document specific glyph alphabet and its utilisation in a text element.

The graphical display after the complete process for the example text line from (Figure 1) is shown in (Figure 5). The image was scanned with a resolution of 300 dpi in 256 grey levels, then binarised and deskewed.

Figure 5. Graphical result for the described workflow. The display is scaled to 400% of the original size and rendered by the Squiggle SVG viewer (Apache, 2005). The upper line is a bitmap facsimile whereas the lower is SVG text referencing the document specific glyphs.

One obvious effect is the better scalability of the vectorised glyphs in comparison to the text image. As the interpretation remains with the reader it is of no concern if there are slight differences between the original image and the vectorised copy. (Figure 5) shows that it is possible to achieve a token-based compression, for instance by reusing the “i” and the dot. Semantically equal characters which have too little similarity of their shape (like the two forms of the “e” in the upper example) will get different prototypes in the alphabet. Furthermore, it is uncomplicated to handle unknown characters or symbols and treat them like text. Another effect is the robustness of the process when it comes to ligatures. They are simply handled as one character of the extracted alphabet. If the segmentation methods are not able to return correctly separated parts for connected characters, the system can still cope with them as new entries in the alphabet.
The goal of producing specific output formats for different platforms and devices can be achieved by means of output transformations. (Figure 6.) shows some simple layout modifications by changing the text decoration and the alignment in a smaller column. Especially applications for mobile devices with small displays could benefit from rearrangement of text to a suitable size. Textual parts can be presented in an easily readable size while images within documents may be scaled down to fit to the screen.

Figure 6. Example for text decoration and rearrangement of the text flow.

5 Conclusion and Future Work

An approach and system design has been shown for an alternative way of preparing and delivering digitised documents for various digital output platforms. The main idea is to avoid mistakes and therefore manual interventions during the process to enable automated workflows. The first results are very promising and encourage the further development of this system. A major advantage of this approach is the ability to handle a variety of documents containing different fonts, languages and even unknown characters. The gist is that the graphical representations of the original characters are preserved by vectorisation and encoding in SVG after the document specific alphabet has been extracted. The interpretation of the produced output document remains with the reader. The system is not prone to recognition errors as it would not erroneously assign a wrong code to a character. Benefits are among others the possibility to modify the layout and the structure of the document. As the property of text – to contain words that consist of characters – is kept, it is possible to change for example the line break within a text block. This enables the production of output formats for special devices like small displays of mobile phones. As the vector format SVG allows scaling without further loss of quality, it is possible to adjust textual parts to a suitable size whereas other image parts are left unchanged.
One disadvantage is the absence of options for further development of the content. It is not possible to copy and paste text encoded in private Unicode code points into standard word processors. However, as this format preserves the graphical representation of the original document, it would be feasible to apply a conventional OCR at the end of the process. Likewise full text search is not available until an OCR is performed. Yet there are alternative methods available based mostly on statistical assumptions which allow summarising documents by highlighting significant words or phrases (Bloomberg & Chen 1996), (Chen & Bloomberg 1998). Additionally, it is conceivable to execute searches that look for words in the document immanent alphabet by

means of sequences in the private Unicode range. However, this requires a special user interface to access the document specific alphabet for entering search queries.
Many of the problems that arise during the described workflow could not be discussed in this paper, for instance on how to achieve confident classification results for the identification of textual blocks and vectorisable characters. Research remains to be done on how to handle the trade-off between the complexity and the accuracy of a document specific alphabet as well as on the optimisation of vectorisation methods for the special task of representing glyphs.

References

Altamura, O., Esposito, F. & Malerba, D. (2001). *Transforming paper documents into XML format with WISDOM++*. International Journal of Document Analysis and Recognition, 4(1), pp. 2–17

Apache Software Foundation, *Squiggle - the SVG Browser*, http://xml.apache.org/batik/svgviewer.html, last visited April 2005

Atallah, M., Genin, Y. & Szpakowski, W. (1995). *Pattern matching image compression: Algorithmic and empirical results.* Technical Report CSD TR-95-083, Computer Science Department, Purdue University

Baird, H. S., Bunke, H. & Yamamoto K. (Eds.) (1992). *Structured document image analysis.* Springer-Verlag, Berlin

Barrett, W. A. & Barzee, R. A. (1998). *Posting Paper on the Web.* Proceedings Vision Interface '98, pp. 381-388

Bloomberg, D. S. & Chen, F. R. (1996). Extraction of text-related features for condensing image documents. In: Vincent L. & Hull J. (Eds.), *Proceedings of the SPIE—Document Recognition III.* The International Society for Optical Engineering (SPIE), Vol. 2660, pp. 72–88

Breuel, T., Janssen, W., Popat, K., & Baird, H. (2002). *Paper to PDA.* In Proc. 16th Int. Conf. on Pattern Recognition (ICPR)

Bunke, H. & Wang, P. S. P. (Eds.) (1997). *Handbook of Character Recognition and Document Image Analysis.* World Scientific

Chen, F. R. & Bloomberg, D. S. (1998). Summarization of imaged documents without OCR, *Computer Vision and Image Understanding* Vol. 70, no. 3, pp. 307-320

Hitz, O., Robadey, L. & Ingold, R. (1999). *Using XML in Document Recognition.* In: Proc. International Workshop on Document Layout Interpretation and its Applications, Bangalore, India

Howard, P. G. (1997). *Text image compression using soft pattern matching*. The Computer Journal, 40, pp. 146-156

Kasturi, R., O'Gorman, L. & Govindaraju, V. (2000). *Document image analysis: A primer*, Sadhana, Vol. 27, Part 1, pp. 3-22.

Kia O. E. (1997). *Document Image Compression and Analysis.* Technical Report: LAMP-TR-010/CFAR-TR-856/CS-TR-3786, University of Maryland, College Park

Kopec, G. & Lomelin, M. (1996). *Document-Specific Character Template Estimation*. IS&T/SPIE 1996 Intl. Symposiumon Electronic Imaging: Science & Technology, San Jose, CA

Kreulich, K. (2003) *Publishing Workflows with XSL-FO.* In XML Europe 2003, London, England

Liang, J., Ha, J., Rogers, R., Phillips, I. T., Haralick, R. M. & Chanda, B. (1996). *The Prototype of a Complete Document Image Understanding System.* IAPR Workshop on Document Analysis System, Malvern, PA

Liang, J., Rogers, R., Haralick, R. M. & Phillips, I. T. (1997). *UWISL Document Image Analysis Toolbox: An Experimental Environment.* Proc. Fourth ICDAR, Ulm, Germany, pp. 984 - 988.

Luczak, T. & Szpakowski, W. (1995). *A lossy data compression based on an approximate pattern matching.* Technical Report CSD TR-94-072, Computer Science Department, Purdue University

Nicolai, C. G. (1737). *Anleitung zu den curiösen Wissenschaften.* Frankfurt und Leipzig

O'Gorman, L. & Kasturi, R. (1997). *Document image analysis.* IEEE Computer Society Press Executive Briefing Series, Los Alamitos, CA

SVG - Scalable Vector Graphics, W3C, http://www.w3.org/Graphics/SVG/, last visited April 2005

Tombre, K., Ah-Soon, C., Dosch, P., Masini, G. & Tabbone, S. (2000). Stable and Robust Vectorization: How to Make the Right Choices. In A. K. Chhabra and D. Dori, (Eds.), *Graphics Recognition - Recent Advances*, pp. 3-18. Springer Verlag, Lecture Notes in Computer Science, vol. 1941

Ullrich, S., Hübler, A. C., Kreulich, K. & Enge, C. (2003). *A Device for Automated Scanning of Books.* 2003 Conference on Electronic Theses and Dissertations Worldwide, Berlin (ETD)

Unicode Inc., *Unicode Standard*, http://www.unicode.org/standard/standard.html, last visited April 2005

XML - Extensible Markup Language, W3C, http://www.w3.org/XML/, last visited April 2005