An Overview of Digitization of Information Resources

Dillip K Swain
Asst Librarian, School of Computer Application, KIIT University, Bhubaneswar, Orissa
E-mail: swaindk_69@yahoo.co.in
K. C. Panda
Professor& Head, PG Dept of LIS, Sambalpur University, Orissa
E-mail: krushna52@yahoo.co.in

Abstract
This paper highlights basic steps and mechanisms involved in creating and documenting an electronic text or similar digital resources emphasizing the tools, techniques, standards that are adopted in the process of creation of e-resources with intent to provide the review of existing mechanisms that are of interest to the librarians who are at the initial stage of digitizing their respective library collection.
Keyword: Scanning, Image capture, OCR, Re-keying, XML, SGML, File format

Introduction

Digitization is quite simply the creation of a computerized representation of a printed analog. There are many methods of digitizing and varied media to be digitized. However, the main focus rests primarily on texts and images, as these are the main objects in the digitization process; therefore, it refers to the conversion of materials that were originally created in another format. Technically, the process of digitization involves converting an analog image into its corresponding numeric values[1]. In this context, some of the fundamental issues like, scanning and image capture, necessary hardware and software selection that are crucial for the process of digitization are briefly discussed in the succeeding sections.

Scanning and Image Capture

The first step in digitization, both text and image, is to obtain a workable facsimile of the page. To accomplish this, the electronic text creator will need a combination of hardware and software imaging tools. This is somewhat difficult area to address in terms of recommending specific product brands, as what is considered industry (or at least the text creation industry) standard is subject to change as technology develops[1]. The word “image” is literally true because, the digital scanner creates an image of the original analog item, whether that item is a photograph, a word processed document, or a hand written letter. The digital mage created by the scanner is stored in numeric form. For example, when a photograph is digitized for viewing on a computer screen, the original continuous tone image is divided into dots with assigned values that are mapped against a grid. The pattern of the dots is remembered and reassembled by the computer upon appropriate command[2]. However, some of the hardware and software frequently used by archives and digital project creators may be discussed.

Hardware-Types of Scanner and Digital Cameras

There are quite a few methods of image capture that are used within the humanities community. The equipment ranges from scanners (flatbed, sheetfed, drum, slide, microfilm) to high end digital cameras. In terms of standards within the digitizing community, the results are less than satisfactory. Projects tend to choose the most available option, or the one that is affordable on limited grant funding. However, two of the most common and accessible image capture solutions are flatbed scanners and high resolution digital cameras[1].

Software

Making specific recommendations for software program is a problematic proposition. Since there have been no agreed standards for digitization with Software, as with hardware, the choices may vary from project to project depending upon personal choice, university recommendations, and often budgetary restrictions. However, there are a few tools that are commonly seen in use with many digitization projects. Regardless of the brand of software purchased, the project will need text scanning software, if there is to be in-house digitization of text and an image manipulation software package, if imaging is to be done. There are a wide variety of text scanning softwares available, all with varying capabilities. However, the primary consideration with any text scanning software is how well it works with the condition of the text being scanned[1].For this purpose Adobe photoshop is the most common choice.

Image Capture and Optical Character Recognition (OCR)

The best method of digitizing text is Optical Character Recognition (OCR). This process is accomplished through the utilization of scanning hardware in conjunction with text scanning software. OCR takes a scanned image of a page and converts it into text. Similarly, image capture also requires image scanning software to accompany the hardware. However, unlike text scanning, image capture has more complex requirements in terms of project decisions and, like almost everything else in the digitization project[1].

Image Types

There are four main types of images: 1-bit black and white, 8-bit greyscale, 8-bit color and 24-bit color. A bit is the fundamental unit of information read by the computer, with a single bit being represented by either a “0” or a “1”. A ‘0’ is considered an absence and a ‘1; is a presence, with more complex representations of information being accompanied by multiple of gathered bits[3].

A 1-bit black and white image means that, the bit can either be black or white. This is a rarely used type and is completely unsuitable for almost all images. While 8-bit greyscale images are an improvement from 1-bit as they encompasses 256 shades of grey. It is often used for non-color images (*http://www.hcu.ox.ac.uk/jtap/*). Moreover, greyscale images are often considered more than adequate, there are times when non coloured images should be scanned at a higher color because the finite detail of the hand will come through distinctly.8-bit color is similar to 8-bit greyscale with the exception that each bit can be one of 256 colors. The decision to use 8-bit color is completely project dependent, as the format is appropriate for web page images but can come out somewhat grainy. However, 24-bit color is the best scanning choice. This option provides the highest quality image, with each bit having the potential to contain one of 16.8 million colours.The arguments against the image format is the *size*, *cost* and *time* necessary[3].

Resolution

The second concern relates to the resolution of the image. The resolution is determined by the number of dots per inch (dpi). This choice is directly related to what is being done with the image. If the image is being archived or will need to be enlarged, then resolution will need to be relatively higher. However, if the image is simply being placed on a web page, then the resolution drops drastically. The higher the dpi, the larger the file size. To illustrate the differences, an informative table created by the Text Center, which examines an uncompressed 1”x 1” image in different types and resolutions may be given below in Table-1 (*ota.ahds.ac.uk/documents/creating/chap1.html*):

Table-1 Different Types of Resolutions

Resolution(dpi)	400x400	300x300	200x200	100x100
2-bit black and white	20k	11k	5k	1k
8-bit greyscale or color	158k	89k	39k	9k
24-bit color	475k	267k	118k	29k

(Source: ota.ahds.ac.uk/documents/creating/chap1.html-12k)

Obviously, the 400 dpi scan of a24-bit color image is going to be the largest file size because, the screen resolution rarely exceeds this amount. Therefore, the dpi choice primarily depends upon the project objectives[1].

File Formats

In terms of text creation, three types of file formats are commonly seen in the process: TIFF, JPEG, and GIF. These are the most common image formats because they transfer to almost any platform or software system.

TIFF (Tagged Image File Format) files are the most widely accepted format for archival image creation and retention as master copy. Most digitization projects begin image scanning with the TIFF format, as it allows gathering as much information as possible from the original and then saves these data. On the other hand, JPEG (Joint Photographic Expert Group) files are the strongest format for web viewing and transfer through systems that have space restrictions. JPEGs are popular with image creators not only for their compression capabilities but also for their quality. GIF (Graphic Interchange Format) files are older format that are limited to 256 colors and don't have the compression capabilities of a JPEG, yet they are strong for graphic art and line drawing[1].

Re-Keying

Unfortunately for the text creator, there are still many situations where the documents or project prohibit the use of OCR. If the text is of a poor or degraded quality, then it is quite possible that the time spent in correcting the OCR mistakes will exceed that of simply typing in the text from scratch. The amount of information to be digitized also becomes an issue. Even if the document is of a relatively good quality, there might not be enough time to sit down with 560 volumes of texts and process them through OCR. Therefore, when OCR is found incapable of handling the project of digitization, the viable solution should be re-keying the text[1]

Markup

Markup is commonly defined as a form of text added to a document to transmit information about both the physical and electronic sources. This is nothing but the typographical design of a document. As Philip Gaskell points out, 'many examples of printers' copy have survived from the hand- press period, some of them annotated with instructions concerning layout, italicization, capitalization, etc. [4]. According to G. T. Tenselle, one might choose a particular text to markup to reflect these editorial decisions, but that text would only be serving as a convenient basis for producing printer's copy[5]. The leap from markup as a method of labeling instructions on printer's copy to markup as a language used to describe information in an electronic document is not so vast.

Postscript and Portable Document Format (PDF)

In 1985, Adobe Systems created a programming language for printers called postscript. In doing so, they produced a system that allowed computers to talk to their printers. This language describes for the printer the appearance of the page, incorporating elements like text, graphics,

color, and images, so that documents maintain their integrity through the transmission from computer to printer. PostScript printers have become industry standard with corporations, marketers, publishing companies, graphic designers, and more. Printers, slide recorders, image setters- all these output devices utilize PostScript technology (*http://www.adobe.com/print/features/psvspdf/main.html*).

Portable Document Format (PDF) was created by Adobe in 1993 to complement their PostScript language. PDF allows the user to view a document with a presentational integrity that almost resembles a scanned image of the source. Another enticing feature, depending on the quality of the printer, is that when a PDF file is printed out, the hard copy output is an exact replication of the screen image. PDF is also desirable for its delivery strengths. Not only does this document maintain visual integrity, but also can be compressed. This compression eases on-line and CD-ROM transmission and assists its achieving opportunities. PDF files can be read through an Acrobat Reader application that is freely available for download via the web. This application is also capable of serving as a browser plug-in for online document viewing. Creating PDF files is a bit more complicated task. To write a PDF document, it is necessary to purchase Adobe software[1].

HTML 4.0

Hyper Text Markup Language (or HTML as it commonly known) is a non-proprietary format markup system used for publishing hypertext on the World Wide Web. To date, it has appeared in four main versions (1.0, 2.0, 3.0, 4.0), with the World Wide Web consortium (W3) recommending 4.0 as the markup language of choice. HTML is a derivative of SGML- the Standard Generalized Markup Language. SGML allows one to create his own markup language but provides the necessary support to ensure its processing and preservation. HTML is a successful implementation of the SGML concepts, and, as a result, is accessible to most browsers and platforms. Along with this, it is a relatively simple markup language to learn, as it has a limited tagset. HTML is by far the most popular web- publishing language, allows users to create online text documents that include multimedia elements (such as images, sounds, and video clips, etc.), and then put these documents in an environment that allows for instant publication and retrieval[1].

There are many advantages to a markup language like, HTML. As mentioned above, the primary benefit is that, a document encoded with HTML can be viewed in almost any browser- an extremely attractive option for creator who wants documents which can be viewed by an audience with varied systems. However, it is important to note that while the encoding the data, there are consistently differences in page appearance between browsers. While W3C recommends the usage of HTML 4.0, many of its features are simply not available to users with early versions of browsers. HTML has no true sense of page structure and files can neither be saved nor printed with any sense of precision[1], however, it attracts its many users for the simple manner with which it can be mastered.

User-definable Descriptive Markup

A user-definable markup is exactly what its name implies. The content of the markup tag is established solely by the user, not by the software. As a result of SGML and its concept of a DTD, a document can have any kind of markup a creator desires. This frees the document from being married to proprietary hardware and software and from its reliance upon an appearance- based markup language. If one decides to encode the document with a non-proprietary language, which we highly recommend, then this is a good time to evaluate the project goals. While a user-definable markup language gives you control over the content of the markup, and thereby more control over the document, the markup can be fully understood by you. Although not tied to proprietary system, it is also not tied to any accepted standard. The markup language defined and

implemented by you is simply that- a personal non-proprietary markup system[1].

However, if the electronic texts require a language that is non-proprietary, more extensive and content -oriented than HTML, and comprehensible and acceptable to a humanities audience, then there is a solution- the Text Encoding Initiative(TEI). TEI is an international implementation of SGML, providing a non-proprietary markup language that has become the de facto standard in Humanities Computing. TEI provides a full set of tags, a methodology, and a set of Document Type Descriptions (DTDs) that allow the detailed (or not so detailed) description of the spatial, intellectual, structural, and typographical form of work[6].

XML: The Future of SGML

SGML has an undeserved reputation for being difficult and expensive to produce because it imposes prohibitive intellectual overheads, and because the necessary software is lacking. While it is true that performing a thorough document analysis and developing a suitable DTD should not be undertaken lightly, it could be argued that, to approach the production of any electronic text without first investing such intellectual resources is likely to lead to difficulties. One possible solution to this dilemma is the Extensible Markup Language (XML) 1.0 which became a W3C Recommendation on 10th February 1998 (*http://www.w3.org/TR//REC-xml*).

Both libraries and document providers can easily modify XML-exchange information in existing applications, since XML supports the definition of language-neutral and platform-neutral facilities. Our electronic document delivery system model prefers XML for **two** reasons: *language theory* and *practical application*. XML enables developers to create and manipulate their own tags and it also works smoothly with Cascading Style Sheets (CSS) to enable developers to present information as it is originally structured. XML is an excellent format for interchanging data, since browsers (like IE5.0) can read XML data[7].

Increasingly, XML applications are appearing on the World Wide Web, from e-commerce to information management. In the case of libraries and archieves, XML enables more flexible management and retrieval than using MARC or a relational database management system[8].

Documentation and Metadata

Metadata are ‘data about data’. The term is perhaps more clearly defined as “data that records information about a resource” [9]. According to *Taylor* (2003), ‘metadata’ is structured data which describes the characteristics of a resource. It shares many similar characteristics to the cataloguing that takes place in libraries, museums, and archives. A metadata record consists of a number of pre-defined elements representing specific attributes of a resource, and each element can have more values[10]. The example of a simple metadata is depicted in Table-2 given below.

Table-2 Sample Metadata Format

Element	Value
Title	Web Catalogue
Creator	Dagniya Auliffe
Publisher	University of Queens land Library
Identifier	http://www.library.uq.edu.au/iad/mainmmenu.html
Format	Text/html
Relation	Library Web Site

Source :http://www.library.uq.edu.au/iad/itmeta4.html

In recent years the issue of metadata has become a serious topic for those concerned with the creation and management of digital resources. When digital resources first started to emerge much of the focus of activity was centered on the creation process without much thought given to

how these resources would be well documented and found by others.

At its inception the web was not designed nor intended as a forum for the organized publication and retrieval of information and therefore, no system for effectively cataloguing information held on the web was devised. Due to this lack of formal cataloguing procedures, the web has evolved into a chaotic repository for the collective output for the world's digital printing presses[11]. Many of the newly emerging metadata standards have been applied and tested in digital library projects. In particular, the Dublin Core Metadata Standard has been a frequent choice[12].

Hardware and Software Requirements for Creation of E-Resources

There are different hardware and softwares available in market for the creation of digital resources. However, in the light of Ohiolink Digital Resource Centers (DRC) recommendations (*http://eprints.nelis.org/archieve*) the following specification for hardware and software requirement for the effective creation of digital resources may be followed:

Hardware

i) Enterprise servers and storage networks;
ii) Server computers should be on the internet backbone ensuring maximum availability and speed; and
iii) Unlimited storage space, massive offsite tape and disk back up systems to ensure the safety and security of content.

Software

Paid/Open source software should have the following:

i) Should provide a universal repository capability of storing and managing all content types anticipated;
ii) Should supply the security and access controls;
iii) Capability to create search and browse mechanisms;
iv) Virtual document and rendition management capabilities to support complex document structures; and
v) Workflow capabilities that facilitate the automatic routing of content and task.

Conclusion

While digitizing information resources, the creators of authentic digital information require extensive skill and ability to track copies of authoritative originals as part of licensing and protection mechanisms. Many technical methods are being developed or offered that need to be thoroughly grasped. Therefore, it is quite inevitable to determine which methods are suited for what purposes with a thorough understanding of the functional requirements of the electronic texts meant for the potential users of digital resources.

References

1. Morrison, et al. (n. d), Guide to Creating and Documenting Electronic Texts. Available from : ota.adhs.ac.uk/documents/creating/Chap.1.html(accessed 27 September, 2009).
2. Chaurasia, N K.(2006), "Digitization of Library Materials", *University News*,Vol.44 No.30, pp.24-30.
3. Robinson, P.(1993), " The Digitization of Primary Textual Sources. Oxford: Office for Humanities Communication Publications, available at: ota.ahds.ac.uk/documents/creating/chap9.html (accessed 12 November, 2007).
4. Gaskell,P.(1995), "A New Introduction to Bibliography". Delaware ,Oak Knoll Press.
5. Tanselle, G.T.(1981), "Recent Editorial Discussion and the Central Questions of Editing", *Studies in*

Bibliography, Vol.34, pp.23–65.

6. Seaman, D(1994). “Campus Publishing in Standardized Electronic Formats — HTML and TEI (online)”, available at: http://etext.lib.virginia.edu/articles/arl/dms-arl94.html (accessed November 7, 2007).
7. Yu (Shien-Chiang). “Developing an XML Framework for an Electronic Document Delivery System”, *The Electronic Library*, Vol.19 No. 2, pp.102-111.
8. Chiang,M., “An Electronic Finding Aid Using Extensible Markup Language (XML) and Encoded Archieval Description (EAD)”, *available at:* http://www.library.qmul.ac.uk/e-resources/ej.htm(accessed 2 September, 2009).
9. National Library of Australia(2000), Safeguarding Australia’s Web resources : guidelines for Publishers , available at http://www.nla.gov.au/guidelines/2000/webresources.html(accessed 27 September, 2009).
10. Taylor, T.(2003), available at: http://www.library.uq.edu.au/iad/ctmeta4.html(accessed October 3,2009).
11. Lynch, C. (2003), “Digital library opportunities”, *The Journal of Academic Librarianship*, Vol.29 No.5, pp. 286-289.
12. Guinchard, C.(2002), “Dublin Core Use in Libraries: A Survey”, *OCLC Systems and services*, Vol.18 No.1, pp. 40-50.