

HTML5 ETDs

Sung Hee Park*, Nicholas Lynberg†, Jesse Racer†, Phil McElmurray*, Edward A. Fox*
Department of: Computer Science*, Electrical & Computer Engineering†, Virginia Tech
{shpark,nlynber,jracer,pmac,fox}@vt.edu

We have developed and demonstrated a software prototype to convert plain ETDs (Electronic Theses and Dissertations) into HTML5. This can be thought of as a migration to a better format for preservation. Browsers for HTML5, the latest revision of HTML, directly support access to multimedia content and hypermedia works. Most current ETDs are submitted in PDF (Portable Document Format). Some are accompanied by multimedia files or other files connected with hypertext links, i.e., are hypermedia works. All of these kinds of files can be integrated using HTML5. For readers of ETDs it would be much more convenient to view an HTML5 file in a browser that directly manages multimedia; thus our goal is to transform to HTML5. Our semi-automatic algorithm first finds links or clues for links (e.g., video or audio file names, figures, references, tables, etc.) in a PDF ETD. Then it constructs an HTML5 file including all suitable links. Thus, HTML5 will make it easier to read and preserve multimedia and hypermedia ETDs. We have transformed 3 multimedia plus 3 linked (hypertext/hypermedia) ETDs into HTML5 in a semi-automatic way. We have developed a software prototype that helps in the migration. For a user's viewing convenience, an ETD digital library can support HTML5, which facilitates viewing of integrated multimedia content following video and audio tagging. In addition to helping with preservation and supporting browsing, this will improve access to multimedia ETDs by mobile devices (e.g., iPods).

1. INTRODUCTION

Recently, as mobile devices (e.g., smart phones, PDAs, etc.) become more and more common, technological changes have emerged, e.g., the *mobile web*. The mobile web can be accessed from popular mobile devices anytime, anywhere. In addition to these changes in computing environments, a standard for the mobile web is being developed: HTML5 [1]. HTML5 is the latest revision of HTML, which is expected to foster the mobile web so that mobile users can easily access rich web resources (e.g., video, canvas), reducing the need for proprietary plug-in applications (e.g., Adobe Flash, Microsoft Silverlight) [8].

In a meantime, for digital libraries, long-term preservation of digital media has been researched [2]. LOCKSS (Lots of Copies Keep Stuff Safe), based at Stanford University Libraries, is a long-term preservation tool. Especially in higher education, projects to make ETDs (Electronic Theses and Dissertations) easily long-term accessible and preserved for the next generation have been in collaboration with ASERL¹ and LOCKSS² [3], and in collaboration with NDLTD³ and MetaArchive⁴ [4].

Those long-term preservation efforts would be more effective when aided by format migration which converts an old format to a new format, in keeping with new technologies and standard file formats (e.g., hardware and software) [5, 6, 7, 8]. Format migration is a fundamental preservation strategy which: preserves content and functionality of a digital object, ensures continued access to the digital object, and minimizes physical and intellectual information loss [7]. One of the possible strategies is to migrate to a system that is compliant with open standards.

In this paper, we have developed and demonstrated a software prototype to convert plain ETDs into HTML5. Also, we have investigated if HTML5 aids migration for the mobile web environment.

In section 2, this paper provides background regarding ETD structure, migration strategies, and format conversions. In section 3, a proposed HTML5 ETD conversion algorithm and implementation is described. Problems, lessons learned regarding migration, and adaptations for mobile computing, are discussed in section 4. Finally, we conclude, noting that migrating to a format for the mobile web will help people browse ETDs with devices like iPods.

2. BACKGROUND

Prior to planning conversion into a new file format, it is important to understand: requirements of the mobile web, evolution of the HTML standard, ETD structure, and issues related to migration and preservation.

A. HTML5

HTML5 is a new revision of HTML4 [9], XHTML1 [10], and DOM Level 2 HTML [11]. It is being developed by the W3C HTML Working Group⁵ and the Web Hypertext Application Technology Working Group⁶ (WHATWG).

HTML5 has many features for the mobile web. For example, the *manifest* attribute will enable mobile users to access offline web pages from the application cache [1]. For web browsers running on mobile devices, the *manifest* feature provides a dramatic improvement

¹ <http://www.aserl.org>

² <http://www.lockss.org>

³ <http://www.ndltd.org>

⁴ <http://www.metaarchive.org>

⁵ <http://www.w3.org/html/wg/>

⁶ <http://www.whatwg.org/>

in loading performance. Another feature is the addition of elements/tags, e.g., <video> and <audio>. The HTML5 video element enables video contents to be shown with web browsers without any plug-in (e.g., Adobe Flash) [8].

These novel features of HTML5 can be leveraged for the migration of ETDs to the mobile web environment. Specifically, multimedia elements such as <video>, <audio>, <canvas>, and <figure> can offer explicit links between ETD text and supplementary multimedia files when there only are implicit links. The explicit links can directly connect separate multimedia and PDF files. Supporting all this is analysis of ETD structure as described in the next subsection.

B. ETD Structure

An ETD web (“splash”) page has metadata (e.g., type of document, authors, etc.) and links to thesis/dissertation files which the metadata describes (see Figure 1 (a)). Most ETDs are presented in a single PDF (portable document format) file. But many also have several multimedia files as supplements.

Such an ETD, with several digital media files, may lack explicit links among its files, beyond what is provided in the metadata. When there are no explicit links, structuring the relationship between files belonging to the ETD can aid navigation and browsing. Figure 1 (b) illustrates an example of a structuring process aimed to identify ETD file relationships.

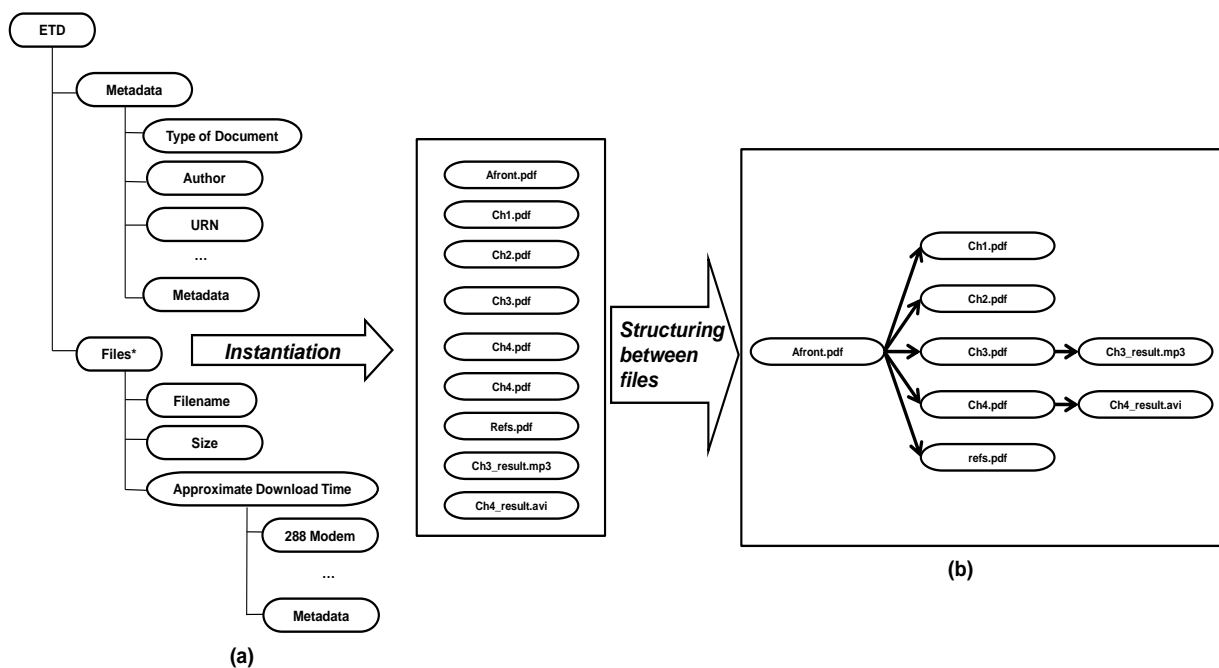


Figure 1. (a) Example of an ETD structure and (b) structuring the relationship between files belonging to the ETD

C. Issues for migration strategy

We developed and demonstrated a software prototype to convert plain ETDs into HTML5. Subsequently, due to ongoing HTML5 standardization [1], we had to answer the following research questions faced during the development effort.

- How will the conversion to HTML5 be conducted?
- Which browsers will support HTML5?

- Which video file formats are supported by current browsers?
- Which video file format converters will convert into supported file formats?
- Which pdf2txt extractors are most effective?
- How will an HTML5 ETD work on mobile devices (e.g., Android phone, iPod, iPad)?

3. HTML5 ETD CONVERTER ALGORITHM & IMPLEMENTATION

One of the primary requirements for HTML5 ETDs was to provide better access from mobile devices. To meet this requirement, we find link candidates between the PDF ETD and multimedia supplements, and then explicitly connect them. Our HTML5 ETD converter consists of two parts: *link information extractor* and *tagger*. Figure 2 shows the conversion process and the modules involved.

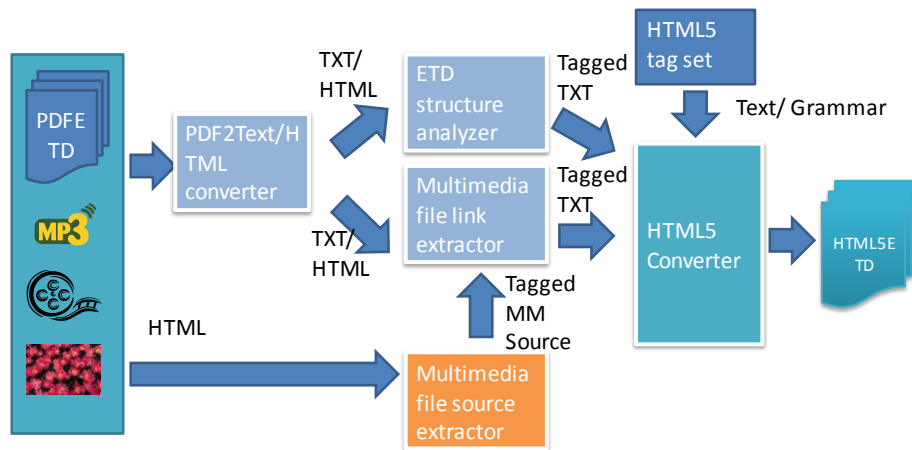


Figure 2. Flow chart of HTML5 ETD converter

A. PDF2TXT/HTML

The PDF2TXT/HTML module converts from the PDF presentation format into an intermediate format (e.g., plain text) or a semi-presentation format (e.g., HTML), so we can find some link candidates and then add useful HTML5 tags (e.g., video, audio, etc.) to the plain text or the HTML file. Obviously, this process would have even wider utility if it could start with document images captured by a scanner, as input data, and then extract text from the document images using Optical Character Recognition (OCR). But that could occur in a prior step; our algorithm instead starts with documents which were saved as text files, not raster images. Accordingly, since the start is plain text, there is the problem of loss of ETD formatting information; that is considered below in section 4.

The algorithm for extracting text from the PDFs is relatively straightforward, but does have some problems. The basic approach is to use the PDFParser class of PDFBox⁷ to parse the entire PDF document. Once this has been done, it is possible to use the PDFTextStripper class (also of PDFBox) to extract the PDF's text from the parser. The text stripper makes it easy to get a simple string representation of text segments within the PDF. It is trivial to prepend (and append) any desired HTML5 tags to each string before writing to a file. If the tags are added in a logical order, the result is a simple HTML page with the extracted text.

B. STRUCTURE ANALYSIS

⁷ <http://pdfbox.apache.org>

Once a PDF file has been converted into plain text/HTML, the “Table of Content (ToC)” section is parsed, and the *inter-structure* between the logical page structure and logical structure is analyzed. When a ToC section is parsed, each heading, separator (e.g., ' ' (blank), '-', ':', etc.), and logical page is segmented and recognized. Analyzing a ToC returns the entire logical structure of the document, which has more semantic information (e.g., chapters, sections) for human users. This information supports inserting more useful HTML5 tags (e.g., header, chapter, section) into the ETD.

This algorithm was separately implemented by the first author as a small project named “Table of Content Analysis for ETD Structuring”. The segmentation of headings, separators, and logical pages from the ToC was implemented using regular expressions (based on rules observed by exploratory data analysis of ETD tables of contents).

C. MULTIMEDIA LINK SOURCE EXTRACTION

In the Multimedia Link Source Extraction, source information for multimedia files which should be linked with original theses/dissertations (e.g., file names) are extracted from the ETD main web pages. This information which is extracted is a value for the 'src' property in the 'video' or 'audio' tags. For example, Figure 3 shows an ETD title page and its multimedia link sources metadata for a specific supplement file as the multimedia link sources (see video file names in a block).

This algorithm is coded in Perl. To parse a HTML file, the Perl package for HTML parsing is used. Afterward, this code is integrated with the HTML5 main graphical user interface written in JAVASDK and the Java Standard Widget Toolkit (Java SWT).

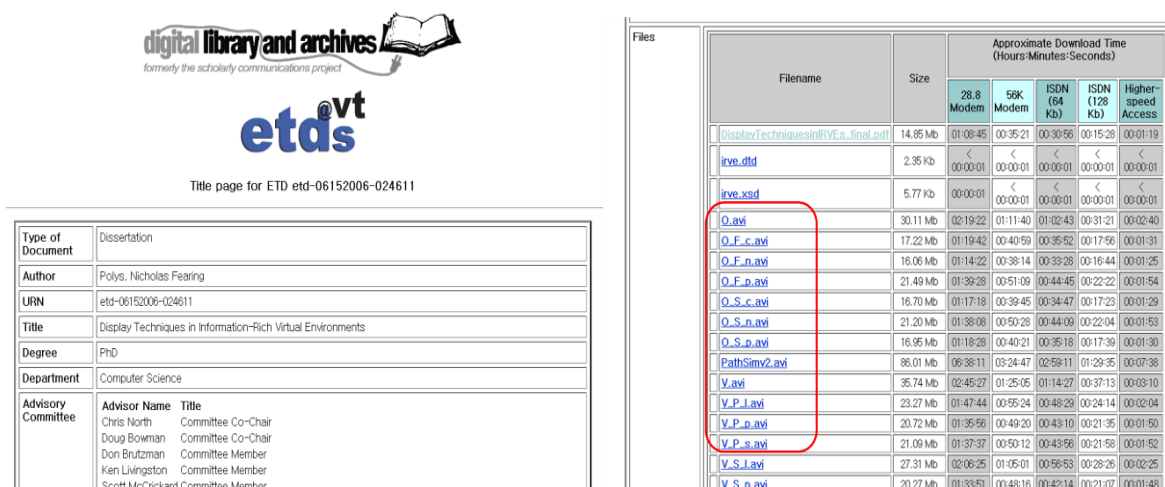


Figure 3. Example of metadata and multimedia link sources in an ETD title page

D. MULTIMEDIA LINK EXTRACTION

Multimedia link extraction is a process which identifies link candidates from the plain ETD text that results from PDF2TXT/HTML. The input of this process is the multimedia link sources identified in the previous process (MULTIMEDIA LINK SOURCE EXTRACTOR). Matches are sought in the plain text. If more than one link candidate has been found, all are tagged. This process returns a tagged text file with multimedia type attributes (e.g., video or audio and so on). Figure 4 presents an example of multimedia link candidates in the ETD *context* to be identified and then, eventually, surrounded by the video element of HTML5.

Link Candidates: Video file names (.avi)	
V_S_s.avi	A video demonstrating the HUD BorderLayout technique with semi-transparent polygon connector (Experiment 4)
V_S_p.avi	A video demonstrating the HUD BorderLayout technique with opaque polygon connector (Experiment 4)
V_P_l.avi	A video demonstrating the HUD Proximity BorderLayout technique with line connector (Experiment 4)
V_P_s.avi	A video demonstrating the HUD Proximity BorderLayout technique with semi-transparent polygon connector (Experiment 4)
V_P_p.avi	A video demonstrating the HUD Proximity BorderLayout technique with opaque polygon connector (Experiment 4)

Figure 4. Example of multimedia link candidates in the ETD context

This process is also implemented in Perl. To match a multimedia link source (e.g., file name) to candidate links, simple string matching is employed. Afterward, this code is integrated with the HTML5 main graphical user interface written in Java and Java SWT.

E. HTML5 CONVERSION

This process combines all information for producing a document in HTML5 format. Useful HTML5 tags, which are selected according to a specific task (e.g., ETD conversion), a plain text ETD with link candidate tags, and candidate link sources are used to generate an HTML5 ETD. Multimedia candidate link tags (e.g., file names, figures, references) are replaced by multimedia link source information (e.g., URL) and HTML5 tags such as <video>, <audio>.

A large part of the conversion to HTML5 is taken care of while outputting the text during the first step, PDF2TXT. This step of the conversion sets up the basic HTML5 page beginning with <!DOCTYPE HTML>, and including head, body, and other tags. The more interesting part of the conversion comes up with the actual video insertion and tagging. After we have a basic HTML document filled with text, the program will look through that output for any video files listed according to a certain format that many of the ETDs use. If such videos are found, the program will insert <video> tags surrounding them and also add information where the source for the video can be found. Figure 5 illustrates a main screen of the HTML5 converter. This tool is written in Java SDK for general functionalities, Java SWT for graphical user interfaces along with the PDFBox, and the Java open library for the PDF parsing and text extracting. It integrates Perl scripts for the multimedia source extractor, multimedia link candidate extractor, and ETD structure analysis. An example HTML5 ETD

and its page source are shown in Figures 6 and 7, respectively. Our test browser for the HTML5 ETD is the latest version of Mozilla Firefox (3.6.3) which supports the video element HTML5, but it only supports the video file format *ogv/ogg*. (Note that the O.ogg and V.ogg file extensions were edited manually for the purpose of demonstration.)

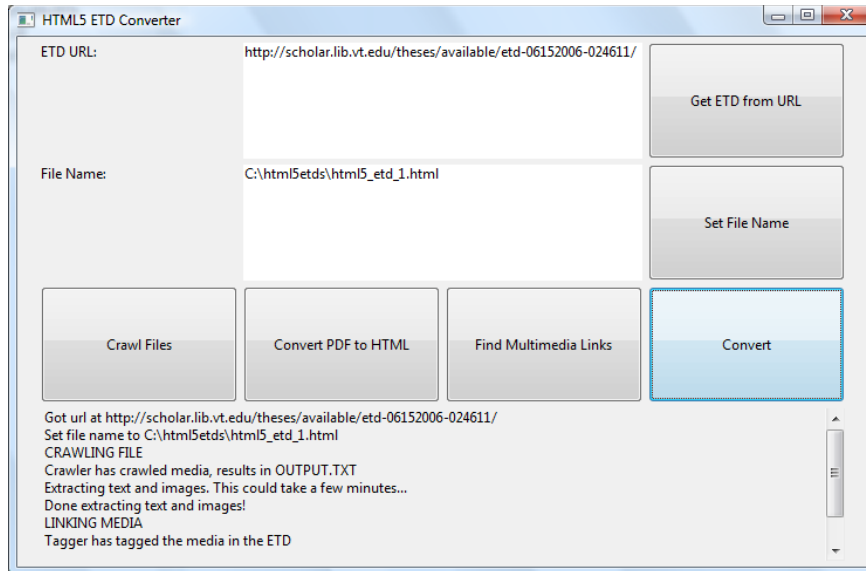


Figure 5. Main screen of HTML5 ETD converter

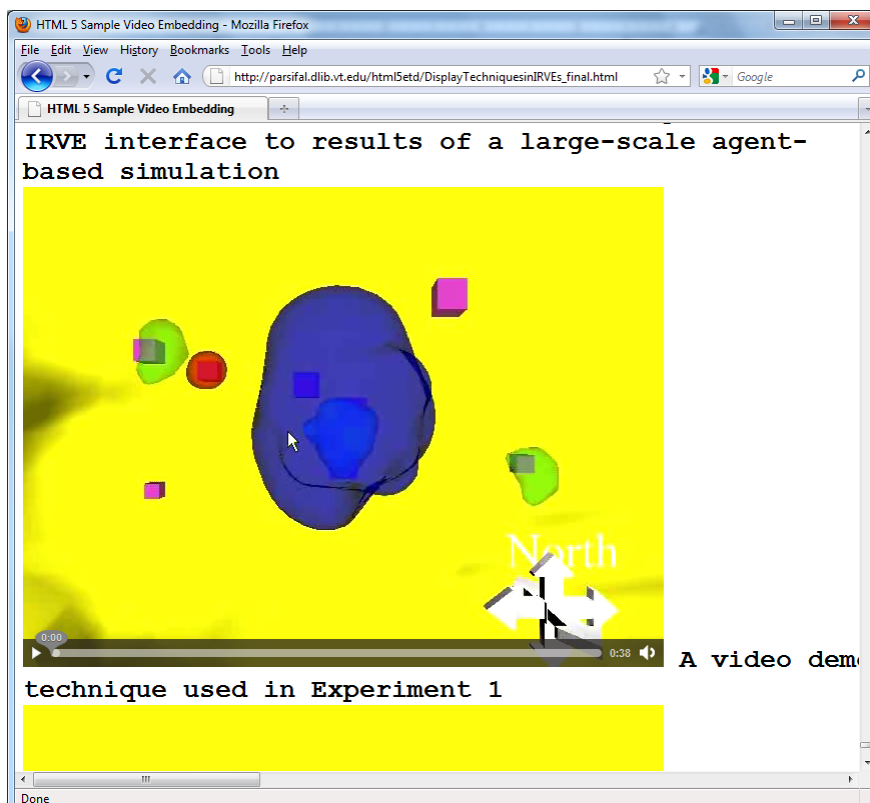
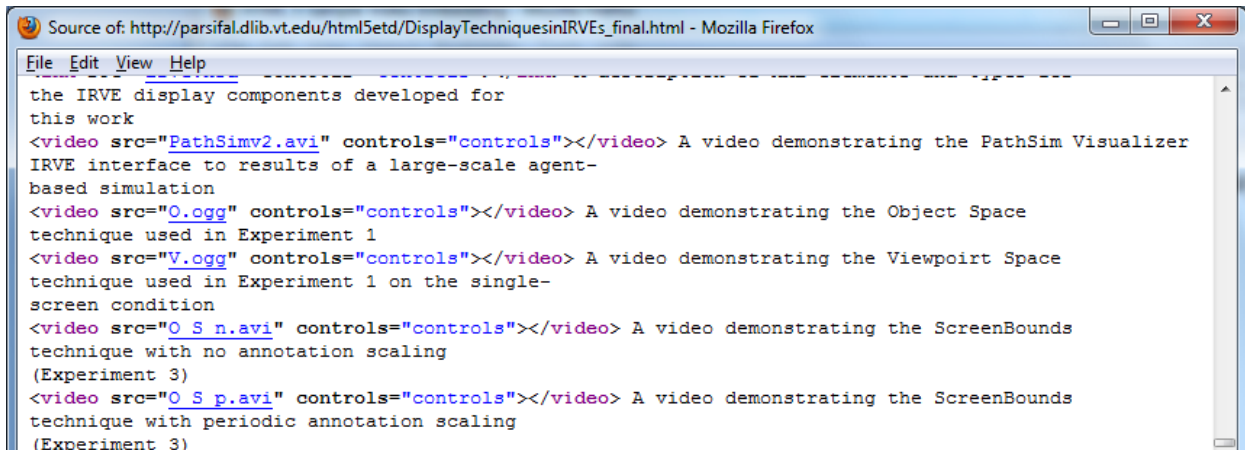


Figure 6. Example of an HTML5 ETD in the Mozilla Firefox (version 3.6.3)



```
Source of: http://parsifal.dlib.vt.edu/html5setd/DisplayTechniquesinIRVEs_final.html - Mozilla Firefox
File Edit View Help
the IRVE display components developed for
this work
<video src="PathSimv2.avi" controls="controls"></video> A video demonstrating the PathSim Visualizer
IRVE interface to results of a large-scale agent-
based simulation
<video src="O.ogg" controls="controls"></video> A video demonstrating the Object Space
technique used in Experiment 1
<video src="V.ogg" controls="controls"></video> A video demonstrating the Viewpoint Space
technique used in Experiment 1 on the single-
screen condition
<video src="O S n.avi" controls="controls"></video> A video demonstrating the ScreenBounds
technique with no annotation scaling
(Experiment 3)
<video src="O S p.avi" controls="controls"></video> A video demonstrating the ScreenBounds
technique with periodic annotation scaling
(Experiment 3)
```

Figure 7. Page source of the figure above

4. DISCUSSION

In this section, we discuss lessons learned through this research work, i.e., problems that we faced and the best solutions that we found.

A. Problems

The first problem we faced was how to migrate from a PDF file into an HTML5 file. To begin with, we wanted to extract all of the text from the PDF file. This was a problem at first; there didn't exist a standard Java library that we could use to extract text from the PDF file. The second problem we faced was that in some of the PDF files we had, there were images. Obviously a text extraction tool would not allow us to extract images from PDF.

After committing to a library, PDFBox, to extract text and images, we ran into a problem with the formatting and styling of the text. When the text is taken out of the PDF, it is a plain text file which no longer keeps the formatting and style information (e.g., size, color, etc.) of the font that was being used. This is bad for preservation and archiving because it is harder to find elements like headings; there is much information loss. But, for mobile devices with a small screen size, connected through a narrow bandwidth wireless network, formatting and styling may not seem to matter. Another problem is that for some images in PDF files, erroneous stacking of parts occurs, leading to finding of many pieces of an image.

Yet another problem is that not all browsers support full HTML5 capabilities. The latest versions of Internet Explorer and Opera do not support the video tag at all but Mozilla Firefox, Google Chrome, and Safari do support it. Also, Firefox, Chrome and Safari all support audio files, but IE and Opera do not.

Further, a goal for HTML5 is support of mobile devices, such as cell phones or iPods. The iPod touch, iPhone, and iPad do support HTML5. But, cell phones with Android 2.1, and Blackberrys, don't support HTML5 video. This is a concern because of the number of people that use mobile devices every day.

B. Solutions

PDFBox was the best solution we found for extracting text and images from PDF. There exist classes in PDFBox to easily take out the text and images from the PDF, and put the text into a string (as needed, with start or end tags added). The images are saved in the source path images file.

As for the problem with multiple parts for one image, we have no real solution. Further research, and possibly changes to PDFBox, may be needed.

Regarding the problems mentioned about browsers, the simplest solution is to be sure to pick a browser that has all the features desired. In addition there is the matter of HTML5's embed tag. Since we want to get away from relying on the user's computer to have the appropriate (plug-in) software, we recommend not using the embed tag to view video.

As for the problem with file types that can be used with the video tag, we propose converting all video files to *ogv*. This will allow for the largest number of browser types to view the videos. We have converter software that will convert to *ogv* format; the only downside to this is that we must download the video, convert it, and then put it into the new ETD file. To avoid this, we suggest a better solution. The files that are not *ogv* already should be converted to *ogv* before putting them onto the website. This would speed up the process, because instead of downloading and converting, which can take a lot of time, we would be able to just point to the *ogv* file, and not actually have it downloaded again.

C. Mobile Adaptation in Digital Libraries

For an ETD's sustainability in the evolving information environment, the ETD should be able to be adaptive to its environment. In particular, the ETD should adapt its structure to the mobile learning environment.

Adaptation issues in digital libraries are generally categorized into two approaches: *system-oriented adaptation* [12, 13, and 14] and *user-oriented adaptation* [14, 15, 16, 17, 18, and 19]. System-oriented adaptation can further be divided into adaptation to a specific software application (e.g., browsers), adaptation to a specific hardware device (e.g., small-size displays), and adaptation to a specific computer network (wireless network). User-oriented adaptation can also be grouped into adaptation to a specific class of user (beginner vs. expert, handicapped) and a specific user task (e.g., learning, collaboration). Accordingly, we stipulate that HTML5 ETDs should be accessible by general users, with mobile web browsers, from wireless networks. In general, digital libraries should be accessible on mobile devices [20]. To check for this, we recommend using mobile learning evaluation metrics [21].

5. CONCLUSION

In this paper, the conversion of PDF ETD and multimedia files into HTML5 ETDs has been studied. This can be thought of as a migration to a better format for preservation. We have transformed 3 multimedia plus 3 linked (hypertext/hypermedia) ETDs into HTML5 in a semi-automatic way. We have faced many research challenges, and developed solutions to the problems encountered.

In the future, beyond migration, further adaptation of HTML5 ETDs to the mobile web and the semantic web can be investigated. Content adaptation should be explored, to meet further requirements of: individuals, mobile web browsers, and small-size screens on mobile devices. Moreover, by using Microdata [22] and RDFa [23], a HTML5 ETD can be adapted to the semantic web, which aims to create machine readable contents that will allow additional services and will aid further adaptation, for emerging environments.

ACKNOWLEDGMENT

Special thanks go to Channy Yun, Nadia Puchalski, Seungwon Yang, and Paul Mather, for their constructive comments on the HTML5 specification, compound digital objects, document readability on mobile devices, and practical issues regarding ETDs, respectively.

REFERENCES

- [1] W3C, HTML5, <http://dev.w3.org/html5/spec/Overview.html>, Accessed 12 May 2010

- [2] David S. H. Rosenthal, Vicky Reich, “Permanent Web Publishing”, Proceedings of FREENIX Track: 2000 USENIX Annual Technical Conference, San Diego, CA, USA, June 18-23, 2000
- [3] John Burger, “ASERL and LOCKSS to Preserve e-Theses & Dissertations”, SOLINET Press release, <http://scholar.lib.vt.edu/theses/ETDsASERLLOCKSS20050711PR.pdf>, July 11, 2005
- [4] Gail McMillan, Katherin Skinner, NDLTD Preservation Strategy with the MetaArchive Cooperative, May 19, 2008, Accessed 14 May 2010, <http://scholar.lib.vt.edu/theses/preservation/NDLTDPreservationPlan20080520.pdf>
- [5] David S. H. Rosenthal, Thomas Lipkis, Thomas S. Robertson, Seth Morabito, “Transparent Format Migration of Preserved Web Content,” D-Lib Magazine, Vol. 11, No., 1, January 2005
- [6] Van Wijk, Caroline, Starting Point for Migration Research. Migration Research Project, Koninklijke Bibliotheek. July 2006. http://www.kb.nl/hrd/dd/dd_projecten/Starting_Point_Migration_Research.pdf
- [7] Jonathan Leidig, AJ Alon, Amine Chigani, Mahima Gopalakrishnan, Sung Hee Park, (Editor: Edward A. Fox), Digital Libraries/File formats, transformation, migration, Accessed 12 May 2010, http://en.wikiversity.org/wiki/Digital_Libraries/File
- [8] Wikipedia, “HTML5.” Web. <http://en.wikipedia.org/wiki/Html5>, Accessed 07 May 2010
- [9] W3C, HTML 4.01 Specification, W3C Recommendation 24 December 1999, <http://www.w3.org/TR/REC-html40/>, Accessed 20 May 2010
- [10] W3C, XHTML™ 1.0 The Extensible HyperText Markup Language (Second Edition), A Reformulation of HTML 4 in XML 1.0, W3C Recommendation 26 January 2000, revised 1 August 2002, <http://www.w3.org/TR/xhtml1/>, Accessed 20, May 2010
- [11] W3C, Document Object Model (DOM) Level 2 Core Specification, Version 1.0, W3C Recommendation 13 November, 2000, <http://www.w3.org/TR/DOM-Level-2-Core/>, Accessed 20, May 2010
- [12] Javed I. Khan, Qiong Gu, “Network Aware Symbiotic Video Transcoding for In Stream Rate Adaptation on Interactive Transport Control”, Proceedings of the IEEE International Symposium on Network Computing and Applications, Oct. 2001
- [13] Randy H. Katz, “Adaptation and Mobility in Wireless Information Systems”, IEEE Personal Communications Magazine, Volume 1, Number 1, First Quarter, 1994
- [14] Bharat Bhargava, Melliyal Annamalai, Evaggelia Pitoura, “Digital Library Services in Mobile Computing”, SIGMOD Record, Vol. 24, No., 4, December 1995
- [15] Mark Perry, Kenton O’ara, Abigail Sellen, Barry Brown, Richard Harper, “Dealing with Mobility: Understanding Access Anytime, Anywhere”, ACM Transactions on Computer-Human Interaction, Vol. 8, No, 4, December 2001, pp. 323-347
- [16] Matt Jones, Emman Thom, David Bainbridge, David Frohlich, “Mobility, Digital Libraries and a Rural Indian Village,” JCDL ’09, pp. 309-312, June 15-19, 2009, Austin, Texas, USA
- [17] Benjamin B. Bederson, Alex Quinn, Allison Druin, “Designing the Reading Experience for Scanned Multi-lingual Picture Books on Mobile Phones,” JCDL ’09, pp. 305-308, June 15-19, 2009, Austin, Texas, USA
- [18] Allison Druin, Benjamin B. Bederson, Alex Quinn, “Designing Intergenerational Mobile Storytelling”, IDC 2009, 325-328, June 3-5, 2009, Como, Italy
- [19] Jonathan Hey, Jaspal S. Sandhu, Catherine Newman, Jui-Shan Hsu, Charlotte Daniels, Esha Datta, Alice M. Agogino, “Designing Mobile Digital Library Services for Pre-engineering and Technology Literacy”, Int. J. Engng Ed. Vol. 23, No. 3, pp. 441-453, 2007
- [20] David Bainbridge, Steve Jones, Sam McIntosh, Matt Jones, Ian H. Witten, “Portable Digital Libraries on an iPod,” JCDL ’08, pp. 333-336, June 16-20, 2008, Pittsburgh, Pennsylvania, USA
- [21] Antti Syvanen, Petri Nokelainen, “Evaluation of the technical and pedagogical mobile usability”, MLEARN 2004, pp. 191-196, July 5-6, 2004, Rome, Italy
- [22] W3C, HTML Microdata Working Draft, <http://www.w3.org/TR/microdata>, Accessed 12 May 2010
- [23] W3C, RDFa Primer, 14 October 2008, <http://www.w3.org/TR/xhtml-rdfa-primer/>, Accessed 11 March 2010