Lab II – Blackboard Archive Extractor

CS 411W Lab II

Prototype Product Specification
For
Blackboard Archive Extractor

Prepared by: Austin Tillery, Team Crystal

Date: 4/14/17

Table of Contents

Example:

List of Figures

Example:

1 Introduction

Blackboard Archive Extractor, or BAE, is an application that allows the user to load a Blackboard Course Archive and explore the contents of the archive. The application will allow professors an automated means of extracting Blackboard content such as PDFs, homework, and quizzes for faculty review.

1.1 Purpose

The end user will be teaching professionals who use Blackboard to teach students and ultimately use the Blackboard archiving system to store old Blackboard content for later purposes such as faculty review. This application will be an automated alternative to the current manual process of extracting Blackboard content archives. But it will not be of use in extracting PLE content or anything unrelated to Blackboard.

1.2 Scope

The distribution of the resulting application will be limited to the testing and development team. The prototype will demonstrate the functionality and efficiency of extracting archive blackboard data files. This will include all archive data files such as, homework assignments, journal posts, quizzes, and the tests will be stimulated. Also the product will create a directory containing all of the extracted and processed contents from the Blackboard Course Archive. The product's ultimate goal is to create an algorithm for converting archive data files into fully functional webpages. As shown in Figure 1, the manifest file is zipped into a zip archive.

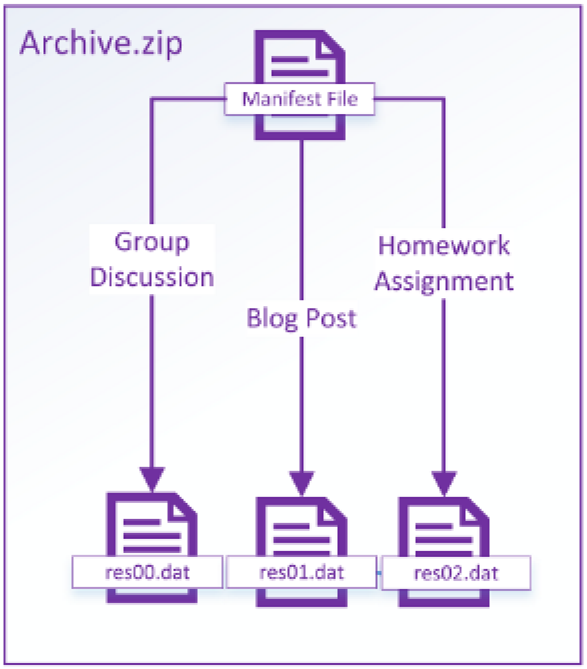


Figure 1 Manifest file

1.3 Definitions, Acronyms, and Abbreviations

Graphical User Interface- The visual way of interacting with the application using items such as buttons

User Interface - the application interface through which a user utilizes the features of the Blackboard Extractor software.

User Experience – The feedback that is collected from users when describing how they feel about Blackboard Extractor

Root Index – Refers to the top-level directory of the file system

Team Foundation Server – Microsoft Product provides source code management

Microsoft**.**NET - Microsoft's implementation of the command line interface and serves as the execution engine of .NET Framework.

1.4 References

CS 410 documents: milestones and prototype

Lab I

CS 411 Software Requirements

Corcoran, B. (2014, July 23). Blackboard's Jay Bhatt Strikes Up the Brass Band. Retrieved October 19, 2016, from https://www.edsurge.com/news/2014-07-23-Blackboard-s-jay-bhatt-strikes-up-the-brass-band?elq=29068c31188e4150bde7dd6db8a3dab7&elqCampaignId=7599

ODU Faculty Handbook. (2005, December 21). Retrieved October 26, 2016, from http://ww2.odu.edu/ao/facultyhandbook/index.php?page=ch07s01.html

BFree: Extract Blackboard content. (n.d.). Retrieved October 19, 2016, from https://its2.unc.edu/tl/tli/bFree/index.html

Blackboard | Reimagine Education | Education Technology. (n.d.). Retrieved October 19, 2016, from http://www.Blackboard.com/

".NET - Powerful Open Source Cross Platform Development." .NET - Powerful Open Source Cross Platform Development. Microsoft, n.d. Web. 15 Dec. 2016.

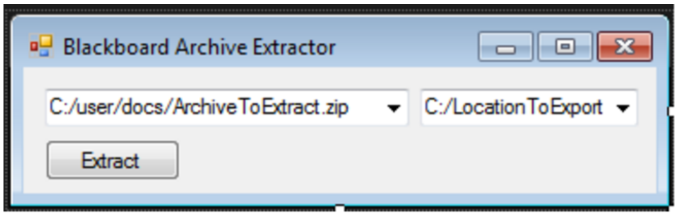
1.5 Overview

This product specification provides the hardware and software configuration, external interfaces, capabilities and features of the Blackboard Archive Extractor prototype. The information provided in the remaining sections of this document includes a detailed description of the hardware, software, and external interface architecture of the BAE prototype; the key features of the prototype; the parameters that will be used to control, manage, or establish that feature; and the performance characteristics of that feature in terms of outputs, displays, and user interaction.

2 Prototype Architecture Description

Interactive GUI

The GUI will be simplified for any teacher to use and will be easy to comprehend. The GUI is a simplified input and output box where users can drag and drop a Blackboard archive to extract. Then the application will automatically open the index file to display the webpage.



C:/user/docs/ArchiveToExtract.zip

C:/LocationToExport

Figure 2 Extract GUI

Automation

For each archive the professor will need to manually extract the archive from Blackboard. The extracted archive is loaded into the application in which the content is automatically processed. The archived data files are automatically linked by matching the IDs given to them by Blackboard. The content is automatically formatted and extracted to a webpage through the use of xlst which parses the xml to html. Then the professor repeats the process for each archive.

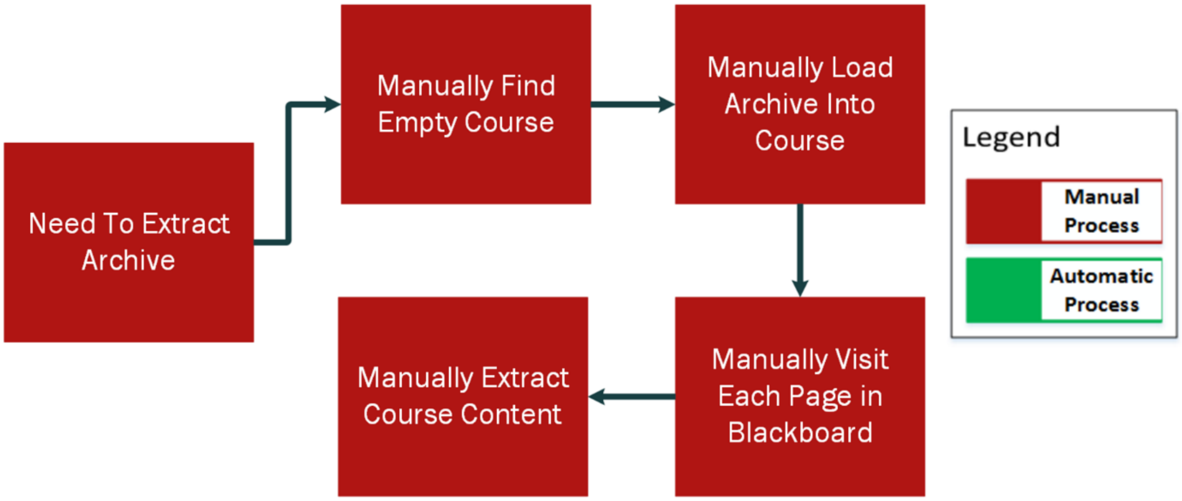

Need To Extract Archive
Manually Find Empty Course
Manually Load Archive Into Course
Manually Visit Each Page in Blackboard
Manually Extract Course Content
Legend
Manual Process
Automatic Process

Figure 3 Current Manual Process Flow

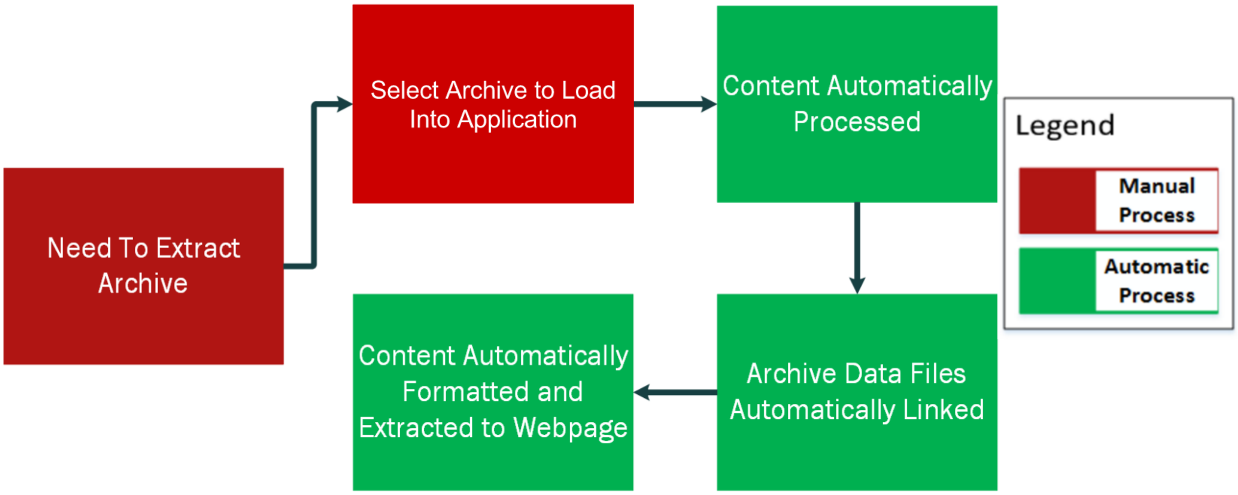

Need To Extract Archive
Select Archive to Load Into Application
Content Automatically Processed
Archive Data Files Automatically Linked
Content Automatically Formatted and Extracted to Webpage
Legend
Manual Process
Automatic Process

Figure 4 Revised Automated Process Flow

Blackboard Archive Extractor is comprised of the following major components:

- <u>Blackboard Extractor Interface</u>- The archive blackboard file is extracted using the GUI or command line.
- <u>Parsing</u>- The XSLT is used to parse the xml to html when the blackboard archive is extracted. XSLT is a language for transforming XML documents into other XML documents, or other formats such as HTML for web pages, plain text or

XSL Formatting Objects, which may subsequently be converted to other formats, such as PDF, PostScript and PNG.

- Linking- The archived data files are automatically linked by matching the IDs given to them by Blackboard.
- Reformatting- The archived data is organized into information that is similar to the original format of the website for instance, the professor's blackboard webpage.

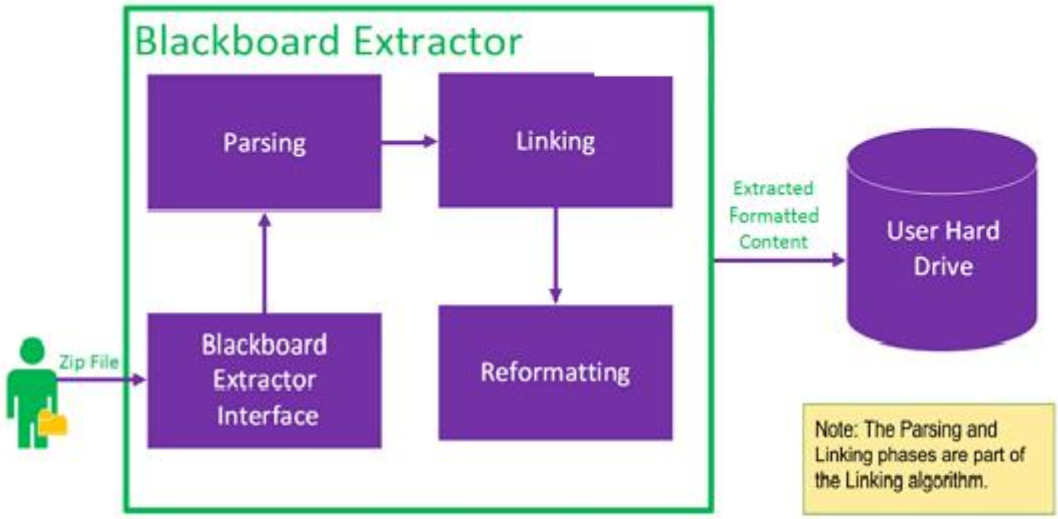

Blackboard Extractor
Parsing
Linking
Zip File
Blackboard Extractor Interface
Reformatting
Extracted Formatted Content
User Hard Drive
Note: The Parsing and Linking phases are part of the Linking algorithm.

Figure 5 Major Functional Requirements

2.1 Prototype Functional Description

When the professor uses the application, it will load a Blackboard Course Archive and explore the contents of the archive. The application will parse through all types of Blackboard content including blog posts, module items, and pdfs. The application will also create a directory containing all of the extracted and processed contents from the Blackboard Course Archive. The extracted data files will be linked together by relation (i.e. parent/child, hyperlinks) to each other using the linking algorithm. The interpreted data will then be formatted by building a file structure containing all of the contents indexed by HTML files. In result the archive will be viewable in a set of HTML pages.

3 Specific Requirements

The development of Blackboard Archive Extractor will require three main specific requirements; internal interface, HTML content, and XML parsing.

3.1 Functional Requirements

Graphical User Interface. When the application is launched with no command line arguments, the graphical user interface will be displayed.

Command Line Arguments. When the application is launched with two command line arguments, the arguments will be interpreted as the following:

HTML Content Pages. The application will extract the content of given Blackboard Course Archive as a series of HTML web pages.

HTML Index Page. The application will create an HTML index page to index the web pages created in requirement 1.1.5.

Manifest XML parser. The application will build the HTML index page based on the top level links found in the manifest file of the given Blackboard Course Archive.

Resource XML parser. The application will build the HTML content pages based on the resource xml files and their relation to the manifest.

File management. The application will move files such as pdfs, PowerPoints, and images from the archive to that they can be accessed from the website.

HTTP Link Validation. The application will validate the availability of any HTTP links that reference locations outside of the blackboard course.

3.2 Assumption and Constraints

The user provides a valid Blackboard Archive file.

If the user fails to provide a valid Blackboard Archive, an error message will be displayed on the screen.

3.3 Non-Functional Requirements

Maintainability

The architecture of the application will allow support for future versions of blackboard. Features introduced in future updates may not be compatible with the Blackboard Archive Extractor prototype.

The Blackboard Archive Extractor prototype will not receive any updates following the completion of the application.

Any links to websites which are no longer available, which return HTTP 400 response codes, will be marked as “dead” and displayed as such on the output website.

Any links which are still available will be hyperlinked on the website.