
Lab 1 – Blackboard Extractor

Austin Tillery

CS 411

Thomas J. Kennedy

22 January 2017

Outline

1. Introduction
 - a) Blackboard in Learning Institutions
 - b) ODU Faculty Review Process
 - c) Solution Overview
2. Blackboard Extractor Product Description
 - a) Product Features
 - i. Interactive GUI
 - ii. Automation
 - iii. User Experience?
 - b) Major Components
 - i. MFCD
 - ii. Parsing
 - iii. Linking
 - iv. Reformatting
 - c) Customer Base
 - i. ODU Faculty
 - ii. All Educators Using Blackboard
3. Blackboard Extractor Prototype Description
 - a) Prototype Goals
 - i. Fully Functioning Application
 - b) Prototype challenges
 - i. Blackboard Changes Structure
 - ii. Customers are Unhappy

List of Figures

Figure 1: Blackboard Logo

Figure 2: Current Manual Process Flow

Figure 3 Interactive GUI

Figure 4: Automated Process Flow

Figure 5: MFCD

Figure 6: Linking Algorithm Iteration

Figure 7: Linking Algorithm Diagram

Figure 8: Formatting Iteration

Introduction

Blackboard in Learning Institutions

Blackboard is an online tool used by schools, colleges, and universities to organize the course materials of a class for both professors and students. It allows professors to provide information such as syllabi and homework assignments to students over the Internet. In 2014, approximately 17,000 schools were using Blackboard, in over 100 different countries including Old Dominion.¹

Blackboard has a number of features that help teachers and students to include:

- Incorporation of files, graphics, and audio/video clips into course materials
- Facilitated communication through:
 - Announcements posted in one central location
 - Email messages sent from within Blackboard
 - Discussion boards
 - Desktop web conferencing
- Assignment management
- Plagiarism detection
- Tests and Surveys
- Online gradebook including rubrics
- Blogs, wikis, and journals
- Grade upload from the Blackboard gradebook



- Automated integration of the course roster
- Utilization of the University NetID login
- Course copying from one semester to the next

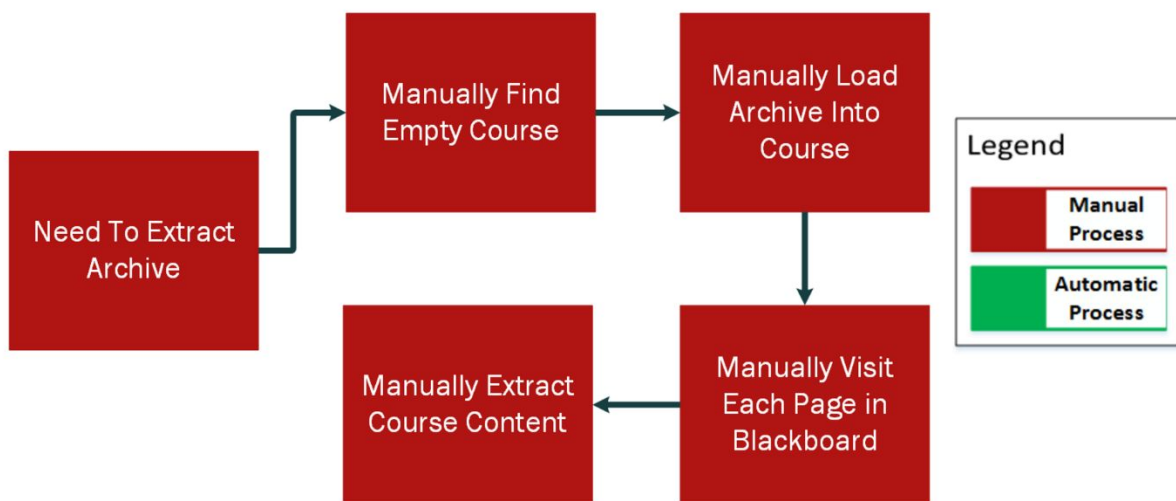
ODU Faculty Review Process

Teachers in educational institutions such as Old Dominion are reviewed by the university board or department head to evaluate their performance. For example, according to the Old Dominion University Handbook, lecturers are reviewed every year, senior lecturers are reviewed every three years, and tenured professors are reviewed every five years.² Professors are required to present all materials used throughout all courses taught since the previous review, including any material stored on Blackboard. Blackboard only stores courses for two years, before which the course must be exported as a Blackboard Archive file. Although this is helpful for teachers who need to store their teaching material, the when the archive file is retrieved there is an excruciating and monotonous amount of steps to make it presentable for review due to a number of issues. The most prominent issue of Blackboard Course Archives is that the process is entirely manual.

Currently, extracting a Blackboard Archive requires a professor to: Find an unused Blackboard course into which he or she can load the Archive. If there is no unused Blackboard course, it is not possible to extract the content. Manually download course content,

page-by-page. Manually format folder structure. This process must be repeated for each course Archive. This has led us to our problem statement: Professors have difficulty retrieving course content after it has been exported to a Blackboard Course Archive.

Current Manual Process Flow



Solution Overview

Our team's solution to this problem is to make a blackboard extractor that allows the extraction process to be automatic. Our team will be automating the process for extracting Blackboard course archives by implementing software that will: process, link, and format data automatically. This will provide users with a solution that takes less time and effort than the current process.

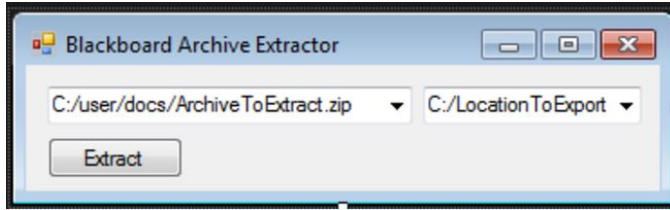
Blackboard Extractor Product Description

Product Features

Blackboard Extractor will be an application which allows a user to load a Blackboard Course Archive and explore the contents of the archive. This application will be able to handle all types of Blackboard content including blog posts, module items, and pdfs. The application will create a directory containing all of the extracted and processed contents from the Blackboard Course Archive.

Interactive GUI

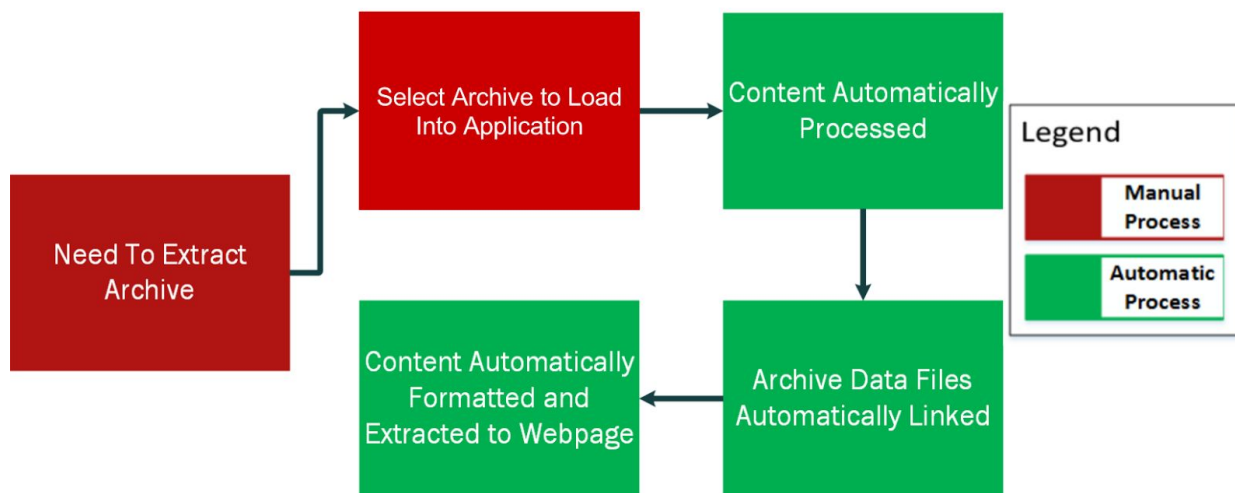
The GUI will be simplified for any teacher to use and will be easy to comprehend. The GUI is a simplified input and output box where users can drag and drop a Blackboard archive to extract. Then the application will automatically open the index file to display the webpage



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 xslt which parses the xml to html. Then the professor repeats the process for each archive.

Revised Automated Process Flow



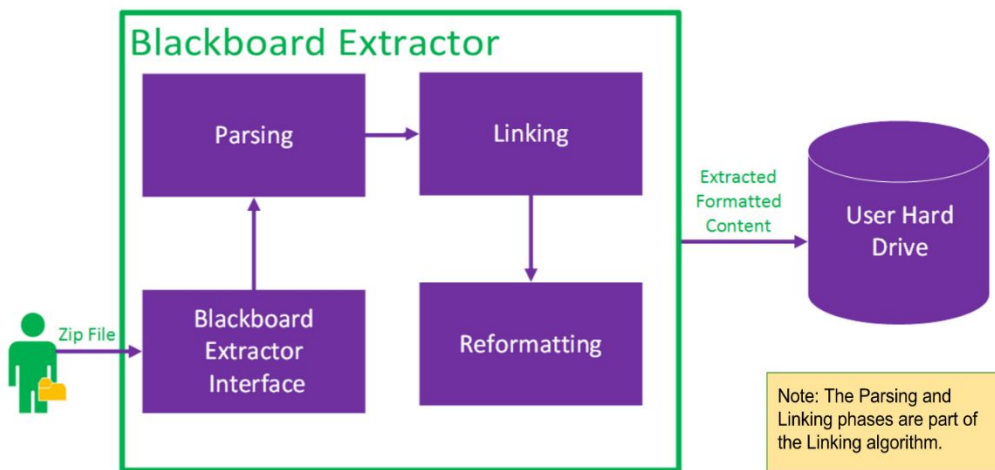
User Experience

The users of the software will be able to provide feedback on how the application can be improved such as GUI interface layout and extraction process.

Major Components

MFCD

Major Functional Components



Parsing

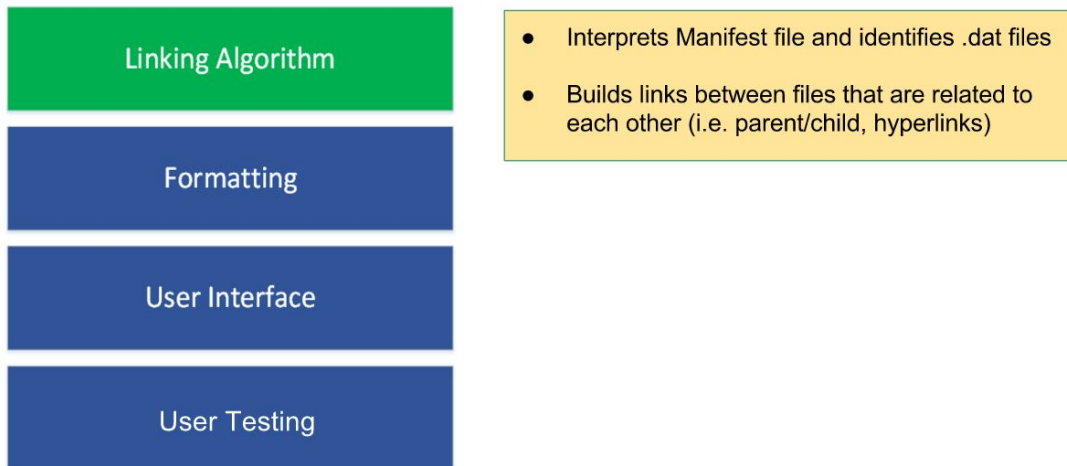
The XSLT is used to parse the xml to html when the blackboard archive is extracted.

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. So for example, to create a file that will convert an XML data file exported from a program we will use to an HTML report. In order for a web browser to be able automatically to apply an XSL transformation to an XML document on display, an XML stylesheet processing instruction can be inserted into XML.

Linking

After the parsing process, the archived data files are automatically linked by matching the IDs given to them by Blackboard.

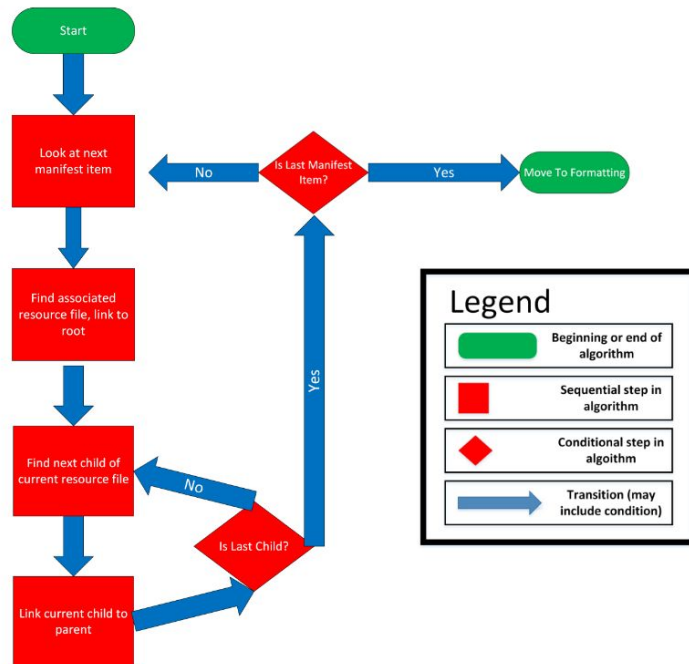
First Iteration: Linking Algorithm



Using the linking algorithm, we will interpret the manifest file and identify .dat files. This builds the links between files that are related to each other for example, hyperlinks. The diagram below shows how the linking algorithm is implemented:

- The application looks at the archive data manifest item.
- The application finds the associated resource file and links it to root.
- The next child of the current resource file is linked to parent and is repeated until the last child.
- The application repeats until the last manifest item in which then it goes to formatting.

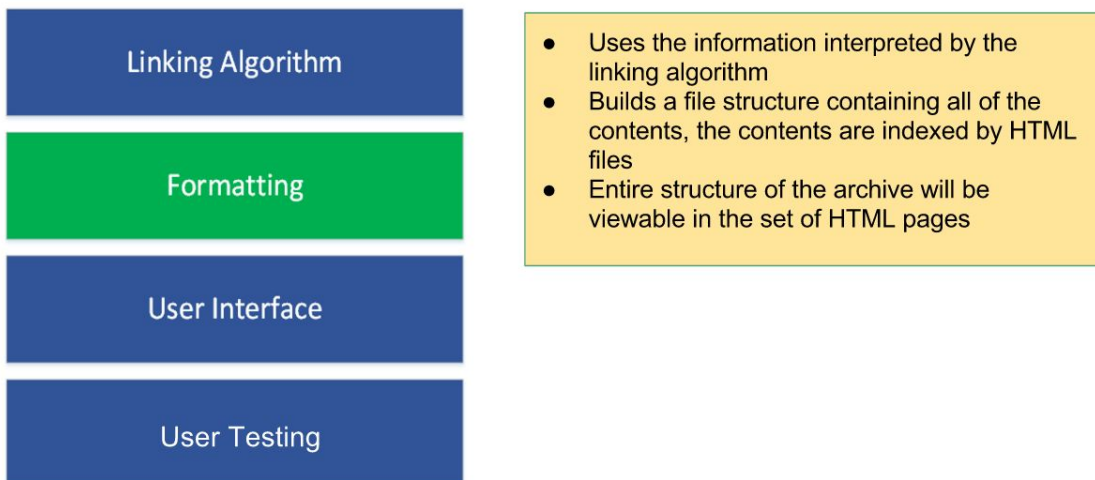
Linking Algorithm



Reformatting

Lastly, the archived data is organized into information that is similar to the original format of the website for instance, the professor’s blackboard webpage.

Second Iteration: Formating



Customer base

The target customer base of Blackboard Extractor will be universities that use the Blackboard system this especially includes ODU faculty and ultimately all educators using Blackboard.

ODU Faculty

ODU professors that use the Blackboard system will greatly benefit from Blackboard Extractor because it will provide automated process and more efficiency in storing their files for later review.

All Educators

Ultimately we want Blackboard Extractor to be the go to application for extracting Blackboard files for review and be used by all educators who use blackboard.

Blackboard Extractor Prototype Description

The Blackboard Extractor will implement all processes required for the final product. All archive data files such as, homework assignments, journal posts, quizzes, and test will be stimulated. The distribution of the resulting application will be limited to the testing and development team.

Prototype Goals

The goal of the prototype is to test the functionality and efficiency of extracting archive blackboard data files. The module tests will validate the correctness of the product algorithms for converting archive data files into fully functional webpages.

Fully functioning application

The application will be able to load a Blackboard Course Archive and explore the contents of the archive. This application will be able to handle all types of Blackboard content including blog posts, module items, and pdfs. The application will create a directory containing all of the extracted and processed contents from the Blackboard Course Archive.

Prototype Challenges

Some problems that may arise might be:

- Customer dislikes user experience and user interface.
- ODU stops using Blackboard within two years.
- Security permission issues.

- Blackboard abandons IMS Course Cartridge Archive Format within two years.
- Blackboard modifies archive formatting within two years.

Glossary

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.

References

1. 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>
2. ODU Faculty Handbook. (2005, December 21). Retrieved October 26, 2016, from <http://ww2.odu.edu/ao/facultyhandbook/index.php?page=ch07s01.html>
3. BFree: Extract Blackboard content. (n.d.). Retrieved October 19, 2016, from <https://its2.unc.edu/tl/tli/bFree/index.html>
4. Blackboard | Reimagine Education | Education Technology. (n.d.). Retrieved October 19, 2016, from <http://www.Blackboard.com/>
5. ".NET - Powerful Open Source Cross Platform Development." .NET - Powerful Open Source Cross Platform Development. Microsoft, n.d. Web. 15 Dec. 2016.