Lab 1 – Blackboard Archive Extractor

Devin R. Haslam

CS 411

Professor Thomas Kennedy

6 February 2017

Table of Contents

List of Figures

1. Introduction

Blackboard Archive Extractor is an application designed to automatically extract and reformat material uploaded from a course archive. This application will automate a manual task which is time consuming and labor intensive. The customers of this product, teachers using Blackboard, will now be able to access old content with less effort than ever before.

1.1 Blackboard in Leaning Institutions

Blackboard is an online tool used by both elementary schools and universities in order to organize course materials for a class. Blackboard allows both professors and students to access syllabi and assignments over the internet instantly. In fact, during the year 2014, around 17,000 schools were using Blackboard. Due to the large amount of customers, Blackboard has continued to update and improve its tools and services. However, there is one aspect of Blackboard that gives difficulties to professors. After 2 years, the content of a course is exported to an archive which is very difficult to work with. The software will solve this problem and help professors have a better experience with providing materials to students.

1.2 ODU Faculty Review Process

Similar to other learning institutions, faculty at Old Dominion University is reviewed periodically to ensure that they are providing a proper education to students. According to the Faculty handbook, this review process will take place every year for lecturers, every three years for senior lecturers, or five years for tenured lecturers. During this performance review, professors are required to present all aspects of a course, including materials given to the class on Blackboard. Due to the ODU policy that Blackboard only stores courses for two years before exporting them to an archive, teachers will be faced with the problem of extracting material from an archived course. Currently, if a professor wants to manually extract an archive, their first step is to upload the archive into a course they are not using on Blackboard. Without an unused Blackboard course, extraction is impossible. Assuming that a professor can find this unused course, the next step is to download each individual file from Blackboard and organize it in a presentable way for the review committee. In contrast, the Blackboard Archive Extractor will not need to use any Blackboard courses solving this dilemma. The solution will be able to extract multiple archives which will expedite the current manual process.

2. Blackboard Extractor Product Description

The Blackboard Archive Extractor will consist of a User Interface that will provide functionality to automatically extract a blackboard archive. This functionality can be further broken down into algorithms that perform Linking, and Reformatting. Although the target customer base is all teachers using Blackboard, the team will focus most effort to satisfy ODU teachers.

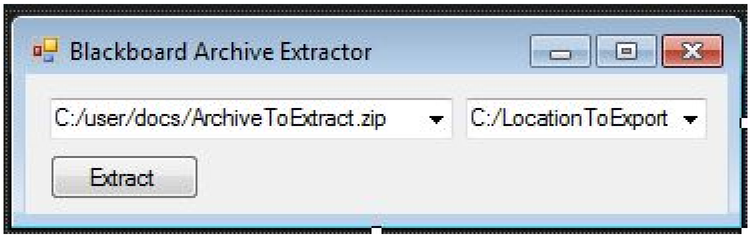
2.1 Product Features

The Blackboard Archive Extractor does not have many features to discuss. Arguably the most notable and subjective feature of the Blackboard Archive Extractor is the User interface. This feature will be focused on greatly because it will affect the perception of the tool. A second feature that we hope to design well is the formatting of the output. We would like to provide output that is easy for the user to access and understand. Lastly, the algorithms of the

Blackboard Extractor are the most clear cut piece of the work. These algorithms will consist of two phases of parsing and linking.

2.1.1 User Interface

The user interface for the program should be easy to use while also looking professional. Considering that the Blackboard Extractor only takes two parameters as input, the user interface should be simple and easy to create. In addition to a browsing feature, we would like the UI to include drag and drop functionality. A teacher should be able to drag an archive to the user interface and the program should execute the same as if the file has been selected using “browse.” Although we will update the User Interface, we have provided an early version of the tool below.



C:/user/docs/ArchiveToExtract.zip

C:/LocationToExport

Figure 1: User Interface

2.1.2 Algorithms and Process Flow

This project has two main algorithms: linking and formatting. The linking algorithm can be further broken down into a parsing and linking phase. The parsing phase interprets the manifest file and identifies the correct files for the linking phase. Following this phase, the linking algorithm then uses this information to build links between files that are related to each other. Both of these phases happen automatically and contribute to the goal of removing the tedious manual steps from this process. The other algorithm of reformatting creates the html files and saves them.

2.1.3 Output

The output will be placed in the directory that the user chooses. The output will consist of a folder containing all of the course content formatted in a way that promotes ease of access. The hierarchy of the files will reflect the structure of blackboard.

2.2 Major Components

The Blackboard Archive Extractor has 3 major components for functionality. The first component has been discussed previously known as the parsing phase. Next the linking phase, which we have also discussed previously, systematically links files that should be connected. Lastly the reformatting phase is executed, which is a component that reformats the html links after the linking phase has concluded.

2.2.1 MFCD

The 3 major components of the Blackboard Archive Extractor can be seen the project's MFCD. The portion of the MFCD labeled "Blackboard Extractor Interface" is a representation of the user interface.

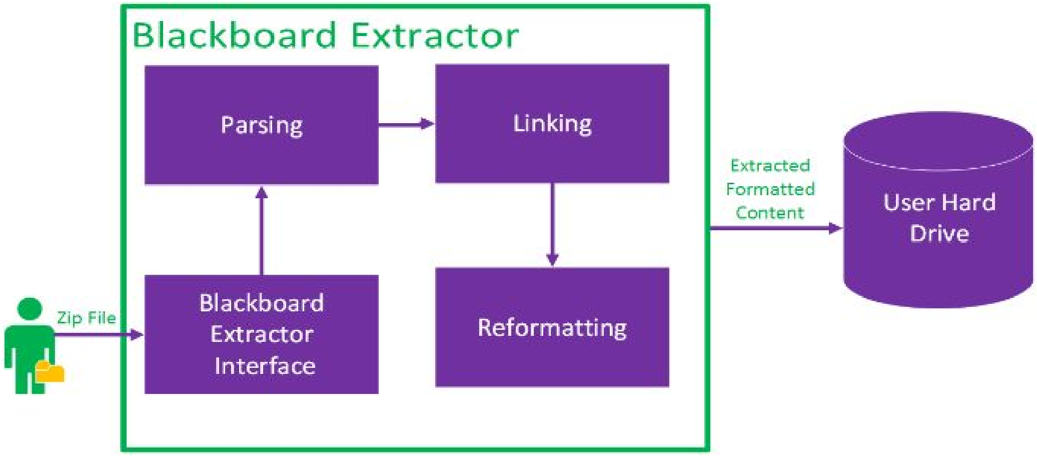


Figure 2: MFCD

2.2.2 Parsing and Linking

Parsing and linking happen simultaneously making up the linking algorithm. This algorithm first opens each manifest file for an element and finds the res.xxxx file for that element. The algorithm will then create a new node for the element, add it to the root node, and then process the resources for the current node. This process consists of classifying the material, checking for children, and then checking the html links.

2.2.3 Reformatting

Reformatting happens after the linking algorithm. When the reformatting phase is mentioned, it pertains to generating the html index files. This process first builds an index file and then saves the html resource. After reformatting, the resources are saved to the disk.

2.3 Customer Base

We are creating the Blackboard Archive Extractor in an attempt to benefit all teaching professionals using Blackboard. However, we expect the primary users to be faculty teaching at

Old Dominion University. We are attempting to automate a process that will save user's time and effort. The Blackboard Archive Extractor has the potential to solve a glaring problem for teachers using Blackboard.

3. Blackboard Extractor Prototype

Unlike other projects, the Blackboard Archive Extractor is expected to not take a large amount of time to complete. Every feature in the full product will be contained in the prototype being constructed. The team designing the Blackboard Archive Extractor is capable of satisfying all prototype goals and having a release ready for testing by ODU faculty.

3.1 Prototype Goals

There is a litany of goals that are expected to be achieved with the prototype. One example of these goals is that the prototype should be able to extract most types of content from an archive. In addition, ideally, the output would be saved on the hard disk. While these two goals are relatively obvious, there are also other aims such as being able to execute the program with a GUI or command line. Overall, the prototype goals are very similar to the goals of the fully functioning application.

3.1.1 Fully Functioning Application

There are very few goals for the fully functioning application that are not reflected in the goals for the prototype. However, one of these rare occurrences is the ability to execute the program from a web application. This goal will not be in the prototype, but would be in the final application. There are no other goals that are in the fully functioning program, but not mentioning in the prototype. However, there are cases where the fully functioning application will be more practical. A great example of this is that while both should be able to extract content from an archive, the prototype will not be able to deal with content that is not normal and not supported.

3.2 Prototype Challenges

We do expect to have challenges when developing theprototype, some challenges more severe than others. Two of the most challenging obstacles will be if Blackboard Changes the structure of archives or the University stops using Blackboard. We are wary of both of these challenges and have taken steps to mitigate the problems.

3.2.1 Blackboard Changes Structure

There is a very high chance that Blackboard changes the structure of its archives in the next two years. If this were the case, much of the program would need to be revised and rewritten. However, we believe that the impact of this happening will only be medium. We can mitigate this risk by using good software development practices and object oriented design principles. This will make updating the application much more simple if it were necessary.

3.2.2 University Stops Using Blackboard

In contrast to the other challenge, if the University were to stop using Blackboard, this would have a very high impact. The program would then be completely useless to the University. There is no way to completely mitigate this problem, but ODU has already considered this option in the recent past. We do not believe that ODU will stop using blackboard within the next two years and therefore rate the probability as medium.

4. Glossary

Blackboard: a virtual learning environment and course management system developed
by Blackboard Inc.

Blackboard Archive: A “frozen snapshot” of a course which contains a permanent record of all content.

Blackboard Archive Extractor: A program in development by team Crystal of the ODU course CS411W. This program will automate the previously tedious process of accessing a Blackboard Archive.

GUI - Graphical User Interface: a type of user interface that allows users to interact with electronic devices through graphical icons and visual indicators.

IMS Course Cartridge: allows Instructors and Administrators to export a package that can be used in any Learning Management System that supports Common Cartridge.

Learning Management System: is a software application for the administration, documentation, tracking, reporting and delivery of electronic educational technology.

Microsoft .NET: A software framework developed by Microsoft that provides capability across several programming languages.

Old Dominion University: a public, co-educational research university located in Norfolk, Virginia.

UI - User Interface: the means by which the user and a computer system interact, in particular the use of input devices and software.

UX - User Experience: the overall experience of a person using a product such as a website or computer application, especially in terms of how easy or pleasing it is to use.

5. References

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/

"Bootstrap · The World's Most Popular Mobile-first and Responsive Front-end Framework." Bootstrap · The World's Most Popular Mobile-first and Responsive Front-end Framework. Twitter Inc., n.d. Web. 04 Feb. 2017.

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

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

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