# Lab 1 – Blackboard Archive Extractor

Chris Soffos

CS 411

Professor Kennedy

2 February, 2017

## Table of Contents

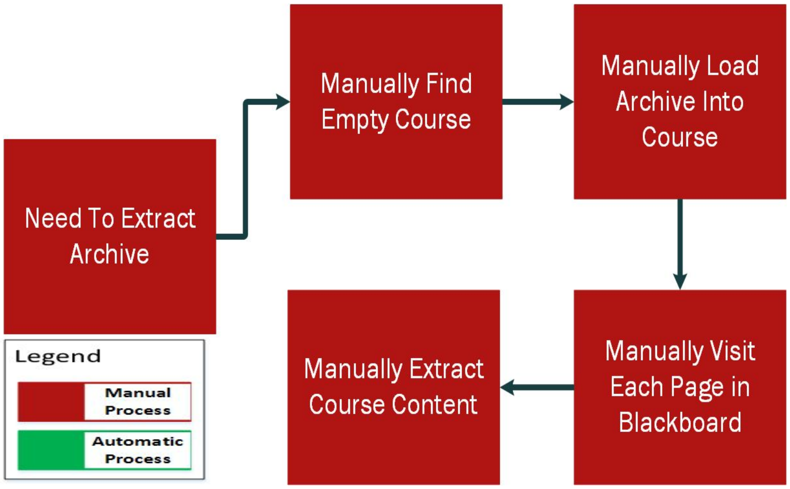
## Table of Figures

# 1 Introduction

Blackboard is a learning management solution designed to organize and deliver course content which is used by approximately 17,000 educational institutions located across over 100 nations[1], including Old Dominion University. This tool also provides a method of exporting content created within the application in a standard archive format.

At Old Dominion University, all professors periodically undergo a formal review of all course materials. According to the Old Dominion University Faculty Handbook, lecturers are reviewed every year, senior lecturers are reviewed every three years, and tenured professors are reviewed every five years[2]. This review process requires that all course material used since the last review be presented to a review board, which could mean a very large amount of material to be presented. It is worth noting that at Old Dominion University, Blackboard courses remain in a professor's course catalog for 2 years, after which time the course is expunged from the Blackboard system. This retention period necessitates the exporting of Blackboard course archives to ensure retention of all course materials until the next review.

Currently, when a professor needs to collect and display his or her course materials for review, the professor must begin the tedious, manual, and unreliable process represented by Figure 1. Beginning with an exported Blackboard archive of a course, the professor must hope that, of the courses he or she has taught over the past two years (ODU's retention period), there is an unused one into which to upload the content of a course scheduled for review. If the professor is able to locate such a course, the process of manually re-created the previous course's folder structure and manually visiting and uploading each page of the previous course can begin. This arduous process must be repeated for each course which is scheduled to be reviewed.


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

*TC "Figure 1: Current manual process flow" \f y \l 1 Figure SEQ Figure * ARABIC 1: Current manual process flow by which a professor presents a course for review.*

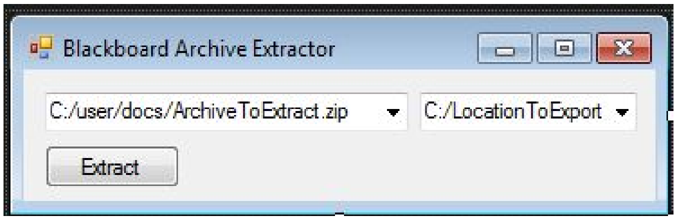
Blackboard Archive Extractor is a desktop application designed to simplify and automate the process of extracting Blackboard course archive content. Blackboard Archive Extractor will process an input Blackboard archive and index and output the contents in the form of a simple and readable website.

# 2 Blackboard Archive Extractor Product Description

The Blackboard Archive Extractor tool will be a desktop application with which a user may interact via the command line or the graphical user interface (GUI). The application's internal structure can be broken down into the functions of extraction, parsing, linking, and formatting, all which will be discussed later. The final output of the tool will be a structured and indexed website, created based on the input Blackboard archive.

## 2.1 Key Product Features and Capabilities

The Blackboard Archive Extractor tool will have two methods of interaction: command line interaction or GUI. The command line options will simply allow the specification of an input archive file and the output folder where the tool should store the contents of the generated website.



Blackboard Archive Extractor

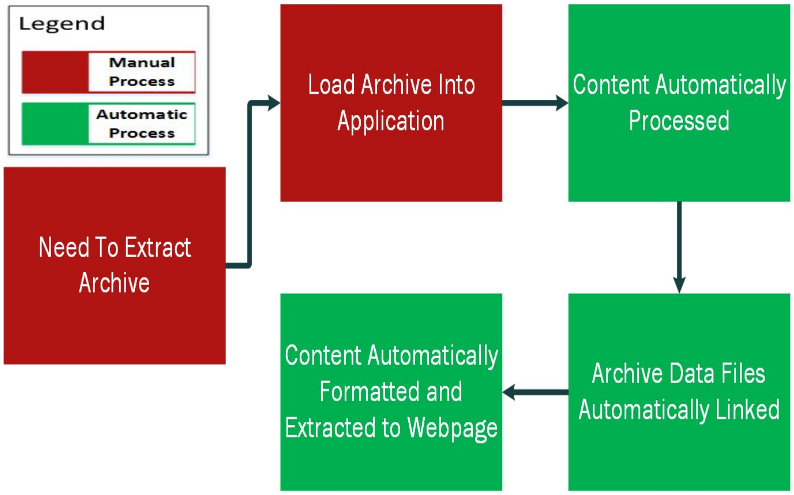
C:/user/docs/ArchiveToExtract.zip

C:/LocationToExport

TC "Figure 2: Blackboard Archive Extractor GUI" \f y \l 1 Figure 2: Blackboard Archive Extractor graphical user interface (GUI)

The GUI, as seen in Figure 2, will consist of: a file selection menu, with which the user will specify a blackboard archive, a folder selection menu, with which the user will specify an output location, and a button which will initiate the process.

The primary focus of Blackboard Archive Extractor is to automate the process of extracting a Blackboard course archive and presenting it in a readable format. This process is broken down into four main functions: extraction, parsing, linking, and re-formatting. These functions will be explained in further detail later in the paper. Once the Blackboard Archive Extractor tool has been incorporated in the professor's process of preparing course content for review, the process becomes much more automated and hands-off, as displayed in Figure 3.



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

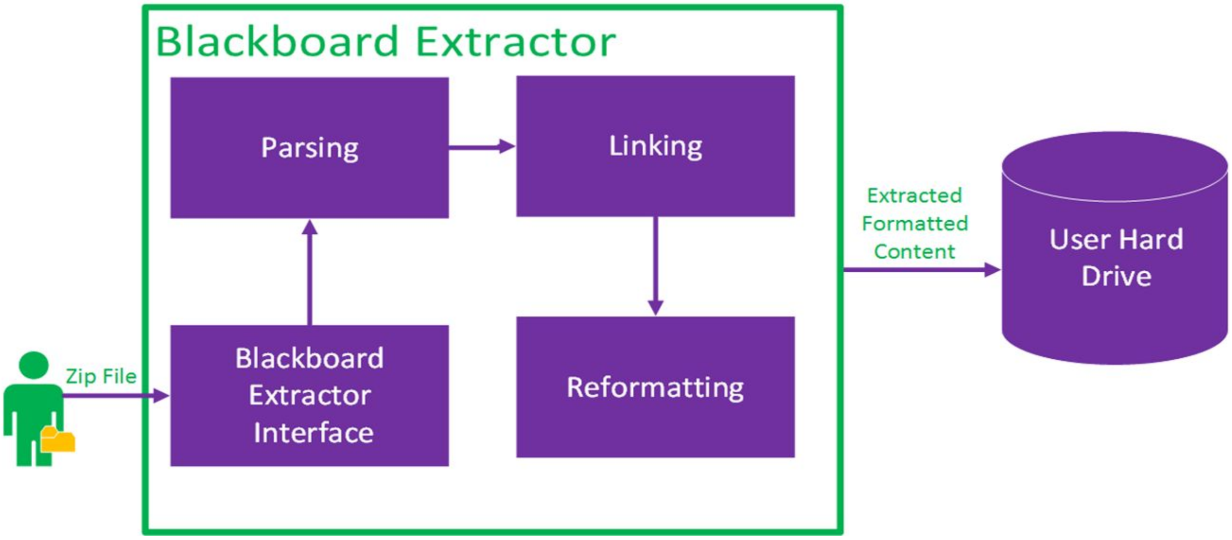
TC "Figure 3: Revised automated process flow" \f y \l 1 Figure 3: Revised process flow after including Blackboard Archive Extractor in the process.

As can be observed in the figure, the only manual parts of the process are exporting the Blackboard course archive and loading the archive into the application. The rest of the process, which when manually performed could take hours, is now handled automatically by the tool. This significantly reduces time wasted on trivial, repetitive actions; time which could be better put to use on another project, or which could be put toward some well-earned leisure time.

The output of the program will be the contents of the input course archive, arranged in an easily-navigable folder structure, an index of the archive's contents, and a website to deliver the contents in an approachable format. The index will be a structured list of resources found within the archive, organized by hierarchy, and will contain links to the various resources referenced. The website be a simple HTML5 site, consisting of a series of linked pages which will display the contents with a high level of fidelity to the formatting and structure with which the Blackboard course was exported.

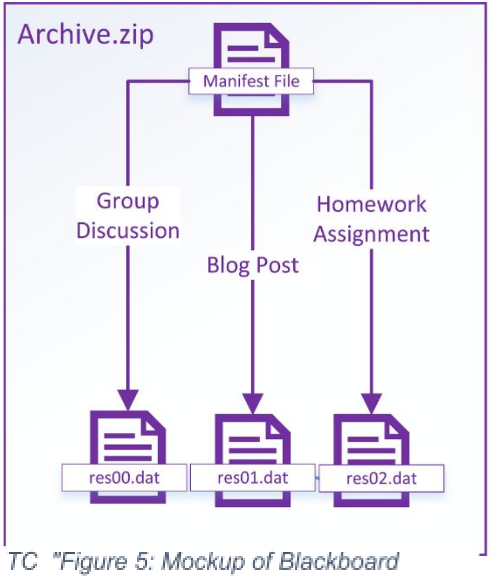
## 2.2 Major Components

The Blackboard Archive Extractor tool will consist of four major components. The interface, with which the user will interact; the parsing algorithm, which will extract and parse the archive, relaying information about the files found within to the linking algorithm; the linking algorithm which will build a parent-child hierarchy of resources found within; and the formatting algorithm, which will generate and output the index and website for the provided Blackboard course archive.



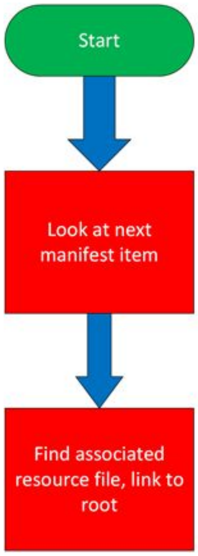
*TC "Figure 4: Major Component Functional Diagram" \f y \l 1 Figure 4: Major Functional Component Diagram*

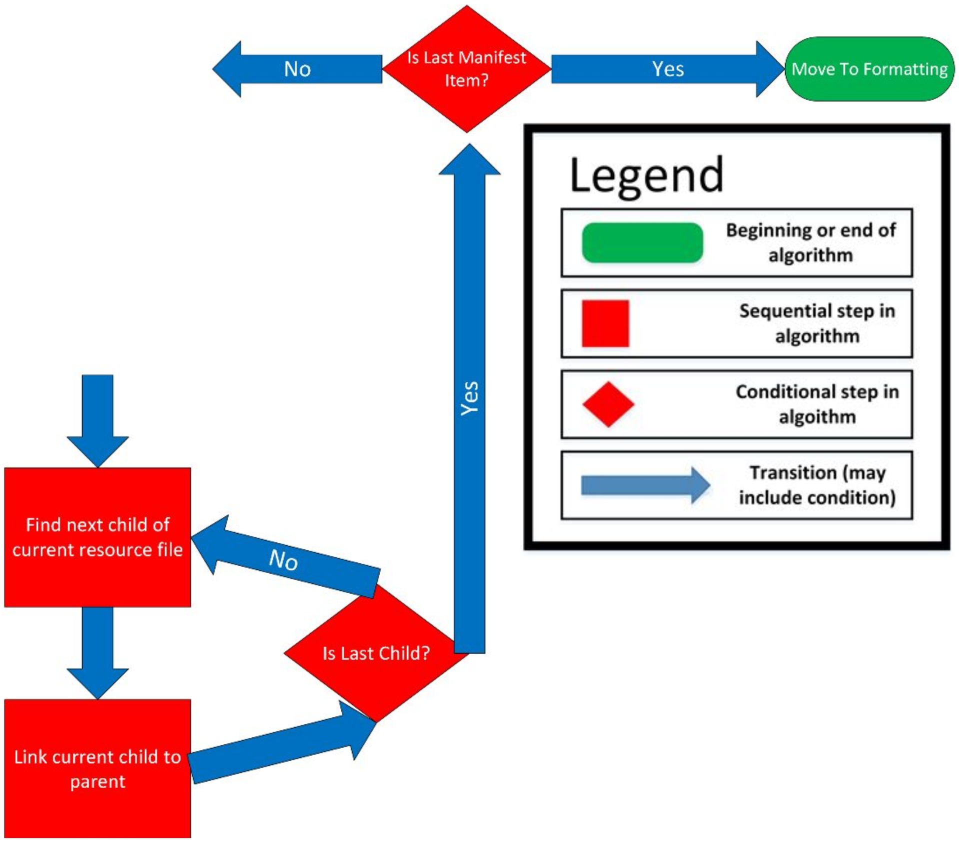
Because the user interface for the tool is minimalistic and was discussed earlier, discussion of it will be omitted here. Upon initiating the process for extracting a Blackboard course archive, the file is sent to the parsing algorithm for initial processing. It is within this section of the program that the archive is extracted. Within the archive, a manifest file detailing the contents can be found (Figure 5). This manifest contains entries describing the resources included within the archive, including the filepath and content type. Contents of the archive are generically named, necessitating the use of the file for piecing together an accurate representation of the original exported Blackboard course.



*TC "Figure 5: Mockup of Blackboard archive structure" \f y \l 1 Figure 5: Example of a manifest file found within a Blackboard course archive.*

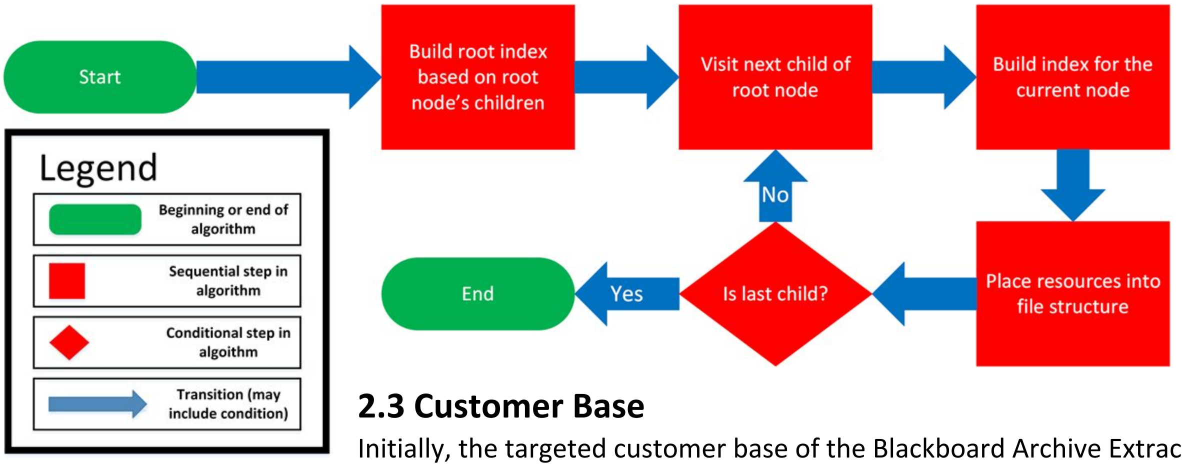
Using information relayed by the parsing algorithm, the linking algorithm will begin to build a hierarchy of files contained within the archive, generating parent-child relationships among related files. The interaction for this process can be seen in Figure 6. This hierarrarchy will consist of objects representative of individual resources referenced by the archive's manifest file. This parent-child structure will facilitate the formatting algorithm's generation of a website which appropriately represents the exported Blackboard course.


Start
Look at next manifest item
Find associated resource file, link to root


No
Is Last Manifest Item?
Yes
Move To Formatting
Legend
🟩 Beginning or end of algorithm
🟥 Sequential step in algorithm
◆ Conditional step in algorithm
→ Transition (may include condition)
Yes
Find next child of current resource file
No
Is Last Child?
Link current child to parent

*TC "Figure 6: Parsing and Linking algorithms" \f y \l 1 Figure 6: Parsing and Linking algorithms and their interaction*

Upon completion of the linking algorithm, the reformatting algorithm will begin to rapidly generate a user-friendly and readable representation of data collected. This representation comes in the form of an index of all resources and a website, which will be output to the user's hard drive. The algorithm for this process is outlined in Figure 7.


Start
Build root index based on root node's children
Visit next child of root node
Build index for the current node
Place resources into file structure
Is last child?
No
Yes
End
Legend
🟩 Beginning or end of algorithm
🟥 Sequential step in algorithm
◆ Conditional step in algorithm
→ Transition (may include condition)

## 2.3 Customer Base

Initially, the targeted customer base of the Blackboard Archive Extractor tool will be faculty of Old Dominion University that require a method of displaying the contents of a Blackboard course archive. This decision was made based upon proximity to the users and the necessity of this tool arising fro m a policy local to Old Dominion University's faculty and its implementation of

Blackboard. Once appropriate testing has been performed, and the applic ation has reached its final iteration, the tool will be released to the general public. At which point, the customer base for the tool will become all educators employed by a University which utilizes the Blackboard LMS to deliver course content to students.

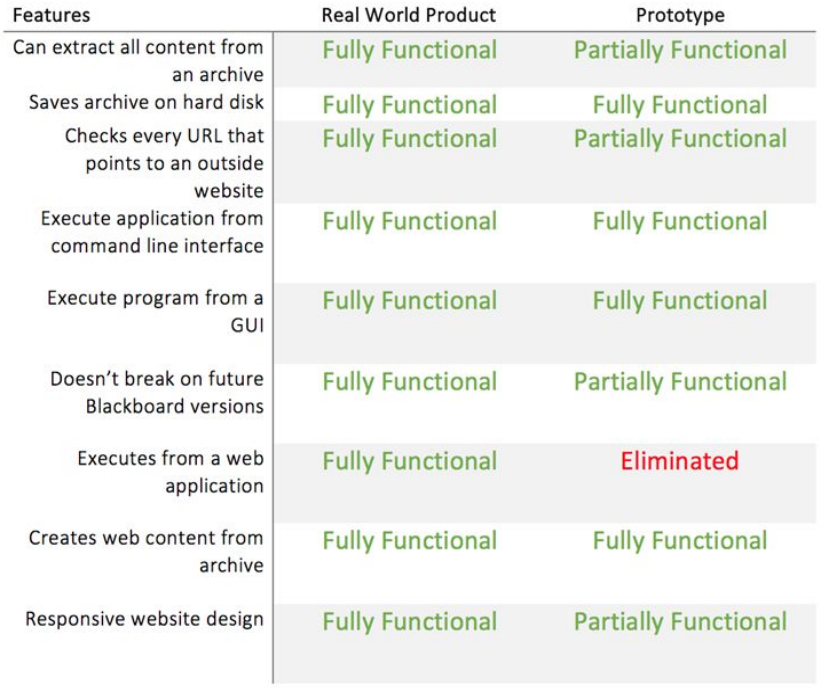
## 3 Blackboard Archive Extractor Prototype Description

The Blackboard Archive Extractor Prototype will be a fully-functional version of the program, with full implementation of all algorithms required in the final product. The distribution of the prototype application will be limited to professors employed by Old Dominion University, and will allow for the extraction of and output of the contents of Blackboard course archives.

## 3.1 Prototype Functional Goals and Capabilities

During the prototyping phase, Team Crystal plans to maintain a high level of fidelity to the originally pitched product, and will deliver a fully-functioning version of the Blackboard Archive Extractor tool. However, in order to rapidly prototype the product, the team has opted to forgo the ability to execute the application from a web browser, which would add unnecessary complication to the proof of concept. Our goals for this prototype are outlined in Figure 8.

This means the tool will be able to perform the process of extracting, parsing, linking, and formatting the contents of any provided Blackboard course archive from beginning to end with no interruption or manual intervention required on the part of the user. The tool will also be modular and be able to check the state of each link referenced in the archive.



| Features | Real World Product | Prototype |
|---|---|---|
| Can extract all content from an archive | Fully Functional | Partially Functional |
| Saves archive on hard disk | Fully Functional | Fully Functional |
| Checks every URL that points to an outside website | Fully Functional | Partially Functional |
| Execute application from command line interface | Fully Functional | Fully Functional |
| Execute program from a GUI | Fully Functional | Fully Functional |
| Doesn't break on future Blackboard versions | Fully Functional | Partially Functional |
| Executes from a web application | Fully Functional | Eliminated |
| Creates web content from archive | Fully Functional | Fully Functional |
| Responsive website design | Fully Functional | Partially Functional |

*TC "Figure 8: Prototype feature chart " \f y \l 1 Figure 8: Comparison of functionality between Real World Product and Prototype of Blackboard Archive Extractor.*

## 3.2 Prototype Challenges

During development of the prototype application, two key challenges must be mitigated. The first challenge is that of Blackboard changing the structure of its archive format. Currently, Blackboard organizes its files in such a manner as outlined in Figure 5, previously referenced. Blackboard exports the contents of its courses in a strictly organized structure, including a "manifest" file which contains information on the various resource files within the archive. If Blackboard were to alter their archive format, this could potentially impact the Blackboard Archive Extractor tool's ability to appropriately handle these course archives and their content. This risk is mitigated by developing a modular, open-source application, allowing for the later modification of the data model used by the program.

The second challenge to the efficacy of the prototype is the appeal of the user interface generated for the website output by the tool. This step is easily mitigated by decoupling the structure of the pages from the formatting. By isolating formatting code to a shared CSS file, the risk is mitigated by allowing the user to modify the formatting of the site to a form that is pleasing to the user based upon his or her preferences.

## 4 Glossary

**GUI:** Graphical User Interface, or a method of visual interaction with a program

**UI:** User Interface, or the method of interaction with a program, whether by text or graphical buttons

**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

**Algorithm:** A process or set of rules to be followed in calculations or other problem-solving operations, especially by a computer

**Microsoft .NET:** A semi-open source, cross-platform, compiled programming framework used to develop executable applications

## 5 References

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.