

CS 411W Lab II

Prototype Product Specification

For

Blackboard Archive Extractor

Prepared by: Chris Soffos, Team Crystal

Date: 03/01/17

Table of Contents

1 Introduction	3
1.1 Purpose	3
1.2 Scope	3
1.3 Definitions, Acronyms, and Abbreviations.....	5
1.4 References	5
1.5 Overview	5
2 General Description	6
2.1 Prototype Architecture Description.....	6
2.2 Prototype Functional Description	6

Table of Figures

Figure 1: Current manual process flow.....	3
Figure 2: Revised automated process flow	4
Figure 3: Blackboard Archive Extractor GUI.....	6
Figure 4: Blackboard Archive Extractor command-line UI.....	6
Figure 5: Blackboard Archive Extractor major functional component diagram	7
Figure 6: Mockup of Blackboard archive structure.....	7
Figure 7: Parsing and Linking algorithms	7
Figure 8: Formatting algorithm	7

1 Introduction

Blackboard Archive Extractor is a tool being developed as an exit project by Team Crystal of Old Dominion University, which seeks to fill a niche which will mitigate a pain point of any professor that uses Old Dominion University’s implementation of Blackboard.

1.1 Purpose

The purpose of Blackboard Archive Extractor is to simplify and automate the process of arranging the contents of a Blackboard course archive for display to a review board. The intent of this product is to provide professors a method by which they may rapidly generate neatly-organized course websites which may then be presented to a course review board. The product will offer all features of previous competitors, such as the deprecated product *bFree*, and improve upon their shortcomings.

1.2 Scope

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 years². This review process requires that all course material used since the last review be presented to a review board, which could mean a non-trivial amount of material to be presented. 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.

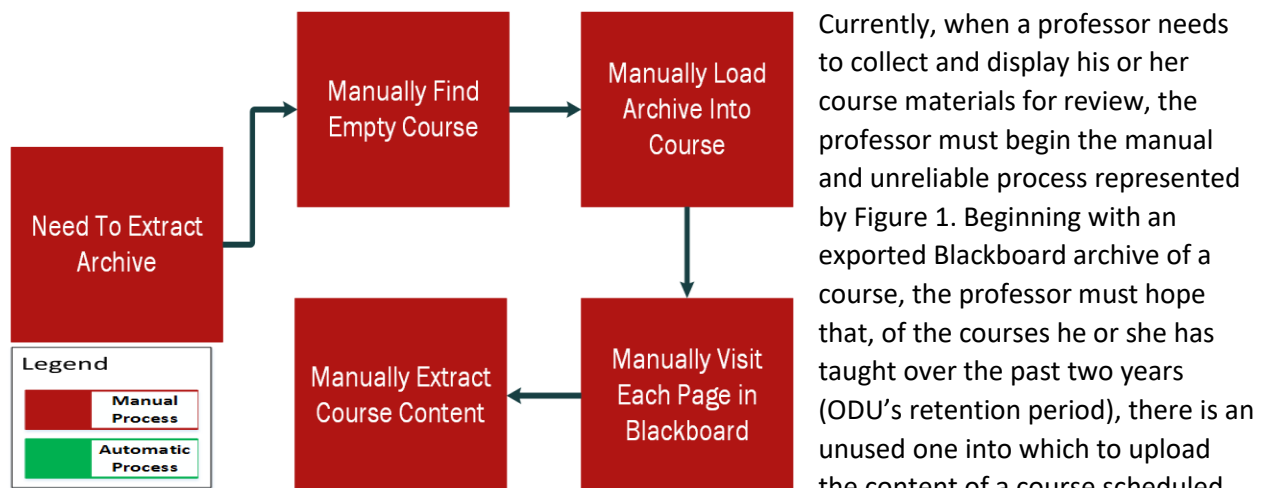


Figure 1: Current manual process flow by which a professor presents a course for review.

Currently, when a professor needs to collect and display his or her course materials for review, the professor must begin the 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 process must be repeated for each course which is scheduled to be reviewed.

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.

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 in Section 2.2 of this paper. Once the Blackboard Archive Extractor tool has been incorporated in the professor's process of preparing course content for review, the process becomes more automated, as displayed in Figure 3.

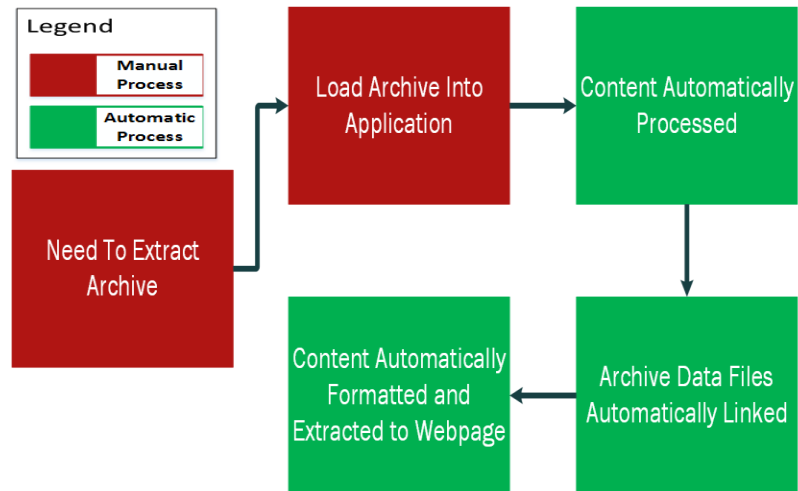


Figure 2: Revised process flow after including Blackboard Archive Extractor in the process.

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 will 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.

In the development of Blackboard Archive Extractor, Team Crystal seeks to simplify and automate a needlessly complex process. In the current workflow of presenting Blackboard course archives, a professor may spend countless hours manually organizing course content for display. Blackboard Archive Extractor will reduce these hours to minutes by automating the tedious and monotonous portions of the workflow.

The prototype which will be used to demonstrate the functionality and usefulness of the product will be able to receive an input Blackboard course archive, process it, and output the archive's contents in the form of an organized website, with an index linking the contents together.

1.3 Definitions, Acronyms, and Abbreviations

Graphical User Interface (GUI): A method of visual interaction with a program

User Interface (UI): The method of interaction with a program, whether by text or graphical buttons

User Experience (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

1.4 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.

CS 411W Lab I – Blackboard Archive Extractor Descriptive Paper

1.5 Overview

This product specification will outline the software requirements, capabilities, and features of the Blackboard Archive Extractor prototype. The following information will include a detailed description of the software and external interface architecture of the Blackboard Archive Extractor prototype; the key features of the prototype; the parameters that will be used to control the prototype; and the performance characteristics of the prototype in terms of inputs, outputs, and user interaction.

2 General Description

Presented in this major section is a general description and outline of the features of Blackboard Archive Extractor which will be present in the released prototype.

2.1 Prototype Architecture Description

Blackboard Archive Extractor is comprised of the following major architectural components:

- A graphical user interface allowing the ability to provide an input Blackboard archive and a destination for the output course website. The interface will also provide a button to begin the execution of the product's core functionality.

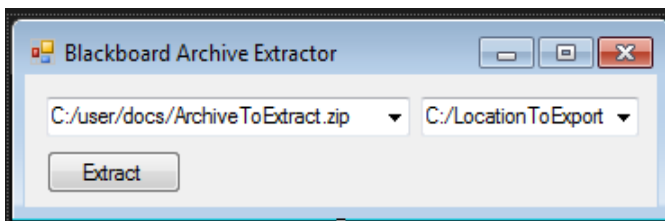


Figure 3: Blackboard Archive Extractor GUI

- A textual, command-line interface allowing the ability to provide an input Blackboard archive and a destination for the output course website.

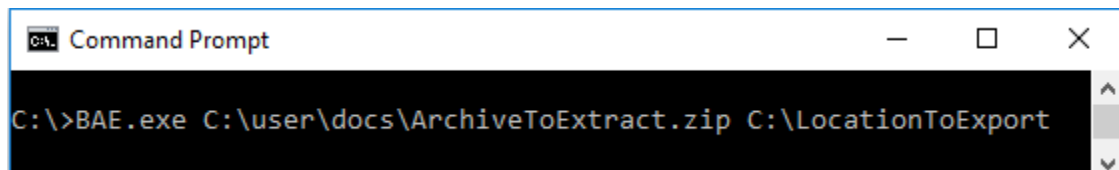


Figure 4: Blackboard Archive Extractor command-line UI

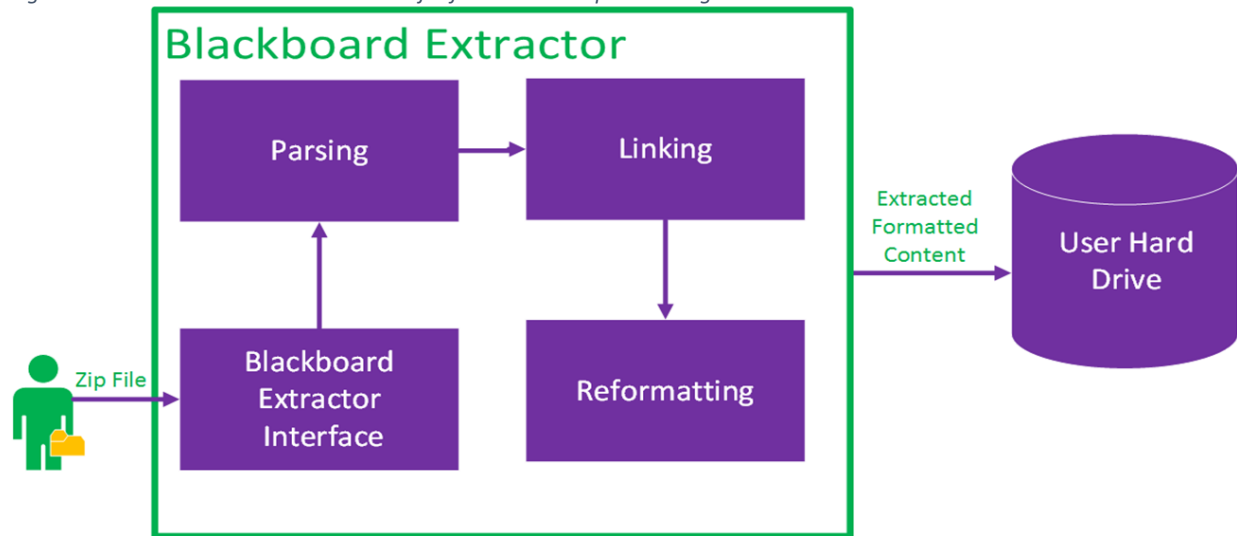
2.2 Prototype Functional Description

Blackboard Archive Extractor is comprised of the following major components, which are displayed graphically in Figure 5:

- User Interface: provides the methods by which the product may be used, including both a textual command-line interface and a graphical window-based interface.
- Parsing Algorithm: The method by which the product will interpret and process the provided Blackboard course archive.
- Linking Algorithm: The method by which the product will form associations between files contained within the Blackboard course archive.

- Reformatting Algorithm: The method by which the product will output the contents of the Blackboard course archive in the form of an indexed website.

Figure 5: Blackboard Archive Extractor major functional component diagram



Because the user interface for the tool is minimalistic and was discussed earlier in Section 2.1, 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.

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 hierarchy will consist of objects representative of individual resources referenced by the archive’s manifest file. This parent-child structure will facilitate the

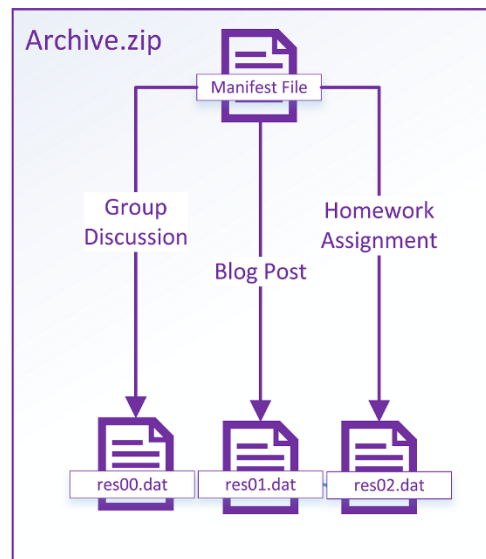


Figure 6: Example of a manifest file found within a Blackboard course archive.

formatting algorithm’s generation of a website which appropriately represents the exported Blackboard course.

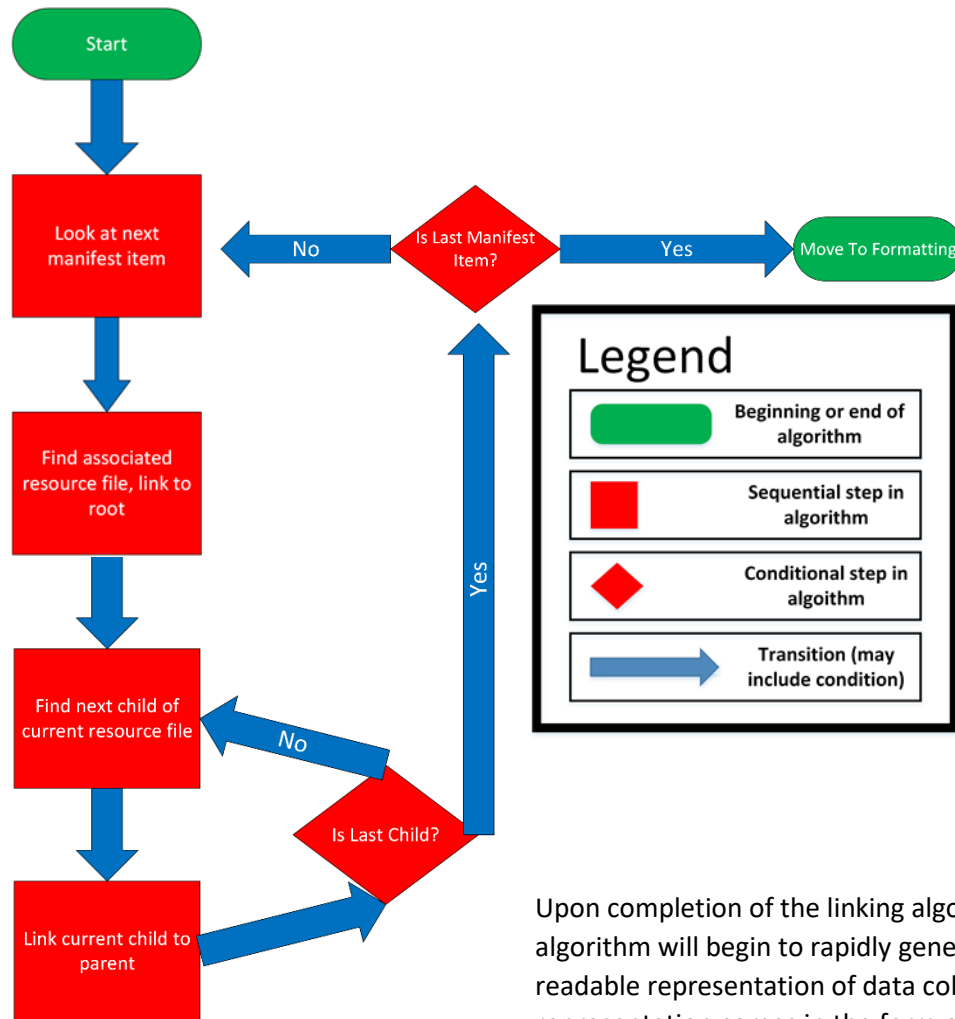


Figure 7: Parsing and Linking algorithms and their interaction

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.

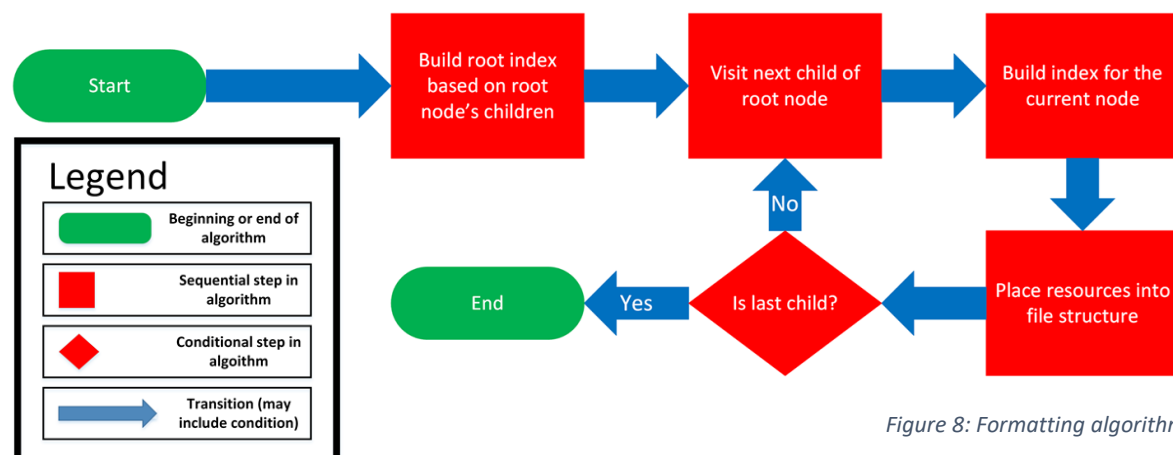


Figure 8: Formatting algorithm