TEXTBOOK PROBLEM DEPENDENCY WEB

An Undergraduate Research Scholars Thesis

by

JOSEPH MARTINSEN

Submitted to the Undergraduate Research Scholars program at
Texas A&M University
in partial fulfillment of the requirements for the designation as an

UNDERGRADUATE RESEARCH SCHOLAR

Approved by Research Advisor: Dr. Philip B. Yasskin

May 2019

Major: Computer Engineering

TABLE OF CONTENTS

ABSTRACT

Textbook Problem Dependency Web

Joseph Martinsen
Department of Computer Engineering
Texas A&M University

Research Advisor: Dr. Philip B. Yasskin
Department of Mathematics
Texas A&M University

After a textbook has been written and published, one may want to customize it for a particular audience; it may be desirable to delete or reorder some of the chapters or sections. However, there may be dependencies among the chapters, sections, examples and exercises which make it very tedious to rearrange the order of not only the chapters but also, the associated exercises. Martinsen is building a graph database to describe the dependencies among the chapters and sections in a portion of the online Calculus book, MYMathApps Calculus, being written by Yasskin. In addition there will be a database of examples and exercises which describes their dependencies on the chapters and sections. Further, he is building a GUI for an instructor or institution to reorder the chapters and sections by drag and drop and have the examples and exercises automatically reorder.

DEDICATION

To. . .

ACKNOWLEDGMENTS

To...

NOMENCLATURE

CONSUMER	One who uses the textbook

1. INTRODUCTION AND LITERATURE REVIEW

1.1 Background Info

Textbooks go through a long and arduous process before a student or professor is able to view and use it. This process requires much work and effort into not only validating the content of the textbook but also validating the ordering of the textbook as whole. Much like a jigsaw puzzle, each chapter fits one after another based on the dependency of the topics being taught. On top of these chapters being ordered, the exercises must also be placed in the correct place in order to not give an exercise that is based on a topic that has not been presented to the reader previously.

After all this work on ordering has been completed (among other meticulous things), the textbook may finally ready for publication. As with many good textbooks, many professors and institutions may enjoy the content within the textbook but would prefer delivering the content to a student in a different order than what the current order the textbook is in currently. Most of the time, the work and effort that has gone into meticulously ordering the chapters and exercises must now be revisited again and modified. This process is nearly as time consuming and laborious as the first going through this process.

1.2 Existing Technology

In the current field, most re-orderings of chapters and exercises are done by hand with human intervention by either textbook authors or commercial institutions. An exception is TopHat (TM) who, for their interactive textbooks, has the ability to reorder in a drag and drop manner but one lacking feature is any sort of feedback of any sort being provided to the user. I was able to test drive this product and it felt much like a changing the order of a power point presentation as it is in its current state. Besides the technology features that

TopHat offers on the rest of their platform, which is outside the scope of this discussion, much of what their reordering seems to achieve can simply be done by reordering the pages of the textbook in some sort of PDF viewer.

From speaking with several Math professors at the Joint Math Meeting 2019, this idea of being able to reorder structural components of their textbook is an issue they commonly face and is not brand new. From each professor I interacted with, there were not aware of a simply and useful that would help them complete this task. There was one platform for identifying dependencies between subjects and textbooks that then allows a professor to generate a curriculum based off these topics, but this is done on a much higher textbook level, not by chapter, section or exercise.

1.3 Approach

My approach to this situation wants to take on a similar approach to what TopHat has done but builds heavily on the fact that the original author has an idea of the flow of a textbook, including changes to the order the repercussions that may result. While it is not feasible for a professor to provide this feedback in person, the realtime feedback aspect can be preserved by creating a platform that *understands* the original authors concerns and worries and then provide them to the consumer of the textbook the moment a change they make may result in possible conflict of dependencies.

2. THE ALGORITHM

2.1 Introduction

The core portion of the platform that hopes to allow reordering of aspects of the textbook can simply summed up as *The Algorithm*. This underlying algorithm is independent of technology stack or implementation. It is simply a high level description and analysis of the proposed solution to the given problem.

With simply an understanding of the algorithm, the implementation should follow in a natural and simple manner.

2.2 The Motivation for the Algorithm

The algorithm has other items to consider outside simply just solving the given issue. As with most algorithms, there is a thought and focus on completing the desired task in an efficient and optimal manner. This means that if the algorithm is able to provide feedback but if it is done in an clunky manner, the algorithm is still considered a failure.

In addition to efficiency, correctness is another major point of focus. In this context, correctness relates to properly conveying the thoughts and concerns of the original author to the professor or institution trying to modify the ordering of a textbook. As a result of a particular action or modification by the consumer, there should never arise a situation where a suggestion or warning provided by the algorithm conflicts with the thoughts or viewpoints of the author of the textbook. These messages provided by the algorithm should be simply an extension of the original if not exactly the same as if the author was at the computer sitting with the user of the platform.

Correctness also relates to truly reordering the textbook as desired and specified by the one who modified the order of the textbook. After a change by the consumer, they should

be able to clearly understand what type of change they are proposing, how this affects the textbook as a whole, how nearby sections or chapters may be affected and finally after committing these changes, the actual textbook should properly reflect these changes as desired by the consumer.

traverse old and new textbooks diff algorithm between two trees

traverse old and new textbooks diff algorithm between two trees

[1] [2] [3]

2.3 The Design

The design includes...

tree of unit (chapters/sections/pages) order – 1a orig and 1b modified

tree of exercise order – 2a orig and 2b modified

tree of unit interdependences 3

tree of exercises dependencies on units 4

flexiblility

1. author creates original trees 1a, 2a, 3, 4

2. adopting institution or instructor specifies desired order of units 1b

program provides warnings of impermissible orders as this tree is created

adopting institution or instructor can override order

3. program automatically creates 2b

adopting institution or instructor previews exercise and can override order

4. actually reorder the files of the text and links between them

3. THE IMPLEMENTATION

3.1 The Textbook

For the implementation of the design, I have gone with proof of concept model. The textbook that will first be integrated with my adviser's Mathematics textbook.

3.2 Textbook Build Process

Talk about the JSON hierarchy that the textbook depends on

3.3 The Tech Stack

Give basic high level overview of the application on how a user interacts with it

3.4 Integration with Textbook

How the json is imported into application then exported for use by the textbook

4. CONCLUSION

4.1 Results

Images of the application and textbook?

4.2 Challenges

Time? Tweaking existing build process of textbook for integration.

4.3 Broader Impact

Allow other authors to utilize this design + implementation?

4.4 Future Plans

Get this functional for entire MYMACalc book

REFERENCES

[1] P. Bille, "A survey on tree edit distance and related problems," *Theoretical Computer Science*, vol. 337, no. 1-3, pp. 217–239, 2005.

[2] D. Tsur, "Faster algorithms for guided tree edit distance," *Information Processing Letters*, vol. 108, no. 4, pp. 251–254, 2008.

[3] B. Vaughn, "Reconciliation." Web, 2019.