The 32nd Annual Conference of the Japanese Society for Artificial Intelligence, 2018

Connecting Learning Footprints Across Versions within E-Book Reader

Christopher C.Y. Yang [*1] Brendan Flanagan [*2] Hiroaki Ogata [*2]

[*1]Graduate School of Informatics, Kyoto University [*2]Academic Center for Computing and Media Studies, Kyoto University

Abstract: This paper deals with the connection of annotations, or we call learning footprints such as marker, memo and bookmark across versions within an e-book reader, we propose the issues with e-book reading and data analysis after new version learning material been uploaded to the e-book reader. We first briefly give an introduction to demonstrate version control system and e-book reader, then we propose a process to combine the core concepts of version control such like Diff and Merge with an e-book reader, we also developed a Transformation model to connect all the learning footprints across versions made by e-book users. In the final, we give the implementation results with our own test data to explain about how this process can fix the proposed issue, and also conclusions and future works to this research.

1. Introduction

Traditional textbook usually not allow teachers or instructors to update their learning materials, so the versions of learning materials is hard to be distributed. But nowadays, e-book reader can overcome this problem, allowing teachers or instructors to update their learning material freely, and also users can make their annotations in the interface of a common e-book reader.

Despite e-book reader allows users to freely create their learning footprints like marker, memo or bookmark, and also allows teachers or instructors to update their learning materials at any time, there is still a problem exist within e-book reader. In a coursework, when teachers update their learning material, there is a new version of learning material will show up, we may face a problem on the modification for all the learning footprints left by users in the interface of e-book reader and also the log in database, thus we will demonstrate the issues in detail in the following section. Due to the proposed issues above, the primary objective of this paper is to connect all the learning footprints across versions within BookRoll, which is an e-book reader and be able to offer many kinds of interaction between users and system, include taking note or creating memo. Since the proposed BookRoll e-book reader allows teachers to update their learning materials frequently for the new information, therefore, several versions can be created in one learning material. Besides, the proposed BookRoll e-book reader records user's activities, but the logs can only be linked with one version of one learning material. According to the issues above, a possible solution is the combination of version control system and e-book reader, so this paper proposes a process to combine the core concepts of version control system which has become a critical part for taking control of the release history on various of learning fields, and meanwhile applied on software development and pedagogy in many researches, and also an e-book reader BookRoll to connect all the learning footprints across versions left by e-book users, there are some vocabularies and background need to be clarified in the first place, so we first give the introduction of version control system, e-book reader and also the issues we propose in the following sections.

1.1 Version control system

Version control, also known as revision control or source control, is the management of multiple versions of various information in a digit format. Such systems are commonly used to keep track of product evolution and backup in case of data loss (German et al. 2016). Changes to these documents are identified by incrementing an associated number or letter code, termed the "revision number" or "revision" (Milentijevic et al. 2008). Each revision is associated with the person making the change. Revisions can be compared, restored, and in some cases, merged.

A user interacts with a version control system by using a set of actions provided by the system. The following part of actions shows the common vocabularies and actions which will be used in this research.

Repository - The Repository is where files' current and historical data are stored, often on a server. In this research, it indicates database of the proposed BookRoll system.

Working copy - The Working copy is the local copy of files from a repository, and it contains only a single version number of the data to be changed. In this research, working copy will be used as a copy to store all the learning footprints made in old version learning material.

Diff - Also called Change, represents a specific modification to a document under version control system.

Merge - A Merge or Integration is an action in which two sets of modifications are applied to a file or set of files.

The proposed Diff program and Merge program will be demonstrated in the following sections.

1.2 E-book reader

E-book reader, also known as digital learning material reader, has become not only one of the most core part of modern formal education, but also an efficient data collection source in learning analytics area. The reading behavior made by students has previously been used to visualize class preparation and review patterns. This research proposed the BookRoll digital learning material reading system. As shown in Figure 1, there are many types of learning footprints users can make, users can use marker function to highlight sections of learning materials in yellow for the sections that were not understood, or red for import sections. Memo function can also be created at any pages with the specific section of the page. Users can also use bookmark function to mark any pages or use full text search function to find the information they need. Currently, learning material contents can be uploaded to BookRoll in PDF format, and be able to support a large scale of devices as it can be accessed through a standard web browser (FLANAGAN et al. 2017).

In this research, the connection of learning footprints across versions will be tracked and controlled properly. Due to the use of Transformation model, there are several reading behaviors in the BookRoll e-book reader need to be clarified. In the repository of BookRoll e-book reader, reading behaviors or action names will be stored in the repository, in Table 1 and Figure 1 we give the explanation of these reading behaviors.

Table 1. Explanation of reading behaviors made by e-book users

Reading behavior	Explanation
NEXT	While a user click button "NEXT" and turn to the next page on e-book reader, the learning footprint will be saved as "NEXT".
PREV	While a user click button "PREV" and turn to the previous page on e-book reader, the learning footprint will be saved as "PREV".
PAGE_JUMP	While a user click button "jump" and jump to another specific page on e-book reader, the learning footprint will be saved as "PAGE_JUMP".
MARKER	While a user click button "MARKER" to highlight the contents on e-book reader, the learning footprint will be saved as "ADD_MARKER".
MEMO	While a user click button "MEMO" to take some notes on the specific page on e-book reader, the learning footprint will be saved as "ADD_MEMO".
BOOKMARK	While a user click button "BOOKMARK" to mark the specific page on e-book reader, the learning footprint will be saved as "ADD_BOOKMARK".

Figure 1. A screenshot of the user interface of e-book reader BookRoll

1.3 Illustration of version control issues in context of e-book reader

Usually many types of reading behaviors can be made by users on an e-book reader, include opening learning material, turning to next page, adding memo etc. In this paper, we deal with learning footprints such as marker, memo and bookmark since these actions can be directly created and seen by users in the user interface of BookRoll. In order to connect all the learning footprints across versions on the e-book reader, we propose a serious issue need to be addressed with.

These reading behaviors, or we call learning footprints in this paper, need to be transferred to the correct positions and page numbers after a new version learning material been uploaded to the e-book reader, since several pages on the old version learning material might be changed or does not exist on the new version learning material and that will become a serious problem for the reader who intend to read the same learning material. As shown in Figure 2, this two screenshots from the proposed BookRoll e-book reader, indicate page 2 on the old version learning material and page 4 on the same learning material but with the new version. This two screenshots show the same page content but in the different page number of the same learning material, when e-book readers intended to read the new version learning material, they can only see their learning footprints in the wrong position, include all the memos, markers and bookmarks made on the old version learning material because these learning footprints did not be transferred to the correct positions and page numbers as shown in Figure 2. The exclamation mark shown in Figure 2 indicate that the positions or page numbers of the learning footprints made on the old version learning material might not be correct on the updated new version learning material.

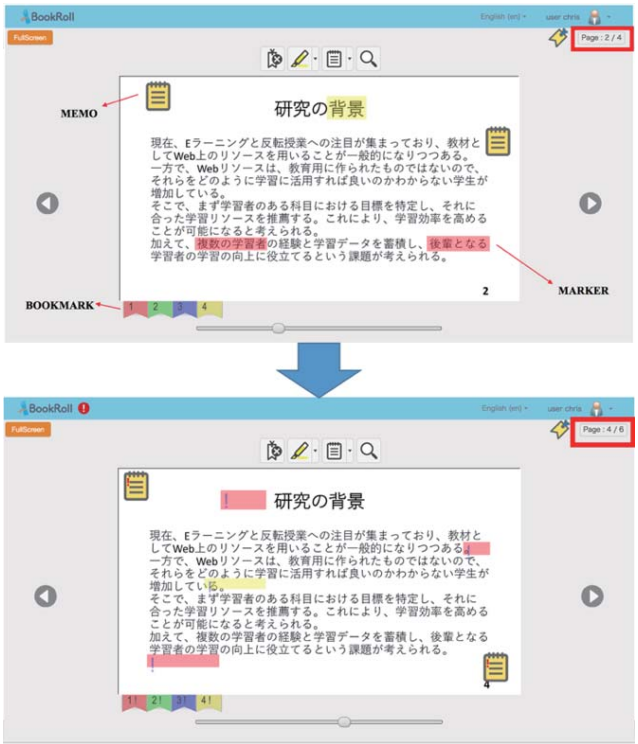

Figure 2. Two screenshots of page 2 on the old version learning material and page 4 on the updated new version learning material

To deal with the proposed issue above, we propose a process using Diff program, Transformation Model and Merge program which will be demonstrated in Methodology section.

2. Related work

In this section, we give several related works talk about how researchers used version control system in the aspect of software development, coursework or project-based learning in the past. In (Fischer et al. 2003), they introduced the population of a Release History Database that combines version and bug report data and further demonstrate some query examples with respect to software evolution analysis. In (Milentijevic et al. 2008), they addressed with the development of a generalized model for version control systems application as a support in a range of project-based learning methods. In (German et al. 2016), they presented a method that continuously mines all known D-VCSs (Distributed Version Control Systems) of a software project to uncover the complete development history of a project. In (Rocco 2011), they mentioned version control is not a peripheral or optional tool but a vital component in the software development toolkit, and also presented a model for incorporating version control into the entire project-based CS curriculum using a modern distributed version control system, Mercurial. In (Glassy 2008), they also used the Subversion version control system over CVS to help make clear how a software engineering student developed programs and gave two figures as their results to emphasize that using version control did help make student development processes more visible, besides, they also mentioned that a brief scan over a student's commit logs showed whether the student was making incremental progress or waiting to work on the assignment until just before a deadline. In (Laadan 2010), they used an approach to teaching operating systems based on virtual appliances, a distributed version control system and live demonstrations, they combined a virtual appliance VMware with a distributed version control system Git to provide reliable storage for students' homework assignments, support students working together on group homework assignments, and manage the submission and grading

of homework assignments, then they leveraged virtual appliances and the version control system to enable students to do live demonstrations of their work as part of grading their assignments to simplify grading, providing better feedback to students and greater interaction between instructional staff and students to facilitate learning.

Despite so many researches focused on software development, pedagogy, coursework or project-based learning, there is still a lack of the deployment on e-book based systems, so as the primary objective, this paper will focus on version control for the connection of learning footprints across versions within an e-book based system. We propose a process include Diff program, Transformation model and Merge program in this paper, and the methodology of this research will be explained in the next section.

3. Methodology

In order to deal with the issues above, we use some test data made by ourselves as our learning footprints for the propose process. As shown in Figure 3, after a new version learning material been uploaded to the BookRoll e-book reader, we will first execute Diff program then be able to generate the page control data which will be shown in Figure 4(the modification of pages between two different versions of the same learning material from the e-book reader). After that, all the learning footprints made by users on the old version learning material will be copied as a working copy, then we will use Transformation model to modify all the learning footprints in the working copy by following the proposed page control data, in the final of this process, these learning footprints will be added back to the main repository by deploying Merge program.

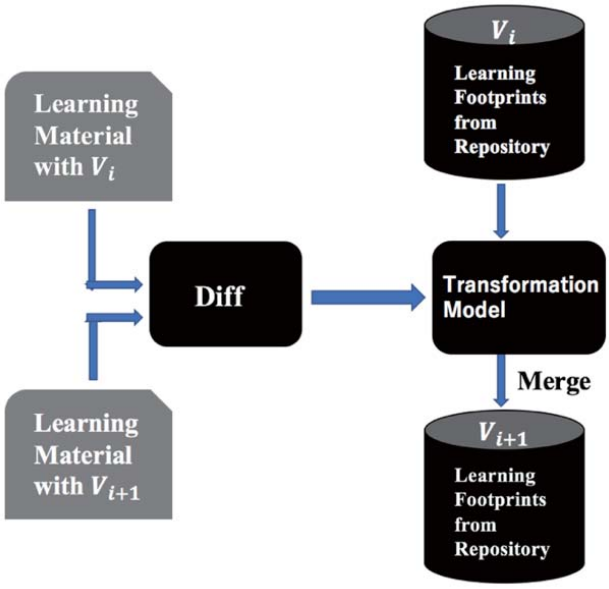

Figure 3. Process of the proposed method

3.1 Diff and page control data

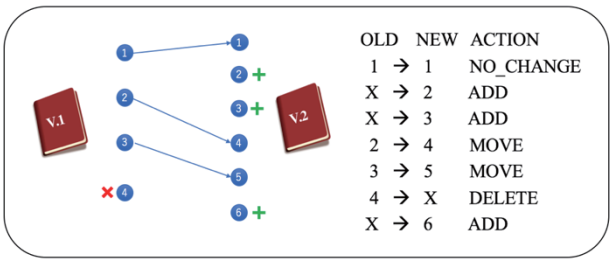

Figure 4. The page control data about how pages changed

As mentioned before, Diff program can be treated as the modification of pages between two different versions of the same learning material from the BookRoll e-book reader (page control data). This is the first part to be executed of this process. As shown in Figure 4, the picture in the left side indicates that there is a learning material in version 1 and version 2 with 4 pages and 6 pages, respectively and also shows how pages changed from version 1 to version 2 in image format. The picture in the right side indicates the page control data in text format, so according to the page control data, page 1, 2, 3 in version 1 learning material are ninety or one hundred percent similar to page 1, 4, 5 in version 2 learning material, respectively, and page 4 in version 1 learning material does not exist in version 2 learning material, meanwhile, page 2, 3, 6 in version 2 learning material are new pages to this learning material. After the detection from Diff program, next step will be the developed Transformation model.

3.2 Transformation model

Transformation model indicates the transformation for learning footprints on pages still exist after new version learning material been uploaded to the BookRoll e-book reader, which means pages being detected as NO_CHANGE or MOVE group from the page control data by Diff program as shown in Figure 4. Meanwhile it will remove all the learning footprints on pages does not exist in new version learning material.

Transformation model will be used to make a copy of these learning footprints and modify, transform them in the working copy. In the transformation model, we propose notations B and B′ as the set of page contents from the old version learning material and the new version learning material, respectively, and real number i and j are page numbers. In this paper, B and B′ will be described as below:

B = {1,2,3,4}, B′= {1,2,3,4,5,6}.

In addition, in the proposed transformation model, an empty set ∅ indicates that this learning footprint will be removed from the working copy.

Transformation for MARKER:

- ***DIFF_ADD/DELETE_MARKER(B_i, B'_j)***

$$= \begin{cases} \emptyset, & if\ \{B_i \notin B'\} \\ \text{ADD/DELETE\_MARKER}(B'_j), & if\ \{B_i = B'_j\}\,; i \neq j \\ \text{ADD/DELETE\_MARKER}(B_i), & if\ \{B_i = B'_j\}\,; i = j \end{cases}$$

Here we only give the example to demonstrate how transformation model work with all the markers in the current learning material, but other learning footprints like all the memos and bookmarks will actually follow the same rules.

3.3 Merge

Merge will be executed after all the learning footprints in the working copy been modified by using transformation model. This working copy will be added back to the original repository again and become learning footprints for the new version learning material by deploying Merge program, the implementation of this section will be summarized in the next section.

4. Implementation

This research implemented the core concepts of version control to modify the learning footprints left on the BookRoll e-book reader by users, and also create and modify the working copy with the learning footprints logged into the repository of the BookRoll e-book reader. The proposed issue can be fixed after the process

proposed by this research.

As shown in Figure 7, after teachers or instructors updated their own learning material, all the learning footprints such as markers, memos and bookmarks created by users on the interface of the BookRoll e-book reader will be transformed to the correct positions and page numbers from page 2 of the old version learning material to page 4 of the new version learning material after deploying the proposed process, and obviously users can easily see all the learning footprints they left on new version learning material in the correct position and page number again.

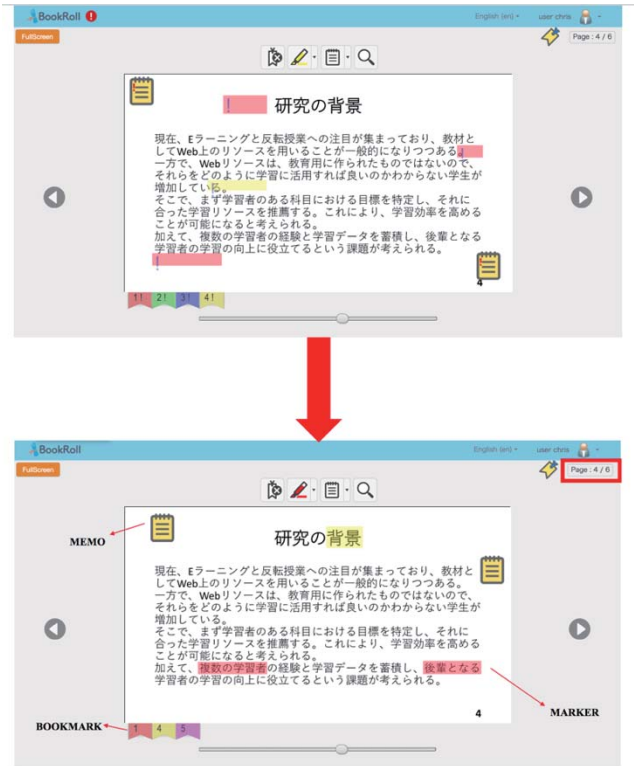

Figure 7. Page 4 on the new version learning material after the propose process

5. Conclusions

In this paper, a process to combine the core concepts of version control with an e-book reader to connect all the learning footprints across versions left from e-book users is presented. It has been proven that using the concept of version control for the connection of learning footprints is important, workable and be able to play a huge part for an e-book based system since instructor may update their learning material in anytime. As shown in Figure 7, all the learning footprints can be connected properly between old version learning material and old version learning material, so obviously, despite there are still several developments of the proposed process need to be enhanced, the proposed issues can be fixed by the proposed process include Diff program, transformation model and Merge program.

6. Future works

To specify the future works to this research, for now, we are just using our own test data made by ourselves as our learning footprints in a test environment, in the future we will use the real learning footprints created by other users, mainly students in a course, as our data sets to integrate this process to the real BookRoll system. Meanwhile, this implementation focuses on ninety or one hundred percent similar pages between two different versions of the same learning material, but in many cases, these two pages might really similar to each other and they will still be determined as different pages due to the proposed Diff program, and all the former learning footprints on those pages will be removed, which means we might lose some important information from those learning footprints, so as our primary objective in the future, a development to determine any kind of changes happened between two different versions of the same learning material is currently ongoing.

References

[Milentijevic 2008] Milentijevic, I., Ciric, V., & Vojinovic: Version control in project-based learning, Computers & Education, 50(4), 1331-1338.

[Ogata 2017] Ogata, Hiroaki, et al: "Learning Analytics for E-Book-Based Educational Big Data in Higher Education.", Smart Sensors at the IoT Frontier, Springer, Cham, 2017. 327-350.

[Fischer 2003] Fischer, Michael, Martin Pinzger, and Harald Gall: "Populating a release history database from version control and bug tracking systems.", Software Maintenance, ICSM 2003. Proceedings. International Conference on. IEEE, 2003.

[German 2016] German, Daniel M., Bram Adams, and Ahmed E. Hassan: "Continuously mining distributed version control systems: an empirical study of how Linux uses Git.", Empirical Software Engineering 21.1 (2016): 260-299.

[Cederqvist 1992] Cederqvist, Per, et al: "Version management with CVS.", URL http://www. cvshome. org/docs/manual (2006).

[Glassy 2008] Glassy, Louis: "Using version control to observe student software development processes.", Journal of Computing Sciences in Colleges 21.3 (2006): 99-106.

[Laadan 2010] Laadan, Oren, Jason Nieh, and Nicolas Viennot: "Teaching operating systems using virtual appliances and distributed version control.", Proceedings of the 41st ACM technical symposium on Computer science education. ACM, 2010.

[Rocco 2011] Rocco, Daniel, and Will Lloyd: "Distributed version control in the classroom.", Proceedings of the 42nd ACM technical symposium on Computer science education. ACM, 2011.

Contact:
Name: Christopher C.Y. Yang
Affiliation: Graduate School of Informatics, Kyoto University
Mail: chris20834@gmail.com