

# A Tool for Collaborative Guitar Chords Creation based on The Concept of The Distributed Version Control

Chakkrit Tantithamthavorn<sup>1</sup>, Papon Yongpisanpop<sup>2</sup>, Masao Ohira<sup>2</sup>,  
Arnon Rungsawang<sup>1</sup> and Kenichi Matsumoto<sup>2</sup>

<sup>1</sup>Department of Computer Engineering,  
Faculty of Engineering,  
Kasetsart University, Bangkok, Thailand  
{ b5105256, fenganr } @ ku.ac.th

<sup>2</sup>Graduated School of Information Science,  
Nara Institute of Science and Technology,  
Nara, Japan  
{ papon-y, masao, matumoto } @ is.naist.jp

## ABSTRACT

The distributed version control system has become very popular in software development process for years. Many software projects, which are developed by a large group of programmers have been using it to manage their source code. We study on another aspect of applying a distributed version control to keep track the version of other artifacts instead of using just for tracking source code. An application for sharing and creating guitar chords is used in our case study. We apply distributed version control concept in order to interoperate musical ideas from multiple users and preserve the versions of guitar chords with their music styles introduced to the community. The results show how distributed version control system can help guitarists to collaborate on creating guitar chords and also reduce the redundancy of derivative versions. This study has approved that distributed version control system can be applied to any projects that need to keep track the version of their artifacts.

## Categories and Subject Descriptors

H.5.3 [INFORMATION INTERFACES AND PRESENTATION]: Group and Organization Interfaces - *Asynchronous interaction, Collaborative computing, Web-based interaction*; H.5.5 [INFORMATION INTERFACES AND PRESENTATION]: Sound and Music Computing - *Methodologies and techniques, Systems*.

## General Terms

Design, Human Factors, Experimentation

## Keywords

Chord Archives, Distributed Version Control, Collaborative Music Creation and Sharing

## 1. INTRODUCTION

Ultimate-guitar.com is a large guitarist community website that has a large number of guitar tablature. Due to the large amount of tablatures, overlapped (same) songs and many derivative versions are usually submitted to the archives. As a result, users are having a problem on how to choose a song. Moreover, if some parts in a guitar chord are incorrect, guitarists cannot continue to play the guitar chord. Since the current guitar chord archives do not provide the permission to edit an archived chord by anonymous users, the online music communities confront with a severe shortage of contributors and lack of collaboration among guitarists. To solve those problems, we came up with new software called ChordBook, which applies distributed version control system to keep track of guitar chord versions, reduce the redundancy and allow guitarists to do the collaboration.

Today's software development is usually done in a distributed environment [1]. Many researchers focus on the area and apply this concept to their work. Both Rocco et al [2] and Ladden et al [3] apply distributed version control to the classroom for improving class teaching. One of the common purposes of distributed version control is to enable groups of people work together on any kind of files without necessarily being connected to a common network. They can change, archive, merge, branch or synchronize files of which the system keeps a history. Another purpose is archiving and backup so any files in the system should be safe and free of data-loss.

In section 2, we purpose the architecture and key designs of ChordBook. In section 3, we demonstrate the software and show the empirical study on the application of distributed version control system. And we conclude everything up in section 4.

## 2. DESIGN

The architecture of ChordBook is separated into two parts. The first part is the software for guitarists to create and share guitar chords. The software is developed as an iPad application due to the portability. The second part is a server repository. It acts itself as a guitar chords archive, provides necessity services for communicating with a client-side application.

The architecture of ChordBook is based on four keys as follows:

1. Contribution – According to gain more guitar chords into the archive, we need to have more contribution from guitarists. ChordBook allows guitarists to easily distribute their guitar chords to the archive.
2. Distribution – According to guitarists are spread all over the world and guitar chords can be done by many guitarists, the server side will be able to keep track of versions and also have a good method to distribute the guitar chords.
3. Availability – ChordBook allow users to check out the songs that they want and it will automatically download and save them into their local storage.
4. Collaboration – A million heads is better than one. Guitarists are able to access any song sheets, which are shared on the server via ChordBook. If other guitarists notice there is a mistake in a song sheet or if they have more information on the song, they will be able to edit the song or provide more in-depth explanations. This collaboration key allows guitarists to help each other to create a better guitar chord.

In the ChordBook client, we use SQLite database as a local repository on iPad to keep track of versions of song sheets. Song sheets will be stored in the database as ASCII text based format as shown in Figure 1. ChordBook allows users to have various versions of song by creating branches for it. Figure 2 shows that a

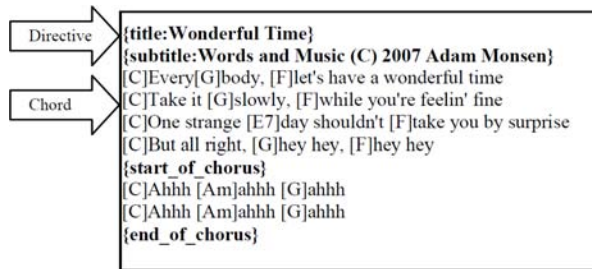


Figure 1. Guitar chord data format

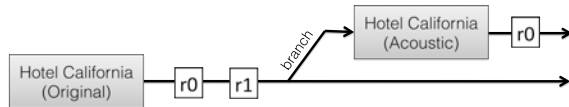


Figure 2. ChordBook Interface on iPad

user can create an acoustic version of the song, which is based on the original one. When the user is done with the song sheet and is ready to share or publish it, he/she can push that song sheet to the server repository. Users do not need moderators or system administrators to ask for a right to access repositories. The distribution process and rights management are done by guitarists among themselves. As a result, no special write access and no politic are needed.

### 3. DEMONSTRATION

The goal of our demonstration is to show how ChordBook can facilitate distributed version control concept in several aspects and discuss the potential applications that are feasible using distributed version control.

To combine musical ideas among guitarists, we get started from “clone” to download song sheet from server to iPad. If user notices there is a mistake or if he/she would like to provide more information on the song, then the user can edit the song, provide more in-depth explanations or discuss it among users. During the mediation, one can “commit” to his/her iPad to keep track of versions in local repository. If a user prefers to distribute or publish his/her guitar chord to the Internet, he/she can “push” the guitar chord to the server. If the server side finds a related song title name, the server will return a set of the result to the user. The user has to decide which one is a better title for the song. We believe that human decision can solve the problem of similar song titles repeatedly created. If the guitar chord is completely new to the server, it will be accepted right away. For an updated version, the server will first check whether that song version is up to date. If it is already up to date, the server will accept user’s request. If it is not, this version will conflict and users will be guided to have discussions among users. ChordBook also have two ways to handle a spam and malicious users. First, the system allows users to report a spam and then will send a request to a moderator. Second, if the system detects some frequently changed or updated guitar chords within minutes, it will lock the song to be edited automatically and stop the update from users for 24 hours.

To avoid the duplication of song titles when there are more than one guitarists want to create the song and commit it into ChordBook, the distributed version control will prohibit the duplication of song titles to be created on the server.

To handle various types of a song as shown in figure 2, e.g., Hotel California will be able to create just once but ChordBook will allow this song to have many types of itself (i.e., acoustic, classic and etc.) using branch to represent types of that song.

### 4. CONCLUSION

In this paper we have described an empirical study on an application of distributed version control to music community. This application is novel in the sense that it fully utilizes a version control to keep track versions of guitar chord song sheets in chord archives. Our study could be applied to other artifact creations that need collaboration among users to keep track the versions of the artifact itself. For the future work we are planning to extend ChordBook to effectively treat other music instrument, enhance flexibility and scalability of the user interface.



Figure 3. ChordBook Interface on iPad

### 5. ACKNOWLEDGMENTS

This research was supported by Nara Instituted of Science and Technology (NAIST), Japan, and Kasetsart University, Thailand. Thanks to anonymous reviewers for their comments.

### 6. REFERENCES

- [1] Brian de Alwis and Jonathan Sillito. 2009. Why are software projects moving from centralized to decentralized version control systems?. In *Proceedings of the 2009 ICSE Workshop on Cooperative and Human Aspects on Software Engineering (CHASE '09)*. IEEE Computer Society, Washington, DC, USA, 36-39.
- [2] Daniel Rocco and Will Lloyd. 2011. Distributed version control in the classroom. In *Proceedings of the 42nd ACM technical symposium on Computer science education (SIGCSE '11)*. ACM, New York, NY, USA, 637-642.
- [3] Oren Laadan, Jason Nieh, and Nicolas Viennot. 2010. Teaching operating systems using virtual appliances and distributed version control. In *Proceedings of the 41st ACM technical symposium on Computer science education (SIGCSE '10)*. ACM, New York, NY, USA, 480