Development and Practice of Programming Learning Course Management System with Version Control Software

Motoki Miura[†] **Susumu Kunifuji**[†]
[†]School of Knowledge Science, Japan Advanced Institute of Science and Technology
{miuramo,kuni}@jaist.ac.jp

Abstract

We have developed "SVNLecture," a programming course management system based on a version control software subversion. SVNLecture is a Web-based system for the management of programming courses. SVNLecture assists the teachers to create SVN repositories for their course and to manage student access permissions for the repositories. We also developed a special SVN client "SVN4Lec" for novices who have limiter computer skills. SVN4Lec provides basic functions such as checkout/update and commit with simple GUI. Therefore, SVN4Lec reduces the burdens of the students with regard to the downloading, deploying, and submitting of their files. We applied SVNLecture and SVN4Lec for a C# programming course for novice learners and determined its effectiveness in practice.

Keywords: Subversion, Course Management System, Trac

1 Introduction

While learning a programming language, the students often edit their source codes and compile them to build a program to check the statements for the purpose of understanding the programming language and for acquiring programming skills. The students sometimes submit their source codes as reports by printing on paper or emailing or uploading them via the Web browser. A teacher and a teaching assistant check the reports and reply to the students. A paper report is suitable for the teaching staff to annotate the comments; however, the students cannot refer to these comments until the reports are returned to them. Submitting the source codes as an email attachment or uploading them via the Web browser requires the students to manage their files by themselves. If they lose their files, then they cannot be recovered. Moreover, the students must learn to submit the source codes via email or a Web browser.

In order to reduce the problems associated with these file management tasks and additional operations for submission, we have developed "SVNLecture," which is a programming course management system based on version control software subversion. SVN-Lecture consists of two parts: a server-side repository and its management Web interface and a client-side program termed "SVN4Lec." SVNLecture and SVN4Lec are designed for effectively introducing a state-of-the-art programming source management environment with simple interfaces, particularly for novice programmers and beginners. Using the client software SVN4Lec, the student can easily download the template files and sample project files prepared and developed by their teachers. After the modification of the files and examination of the behavior of the program, the students can easily submit his/her assignments by the click of a button.

2 Related Studies

The PHP simple online submission system (phpSOSS) [1] is a Web-based system that accepts the workfiles of the student. The system is developed on the basis of open source products such as PHP and MySQL. The phpSOSS provides simple interfaces to manage the submitted files. EduComponents [2] is a plug-in of Plone CMS. The plug-in provides automatic scoring function for reducing the burden of the teaching staff. The programs in Python, Haskell, Scheme, CommonLisp, and Prolog can be handled and evaluated by the plug-ins, and the results can be sent to students.

There are several studies that discuss the utilization of the version control system for managing the workfiles of students, such as Reid et al. [3] for the CVS (concurrent versions system) and Glassy [4] for Subversion. In particular, the study by Reid et al. provides a detailed description of the pros and cons of utilizing the version control system on the basis of their prac-

tice. However, both the practices and studies selected students who majored in computer science as learners. The students were required to learn not only programming but also the state-of-the-art project management with other members in order to become a professional developers and project managers. Thus, they could accept the use of command line tools of the conventional CVS and Subversion. In the practice of Glassy, the teacher asked the students to create their own repositories and import initial files so that they can get accustomed to the version control system. Our practice is different from abovementioned ones in two points: (1) In our practice, students are provided with a special GUI client. Therefore, even beginners can utilize the mechanism of version control. (2) We utilized the "Trac" system to provide a Web interface to confirm the status and diff lines of any versions.

3 Subversion/Trac

In this section, we describe a version control system Subversion [5] as well as the project management tool, Trac.

3.1 Subversion

Subversion [5] was developed by CollabNet Inc. Its basic functions are derived from the CVS; however, it shows improved functionality with regard to the unified managing of directories, binary, and text files. Thus, Subversion is simpler yet powerful than CVS.

Generally, Subversion utilizes a *repository* to store the managed files. First, the users "check-out" the latest snapshot of the repository. The latest snapshot in the storage of the user is called the "working copy." After the user edits the files of the working copy, he/she commits the modifications of files (result of editing) into the repository. The user can add messages to the committed files. The repository stores a snapshot of the files as a revision. Once the check-out had performed, "update" operation should be used to download the latest revision.

For students, Subversion offers the following benefits:

- A student can easily restore files by selecting the previous revision. This assists the students in reducing the burden if he/she failed to edit the files. This feature encourages the student to attempt advanced and drastic modification of his/her files.
- Once committed, the snapshots of the files are perfectly stored. Even if the storage medium of the student is broken or damaged, the snapshot of the committed files can be easily recovered.
- The student can put their working copy on several computers and environments (such as university, dormitory, and home). He/she easily organize these working copies by updating before editing.
- The students can store their working copies on several computers and environments (such as the university, dormitory, and home). He/she can easily organize these working copies by updating them before editing. The student must only (1) download template files and sample project files, (2) edit only the necessary parts and files, (3) and commit. If the tool is not used, the student has to download the compressed archive, extract it, and then deploy the files. When the files are already organized by the teachers, the students do not have to tackle complicated path settings.
- The history of snapshots is stored in the repository as revisions. Thus, the student can gauge his/her progress of learning by browsing the revision log.

Incidentally, Subversion is also advantageous to the teaching and administrative staff for managing the progress of the students. All the revisions stored in the repository are the only "differences" between pre- and post-snapshots. Moreover, the teacher can distribute the initial template files and project files for each student as a set of "references," not as "duplicated copies." Therefore, the required storage is considerably reduced, even when the repository stores additional data for modifications on the committed files.

3.2 Trac

Trac [6] is a project management tool developed by Edgewall Software in Sweden. Trac provides a Wiki, a repository browser, a ticket system to track bug, and so on. In particular, the repository browser can be helpful for students to confirm their commit and its log message on the Web browser, and highlight the differences between the two revisions.

4 SVNLecture

To facilitate course content management with Subversion, we have developed a metacourse management system called SVNLecture. SVNLecture consists of (1) a Web interface for the lecturer/supervisor, (2) a Web interface for the students, and (3) specialized client named "SVN4Lec" for students.

4.1 Design Criteria

For the design of the programming learning environment with Subversion, we consider the following important factors.

1. Even novices and beginners can utilize the tool to benefit from the functions of Subversion.
2. The teaching staff can easily perform management tasks such as the creation of new repositories and the registration of the students.

Considering the first criterion, the number of additional operations such as the installation and setup of the software should be minimized. Therefore, we have prepared a special subversion client that involves necessary setup preferences as a Java application. The reason for selecting Java is that the same binary runs on several VMs of the platforms. To avoid deployment errors such as the linkage of libraries and omit installation, we organized the software into one Java archive format file. Users can invoke the software by double-clicking on the file icon. As the second criterion, we developed a Web-management interface for lecturers, teaching staff, and supervisors.

4.2 Student Scenario

To explain the flow of the lecture with SVNLecture, we describe a scenario in a step-by-step manner.

A student receives an email containing a request URL for the creation of an account of SVNLecture from a teaching staff member. When the student sets his/her password via the URL, the student account becomes valid.

The student is required to visit SVNLecture Web manager (Figure 1) and login with the password. Subsequently, the personal page of the student will be loaded, and he/she can download

Figure 1. SVNLecture WebManager

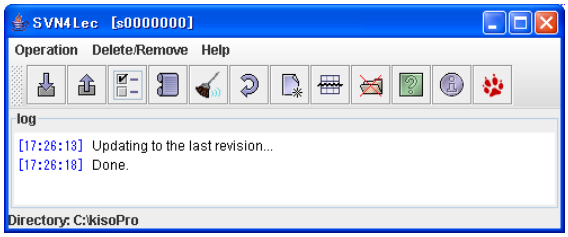

Figure 2. SVN4Lec: A Subversion client specialized for lecture

the SVN4Lec.jar file that is the specialized subversion client for the lecture.

When the student clicks on SVN4Lec.jar, the main window of SVN4Lec (Figure 2) appears. The main window consists of a menu bar, tool bar, and log message window. To download the template files or initial project files, the student clicks on the "Update/Checkout" button located at the leftmost end of the toolbar. After authentication, SVN4Lec downloads the initial files into the student folder of the repository.

After downloading the files, the student works on an assignment as usual by editing the files and building the program to check whether the behavior of his/her program satisfies the assignment. For submission, the student presses the "commit" button, which is the second button from the left. A window appears and it enables the user to confirm the committed files and input messages regarding the modifications. Finally, the snapshot of the committed files is stored as a *revision* with the message into the repository.

To check the submission, the student presses the "Open Trac page" button located at the right-

most end of the toolbar. SVN4Lec opens a repository browser of Trac. The student can confirm the commit status, revision history, differences in modification (see Figure 6 and Figure 7), and comments from the teaching staff.

After this, the student presses the "Update/Checkout" button to synchronize the working copy with the repository before working on the assignments. By this operation, the student can download additional distributed files and merge the annotations/comments written by the teaching staff between the lines of source code into his/her working copy.

4.3 Teaching Staff Scenario

A teacher who conducts a lecture first logs in to the Web manager (Figure 1) through an administrator account. With this administrator account, the teacher can create a new repository for the lecture and manage the participants in the course. A participant registration page (Figure 3) enables the teacher to add, freeze, and delete student accounts by entering the account name and email. According to our fundamental and simple policy, there should be one repository for one lecture in SVNLecture. When the teacher creates a directory, whose name is the same as that of the student account, the student can only access the files contained in the directory. Figure 4 shows an example of the structure of the repository. Incidentally, the "base" refers to an administrative directory for preparing the files before distribution.

Next, the teacher sets parameters for SVN4Lec in SVN4Lec.jar. These parameters are the URL of the repository and the URL of the lecture information page, if provided. The teacher stores the Java archive file at any location where the students can access it, such as a Web server or a shared folder. Then, the teacher prepares instructions for the students on how to download the JAR file as a Wiki page on Trac.

After preparing the student accounts, directories for students, modified SVN4Lec.jar, and instructions, the teacher sends an email containing a request URL for creating an account for each student via the SVNLecture system.

When a student commits, the teaching staff can receive a notification email. The notification email contains information regarding the person who committed the files and the committing time

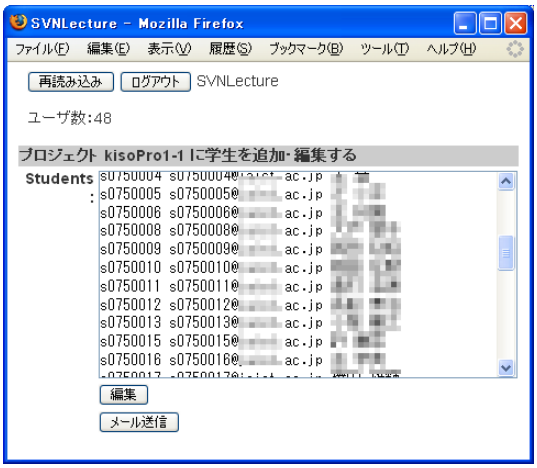

Figure 3. SVNLecture: Manage Students

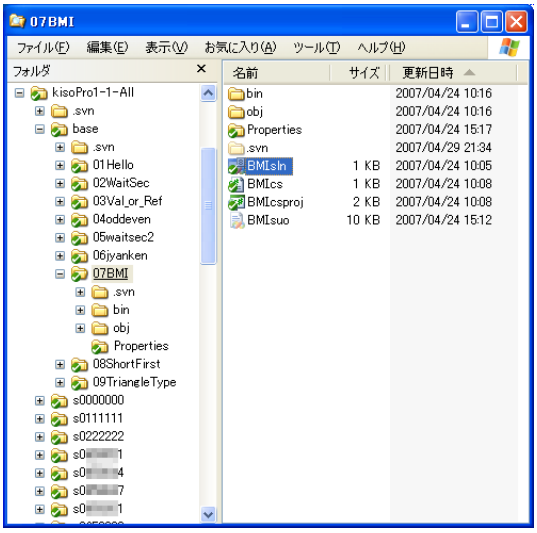

Figure 4. Repository Structure

and the message on the revision. The copy of the notification email is also sent to the student. The teacher can open the URL mentioned in the email to check the revision log on Trac. Apparently, the teaching staff has the authority to retrieve the working copies of all the directories including the progress of the students. Therefore, the teaching staff can add comments to the source code files or modify them directly. It should be noted that the modifications by any teaching staff are also recorded as revisions.

Even if the abovementioned mechanism is presented, it is difficult to insert comments in all the student workfiles. To reduce this inconvenience, we provide a feedback help tool. This tool enables the teacher to collectively insert comments into the workfiles. Once the teacher

prepares a CSV file that includes the assignment title, path to workfiles, and the score and comment for each student (see Figure 5), the tool reads the items to be written in the corresponding files and commits with the comments automatically. Students can check the message from the Trac pages (see Figure 6 and Figure 7) as well as the source codes.

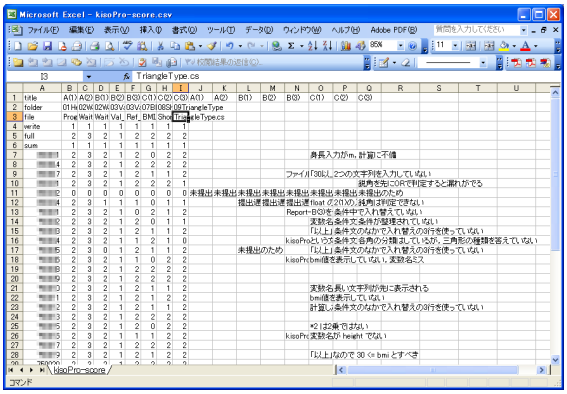

Figure 5. CSV file for evaluation and comment feedback

Figure 6. Revision View

5 Implementation

We describe the details of the system implementation of SVNLecture and SVN4Lec, which are not mentioned above.

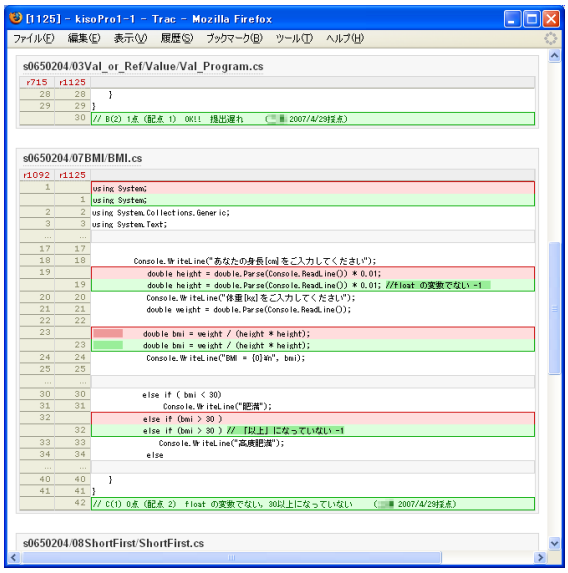

Figure 7. Scores and Comments by Teacher

5.1 SVNLecture

SVNLecture, a Web interface for managing the repository and student participation, is developed using PHP script and MySQL. As shown in 4.3, SVNLecture does not intend to create SVN repositories for each student. Rather, we recommend teachers to create one repository per course (known as "unified repository architecture"). This approach is advantageous not only for saving the disk space of the repository but also for managing the workfiles of the students.

Meanwhile, the unified repository architecture requires an SVN hosting server to control the access with authentication. If this mechanism is not used, the students can freely access the workfiles of other students. Although this behavior is effective for collaborative development, it should be prohibited by the system. In order to effectively control the access permissions, we have implemented a function for the dynamic configuration of the access list.

The authentication of the SVN can be carried out by two configuration files: an access control list (ACL) and a password. When the teacher adds participant accounts to the lecture, SVNLecture automatically updates the ACL file, as shown in Figure 8. It should be noted that "kisoPro1-1" represents a repository name, and s0xxxxxx indicates a student account. In addition to the modification of the ACL, SVNLecture modifies the password file (`.htpasswd`). Since we chose the SVN

```
[kisoPro1-1:/]
teacher = rw
* =
[kisoPro1-1:/base]
teacher = rw
* = r
[kisoPro1-1:/s0000000]
s0000000 = rw
[kisoPro1-1:/s0111111]
s0111111 = rw
[kisoPro1-1:/s0222222]
s0222222 = rw
.....
```

Figure 8. Access Control List for Repository (svnaccess). "kisoPro1-1" is the name of lecture/repository.

module of the Apache Web server to provide repository service via HTTP, SVNLecture utilizes the password files to authenticate the users. An `AuthUserFile` directive of the Apache configuration file specifies the password file (see Figure 9). After authentication, an ACL specified by `AuthzSVNAccessFile` directive checks the permission in order to limit the access to the repository. The Trac system also refers to the same two files to provide consistent permissions.

SVNLecture provides a Web interface for changing and resetting the passwords for the members. When a member changes his/her password, SVNLecture replaces a corresponding line of the password file. The updated password file becomes valid without restarting Apache. The resetting interface is crucial for reducing the difficulties encountered by the teaching staff because the students tend to lose their passwords. Apparently, the student can change his/her password only on the Web.

5.2 SVN4Lec

SVN4Lec is developed as a Java application. We adopt the svnkit[7] SVN library to introduce the functionality of the Subversion core.

SVN4Lec reads a configuration file (`config.txt`) embedded inside of the JAR file. The configuration file includes three types of URLs: checkout base, help, and system update. These URLs will be modified to suit with the courses.

When the checkout/update button was pressed, SVN4Lec first checks the existence of the .svn directory under the current path. If it exists, the "update" operation is selected since the directory is already version-controlled. Otherwise, the initial "checkout" operation is selected. SVN4Lec asks a user to input his/her account name and password for checkout/update for the first time. With this account information, SVN4Lec attempts to access the repository of the student. The student repository URL is determined by concatenating the *checkout base* and *account name*.

Similar to a general SVN client, SVN4Lec provides the "update to revision" function to enable students to retrieve the previous snapshot. In addition, there are several specific functions such as "delete password cache file," "disconnect (unversioned by removing the .svn directories recursively)," "automatic update," and "open Trac page" for the exclusive use of a lecture.

6 Practice

We adopted the tools for a programming course for novice students. The course was held in April and May 2007. The objective of this course was to learn the fundamentals of a programming language for beginners in C#. The number of students was around 40.

6.1 Course Design

All of the students were graduates, but they had very less experience in programming. In this course, we considered "exercise as important" for understanding the concepts of programming. Therefore, we planned to assign the homework for programming every week. However, we wanted to avoid unnecessary errors such as simple grammar mistakes, compile errors, runtime errors, and misuse of build commands. To fulfill these conditions, we adopted the Visual Studio .NET 2005 (VS2005) IDE as a default environment. Considering the level of the students, we focused on the console applications such as assignments. We prepared initial VS2005 console application project files for each assignment and distributed them to the students because of the following two reasons: (1) Even though a task for developing a new project in VS2005 is not very difficult, it may confuse the first-time programmers. (2) We wanted to collectively handle and evaluate the workfiles of all the students. If the initial files are distributed, the students rarely change the filename. We distributed the initial files by using a couple of scripts with the "svn copy" command.

```
<Location /lecture/svn>
DAV svn
SVNParentPath /home/svn/lecture_material/reposroot
SVNIndexXSLT "/svnindex.xsl"
AuthType Basic
AuthName "SVNLecture"
Require valid-user
AuthUserFile /home/svn/.htpasswd
AuthzSVNAccessFile /home/svn/lecture_material/svnaccess
</Location>
```

Figure 9. Excerpt from Apache Configuration file (httpd.conf)

We introduced SVN4Lec and explained the usage of SVN4Lec for approximately 15 min in the lecture. The explanation involved (1) the fundamental operations of updating and committing (described in section 4.2), (2) some advanced functions of restoring ("update to revision" function) and reverting, and (3) typical steps of working an assignment (including confirmation with Trac). We also explained how to edit and build a program with VS2005.

SVN4Lec provides a function for the addition of extra files by students. The students can select files to be added from an unversioned file list in the commit window. We did not explain how to add their files to the repository in the instruction because it is not necessary for the completion of this assignment. However, several students added their own files to the repository while committing. Incidentally, temporarily generated files and unnecessary files in VS2005 project (.suo, .exe, below obj/ and bin/) were excluded from the unversioned file list in SVN4Lec.

6.2 Findings

We observed the following advantages.

- Students could download the initial project files and commit their work without unanticipated problems.
- They could easily re-submit their files.
- The teaching staff could make full use of a differential view of Trac repository browser for reviewing. We set a two-step assignment for students. In the first step, the student was instructed to just input the source code and check the behavior. In the second step, the student was required to modify the code and commit. Thus, the differential view of Trac highlighted only the modifications. The teaching staff could easily review with the differential view, even when the original source code was huge.
- The shared source code was quite effective to communicate rather than to email. The teaching staff could directly add comments to the source code of the student.
- The shared source code was helpful for troubleshooting. The teaching staff could easily find the causes for the problems faced by the students by building the application themselves instead of the students.

Several problems were also found in the practice.

- "Update to revision" could retrieve the previous snapshot. However, the modification on the previous snapshot was not supported by SVN4Lec. Thus, the students often faced difficulties when they committed. The interim solution is to overwrite the file after updating the latest revision.
- "Updating before working" should be informed effectively; In particular, the modifications of the teaching staff may conflict with those of the students.
- The error message of SVN4Lec was very inconvenient for fixing of the problem by the students.
- The updation took time to process the checking of the modifications by SVN4Lec when there was an increase in the number of files. This behavior depends on the machine specifications. We utilized PCs with a memory of 640 MB.

After the experiment, we solved the first problem of the modification of the previous snapshot by introducing a rollback function that merges the differences between previous snapshot and current revision on the current revision. This improvement will reduce the trouble on backward references.

6.3 Discussion

The total size of working copy at the end of the lecture, which includes only one snapshot, was 69 megabytes; Whereas the total size of repository on the server, which includes all snapshots, was 13 megabytes. That means the efficiency of subversion on file management. The total number of commits was 2,065. This number included the 1,020 commits for material distribution or comments by the teacher. That implies that the students committed only for the uploading of their assignments and not for their personal records.

We provided a simple questionnaire with the following three items: (1) Is the "update to revision" function useful? [RB] (2) Is the repository browsing function on the Web useful? [Diff] (3) Is SVN4Lec effective for lecture? [Effect]. According to the result (see Figure 10), the response distributions were similar among the items. There were two groups: one supported the management of assignment by subversion, while the other did not. The latter might have been confused by the complicated mechanism of subversion. We will continue to evaluate its effects on the learner's perception.

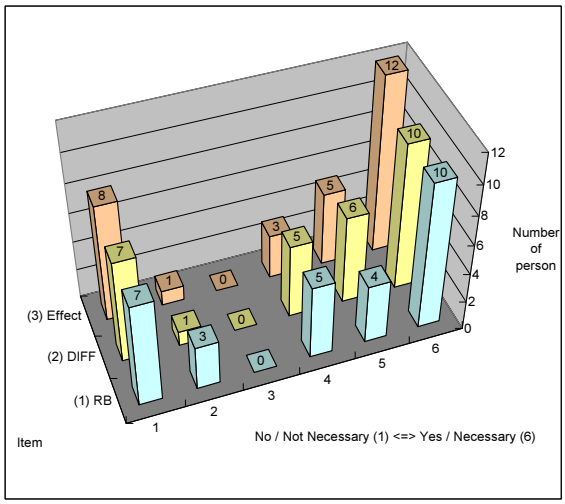

Figure 10. Questionnaire Result

7 Conclusion

We have developed a programming course management system based on Subversion and Trac. To ensure that the state-of-the-art environment is applicable for beginners, we also developed a specialized Subversion client SVN4Lec. For the teaching staff, we prepared a Web interface for managing a repository and the students for a course. We described the merit and use cases of the system as well as the system implementation details. We applied the system for a C# programming course and stated the findings and revealed the problems.

Currently, the study only focuses on the effectiveness of the course from viewpoint of the administration and not the student. For future study, the changes in the mindset of the students due to the introduction of this system should be focused upon in long-term experiments. The students may have different perceptions regarding the file management tasks by the simple rollback function. The perception of the students must be considered because this may evolve the learning methodology in programming courses.

In order to ensure that the effective solutions are available, we have been provided the source code of SVN4Lec as an open source code[1].

References

[1] David Wolff. A Web-based Tool for Managing the Submission of Student Work. *Journal of Computing Sciences in Colleges*, 20(2):144–153, December 2004.

[2] Mario Amelung, Michael Piotrowski, and Dietmar Rösner. EduComponents: Experiences in E-Assessment in Computer Science Education. In *ITICSE '06: Proceedings of the 11th annual SIGCSE conference on Innovation and technology in computer science education*, pages 88–92. ACM Press, June 2006.

[3] Karen L. Reid and Gregory V. Wilson. Learning by Doing: Introducing Version Control as a Way to Manage Student Assignments. In *SIGCSE '05: Proceedings of the 36th SIGCSE technical symposium on Computer science education*, pages 272–276. ACM Press, February 2005.

[4] Louis Glassy. Using Version Control to Obverve Student Software Development Processes. *Journal of Computing Sciences in Colleges*, 21(3):99–106, February 2006.

[5] Ben Collins-Sussman, Brian W. Fitzpatrick, and C. Michael Pilato. *Version Control with Subversion*. Oreilly & Associates Inc., 2004.

[6] Edgewall Software. The Trac Project. `http://trac.edgewall.org/`.

[7] TMate Software. SVNKit – Subversion for Java. `http://svnkit.com/`.

[1] `http://css.jaist.ac.jp/~miuramo/svnlec/`