

Subversion を用いた学習管理システムの構築と プログラミング初学者への適用

三 浦 元 喜[†] 國 藤 進[†]

バージョン管理システム Subversion を使用して教材を配布、実習成果物やレポートを収集する学習管理システム SVNlecture を構築した。SVNlecture は講義用リポジトリの作成と、学習者の登録とアカウント管理、学習者の個別フォルダへのアクセス権の設定などを簡単に行うための Web インタフェースである。また初心者が簡単に講義用リポジトリにアクセスし、作業コピーを作成したり実習成果ファイルを提出できるようにするために、SVNlecture 専用 Subversion クライアントを実装した。受講者は、ボタンを押すだけでチェックアウトや更新、コミットなどの操作をリポジトリのパスを意識することなく完了できる。そのためファイルの管理や提出にかかる負担が軽減でき、本来のプログラミング学習に集中しやすくなる。プログラミング初学者に C# を教える講義において構築したシステムを運用し、有効性に関する知見を得た。

Development and Practice of Programming Course Management System with Subversion

MOTOKI MIURA[†] and SUSUMU KUNIFUJI[†]

We have developed "SVNlecture," a programming course management system based on a version control system subversion. SVNlecture is a web-based system to manage programming courses. SVNlecture helps teachers to create SVN repositories for their course, and to manage permissions for students' access into the repositories. We also prepared a special SVN client "SVN4Lec" for novices who has few skills of computer literacy. SVN4Lec provides basic functions such as checkout/update and commit with simple GUI. Therefore SVN4Lec reduces students' burdens of downloading, deploying and submitting their files. We applied SVNlecture and SVN4Lec for C# programming course for novice learners, and obtained several effectiveness on the practice.

1. はじめに

プログラミング学習においては学生が実際にソースコードを編集し、動作を確認しながら理解を深めるといった活動がよく行われる。その活動状況を教員や TA が確認する手段として、学生が作成したプログラムを紙に印刷してレポートとして提出したり、メールや Web ブラウザ経由で送信するといった方法がとられることが多い。紙媒体で提出されたプログラムはコメントを直接書き込むことができる反面、紙が返却されるまで学生はフィードバックを受けることができないという短所がある。メールや Web ブラウザ経由でプログラムを電子的にやりとりする場合、学生は必要なプログラムをメールに添付したり、アップロードす

るための操作方法を別途覚える必要がある。また学生がソースコードの管理やバックアップを自ら行う必要があり、ファイルを誤って削除してしまうといった状況に対応しづらい。

我々は上記に挙げたファイル管理やアプリケーション操作にかかる手間を軽減し、できるだけ本来のプログラミング学習に集中しやすくすることを目的として、バージョン管理ソフトウェアである Subversion を利用したプログラミング学習管理システムを構築した。本システムはプログラミング初学者にも簡単に使えるよう設計している。本システムを用いると、学生は教員が配布するプログラムの雛型ファイルをダウンロードし、その雛型ファイルを直接編集し動作確認をしたのち、ボタンを押すだけで編集内容を簡単にアップロードすることができる。

[†] 北陸先端科学技術大学院大学 知識科学研究科
School of Knowledge Science, Japan Advanced Institute of Science and Technology

2. 関連研究

phpSOSS(PHP Simple Online Submission System)⁷⁾は、学生の成果物ファイルを受領する Web ベースのシステムである。オープンソースである PHP や MySQL を使用して構築されている。phpSOSS は提出されたファイルを管理する機構を備えたシンプルなシステムである。EduComponents¹⁾は、教員の手間を軽減するための自動採点機能を備えた Plone CMS 用プラグインである。Python, Haskell, Scheme, Common-Lisp, Prolog のプログラムを自動的に評価しフィードバックする機能を備えている。

本稿に類似したバージョン管理システムを利用した実践として、Reid らによる CVS(Concurrent Versions System)を用いた実践⁵⁾や Glassy による Subversion を用いた実践⁴⁾が報告されている。特に Reid らによる報告には、CVS を用いた成果物管理の長所と短所が実践を踏まえた知見として詳細に述べられている。しかし、どちらの実践もコンピュータサイエンスを専攻する学生が、バージョン管理システムを用いながら実践的なプロジェクト管理手法を体得することが目的の 1 つに置かれている。そのため、学生は既存の CVS や Subversion のコマンドラインツールの使い方をあらかじめ理解する必要があった。Glassy の実践ではリポジトリの作成と教員の初期配布ファイルのインポートを学生に行わせるなど、よりバージョン管理システムへの習熟に主眼を置いている。我々の実践が上記の実践と異なる点は (1) 専用クライアントを準備することにより、コンピュータにあまり慣れていない初学者でも簡単に利用できるようにした (2) Trac (3.2 参照) を利用し、Web ブラウザから成果物ファイルの更新状況を確認できるようにした、の 2 点である。

3. Subversion/Trac

本章では、バージョン管理システム Subversion²⁾ およびプロジェクト管理ツール Trac について述べる。

3.1 Subversion

Subversion²⁾は、CollabNet Inc. が開発したオープンソースのバージョン管理システムである。基本的には CVS の考え方に基いているが、ディレクトリのバージョン管理機能や、テキストファイルとバイナリファイルを統一的に扱う仕組みにより CVS よりもシンプルで使いやすいシステムに改善されている。

Subversion では、リポジトリ (Repository) と呼ばれるデータがバージョン化したファイルを管理している。利用者は (1) リポジトリからバージョン化された

ファイルの作業コピーを作成し (チェックアウト)、(2) 編集作業を行ったあとで (3) 編集後のファイルの状態に作業内容に関する覚え書き (メッセージ) を付け、「リビジョン」としてリポジトリに格納する (コミット)、という 3 つのステップを踏む。一度チェックアウトした後は (1) のチェックアウトの代わりに、作業コピーを最新の状態にする (アップデート) を行い、編集およびコミットを行う。

Subversion を使用して成果物ファイルを管理すると、学生にとって以下の利点がある。

- 学生はプログラムの編集に失敗しうまく動かなかったときに、簡単に過去の状態 (リビジョン) に戻ることができる。そのため、プログラムを修正しながら動作を試してみるといった作業を安心して行える。
- 一旦コミットした情報はリポジトリで管理しているため、作業コピーが万一失われたとしてもコミット済みのデータを簡単に復元できる。
- 学生は作業コピーを複数の環境に置くことができる。例えば、大学の計算機と個人用の計算機の双方に作業コピーを作成しておき、家に帰ったときは個人用の計算機で作業するといった利用ができる。
- 学生は教員が準備した雛型ファイルを簡単にダウンロードし、必要な箇所だけを編集して提出できる。そのため圧縮ファイルを展開したり配置したりする手間が軽減できる。またプログラムの実行に必要なプロジェクトファイル等の設定も軽減できる。
- コミット履歴はリビジョンとしてリポジトリに残っているため、学生は過去の学習経過をふりかえることができる。

また教員側の運営管理上の利点として、編集作業で生じた変更点のみをリポジトリに格納するため、リポジトリのファイルサイズを節約できる点が挙げられる。リポジトリ内でのコピーは「コピーを行った」という情報のみ記録するため、通常のコピーに比べて必要なファイル容量は大幅に縮小できる。

3.2 Trac

Trac³⁾とはスウェーデンにある Edgewall Software が提供している Wiki, Subversion リポジトリブラウザ、バグトラッカ (チケットシステム) 等の機能を備えたプロジェクト管理ツールである。リポジトリブラウザを用いると、学生は Subversion リポジトリに格納されたファイルやコミット履歴を閲覧したり、変更箇所のみを強調表示することができる。システム自体は

Python で作成されている。インタクト株式会社*が日本語化を行っている。

4. SVNLeague

Subversion を利用したプログラミング演習環境の構成を容易にするため、我々はプログラミング学習管理システム SVNLeague を構築した。SVNLeague は、学生用の Subversion クライアント SVN4Lec と、学生アカウントやリポジトリ管理を行う Web システムから構成される。

4.1 システム設計思想

我々は Subversion を利用したプログラミング学習環境において、以下の 2 点を重視した。

- (1) 計算機にあまり慣れていない利用者やプログラミング初学者でも簡単に使用でき、Subversion の機能がもたらす恩恵を受けられるようにすること
- (2) 教員がプログラミング講義を行ううえで、必要となるリポジトリ作成や学生管理等の諸々のタスクを簡潔に行えるようにすること

(1) を考慮すると、学生にインストールや設定といった複雑な操作をさせるのは望ましいことではない。そこで我々はあらかじめ講義用の設定項目を埋め込んだ状態で学生に配布できる Subversion クライアント SVN4Lec を Java アプリケーションとして構築した。Java を選択した理由は同一のファイルが環境を選ばず動作するためである。インストールにおける問題を回避するため、ライブラリや外部アプリケーションに依存せず、ファイル 1 つのみで実行できるよう Java Archive として構築した。また (2) を考慮し教員が手軽に Subversion を利用した講義を実施できるようにするため、Web ベースのリポジトリ/学生管理システム SVNLeague を構築した。

4.2 学生のシステム使用の流れ

説明のため、学生がプログラミング講義における演習を行う際の流れを示す。

学生は、講義を運営する教員から、SVNLeague のアカウント設定のための URL を含むメールを受け取る。学生がその URL を開きパスワードを設定すると、初期パスワードが解除されアカウントが有効になる。

学生が図 1 に示す Web ページから、設定したアカウントでログインすると、図 2 に示す学生用個人ページが開き、講義専用 SVN クライアント SVN4Lec がダウンロードできる。

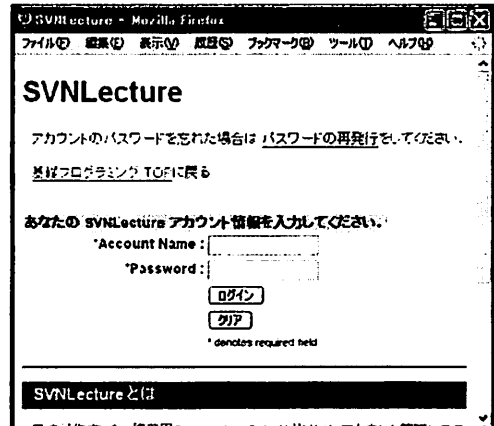


図 1 SVNLeague ログインページ
Fig. 1 SVNLeague WebManager

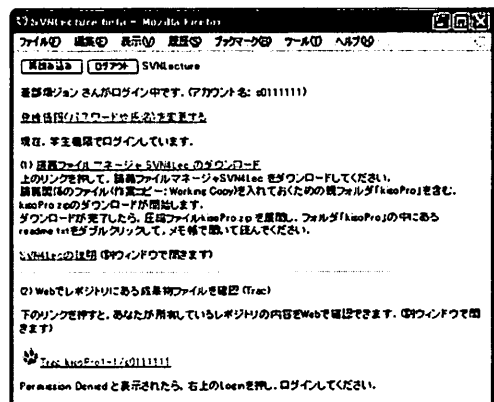


図 2 SVNLeague 学生個人ページ
Fig. 2 SVNLeague: Personal page for student

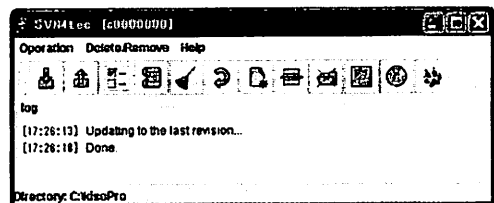


図 3 講義用 Subversion クライアント SVN4Lec
Fig. 3 SVN4Lec: A Subversion client specialized for lecture

学生が単一の Java アーカイブファイルで構成された SVN4Lec.jar をダブルクリックすると、図 3 の画面が現れる。学生がツールバーの一番左のボタン (Update/Checkout) を押し、アカウント名とパスワードを入力すると、リポジトリの学生ディレクトリ内にあ

* <http://www.i-act.co.jp/project/products/products.html>

る初期配布ファイルを取得できる。

学生は初期配布ファイルを編集し課題演習を行う。課題演習が完了したら、SVN4Lec ツールバーの左から2番目のボタンを押す。するとコミット確認画面が表示され、学生がコミットメッセージを入力してボタンを押すと、ファイルの編集内容(差分)が演習成果物としてリポジトリに記録される。

SVN4Lec ツールバーの一番右のボタンを押すと、Web ブラウザが起動し、Trac のページが表示される。図7や図8のように自分の編集履歴や、教員によるコメントを確認することができる。

次回以降、学生は演習を始める前に Update/Checkout ボタンを押し、作業コピーを最新の状態に更新する。これにより、教員が追加で配布した雛型ファイルを取得したり、教員のコメントやアドバイスを作業コピー内のソースコードに反映させたりすることができる。

4.3 教員のシステム利用の流れ

講義を運営する教員は、最初に図1に示す画面から教員アカウントにてログインする。教員アカウントでは、講義用リポジトリを作成したり、そのリポジトリにアクセスできる学生アカウントを追加・編集したりできる。学生アカウントの追加、一時停止および削除は、学生アカウント名とメールアドレスを図4に示す画面に入力することで行う。SVN4Lec では、基本的に1つの講義で1つの講義用リポジトリを使用するというシンプルな構成方針をとっている。リポジトリのトップディレクトリに学生アカウント名と同一名のディレクトリを作成すると、学生はそのディレクトリ以下のみアクセスできるようになる。図5に、具体的なリポジトリ構成例を示す。ちなみに base は、教員による配布ファイル準備用のディレクトリである。

次に教員は SVN4Lec.jar 内に含まれている設定ファイルに、講義用リポジトリの URL と、必要に応じて講義サポート用 Web ページの URL を記入する。そのファイルを任意の Web サーバ上に配置し、Trac の Wiki ページに、その講義用 SVN4Lec.jar への URL を記載し、ダウンロード可能にする。

学生アカウント、学生用ディレクトリ、講義用 SVN4Lec.jar へのリンクを準備したら、学生にパスワード設定依頼メールを送信し、アカウント設定を依頼する。

学生が演習を行った結果をコミットすると、誰がいつ、どんなメッセージを付与してコミットしたかの情報が教員宛てにメールで送信される。教員はメールをクリックし Trac のリポジトリブラウザを開くことで学生の作業内容(差分)を確認することができる。ま

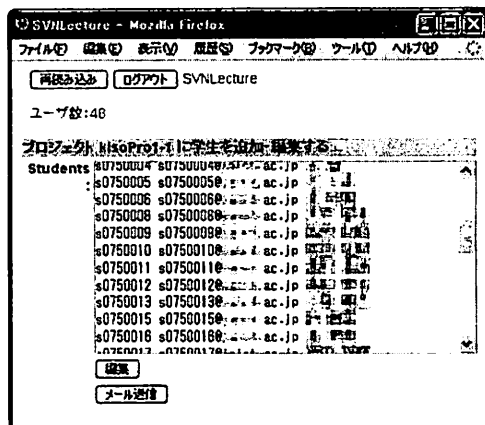


図4 SVN4Lec: 学生管理画面
Fig.4 SVN4Lec: Manage Students

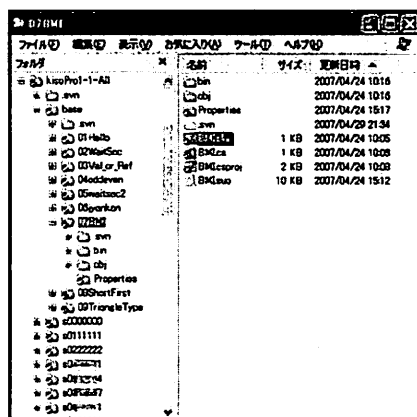


図5 リポジトリ構成
Fig.5 Repository Structure

た、教員用アカウントでリポジトリの作業コピーを取得し、すべての学生の作業内容を確認することができる。たとえば演習課題についてのコメントをソースコード内に直接記載しコミットすることも可能である。

しかし教員が1つ1つのソースコードに個別にコメントを付与するのは手間がかかる。そこでコメントを一括してフィードバックする簡単なツールを構成した。図6に示すような課題ファイルのパスと、演習課題のスコアやメッセージを学生個別に入力した CSV ファイルをツールに読み込ませると、メッセージと点数を各学生のソースコード(作業コピー)に書き込み、コミットすることができる。学生は Trac の画面(図7, 図8参照) やソースコードからメッセージを確認できる。

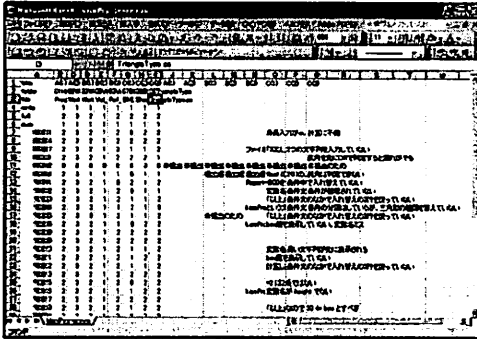


図 6 採点用 CSV ファイル

Fig. 6 CSV file for evaluation and comment feedback

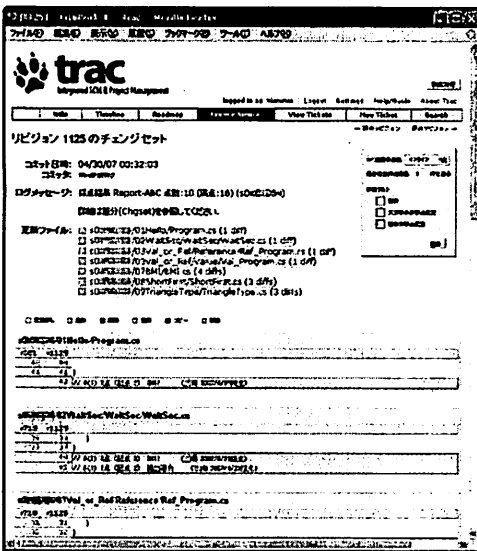


図 7 リビジョン表示

Fig. 7 Revision View

5. 実 装

SVNLecture および SVN4Lec について、上記で説明しなかった実装の詳細について述べる。

5.1 リポジトリ/学生管理 Web システム SVN-Lecture

リポジトリや学生を管理する Web システム SVN-Lecture は、PHP と MySQL を用いて実装した。4.3 で述べたように、SVNLecture では、学生毎に別々のリポジトリを作成するのではなく、1つのリポジトリ内に学生用フォルダと教材配布用 (base) フォルダを置く設計としている。このようにすることでリビジョン内コピーとして教材が配布できるためリポジトリの

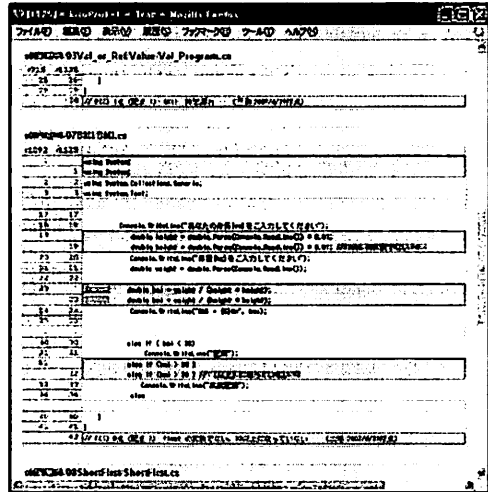


図 8 講師の採点結果とコメント

Fig. 8 Scores and Comments by Lecturer

実データ領域を節約できる。また教員が学生の成果物を一括してダウンロードする際にも有利となる。

上記のように単一リポジトリ構成を採用した場合、学生フォルダ毎のアクセス制御が必要となる。そこで、SVNLecture ではアクセス制御のための設定ファイルを Web インタフェースから動的に再構成する機構を設けた。教員がリポジトリを作成したのち、ユーザを追加または編集すると、PHP スクリプトが図 9 のリポジトリアクセス制御ファイル (ACL) と Apache 用認証ファイル .htpasswd を自動的に更新する。kisoPro1-1 はリポジトリ名を表す。リポジトリへのアクセスを HTTP で行くと、まず Apache の設定ファイル (図 10) 内の AuthUserFile ディレクティブで指定したファイル (.htpasswd) で認証を行う。認証後、AuthzSVNAccessFile ディレクティブで指定した ACL により、認証ユーザのリポジトリへのアクセスを制限することが可能となる。Trac についてもこれらの設定ファイルを参照させることで、リポジトリと同様の認証を行うことができる。ちなみに学生が Web インタフェースからパスワードを変更した場合は、Apache 用認証ファイルのみを更新する。この仕組みにより、教員が行う学生管理の手間を大幅に軽減できる。また学生も自由にパスワードを変更できる。

5.2 SVN4Lec

SVN4Lec は Java アプリケーションとして実装した。Subversion の機能は svnkit⁶⁾ ライブラリを使用して構築した。

SVN4Lec は起動時に、自身の JAR ファイル内部

```
<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>
```

図 10 Apache の設定ファイル (httpd.conf) の一部
Fig. 10 Excerpt from Apache Configuration file (httpd.conf)

```
[kisoPro1-1:/]
teacher = rw
* =
[kisoPro1-1:/base]
teacher = rw
* = r
[kisoPro1-1:/s00000000]
s0000000 = rw
[kisoPro1-1:/s0111111]
s0111111 = rw
[kisoPro1-1:/s0222222]
s0222222 = rw
.....
```

図 9 リポジトリアクセス制御ファイル (svnaccess)
Fig. 9 ACL for Repository (svnaccess)

の設定ファイル config.txt を参照し、(1) チェックアウト先の URL、(2) ヘルプボタンを押したときに表示する URL、(3) システム自動アップデート先 URL などの講義固有の情報を読み込む。実際のチェックアウト時には、学生が入力したアカウント名を付加したものをチェックアウト先 URL とすることにより、学生は透過的にリポジトリ内の自分のディレクトリ以下のみをチェックアウトできる。

SVN4Lec には、通常のクライアントと同様、過去のリビジョンを一覧表示し、必要に応じて戻る (Update to Revision) 機能を備えている。このほか、パスワードキャッシュを消す機能や、.svn フォルダを再帰的に削除することにより SVN 管理下から外す機能、Trac ページを開く機能を備えている。

6. 実 践

2007 年 4 月～5 月にプログラミング初学者に C# を通じてプログラミングの基礎を導入する講義を実施し、その際の演習に SVNLecture を運用した。受講者は 40 名で、文系の学部を卒業した大学院修士課程の 1 年次が中心である。演習では、プログラミング統合開発環境である Visual Studio .NET 2005 (VS2005)

を使用して、基本的なコンソールアプリケーションを作成してもらった。VS2005 を利用した理由として、単純な構文ミスによる文法エラーを修正しやすくなり、変数名やメソッド名の補完機能や自動インデント機能により、プログラミング作業以外の負荷を軽減する目的があった。しかしプログラミング初学者にとっては、VS2005 は上記の点で便利である反面、新規プロジェクト作成や設定などの付加的な作業は学習の負担となると考えられる。そこで講義では、VS2005 のプロジェクトを教員が初期ファイルとしてあらかじめ作成しておき、それを学生が取得して編集し提出するという形式をとった。個々の学生への初期ファイル配布は、svn copy コマンドを起動するシェルスクリプトを用いた。

演習方法の説明時には、4.2 で述べた SVN4Lec の Update と Commit の方法、過去の状態に戻す方法 (Revert, Update to Revision)、Trac での提出物確認の手順と、VS2005 での編集と実行方法について 15 分程度の時間を割いて説明した。

SVN4Lec はリポジトリに存在しないファイルをコミット時に追加する機能を備えているが、その機能や説明はあえて行わなかった。その理由は、学生はチェックアウトしたファイルの修正のみで課題を達成できるよう演習を設計しており、また提出ファイルの名前を統一したほうが教員が確認しやすいというねらいがあったためである。実際にはファイルやフォルダを追加で登録した学生もいた。ちなみに学生が不要なファイルを追加することを防ぐため、SVN4Lec では VS2005 がビルド時および実行時に生成するファイル (設定ファイルや実行バイナリ) は追加できない仕様とした。

6.1 利点と問題点

実践で得られた利点を以下に挙げる。

- 学生は特に問題なく、初期ファイルをダウンロードし、課題を提出することができた。

- 学生は課題を簡単に送信できるため、気軽に再提出することができた。
- 教員は Trac の差分表示を活用することで、課題チェックを効果的に行い採点の負荷を軽減することができた。例えば課題として、あるプログラムを入力し動作確認した時点で一旦コミットしてもらい、その後細かな修正を加えて挙動を変更し、再度コミットしてもらう。そうすると、2 度目のコミット時に細かな修正の内容を差分表示により一目で確認することができる。
- 必要があれば直接ソースコードを編集し、コメントでアドバイスできる。
- 必要があれば教員がビルドして動作確認し、問題を発見できる。
- TA のアカウントを作成すれば TA も同時に成果物ファイルにアクセスできるようになる。(ちなみに今回の実践では TA は配置していない)

逆に、実践によって明らかとなった問題点を以下に掲げる。

- Subversion の制約として、最新版ではないリビジョンからの変更を直接コミットすることはできない(明示的に、最新版にマージする必要がある)。このため、過去の状態に戻ったところから修正し、コミットしようとしたときにエラーが発生する。
- 教員がソースコードにコメントを書き込んだ場合、学生が Update を確実に実施しないと、上記の問題が発生する。
- SVN4Lec のエラーメッセージが不親切であり、学生が自分で問題を解決しづらい。
- 処理能力が貧弱な計算機環境だと、フォルダ数が多い場合に更新(Update)処理にやや時間がかかる。

初期の頃には学生演習用計算機(もしくは端末)が接続する環境(Windows サーバ)により、Java がインストールされていなかったり、VS2003 しか利用できない環境があったため、JAR ファイルを起動しようとして ZIP 展開してしまったり、VS2005 のプロジェクトが開けない、などの多少の混乱があった。しかし、演習環境に対する注意を促した後は、大きなトラブルは発生しなかった。受講者は SVN4Lec と VS2005 という 2 つの独立したシステムを切り替えながら課題を遂行する必要があったが、ファイル取得・課題提出ツールと開発ツールという概念の切り分けは明解で理解しやすいものであったため混乱はほとんどなかった。ちなみに VS2005 のエディタには編集集中ファイルが変更されたことを検知し、再読み込みを行うかどうかをユーザに尋ねる機能があり、リビジョン切り替え時

に VS2005 のエディタを終了する必要がなかったことも幸いした。また今回は大学院生を対象とした比較的少人数の講義に適用したことにより、導入の混乱を軽減できた可能性もある。そのため大学生を対象とした多人数講義といった、より厳しい条件での適用を行い検証する必要がある。

今後 SVN4Lec クライアントの改良として、(1) 過去のリビジョンに戻した後で行った修正を最新リビジョンとして活かすためのマージ機能 (2) 特定の課題サブフォルダ内のファイルのみを過去に戻す機能、を追加する予定である。また SVN4Lecture についてグループ作業に対応可能な柔軟なパーミッション設定を行えるようにしていきたいと考えている。

6.2 議 論

講義終了時点での全学生(40 名)の成果物ファイルの最終リビジョンのみをチェックアウトした、作業コピーの総サイズは約 69MB であった。それに対し、実際にサーバが管理しているリポジトリのファイルサイズは約 13MB 程度であり、多くの学生の作業ファイルを効率的に管理できていたことが確認できた。ちなみにコミット数は全部で 2,065 回であったが、そのうちのおよそ半数にあたる 1,020 コミットは教員の教材配布(コピー)やコメント送信により発生したリビジョンである。

また、受講者個別の操作回数を調査するため、講義期間の Apache のアクセスログ解析を行った。Apache のアクセスログには認証を通過したアカウント名が記されているため、いつ、どの受講者がどんな操作を行ったかをおおまかに推測することができる。図 1 に、受講者 40 名の操作回数の平均値と標準偏差を示す。

表 1 受講者の操作回数
Table 1 Operation times of students

	平均	標準偏差	最大	最小
更新・チェックアウト	65.8	65.9	352	19
コミット成功	17.2	6.7	41	7
コミット失敗	8.2	5.9	24	0

コミット失敗の回数が多いのは、過去のリビジョンからの修正(マージ)をプログラムがサポートしていなかったことが最大の要因として挙げられる。しかし、逆に考えれば、ある程度多くの受講者が「過去のリビジョンに戻る機能」を利用していたことが結果から推測できる。

コース終了後に大学が実施する匿名の授業評価アンケート内にて、Subversion を用いた講義および SVN4Lec に関する表 2 の追加項目を設定し、学生

表 2 アンケート項目 (質問項目ラベルは図 11 と対応)
Table 2 Questionnaire items.

項目	質問項目ラベル
自分のプログラムソースコードを過去の状態に戻せることは便利だと思いますか？	RB
編集差分が Web ページで見られることは便利だと思いますか？	Diff
教材をダウンロードしたり、レポートを提出したりするのに SVN4Lec は効果的だと思いますか？	Effect

の意見を集めた。その結果 (図 11) によると、質問項目間の差は少なく、全体的にバージョン管理ツールを使った演習課題管理を支持するグループと、その必要性を感じていないグループの 2 つに分かれていた。今後、SVN4Lec を使用しない他の講義との比較を行い、効果を検証していきたいと考えている。

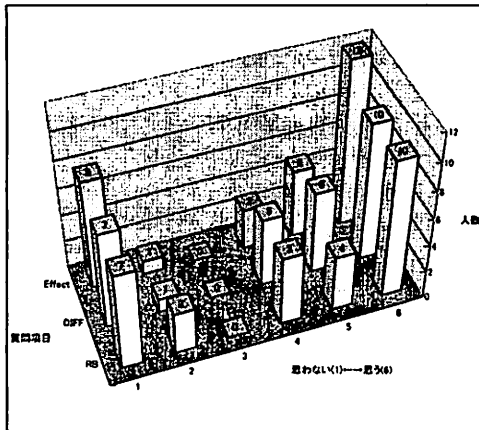


図 11 アンケート結果
Fig. 11 Questionnaire Result

7. まとめと今後の課題

バージョン管理システム Subversion を利用した学習管理システム SVN4Lec と、専用クライアント SVN4Lec を構築した。特にクライアントの改良により (1) 起動後アカウント情報を入力するだけでチェックアウトできる機能、(2) ツールボタンによるシンプルな操作性 (3) ワンボタンによるリポジトリブラウザ起動機能をインストール・設定不要な機構によって提供した。これらの改良により、プログラミング初学者の演習にバージョン管理システムを導入・適用しやすくなった。

差分管理によって、学生がやり直したり過去への状態に戻りやすくなるため、通常的环境に比べて、学生の課題遂行におけるファイル管理の意識が変容する可能性もある。しかし現状のシステムでは学習者の効果よりも教員の運営上の利点が中心となっている感否

めない。教員にとっての運用の効率化が学習者へのきめ細やかな対応に還元できるという考え方もあるが、差分管理をどう活用すればプログラミング学習の本質を変革できるかは今後の課題である。

差分管理システム自体はプログラミングに限らず、卒業論文や Web 作成などの共同作業・進捗管理システムとしても有効である。そのため本システムの適用範囲は広い。SVN4Lec の PHP スクリプトおよび SVN4Lec の Java ソースコードは <http://css.jaist.ac.jp/~miuramo/svnlec/> にてオープンソースとして公開している。

参考文献

- Amelung, M., Piotrowski, M. and Rösner, D.: EduComponents: Experiences in E-Assessment in Computer Science Education, *ITICSE '06: Proceedings of the 11th annual SIGCSE conference on Innovation and technology in computer science education*, ACM Press, pp.88-92 (2006).
- Collins-Sussman, B., Fitzpatrick, B.W. and Pilato, C. M.: *Version Control with Subversion*, O'Reilly & Associates Inc. (2004). (邦訳版は「Subversion によるバージョン管理」<http://svnbook.red-bean.com/index.ja.html> から閲覧可)。
- Edgewall Software: The Trac Project. <http://trac.edgewall.org/>.
- Glassy, L.: Using Version Control to Observe Student Software Development Processes, *Journal of Computing Sciences in Colleges*, Vol.21, No.3, pp.99-106 (2006).
- Reid, K.L. and Wilson, G.V.: Learning by Doing: Introducing Version Control as a Way to Manage Student Assignments, *SIGCSE '05: Proceedings of the 36th SIGCSE technical symposium on Computer science education*, ACM Press, pp.272-276 (2005).
- TMate Software: SVNKit - Subversion for Java. <http://svnkit.com/>.
- Wolff, D.: A Web-based Tool for Managing the Submission of Student Work, *Journal of Computing Sciences in Colleges*, Vol.20, No.2, pp. 144-153 (2004).