

学習履歴の収集・参照機能を備えた プログラミング講義支援システム

草野 裕紀 山本 昭成 平野 洋行 市村 哲

東京工科大学

1. はじめに

近年、情報系大学ではプログラミングは必須科目となっている。しかし、初めて習う学生にとっては苦手とする場合が多く、中には他人のソースを丸写ししたものをそのまま課題として提出する人がおり、講師が学生の理解度をリアルタイムかつ正確に把握することができていないと言える。

そこで、演習内での学生の作業内容をログとして収集し、エラー内容や学生の苦手な箇所を把握し、さらにリアルタイムに助言を与えることが可能なプログラミング講義支援システムを提案し構築する。

2. 背景と問題点

現在のプログラミング講義は、スライドやプリントを使用して行われるものが一般的である。講師は学生にサンプルプログラムを見せながら解説を行ない、講義の最後に演習問題を解かせる。

しかし、演習問題の出題形式は問題文と実行例のみを載せるものが主流であり、問題の主要部分以外に時間をかけてしまい、効率的な学習ができていない学生が多いという問題がある。例えば、for 文を学習する授業の演習において、if 文でつまづいていると、for 文を理解するための時間を無駄にしてしまう。限られた演習時間で効率的に解答させるには、講師が学生の苦手な箇所、躓きやすい箇所をリアルタイムかつ正確に把握し、問題解決のヒントを提供する必要があると考えた。過去の支援システムに、講師側が学生に対しリアルタイムでアンケートを作成し、理解状況を回答させるというものがあるが、回答の内容はあくまで学生の自己申告であり、客観的に見て理解できているかは知るこ

とができない。正確な情報を得るためには、学生の作業記録をログとして取得し、分析する必要があると考えた。

3. 提案

学習履歴の収集・参照機能を備えたプログラミング講義支援システムを提案する。このシステムは講師が作成した問題に対し、学生がエディタを用いて解答する形となる。エディタにはプログラムのコンパイル・実行機能があり、ここで発生したエラーの種類や修正したソースの差分が学生ごとにログとして収集され、サーバに保存される。講師はログを閲覧することで、プログラミングの苦手な学生や学生が躓きやすい箇所を知ることができる。

3.1. システム概要

本システムの概要図を図1に示し、使用手順について述べる。

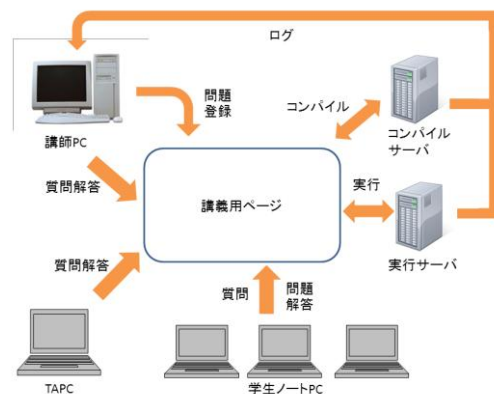


図1 システムの概要図

まず、講師は問題を作成する。エディタを用いて問題文、課題番号、ファイル名を入力し、講義用ページに登録する。次に、学生は講義ページ上で課題を選択し解答を行う。エディタ上でプログラムを作成し、コンパイルボタンを押すとコンパイルサーバ上でコンパイルが行われ、結果がシェルに出力される。さらに実行ボタン

Programming lecture support system equipped with gathering and reference function of a study history

Yuki Kusano

Akinari Yamamoto

Hiroyuki Hirano

Satoshi Ichimura

Tokyo University of Technology (zz)

を押すと、実行サーバ上でプログラムが実行され、実行結果がシェルに出力される。この時、学生が行ったコンパイル回数、実行回数、発生したエラー個数、エラーの内容はログとして収集される。発生したログの一覧は講師の PC で閲覧することができる。発生回数の多いエラーについて、講師は掲示板を使い、リアルタイムでヒントを与えることができる。

4. 実装

4.1. 講義サイト

講師は課題登録画面から問題の情報を入力し、登録する。学生は登録フォームからユーザの新規登録を行うことで、ユーザ用の講義ディレクトリが自動生成される。受講する講義のページから課題を選択すると、プログラミングエディタが起動し、問題が表示される。

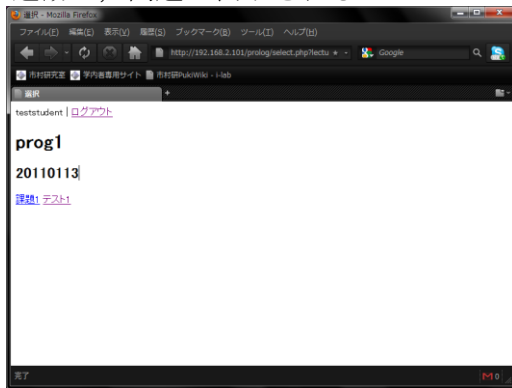


図2 講義サイト (課題選択画面)

4.2. プログラミングエディタ

講義サイトで課題を選択することで、プログラミングエディタが起動する。エディタはコーディングを行うテキストエリア、実行結果を表示するシェル、質問投稿フォーム、各種ボタンで構成されている。

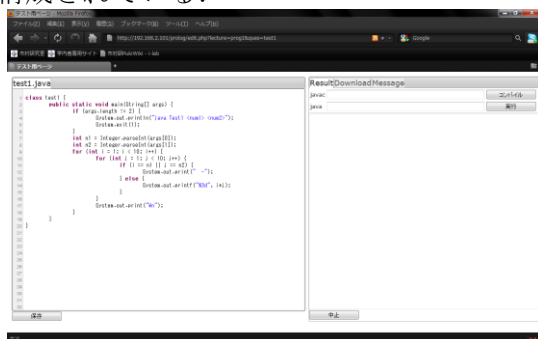


図3 プログラミングエディタ

4.3. 掲示板

エディタの入力フォームから投稿された質問が表示される。質問が投稿されるたびに新たに

テキストエリアが生成され、対応したテキストエリアに書き込むことによって返答を行うことができる。返答は講師、TA、学生が行うことができる。

4.4. コンパイルサーバ

問題解答画面 CGI プログラムを呼び出して処理を行う。以下に処理の流れを示す。

- 1). 問題解答画面から CGI プログラムを引数付きで呼び出す。
- 2). system コマンドによりファイルをコンパイルする。この時出力されるログをテキスト形式で保存し、データベースに登録する。
- 3). 登録されたログを講義用ページに表示する。

4.5. 実行サーバ

コンパイルサーバの処理と同様に CGI プログラムにて処理を行う。以下に処理の流れを示す。

- 1). 問題解答画面から CGI プログラムを引数付きで呼び出す。
- 2). System コマンドによりファイルを実行する。実行時のログをテキスト形式で保存し、データベースに登録する。
- 3). 登録されたログを講義用ページに表示する。

4.6. データベース

MySQL サーバを利用し、学生と収集するログのデータベースを作成する。コンパイルや実行ログなどのデータを登録する。

5. まとめ

現在のエディタには実行中のプログラムを強制終了させる機能がまだ実装されておらず、プログラムが暴走した際の対策ができていないと言える。この機能を実現させるにはサーバを Linux で構築する必要があるため、今後はサーバの移行と Linux 上での動作確認を行っていきたい。また、実際に講師、学生に使用してもらい、ログから取得できるデータの信頼性や本システムの改善点について考察していきたい。

参考文献

- 1) 長瀧寛之, 永井孝幸, 都倉信樹: 「授業内の学生の反応を記録・解析するシステムの運用報告」, 情報処理学会研究報告. コンピュータと教育研究会報告 2003(13), 47-54, 2003-02-07
- 2) 日比野 祐介・梶 勇太: 「学生と TA 間でトラブル状況を共有できるプログラミング講義支援」 市村研究室 2009