情報処理学会論文誌 Vol.54 No.12 2518–2527 (Dec. 2013)

推薦論文

プログラミング演習授業における学習状況把握支援の試み

市村 哲[1,a)] 梶並 知記[2] 平野 洋行[1]

受付日 2013年2月12日, 採録日 2013年9月13日

概要：プログラミング授業は大学や専門学校等の情報系学科では必須科目となっているが，大学で初めて習う学生の多くはプログラミングに対する苦手意識が強い．そこで本研究では，プログラミング初学者がかかえる問題を早期に発見できるプログラミング学習支援システムを構築し，実際の演習授業で運用を試みた．つまずいている学生の早期発見と，多くの学生が共通にかかえる問題の発見を可能にすることが本研究の目的である．本研究の目的を達成するために，Web ブラウザ上で動作するプログラミング演習支援システムを構築し，学生の操作ログ・エラーログを収集して解析できるようにした．評価実験と改良を繰り返し行い，結果として，学生の学習状況を把握しやすくなったことが分かった．

キーワード：教育工学，プログラミング学習支援，Web アプリケーション，Web IDE サービス

A Trial to Support Understanding a Programming Exercise Class

Satoshi Ichimura[1,a)] Tomoki Kajinami[2] Hiroyuki Hirano[1]

Received: February 12, 2013, Accepted: September 13, 2013

Abstract: The programming exercise is a required subject in the department of information systems in colleges and universities. Most of the students are programming beginners right after they entered university. Our goals are to enable to detect the students who have stumbled, and know problems that many students are facing in common. For this purpose, we built programming exercise support system that runs on a Web browser. The system is able to collect and analyze students' error logs. As a result of the repeated experiments and improvements, the system enabled lecturers to easily grasp the situation of the programming class.

Keywords: educational technology, support of programming education, Web application, Web IDE service

1. はじめに

プログラミング授業は大学や専門学校等の情報系学科では必須科目となっていることが多い．授業は，教科書や電子スライドを使用する座学と演習を組み合わせて進められることが一般的である．しかしながら，大学で初めて習う学生の多くはプログラミングに対する苦手意識が強いというのが現状である．また，必須科目であるプログラミング講義・演習を履修する学生の数は一般的に多く，講師が学生ひとりひとりに対応することが難しいという問題がある．このため，プログラミング講義・演習クラスには，講師のほかにティーチングアシスタント（以後，「TA」と略す）と呼ばれる補助教授者（多くの場合，大学院生）が付くことがあり，講師と TA が協力して学生のサポートを行う例が少なくない．TA は学生が手を挙げて質問をした場合に，その学生の座席に行って必要なアドバイスを与える役割を担う．

しかしながら，分からないのに手を挙げない，何を質問すればよいかが分からないという学生に対しては対応できないという問題や，講師が想定しないポイントで多くの学

[1] 東京工科大学コンピュータサイエンス学部
School of Computer Science, Tokyo University of Technology, Hachiouji, Tokyo 192–0982, Japan
[2] 神奈川工科大学情報学部情報工学科
Faculty of Information Technology, Kanagawa Institute of Technology, Atsugi, Kanagawa 243–0292, Japan
[a)] ichimura@stf.teu.ac.jp

本稿の内容は 2012 年 7 月のマルチメディア，分散，協調とモバイル（DICOMO2012）シンポジウム 2012 にて報告され，グループウェアとネットワークサービス研究会主査により情報処理学会論文誌ジャーナルへの掲載が推薦された論文である．

© 2013 Information Processing Society of Japan

生がつまずいている場合にそのことを発見するのにきわめて時間がかかるという問題がある．そこで今回，プログラミング初学者（プログラミング学習を始めて1〜2年目の学生）がかかえる問題を早期に発見できるプログラミング学習支援システムを構築し，実際の演習授業で運用を試みた．本研究の目的は以下のとおりである．

(1) つまずいている学生の早期発見
(2) 多くの学生が共通にかかえる問題を発見

また上記本研究の目的を達成するために，Webブラウザ上で動作するプログラミング演習支援システムを構築し，学生の操作ログ・エラーログを収集して解析できるようにした．

評価実験と改良を繰り返し行い，結果として，講師やTAが，学生の状況を把握しやすくなることが示唆された．

2. 背景と関連研究

本研究が対象とする学生はプログラミング学習を始めて1〜2年目の初学者である．このような初学者を対象としたプログラミング講義では，講義の前にプリントやスライド形式（パワーポイント等）の講義資料を配布したり，教科書やFAQを参照させたりして，それらをもとに講義を進めていく形式が一般的である．また，講義資料には新しく習うプログラミングの文法や提出する課題の内容が書かれており，その資料に沿って講師はプログラムの説明をし，学生に課題を行わせるという形式が主流である．学生に行わせる課題は，長くても数十行程度の短いソースコードを題材とすることが多い．しかしながら，大学で初めて習う学生の多くは課題を解くのに苦労しており，プログラミングに対する苦手意識が強いというのが現状である．

従来，プログラミング講義を支援する研究として，プログラムの処理の1つ1つを部品化し，それらを組み合わせることで視覚的にプログラムを作成できるシステム [1] や，学生同士で協調してプログラミングを行わせることで，新しいテクニックをお互いに学ばせることができるシステム [2]，失敗学に基づいた内省促進によるプログラミング教育支援手法 [3]，入力支援および実行状況の表示等を備えた入門教育用プログラミング環境 [4]，WEB上でプログラムの穴埋め問題を行いコンパイルが可能なシステム [5]，バーチャルに要求から納品までソフトウェアの開発が行えるシステム [6] が提案されている．しかし，これらのシステムは，プログラミングの簡略化や，学生間のコミュニケーション，振り返り学習を主に支援するものであり，本稿の提案とは方向性が異なっている．また関連研究にはプログラミング課題提出後の分析ツールが多いが，授業中に問題をかかえる学生をリアルタイムに発見するための支援ツールはほとんど見当たらない．

著者らは過去においてプログラムの作成過程を動画として自動作成できるソフトウェア [7], [8] を開発した．そして，TAが学生にエラーの解決方法を教えている様子を記録しておき，それを共有すれば上記の問題を緩和できるのではないかと考え，TAが学生にエラーの解決方法を教えている状況を動画で記録し，学生とTA間で共有することができるプログラミング演習支援システムを構築した [9]．システムは，TAが学生のトラブルを解決している過程を記録して資料化するクライアントツールと，この資料を閲覧できるようにする資料閲覧サイトから構成されている．

プログラミング学習を始めて1〜2年目の学生に対して評価実験を実施した結果，前記クライアントツールと資料閲覧サイトを利用した方が学生単独でエラー解決方法を学習しやすく，また，学生が自ら積極的にエラー解決を行おうとする可能性が高まることが示唆された．しかしながら，分からないのに手を挙げない学生に対しては対応できないという問題や，講師が想定しないポイントで多くの学生がつまずいている場合にそのことを発見するのに時間がかかるという問題が残った．

過去の研究において明らかとなった本研究の課題は以下のとおりである．

課題1 つまずいている学生の早期発見が困難
課題2 多くの学生が共通にかかえる問題の発見が困難

課題1について，手が挙がるまで誰がどこでつまずいているか分からないことが問題である．分からないのに手を挙げない，何を質問すればよいかが分からないという学生は，分からないまま授業を終えてしまうことが多々あり，プログラミングに苦手意識を持つようになる．課題2について，学生の質問に個別対応したTAから話を収集するまで授業の問題点が分からないことが問題である．さらに，個別対応したTAから情報を収集することは実際には難しくあまり行われていないのが現状であり，多くの学生がつまずいているポイントに気づくことが困難である．

したがって，本研究の目的はこれらの課題を解決することであり，つまずいている学生を早期発見すること，および，多くの学生が共通にかかえる問題を発見することである．この目的に基づき本研究では，Webブラウザ上で動作するプログラミング演習支援システムを構築し，学生の操作ログ・エラーログをサーバ側で収集して解析し，講師やTAに分かりやすく提示できるようにすることを目指した．

類似のシステムとしては，近年登場しはじめている，Webブラウザ上で動作するプログラム開発環境（Web IDEサービス）がある．たとえば，Eclipse開発チームが構築しているEclipse Orion [10] や，Mozilla LabsのSky writer [11] 等があげられる．また，Ideone.com [12] は，Webブラウザに入力したプログラムをサーバ上でコンパイルおよび実行し，コンパイル結果および実行結果をSOAP形式で返

© 2013 Information Processing Society of Japan

すサービスであり，標準出力に出力された文字列を実行結果として返すほか，入力データをあらかじめ登録しておけば標準入力を要求するプログラムの実行も可能である．特に，手軽にプログラムを作って動作を確認したり，いつでもどこでもプログラミングを勉強したいというような場合に Web IDE は便利である．ただし，これら Web IDE サービスは，開発者が独力でプログラム開発をするためのツールであり，操作ログ・エラーログを収集して解析する機能は有していないため本研究の課題を解決することはできない．

また，学生がプログラミング学習を自主的に行える E ラーニングシステムが提案されている [13]．学生は Web を使い自分の課題提出状況を確認したり，講師に質問や理解状況を送信したりできる．ただし，操作ログ・エラーログを収集して解析する機能は有していないため本研究の課題を解決することはできない．

3. 事前実験

本研究は，Web ブラウザ上で動作するプログラミング演習支援システムを構築し，学生の操作ログ・エラーログをサーバ側で収集して講師や TA に提示するシステムの構築を目的としている．Web ブラウザ上で動作する Web IDE のような演習環境を提供することで，従来の IDE のようなインストールの手間や，プラットフォームの違いによる動作の違いの問題を避けることができるという利点がある．

著者らは，本システムの構築に先立ち，事前に検証用プロトタイプシステムを実装し，実際の授業に試験的に投入して具体的に何が本質的な課題となるのかを事前実験によって調査すべきと考えた．そして，この事前実験を実施するための準備として，Web ブラウザ上で動作し，学生の操作ログ・エラーログをサーバに収集できる検証用プロトタイプシステムを構築した．

3.1 検証用プロトタイプ

検証用プロトタイプの実装について述べる．プロトタイプシステムが備える機能は以下のとおりである．

機能 1 ソースコードを編集，コンパイル，実行できるエディタ機能

機能 2 コンパイルおよび実行結果ログの収集機能

機能 1（エディタ機能）について述べる．学生には Web ブラウザ上で動作するプログラミングエディタを提供した．JavaScript で記述された Web ベースのソースコードエディタ CodeMirror [14]（オープンソースソフトウェア）を改造し，Web ブラウザからコンパイルやプログラムの実行ができるようにした．また，Web ブラウザ上の当該エディタ機能と Web サーバとは，jQuery（Web サーバと Web ブラウザ間の非同期通信をシンプルなコード記述で可能とする JavaScript ライブラリ [15]）を用いて非同期通信するように変更した．

CodeMirror には，最近の IDE には機能的に及ばないものの，ソースコード中の予約語やコメントを色分表示，“ や (を自動補完，改行時の自動インデント整形，等のプログラミング支援機能が備わっており，新たなソースコードエディタを一から実装することなく，独自の Web IDE システムを構築できるという利点がある．Java，C，C#，PHP 等の複数の言語を扱えるが，本研究は Java プログラミング演習授業を対象としているため，Java ソースコードを編集するためのエディタとして用いた．

図 1 に示すように，左側がソースコード入力枠，右側がコマンドプロンプト枠（右側はタブ操作によって，掲示板枠，ヒント表示枠に切替え可）である．コマンドプロンプト枠には，コンパイルエラーや，実行結果が，PC の通常のコマンドプロンプトに表示されるのと同じ様式で表示される．なお，コマンドプロンプトにコンパイル命令や実行命令を入力する場合は，コマンドプロンプト枠の上部のテキスト枠に入力するようになっている．

ソースコードのコンパイル，および，プログラムの実行は，Web サーバの CGI の仕組みによって行われる．加えて，Web ブラウザと Web サーバ間の通信は jQuery により非同期で行われ，コンパイルやプログラム実行を行っても（「コンパイル」ボタン，「実行」ボタンを押しても），Web ページの遷移は発生しない．また，プログラムの実行時に無限ループに入る等の問題に備え，CGI プログラムを強制終了するための「中止」ボタンを Web 画面上に設けた．

なお，実際のプログラム演習に用いたソースコード（for 文，System.out.print 文を含む 10 行のソースコード）のコンパイル操作を，本システムの Web ブラウザから実行した場合にかかった時間は平均 1,155 ms であった．一方本システムを用いず，Web サーバのコマンドプロンプトから同じソースコードをコンパイルした場合にかかった時間は平均 1,127 ms であった．このことから，本システムの Web ブラウザと Web サーバの通信やデータベース処理が，パフォーマンスを大きく低下させる原因とはならないと判断した．また，プログラム演習に用いる程度のソースコードであれば，10 名程度がいっせいに Java コンパイラでコンパイル操作を実行した場合でもサーバ（LinuxOS）の動作に問題が生じないことを実際に試して確認した．

学生が本システムを利用する際には自分の ID とパスワードで本システムにログインするが，ログインすると本日の課題が一覧表示されており，課題を選択すると，図 1 のエディタが開く．このとき，課題によってはソースコードの一部（課題部分が欠けているソースコード）がソースコード入力枠に自動的に表示される．

© 2013 Information Processing Society of Japan

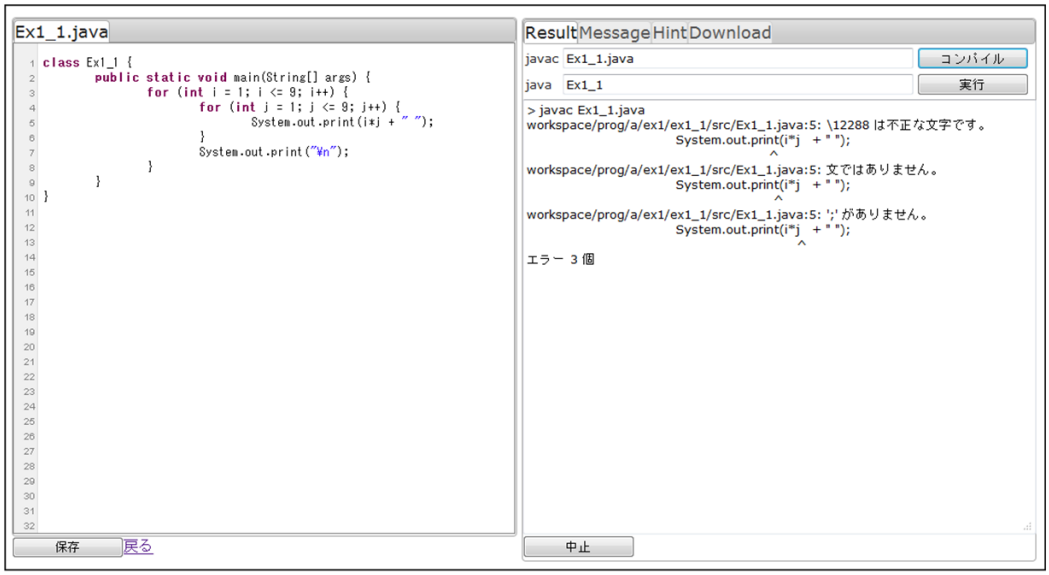

図 1 Web ブラウザ上のエディタ
Fig. 1 Editor on Web browser.

機能 2（ログ収集機能）について述べる．Java コンパイラが出力したエラーメッセージや，学生が作成したプログラムの実行結果は，Web サーバによって取得され，データベースに記録される．Java コンパイラが出力したエラーメッセージには，エラー個数，エラー名，エラー発生行，エラー発生箇所，エラー原因解説文が含まれる．一方，プログラムの実行結果は，プログラムが標準出力に出力した文字列である．

なお，今回の実装では Java 言語を対象としたが，コマンドラインで実行できるコンパイラを備える言語であれば基本的に CGI プログラムとして Web サーバ上で動作するため，わずかな変更で C，C++，C#，PHP といった他の言語を提案システムでサポート対象することが可能である．

また，講師や TA が本システムを利用する際には自分の ID とパスワードで本システムにログインし，管理画面（ログ閲覧画面）で収集したログを閲覧することができる．

3.2 事前実験

構築した検証用プロトタイプを使用してプログラミング演習を行う模擬授業を 2 回行った．事前実験という位置づけである．1 回目は，2011 年 3 月に実施し，情報系学部 3 年生 6 名を対象とし，Java プログラミング問題を 30 分間行わせた．2 回目は，2011 年 6 月に実施し，情報系学部 4 年生 7 名を対象とし，Java プログラミング問題を 30 分行わせた．そして 1 回目，2 回目ともに，プログラミング演習時に収集したログを，模擬授業を行った講師に見せて使いやすさや使用感について尋ねた．

事前実験で講師が閲覧した管理画面を**図 2** に示す．左列から順に，授業 ID，授業日，課題 ID，コンパイル日時，ファイル名，エラー行，エラー数，エラー種，状態種，ユーザ ID である．

3.3 事前実験の結果と考察

実験の結果としては，つまずいている学生を早期発見することができなかった．講師からは，表示されるログが大量であり，どれが重要でどれが重要でないかを素早く判断できない等の指摘を受けた．以下に，講師から指摘された問題点を列挙する．

問題 1 ログの内容から，どれが重要な問題かを発見することが困難．

問題 2 コンパイル時と，プログラム実行時にログを収集するだけでは，コンパイルまでたどりつけない学生を発見できない．

問題 3 コンパイル失敗時や実行失敗時のソースコードが保存されていないため，エラー原因を特定できない場合があった．

問題 4 問題をかかえている学生の存在が分かっても，どの座席に座っている学生か分からない．

さらに，事前実験終了後に，著者らがエラーログを詳細に解析したところ，エラーの種類によって発生頻度や解決にかかる時間が大きく違っていることや，多く発生しているエラーが必ずしも TA が対応すべきエラーとはいえないことが分かった．たとえば，最も多く発生したエラーの 1 つとして，変数やメソッドが未宣言，スペルミス，メソッド名の後に () がない等の場合に出現する「シンボルがありません」というエラーがあるが，これはすぐに解決できる場合がほとんどであり，その解決に時間がかかる学生は存在しなかった．

以上に述べた，講師に指摘された問題，および，エラー

© 2013 Information Processing Society of Japan

lecid	date	exid	time	file	line	number	compile	state	use
prog	ex1	ex1_1	2011/10/25 14:10:50	Ex1_1.java	40	1	1	0	a
prog	ex1	ex1_1	2011/10/25 14:10:50	Ex1_1.java	40	2	1	0	a
prog	ex1	ex1_1	2011/10/25 14:10:50	Ex1_1.java	40	3	1	0	a
prog	ex1	Ex1_2	2011/10/25 14:11:57	Ex1_2.java	6	1	1	0	a
prokiso	20111025	kiso	2011/10/25 16:32:39	Report5_1.java	12	1	1	2	c01
prokiso	20111025	kiso	2011/10/25 16:36:39	Report5_1.java	6	1	1	2	C01
prokiso	20111025	kiso	2011/10/25 16:36:39	Report5_1.java	6	2	1	2	C01
prokiso	20111025	kiso	2011/10/25 16:36:39	Report5_1.java	6	3	1	2	C01
prokiso	20111025	kiso	2011/10/25 16:36:39	Report5_1.java	10	4	1	2	C01
prokiso	20111025	kiso	2011/10/25	Report5_1.java	10	5	1	2	C01

図 2 ログ閲覧画面（プロトタイプ実装）
Fig. 2 Log-browser on prototype system.

ログを解析した結果から考察した結果は以下のとおりである．

- 個人の問題を発見するポイントとして，いつまでも消えないエラーを発見しやすくすることが重要．
- 全体の問題の発見するポイントとして，多くの人が発生させているエラーを発見しやすくすることが重要．

なお，プログラミング演習支援の関連研究として，学生の操作ログ・エラーログを収集して講師や TA に提示するシステムの構築例が報告されている．たとえば，キー入力，コンパイル結果等を監視するクライアントソフトを学生の PC にインストールし，クライアントから集取された情報を講師の PC 画面で表示できる学習支援環境 [16]，学生の PC にインストールされたクライアントソフトを使ってプログラムを入力，コンパイル，実行し，その画面出力結果を定期的に講師の PC に送信し，学生が入力中のソースコードの行数等の概要を講師が見ることができる学習支援環境 [17]，Web サーバスクリプトの作り方を学習させることを目的として，学生がアップロードしたり実行したりしたスクリプトソースコードや実行結果の全エラーログを講師が手元の PC で確認できる学習支援環境 [18] がある．しかしながらこれら関連研究は，著者らの検証用プロトタイプ同様に，いつまでも消えないエラーを発見しやすくしたり，多くの人が発生させているエラーを発見しやすくするための機能が備わっておらず，本研究の本質的な課題を解決することはできない．表示される大量のログの中から，どれが重要でどれが重要でないかを授業中に素早く判断できないという問題が残っている．

また，プログラミング学習のためにソースコードの穴あき問題を Web に掲示しておき，学生がソースコードを入力してコンパイル・実行し，正しいかどうかを学生および講師が確認できる学習支援環境システムの構築例 [19], [20] が報告されている．学生が穴あき部分に入力したソースコードの自動解析機能が備わっており，学生が入力中のソースコード中の変数の数，分岐命令や繰返し命令の数，未使用変数の数等を調べ，学生にアドバイスを与えること等もできる [19]．しかしながら，これら関連研究は学生の操作ログ・エラーログを収集して講師に提示できるものの，いつまでも消えないエラーを発見しやすくしたり，多くの人が発生させているエラーを発見しやすくするための機能は備わっておらず，著者らの検証用プロトタイプ同様の問題が残っている．

4. 提案システム

事前実験により，「個人の問題を発見するポイントとして，いつまでも消えないエラーを発見しやすくすることが重要」であり，「全体の問題の発見するポイントとして，多くの人が発生させているエラーを発見しやすくすることが重要」であることが分かった．この事前実験結果を受けて，本研究ではログ解析・提示機能を提案する．

提案システムに実装したログ解析・提示機能について説明する．講義中に講師が見る管理画面は**図 3** のようなものである（初期状態では，左パネルのみが表示される）．左パネルに，学生ごとの名前，座席番号，課題番号，直近の操

© 2013 Information Processing Society of Japan

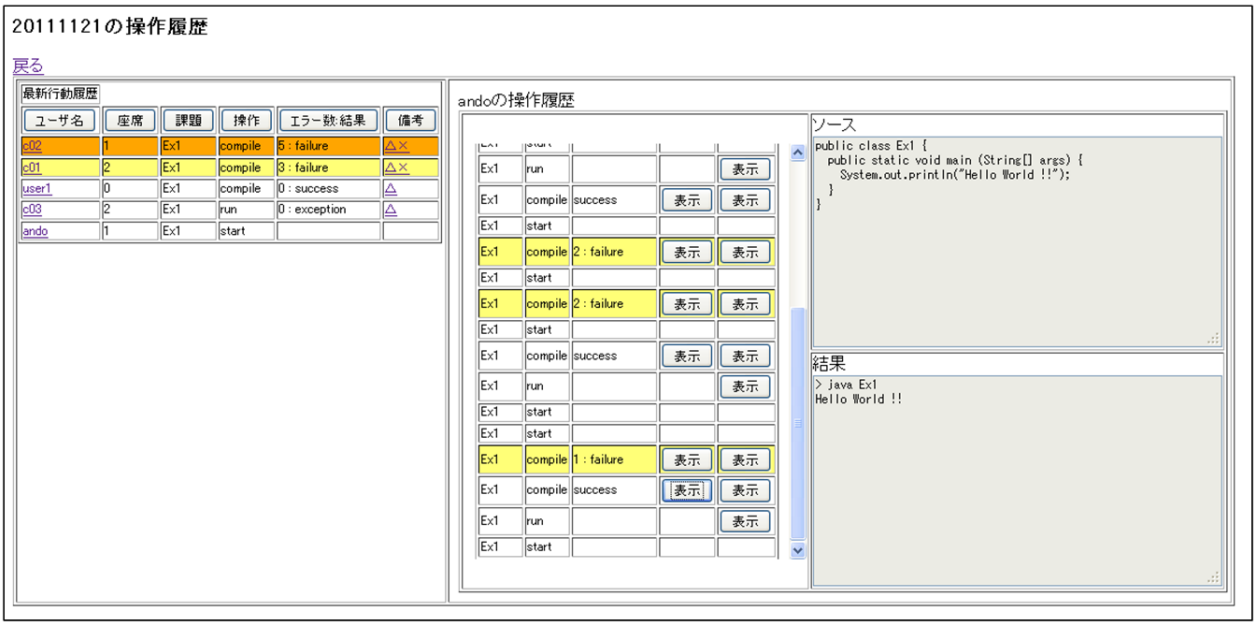

図 3 ログ解析・提示画面

Fig. 3 Analysing and browsing log data.

エラー名	数	平均解決時間	最短時間	最長時間	分散	偏差
;がありません。	2	0h1m59s	0h0m3s	0h3m56s	13572s	0h1m56s
シンボルを見つけられません。	1	0h0m9s	0h0m9s	0h0m9s	0s	0h0m0s
identifier がありません。	2	0h0m8s	0h0m8s	0h0m8s	0s	0h0m0s
メソッドの宣言が不正です。戻り値の型が必要です。	2	0h0m8s	0h0m8s	0h0m8s	0s	0h0m0s
class、interface、または enum がありません。	4	0h0m7s	0h0m7s	0h0m7s	0s	0h0m0s

図 4 エラー集計ログ

Fig. 4 Error summary log.

作，エラー数，同一エラー継続時間，非操作時間がリスト表示されており，同一エラーが長時間継続している学生や非操作時間が長時間継続している学生は色分け表示される．

前記事前実験の問題 1 に対し，重要なエラーを優先的に表示する必要あると考え，同じエラーが継続的に出ている学生や，多くのエラーを出している学生を検出して表示するようにした．具体的には，所定時間以上（デフォルト＝5 分）同じエラーが継続して出ている学生，および，所定回数以上（デフォルト＝10 個）エラーを繰り返し出している学生を赤色で表示する機能を実装した（エラー数が 0 の場合は白，1 から 4 個の場合は黄色，5 から 9 個の場合はオレンジ色，10 個以上を赤とした）．

問題 2 に対し，前回の操作から一定時間経過し，非操作時間が継続している学生を検出して表示するようにした．具体的には，10 分以上操作していない学生には△印，15 分以上操作していない学生には×印を付加している．

問題 3 に対し，コンパイルまたはプログラム実行した時点のソースコードを，エラーログ（エラー個数，エラー名，エラー発生行，エラー発生箇所，エラー原因解説文），操作ログと対応付けてデータベースに保存し，表示の際にも，クリックのみで素早く対応関係を参照できるようにした．

問題 4 に対し，机に貼られた座席番号，または，自分が座っている場所の座席番号を座席表から探して，システムログイン時に学生自らが入力できるようにした．

講師が行をクリックすると，その行の学生についての詳細情報が中央パネルに表示される．中央パネルには，対象学生の操作履歴が上から下に並んで表示されており，過去にさかのぼって，学生がどのようなエラーや操作イベントを発生させたかが分かる（最下部が最新履歴）．各行には，そのエラーや操作イベント発生時点の「ソースコードを表示するためのボタン」と，その時点の「プログラム実行結果を表示するためのボタン」が表示されており，「ソースコードを表示するためのボタン」が押された場合には右パネルの上部に，「プログラム実行結果を表示するためのボタン」が押された場合には右パネルの下部にそれぞれ表示される．

加えて，多くの学生が共通にかかえる問題を発見するという観点から，授業中に発生したエラーごとに，解決時間の平均や分散を集計表示する機能を実装した．図 4 に示す．この表示はリアルタイムに変化し，授業中のどの時点でも確認することができる．講師が想定しないポイントで多くの学生がつまずいている場合にそのことを発見するための手段として構築した．授業の改善点を発見する用途で用いられる．

© 2013 Information Processing Society of Japan

5. 評価実験

提案システムを実際のプログラミング演習授業に投入し，実際の授業で学生に使用させる実験を行った．本実験の目的は，以下の点を検証することである．

実験目的 1 つまずいている学生を早期に発見することができるか．

実験目的 2 多くの学生が共通にかかえる問題に気づくことができるか．

実験目的 1 は本研究の目的 1（つまずいている学生の早期発見）に対応しており，実験目的 2 は本研究の目的 2（多くの学生が共通にかかえる問題を発見）に対応している．したがって，実験目的 1 と実験目的 2 に効果があったことが示されれば，本研究の目的が達成されたと判断できる．

5.1 実験 1

実験 1 は，情報系学部 1 年生対象の「プログラミング基礎 2」という Java プログラミング演習授業において実施することとした．講師 1 名および TA 3 名（全員大学院生）がこの授業の教授者である．

実際の学部授業であることから，本実験のために授業に支障が生じたり，学生から何らかの強い不満がでたり，学生間になんらかの不公平が生じたりしてはならない．そこで評価実験を 2011 年 11 月 22 日 4，5 限に実施することに定め，それに先立ち，2011 年 10 月 25 日 4，5 限の授業時間を利用して本システムを当該授業で試験的に運用してみた．事前に試験運用をすることによって，学生が本システムの使い方をひととおり習得した状態で評価実験を開始できるメリットもあると考えた．

結果的に 10 月 25 日の試験運用では懸念した問題は生じず，むしろ本システムの利用は学生からおおむね好評であることが確認できた．今までのプログラミング授業において，ほとんどの学生はメモ帳等の一般的なテキストエディタを利用してきたが，Web ブラウザ上の CodeMirror にプログラムを入力することに関しても不満はほとんど出なかった．むしろ，ソースコード中の予約語が色分表示されたり，“ や (が自動補完されたり，改行時に自動インデント整形されたりするため好評であった．よって，予定どおり評価実験を 11 月 22 日に実施することとした．評価実験の実施条件を以下にまとめる．

実験 1	
日時	2011 年 11 月 22 日 4，5 限
演習内容	Java の abstract，implement 構文
演習課題	48 行と 65 行のプログラム 2 問
学生数	14 名（当日受講者全員）
TA 数	3 名

表 1 実験 1 の結果（学生回答，個別問題の発見）
Table 1 Experimental results on students about personal problems.

	悩んでいるとき，手を挙げる前に TA が来てくれたことがありましたか
はい	11 名
いいえ	1 名

表 2 実験 1 の結果（学生回答，共通問題の発見）
Table 2 Experimental results on students about common problems.

	講師から出されたヒントは自分が悩んでいる箇所の内容でしたか
はい	5 名
いいえ	3 名
すでに解決済み	4 名

なお，Web ブラウザの種類によって不具合が生じる可能性をなくすため，学生には Firefox を使用して本システムにアクセスするように周知した．また Web サーバには LinuxOS を用いた．

また講師と TA には，管理画面（図 3）の左パネルを頻繁に確認することを依頼するとともに，このパネル中で色付け表示されたり備考に記号が表示されたりした学生がつまずいている学生である可能性が高いことを伝えた．

5.2 実験 1 の結果

実験 1 の結果について述べる．本システムを使った授業後に，授業に出席していた，学生（全員），講師，TA（全員）に対してアンケートを実施することによって行った．

学生に対して，**表 1** に示されるアンケート，および，**表 2** に示されるアンケートを実施した．アンケート項目，および，アンケート項目に対する回答は表に示されるとおりである．

加えて，学生には，システムを利用して授業を受けた感想について自由筆記アンケートも行った．その結果，以下の意見を得ることができた．

(1) 消えないエラーに悩んでいるときに，手を挙げる前に TA が来てアドバイスをくれた．
(2) Java のコンパイルと実行機能が揃っているため今後の講義でも十分利用できそう．
(3) 通常のテキストエディタより Web 上のエディタの方が使いやすく，苦手意識を克服できそう．

また，講師と TA に対して，**表 3** に示されるアンケートを実施した．アンケート項目，および，アンケート項目に対する回答は表に示されるとおりである．

加えて，講師と TA には，システムを利用した感想，および，授業を観察した結果について自由筆記アンケートをした．その結果，以下の意見を得ることができた．

© 2013 Information Processing Society of Japan

表 3 実験 1 の結果（講師・TA 回答）
Table 3 Experimental results on teaching staff.

	悩んでいそうな学生を発見したことがありましたか
はい	4 名
いいえ	0 名

(1) 管理画面（図 3）の左パネルで赤く示された学生を中心に個別指導を行った．
(2) 管理画面（図 3）の左パネルでエラー個数が多い学生に注目していた．
(3) 学生が挙手する前に他の TA が指導しに行く姿が見られた．

5.3 実験 2

実験 1 に続き，評価実験の規模を大きくした実験 2 を実施した．学生数が多く TA が不足気味の状況で提案システムを評価することが目的の 1 つである．

実験 2 は，実験 1 と同じ「プログラミング基礎 2」の別クラスにおいて実施した．実験 1 と実験 2 の講師は異なっている．実験のために授業に支障が生じたり強い不満がでたりしてはならないことから，評価実験を 2013 年 6 月 7 日 4，5 限に実施することに定め，それに先立ち，2013 年 5 月 31 日 4，5 限の授業時間を利用して提案システムを当クラスで試験的に運用した．5 月 31 日の試験運用で問題は生じなかったことから，計画どおり評価実験を 6 月 7 日に実施することとした．

実験 2 では TA 4 名と学生数 74 名を実験被験者とした．実施条件を以下にまとめる．

実験 2	
日時	2013 年 6 月 7 日 4，5 限
演習内容	Java の繰り返し構文（応用）
演習課題	9〜12 行のプログラム 5 問
学生数	74 名（当日受講者全員）
TA 数	4 名

また実験 2 では，教室内の TA 4 名と学生 74 名を，提案システムを利用するグループ（以後，使用グループ）と，提案システムを利用しないグループ（以後，不使用グループ）に分割し，それぞれのグループの人数が TA 2 名＋学生 37 名となるようにした．使用グループと不使用グループに分割することにより，提案システムを利用した場合と利用しなかった場合の違いを比較することが目的である．それ以外の条件は実験 1 と同一である．

5.4 実験 2 の結果

実験 2 の結果について述べる．

両グループの TA に対して，手が挙がる前にサポートできた学生数を尋ねた．結果は**表 4** に示されるとおりとなった．結果より，同じ授業内においても，システム使用時の方がより多くの学生の問題を早期に発見しサポートできることが分かった．

表 4 実験 2 の結果（TA 回答）サポートできた学生数
Table 4 Experimental result - the number of helped students.

	手が挙がる前にサポートできた学生数
使用グループの TA	計 18 名
不使用グループの TA	計 6 名

表 5 実験 2 の結果（TA 回答）早期発見できたと感じるか
Table 5 Experimental result - stumbled student was quickly found or not.

	問題を早期に発見できたと感じるか
使用グループの TA	平均 4.5
不使用グループの TA	平均 2.0

次に，両グループの TA に対して，問題を早期に発見できたと感じるかどうかを 5 段階（1 まったく感じない，2 感じない，3 どちらともいえない，4 感じる，5 とても感じる）で回答させた．結果は**表 5** に示されるとおりである．

加えて，使用グループの TA2 名には，システムを利用した感想について自由筆記アンケートをした．その結果，以下の意見を得ることができた．

(1) 困っている学生を発見しやすかった．
(2) いつもより積極的にサポートできたことは良かった．

なお，実験 2 では，実験目的 2（多くの学生が共通にかかえる問題に気づくことができるか）についての検証は行わないこととした．講師が口頭または板書で問題解決のヒントを出すと，使用グループと不使用グループの両方の学生にそのヒントが伝わってしまい，グループ別の評価ができないと考えたためである．

5.5 考察

実験目的 1（つまずいている学生を早期に発見することができるか）の評価としては，実験 1 の表 1，表 3 の結果，および，実験 2 の表 4，表 5 の結果から，注目すべき学生を手が挙がる前に発見できたことが分かった．加えて，学生や TA への自由筆記アンケートの結果からもそのような傾向がうかがわれた．事前実験用に構築した検証用プロトタイプでは，いつまでも消えないエラーや，つまずいている学生を早期発見することができなかったが，今回構築した提案システムではこれらの状況が改善されたことから，提案システムに効果があったと判断した．

また，実験目的 2（多くの学生が共通にかかえる問題に気づくことができるか）の評価としては，表 2 の結果から，講師が課題のヒントとして説明した内容と学生が悩ん

© 2013 Information Processing Society of Japan

でいる箇所が一致したという意見が一定数得られたことから，多くの人が発生させているエラーを発見しやすくしたことの効果があったと判断した．なお，表 2 の結果は，学生の約 1/3 が課題を完了したタイミングでヒントを出すべきと講師が判断した場合の結果である（このため「すでに解決済み」と回答した学生が 4 名存在した）．講師によってこのタイミングは異なると思われるためさらなる検証が必要である．

また，実験を通して得られた他の知見として，「; がありません.」「) がありません」「, がありません」のように，類似したエラーメッセージが多数発生してログが見にくくなる状況がしばしば生じることが分かった．この対策として，新たにエラー分類登録機能を追加し，個々のエラーメッセージの代わりに「〜がありません」のような事前に登録した分類名を表示できるようにした．

ただし，現状の提案システムは，コンパイルエラーが出力されたり実行エラーが出力されたりする場合に対しては効果が見込まれるが，一方，実行時に実行エラーが出力されないバグを発見しやすくするための機能を有していない．このため，たとえば，ロジックの誤りにより計算結果が間違うといった問題には対応できない．この問題への対策として，今後，文献 [19] で述べられているような，学生のソースコードと講師のソースコードで，変数の数，分岐命令や繰返し命令の数，制御命令の出現順序等を比較し，学生にアドバイスを与える自動解析機能を追加することを検討したい．

6. まとめ

本研究では，プログラミング演習講義において，つまずいている学生の早期発見と，多くの学生が共通にかかえる問題の発見を可能にすることを狙いとした，プログラミング学習支援システムを構築した．目的を達成するために，Web ブラウザ上で動作するプログラミング演習支援システムを構築し，学生の操作ログ・エラーログを収集して解析できるようにした．

評価実験の結果，つまずいている学生を早期に発見することができるかの評価としては，いつまでも消えないエラーが発見しやすくなり，また，注目すべき学生を手が挙がる前に発見できたことが分かった．また，多くの学生が共通にかかえる問題に気づくことができるかの評価としては，講師が課題のヒントとして説明した内容と学生が悩んでいる箇所が一致したという意見が得られた．これらのことから提案システムに効果があったと判断した．

参考文献

[1] 野口孝文：ゲーム作成を課題にしたプログラミング教育とその分析方法の開発，電子情報通信学会技術研究報告，Vol.104, No.222, pp.1–6 (2004).

[2] 北　栄輔，山梨樹里：Peer Review に基づいたプログラミング実習授業支援ツールの開発，名古屋高等教育研究，Vol.7, pp.341–353 (2007).

[3] 知見邦彦，櫨山淳雄，宮寺庸造：失敗知識を利用したプログラミング学習環境の構築，電子情報通信学会論文誌 D-I，Vol.J88-D-I, No.1, pp.66–75 (2005).

[4] 中村亮太，西田知博，松浦敏雄：プログラミング入門教育用学習環境 PEN，情報処理学会研究報告：コンピュータと教育研究会報告，Vol.2005, No.104, pp.65–71 (2005).

[5] Truong, N., Bancroft, P. and Roe, P.: Learning to Program Through the Web, *Proc. ACM 10th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education*, pp.9–13 (2005).

[6] Way, T.P.: A Company-based Framework for a Software Engineering Course, *Proc. 36th SIGCSE Technical Symposium on Computer Science Education*, pp.132–136 (2005).

[7] 山下亮輔，古川雅基，安田　光，井上亮文，市村　哲：動画を用いたプログラミング演習支援システム，情報処理学会研究報告：GN，Vol.2009, No.33, pp.67–72 (2009).

[8] 梶並知記，安田　光，井上亮文，市村　哲：プログラムコーディング過程を記録した動画教材の作成作業を支援するインタフェース，情報処理学会 DICOMO 2011，6B-2, pp.1035–1042 (2011).

[9] 安田　光，井上亮文，市村　哲：学生とティーチングアシスタント間でトラブル解決過程を共有できるプログラミング演習支援システム，情報処理学会論文誌，Vol.53, No.1, pp.1–10 (2012).

[10] Orion – Eclipse, Eclipse の Web IDE サービス (2013)，入手先 ⟨http://wiki.eclipse.org/Orion⟩.

[11] Skywriter, Mozilla Labs の Web IDE サービス (2013)，入手先 ⟨http://mozillalabs.com/skywriter/⟩.

[12] Ideone.com, SOAP API を備えた Web IDE サービス (2013)，入手先 ⟨http://ideone.com⟩.

[13] 濱野，妻鳥：プログラミング実習支援システムにおける学習者支援，高知工科大学学士学位論文 (2010)，入手先 ⟨http://www.kochitech.ac.jp/library/ron/2009/2009info/1100332.pdf⟩.

[14] CodeMirror，Web ブラウザでコードエディタを使用するための JavaScript のコンポーネント (2013)，入手先 ⟨http://codemirror.net/⟩.

[15] jQuery 公式サイト (2013)，入手先 ⟨http://jquery.com/⟩.

[16] 寺元：プログラミング教育を支援する問題解決環境に関する研究，宇都宮大学大学院博士論文 (2011)，入手先 ⟨http://www.ee.utsunomiyau.ac.jp/~kawatalab/research/pse/teramoto.pdf⟩.

[17] 張，西田，阿倍，石橋，松浦：プログラミング授業を支援する学習環境 PEN+，*Journal of Infomatics*, Vol.5, No.1 (2008)，入手先 ⟨http://creativecity-j.gscc.osakacu.ac.jp/JI/article/download/75/67⟩.

[18] 青木，大木，片山：Web を活用したプログラミング授業支援の試み，論文誌情報教育方法研究，Vol.4, No.1, pp.1–3 (2001)，入手先 ⟨http://www.juce.jp/archives/ronbun_2001/01.pdf⟩.

[19] Truong, N., Bancroft, P. and Roe, P.: Learning to Program Through the Web, *Proc. ACM ITiCSE'05*, pp.9–13 (2005).

[20] Hitz, T. and Koegeler, S.: Teaching C++ on the WWW, *Proc. ACM ITiCSE'97*, pp.11–13 (1997).

推薦文

プログラミング演習でつまずいている学生は自ら質問をしない傾向にあるが，そのような学生の状態を早期に把握

© 2013 Information Processing Society of Japan

し指導につなげることを目指した研究である．実地で必要な機能を実装し，運用・評価している．効果的に問題をかかえる学生をリアルタイムで発見できる方法を実現しており，推薦論文に値する．

（グループウェアとネットワークサービス研究会主査 小林　稔）

市村 哲

1989 年慶應義塾大学理工学部計測工学科卒業．1994 年同大学大学院理工学研究科博士後期課程修了．博士（工学）．同年富士ゼロックス（株）入社．1997〜1999 年富士ゼロックスパロアルト研究所（FXPAL）駐在．2002 年より東京工科大学．2011 年同大学教授．グループウェア，ネットワークサービス，生体情報活用等の研究に従事．『IT TEXT 基礎 Web 技術』，『IT TEXT 応用 Web 技術』（オーム社）．DICOMO 2011 最優秀論文賞受賞．ACM，電子情報通信学会各会員．

梶並 知記

2004 年東京都立科学技術大学工学部電子システム工学科卒業．2010 年首都大学東京大学院システムデザイン研究科博士後期課程修了．首都大学東京システムデザイン学部特任研究員，東京工科大学コンピュータサイエンス学部助教を経て，2012 年より神奈川工科大学情報学部助教．博士（工学）．情報可視化技術を用いた意思決定支援，創造活動支援の研究に従事．ACM，日本知能情報ファジィ学会，人工知能学会，電子情報通信学会，情報知識学会，日本デジタルゲーム学会各会員．

平野 洋行

2010 年東京工科大学コンピュータサイエンス学部卒業．2012 年同大学大学院バイオ・情報メディア研究科修士課程修了．

© 2013 Information Processing Society of Japan