

多人数でのプログラミング演習における 学習者のコンパイルエラー状況の把握システム

彦坂知行*1・北英彦*1

Email: hikosaka@ce.elec.mie-u.ac.jp

*1: 三重大学大学院工学研究科電気電子工学専攻

◎Key Words プログラミング演習, 机間巡回, 演習状況把握

1. はじめに

プログラミングの授業では、一般に、プログラム作成を行う演習を行っている。しかし、多人数の場合、演習中にプログラムの作成が円滑に進まず指導を必要としている学習者を机間巡回で見つけることは困難である。著者らはプログラミング演習中に、学習者の状況をリアルタイムで把握することができるプログラミング演習システム PROPEL (PROgramming Practice Easy for Learners) を開発・運用している⁽¹⁾⁽²⁾。

このシステムでは従来、学習者が一定時間プログラムをまったく変更しなかった場合しかシステムは検知することができておらず、指導が必要な学習者の一部しか講師はシステムを介して把握することができなかった。本研究では、指導が必要な学習者をより早く講師が把握できるようにするために、コンパイルエラーまたはコンパイル前に1行単位で検出できるエラーが長時間あるいは多数残っている学習者の一覧を講師に提供する。

2. プログラミング演習支援システム

プログラミング演習中の受講者の中には、何をやっていいかわからず手が止まっている者がいる。このような受講者には、講師が早い段階で的確なアドバイスを必要がある。そのために、著者らは、プログラミング演習システム PROPEL の開発を行っている。

このシステムでは、学生はプログラミング演習において必要な作業のうち「デバッグ」以外の「コーディング」「コンパイル」「プログラムの実行」「保存」「提出」を Web 上のひとつの画面上で行うことができる。また、受講者が講師を呼び出すことができる「呼び出し」機能を提供している。

システム構成図を図1に示す。学習者は Web ブラウザを使用し、図2に示す画面でプログラムを編集し、保存したプログラムは Web サーバー上に保存される。また編集中のプログラムも30秒ごとに自動保存される。講師は Web ブラウザで受講者の状況を確認する。

講師は、図3に示す講師用画面を閲覧することで、クラス全体の進行状況を把握することができる。受講者が「呼び出し」機能を用いた場合には、座席が赤色で表示される。また、受講者がコーディング途中のプログラムが5分以上同じ状態、すなわち入力や修正が全く行われていない場合には、黄色で、さらに15分以上だと橙色で表示される。これにより、講師はプログ

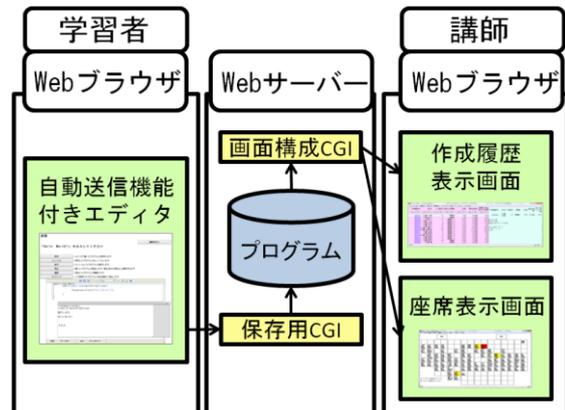


図1 PROPELのシステム構成図



図2 プログラムの編集画面

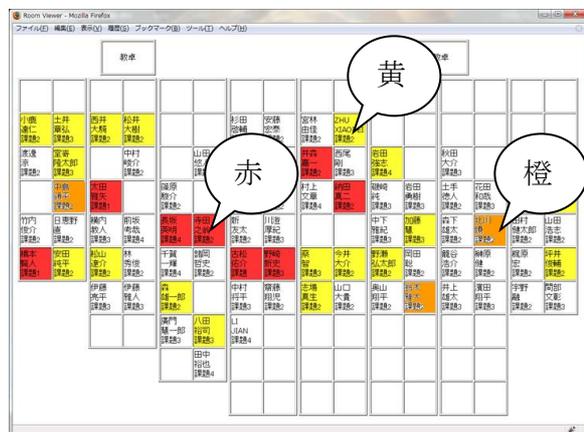


図3 講師用画面：演習室内座席表

ラムの作成に行き詰っている受講者のところへ迅速に指導しに行くことができる。

講師は、また、図4に示す受講者の一覧を表示した講師用画面を閲覧することでも、クラス全体のプログラミング演習の進行状況を把握することができる。受講者を指定すると、この画面の右側の部分にその受講者が現在コーディング中のプログラムが表示される。これにより、気になる受講者のプログラムの現在の様子を知ることができる。図4のプログラム作成状況の部分で拡大したものを図5に示す。

プログラミング演習システム PROPEL は、携帯可能で無線LANによっていつでもシステムにアクセスすることのできる情報端末 iPad 用の講師用画面を提供している²⁾。iPad 等を利用することで、講師は教室内を巡回中でもシステムの利用を通じてクラス全体のプログラミング演習の進行状況を知ることができる。

学籍番号	氏名	PC番号	現在の課題	行数	最終更新	課題に投入した時間
408313	木田 隼矢	124	課題2	18行	10分前	33分
408344	長坂 実明	108	課題4	43行	40分前	63分
408345	中田 文治	99	課題2	41行	40分前	61分
408312	井筒 真一	67	課題2	20行	37分前	64分
408303	野崎 龍平	80	課題2	28行	36分前	57分
408302	橋本 真一	140	課題2	16行	40分前	65分
408304	松井 大輔	58	課題2	28行	39分前	55分
408305	松井 大輔	122	課題2	28行	37分前	53分
408306	松井 大輔	114	課題2	28行	24分前	47分
408311	松井 大輔	47	課題4	16行	37分前	22分
408312	森 健一郎	110	課題2	42行	19分前	67分
408343	八田 昭司	102	課題2	11行	14分前	14分
408308	ZHU XIAOPING	56	課題2	28行	14分前	29分
408309	松山 孝一	106	課題2	28行	13分前	49分
408377	安田 純平	134	課題2	27行	13分前	35分
408344	松井 大輔	6	課題2	27行	13分前	41分
408313	木田 隼矢	124	課題2	18行	10分前	33分
408344	松井 大輔	40	課題2	40行	40分前	62分
408341	青田 文治	99	課題2	41行	41分前	61分
408312	井筒 真一	67	課題2	30行	37分前	64分
408303	野崎 龍平	80	課題2	28行	36分前	57分
408302	橋本 真一	140	課題2	16行	40分前	65分
408304	松井 大輔	58	課題2	28行	39分前	55分
408305	松井 大輔	122	課題2	28行	37分前	53分
408306	松井 大輔	114	課題2	28行	24分前	47分
408311	松井 大輔	47	課題4	16行	37分前	22分
408312	森 健一郎	110	課題2	42行	19分前	67分

図4 受講者一覧、および、ある受講者のプログラム

学籍番号	氏名	PC番号	課題	行数	最終更新	取り組んだ時間
408304	東 靖久	4	課題2	18行	10分前	33分

```

#include <stdio.h>
int str_ch2num(const char str[], int cd, int dc)
{
    int i;
    int count =0;

    for (i = 0; str[i] != '\n' ; i++)
        if (str[i] == cd || dc)
            count++;
    return(count);
}

int main(void)
{

```

図5 ある受講者の現在のプログラム

実際に運用した結果、学習者の現在取り組んでいる課題や無操作時間から学習者の状況を把握し、指導を行うことができている。

3. PROPEL の問題点

PROPEL には2つの問題点がある。1つ目は一定時間入力を行っていない学習者しか検出できないことである。2つ目は学習者がコンパイルするまでシステムによる学習者のプログラムのチェックを行えない点である。以下にそれぞれについて先行研究について述べる。

4. 早期エラーチェック

演習ではプログラムの間違いを早期に発見し、早期に修正することが重要であると考えられる。コンパイルエラーの情報は、学習者がコンパイルするまで得ることができない。小島は初心者が起こしやすいエラーについて調査し、プログラムが完成する前でも、1行単位で間違いがないかどうかチェックすることのできる方法を提案した⁴⁾。この方法によりプログラム作成中でも間違いを検出し、早期に学習者に間違いをシステムが表示することで講師の手間を削減できる。チェックするエラーについては表1に示す。

小島の提案では早期エラーチェックの情報は学習者に対して表示するが、本研究では講師に対して情報を表示することで指導すべき受講者を早期に発見し、指導を行えるようにする。

5. コーディング状況の可視化

現在、PROPEL では文字入力を一定時間行っていない学習者については、図6のように座席表上に表示することができる。しかしエラーをなくすることができず長時間、試行錯誤している学習者については表示することができなかった。高橋は学習者のエラーの発生状況をタイムチャートとして表示することによりコーディング状況を可視化するシステムを提案した³⁾。

図6に高橋が作成したシステムの画面を示す。左側にエラーの内容が表示され、右側にそのエラーが発せられている期間がタイムチャートで表示される。エラーの種類としては、コンパイルエラー、小島提案の早期チェックエラー、条件を緩めた自動テスト⁵⁾による動作エラーがある。自動テストについては後述する。

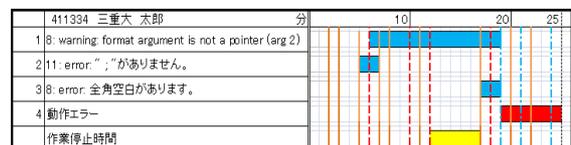


図6 高橋が提案したタイムチャート

6. 提案するシステム

本研究では高橋と小島の案をもとにエラーの情報を活用した状況把握システムの実装を行う。

提案するシステムの画面を図7に示す。画面の左側には従来のPROPELの機能で座席表を表示する。右側上部に学習者のエラーの残留時間を表示する。右側下部に選択した学習者のプログラムを表示する。

右側上部の情報はエラーの残留時間が長い順番にソートされている。学生のエラー時間の順位、氏名、学籍番号、エラーの残留時間について表示する。エラーの残留時間については棒グラフで表示することで講師が状況を速やかに把握できるようにした。また学習者が発生させているエラーの種類について、どの種類であるかを分かるように、各エラーに対して省略記号で表示する。それぞれの省略記号の意味を表1に示す。

また三重大学電気電子工学科1年生を対象としたプログラミング演習においてコンパイルエラーメッセージの集計を行った。結果を表2に示す。

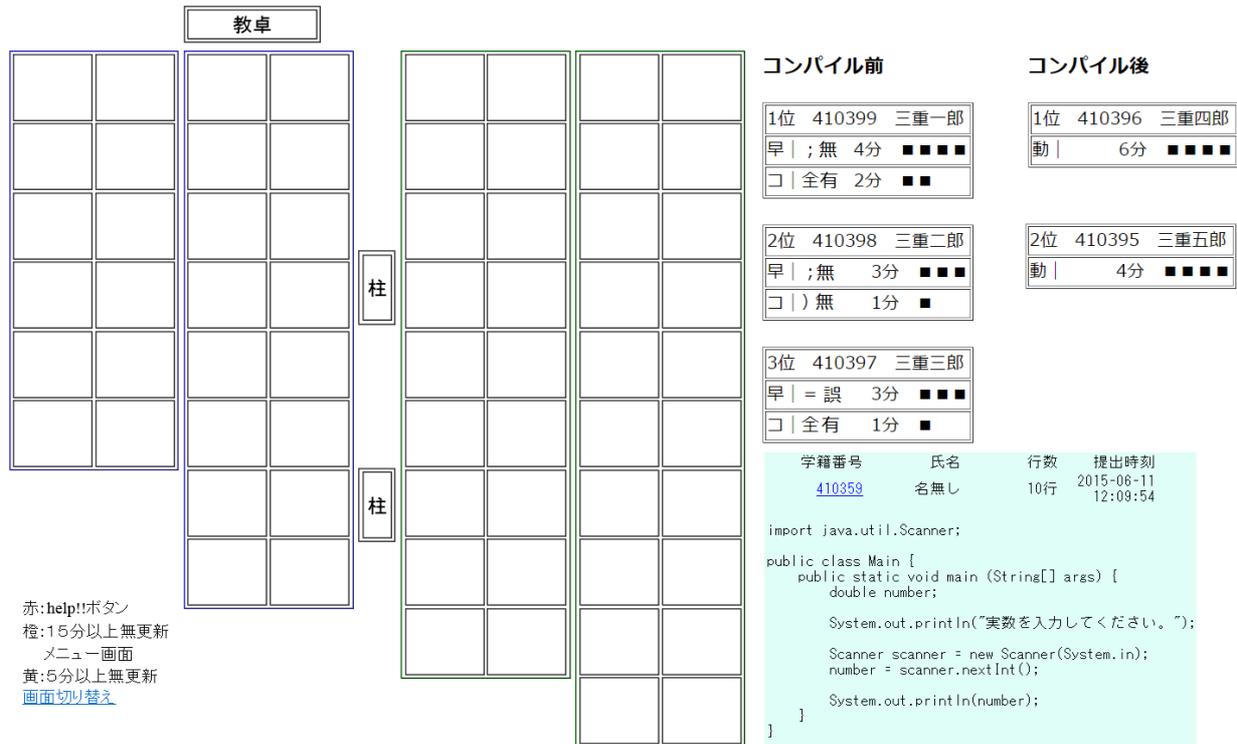


図 7 提案するシステムの画面設計

この結果に基づいて、簡易エラーチェックと同様にそれぞれ省略記号を決めた。表 3 に示す。

表 1 簡易チェックの省略記号と意味

記号	意味
名未	ライブラリ関数, 変数名のタイプミス.
; 無	文末セミコロン忘れ.
" 無	" "" の書き忘れ.
全有	全角文字の使用
比誤	不等号の順番間違い. 「=<」などと間違っ使用.
比=	=と=の間違い
比連	比較の間違い 例 : a<b<c
他 ;	余計なセミコロンがある. 例 : for (i=0; i<1; i++;
他,	, と&&の間違い

表 2 コンパイルエラーの発生数上位 10 位(10704 個中)

エラーメッセージ	発生数
シンボルを見つけれられません	2104
;!がありません	2037
文ではありません	893
¥12288 は不正な文字です	715
式の開始が不正です	605
構文解析中にファイルの終わりに移りました	449
)がありません	431
型の開始が不正です	309
<identifier>がありません	270
互換性のない型	235

表 3 コンパイルエラーの省略記号

省略記号	エラーメッセージ
名未	シンボルを見つけれられません
名無	<identifier>がありません
; 無	;!がありません
文非	文ではありません
式始	式の開始が不正です
型始	型の開始が不正です
全有	¥12288 は不正な文字です
解終	構文解析中にファイルの終わりに移りました
) 無)がありません
型互	互換性のない型

7. 自動テストによる動作エラー

コンパイルされたプログラムはシステムによって正しく動作するか自動テストを行う。自動テストは講師が設定したテストケースを満たすかどうかをテストするブラックボックステストを用いる。

ただし、テストは通常ソフトウェア開発で使われるソフトウェアテストではなく望月が提案したプログラミング初心者向けに、出力に関する条件を緩めたテストを用いる。通常、演習問題では出力の書式が厳密には決められていない。そのため、学習者ごとに空白の入れ方などの点で出力の細かい表現は異なる。そのため、この自動テストでは、ある入力に対して正解となる出力に含まれなければならない文字列を指定し、その文字列が学習者のプログラムの出力に含まれるかどうかで正誤を判定する。

8. 期待される効果

画面に表示する情報についてそれぞれ、利用方法、講師の対応について述べる。

- エラーの残留時間
コンパイルエラー，早期エラーチェックのエラーが出続けている時間である。この時間が長い順で学習者が表示されるため，講師は学習がうまくいっていない学習者を発見し，指導を行うことができる。また時間は棒グラフでも表示されるため，講師は直観的にエラーの残留時間が長さを把握できる。
- エラーの省略記号
学習者が頻繁に発生させる各エラーを1文字で示した記号である。講師はこの情報から学習者がどのエラーを出しているのかを素早く把握することができ，指導を行うときにプログラムを読む前にエラーの原因を推測することができる。

本システムにより，講師はエラーを発生して長時間試行錯誤している学習者に対し，速やかに指導を行うことができると考えられる。

また学習者がコンパイルを行う前でも，早期チェックにより早期に誤りを検出することで，早期に指導を行うことができると考えられる。

9. 類似研究

プログラミング演習中の状況把握に関する研究としては以下の研究がある。

加藤らは，学習者の作業の進捗集計，エラーの分類集計，作業の遅れている学習者の検出等の機能を持つ学習状況を把握するシステムを開発した⁶⁾。作業の進捗集計とは編集開始，コンパイル，実行，正解，提出の各作業の人数を表示できる機能である。エラーの分類集計機能とは発生しているコンパイルエラーメッセージを分類・集計し，講師がクラス全体で発生しているエラーの内容を把握できるようにする機能である。作業の遅れている学習者の検出機能とは学習者の作業時間に外れ値の分析を行い，クラス全体と比べて作業が遅れている学習者を検出・表示する機能である。

このシステムと比較すると，提案するシステムはエラーの残留時間を利用している。そのため遅いペースではあるが着実に進んでいる学習者を検出せずに，長時間エラーが直せず困っている学習者を検出できるという利点があると考えられる。

また倉澤らは学生のコンパイルエラーメッセージ履歴から，プログラムの動作理解が困難な原因と場所を推定・集計するシステムを開発した⁷⁾。このシステムではコンパイルエラーメッセージを過去2年分収集し，それぞれにエラータイプという257種類のIDを割り当てた。エラーの要因を分析し，学習者のコンパイル時にエラーメッセージに対して過去出現頻度の高いエラー要因を表示する。

上記2つのシステムは座席表が表示されないため，講師が指導を行うとき手間がかかると考えられる。本研究のシステムは学習者の座席と対応しているため，より早く指導に行くことがより可能であると考えられ

る。また上記2つのシステムは学生がコンパイルを行うまで間違いをチェックすることができないため，それまで状況を把握できない問題点がある。

10. おわりに

プログラミング演習システム PROPEL は，講師が適格かつ早い段階で受講者にアドバイスできるようにすることを目的としている。本研究では従来使用していなかったコンパイルエラー，早期チェック，ソフトウェアテストの情報を利用して，講師が早期にコーディング状況を把握できるシステムの実装を行った。また1行単位で検出できるエラーチェックを利用することにより早期にエラーを発見することができるようにした。座席表と対応させてこれらの情報を1画面に表示することによって，講師は速やかに学習者に指導を行うことができる。

参考文献

- (1) 伊富昌幸，北英彦，高瀬治彦，林照峯：コーディング状況に応じたアドバイスを可能にするプログラミング演習システムに関する研究，コンピュータ利用教育協議会，2010 PC カンファレンス (2010)。
- (2) 小川正，西口大亮，北英彦：プログラミング演習におけるiPadなどの携帯デバイスの利用による指導の円滑化，コンピュータ利用教育協議会，2011 PC カンファレンス (2011)。
- (3) 小島佑介，高橋功欣，北英彦：プログラミング演習における効率のよい指導のためのエラー早期指摘，コンピュータ利用教育協議会，2011 PC カンファレンス (2011)。
- (4) 高橋功欣，小島佑介，北英彦：プログラミング演習における指導のための受講者のコーディング状況の可視化，コンピュータ利用教育協議会，2011 PC カンファレンス (2011)。
- (5) 望月将行：“プログラミング初心者のための自動テスト機能を備えた演習支援システムに関する研究”，三重大学工学研究科修士論文 (2004)
- (6) 加藤利康，石川孝：プログラミング演習のための授業支援システムにおける学習状況把握機能の実現，情報処理学会論文誌，55巻，8号，pp.1918-1930 (2014)。
- (7) 倉澤邦美，鈴木恵介，飯島正也，横山節雄，宮寺庸造：プログラミング演習における一斉指導のための学習状況把握支援システムの開発，電子情報通信学会技術研究報告 ET，教育工学 104(703)，pp.19-24 (2005)。