1ZC-1

プログラミング演習における学生のコーディング過程可視化システム

齊藤　俊 † 山田　誠 ††　井垣　宏 ‡　井上　亮文 †

† 東京工科大学コンピュータサイエンス学部　†† 東京工科大学院バイオ・情報メディア研究科　‡ 大阪大学大学院情報科学研究科

1　はじめに

近年，プログラミング演習に特化した支援システムが広く研究されている [1]．演習とは，講師がある単元について解説を行った後，学生は残りの時間でその内容に応じた複数の課題を時間内に解く形式を指す．

プログラミング演習の問題点は，講師が演習中に学生を直接チェックするか課題提出後でなければ，進捗や理解度を把握できないことである．課題提出後であっても，答案のソースコードのみでは学習状況や理解度を把握するのは難しい [2]．結果として，講師は支援すべき学生であるプログラミング演習中，あるエラーに関して長時間悩んでる学生や，全体の進捗に対して遅れている学生を見落としてしまうことが多い．

そこで本稿では，プログラミング演習中に自主的に手を挙げない等の従来見逃していた支援すべき学生を検出し，学生間の比較により優先して支援すべき学生を発見可能なシステムを提案する．

2　関連研究

藤原らは学生のコンパイル回数，実行回数，コンパイルエラー数，行数等のデータにおける時間に応じた変化に，学習者の取り組み状況が反映されると考え，複数の関連する学習履歴が持つ特徴を定量化することによって学生の状況を表現する手法を提案している [3]．これにより，講師が定義した特徴に一致する学生を検出することができる．しかし，講師が予め学習状況の定量化を行わなければ検出できず，定量化の際には講師に負担がかかる．また，学生間の比較をしていないため優先して支援すべき学生を発見できない．

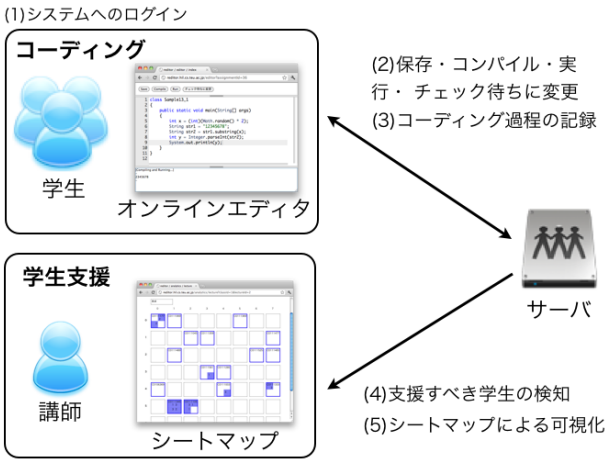

図 1: システム概要

表 1: 遅れ理由とそれを検知するためのメトリクス

学生の遅れ理由	メトリクス
D1 全体的に遅れている	M1 演習ごとの総行数
D2 特定の問題で詰まっている	M2 課題実装時間
D3 特定の時間帯で手が止まっている	M3 編集操作数
D4 エラーで詰まっている	M4 エラー対応時間

3　提案

図 1 に本システムの概要を示す．(1) 学生と講師はシステムにログインし，学生はコーディング，講師はシートマップと呼ばれる学生の遅れを可視化するツールを用いて学生支援を行う．(2) 学生はオンラインエディタ（以下，エディタ）の機能でソースコードを [保存]，[コンパイル]，[実行] し，[チェック待ちに変更] ボタンで講師や TA にチェック待ちを知らせる．(3) エディタは，(2) と同時に記録した編集操作をサーバに送信する．(4) システムは分析結果から支援すべき学生を検知する．(5)(4) の分析結果をシートマップに可視化する．

3.1　支援すべき学生の検知

表 1 に学生の遅れ理由 D1〜D4 を検知するためのメトリクス M1〜M4 を示す．提案するメトリクスについて次に説明する．

M1 演習ごとの総行数

1 回の演習における全ての課題のソースコードの累積行数を示す．

Programming Process Visualization System for Students in Programming Exercise
† Syun SAITO(syuns@hil.cs.teu.ac.jp)
†† Makoto YAMADA(myamada@star.cs.teu.ac.jp)
‡ Hiroshi IGAKI(igaki@ist.osaka-u.ac.jp)
† Akifumi INOUE(akifumi@cs.teu.ac.jp)
School of Computer Science,Tokyo University of Technology (†)
Graduate School of Bionics, Computer and Media Science,Tokyo University of Technology (††)
Graduate School of Information Science and Technology,Osaka University (‡)
1404-1 Katakura, Hachioji, Tokyo 192-0982, Japan
1-5 Yamadaoka, Suita, Osaka 565-0871, Japan

表 2: 学生を支援した状況とその件数

	ランキングに表示されていて自主的に手を挙げていない	ランキングに表示されていて自主的に手を挙げている	ランキングに表示されず自主的に手を挙げている
フォローは必要	17	8	1
フォローは不要	5	0	0

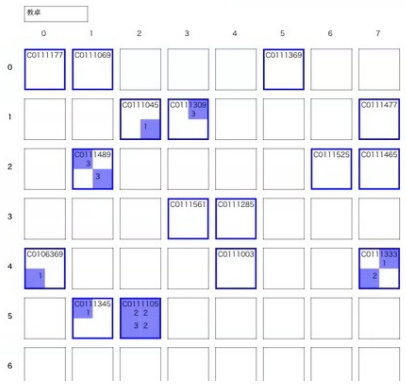

図 2: シートマップによる可視化

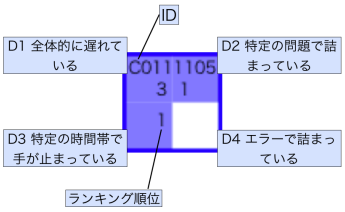

図 3: 遅れ理由 D1〜D4 とシートの対応関係

M2 課題実装時間

特定の課題のエディタを開いた時刻から，課題を解き終わって [チェック済みに変更] ボタンをクリックするまでに，エディタを開いている時間の合計を示す．

M3 編集操作数

特定の課題のソースコードを書いたり消したりする一連の動作数を示す．例えば，以下の操作を行ったとする．(1) エディタに「a」を入力する．(2) エディタから「a」を消す．(3) エディタにクリップボードの「class」を貼り付ける．(4) エディタから「class」を選択して消す．以上の (1)〜(4) のそれぞれが 1 ステップで編集操作数は 4 である．

M4 エラー対応時間

特定の課題のソースコードをコンパイル，実行したときに発生したエラーの継続時間を示す．

3.2 シートマップによる可視化

図 2 に講師が支援すべき学生を決定するときに使用するシートマップを示す．学生の座席位置に表示される**図 3** のシートの 4 つの領域は，遅れ理由 D1〜D4 と対応関係にある．講師は対応する領域の多さとランキング順位から支援する学生を決定する．

4 評価実験

実際のプログラミング演習において，本システムが D1〜D4 を検知できるか評価する実験を行った．対象は学部 1 年生が受講する Java プログラミングの講義である．講義は 90 分× 2 コマで，講師 1 名と TA3 名，学生 16 名で行われた．

4.1 実験手順

学生はエディタを使用して課題を解き，講師，TA は**図 2**，**図 3** のシートマップを閲覧しながら学生の支援にあたる．そのとき，学生の状態 (1. 日時，2. ユーザ ID，3. 課題番号，4. 手を挙げたか，5. フォローが必要だったか，6. シートマップに表示されている D1〜D4 の項目とその順位，7. 自由記入) を記録する．

学生の状態の 4，5，6 から，学生が手を挙げたときと挙げなかった場合のシステムが検出した支援すべき学生の検出率を求める．

4.2 結果

表 2 に学生を支援した状況とその件数を示す．学生が手を挙げて，支援が必要だった件数は 9 件中 8 件で，システムの検出率は 88.8%であった．学生が手を挙げず，システムが検出して講師，TA が学生を支援した件数は 22 件中 17 件で，システムの検出率は 78%であった．

5 おわりに

本稿では，実験結果から提案手法により遅れを検知し，ランキングによる学生間の比較，シートマップに可視化することで，高い検知率で優先して支援すべき学生を発見できることを確認した．今後は，検知率の向上を目指しシステムの改善を行っていきたい．

参考文献

[1] 宮地, 高橋. 構造誤り検出機能を有するアセンブラプログラミング演習支援システムの実現と評価. 信学論, Vol. J91-D, No. 02, pp. 280–292, 2008.

[2] 知見, 坂庭, 櫨山, 宮寺. 失敗知識を利用したプログラミング学習環境の構築. 信学技報. ET, 教育工学, Vol. 104, No. 48, pp. 7–12, 2004.

[3] 藤原, 田口, 島田, 高田, 島川. ストリームデータによる学習者のプログラミング状況把握. DEWS2007 論文集, D9-5, 2007.