

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

加藤 利康^{1,a)} 石川 孝²

受付日 2014年1月29日, 採録日 2014年5月17日

概要: 本論文は, オンライン環境のある教室でのプログラミング演習のための Web ベースの授業支援システムにおいて, 机間巡回では把握が困難なクラス全体と問題のある学生の学習状況を教員が随時に把握できるようにする学習状況把握機能の実現を目的とする. この実現の方法は, 演習課題の配布, 解答プログラムの作成, コンパイル, 実行, および解答の提出を行える授業支援システムを基礎として, プログラミング演習における学習状況把握に対する要求分析と先行研究の調査に基づいて, 機能の設計と評価を行う. クラス全体に対する機能は, 教員があらかじめ用意した模範解答プログラムに基づく正解判定による解答開始から解答提出までの作業進捗集計と, 模範解答プログラムを基準としたコンパイルエラー行の同定によるエラー分類集計である. これらの機能は, コンソール出力を行う Java プログラミングの課題に対して, 十分な精度で適用が可能である. 問題のある学生に対する機能は, 作業進捗の外れ値分析による作業が遅れている学生の検出と, 該当学生の作業履歴の提示である. 検出機能は, 外れ値の基準値を教員が適切に設定することによって, 対処可能な程度に問題のある学生を絞り込むことが可能である. これら 2 種類の学習状況把握機能を付加した授業支援システムは, 実際の演習授業において, システムが提示する情報に基づく学習指導が全指導件数の約半数観察されたことから, プログラミング演習における学習状況把握に有効である. 本論文で実現した学習状況把握機能は, 教員が模範解答プログラムを用意することで, クラス全体の作業進捗とエラー分類を提示し, また, 作業が遅れている学生を検出してその作業履歴を提示することによって, 机間巡回では把握が困難な学習状況を教員が随時に把握することを可能にする.

キーワード: プログラミング演習, 授業支援システム, 学習状況把握, 外れ値分析

Realization of Functions of Assessing Learning Conditions in Learning Management Systems for Programming Practicum

TOSHIYASU KATO^{1,a)} TAKASHI ISHIKAWA²

Received: January 29, 2014, Accepted: May 17, 2014

Abstract: In this paper, we aim to realize some functions related to the assessment of learning conditions in order to help teachers understand the learning situation in their classroom, which can be difficult to grasp. We implement the functions in a Web-based learning management system for programming exercises to be completed in the classroom in an online environment. Using this method, we design and evaluate the functions based on a survey of previous studies and a requirements' analysis for the assessment of learning conditions during programming exercises. This realization of functions is based on the use of a learning management system that enables the distribution of exercises, the creation of an answer program, compilation, running, and submission of answers. The functions for the whole class include a classification error count function and a work progress summary, prepared in advance; both of which are based on the model answer program. These functions can be applied with sufficient accuracy with respect to exercises in the Java language for performing console output. Another function, for individual students, is the presentation of the student's work history and the detection of students whose progress is delayed by an outlier analysis of work progress. This feature can identify students with problems in manageable degree and help teachers properly set reference value for outliers. Teaching guidance, based on information presented by the system in a real class, has been observed in about half of the total number cases. Thus, we can conclude that the learning management system incorporating these features, is effective in helping students learn situational awareness in programming exercises. The functions assessing learning conditions present the error classification summary and progress of the whole class for the reference of the teacher, to allow a model answer program. By presenting students' work history and detecting students whose progress is delayed, this function allows the teacher to grasp at any time, while going around the classroom, any difficulties in the learning conditions.

Keywords: programming practicum, learning management system, assessing learning conditions, outlier analysis

1. はじめに

教員の教育力向上の手段として、情報通信技術 ICT を活用した授業支援システムが利用されている [7], [14], [22]. 授業支援システムは、一般的に授業の資料配布と課題提出などの機能を提供する情報システムの総称である [13]. 授業支援システムを利用する利点は、教員が対面授業では得ることができない個別学生の学習過程や行き詰まり原因の分析などを行えることである [6]. そのため、高等教育機関における授業支援システムの利用率は、年々増加している [22].

高等教育機関におけるプログラミング演習の授業においても種々の授業支援システムが利用されている [10]. プログラミング演習は、学生が個別に演習課題に取り組むため、順調に演習課題を進めている学生と比較して遅延や停滞している学生が多く存在している授業である [23]. そのため、教員には、学生からの質問に対応するだけでなく、理解度不足や諦めによる行き詰まり状態にある学生を把握して、その状況に応じた学習の支援を行うことが望まれている [15]. プログラミング演習の支援に対して教員は、机間巡回に加えて、解答提出と学生画面などの学習状況の確認により学習指導を行っている。プログラミング演習の支援の現状は、教員と TA (Teaching Assistant) の数が限られていることや、学生画面からでは行き詰まっていることが分かりにくいことなどから、クラス全体の作業進度や学習指導を必要としている学生の把握が困難である [2]. プログラミング演習のための授業支援システムには、演習課題の配布、解答プログラムの作成、コンパイルなどの機能がある。しかしながら、対面授業における上記の学習状況を把握する実用的な機能は、4章の先行研究で述べるように、情報処理学会論文誌や電子情報通信学会などの授業支援システムの学習状況把握に対する研究例では実現されていない。

そこで本研究は、オンライン環境のある教室でのプログラミング演習のための Web ベースの授業支援システムにおいて、机間巡回では把握が困難なクラス全体と問題のある学生の学習状況を教員が随時に把握できるようにする学習状況把握機能の実現を目的とする。本研究におけるクラス全体とは、受講者全員のことであり、また、問題のある学生とは、クラス全体と比較して演習課題に対する作業が遅延している学生のことであり、本研究で実現する機能は、プログラミング演習において、クラス全体と個別の学

生に学習指導が必要な情報を教員に提示することで机間巡回では把握が困難な学習状況を把握できるようになり、その学習状況に応じた学習指導を可能にする。

本研究における学習状況把握機能の実現の方法は、プログラミング演習における学習状況把握に対する要求分析と先行研究の調査に基づいて、目標とする機能の設計と評価を行う。学習状況把握は、対面授業において一般的に行われている。そのため、本研究は、一般的な対面授業における学習状況把握に対する要求分析によって目標とする機能を抽出し、その機能に関する先行研究の提案手法から本研究の課題と改良点を明確にする。本研究における学習状況把握機能の実現は、学習状況を教員が随時に把握できるようにするため、プログラミング演習における学生の学習履歴データをリアルタイムに収集できる授業支援システムの利用を前提とする。

本論文は、まず2章でプログラミング演習支援の課題について述べ、3章でプログラミング演習における学習状況把握に対する要求分析に基づいて具体的な機能を定義し、4章でプログラミング演習のための定義した学習状況把握機能に関する先行研究について述べる。次に5章は、クラス全体の学習状況を把握する機能とその評価について記述し、6章で問題のある学生を把握する機能とその評価について記述する。そして7章は、これらの実現した学習状況把握機能を付加した授業支援システムの実際の演習授業での評価について述べ、8章で結論と今後の課題を述べる。

2. プログラミング演習支援の課題

プログラミング演習のための授業支援システムの課題は、対面授業における学習状況把握に関する機能が実現されていないことである。このことから本研究は、一般的な対面授業における学習状況把握に対する要求分析によって実現する機能を抽出する。

初学者を対象としたプログラミング演習の特徴は、学生が個別に演習課題に取り組むため、学生の理解度によって作業進度に差があることが多い [23]. 作業進度に差があるのは、学生がプログラムの書き方やエラーの対処方法が分からずに行き詰まっているためである。行き詰まっている学生は、プログラミング中のトラブルや理解度不足からくる行き詰まり状態の継続により、プログラミングそのものに対するモチベーションが悪化しているといわれている [2]. プログラミング演習は、このような問題のある学生を早い段階で見つけ出し、教員による学習指導を行うことが望まれている [15].

プログラミング演習の支援は、一般的に教員が机間巡回による学習指導を行っているが、クラス全体の作業進度や学習指導を必要としている学生の把握が困難な状況にある [2]. 教員は、解答提出と学生画面などの学習状況の確認により学習指導を行っている。これらの学習状況の把握に

¹ 日本工業大学大学院情報工学専攻
The Graduate School of Engineering, Nippon Institute of Technology, Minami-saitama-gun, Saitama 345-8501, Japan

² 日本工業大学情報工学科
Nippon Institute of Technology, Minami-saitama-gun, Saitama 345-8501, Japan

a) c3115001@cstu.nit.ac.jp

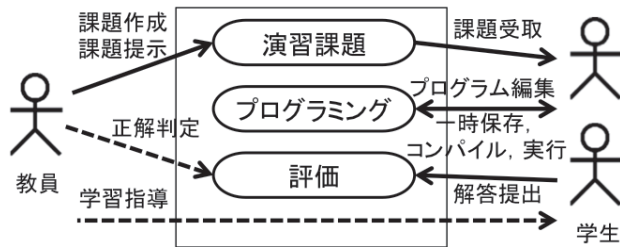


図 1 プログラミング演習支援システムの機能

Fig. 1 Functions in programming practicum support systems.

は、教員と TA の数が限られているためクラス全体を把握できない問題がある。加えて、学生がクラス全体と比較して作業が遅れていること、および行き詰まっていることを把握しにくい問題がある。これらの問題に対する支援方法は、教員の経験や勘に頼る以外に適切な方法を見い出せていないのが現状であるといわれている [15]。一般的には、教員が学生からの質問に対応し、個別学生の事例からクラス全体に対する学習指導を行っている。この現状に対しては、学生が行き詰まってから質問してくる前に、教員が学習状況に応じてつまずくポイントや学習方法に関するアドバイスをを行うことが望まれている [15]。

プログラミング演習のため授業支援システムには、授業の全体的な演習環境を実現するプログラミング演習支援システムと、提出された解答プログラムの評価を支援するプログラミング評価支援システムなどがある [26], [27]。プログラミング演習支援システムの目的は、プログラミング演習に対する教員と学生の作業負担を軽減することである。プログラミング演習支援システムの機能 (図 1) は、教員に対する機能と学生に対する機能で構成される。

- 教員に対する機能
 - 課題作成：演習課題の内容、出題日時を入力して登録する。
 - 課題提示：指定された出題日時の演習課題を提示する。
- 学生に対する機能
 - 課題受取：演習課題を選択して解答を開始する。
 - プログラム編集：演習課題に対する解答プログラムを編集する。
 - 一時保存：解答プログラムを一時保存する。
 - コンパイル：解答プログラムをコンパイルして結果を提示する。
 - 実行：解答プログラムを実行して結果を提示する。
 - 解答提出：解答プログラムを提出して演習課題を完了する。

授業支援システムには、情報処理学会論文誌や電子情報通信学会などの授業支援システムの学習状況把握に対する研究例 10 年分 (2001~2010) の範囲では授業の目的や実施上の制約などから、次の 3 つの課題がある [10]。

(1) 学習の進捗状況を把握できない

授業支援システムは、課題の提出状況を提示しているが、課題に対する進捗状況は途中経過を集計していないため提示していない。

(2) リアルタイムに理解度を把握できない

授業支援システムは、学習履歴データを随時に集計・分析していないため理解度に関する情報を提示していない。

(3) 行き詰まりの原因・場所を把握できない

授業支援システムは、学習履歴データを分析していないため行き詰まりの原因・場所を提示していない。

3. プログラミング演習における学習状況把握に対する要求分析

一般的な対面授業における学習状況把握の目的については、教育学において、次の 3 つの原則がある [17], [19], [20]。

(1) 学習目標の達成状況を把握する

学習目標の達成状況を把握する目的は、クラス全体と個別の学生が取り組んでいる問題を認識し、後続の授業展開の方針を練れるようにすることである。

(2) 学習内容の理解度を把握する

学習内容の理解度を把握する目的は、多くの学生が共通してつまずいているポイントを認識し、クラス全体の学習を整理できるようにすることである。

(3) 学生をつまずきを把握する

学生をつまずきを把握する目的は、問題の解き方が全然分からない学生や途中で行き詰まっている学生の学習過程を認識し、その学生への学習指導を行えるようにすることである。

対面授業における学習状況把握の方法には、机間巡回と小テストの実施があり、授業支援システムの学習状況把握機能も利用されている [10]。授業支援システムの学習状況把握機能には、出欠管理機能、解答提出機能、小テスト機能などがある。出欠管理機能は、授業支援システムに対する学生のログイン情報を集計して提示するため、学生がログインするだけで受講者の確認を可能にする。解答提出機能は、学生からの課題やレポートなどのファイルを受け付けて学生ごとに集計して提示するため、進捗状況の確認を可能にする。小テスト機能は、システム上において小テストの実施を可能にすることで解答を自動的に採点・集計して正答率を提示するため、小テストの終了直後における理解度の確認を可能にする。

本研究におけるプログラミング演習のための学習状況把握機能は、教育学の原則に基づいて、次の 4 つの機能とする。

(1) 作業進捗集計機能

作業進捗集計機能は、教育学の原則における学習目標の達成状況を本研究では演習課題に対する作業進捗と定義して提示する。演習課題に対する作業進捗に対応させたのは、

プログラミング演習では演習課題を解くことが学習目標であり、その演習課題に対して実行までできているのか、コンパイルで止まっているのかなどが達成状況となるためである。この機能に対する要求は、教員がクラス全体の演習課題に対する作業進捗を把握できるようにすることである。

(2) エラー分類集計機能

エラー分類集計機能は、教育学の原則における学習内容の理解度を本研究ではクラス全体のエラー状況と定義して提示する。クラス全体のエラー状況に対応させたのは、多くの学生が共通してつまづいているポイントが、プログラミング演習ではプログラミングにおけるエラーとなるためである。この機能に対する要求は、教員がクラス全体で発生している共通するエラーを把握できるようにすることである。

(3) 問題のある学生の検出機能

問題のある学生の検出機能は、教育学の原則における学生をつまづきを本研究では作業が遅れている学生と定義して提示する。作業が遅れている学生に対応させたのは、問題の解き方が全然分からない学生や途中で行き詰まっている学生が、プログラミング演習では演習課題に対する作業がクラス全体と比較して遅れているためである。この機能に対する要求は、教員が問題のある学生を把握できるようにすることである。

(4) 作業履歴の提示機能

作業履歴の提示機能は、学生をつまづきにおける学習過程を本研究では作業履歴と定義して提示する。作業履歴に対応させたのは、学習過程が、プログラミング演習では演習課題に対する作業内容と作業結果となるためである。この機能に対する要求は、教員が学生の作業履歴を把握できるようにすることである。

これらの機能に対して本研究は、先行研究を調査して本研究が目的とする学習状況を教員が随時に把握できるように設計する。

4. 目標とする学習状況把握機能に関する先行研究

先行研究は、その目的によって本研究が目標とする4つの機能に分類して、先行研究の提案手法における課題と改良点からそれぞれの機能に対する目標を明確にする。

(1) 作業進捗集計機能

宮地らの提案手法 [24] は、専用の言語で記述した模範解答と模範解答プログラムを教員があらかじめ用意し、模範解答プログラムによって提出された解答プログラムを自動的に正解判定して正誤情報を教員に提示する。その有効性は、クラス全体の解答プログラムの正誤を把握して、理解度不足の学生に対する学習指導を支援できることである。

本研究における作業進捗集計機能の課題は、演習課題に対する解答開始から解答提出までの作業進捗を把握できる

ようにすることである。解答開始から解答提出までの作業進捗を対象とした理由は、コンパイルや実行で継続的なエラーが発生しており、次の作業に移れず意欲を失ってしまうといわれている [15] 学生を早い段階で見つけるためである。具体的な改良点は、解答開始、コンパイル、実行、正解、解答提出の各作業における作業進捗を把握できるようにする。作業進捗における正解は、教員があらかじめ用意した模範解答プログラムに基づき正解判定を行えるようにする。

(2) エラー分類集計機能

倉澤らの提案手法 [11] は、過去のコンパイルエラーからエラー要因の候補を教員があらかじめ分析して用意し、演習授業中のエラーメッセージとエラー要因の候補からエラーの原因と場所を推定・集計して教員に提示する。その有効性は、共通するコンパイルエラーの傾向を教員に提示して同時に複数の学生に対する学習指導をできるようにすることである。

長らの提案手法 [18] は、授業の学習内容に合わせたエラーのパターンとパターンに対応するアドバイスを教員があらかじめ用意し、学生が発生させたエラーに対応するアドバイスを自動的に学生に提示する。その有効性は、発生したエラーの件数と内容を教員が確認することでクラス全体のコンパイルエラーの傾向を把握できることである。

市村らの提案手法 [4] は、プログラミング演習中のエラーを取得し、エラー種類で分類集計を行ってエラー種類と件数、およびエラーに対する解決時間を教員に提示する。その有効性は、教員が多くの学生が共通にかかえる問題を発見し、問題の解決に向けた学習指導ができることである。

本研究におけるエラー分類集計機能の課題は、教員によるエラーの分析を必要とせずに機能を実行したときのクラス全体で発生している直接の原因となるコンパイルエラーを把握できるようにすることである。直接の原因となるコンパイルエラーを対象とした理由は、1人の学生が一度に複数のエラーを発生させた場合に教員がすべてのエラーを分析する必要があるためである。具体的な改良点は、エラーメッセージの先頭のエラーを抽出して解答プログラムと模範解答プログラムの差分を用いてそのエラーの場所を推定・集計できるようにする。

市村らの先行研究 [4] は、プログラミング演習中のエラーを分類集計する点で共通しているが、市村らの提案手法がエラー種類だけを分類するのに対して、本研究の分類集計はコンパイルエラーを模範解答プログラムの行に対応させて分類する点が異なる。さらに、本研究の手法は模範解答プログラムを用意するだけで、コンパイルエラーの内容を詳細に分類できる。

(3) 問題のある学生の検出機能

藤原らの提案手法 [21] は、解答プログラムに対するコンパイル回数と実行回数の履歴を組み合わせた作業遅れのバ

ターンを教員があらかじめ分析して用意し、演習授業中に一定時間ごとにコンパイル回数と実行回数を収集してパターンに一致する学生の情報を教員に提示する。その有効性は、コンパイルと実行の作業が遅れている学生を教員が把握できることである。

大城らの提案手法 [8] は、演習課題ごとに提出された解答プログラムを集計し、学生の座席位置に合わせて個別の作業進捗を教員に提示する。その有効性は、解答提出の作業が遅れている学生を教員が把握できることである。

市村らの提案手法 [4] は、操作ログとエラーログから5分以上同じエラーが継続している学生と10分以上作業が行われていない学生を教員に提示する。その有効性は、問題のある学生を手が上がる前に発見し、問題の解決に向けた学習指導ができることである。

井垣らの提案手法 [2] は、解答プログラムの行数と解答開始から解答提出までの作業時間などを集計し、作業進捗の遅延ランキングを行って教員に提示する。その有効性は、作業が遅れていて学習指導を必要とする学生を早期に効率よく発見できることである。

匂坂らの提案手法 [16] は、教員があらかじめ用意したWebのテストを学生に解答してもらい、その解答から診断モデルの生成を行う。次に、この手法は、基本文法を理解していないことや、if文は理解しているがfor文は理解していないなどの分類された特徴とアドバイスを学生に提示する。その有効性は、学習における具体的な改善箇所や学習のポイントを確認できることである。

本研究における問題のある学生の検出機能の課題は、教員による作業遅れのパターンの分析を必要とせずに解答開始から解答提出までの各作業で遅れている学生を把握できるようにすることである。具体的な改良点は、解答開始、一時保存、コンパイル、実行、解答提出の各作業を分析対象として、各作業の完了時刻を統計分析することで作業が遅れている学生を検出できるようにする。

市村らの先行研究 [4] は、作業時間を分析対象とする点は共通しているが、市村らの提案手法が一定時間を基準にして検出するのに対して、本研究の検出機能は作業完了時刻を外れ値分析している点が異なる。また、井垣らの先行研究 [2] は、作業時間を分析対象とする点で共通しているが、井垣らの提案手法がエラー継続時間を一定値との比較によって検出するのに対して、本研究の検出機能は作業完了時刻を外れ値分析している点が異なる。本研究の手法は、コンパイル、実行、提出の各作業について統計的に有意に遅れている学生を発見できる。さらに、匂坂らの先行研究 [16] は、Webのテスト結果からシステムが理解度に合ったアドバイスを学生に提示するのに対して、本研究の提案機能はエラーログ・操作ログから学生が共通にかかえる問題や行き詰まっている学生を教員に提示する点が異なる。

(4) 作業履歴の提示機能

倉田らの提案手法 [12] は、模範解答プログラムを教員があらかじめ用意し、提出された解答プログラムの実行結果と模範解答プログラムの実行結果を自動的に正解判定して出来具合を蓄積し、作業履歴として教員に提示する。その有効性は、作業履歴を確認して、理解度不足の学生に対する学習指導をできるようにすることである。

本研究における作業履歴の提示機能の課題は、演習課題に対する解答開始から解答提出までの作業内容と作業結果を把握できるようにすることである。具体的な改良点は、解答開始、一時保存、コンパイル、実行、解答提出の各作業における解答プログラムとその実行結果を時系列に提示できるようにする。

本研究が目標とするクラス全体の学習状況を把握する機能は、作業進捗集計機能とエラー分類集計機能とする。問題のある学生を把握する機能は、問題のある学生の検出機能と作業履歴の提示機能とする。以降の章は、これらの機能の実現について詳述する。

5. クラス全体の学習状況を把握する機能の実現

クラス全体の学習状況を把握する機能の実現の方法は、作業進捗集計機能とエラー分類集計機能について、それぞれの要求定義から機能設計を行い、アルゴリズムを定義して実装と評価を行う。

5.1 作業進捗集計機能

(1) 要件定義

- 使用目的：教員がクラス全体の演習課題に対する作業進捗を把握できるようにすること。ここでの作業進捗とは、解答開始、コンパイル、実行、正解、解答提出の各作業における人数のことである。
- 機能要件：演習課題に対する解答開始、コンパイル、実行、正解、解答提出の各作業における人数とその学生を提示できるようにすること
- 性能要件：機能要件における各作業の人数を集計できるようにすること

(2) 機能設計

作業進捗集計機能の目的は、演習課題に対する各作業を終えた人数とその学生を随時に提示することである。そのため、作業進捗集計機能は、教員が機能にアクセスしたときに学生ごとの作業完了時刻を集計する。作業完了時刻の集計には、学生が各作業を行ったときに授業支援システムが収集している作業時刻を利用する。また、作業における学生の提示内容は、教員が学習指導を行えるようにするため、各作業完了時刻に加えて学生の学籍番号と氏名、および座席番号とする。学生の学籍番号と氏名、および座席番号は、授業支援システムのログイン時に入力してもらうこ

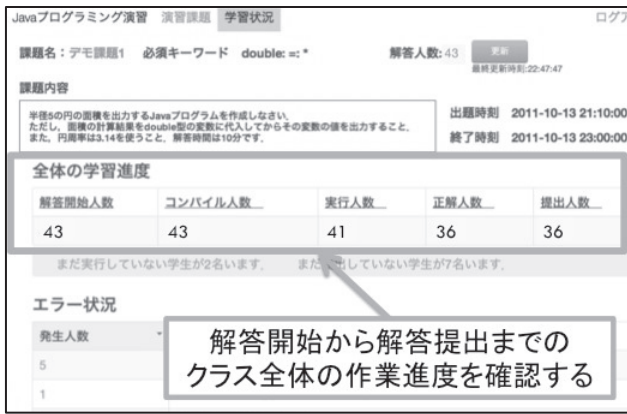


図 2 学習状況画面における作業進度

Fig. 2 Works progress in the learning conditions screen.

とで取得できるようにする。

なお、正解の作業に対する作業完了の定義は、学生の解答プログラムと実行結果を教員があらかじめ用意したキーワードと模範解答プログラムの実行結果によって正解判定を行い、等しい場合とする。正解判定の例は、演習課題の内容が for 文を使って 1 から 10 までの合計を出力する場合、キーワードに for を設定した模範解答プログラムによって、解答プログラムに for が使われており実行結果が 55 の場合に正解となる。この正解判定は、出力結果が得られて文法を問うような演習課題に適用することができる。

(3) アルゴリズム

作業完了時刻の集計アルゴリズムは、各作業の終了人数を随時に提示するため、下記とする。

1. 教員が機能にアクセスしたときに処理を開始する。
2. クラス全体の各作業完了時刻集合を入力する。
3. 各作業完了時刻の数を合計する。
4. 各作業の合計数を出力する。

(4) 実装

作業進度集計機能のユーザインタフェースは、演習課題に対する各作業を終えた人数とその学生を提示するため、以下とする。

- 学習状況画面 (図 2)：解答開始、コンパイル、実行、正解、解答提出の作業人数を提示する。提示された項目の人数をクリックすると、分類別学習状況画面へ移行する。
- 分類別学習状況画面 (図 3)：学習状況画面の各項目に該当する学生の学籍番号、氏名、座席番号、作業完了時刻を提示する。学籍番号をクリックすると、作業履歴画面 (後述 6.2 節) へ移行する。

(5) 評価

作業進度集計機能における妥当性は、機能が提示する情報により、演習課題に対する解答開始、コンパイル、実行、正解、解答提出の作業進度を教員が把握できるかを検証する。検証方法は、実際のプログラミング演習の授業担当教



図 3 分類別学習状況画面

Fig. 3 Classification-based learning conditions screen.

表 1 作業進度の把握に対するインタビュー

Table 1 Interview results for grasp of the work progress.

	教員 A	教員 B
質問：作業進度集計機能の提示内容に対する意見をお聞かせください。	演習課題の未着手者、未完了者を容易に調べることができる。	クラス全体の作業進度を把握しながら課題ごとのヒントが出せる。

員 A, B に学習状況画面の作業進度集計機能について試用評価してもらい作業進度の把握に対するインタビューを行う。インタビューの結果 (表 1) において提示内容を肯定する回答を得られたことから、本機能は妥当である。

5.2 エラー分類集計機能

(1) 要件定義

- a. 使用目的：教員がクラス全体で発生している共通するエラーを把握できるようにすること。共通するエラーとは、エラー種類が等しいことである。エラー種類とは、エラーメッセージに含まれているエラー内容を具体的に示す文言である。
- b. 機能要件：クラス全体に共通するコンパイルエラーの件数とエラー種類を提示できるようにすること
- c. 性能要件：直接の原因となるコンパイルエラーの分類集計をできるようにすること。直接の原因とは、先頭のエラーのことである。

(2) 機能設計

エラー分類集計機能の目的は、解答プログラムとコンパイルのエラーメッセージ、および模範解答プログラムから、随時にエラー分類を行い、集計して提示することである。そのため、エラー分類集計機能は、教員が機能にアクセスしたときにエラー分類を行い、集計する。エラー分類は、共通するコンパイルエラーの場所を推定するためにコンパイルエラー行の同定としてエラーメッセージの先頭のエラー行が示す行番号と模範解答プログラムの行番号を対応させるようにする (以下、対応行番号と呼ぶ)。先頭のエラーを対象とする理由は、最初のエラーが他のエラー要因

にもなっているためである。エラー分類の対象言語は、評価実験の授業が Java 言語を用いているため、Java 言語とする。エラー分類の分析対象は、機能を実行したときに発生しているエラーを取得するため、コンパイルの次の作業である実行に進んでいない学生の最後に発生したコンパイルエラーである。分析対象をコンパイルエラーとした理由は、プログラミング演習において教員による学習指導の必要性があるといわれているためである [11]。

エラー分類におけるコンパイルエラー行の同定方法は、Unix の diff [1] によるテキストの差分（行の対応が可能）をとる方法を採用する。diff におけるコマンドは、複数のスペースやタブを 1 つとカウントする -b オプションにより「diff -b 解答プログラム 模範解答プログラム」とする。テキストの差分をとる方法は、複数の学生が異なるエラー行でエラーが発生していても、同じエラー内容として分類できる。また、対応行番号は、diff の結果において「解答プログラムのエラー行 c 模範解答プログラムの行」（変更 change）とする。エラー分類集計機能の特徴は、コンパイルエラーのある行を模範解答プログラムの対応する行で分類するため、学生の解答プログラムの書き方によらずに、エラー分類をエラー行とエラー種類の組で同定できることである。このエラー分類は、エラーメッセージにエラーの行番号とエラー種類が含まれていればエラー分類ができるため、C 言語や C++ 言語などのプログラム言語にも応用できる。

(3) アルゴリズム

エラー分類のアルゴリズムは、クラス全体に共通するコンパイルエラーを随時に提示するため、下記とする。

1. 教員が機能にアクセスしたときに処理を開始する。
2. 解答プログラムとエラーメッセージ、および模範解答プログラムを入力する。
3. エラーメッセージから、先頭のエラー行が示す行番号とエラー種類を抽出する。
4. 解答プログラムの行を模範解答プログラムの行に、文字列の差分を取り、エラー行に対応する模範解答プログラムの対応行番号を抽出する。
5. 対応行番号が存在しなければ、対応行番号を不明とする。
6. 対応行番号とエラー種類の対を出力する。

(4) 実装

エラー分類集計機能のユーザインタフェースは、クラス全体に共通するコンパイルエラーを随時に提示するため、以下とする。

学習状況画面（図 4）：エラー状況として対応行番号とエラー種類の対を集計数順に提示する。提示された項目の数をクリックすると、分類別学習状況画面（図 3）へ移行する。



図 4 学習状況画面におけるエラー分類集計

Fig. 4 Error classification aggregation in the learning status screen.

表 2 エラー分類の精度

Table 2 Accuracy of the error classification.

解答プログラム数	正解	不正解	精度
5708	5616	92	98.39%

(5) 評価

エラー分類集計機能における妥当性は、クラス全体に共通して発生しているコンパイルエラーが分類できているかを計測するため、エラー分類の精度によって検証する。検証方法は、Java 言語のコンソール出力を行うプログラミングの課題に対するコンパイルエラーのある 5,708 個の解答プログラムをエラー分類して、対応行番号が示した模範解答プログラムの行とエラーの行番号が示した解答プログラムの行が一致する精度を計測する。計測の結果、エラー分類の対応は、98.39%であった（表 2）。エラー分類集計機能は、コンソール出力を行う Java プログラミングの課題に対して、十分な精度で適用が可能である。また、表 2 における不正解の例は、for 文のブロックにおいて、“{”に対応する“}”を入力し忘れた場合である。この場合、エラー行は、Java のコンパイラにおいて、解答プログラムの“{”に対応する最後の行番号が示される。そのため、エラー分類は、実際にエラーのない行で模範解答プログラムとの比較を行ってコンパイルエラー行を同定することに失敗する。したがって、エラー分類の限界は、コンパイラが示すエラーの行番号が実際にエラーのない行の場合にエラー分類が失敗するため、コンパイラが示すエラーの行番号によってエラー分類の成否が決まることである。

6. 問題のある学生を把握する機能の実現

問題のある学生を把握する機能の実現の方法は、問題のある学生の検出機能と作業履歴の提示機能について、それぞれの要求定義から機能設計を行い、アルゴリズムを定義して実装と評価を行う。

6.1 問題のある学生の検出機能

(1) 要件定義

- 使用目的：教員が問題のある学生を把握できるようにすること。問題のある学生とは、クラス全体と比較して作業が遅れている受講生のことである。クラス全体と比較して作業が遅れているとは、クラス全体で相対的に作業を行っていないことである。
- 機能要件：演習課題に対する解答開始、一時保存、コンパイル、実行、解答提出において、クラス全体と比較して作業が遅れている学生の人数を提示できるようにすること
- 性能要件：機能要件における作業が遅れている学生を検出できるようにすること。

(2) 機能設計

問題のある学生の検出機能の目的は、作業が遅れている学生を随時に検出して提示することである。そのため、問題のある学生の検出機能は、クラス全体の作業完了時刻の分布に対して、機能を実行したときの現在時刻が統計的に外れ値であるときに作業が完了していない学生を検出する。現在時刻を外れ値としている理由は、まだ作業を行っていない学生についてはその作業時刻データがないためである。したがって外れ値の現在時刻は、作業を完了していない学生が遅れていると判断できる。外れ値による検出には、しきい値による方法があるが、作業時刻に応じて必ず一定の作業の遅れが検出されてしまう問題がある。作業の遅れは、演習課題に学生が個別に取り組むため、必要以上の学習指導は行わずにクラス全体と比較して異常に作業が遅い学生に対して検出する必要があるといわれている [5]。

外れ値の分析 [25] には、作業の遅れを作業の経過時間による外れ値とみなすためにスミルノフ・グラブス検定を利用する。スミルノフ・グラブス検定は、正規分布すると予想される標本値のうち、最大または最小の値が外れているかどうかを検定する手法である [3], [9]。本研究は、作業の完了時刻に現在時刻を追加して正規分布と仮定することで現在時刻が外れ値と判定された場合に、作業を開始していない学生を問題があると定義する。作業時刻による問題のある学生の検出には、提出において学生が早期に提出した場合、残り全員の未提出者が作業の遅れとして検出されてしまう問題がある。しかしながら、この検出は、提出者が現れるたびに検出量が変わるため、クラス全体の作業進度に応じて異常に作業が遅れている学生を提示することができる。

(3) アルゴリズム

問題のある学生の検出アルゴリズムは、作業が完了していない学生を随時に検出するため、下記とする。

- 教員が機能にアクセスしたときに処理を開始する。
- 現在時刻と作業時刻集合および、有意水準を入力する。

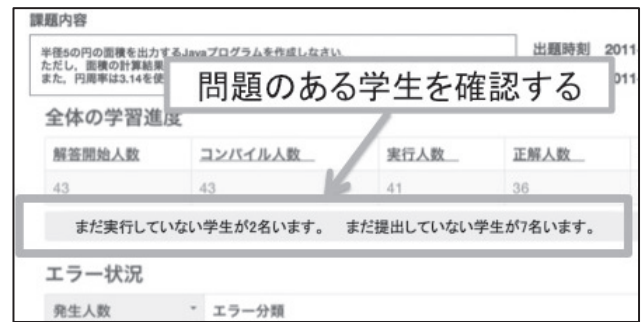


図 5 学習状況画面における問題のある学生

Fig. 5 Stuck students in the learning conditions screen.

- 作業ごとの作業時刻集合に現在時刻を追加して外れ値分析を行う。
- スミルノフ棄却検定表 [9] と作業時刻集合の件数から、外れ基準値を算出する。
- 各作業時刻について検定統計量を求める。
- 検定統計量の最大値を抽出する。
- 検定統計量の最大値が外れ基準値より大きいならばその作業時刻を外れ値とする。
- 外れ値が現在時刻でなければ、このアルゴリズムを終了する。
- 作業時刻集合に作業時刻が存在しない学生に関する情報を出力する。

(4) 実装

問題のある学生の検出機能のユーザインタフェースは、作業が遅れている学生を提示するため、以下とする。

学習状況画面 (図 5)：作業が遅れている学生がいる場合、その作業名と人数を提示する。提示された項目の人数をクリックすると、分類別学習状況画面 (図 3) へ移行する。

(5) 評価

問題のある学生の検出機能の妥当性は、クラス全体の作業完了時刻の分布に対して作業が完了していない学生を検出できるかを計測するため、有意水準の変化による外れ値検出回数によって検証する。検証方法は、実際の授業で収集した学生が作業を行った時点における作業履歴データに対して、有意水準を 5% から 40% の範囲で変化させたときの外れ値検出回数を計測する。計測の結果、外れ値検出回数は、有意水準に比例して増加した (図 6)。この結果から、問題のある学生の検出機能は、有意水準を教員が適切に設定することによって対処可能な程度に問題のある学生の絞り込みが可能である。

6.2 作業履歴の提示機能

(1) 要件定義

- 使用目的：教員が学生の作業履歴を把握できるようにすること。ここでの作業履歴とは、作業時刻を含む作業内容と作業結果のことである。作業内容とは、

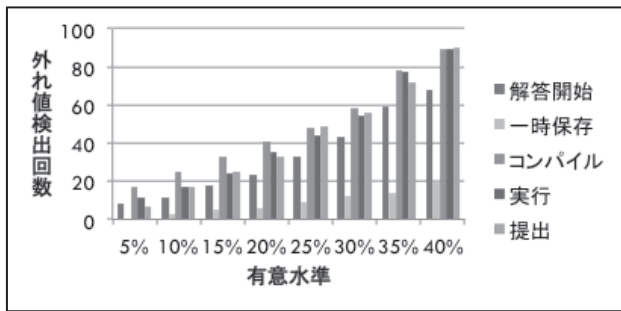


図 6 有意水準の変化による外れ値検出回数

Fig. 6 Numbers of outlier detection in the change of significance level.

各作業における解答プログラムのことである。作業結果とは、解答プログラムに対する実行結果のことである。

- b. 機能要件：学生が解答開始してから、教員が機能を実行したときまでの演習課題に対する各作業の作業内容と作業結果を時系列に提示すること
- c. 性能要件：作業内容と作業結果を集計できるようにすること。

(2) 機能設計

作業履歴の提示機能の目的は、作業時刻と作業内容、および作業結果を、学生の氏名と学籍番号、および座席番号と合わせて時系列に随時に提示することである。そのため、作業履歴の提示機能は、教員がアクセスしたときに最新の作業履歴が先頭となるように作業時刻をもとに降順に並び替えて作業履歴を提示する。この作業履歴は、作業ごとに提示されるようにする。なお、作業内容と作業結果は、解答開始と一時保存などの作業においてデータがない場合、ハイフン記号とする。

(3) アルゴリズム

作業履歴の提示アルゴリズムは、学生が解答開始してから、教員が機能を実行したときまでの作業履歴を提示するため、下記とする。

1. 教員が機能にアクセスしたときに処理を開始する。
2. 学生の演習課題に対するすべての作業時刻と作業内容、および作業結果を入力する。
3. 作業内容あるいは作業結果にデータがなければ、それぞれにハイフン記号を設定する。
4. 作業時刻と作業内容、および作業結果を組として作業時刻をもとに降順に並べ替える。
5. 並べ替えた作業時刻と作業内容、および作業結果を出力する。

(4) 実装

作業履歴の提示機能のユーザインタフェースは、作業ごとに作業履歴を提示するため、以下とする。

作業履歴画面 (図 7)：学生の演習課題に対する解答開始、一時保存、コンパイル、実行、解答提出の作業時刻、



図 7 作業履歴画面

Fig. 7 Work history screen.

表 3 作業履歴の把握に対するインタビュー

Table 3 Interview results for grasp of the work history.

	教員 A	教員 B
質問：作業履歴の提示機能の提示内容に対する意見を教えてください。	作業時刻をもとに、学生がどのような作業を行っているのかを把握できる。	同じエラーを何回も発生させていることが把握できる。

および解答プログラムと作業結果を時系列に提示する。提示された作業時刻をクリックすると、その作業内容と作業結果が提示される。

(5) 評価

作業履歴の提示機能における妥当性は、機能が提示する情報により、学生の作業履歴を教員が把握できるかを検証する。検証方法は、実際のプログラミング演習の授業担当教員 A, B に作業履歴画面を試用評価してもらい作業履歴の把握に対するインタビューを行う。インタビューの結果 (表 3) において提示内容を肯定する回答を得られたことから、本機能は、妥当である。

7. 実現した学習状況把握機能を付加した授業支援システムの評価実験

7.1 実験の目的

実験の目的は、実現した学習状況把握機能 (以下、実現機能と呼ぶ) の実際の授業における有用性と有効性を評価することである。有用性の評価は、実現機能が実際の授業で授業担当教員に使われるかどうかの計測と、実現機能の有用性に対する授業担当教員の主観的評価によって行う。有効性の評価は、実現機能が授業担当教員による学習指導につながるかどうかの計測と、実現機能の有効性に対する授業担当教員の主観的評価によって行う。

表 4 実験における授業科目の学習内容と演習課題の数

Table 4 The numbers of exercises issue and learning content, in the experiment.

学習内容	演習課題の数
算術演算子と式	3
条件分岐	4
論理演算子	4
繰り返し	4

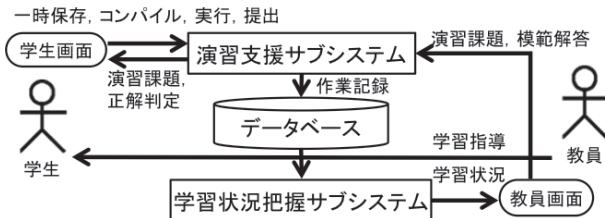


図 8 実験システムの構成

Fig. 8 Structure of the experimental system.

7.2 実験の方法

(1) 授業概要

実際の授業とする授業科目は、Java 言語によるコンソールアプリケーションを中心に学習する大学 1 年次の「プログラミング基礎・演習」である。実験における授業科目の学習内容と演習課題の数は、表 4 である。この授業科目は、クラス数が 2 つ、教員が各 1 人、TA がそれぞれ 2 人と 4 人、学生がそれぞれ 38 人と 73 人である。

(2) 実験システム

実験システムは、プログラミング演習において実現機能を教員に提供できるようにする。実験システムの構成は、2 章で述べた教員に対する機能と学生に対する機能を提供する演習支援と、実現機能を提供する学習状況把握の 2 つのサブシステムとデータベースである (図 8)。この実験システムには、プログラミング演習における学生の学習履歴データをリアルタイムに収集できる Web ベースのクライアントサーバ方式を利用する。実験システムでは、クライアントが学生画面と教員画面であり、サーバが演習支援サブシステムとデータベース、および学習状況把握サブシステムである。クライアントの学生画面は、入力された解答プログラムに対して一時保存、コンパイル、実行、提出などをサーバに依頼する (図 9)。クライアントの教員画面は、演習課題と模範解答プログラムをサーバに送信して結果を受け取る。サーバは、受信したプログラムに対してコンパイルや実行を行い、データベースに作業記録し、クライアントに結果を返す。

実験システムのプログラム言語は、PHP, HTML, Ajax, JavaScript, MySQL である。データベースは、以下の 7 つの表から構成される (図 10)。

- 教員：教員の ID とパスワード



図 9 学生画面

Fig. 9 Student screen.

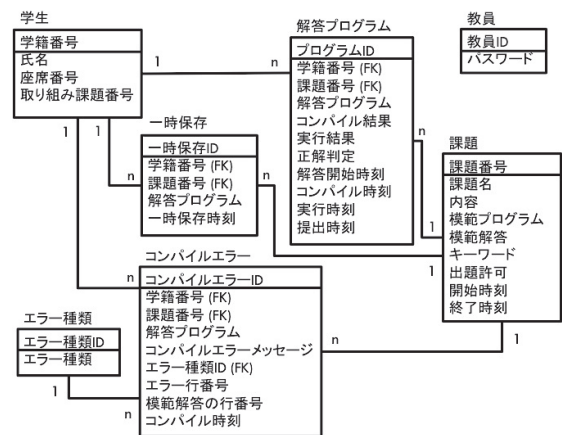


図 10 表間の関係

Fig. 10 Table relationships.

- 学生：学生の学籍番号，氏名，座席番号
- 課題：課題名，課題内容，開始時刻，終了時刻
- 解答プログラム：学生の学籍番号，プログラムのソースコード，およびコンパイル，実行，正解判定，提出についての結果と作業時刻
- 一時保存：解答プログラムのソースコードと一時保存時刻
- コンパイルエラー：コンパイルエラーの先頭のエラー行に対応する模範解答プログラムの行番号とエラー種類の ID
- エラー種類：エラー種類の ID とエラー種類の内容

(3) 実験条件

実験回数は、実現機能による利用者の慣れと効果の有無を考慮して、同じ学習内容について実現機能を使った授業と使わない授業に分けて交互に 2 回ずつ計 4 回とする。実験の環境は、教卓に教員用の PC が 1 台と、教室内に学生用の PC が 1 人 1 台、およびすべての PC がネットワークを介してアクセスできる授業支援システムで構成される。

問題のある学生の検出機能における有意水準は、25% 固定とする。有意水準を 25% に設定したのは、6 章の問題のある学生の検出機能の評価において、外れ値が検出される

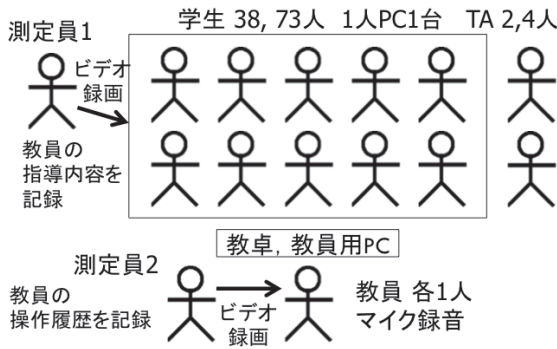


図 11 計測環境
Fig. 11 Measurement environment.

ときとまったく検出されないときの中間の値だったためである。

(4) 計測方法

実現機能の有用性に対する計測項目は、実現機能が教員に使われるかを測るため、実現機能の提示画面を指導の原因とした指導回数とする。また、実現機能の有効性に対する計測項目は、学習指導につながるかを測るため、指導の結果における指導内容に基づく実現機能ごとの指導回数とする。指導の原因と結果の計測方法は、教員の音声録音および、授業風景と教員画面の操作履歴をビデオ録画する方法を採用する(図 11)。指導の原因は、学習状況画面、作業履歴画面、巡回、挙手のいずれかとする。指導の結果は、一斉指導か個別指導の分類と指導の具体的な内容の対である。なお、教員が TA に指示した指導は記録するが、TA 自らによる指導は記録しない。また、指導後に続けて質問されて指導した内容は、記録しない。加えて、実験の最後には、実現機能の有用性と有効性に対する教員の主観的評価についてのインタビューを行う。

7.3 実験の結果

実現機能の提示画面を指導の原因とした学習指導が 30 回あった(表 5)。作業進捗集計機能が提示するクラス全体の作業進捗に対する学習指導が 5 回あった(表 5 における※ 1)。エラー分類集計機能が提示するクラス全体に共通するエラーに対する学習指導が 3 回あった(表 5 における※ 2)。問題のある学生の検出機能が提示する問題のある学生に対する学習指導が 2 回あった(表 5 における※ 3)。作業履歴の提示機能が提示する作業履歴画面の確認後による学習指導が 22 回あった。なお、表 5 における()内の数値は、共通する指導内容の回数である。また、指導の原因が巡回と挙手による学習指導は、実現機能を使わない場合が 64 回、使った場合が 33 回あった(表 6)。

有用性と有効性に対するインタビューの結果は、次のとおりであった。

- 質問：学習状況把握機能について良かった点と悪かった点を教えてください。

表 5 実現機能の提示画面による指導の原因と結果

Table 5 Contents of the cause and effect of guidance by the presentation screen of the proposed function.

原因	学習状況把握画面		作業履歴画面	
	分類	一斉	一斉	個別
結果	主な指導内容	<ul style="list-style-type: none"> ・全体の作業進捗から遅れている学生へサポートの要求を促す※1(5) ・論理演算子を用いること※2(2) 	文字列の連結方法を確認すること※2	コンパイルの時間間隔が長い学生を促す※3(2)
指導回数		8	2	20
小計		8	22	
合計		30		

※1, 2, 3 は、従来行われなかった学習指導
()内は共通する指導内容の回数

表 6 巡回と挙手による指導回数

Table 6 Numbers of guidance by a hand rising and patrol.

実現機能		なし	あり
原因	巡回	42	23
	挙手	22	10
合計		64	33

- 教員 A の回答：
 - 良かった点
 - 演習の解答開始から解答提出の確認までがシステムとして自動化されているので演習課題の指示、解答提出の確認の負担が減る。
 - コンパイル時のエラー状況が把握できるので、学生がどのような状況で課題に行き詰っているのかを把握でき、学習指導する前の事前情報として利用できる。
 - 悪かった点
 - 特にない。
- 教員 B の回答：
 - 良かった点
 - 人数の多いコンパイルエラーについて一斉にコメントが行える。
 - 学習状況が教壇からリモートで確認できるので、内容をもとに TA に指導の指示が出せる。
 - 作業履歴の情報により、作業が遅れている具体的な原因が分かる。
 - 悪かった点
 - 学習状況画面の更新をしなくても、状況の変化があった場合にメッセージが提示されると把握が簡単になる。
 - 教室の座席位置に合わせて各学生の学習状況を提示されると学習指導しやすい。
- 質問：学習状況把握機能の提示内容について、学習指

表 7 教員からの指導の比率

Table 7 Ratio of guidance from teachers.

実現機能		なし	あり
原因	提示画面と巡回	65.6% (42)	84.1% (53)
	挙手	34.4% (22)	15.9% (10)
指導回数の合計		64	63

() 内は指導の原因に対する指導回数

導に対する効果を教えてください。

- 教員 A の回答：
 - 作業進捗の情報は、作業進捗が全体的に遅れていればクラス全体に対して学習指導が必要な学習状況が分かる。
 - エラー分類集計の情報は、学生の文法に対する理解度不足がクラス全体に共通か個別かを判断できる。
- 教員 B の回答：
 - 問題のある学生の情報は、他の学生より作業が遅れていて学習指導を必要とする学生を発見できる。

7.4 結果の考察

システムの提示する情報に基づく学習指導が、実現機能の提示画面と巡回および挙手による指導回数を合わせた 63 回中 30 回の約半数観察された。加えて、有用性に対するインタビューの結果が、学習指導に役立つ内容を示す回答であった。これらの結果から、実現機能は、学習指導を必要とする有益な情報を提示しており、プログラミング演習における学習状況把握に有効である。

実現機能を使ったときにしか行われなかった学習指導が観察された(表 5 における※ 1, 2, 3)。加えて、有効性に対するインタビューの結果が、学習指導につながる内容を示す回答であった。また、実現機能を使った授業では、使わない授業と比較して教員からの指導の比率が高い(表 7)。表 7 における () 内の数値は、指導の原因に対する指導回数である。表 7 より教員が能動的に指導する比率は、実現機能の提示画面により学習状況を把握できたため、増加した。学生から催促されて指導する比率は、学生からの質問の前に提示画面から問題を発見できたため、減少した。実現機能を使った授業による教員からの指導の比率は、カイ二乗検定による有意水準 1% で有意に高い。これらの結果から、実現機能は、机間巡回では把握が困難な学習状況に対する学習指導を行っていたこと、および学生から助けを求められる前に学習指導を行っていたため、プログラミング演習における学習指導に有効である。

実験システムの改善点として、エラー分類集計機能が提示するコンパイルエラーの集計数には、1 件のエラーの場合に、個別に発生しているのか断続的にクラス全体で発生

しているのかが分からない問題がある。この問題は、現在のエラーの件数や累計件数に対して統計的な外れ値分析を行うことで判断できる。エラー分類集計機能は、機能の提示するコンパイルエラーがクラス全体に共通か個別かを的確に判断することが必要である。

8. おわりに

本論文は、プログラミング演習のための授業支援システムにおいて、机間巡回では把握が困難なクラス全体と問題のある学生の学習状況を教員が随時に把握できるようにする学習状況把握機能の実現について述べた。実現したクラス全体に対する機能は、教員があらかじめ用意した模範解答プログラムに基づく正解判定による解答開始から解答提出までの作業進捗集計と、模範解答プログラムを基準としたコンパイルエラー行の同定によるエラー分類集計である。これらの機能は、コンソール出力を行う Java プログラミングの課題に対して、十分な精度で適用が可能である。また、問題のある学生に対する機能は、作業進捗の外れ値分析による作業が遅れている学生の検出と、該当学生の作業履歴の提示である。検出機能は、外れ値の基準値を教員が適切に設定することによって、対処可能な程度に問題のある学生の絞り込みが可能である。これら 2 種類の学習状況把握機能を付加した授業支援システムは、実際の演習授業において、システムが提示する情報に基づく学習指導が全指導件数の約半数観察されたことから、プログラミング演習における学習状況把握に有効である。

今後の課題は、7.4 節の結果の考察で明らかになったエラー分類集計機能が提示する 1 件のエラーに対する個別かクラス全体の問題かについて、エラーの件数に統計的な分析を行ってクラス全体に共通か個別かを判断できるようにすることである。

謝辞 評価実験に協力していただいた日本工業大学情報工学科の高瀬浩史准教授と勝間田仁講師に感謝の意を表す。また、実験システムの開発に協力してくれた、研究室の鈴木裕司氏、高石晃成氏、吉田真世登氏に感謝の意を表す。

参考文献

- [1] Diffutils, available from (<http://www.gnu.org/software/diffutils/diffutils.html>).
- [2] 井垣 宏, 齊藤 俊, 井上亮文, 中村亮太, 楠本真二: プログラミング演習における進捗状況把握のためのコーディング過程可視化システム C3PV の提案, 情報処理学会論文誌, Vol.54, No.1, pp.330–339 (2013).
- [3] 市原英行, 口井敏匡, 山達正明, 坂口英明, 植村 博, 樹下行三: イメージセンサに対する統計型エラーモデルとそのテスト手法について, 電子情報通信学会論文誌, Vol.J89-D, No.8, pp.1663–1672 (2006).
- [4] 市村 哲, 梶並知記, 平野洋行: 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, 情報処理学会論文誌, Vol.54, No.12, pp.2518–2527 (2013).

- [5] 植野真臣：eラーニングにおける所要時間データの異常値オンライン検出，電子情報通信学会論文誌，Vol.J90-D, No.1, pp.40-51 (2007).
- [6] 植野真臣：eラーニングにおけるデータマイニング，日本教育工学会論文誌，Vol.31, No.3, pp.271-283 (2007).
- [7] 江本理恵：ICTを活用した教育支援システムの導入とファカルティ・ディベロップメント—岩手大学の事例から，国立教育政策研究所紀要第139集 (2010).
- [8] 大城正典，山川裕子，Mackin Kenneth J.：プログラミング学習支援システム CAPTAIN における学習状況把握機能の開発，日本教育工学会研究報告集，Vol.10, No.1, pp.81-84 (2010).
- [9] 奥田千恵子：医薬研究者のためのケース別統計手法の学び方，金芳堂 (1999).
- [10] 加藤利康，石川 孝：優れた授業実践のための7つの原則に基づく授業支援システムの要求分析，情報処理学会コンピュータと教育研究報告，Vol.2011-CE-110, pp.1-8 (2011).
- [11] 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，Vol.104, No.703, pp.19-24 (2005).
- [12] 倉田英和，富永浩之，林 敏浩，垂水浩幸：実行テストによるプログラム判定を用いた初級Cプログラミング演習支援と授業実践，情報処理学会研究報告，Vol.2007-CE-101, pp.11-18 (2007).
- [13] 佐伯 胖：学びとコンピュータハンドブック，東京電機大学出版局，pp.200-203 (2008).
- [14] 清水康敬：ICT活用によるFDの現状とNIMEの取り組み，メディア教育研究，Vol.4, No.1, pp.1-8 (2007).
- [15] 匂坂智子，渡辺成良：プログラミング初学者の学習方略と段階的理解度に関する調査および支援ルールの作成について，教育システム情報学会誌，Vol.26, No.1, pp.5-15 (2009).
- [16] 匂坂智子，渡辺成良：プログラミング初学者のためのWeb-based 学習診断システムの開発と評価 (特集eラーニング環境のデザインと実践運用)—(評価・診断システム)，教育システム情報学会誌，Vol.27, No.1, pp.29-38 (2010).
- [17] 田中耕治：よくわかる授業論，ミネルヴァ書房 (2007).
- [18] 長 慎也，笈 捷彦：proGrep-プログラミング学習履歴検索システム，情報処理学会研究報告，Vol.2005-CE, No.15, pp.29-36 (2005).
- [19] 西澤泰彦：多人数講義における問題点と教育方法，名古屋高等教育研究，Vol.6, pp.45-57 (2006).
- [20] 日本教育工学会：教育工学辞典，実教出版 (2000).
- [21] 藤原理也，田口 浩，島田幸廣，高田秀志，島川博光：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，電子情報通信学会第18回データ工学ワークショップ，pp.1-6 (2007).
- [22] 放送大学学園：平成21年度・22年度文部科学省先導的大学の改革推進委託事業「ICT活用教育の推進に関する調査研究」(2011).
- [23] 堀口悟史，井垣 宏，井上亮文，山田 誠，星 徹，岡田謙一：講義資料閲覧ログを用いたプログラミング講義進捗管理手法の提案，情報処理学会論文誌，Vol.53, No.1, pp.61-71 (2012).
- [24] 宮地恵佑，高橋直久：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，電子情報通信学会論文誌，Vol.J91-D, No.2, pp.280-292 (2008).
- [25] 山西健司：データマイニングによる異常検知，共立出版 (2009).
- [26] 米沢宣義，志村 武，南 敏：多人数初心者向きプログラミング教育システムの基本設計について，情報処理学会論文誌，Vol.27, No.1, pp.96-102 (1986).
- [27] 渡辺博芳，荒井正之，武井恵雄：事例に基づく初等アセ

ンブラプログラミング評価支援システム，情報処理学会論文誌，Vol.42, No.1, pp.99-109 (2001).



加藤 利康 (学生会員)

2001年日本工業大学情報工学科卒業。2003年同大学大学院工学研究科博士前期課程修了。2003年株式会社ノアシステムにてWebシステムの開発に従事。2005年神田情報ビジネス専門学校情報デザイン科教諭。2011年より日本工業大学大学院工学研究科博士後期課程在籍中。ICTを用いた授業支援システムの研究に従事。電子情報通信学会，教育システム情報学会各会員。



石川 孝 (正会員)

1976年東北大学理学部物理学科卒業，1976～1994年ぺんてる株式会社電子研究所ほか，1994～2000年木更津工業高等専門学校情報工学科助教，2000年より日本工業大学情報工学科教授。博士(工学)。専門は機械学習。人工知能学会会員。日本ソフトウェア科学会ネットワークが創発する知能研究会プログラム委員。