Vol.2013-CE-120 No.2
2013/7/6

プログラミング演習支援システムにおける学習状況把握機能の提案

加藤利康[†1]　　石川孝[†2]

多数の学生が個々に PC を使うプログラミング演習の授業を支援するプログラミング演習支援システムでは，演習の課題に対する学生の作業進度や学生が抱える多様な問題などの学習状況を教員がリアルタイムに把握することが課題となっている．本論文は，ネットワーク環境のある教室での授業支援を目的としたプログラミング演習支援システムにおいて，課題の解答開始から提出までの作業ごとのクラス全体の学習状況と個別学生の学習状況に関する情報を教員にリアルタイムに提供する学習状況把握機能を提案する．この提案機能は，クラス全体の作業進度と作業履歴，コンパイルエラー分類，作業進度の遅れ情報を提供する．実際の授業においてプロトタイプシステムを用いて提案機能の有効性を評価した結果，授業中の教員の指導内容の約半数が学習状況把握機能の利用に起因しており，作業進度に合わせた指導を行なっていることから，提案機能はプログラミング演習におけるリアルタイム学習状況把握に有効である．

A Proposal of Functions in Programming Practicum Support Systems for Assessing Learning Conditions

TOSHIYASU KATO[†1]　TAKASHI ISHIKAWA[†2]

That in programming practicum support system, teachers are real-time visibility into the status of student learning and work progress to the problem of exercises is an issue. In this paper, propose a learning condition ascertainment functions provided to teachers to real-time information about the work of individual students and work of the whole class. This proposal function provides a delay information work progress of the whole class, work history, compile error classification, the work progress. The results of the evaluation experiment of effectiveness, about half of the teaching content is due to the use of learning condition ascertainment function in an actual class. In addition, the proposed functions is effective in real-time condition ascertainment in learning programming exercises from the fact that teachers are teaching to fit the work progress.

1. はじめに

授業の質向上の取り組みにおいて授業支援システムが活用されており，本研究はプログラミング演習のための支援システムを対象とする．授業支援システムは，一般的に情報通信技術 ICT を利用して授業を支援するシステムである[6]．授業支援システムには，一般の講義で使用される汎用システムのほかに，語学教育のためのシステムやプログラミング教育のためのプログラミング演習支援システムなどの専用システムがある．プログラミング演習支援システムは，一般にネットワーク環境で PC を使える教室において，教員による演習課題の作成と配布，および学生によるプログラミングと解答の提出などを支援する[10]．また，教員による学生画面のモニターなどの学習状況の把握のための機能も補助的に使われている[9]．

プログラミング演習支援システムは，演習の課題に対する学生の作業進度や学生が抱える多様な問題などの学習状況を教員がリアルタイムに把握することが課題となっている．プログラミング演習は，学生が実際にプログラムを作成することを課題とした授業である．この授業は，講義形式の授業とは違って，多数の学生が非同期で課題に取り組むために，学生が抱える多様な学習上の問題に対して教員がきめ細かく対応することが困難である．そのため授業補助者として学生の TA（Teaching Assistant）を配置することや複数の教員で担当するなどして，学生の学習状況の把握と個別指導を行っている．プログラミング演習のこのような問題に対して，学生の学習状況をリアルタイムで把握したり，学生の学習上の問題点を分析したりすることで教員の学習状況把握を目的とした多くの研究がある[4]．しかし，これまでの研究は，2 章の関連研究で述べるような種々の理由で実用化には至っていない．

そこで本研究は，プログラミング演習支援システムにおいて，演習の課題に対する解答開始から提出までの作業ごとのクラス全体の学習状況と個別学生の学習状況を教員がリアルタイムに把握する学習状況把握機能を実現することを目的とする．学習状況把握機能の実現という課題に対しては，演習の課題に対する作業開始時刻やコンパイルエラーメッセージなどを分類集計して提供することで解決する．具体的には，クラス全体についての課題に対する作業進度と作業履歴を提供する学習状況提示機能，共通的なエラーを提供するコンパイルエラー分類機能，および作業進度の遅れている学生の情報を提供する作業進度の遅れ検出機能を提案する．

本論文は，情報システムの一般的な開発ステップ[7]に従って，提案する機能の定義と分析，設計，実装，評価，および運用について記述する．定義ステップは，関連研究が

†1 日本工業大学大学院
The Graduate School of Engineering, Nippon Institute of Technology
†2 日本工業大学
Nippon Institute of Technology

ⓒ2013 Information Processing Society of Japan

提案する方法を一般化して要求を定義する．分析ステップは，利用者の情報要求を具体化するため，実際の授業を観察して利用シナリオを作成する．設計ステップは，利用シナリオに基づいて要求を実現する機能を定義する．実装ステップは，システム要件に基づいて要求機能を実装する．評価ステップは，実際の授業において要求機能の有効性を検証する．運用ステップは，システムを導入して実際の使用に基づいて改良する．

2. プログラミング演習の学習状況把握

プログラミング演習は，教員が限られた時間の中で多人数の学生を指導しなければならないため学習状況の把握を必要としており，授業における課題の作成から提出までの教員と学生の様々な行動の負担を軽減するプログラミング演習支援システムが使われている授業である．

2.1 関連研究

プログラミング演習支援システムの学習状況把握における関連研究には，作業進度の提示，作業進度の遅れ検出，一斉指導に対する支援，正誤判定の支援がある．

作業進度の提示[3]は，演習中のクラス全体についての課題に対する作業進度を学生の在席位置に合わせて教員に提示する．その有効性は，提出の作業が遅れている学生を教員が分析して把握できることである．

作業進度の遅れ検出[11]は，プログラムの行数やコンパイル回数を全体の学生と相対的に分析して作業が遅れている学生の情報を教員に提示する．その有効性は，コンパイルの作業が遅れている学生の情報を把握できることである．

一斉指導に対する支援[5]は，過去のコンパイルエラーによって演習中のコンパイル作業履歴を分析して学生のエラーの原因や場所を推定・集計して教員に提示する．その有効性は，共通の問題を抱える学生のエラー内容を教員へ提供して一斉指導を支援できることである．

正誤判定の支援[12]は，あらかじめ用意した解答プログラム例によって学生の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供する．その有効性は，クラス全体のプログラムの正解判定を把握できることである．

これらの関連研究が提案する方法は，ある 1 つの作業に対する指導の支援を目的としており，本研究が提案する機能とは目的が異なる．作業進度の提示は，課題に対する作業進度を提供する点では提案する機能と同様のコンセプトを持つが，提出の作業のみを対象としている点で異なる．

2.2 要求定義

情報システムの開発における要求定義は，利用者がどんな目標を達成したいのか，どんな形で収益をあげていきたいのかという要求を決定することである．

本研究における要求定義は,以下に述べる手順によって，2.1 で述べた関連研究が提案する方法からプログラミング演習支援システムにおける学習状況把握機能に対する要求を決定することである．

(1) 関連研究が提供する学習状況の抽出

学習状況の抽出は，関連研究が教員へ提供する具体的な学習状況の内容について列挙することである．学習状況の具体的な内容は，以下のとおりである．

- クラス全体についての課題の提出における作業進度
- 提出が遅れている学生
- 共通的なエラーでつまずいている学生
- 共通的なエラー内容
- コンパイルが遅れている学生
- 学生のプログラム履歴
- 学生のプログラムに対する正誤

(2) 要求内容の分類整理

要求内容の決定は，抽出された学習状況の内容を分類することである．分類した学習状況の内容は以下のとおりである．

- クラス全体についての課題に対する作業進度
- 共通的なエラー
- 作業進度の遅れ
- 学生のプログラム履歴

以上の手順からプログラミング演習支援システムにおける学習状況把握機能の要求は，クラス全体についての課題に対する作業進度を把握すること，共通的なエラーでつまずいている学生を把握すること，作業進度が遅れている学生を把握すること,学生の作業履歴を把握することである．

2.3 シナリオに基づく設計

情報システムの開発における分析は，システム化したあとで仕事の流れがどのような流れに変化するのかを調査する開発手法を用いて明確にすることである．

本研究で用いる情報システムの開発手法はシナリオに基づく設計[2]である．シナリオに基づく設計は，システムとソフトウェアを分析し設計するときに，それらがどのように人間活動の文脈を変換し，またどのように人間活動の文脈によって制約されているのかを検討するための手段である．その直接的なアプローチの 1 つは，典型的で重要な人間活動を明確に想像して記述することである．そのような記述は，多くの場合シナリオと呼ばれ，システムの使用状況に関する推論を支援する．シナリオは，つぎの 3 つの特徴要素をもっており，それらは状況設定についての言及または仮定である．

- 少なくとも 1 人のアクタ：システムの利用者を表す
- 少なくとも 1 つのアクタの目的：利用者のめざすことがらを表す
- プロットの系列：利用者が目的を達成するために行うタスクや，利用者に対して起きること，システムとの関連におけるシーンを表す

シナリオの記述は，タスクがどのように実行されるか，またそれらのタスクを支援する機能をシステムが提供する方

法について，それぞれのタスクの概略を描く．このシナリオは，システムに必要な機能を見出し，それらを実現したシステムの有効性を評価するのに用いられる．

2.4 学習状況把握のシナリオ

シナリオの作成は，学習状況把握機能に対する利用者の情報要求を具体化するため，プログラミング演習の実際の授業を観察して，学習状況把握機能の利用シナリオを記述することである．

プログラミング演習支援システムにおける学習状況把握機能の利用におけるアクタは，1 人の教員と複数の学生である．要求分析においては典型例を想定して例外を扱わないことになっているが，プログラミング演習支援システムでは理解していない学生や意欲のない学生などの状態の異なる学生がいるために，複数の学生がアクタとなる．つぎに，教員の主目的は，学生を指導することであり，学生の主目的は，課題を解答することである．これらの目的は抽象的であるため，主目的を達成するための副目的をあげて目的を階層的に展開すると，図 1 の目的とその間の関連が得られる．これらのアクタと関連付けられた目的からプロットの系列を具体化すると，学習状況把握機能の利用シナリオは，図 2 になる．

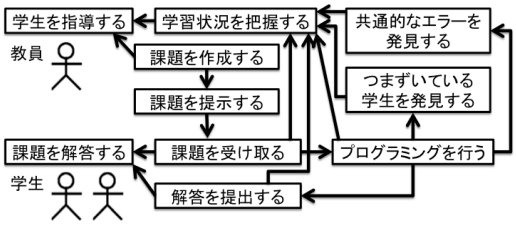

図 1 アクタの目的の関連

教員は課題を作成して学生へ提示し，学生は提示された課題を受け取る．学生は受け取った課題に対してプログラムの編集，一時保存，コンパイル，実行といったプログラミングを行う．教員はシステムが学習状況を分析して提示した情報を確認する．教員は，クラス全体の作業進度や共通的なエラーを確認して，クラス全体に対する指導を行う．また，つまずいている学生の作業履歴を確認して，個別の指導を行う．学生は実行時に正解判定を確認して解答を提出し，教員は提出を確認する．

図 2 学習状況把握機能の利用シナリオ

学習状況把握機能の利用シナリオから学習状況把握機能に対する利用者の情報要求は，クラス全体についての課題に対する解答開始から提出までの作業進度，共通的なエラー内容と人数, クラス全体と比較した作業進度の遅れ, 個々の学生の作業履歴である．

3. 学習状況把握機能の設計

情報システムの開発における設計は，最終的なシステムがどのような形で要件を実現していくのかを機能として決定することである．

3.1 機能定義

情報システムの開発における機能定義は，提案機能における入力と出力，およびアルゴリズムを決定することである．

本研究における機能定義は，2 章で述べた学習状況把握機能に対する利用者の情報要求から決定することである．学習状況把握機能の定義は，学習状況提示機能，コンパイルエラー分類機能，作業進度の遅れ検出機能の 3 つのサブ機能である．

3.2 学習状況提示機能

(1) 定義

学習状況提示機能の定義は，クラス全体についての課題に対する作業進度と学生の作業履歴を提供することである．

(2) 入力データ

学習状況提示機能における入力は，クラス全体についての課題に対する開始時刻，コンパイル時刻，実行時刻，提出時刻の作業時刻集合と課題に対する個々の学生の正解判定，および学籍番号，課題番号である．

(3) 出力データ

学習状況提示機能における出力は，解答開始，コンパイル，実行，正解判定，提出を行なった作業人数集合，および課題に対する解答開始，コンパイル，実行，正解判定，提出の作業時刻集合とコンパイル，実行，正解判定，提出における作業内容集合である．

(4) アルゴリズム

学習状況提示機能のアルゴリズムは，下記のとおりである．

1. 作業時刻集合，正解判定，学籍番号，課題番号を入力する．
2. 作業時刻ごとに学生の人数を集計する．
3. 正解判定が正解である学生の人数を集計する．
4. 学籍番号と課題番号から時系列に解答開始時刻, コンパイル，実行，正解判定，提出における作業時刻とプログラムを抽出する．
5. 作業人数集合, および作業時刻集合と作業内容集合を対で出力する．

3.3 コンパイルエラー分類機能

(1) 定義

コンパイルエラー分類機能の定義は，コンパイルのエラーメッセージと教員が作成した解答プログラム例を用いてコンパイルエラーの傾向を分類することである．

(2) 入力データ

コンパイルエラー分類機能の入力は，学生が課題に対し

て作成したプログラムと学生がコンパイルしたときにエラーの際に出力されるエラーメッセージ，および教員が課題に対して作成した解答プログラム例である．

(3) 出力データ

コンパイルエラー分類機能の出力は，エラーメッセージの先頭のエラー行に対応する教員の実行結果の対応行番号と先頭のエラー行におけるエラー種類である．

(4) アルゴリズム

コンパイルエラー分類機能のアルゴリズムは，下記のとおりである．対応行番号の抽出には，Unix の diff[1]を用いる．

1. プログラムとエラーメッセージ，および解答プログラム例を入力する．
2. エラーメッセージから，先頭のエラー行番号とエラー種類を抽出する．
3. プログラムの行を解答プログラム例の行に，文字の違いが最も小さくなるように対応付け，エラー行に対応する解答プログラム例の対応行番号を抽出する．
4. 対応行番号が存在しなければ，対応行番号を不明とする．
5. 対応行番号とエラー種類の対を出力する．

3.4 作業進度の遅れ検出機能

(1) 定義

作業進度の遅れ検出機能の定義は，外れ値検定[13]を用いて課題に対するコンパイルや実行などの作業進度が遅れている学生を発見することである．

(2) 入力データ

作業進度の遅れ検出機能の入力は，作業進度の遅れ検出機能を実行したときの現在時刻と課題についてクラス全体の学生の開始時刻，一時保存時刻，コンパイル時刻，実行時刻，提出時刻の作業時刻集合，および外れ値と判定する基準の有意水準である．

(3) 出力データ

作業進度の遅れ検出機能の出力は，学籍番号と氏名である．

(4) アルゴリズム

作業進度の遅れ検出機能のアルゴリズムは，下記のとおりである．外れ値検定には，スミルノフ・グラブス検定[8]を用いる．

1. 現在時刻と作業時刻集合および，有意水準を入力する．
2. 作業ごとの作業時刻集合に現在時刻を追加して外れ値検定を行う．
   - 2-1. 外れ基準値を算出する．
   - 2-2. 各作業時刻について検定統計量を求める．
   - 2-3. 検定統計量の最大値を抽出する．
   - 2-4. 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．
   - 2-5. 外れ値が存在しなければ，このアルゴリズムを終了する．
   - 2-6. 現在時刻が外れ値ならば，3 へ進む．
   - 2-7. 作業時刻集合から，外れ値の作業時刻を除いて 2-2 へ戻る．
3. 作業時刻集合が提出の作業ならば，作業時刻が存在しない学生を選択し，そうでなければ，作業時刻集合の学生を選択する．
4. 選択された学生の学籍番号と氏名を出力する．

3.5 設計評価

情報システムの開発における設計評価は，提案機能が入力に対して出力が設計どおりに作成されていることを確認するため，機能の有用性について検証することである．

本研究における設計評価は，学習状況把握機能の 3 つのサブ機能の有用性について，学習状況提示機能を教員のインタビューの回答から，コンパイルエラー分類機能をコンパイルエラー分類の精度から，作業進度の遅れ検出機能を有意水準の変化による外れ値検出回数から，それぞれ検証することである．

(1) 学習状況提示機能の評価

学習状況提示機能の有用性は，後述する 4 章の評価実験において，学習状況提示機能に対する 2 人の教員のインタビューの回答から評価する．インタビューの質問と回答は，つぎのとおりである．

- 質問：学習状況提示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．
- 教員 A の回答：

  良かった点
  - 作業進度が課題ごとに把握できる．
  - クラス全体の作業進度を把握しながら課題ごとへのヒントを出せる．
  - エラー状況や作業履歴が把握できるので，学生がどのような状況で課題につまずいているかを把握できる．

  悪かった点
  - 特別にない．
- 教員 B の回答：

  良かった点
  - 課題の未着手者，未完成者を容易に調べることができる．
  - 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確認できるので，内容をもとに TA に指導の指示が出せる．

  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表示されると把握が簡単になる．

インタビューの回答から，学習状況提示機能は，クラス

全体についての課題に対する作業進度や学生の作業履歴を提供しており，有用である．

(2) コンパイルエラー分類機能の評価

コンパイルエラー分類機能の有用性は，実際の授業で学生が作成したプログラムに対するコンパイルエラー分類の精度によって評価する．評価に用いた学生のプログラムは，4 章の評価実験において収集したコンパイルエラーのある 5708 個のプログラムである．コンパイルエラー分類の精度は，対応行番号が示した解答プログラム例の行と，エラー行番号が示す学生のプログラムの行の差が最も小さいことを正解として評価した結果，98.39%である（表 1）．また，表 1 における不正解は，対応行番号が決定できる場合についての誤りである（表 2）．誤りの原因は，エラー行を指定している場所が，解答プログラム例と差異のない場所であり，実際にエラーを起こしている場所と異なるためである．誤りにおける対策は，行だけではなく全体を通して文字の違いを対応付けることである．

表 1 コンパイルエラー分類の精度

プログラム数	正解数	不正解	精度
5708	5616	92	98.39%

表 2 コンパイルエラー分類の不正解

エラー種類	件数	誤り率	構成比率
class、interface、または enum がありません。	37	0.65%	40.21%
<identifier> がありません。	11	0.19%	11.96%
式の開始が不正です。	11	0.19%	11.96%
'else' への 'if'がありません。	10	0.18%	10.87%
不正な文字です。	7	0.12%	7.61%
すでに定義されています。	5	0.09%	5.43%
その他	11	0.19%	11.96%

(3) 作業進度の遅れ検出機能の評価

作業進度の遅れ検出機能の有用性は，有意水準の変化による外れ値検出回数によって評価する．外れ値検出回数は，4 章の評価実験において収集した学生全員が作業を行った時点においての作業データに対して，外れ値として検出された学生の人数である．また，有意水準を 1%～100%の範囲で変化させて算出する（図 3）．作業進度の遅れ検出機能は，外れ値検出回数が，有意水準に比例して増加することから，教員が必要に応じて外れ値検出の量を制御できるため有用である．

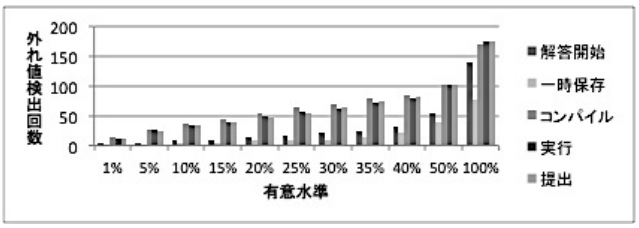

図 3 有意水準の変化による外れ値検出回数

4. 学習状況把握機能の評価

情報システムの開発における評価は，導入されるシステムとして品質が十分であるのか，利用者の視点でシステムとしての要件が満たされているかどうかを実運用と同じ形式で試験することである．

4.1 評価の方法

本研究における評価は，学習状況把握機能の効果を実際の授業においてプロトタイプシステムを用いて評価実験を行い，その有効性を考察することである．効果に対する仮説は，提案機能によって教員の指導回数が増えること，および指導内容が変わることである．評価実験の対象のプログラミング演習は，大学 1 年の「プログラミング基礎・演習」である．この授業科目は，手続き型プログラムの作成を目的として，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．

4.2 プロトタイプシステム

情報システムの開発におけるプロトタイプシステムは，利用者に提案機能の操作感を実際に確かめてもらう手段を可能にする仕組みである．

本研究におけるプロトタイプシステムは，プログラミング演習の授業において学習状況把握機能を教員に提供することである．プロトタイプシステムの要件は，教室にネットワーク環境があること，および教員と学生の PC に Web ブラウザが必要なことである．

プロトタイプシステムの構成は，プログラミング支援機能と 3 章で述べた学習状況把握機能である．プログラミング支援機能は，2 章で述べた関連研究の方法を一般化して定義し，教員に対する機能と学生に対する機能に分類される（図 4）．

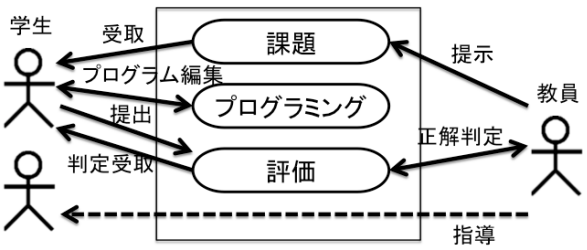

図 4 プログラミング支援機能の概要

プロトタイプシステムの構造は，クライアントサーバ方式を用いて，プログラミング支援機能と学習状況把握機能，およびデータベースである（図 5）．プログラミング支援機能は，クライアントからの要求をサーバで処理してデータベースに登録し，クライアントへ結果を送信する．学習状況把握機能は，クライアントからの要求によりデータベースから取得し，分析してクライアントへ結果を送信する．

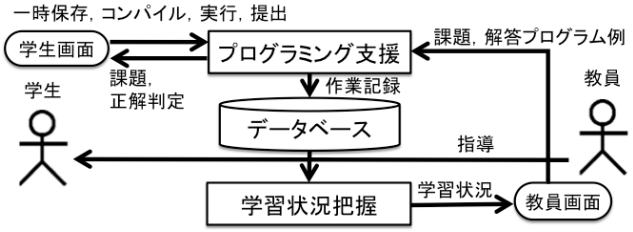

図 5 プロトタイプシステムの構造

プロトタイプシステムの実装は，PC のプラットフォームに依存しないように，ユーザが Web ブラウザを使って操作する Web アプリケーションとして実現する．プロトタイプシステムのプログラムは，サーバサイドアプリケーションとして，PHP, HTML, Ajax, JavaScript 言語を用いる．また，データベースシステムは，MySQL を用いる．データベースは，教員，学生，課題，解答プログラム，一時保存，コンパイルエラー，エラー種類の 7 つの表から構成される．データベースの各表は以下の情報を記録する．

- 教員：教員の ID とパスワード
- 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実行時刻
- 一時保存：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解答プログラム例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

プロトタイプシステムのユーザインタフェースは，教員用と学生用から構成される．各画面とその操作は，以下のとおりである．

(1) 教員の操作

- ログイン画面：教員の ID とパスワードを入力すると，教員のトップページである演習課題画面へ移行する．
- 演習課題画面：課題の出題状況を表示し，課題作成画面，課題作成画面，出席者一覧画面，学習状況画面へ移行する．
- 課題作成画面：課題名，課題内容，解答プログラム例を入力し，コンパイルと実行を行って，実行結果を確認してから，課題の実施日，開始時刻，終了時刻を入力して登録する．
- 出席者一覧画面：ログインしている学生の氏名，座席番号，解答中の課題名を表示する．学生の氏名をクリックすると作業履歴画面へ移行する．
- 学習状況画面：課題ごとの課題名，課題内容，解答人数，出題時刻，終了時刻，全体の作業進度として解答開始人数，コンパイル人数，実行人数，不正解人数，提出人数，またエラー状況としてコンパイルエラー分類の結果を人数順に表示する（図 6）．表示された項目のそれぞれ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：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，行動時刻を表示する．学籍番号をクリックすると，作業履歴画面へ移行する．
- 作業履歴画面：学生の課題に対する解答開始，一時保存，コンパイル，実行，提出の時刻，およびプログラムと作業に対する作業内容を表示する（図 7）．

図 6 学習状況画面

図 7 作業履歴画面

(2) 学生の操作

- ログイン画面：学籍番号，氏名，座席番号を入力し，課題選択画面へ移行する．

- 課題選択画面：課題の一覧を表示し，選択する課題名をクリックして，演習画面へ移行する．
- 演習画面：課題に対するプログラムを入力し，コンパイル，実行して結果を表示する（図 8）．実行結果が正解の場合は，解答を提出する．提出すると課題選択画面へ移行する．

図 8 演習画面

4.3 評価実験

実験の目的は，実際の授業において学習状況把握機能の効果を検証することである．

検証の方法は，2 人の教員が同じ学習内容について学習状況把握機能の有無による演習を交互に 2 回ずつ実験を行い，指導回数と指導内容を測定して比較することである．指導回数の測定は，教員の音声録音と室内をビデオ録画し，指導内容については，教員の音声録音と教員画面をビデオ録画することである（図 9）．

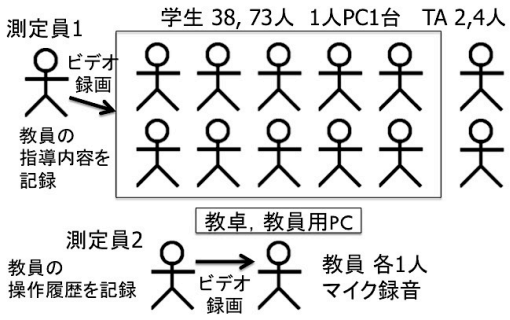

図 9 効果の測定環境

指導回数の測定結果は，学習状況把握機能の有無による総指導回数および教員別と学習内容別の指導回数に統計的に有意な差がない（表 3）．

指導内容の測定結果は，学習状況把握機能によってしか把握できない学習状況に対する指導があり，指導内容が変化する（表 4）．

表 3 学習状況把握機能の有無による指導回数

学習状況把握機能		あり	なし
総指導回数		65	64
教員	A	38	37
	B	27	27
学習内容	算術演算子と式	16 (A)	9 (B)
	条件分岐	15 (B)	23 (A)
	論理演算子	22 (A)	18 (B)
	繰り返し for	12 (B)	14 (A)

表 4 学習状況把握機能の有無による指導内容

機能	原因	結果		件数	合計
		指導分類	指導内容		
あり	エラー状況	一斉	文字列の連結方法を確認すること	1	1
	分類別学習状況	一斉	・全体の作業進度から遅れている学生へサポートの要求を促す※ (5) ・論理演算子を用いること	9	29
		個別	・コンパイルの時間間隔が長い学生を促す※ (2) ・不等号の使い方を確認すること (4)	20	
	巡回	一斉	switch 文について，色々なパターンを試すこと	1	27
		個別	プログラムの記述を確認すること (6)	26	
	挙手	一斉	課題の内容をよく読むこと	2	8
		個別	for 文の中の変数の宣言方法を確認すること	6	
なし	巡回	一斉	if 文について，偶数判定の方法を確認すること	7	42
		個別	・偶数の判定方法を確認すること (8) ・文字列の連結方法を確認すること (5)	35	
	挙手	一斉	if, else 文を使うこと	4	22
		個別	・偶数の判定方法を確認すること (3) ・文字列の連結方法を確認すること (3)	18	

4.4 考察

学習状況把握機能の考察は，学習状況把握機能が提供する学習状況の情報によって教員の指導に対する効果を明らかにすることである．

(1) 指導回数に対する効果

学習状況把握機能は，学習状況把握機能によって指導にあたった割合が，65 回中 30 回の 46%あるところから，指導に有益な学習状況の情報をリアルタイムに提供している．

ⓒ2013 Information Processing Society of Japan

(2) 指導内容の変化に対する効果

学習状況把握機能は，作業進度が遅れている学生を個別指導しているところから，学習状況把握機能によってしか把握できない指導を行なっているため，指導に有益な学習状況の情報をリアルタイムに提供している．

(3) 効果に影響する要因

指導回数に差がないことの要因は，教員が学習状況把握機能を使うためには机間巡回の間に教卓へ戻らなければならないことである．

学習状況把握機能による学習状況把握は，教員が学習状況画面を手動で更新して確認しているため，作業進度が遅れている学生の情報を見落とす可能性がある．

5. プログラミング演習支援システムの導入

情報システムの開発における導入は，利用者がシステムを利用できることを確認し，問題があれば解消していくことである．

本研究における導入は，4 章で述べた評価実験で明らかになった問題を解消して実際の授業で使われていることの有用性を評価することである．学習状況把握機能の効果に影響する要因の机間巡回の間に学習状況把握機能が使えない問題は，タブレット端末を利用することで解消される．図 10 は，スクリーンに課題の内容や要点を提示し，タブレット端末を利用して学生の学習状況を把握しながら個別指導を行なっている様子である．

学習状況把握機能は，本論文の投稿段階で運用途中でありプログラミング演習支援システムにおける学習状況把握機能としての利用を開始しているため，教員の学習状況把握に役立っている．

図 10 タブレット端末の利用における授業風景

6. おわりに

本研究は，プログラミング演習支援システムにおいて，演習の課題に対する解答開始から提出までの作業ごとのクラス全体の学習状況と個別学生の学習状況を教員がリアルタイムに把握する学習状況把握機能を実現した．本研究で提案した学習状況把握機能は，クラス全体の学習状況を把握するという課題を，作業開始時刻を集計する学習状況提示機能と，コンパイルエラーメッセージを分類集計するコンパイルエラー分類機能により解決した．また，個別学生の学習状況を把握するという課題を，作業開始時刻の外れ値を分析する作業進度の遅れ検出機能により解決した．これらの提案機能は，実際の授業における評価実験の結果，教員の指導内容の約半数が学習状況把握機能の利用に起因しており，作業進度に合わせた指導を行なっていることから，リアルタイム学習状況把握に有効である．

今後の課題は，作業進度が遅れている学生の情報を教員がリアルタイムに把握するため，作業進度の遅れ検出機能を定期的に実行して教員へ通知することである．

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る．

参考文献

1) Diffutls, http://www.gnu.org/software/diffutils/diffutils.html
2) John M.Carroll: シナリオに基づく設計, 共立出版 (2003).
3) 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).
4) 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, No.6, pp.1-8 (2011).
5) 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, vol.104, no.703, pp.19-24 (2005).
6) 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).
7) 佐川 博樹: よくわかる最新システム開発者のための要求定義の基本と仕組み, 秀和システム (2010).
8) 東京大学教養学部統計学教室: 統計学入門 (基礎統計学), 東京大学出版会 (1991).
9) 西 輝之, 劉 渤江, 横田 一正: デバッガとの連携による C 言語学習支援システムの提案, 電子情報通信学会技術研究報告 vol.106, no.583, pp.173-178 (2007).
10) 長谷川 伸, 松田 承一, 高野 辰之, 宮川 治: プログラミング入門教育を対象としたリアルタイム授業支援システム, 情報処理学会論文誌, vol.52, no,12, pp.3135-3149 (2011).
11) 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).
12) 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).
13) 山西 健司: データマイニングによる異常検知, 共立出版 (2009).

ⓒ2013 Information Processing Society of Japan