

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

安田 光¹ 井上 亮文¹ 市村 哲^{1,a)}

受付日 2011年4月15日, 採録日 2011年10月3日

概要: 現在のプログラミング講義・演習科目では, 完成したソースコードを示して解説する手法が主流であるが, この手法では, 学生がエラー解決方法を習得することが難しいという問題がある. そこで本研究では, ティーチングアシスタント (TA) が学生にエラーの解決方法を教えている状況を動画記録し, 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. 本システムは, 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. 評価実験の結果, トラブルを解決している過程を記録して資料化することに一定の有用性があることが示唆された.

キーワード: Web, プログラミング学習, 動画資料

Programming Education System that can Share Problem-solving Processes Between Students and Teaching Assistants

KO YASUDA¹ AKIFUMI INOUE¹ SATOSHI ICHIMURA^{1,a)}

Received: April 15, 2011, Accepted: October 3, 2011

Abstract: Recently, programming class is mostly done using electronic slide-show that shows completed source codes. It is difficult, however, for students to learn how to resolve errors. In this paper, we propose a process-recording-based programming education system composed of the process-recording tool that produces a video file and the process-recording material viewing sites where the process-recording material shows how to solve problems in programming. As a result of the experiment, recording the process of debugging and sharing it were proved somewhat effective to teach how to resolve errors.

Keywords: web, programming education, movie material

1. はじめに

情報系大学や情報系学部では, プログラミングを学ぶ講義および演習が必須科目となっていることが多い. 現状, 完成したプログラムのソースコードを表示して解説を行う方法が主流となっているが, この手法では, プログラミングの際に発生するエラーの解決方法を学ぶことは難しいという指摘がある [8], [11]. 通常, 学生は自分の限られた知識の範囲内でエラー解決を行う必要がある.

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 1 人 1 人に対応することが難しいという問題がある [12]. 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. しかしながら, それでも学生数に対して TA の数は少なく, TA が到着するまで学生はしばしば長時間待たされる場合がある. また, TA が保有しているエラー対処方法やノウハウは, 質問した一部の学生にしか伝わらず保存もされないという問題が生じている.

¹ 東京工科大学
Tokyo University of Technology, Hachioji, Tokyo 192-0982,
Japan

^{a)} ichimura@cs.teu.ac.jp

これらの問題に対し著者らは、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た。そしてこの考えに基づき、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。本システムは、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。

学生がエラー解決方法を学習しやすく、また、学生が自らエラー解決が行えると実感できる環境を提供することで、プログラミング授業における学習効果を向上させることが本研究の目的である。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、提案システム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。これにより本研究の目的が達成されたと判断した。

2. プログラミング教育支援に関する従来研究

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、本論文の提案とは方向性が異なっている。また、学生にプログラムの作成過程を説明する機能は備えていなかった。

そこで、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], [12] を開発した。講義前に講師がソースコードを作成する過程を動画で収録しておき、講義中に学生に見せることができるようになっていた。プログラム作成過程を動画収録するツールとしては、他にも文献 [8] および [11] に記載されているツールが存在する。しかしながらこれらのシステムは、講義中に見せる動画資料を講師が講義前に作成するためのものであり、本論文で提案するような TA が学生のトラブルを解決している状況を収集・蓄積・共有する機能は備えていなかった。また、著者らが調査した限りにおいては、講師や TA 等の教示者が学生等の学習者にエラー解決方法を示している過程を動画記録して資料として共有するシステムは

他に見当たらない。

3. 本研究の目的と位置づけ

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PPT(パワーポイント)スライド形式の講義資料を配布し、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となっている。

現在、プログラミング講義で使われる講義資料の例を図1に示す。講義資料にはサンプルプログラムのソースコードが載っており、学生はそれを入力し実行することでプログラムを学んでいく。しかし、このような静止画の講義資料では、プログラムが作成された過程を読み解くことが難しく、また、プログラミングを行う際に起こるエラーの解決方法を学ぶことができないという指摘がある [8], [11]。

講師や TA は、通常、学生に比べて高いプログラミング作成スキルを持った熟練者であり、学生らが保有していないプログラム作成ノウハウを持っていることが多い。また、学習者にとって、熟練者のプログラミング技術を見ることは、自身の学習の助けになるといわれている [8]。

以上のことから、明らかとなった本研究の課題を以下に述べる。

- 課題1 エラーへの対処方法が共有または公開されおらず、学生は自分の知識の範囲内でエラー解決を行う必要がある。
- 課題2 TAのエラー対処方法やノウハウが一部の学生にしか伝わらず、また、保存もされない。
- 課題3 エラー解決方法が分からない学生は独力で解決することをあきらめてしまう。

課題1について、現状の講義・演習においては、エラーへの対処方法が共有または公開されていないことが一般的

```

while 文
例4 1からnまでの自然数の和がはじめて
100を超えるときのnの値は？
int i=0,sum=0;
while (sum<=100) {
    i=i+1;
    sum=sum+i;
}
System.out.println(i);
    
```

図1 講義資料の例

Fig. 1 Example of lecture material.

であり、学生は自分のきわめて限られた知識の範囲内でエラー解決を行う必要がある。

課題2について、学生が自力でトラブルを解決することができない場合は、TAを呼びエラーの解決方法を尋ねるが、そのときTAから与えられた有益な情報は当該学生に伝わるだけであり、保存もされないため他の学生に伝わらない。

課題3について、Webに掲載された授業資料や書籍を見てもエラーの解決方法が書かれていることはまれであり、エラーを解決できない学生は、TAの到着まで待つしかない。直面しているエラーの解決方法を自分自身でより粘り強く調べることができるようになることが望ましい。

なお、特に課題1に関し、学生がどのようなエラーを起すかを事前に予測することは困難であり、講師やTAがそのための資料を事前に作成することはきわめて難しいのが現実である。プログラミングを行う際に起こるトラブルとして、コンパイルエラーと実行時エラーがある。学生は、この両方のエラーに対する対処方法を学ぶ必要がある。コンパイルエラーは、入力したソースコードの構文が誤っている場合や必要なライブラリがリンクされていない場合等に発生し、このままではプログラムの実行を行うことができない。コンパイルエラーメッセージには問題のあるソースコードの箇所とそのエラー内容を説明する記述が含まれており、学生は、そのエラーメッセージをもとにエラーを解決していく方法を習得する必要がある。一方、実行時エラーは、コンパイルは成功するが、プログラムを実行する際に強制終了したり、意図しない結果が出力されたりするというものである。変数の値が初期化されず使われている場合や、ループや分岐の条件が適切ではない場合等に発生する。そのような問題の原因を見つけ出すためのエラー解決方法の習得が必要となる。

以上の3つ課題に鑑み、学生がエラー解決方法を学習しやすく、また、学生が自らエラー解決が行えると実感できる環境を提供することで、プログラミング授業における学習効果を向上させることを本研究の目的と定めた。本論文においては「本研究が対象とする学習効果が向上した状態」とは、「学生がエラー解決方法を学習しやすく、また、学生が自らエラー解決が行えると実感できる状態」である。そして、上記3つの課題を解決することが、本研究の目的を達成するための要件である。

なお、提案システムの導入によって、講師やTAが複数の学生に対し同じ説明を繰り返す必要が低減し、指導が効率化されることが期待できるが、本論文では効率化の追求については対象外としている。

4. プログラミング演習支援システム

4.1 概要

本論文で提案するプログラミング演習支援システムは、

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。本システムはプログラミング演習またはプログラミング講義の時間内に利用されることを想定している。

クライアントツールはプログラミング講義を受講する各学生のPC上で動作するソフトウェア（Windowsアプリケーション）である。一方資料閲覧サイトは、クライアントツールから送信された資料を掲載するWebサイト（PHPで開発されたWebアプリケーション）である。同一講義内において、すべてのクライアントツールと資料閲覧サイトは常時ネットワーク接続されており、クライアントツールから送信された資料（動画資料を含む）は即座に資料閲覧サイトにアップロードされる。また、資料閲覧サイトには、当日の課題、当日使用された講義スライド（静止画）、サンプルソースコード、問題解決上のヒント等も掲載される。学生は、資料閲覧サイトを適宜参照しながら、クライアントツール上でプログラムを作成するというようにして学習を行う。ただし、複数の学生が同時に同じ課題に取り組んでいる場合、他の学生のプログラムが修正されている様子が動画で閲覧できてしまうと、他の学生の解答や、TAの解答例を見てしまうことになる。これを防ぐために、通常学生が閲覧できるのは授業開始前にアップロードされた動画資料のみであり、授業開始後にTAによってアップロードされた動画資料の閲覧は制限されている。

4.2 機能とシステム利用の流れ

提案システムが提供する機能とシステム利用の流れについて述べる。まず、本章で説明する提案システムの3機能を以下に列挙する。

- 機能1（動画記録機能） 学生PC上のクライアントツールをTAモードで動作させてTAがソースコードを修正する過程を動画として記録する機能
- 機能2（資料収集機能） TAが作成した動画資料に各種メタ情報を付けて資料閲覧サイトに収集する機能
- 機能3（資料検索機能） 資料閲覧サイトにおいて各種メタ情報を指定してエラー解決動画を検索する機能

4.2.1 機能1（動画記録機能）

図2にシステムを利用する流れを示す。学生は講義中に出される課題をクライアントツール上でプログラミングする。本クライアントツールは、学生モードとTAモードの2つのモードを有している。学生モードでは、プログラムのソースコードの読み込みや新規作成、編集、コンパイル、実行、コンパイルエラー表示等の機能が利用でき、プログラミングを行う際に必要な作業がひととおり行えるようになっている。学生が自力で問題を解決することができない場合は、TAを呼び、助言を受ける。

TAは学生から質問を受けた箇所について助言を行う。

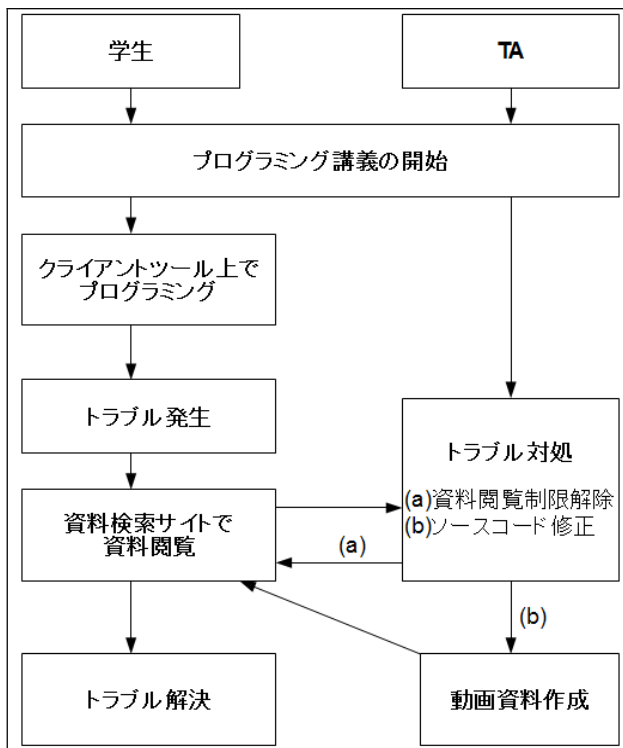


図 2 システムを利用する流れ
Fig. 2 Flow-chart.

このとき、TA は必要に応じて TA 自身のログイン名とパスワードを入力して学生 PC 上のクライアントツールを TA モードで動作させる。TA モードでは資料閲覧サイトの閲覧制限が解除となり、すべての資料（授業開始後にアップロードされた動画資料を含む）の閲覧が可能になる（図 2(a)）。トラブルに対し適切な解決方法が資料閲覧サイトにあれば学生に閲覧させ解決を促す。もし適切な資料がなく状況が改善しなかった場合に、TA はプログラムの修正を実演してみせる。このとき TA モードでは、ソースコードを修正する過程を動画として記録することが可能である。ふだんどおりプログラミングを行うだけで動画資料を作成できる。TA はプログラムの修正を実演してみせ、動画資料を作成する（図 2(b)）。

動画作成にかかる時間は、200 文字程度の修正であれば「動画作成」ボタンを押してから約 2 秒である。なお、現状のクライアントツールの使用方法については講師と TA の区別はなく、講師は講師自身のログイン名とパスワードを入力して学生 PC 上のクライアントツールを TA モードで動作させることが可能である。

4.2.2 機能 2（資料収集機能）

TA は作成された動画資料をクライアントツールから資料閲覧サイトに送信できる。このとき、作成された動画資料と一緒に、コンパイルエラーメッセージまたは実行エラーログ、修正前後のソースコード、課題番号、投稿日時、フリーワード等の各種メタ情報が資料閲覧サイトに送信される。

4.2.3 機能 3（資料検索機能）

資料閲覧サイトでは、TA が入力した課題番号やフリーワードをもとに資料検索を行うことができる。資料閲覧サイトには動画プレイヤーのほかにこれらの各種メタ情報が表示されている。

4.2.4 各機能と課題の対応

本章で述べた各機能と 3 章で述べた本研究の課題の対応について述べる。学生 PC 上のクライアントツールを TA モードで動作させて TA がソースコードを修正する過程を動画として記録する機能 1（動画記録機能）は、課題 1 に対応している。また、TA が作成した動画資料に各種メタ情報を付けて資料閲覧サイトに送信する機能 2（資料収集機能）は、課題 2 に対応している。さらに、資料閲覧サイトにおいて各種メタ情報を指定してエラー解決動画を検索する機能 3（資料検索機能）は、課題 3 に対応している。

5. 実装

クライアントツールは TA が学生にエラーの解決方法を教えている状況を動画等で記録するツールである。また資料閲覧サイトは、クライアントツールが作成した資料を学生と TA 間で共有するための Web アプリケーションである。

それぞれの実装について説明する。

5.1 クライアントツール

クライアントツールはプログラミング講義を受講する各学生の PC 上で動作し、学生モードと TA モードの 2 つのモードを有している。表 1 に各モードの機能を示す。

図 3 に学生モード時のクライアントツールを、図 4 に TA モード時のクライアントツールを示す。

学生モードでは、プログラムのソースコードの編集、コンパイル、実行等のプログラミングを行う際に必要な作業がひととおり行えるほか、コンパイルエラーメッセージを検索キーワードとして、Web または本システムに登録されているヘルプページ内を即座に検索できるようになっている。

TA モードでは、学生モードと同様にプログラムのソースコードの読み込みや新規作成、コンパイル、実行、コンパイルエラーの閲覧ができるほか、TA がソースコードを修正する過程を自動的に動画記録することができる。ふだんどおりプログラミングを行うだけで容易に動画資料を作成できる。ソースコードを修正する過程は、TA が文字を入力するたびにエディタ部分をキャプチャ画像として生成・保存し、生成されたキャプチャ画像を連結することで動画を作成している [9]。

また TA モードにおいて、動画中に解説を加えたい場合には、テキストを入力し「コメント」ボタンを押すことにより、図 5 に示すように動画中に吹き出しコメントとして

表 1 各モードの機能

Table 1 Functions for each mode.

機能	学生モード	TA モード
ソースコードの読み込み, 新規作成, 編集, コンパイル, 実行, エラー閲覧	○	○
修正過程の動画記録, 動画へのコメント挿入, 資料へのタグ付け, 資料のアップロード	—	○

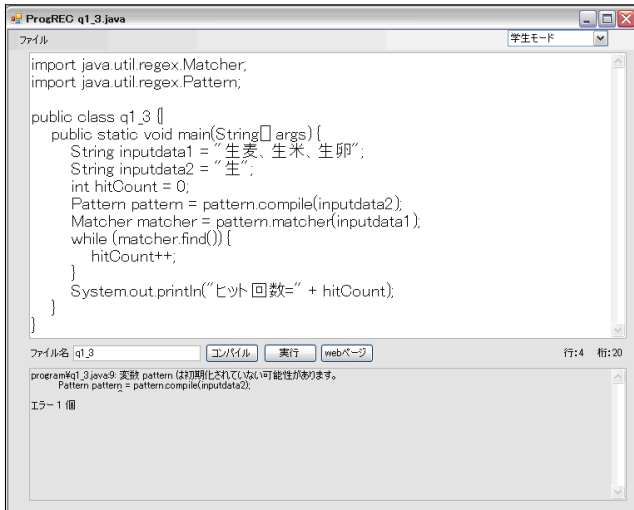


図 3 学生モード

Fig. 3 Mode for student.

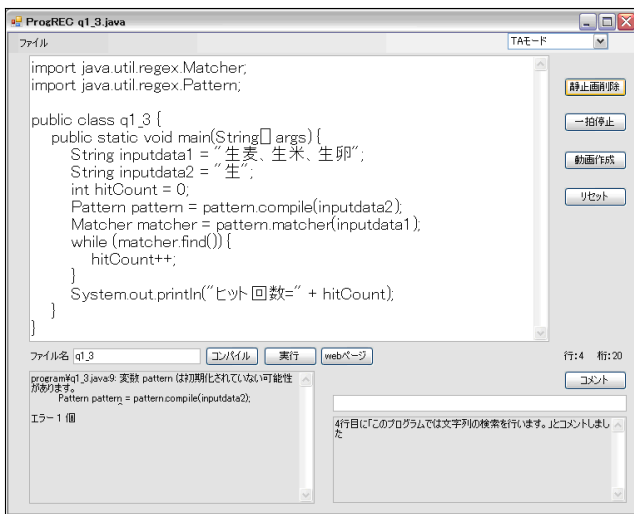


図 4 TA モード

Fig. 4 Mode for TA.

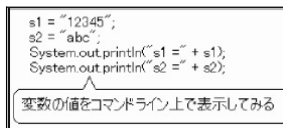


図 5 吹き出しコメント

Fig. 5 Balloon comments.

説明を加えることができる。さらに、動画資料の再生速度を変更すること等が可能である [12]。

TA がソースコードの修正を失敗した場合には、修正し

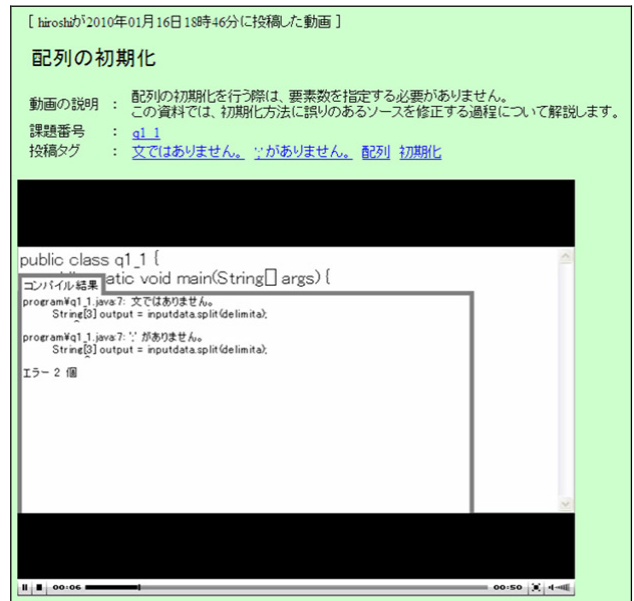


図 6 動画資料再生 Web ページ

Fig. 6 Web Page showing lecture-video.

たソースコードをリセットする機能を利用して最初の状態へと戻すことができる。なお、クライアントツールで作成される動画資料については、Web ブラウザ上で再生が可能な FLV (Flashvideo) 形式 [10] で出力できるようになっている。ソースコードの修正を終えたあと、TA は動画資料のタイトルや説明、および、該当する課題番号を入力する。そして、これらの情報をクライアントツールからサーバへとアップロードするが、作成された動画資料、コンパイルエラーメッセージ、修正前後のソースコードと一緒に送信される。従来の指導方法と比較しても、さほど大きな手間や負担を強いることなく、エラー解決過程の収集および共有を行うことが可能である。

5.2 資料閲覧サイト

資料閲覧サイトでは、TA がクライアントツール上で、資料をデータベースへと送信する際に入力した課題番号やキーワードをもとに資料検索を行うことができる。

資料閲覧サイトには Flash 動画プレイヤーが埋め込まれており、FLV 形式の動画資料を再生できる。また、資料のタイトルや説明文、課題番号、投稿日時等の各種メタ情報を確認できる。さらに、課題番号とコンパイルエラーメッセージ (投稿タグ) はそれぞれリンクボタンになっており、クリックすることで、その文字列を用いた Web 検索、データベース検索が素早く行えるようになっている。図 6 に動画資料再生ページを示す。

6. 評価実験

学生がエラー解決方法を学習しやすく、また、学生が自らエラー解決が行えると実感できる環境を提供することで、プログラミング授業における学習効果を向上させるこ

とが本研究の目的である。この目的に基づき、本論文が対象としている学習効果が向上したかどうか（学生がエラー解決方法を学習しやすく、また、学生が自らエラー解決が行えると実感できる状態に近づいたかどうか）を評価する実験を行った。

6.1節では、エラーの解決方法を学ぶために動画を用いることが適しているかどうかについて検証し、6.2節では提案システムによって本研究が対象とする学習効果が向上するかどうかを検証する。

6.1 動画資料の有効性検証

エラーの解決方法を学ぶための資料として、動画を用いることが適しているかどうかを調べるためにシステム実装前に予備実験を行った。講師やTAは学生らが保有していないコーディングノウハウ（print文による変数表示や実行箇所トレースのテクニック、プログラム入力作業やデバッグ作業の省力化テクニック、問題解決順序の判断方法等）を持っており、これらのノウハウを学生に見せることで学生のプログラミングスキルが向上するといわれている[8]。このことを著者らが想定する環境において実際に検証することが本実験の目的である。

ソースコードを修正入力している過程を表示する動画資料Aと、完成したソースコード全体を表示した静止画資料Bを作成し、それぞれを情報系学部の1年生から4年生までの大学生16名に視聴させ、アンケート調査を行った。アンケート項目は6項目で、1（あてはまらない）～5（あてはまる）の5段階評価によって回答させた。同時に被験者へインタビューを行い、感想を聞いた。

アンケート調査後、Wilcoxon符号付き順位和検定を行った結果、表2のように「理解しやすい」「見やすい」「飽きにくい」「魅力がある」「自然である」「また視聴したい」の6つの項目について、有意水準1%で有意差が認められた。インタビューでは、「プログラムのソースコードを入力する過程があると分かりやすく、動きがあるので見ていて飽きにくい」という意見を得ることができた。これらのことから、プログラミング講義の資料として利用する際は、静止画による解説よりも、動画による解説の方がより有効であると分かった。そこで、本研究で提案するシステムでは、動画資料Aの形式を学生らに提供することとした。

表2 予備実験アンケート調査でのWilcoxon符号付き順位和検定
Table 2 Wilcoxon signed-rank test of pre-experiment.

アンケート項目	動画資料Aの平均	静止画資料Bの平均	Wilcoxon符号付き順位和検定p値
Q1 理解しやすい	4.3	2.4	0.00003 **
Q2 見やすい	4.0	2.6	0.00468 **
Q3 飽きにくい	3.5	1.7	0.00002 **
Q4 魅力がある	3.6	1.7	0.00004 **
Q5 自然である	3.9	2.8	0.00623 **
Q6 また視聴したい	3.1	1.8	0.00294 **

n = 16, ** ... P < 0.01, * ... P < 0.05

6.2 提案システムの有効性評価

プログラミング初学者にとって、提案システムを導入した演習環境が、従来のプログラミング演習環境と比較して、よりエラーの原因を理解しやすい環境かどうかを検証するために実験を行った。

従来どおり書籍とWebを用いてトラブルに対処する方法と、書籍とWebに加え本研究で作成した資料閲覧サイトを利用してトラブルに対処する方法の2つの方法を被験者に実施してもらい、アンケート調査を行った。被験者には、プログラミング学習を始めて1～2年目の大学生11名を選んだ（情報系学部のプログラミング基礎演習クラスから学生を召集し本実験の被験者とした）。

6.2.1 実験方法

実験では、コンパイルエラーを起こすソースコードと実行エラーを起こすソースコードを修正するタスクを課し、書籍とWebを利用した修正と、書籍とWebに加え資料閲覧サイトを利用した修正を行ってもらった。

具体的には、コンパイルエラーを起こすソースコード2本と実行エラーを起こすソースコード2本を用意した。被験者は6名と5名の2グループ（XおよびY）に分けられており、Xはコンパイルエラーを起こすソースコードの1本目と実行エラーを起こすソースコードの1本目を書籍・Webのみを利用して解き、コンパイルエラーを起こすソースコードの2本目と実行エラーを起こすソースコードの2本目を書籍・Web・資料閲覧サイトを利用した方法で解いた。一方Yは、コンパイルエラーを起こすソースコードの2本目と実行エラーを起こすソースコードの2本目を書籍・Webのみを利用して解き、コンパイルエラーを起こすソースコードの1本目と実行エラーを起こすソースコードの1本目を書籍・Web・資料閲覧サイトを利用して解いた。

なお、資料閲覧サイトの動画資料はTAによって作成されたものであり、print文を段階的に挿入することにより変数の表示を適宜行うテクニックが含まれていた。

実験中、1つのソースコードの修正が終わるたびに、各アンケート項目に対し、1（あてはまらない）～5（あてはまる）の5段階で回答させた。

次に、「書籍・Web」と「書籍・Web・資料閲覧サイト」のアンケート結果に差があるかどうか、コンパイルエラーと実行時エラーのソースコードそれぞれの場合について、検定を行った。

以下に実験条件の一覧を示す。

- 1A：書籍・Web（コンパイルエラー）
- 1B：書籍・Web・資料閲覧サイト（コンパイルエラー）
- 2A：書籍・Web（実行時エラー）
- 2B：書籍・Web・資料閲覧サイト（実行時エラー）

本実験の評価項目についてまとめる。3章に述べた本研究の3つの課題に対して、以下の2つの仮説が成立すれば、

提案システムが有効な手段であることが示される。

- 仮説1 提案システムを利用した方が (TA がエラーを解決している過程を動画共有した方が), 学生がエラーの原因を理解しやすい (課題1 および2 に対応).
- 仮説2 提案システムを利用した方が (動画閲覧サイトを利用した方が), 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 (課題3 に対応).

なお前述したとおり, 本論文では学生の学習効果の面から評価することを目的としており, 講師や TA の指導の効率化については評価の対象外である。

6.2.2 実験結果

表3 にコンパイルエラーの場合の Wilcoxon 符号付順位和検定の結果, 表4 に実行時エラーの場合の Wilcoxon 符号付き順位和検定の結果を示す。検定の結果, 「書籍・Web」と「書籍・Web・資料閲覧サイト」とでは, コンパイルエラーの場合および実行時エラーの場合ともに有意差が認められた。

以上のことから, プログラミングでコンパイルエラーが起きた場合, 書籍や Web のみで対処するよりも, 資料閲覧サイトを併用して対処した方がエラーの原因を理解し, エラーを解決しやすいことが分かった。また, 実行時エラーが起きた場合, エラーの原因を理解し, 解決しやすいことが分かった。さらに「エラーを解決しやすい」という項目の結果より, トラブルの原因が比較的明確なコンパイルエラーより, 原因が曖昧な実行時エラーの対処を行う場合の方がエラーを解決する資料として有効であり, 資料閲覧サイトの利用価値が高いことが分かった。まとめると, 主に Q1, Q2 の結果から, 仮説1 が成立したと考察できる。また, Q3, Q4 の結果から, 仮説2 が成立したと考察できる。よって両仮説が成立したと認められることから, 本研究が対象とする学習効果が向上したと判断した。

表3 コンパイルエラーの場合の Wilcoxon 符号付き順位和検定
Table 3 Wilcoxon signed-rank test of compile error.

アンケート項目	1A の平均	1B の平均	Wilcoxon 符号付順位和検定 p 値
Q1 エラーの原因を理解しやすい	2.2	3.7	0.0234 *
Q2 エラーを解決しやすい	2.5	3.7	0.0078 **
Q3 対処方法に魅力がある	2.4	3.9	0.0156 *
Q4 この対処方法をまた利用したい	2.7	3.9	0.0313 *

n = 11, ** ... P < 0.01, * ... P < 0.05

表4 実行時エラーの場合の Wilcoxon 符号付き順位和検定
Table 4 Wilcoxon signed-rank test of run-time error.

アンケート項目	2A の平均	2B の平均	Wilcoxon 符号付順位和検定 p 値
Q1 エラーの原因を理解しやすい	2.6	4.2	0.0078 **
Q2 エラーを解決しやすい	2.6	4.2	0.0078 **
Q3 対処方法に魅力がある	2.5	4.3	0.0098 **
Q4 この対処方法をまた利用したい	2.7	4.0	0.0469 *

n = 11, ** ... P < 0.01, * ... P < 0.05

7. システムの使用感についての調査および考察

本研究で作成した資料閲覧サイトを被験者に利用させ, 本システムの使用感に関するアンケート調査 (主観評価) を行った。7.1 節では, 本アンケート調査について述べ, 7.2 節では本アンケート調査の結果に基づいて行った機能追加に関して述べる。

7.1 使用感についての調査

7.1.1 学生に対する調査

本アンケートは, 6章の「提案システムの有効性評価」に参加した被験者 11 名を含む大学生 13 名 (プログラミング学習を始めて 1~2 年目の大学生 13 名) に対して行った。

6章に述べた提案システムの有効性評価実験後に, 3つのアンケート項目に対し, 1 (あてはまらない)~5 (あてはまる) の 5 段階評価で回答させた。表5 に評価結果を示す。

アンケート評価の結果, すべてのアンケート項目が平均 3 (どちらともいえない) 以上となった。プログラミング初学者にとっておおむね好評または有効であることが分かった。なお, 表5 記載の評価結果は, 表3 および表4 記載の評価結果と評価内容の面でかなり重複していると考えられる。表3 および表4 の結果については, もし仮に「書籍と Web のみを利用した場合の使用感」をアンケート調査したとすれば平均は 3 未満となることが容易に推測される。

7.1.2 TA に対する調査

本研究で作成したクライアントツールの TA モードを被験者に利用させ, 本システムの使用感に関するアンケート調査 (主観評価) を行った。被験者は, 情報系学部の TA として, 実際の講義にたずさわったことがある大学院生 10 名を選んだ (情報系学部のプログラミング基礎演習クラスで TA 業務を経験したことのある大学院生を複数研究室から召集し, 本実験の被験者とした)。実験では, コンパイルエラーと実行時エラーの 2 通りのトラブルの対処を, それぞれ 2 つのプログラムに対して行ってもらった。実験終了後, 2 つのアンケート項目に対し, 1 (あてはまらない)~5 (あてはまる) の 5 段階評価を行ってもらった。

なお, 本システムは, エラー解決過程の収集および共有を行うことが目的であり, そのために TA の負担が多少増加することはやむをえないと考えている。そこで仮説として, 評価が 1 (あてはまらない) や 2 (ややあてはまらない)

表5 学生による評価のアンケート結果
Table 5 Results evaluated by students.

		平均
Q1	資料閲覧サイトは使いやすい	3.5
Q2	動画資料はプログラミングの理解に役立つ	3.7
Q3	実際のプログラミング講義で利用したい	3.4

n = 13

表 6 TA による評価のアンケート結果
Table 6 Results evaluated by TAs.

		平均
Q1	クライアントツール(TA モード)は使いやすい	3.2
Q2	実際の講義で利用したい	3.2

n = 10

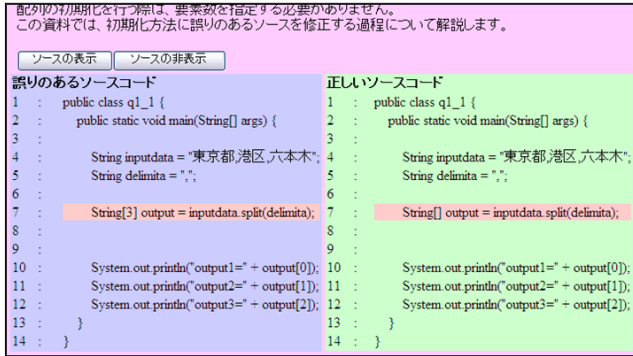


図 7 ソースコード比較

Fig. 7 Comparing source codes.

い) となるような場合以外は許容される範囲であると定めて調査を行った。表 6 に TA による評価の結果を示す。

アンケート結果, 平均 3 (どちらもいえない) 以上の評価を得ることができた。

7.2 機能追加

使用感についてのアンケート調査を行っている過程において, ソースコードのどこを直したのかが分かりやすくなるとよとの要望が複数学生からでた。そこで, クライアントツールにアップロードされた TA 修正前のソースコードと TA 修正後のソースコードとを並べて表示するソースコード比較ページを追加実装した。ソースコード中で編集された行の色が変わっているため, どのように編集されたのかを分かりやすくすることができた。図 7 にソースコード比較ページを示す。

8. おわりに

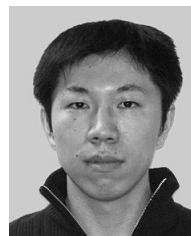
本研究では, TA が行ったトラブル解決手法を他の学生らと共有する方法に着目し, トラブルを解決するための資料を作成できるクライアントツールと, 資料を閲覧できる資料閲覧サイトからなるプログラミング演習支援システムを構築した。

プログラミング初学者に対して評価実験を実施した結果, 提案システムを導入した演習環境の方が, 従来の演習環境より, 学生が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。これにより本研究が対象とする学習効果が向上したと判断した。また, クライアントツールと資料閲覧サイトについて, TA と学生の両方に対して主観評価実験を行った結果, 利便性と有用性について

一定の評価を得た。以上のことから, 本論文で提案したエラー解決の過程を収集・蓄積・共有することの有効性を示すことができた。本研究は科研費 (20300276) の助成を受けたものである。

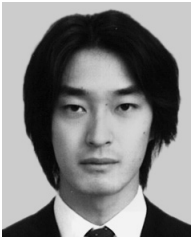
参考文献

- [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, 樋山淳雄, 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] Bennedsen, J. and Caspersen, M.E.: Revealing the Programming Process, *Proc. ACM 36th SIGCSE Technical Symposium on Computer Science Education*, pp.186-190 (2005).
- [9] FFmpeg (2011), available from (<http://ffmpeg.org/>).
- [10] Adobe Flash Player (2011), available from (<http://www.adobe.com/products/flashplayer/>).
- [11] Pais, R. and Barros, J.P.: Use of Flash Movies for Teaching GUI Programming, *Proc. ACM ITiCSE'05*, p.390 (2005).
- [12] 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 DICO 2011, 6B-2, pp.1035-1042 (2011).



安田 光 (学生会員)

2009 年東京工科大学コンピュータサイエンス学部卒業, 同年ケル株式会社入社。2010 年ケル株式会社退社。同年東京工科大学大学院バイオ・情報メディア研究科博士前期課程入学。現在, 東京工科大学大学院バイオ・情報メディア研究科博士前期課程に在学。



井上 亮文 (正会員)

1999年慶應義塾大学理工学部計測工学科卒業。2005年同大学大学院後期博士課程修了。博士(工学)。現在、東京工科大学コンピュータサイエンス学部講師。グループウェア、音楽情報処理の研究に従事。本会論文誌編集委員。ヒューマンインタフェース学会、ACM各会員。



市村 哲 (正会員)

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。