

学生と教師間でトラブル状況を共有できる プログラミング講義支援システム

提箸浩志 梶勇太 日比野祐介 市村哲

東京工科大学

1 はじめに

情報系大学ではプログラミング講義は必修科目である。しかし、現在のプログラミング講義では、プリントやスライド資料を使い、完成したソースコードを教師が説明するという方法が主流で、エラーの解決や、デバッグの方法を学ぶことができない。そのため、大学へ入学して初めてプログラミングを経験する学生にとっては、プログラミングを苦手とする場合が多い。また、トラブルを抱える多くの学生に対し小数のTA（ティーチング・アシスタント）が個別に対応するため、学生たちが抱えているトラブルを講義中にすべて解決できないという問題や、複数の学生が同じトラブルを抱えているのにTAが個別に対応するので効率がよくないという問題がある。

本研究では、プログラミングにおけるトラブルを解決する過程を動画資料として収録するクライアントツールと、学生が自分でトラブルを解決するための動画資料検索サイトからなるプログラミング講義支援システムを提案する。

2 プログラミング講義

2.1 講義の問題点

現在、プログラミング講義で使われる講義資料は、図 1 のようなプリントやスライド資料が主流である。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、このような静止面の講義資料では、プログラムが作成された過程を読み解くことが難しいため、実際にプログラミングを行う際に起こるエラーの解決や、デバッグの方法を学ぶことができない。

また、講義中にコンパイルエラーなどのトラブルを抱えた場合、学生は教師またはTAを呼び、解決策を聞くというのが一般的である。しかし、講義を受講する学生の数は多く、それに対しTAの数は少ないため、なかなか個別に対応することが難しく、講義中にトラブルを解決することができないという問題がある。さらに、複数の学生が

同じトラブルを抱えているのに、TAが個別に対応するので効率が悪いという問題もある。

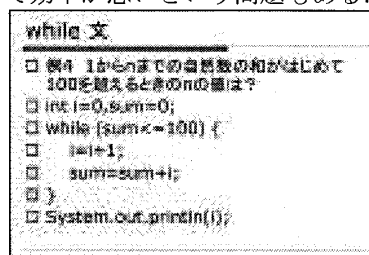


図 1 現在の講義資料例

2.2 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ひとつひとつを部品化し、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]が考えられている。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を主に支援しており、エラー解決などのトラブルを支援することは難しい。

3 予備実験

予備実験を行い、コンパイルエラーなどのトラブルを解決するために、どのような講義資料が適しているのかについて調査を行った。実験内容は、プログラムのソースコード全体を記した静止面に音声による解説を加えた講義資料と、ソースコードの入力過程を示した動画に音声による解説を加えた講義資料の2つを、大学生16名に閲覧させ、それぞれについてアンケート調査を行うというものである。なお、ソースコードの入力過程を示した動画とは、ソースコードの文字が、入力した順番に1文字ずつ表示されるものとなっている。

アンケート調査後、ウィルコクソンの符号付順位和検定を行った結果、「理解しやすい」「見やすい」「また視聴したいと思う」などのアンケート項目について有意水準1%での有意差が認められ、プログラミング講義で利用する講義資料は、ソースコードの入力過程を示した動画に音声の解説を加えたものの方が有効であることがわかった。今回提案するシステムでは、この形式の動画資料を学生たちへ提供することにした。

Programming Education System that
can share problem as student and teacher

Hiroshi Sagehashi, Yuuta Kunugi, Yuusuke Hibino,
Satoshi Ichimura
Tokyo University of Technology

4 提案

本論文では、プログラミング講義で学生が起こしたトラブルの解決過程を、エラー解決やデバッグ方法を学ぶための動画資料として収録する「クライアントツール」と、見たい動画資料を見つけ出し、その資料を閲覧するための「動画資料検索サイト」からなるプログラミング講義支援システムを提案する。図 2 にシステムの全体図を示す。

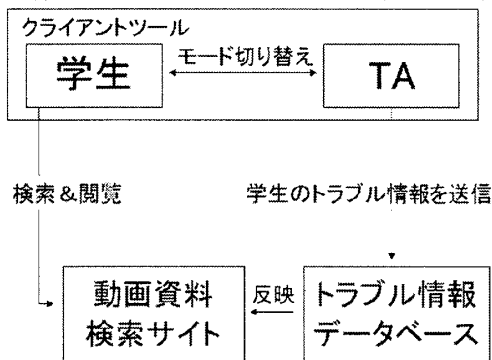


図 2 システムの全体図

5 実装

5.1 クライアントツール (学生モード)

講義開始後、学生はクライアントツールを用いてプログラミングを行う。クライアントツールにはソースコードの編集、コンパイル、実行機能があり、コンパイルと実行の結果もツール上で確認することができる。

作成しているプログラムでコンパイルエラーや実行時エラーなどのトラブルが起きた場合、動画資料検索サイトで動画資料を閲覧し、学生は自分の力で解決を試みる。

5.2 クライアントツール (TAモード)

学生が動画資料を閲覧してもトラブルを解決できない場合や、そもそも閲覧したい動画資料が存在しなかった場合は、TAを呼んでトラブルに対処してもらう。

学生に呼ばれたTAは、まずクライアントツール上でパスワードを入力することで学生モードからTAモードへと変更する。すると、TAがソースコードを修正する過程が自動的に記録され、動画資料として作成されるようになる。また、ソースコード中に解説を加えたい場合には、図 3 のように、動画中に吹き出しコメントとして説明を加える機能を持っている。さらに、動画資料の再生速度を変更することがクライアントツール上で可能である。TAがソースコードの修正を失敗した場合には、記録した動画のデータを削除する機能や、修正したソースコードをリセットする機能を利用して解決する。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動画に関する

説明文、動画資料を検索するためのタグ、編集したソースコードに該当する課題番号を入力し、動画データとまとめてクライアントツールからトラブル情報データベースへと送信する。

```

s1 = "12345";
s2 = "abc";
System.out.println("s1 = " + s1);
System.out.println("s2 = " + s2);
    
```

変数の値をコマンドライン上で表示してみる

図 3 吹き出しコメントの挿入例

5.3 動画資料検索サイト

動画資料検索サイトで動画資料を検索する方法として「課題番号検索」、「タグ検索」、「フリーワード検索」の3つがある。

課題番号検索では、TAがクライアントツール上で、動画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をもとに検索を行う。これにより学生は、自分が作成中の課題に合った動画資料を見つけ出すことができる。タグ検索では、課題番号同様、TAがクライアントツール上で入力したタグ(動画資料に関係のあるキーワード)の文字列をもとに検索を行う。また、コンパイルエラーを解決する動画資料の場合には、コンパイルエラーの内容がそのまま検索用タグとして利用できるようになっている。フリーワード検索では学生が任意に文字列を入力し、それをもとに、TAがクライアントツール上で入力したタグ、動画資料タイトル、動画資料の説明文に含まれる文字列の内容を検索する。

6 まとめ

今後、クライアントツールに音声録音機能を追加し、吹き出しコメントによる解説だけでなく、教師の音声による解説を付加する機能を追加していきたい。また、現在の動画資料はソースコードを1文字ずつ表示する方法を取っているが、変数の宣言や予約語を1単語単位で表示していく方法も検討している。さらに、実際のプログラミング講義において本システムを利用してもらい、有用であるかを評価していく予定である。

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).