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

安田 光[†]　　市村 哲[†]

現在のプログラミング講義は，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．そのため，エラー解決やデバッグ手法を学ぶことは難しい．また，履修する学生が多くTAが不足するため，トラブルには効率的な対応が必要である．そこで本研究では，トラブル状況を共有するため，学生がプログラミングの際に起こしたトラブルと解決方法を資料として収録するクライアントツールと，資料を閲覧するための資料検索サイトからなるシステムを構築した．クライアントツールでは，トラブルの発生したソースコードをTAが修正する過程を動画として記録するなどができる．

Programming Education System that can share Problems

KO YASUDA[†]　　SATOSHI ICHIMURA[†]

Recently, programming class is mostly cone using PowerPoint slide-show and paper. It is difficult for students to learn how to debug and resolve errors. Although, many students are not good at programming, the number of teaching assistants (TA) is not enough to teach students in the class. In this paper, we propose a process-recording-based programming education system composed of the process-recording material making tool and the process-recording material search sites where the process-recording material show how to solve problems in programming. When the students can not solve the problem, a TA corrects the source code. The TA can make the lecture-video by typing a usual program and insert balloon comments into a movie. Finally, the lecture-video is sent to the server from the process-recording material making tool.

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が必須科目となっている．現在のプログラミング講義では，完成したプログラムのソースコードをスライド資料上へ表示することで解説を行う方法が主流であり，このようなスライド資料は静的な講義資料であると言える．しかし，静的な講義資料で示されるプログラムは完成されたソースコードであるため，プログラムの動きや処理の流れまで読み解くことは難しく，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は難しい．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は，プログラミングを苦手とする場合が多い．

また，必須科目であるプログラミング講義を履修する学生の数は非常に多く，TAと学生ひとりひとりが個別に対応することが難しいため，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にもかかわらずTAが個別に対応するので効率が悪いという問題がある．

そこで本研究では，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を提案する．学生がプログラミングを行う際に起こしたトラブルを資料として収録するクライアントツールと，資料を閲覧するための資料検索サイトからなるシステムを構築した．

本システムを使用した評価実験によって，プログラミングを行う際に起こるトラブル状況を共有することがトラブル解決に有用であることが示された．

2. 背景・問題点

情報系大学や情報系学部においてプログラミング講義は必須科目である．現在の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主流である．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いため，プログラミングを行う際に起こるエラーの解決方法や，デバッグの方法を学ぶことができない．

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．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などに発生し，このままではプログラムの実行を行うことができない．コンパイルエラーには問題のあるソースコードの個所とそのエラー内容を説明する記述が含まれているため，その内容をもとにエラーを解決していく．

†東京工科大学
Tokyo University of Technology

それに対し実行時エラーは，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されたり，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にはデバッグ作業が必要となる．

3. 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]，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など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，振り返り学習を主に支援しており，エラー解決などの支援することは難しい．

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や，プログラミング講義における学生・講師間のコミュニケーション不足を解決する WEB を介した学生・講師間のコミュニケーションツールを用いたシステム[7]を開発した. 類似手法として, Revealing the Programming Process[8]が存在する．しかしながらこれらのシステムは，学生からトラブルの掬い上げを行うことができたが，トラブルを解決する直接的な手段は提供していなかった．

4. 提案

本論文では，プログラミング講義の際，学生が起こしたトラブルの解決方法を TA が指導し，その指導した解決方法をエラーの解決やデバッグ方法を学ぶための資料として収録するクライアントツールと，学生が必要としている解決方法を学生と TA 間で共有する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．本システムは学生が抱えるトラブル状況を講師や TA が収集･蓄積･共有することを主目的としている．主な対象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．

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．

図 1 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．プログラムをコンパイルおよび実行した際にトラブルが発生したら，資料検索サイトを参考に自己解決を試みる．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，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い，トラブルに対し適切な解決方法が資料検索サイトにあれば学生に閲覧させ自己解決を促す(図 1(a))．適切な資料がなく助言を行っても状況が改善しなかった場合に，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 1(b)).

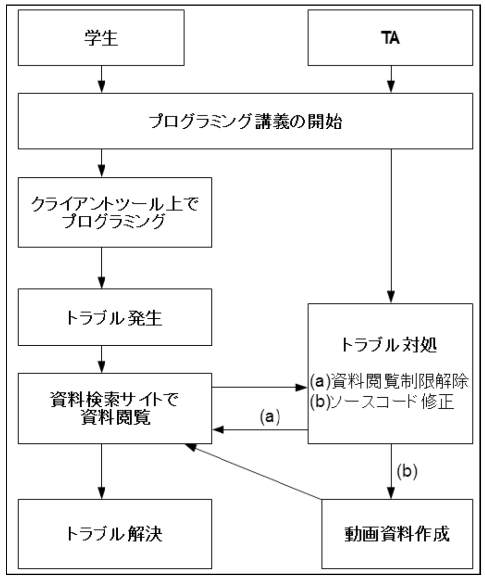

図 1 システムを利用する流れ
Figure 1 flow of system

以下，学生モードと TA モードについて詳細に述べる．学生モードでは学生がプログラミングを行うことができ，コンパイルエラーなどのトラブルが起きた場合には，資料検索サイトを利用できる．資料検索サイトではクライアントツールで作成された資料を閲覧することが可能で，学生に対し TA を呼ばずに自分の力で解決するように促すことができる．そのため，TA は学生から同じ質問を受ける回数が減り，少ない

TA でより多くの学生のプログラミング作業を助けることができる可能性がある．さらに，後述する動画資料を参照することで，文字のみの講義資料からでは学ぶ事が難しいデバッグテクニックを学ぶことができる．なお，閲覧する資料には制限がかけられており，当日の課題の解答になるソースコードや動画資料を学生は閲覧できないようになっている．

TA モードでは，当日の課題の解答ソースコードや動画解説資料を含めた全てを検索または一覧表示できる．また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も可能である．普段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．またこの動画資料をクライアントツールから資料サイトに送信できる．200 文字程度の修正であれば「動画作成」ボタンを押して約 2 秒で動画が作成される．

5. 実装

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．

前述したとおり，学生モードでは資料検索サイトの利用に閲覧制限が掛けられており，当日の課題の解答になるソースコードや動画資料は見られなくなっている(表 1 中の△印).

表 1 各モードの機能
Table 1 the function of modes

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	○	○
資料検索サイトの表示	△	○
資料の作成	−	○

クライアントツールはプログラミング講義の時間内に利用することを想定しており，学生はクライアントツールの学生モードで，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，コンパイルエラーの確認と，プログラミングを行う際に必要な作業が一通り行えるようになっている．また，コンパイルエラーなどのトラブルが起きた場合のために，資料検索サイトを表示するためのボタンがついている．

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確認ができる．TA モードの特徴として，TA がソースコードを修正する過程を自動的に記録することができる．普段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[9]している，これにより，TA のタイピング速度に関係なく指定した速度での動画が作成可能となっている．

また，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 2 に示すように動画中に吹き出しコメントとして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や，動画に 1 秒分の時間的スペースを空けるための時間を挿入することが可能である．TA がソースコードの修正を失敗した場合には，修正し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は，Web 上での利用を考えて FLV（Flashvideo）形式[10]で出力す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文，検索用タグ，編集したソースコードおよび，該当する課題番号を入力し，動画データと一緒にクライアントツールからサーバへと送信する．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，本システムの目的であるトラブル状況の収集および共有を行うことができると期待できる．

図 2 吹き出しコメント
Figure 2 balloon comment

5.2 資料検索サイト

資料検索サイトは，プログラミング講義を受講する学生が自分の PC 上で閲覧することができる．資料を検索するために以下のような複数の方法を用意している．

- 課題番号検索
- タグ検索
- フリーワード検索
- 修正前・修正後ソースコードの比較

- 動画資料の閲覧

資料検索サイトで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，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の文字列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資料を見つけ出すことができる．タグ検索では，課題番号司様．TA がクライアントツール上で入力したタグ（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．これにより学生は，自分が見たい内容を含む資料を，タグ検索を用いて簡単に一覧として表示することができる．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，資料タイトル，資料の説明文に含まれる文字列の内容を検索できる．

加えて，TA が編集する前のソースコードと編集したあとのソースコードを並べて表示することができ,さらにソースコード中で編集された行のみ着色されているため,プログラムがどのように編集されたのかが一目でわかるようになっている．図 3 にソースコード比較ページを示す．

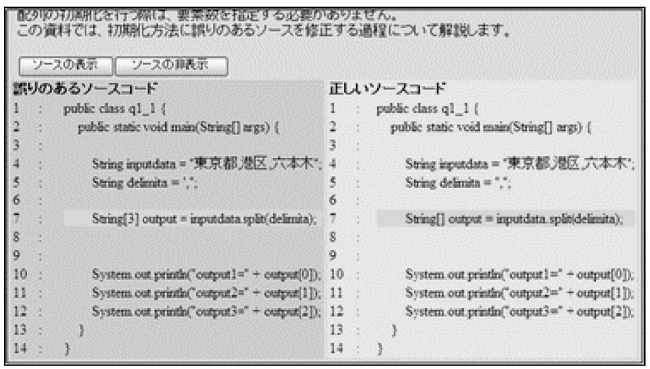

図 3 ソースコード比較ページ
Figure 3 page for comparing source codes

図 4 動画資料再生ページ
Figure 4 page of lecture-video

各資料には専用のページがあり，そこに埋め込まれた Flash 動画プレイヤーを用いて FLV 形式の動画資料を再生できる．また，資料のタイトル，資料の説明文，課題番号，タグの一覧，投稿日時が確認できる．課題番号と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課題番号検索，タグ検索が素早く行えるようになっている．図 4 に動画資料再生ページを示す．

6. 評価・考察

6.1 TA による評価

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してもらい，アンケート調査を行った．被験者は，情報系大学の 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．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． 表 2 に TA による評価の結果を示す．

表 2 TA による評価のアンケート結果
Table 2　the results from TA

		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

Q1，Q2 のアンケート結果では，平均 3 以上の評価を得ることができた．

6.2 学生による評価

本研究で作成したクライアントツールの学生モードを被験者に利用してもらい，アンケート調査を行った．被験者には，大学入学後約 1 年間プログラミングの学習をした大学生 13 名を選んだ．実験でに，あらかじめエラーの発生するソースコードを準備し，エラーが発生することを確認してもらった後に，資料検索サイトを用いてエラーを解決するための動画を検索・閲覧してもらった．なお本評価実験では，前節の「TA による評価」で利用したプログラムのソースコードと，この時 TA によって作成された動画資料を利用した．

実験終了後， 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 段階評価を行ってもらった． 表 3 に学生による評価の結果を示す．

アンケート評価の結果，全てのアンケート項目で平均 3 以上の評価を得ることができた．

表 3 学生による評価のアンケート結果
Table 3 the results from student

		平均
Q1	資料検索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

6.3 トラブルに対処する資料の評価

本研究で作成した資料検索サイトを利用してプログラミングにおけるトラブルに対処する方法と，従来通り書籍と Web を用いてトラブルに対処する方法の，2 つを被験者にそれぞれ利用してもらい，アンケート調査を行った．被験者には，大学入学後約 1 年間プログラミングの学習をした大学生 11 名を選んだ．実験では，コンパイルエラーおよび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，資料検索サイトを利用した修正の 2 通りを行ってもらった．なお，資料検索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 文を挿入することにより変数の表示等を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，5 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．

次に，「書籍・Web」と「資料検索サイト」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．

以下にトラブル対処方法の一覧を示す．

A1：書籍・Web（コンパイルエラー）
A2：書籍・Web（実行時エラー）
B1：資料検索サイト（コンパイルエラー）
B2：資料検索サイト（実行時エラー）

表 4 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 5 に実行時エラーの場合の Wilcoxon 符号付順位和検定の結果を示す．

検定の結果，「書籍・Web」と「資料検索サイト」では，コンパイルエラーの場合の Q3，Q5 および，実行時エラーの場合の Q5 のアンケート項目で，有意水準 5%で有意差が認められた．また，コンパイルエラーの場合の Q1，Q2，Q4，実行時エラーの場合の Q1，Q2，Q3 のアンケート項目で，有意水準 1%で有意差が認められた．

表 4 コンパイルエラーの場合の Wilcoxon 符号付順位和検定
Table 4　Wilcoxon signed-rank test of compile error

アンケート項目		A1 の平均	B1 の平均	Wilcoxon 符号付順位和検定 p 値
Q1	エラー原因を探しやすい	2.5	3.7	0.002223 **
Q2	エラーの原因を理解しやすい	2.2	3.7	0.00444 **
Q3	エラーを解決しやすい	2.5	3.7	0.02103 *
Q4	対処方法に魅力がある	2.4	3.9	0.004278 **
Q5	この対処方法をまた利用したい	2.7	3.9	0.03011 *

n = 11, ** … P < 0.01, * … P < 0.05

表 5 実行時エラーの場合の Wilcoxon 符号付順位和検定
Table 5 Wilcoxon signed-rank test of run-time error

アンケート項目		A2 の平均	B2 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	0.001871 **
Q2	エラーを解決しやすい	2.6	4.2	0.00182 **
Q3	対処方法に魅力がある	2.5	4.3	0.001616 **
Q4	この対処方法をまた利用したい	2.7	4.0	0.02566 *

n = 11, ** … P < 0.01, * … P < 0.05

また，資料検索サイトにある動画資料を閲覧してもらい，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 段階評価を行ってもらった．同時に，自由記入欄に意見や感想を記述してもらった．表 6 にアンケート結果を示す．

動画資料の機能についてアンケート評価を行った結果，全てのアンケート項目で平均 3.7 以上の評価を得ることができた．そのため，実際の講義で学生に動画資料を閲覧してもらう場合でも，動画の表示は見やすく，プログラミングを行う際のテクニックやデバッグの方法が理解しやすいと考える．

表 6 動画資料に対するアンケート結果
Table 6 the result of lecture-video

		平均
Q1	デバッグ方法は参考になった	3.7
Q2	吹き出しコメントはわかりやすかった	3.7
Q3	1 文字ずつ表示する形式は見やすい	3.8

n = 11

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 で対処するよりも，資料検索サイトを利用して対処した方が，自分が見たい資料を見つけやすく，エラーの原因を理解し，エラーを解決しやすいことがわかった．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わかった．「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，資料検索サイトの利用価値が高いことがわかった．

7. 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たち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，自分が見たい資料を簡単に検索できる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構築した．

クライアントツールと資料検索サイトについて，TA 側と学生側の両方から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．また，動画資料について評価実験を行った結果，プログラミングを行う際に起こるトラブルを解決する資料として，書籍や Web よりも有用であることを示した．

動画資料に対し，ソースの色分けや音声が欲しいといった要望もあったため，動画資料に対し色情報や音声情報などを付加する機能が必要と考える．

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, 櫨山淳雄, 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) Nghi Truong, Peter Bancroft, Paul Roe: Learning to program through the web, In Proceedings of the 10th annual SIGCSE conference on Innovation and technology in computer science education, pp.9-13 (2005)
6) Thomas P. Way: A company-based framework for a software engineering course, In Proceedings of the 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講義支援システム, 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. 2005. Revealing the programming process. In Proceedings of the 36th SIGCSE Technical Symposium on Computer Science Education (St. Louis, Missouri, USA, February 23 - 27, 2005). SIGCSE '05. ACM, New York, NY, 186-190.
9) FFmpeg
http://ffmpeg.org/
10) Adobe Flash Player
http://www.adobe.com/products/flashplayer/