

# 穴埋め問題を用いたプログラミング教育支援ツール pgtracer のログデータ分析機能

太田康介<sup>†1</sup> 掛下哲郎<sup>†1</sup>

我々は穴埋め問題を用いたプログラミング教育支援ツール pgtracer を開発している。本ツールは学生に対し、C++のプログラムとトレース表の組に対する穴埋め問題を出題する。学生が問題に解答すると、pgtracer は学生が入力した解答、所要時間、採点結果などをログデータとして収集する。本稿では、収集したログデータを分析する 7 種類の分析機能について述べる。ログデータ分析機能は、学生毎の分析機能、問題毎の分析機能、解答過程や学習履歴の分析機能に大別されており、様々な詳細度で分析結果を表示できる。教員はログデータ分析機能を活用することで、学生の理解度や問題の難易度を把握できる。教員は分析結果を考慮して、学生に適切な問題を作成することができる。

## A Programming Education Support Tool pgtracer utilizing Fill-in-the-Blank Questions : Log Data Analysis Function

Kosuke Ohta<sup>†1</sup> Tetsuro Kakeshita<sup>†1</sup>

We are developing a programming education support tool pgtracer utilizing fill-in-the-blank questions. This tool provides fill-in-the-blank questions composed of a C++ program and a trace table to students. When a student answers a question, pgtracer collects student's answers, elapsed time, evaluation result, etc. as student log data. In this paper, we develop seven functions for log data analysis. The data analysis functions are classified into the analysis function for a student, that for a question, and that for a process of an answer. A teacher can analyze understanding level of the students and difficulty level of the problems utilizing the analysis functions. A teacher can create adequate questions to the students by considering the analysis results.

### 1. はじめに

プログラミング教育は理工系の大学や高専において重要性が高いが、学生の学力低下に関する懸念や、プログラミング実習時に教員や TA 等だけでは十分な指導が行えない等の課題がある。そこで我々は穴埋め問題を用いたプログラミング教育支援ツール pgtracer の開発・運用を行っている[1,2,3]。

本ツールは大学等で広く普及している e-Learning システム Moodle[4]のプラグインモジュールとして動作し、プログラムとトレース表に対して、いくつかの箇所穴抜きを行った穴埋め問題を学生に出題する。pgtracer では学生が問題を受験する際の内容を細かく収集しており、受験毎の点数、開始時刻、終了時刻などを学習履歴として、1 つ 1 つの穴抜きに解答する際の答案、所要時間などを解答履歴として、最終的な解答や正誤などを最終解答履歴として収集する。教員はこれらのログデータを分析することで、学生の理解度の把握や得意箇所を特定し、教育改善に役立てることができる。

本稿では、収集したログデータを pgtracer 上で分析できる、ログデータ分析機能について示す。ログデータ分析機

能は、学生毎の分析、問題毎の分析、学習履歴の分析からなる 7 つの機能を実装した。2 節ではプログラミング教育支援ツール pgtracer とはどのようなものなのかについて述べる。3 節では今回実装したログデータ分析機能の概要について述べ、4 節で学生毎の分析機能、5 節で問題毎の分析機能、6 節で学習履歴の分析機能について述べる。7 節では各機能の実装について、8 節では実装した機能に対しての試用者からのレビューコメントとそれに対する検討について述べる。

### 2. プログラミング教育支援ツール pgtracer

pgtracer はプログラミング初学者や低学力の学生といった 1 からプログラムを作成できない学生を対象としたシステムとして開発を行っている。そのため、プログラムを 1 から記述するといった出題形式ではなく、穴埋め形式の問題を採用している。穴埋め形式にすることで穴埋めの位置や大きさを変更することで容易に難易度を変化させることが可能で様々な問題を作成できる。

本ツールで出題する穴埋め問題はプログラムだけでなく、プログラムの実行過程を示したトレース表にも穴抜きを行える(図 1)。pgtracer ではプログラムとトレース表が一貫した状態になるようにする作業をプログラミングとし、

<sup>†1</sup> 佐賀大学  
Saga University

記述されているコメント（アルゴリズム）からプログラムを作成するだけでなく、トレース表の変化からプログラムを再現する作業もプログラミングに含める。

解答画面 試験モード

注意事項: ①プログラム内のは半角スペースを表しています。  
 ②トレース表内の出力値において改行を表したい場合は\nを入力して下さい。  
 ③画面サイズが足りず問題が見える場合は、ページのズームを縮小して下さい

step	プログラム	ルーチン	step	main	main	main
	#include <iostream > using namespace std; int main () {					
1	int num1, num2;	main	1	?	?	
2	int tmp;	main	2	?	?	?
	//標準入力から入力を受け取る	main	3	?	1	?
3	cin >> num1;	main	4			
4	cin >> num2;	main	5			
	//入力した2つの整数の値を入れ替える	main	6			
5	tmp = num1;	main	7			
6	num1 = num2;	main	8	1	99	1
7	num2 = tmp;					
8	return 0;					
	}					

解答終了

図 1 pgtracer の問題画面

図 2 は pgtracer を活用したプログラミング教育過程の概念図である。教員はまず、問題作成機能を用いて学生に出題する穴埋め問題を作成・編集する。pgtracer では穴埋め問題をプログラムとトレース表、穴抜きの情報を持つプログラム用マスク、トレース表用マスクの 4 種類の XML ファイルの組み合わせで定義する。本ツールではソースプログラムからプログラムとトレース表の XML ファイルを、GUI 上での操作でマスク用の XML ファイルを編集する機能を提供している。プログラムの穴抜きはプログラムの最

小単位である「トークン」、いくつかのトークンをまとめた「複数のトークン」、プログラムの 1 行である「文全体」に対して設定できる。一方、トレース表においては「ステップ番号」や「変数名」、「変数値」、「出力値」に対して穴抜きを設定できる。教員はこれらの穴抜きを任意に組み合わせ、問題のタイトルと難易度、配点を設定し 1 つの問題として pgtracer に登録できる。

学生が pgtracer にアクセスすると登録されている問題がタイトル、難易度毎に一覧表示され、問題を選択すると、図 1 のような穴埋め問題が出題される。pgtracer では、学生が 1 つ 1 つの穴抜きに解答する際の学生が入力した解答や所要時間を解答履歴として収集する。学生が解答を終えると pgtracer は自動採点を行う。自動採点では、 $c=a*b$  が正答の穴に  $c=b*a$  と解答した場合など、意味的には間違っていないが正答とは一致しない場合も考慮し、正答との一致判定だけでなく、解答プログラムの実行を通じた正誤判定も行っている。自動採点の結果、得点や学習開始時刻、終了時刻は学習履歴として、それぞれの穴抜きの学生の最終的な解答や正誤は最終解答履歴として収集する。

教員は収集したログデータを分析し、個々の学生や全体の理解度・不得意箇所を把握する。また、問題を構成する穴抜きの種類と所要時間から穴埋め問題の難易度を評価する[5]。その後、分析結果をもとに、理解度が低かった箇所や学生の不得意な箇所を重点的に教育できるように問題を作成する。この PDCA サイクルを繰り返すことにより、プログラミング教育を継続的に改善できると考えている。

これまで我々は問題作成機能や問題の出題機能、ログデータ収集機能の実装を行ってきた。本稿では残りの機能であるログデータ分析機能の企画および実装を行う。

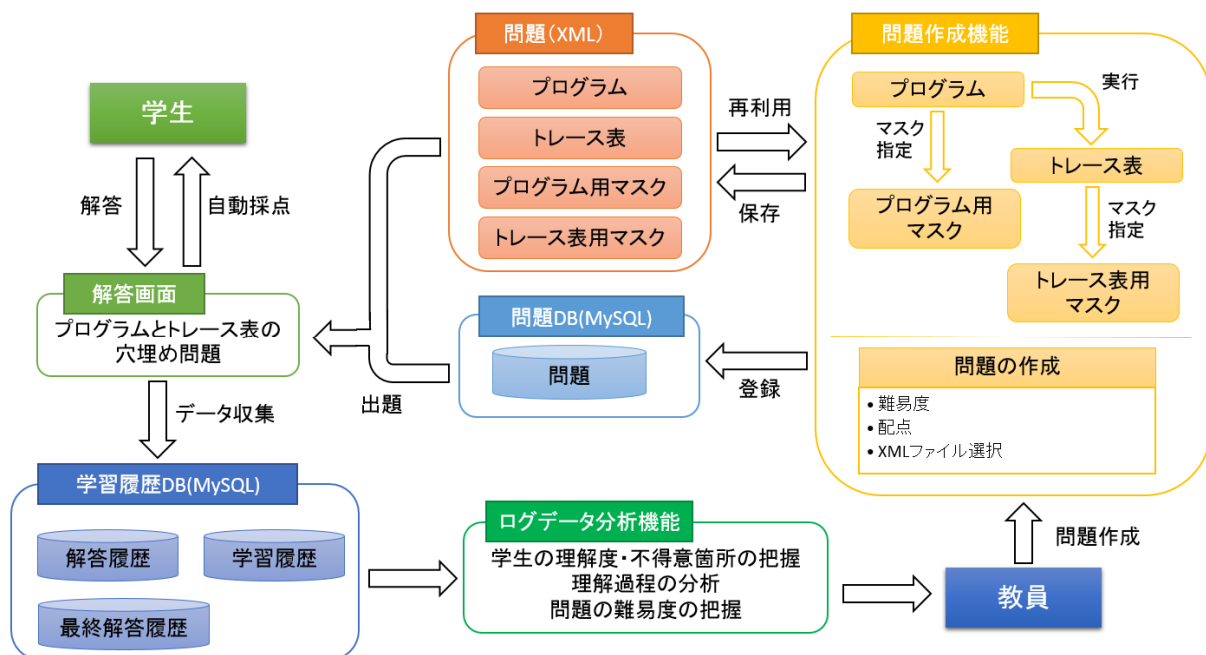


図 2 pgtracer を用いたプログラミング教育過程

### 3. ログデータ分析機能の概要

pgtracer で収集するログデータを表 1～表 3 に示す。

学習履歴 (表 1) は学生が解答する問題を選択した時点で生成され、解答終了した時点で更新される。解答履歴 (表 2) は学生が個別の穴抜きに解答した際に生成される。解答履歴を用いることで、学生の解答過程を分析できる。また、最終解答履歴 (表 3) は、学生が解答を終了し、それぞれの穴抜きの自動採点が行われる際に生成される。学生は、穴埋め問題を解く過程で、いったん記入した解答を変更できる。解答履歴は中間解答を含むが、最終解答履歴は最終的な答案のみを含む点が異なっている。

表 1 学習履歴

名前	データ型	説明
id(主キー)	int	学習履歴 ID
user_id	int	ユーザ ID
question_id	int	問題 ID
point	int	得点
start_time	int	学習開始時刻
end_time	int	学習終了時刻

表 2 解答履歴

名前	データ型	説明
id(主キー)	int	解答履歴 ID
study_id	int	学習履歴 ID
blank_path	varchar	穴抜きの XPath
answer	varchar	学生の解答
correct_answer	varchar	正答
elapsed_time	int	穴埋めに要した時間

表 3 最終解答履歴

名前	データ型	説明
id(主キー)	int	最終解答履歴 ID
study_id	int	学習履歴 ID
blank_path	varchar	穴抜きの XPath
answer	varchar	学生の解答
correct_answer	varchar	正答
correct	int	自動採点の結果

ログデータ分析機能は、様々な観点や詳細度でログデータを分析する機能を提供する。ログデータの分析には様々な方法が考えられるが、主要な観点としては、学生毎の分析、問題毎の分析、解答過程や学習履歴の分析の3つが挙げられる。

- 学生毎の分析機能

学生毎の分析機能では、1人の学生と問題全体に対する分析機能を提供する。提供する機能は、学生毎の利用状況表示機能、学生毎の受験履歴表示機能の2つである。

これらの機能を用いることで、教員は学生がどの程度 pgtracer を利用したか、個々の学生の各問題の受験結果、学生が問題に対してどのような解答をしているのかを分析できる。

- 問題毎の分析機能

問題毎の分析機能では、1つの問題に対する学生全体の分析機能を提供する。提供する機能は、問題毎の学習履歴全体の分析機能、問題毎の受験者一覧表示機能、問題分析機能の3つである。

これらの機能を用いることで、教員は問題にどの程度の学生が解答しているか、個々の問題の各学生の受験結果、問題を構成する穴抜きの分析といった分析方法などが考えられる。

- 解答過程や学習履歴の分析機能

解答過程や学習履歴の分析機能では、1人の学生と1つの問題の組み合わせに対する分析機能を提供する。提供する機能は学生の問題毎の受験履歴表示機能と解答過程の分析機能の2つである。

教員は学生毎の受験履歴表示機能や問題毎の受験者一覧表示機能で選択した学生と問題の組み合わせに対して、より詳細な受験履歴を確認し、個々の受験に対して解答過程の分析機能で詳細な解答過程を分析できる。

これらの機能の画面遷移図を図 3 に示す。関連する機能間では双方向での遷移ができるように分析機能を設計した。各機能の詳細については4節以降で説明する。

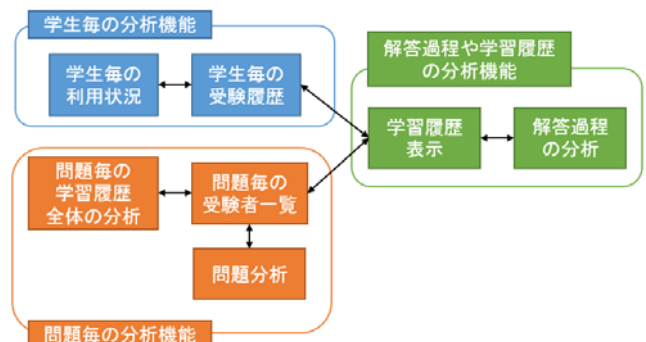


図 3 分析機能の画面遷移図

### 4. 学生毎の分析機能

#### 4.1 学生の利用状況表示機能

学生の利用状況表示機能 (図 4) では、学生毎の pgtracer の利用状況を表示する。登録されている問題のうち何問に解答したかを示す「解答問題数」、同じ問題に何度も解答した場合を含めた「延べ解答回数」、解答に要した時間の合計を表す「合計利用時間」、解答した問題の最高点の合計÷解

答した問題の満点の合計を表す「正解率」を表示する．表示されているデータは、複数の項目を用いてソートが可能となっている．

学生の利用状況

登録問題数 80問

学生	解答問題数(問)	延べ解答回数(回)	合計利用時間	正解率(%)	
	88	103	4時間10分13秒	96.14	表示
	56	70	3時間28分39秒	100.00	表示
	52	70	5時間39分47秒	100.00	表示
	47	54	3時間30分51秒	100.00	表示
	47	52	3時間32分12秒	96.21	表示
	47	64	3時間4分6秒	100.00	表示
	46	47	1時間33分41秒	94.78	表示
	45	50	3時間18分17秒	96.64	表示
	45	56	3時間17分27秒	100.00	表示
	45	67	2時間35分55秒	100.00	表示
	44	48	2時間38分24秒	95.27	表示

図 4 学生の利用状況表示機能

教員はこの機能を利用することで、学生の学習状況や学習意欲を把握できる．例えば、解答問題数でソートすることで指定された問題数に解答しているかを確認できる．延べ解答回数や合計利用時間でソートすることで学習意欲が高い学生を探し出すことができる．また、延べ解答回数や合計利用時間からは、学生にかかる負担がわかる．これらを参照しながら問題を出題することで、学生の負荷を調節できる．

#### 4.2 学生毎の受験履歴表示機能

学生毎の受験履歴表示機能（図 5）では、利用状況表示機能で選択された学生の問題の受験結果を表示する．表示する内容は、問題を初めて見た結果である初回と学習を繰り返した結果である最高点時の「点数」、「学習開始時刻」、「所要時間」と問題に何回解答したかを示す「解答回数」を表示する．学生が解答していない問題には各項目をハイフンで表示する．表示されているデータは、複数の項目を用いてソートが可能となっている．

学生の利用状況

登録問題数 80問

テーマ	タイトル	難易度	初回受験結果			最高点受験結果			解答回数	
			点数	学習開始時刻	所要時間	点数	学習開始時刻	所要時間		
評価実験(自習用)	問題19	1	100	2015/01/08 17:11:02	0時間0分24秒	100	2015/01/08 17:11:02	0時間0分24秒	1	表示
評価実験(自習用)	問題19	2	100	2015/01/08 23:02:05	0時間0分33秒	100	2015/01/08 23:02:05	0時間0分33秒	1	表示
評価実験(自習用)	問題05	1	100	2015/01/08 16:44:26	0時間0分36秒	100	2015/01/08 16:44:26	0時間0分36秒	1	表示
評価実験(自習用)	問題18	2	91	2015/01/08 23:01:20	0時間0分37秒	91	2015/01/08 23:01:20	0時間0分37秒	1	表示
評価実験(自習用)	問題13	2	100	2015/01/08 22:47:40	0時間0分40秒	100	2015/01/08 22:47:40	0時間0分40秒	1	表示
評価実験(自習用)	問題19	1	100	2015/01/04 01:29:09	0時間0分42秒	100	2015/01/04 01:29:09	0時間0分42秒	1	表示
評価実験(自習用)	問題17	2	100	2015/01/08 23:00:31	0時間0分43秒	100	2015/01/08 23:00:31	0時間0分43秒	1	表示
評価実験(自習用)	問題22	1	70	2015/01/08 17:18:24	0時間0分44秒	70	2015/01/08 17:18:24	0時間0分44秒	1	表示
評価実験(自習用)	問題10	1	100	2015/01/08 16:53:02	0時間0分50秒	100	2015/01/08 16:53:02	0時間0分50秒	1	表示
評価実験(自習用)	問題08	2	86	2015/01/08 17:32:16	0時間0分53秒	86	2015/01/08 17:32:16	0時間0分53秒	1	表示
評価実験(自習用)	問題03	2	100	2015/01/04 01:58:43	0時間0分55秒	100	2015/01/04 01:58:43	0時間0分55秒	1	表示
評価実験(自習用)	問題02	2	63	2015/01/08 17:21:24	0時間0分55秒	63	2015/01/08 17:21:24	0時間0分55秒	1	表示
評価実験(自習用)	問題05	2	66	2015/01/08 17:25:17	0時間0分55秒	66	2015/01/08 17:25:17	0時間0分55秒	1	表示

図 5 学生毎の受験履歴表示機能

教員はこの機能を利用することで、学生が指定されている問題に解答しているか確認できる．また、学生が苦手と

する問題を探し出せる．例えば、問題名でソートし各項目を見ることで指定された問題に解答しているか確認でき、点数でソートすると解答していない問題をまとめて確認できる．また、点数や所要時間でソートすることで、学生が苦手とする問題を探し出すことができる．初回受験時の結果と最高点時の結果を見比べることで、繰り返しの効果などの程度あったかを確認できる．この機能で特定した問題を学生がどのように解答しているか等、さらに細かく分析したい場合は解答過程や学習履歴の分析で提供する機能を利用する．

### 5. 問題毎の分析機能

#### 5.1 問題毎の学習履歴全体の分析機能

問題毎の学習履歴全体の分析機能（図 6）では、登録されている各問題の初回と最高点時の「平均点」、「平均所要時間」、「解答者数」、「平均学習回数」を表示する．学生が誰も解答していない問題は、各項目をハイフンで表示する．表示されているデータは、複数の項目を用いてソートが可能となっている．

問題毎の平均点

テーマ	タイトル	難易度	初回受験		最高点		解答者数	平均学習回数	
			平均点	平均所要時間	平均点	平均所要時間			
評価実験(自習用)	問題09	2	100.00	145.54	100.00	145.54	28	1.00	表示
評価実験(自習用)	問題16	1	100.00	288.40	100.00	288.40	20	1.00	表示
評価実験(自習用)	問題16	2	100.00	177.05	100.00	177.05	20	1.00	表示
評価実験(自習用)	問題17	2	100.00	91.64	100.00	91.64	28	1.00	表示
評価実験(自習用)	問題19	1	100.00	85.00	100.00	85.00	28	1.00	表示
評価実験(自習用)	問題19	2	100.00	70.21	100.00	70.21	28	1.00	表示
評価実験(自習用)	問題21	2	100.00	463.93	100.00	463.93	28	1.00	表示
評価実験(自習用)	問題15	2	99.89	230.18	99.89	230.18	28	1.00	表示
評価実験(自習用)	問題11	1	99.79	295.93	100.00	290.43	28	1.07	表示
評価実験(自習用)	問題17	1	99.71	235.93	99.86	228.00	28	1.04	表示

図 6 問題毎の学習履歴全体の分析機能

教員はこの機能を利用し、平均点や平均所要時間、平均受験回数でソートすることで学生全体が苦手とする問題を探し出せる．また、初回受験時の平均と最高点時の平均を見比べることで繰り返しの結果どの程度の効果があったかを確認できる．また、問題の難易度が適切に設定されているかを確認する．例えば、同じタイトルの異なる難易度間において、難易度が低い方が点数は高く、所要時間は短くなると思われる．しかし、逆にになっている場合は穴抜きの設定や、難易度の設定がうまくいってないと思われる．

#### 5.2 問題毎の受験者一覧表示機能

問題毎の受験者一覧表示機能（図 7）では、選択された問題の受験結果の一覧を表示する．表示する内容は、学生毎に初回と最高点時の「点数」、「学習開始時刻」、「所要時間」と問題に何回解答したかを示す「解答回数」を表示する．問題に解答していない学生は、各項目をハイフンで表示する．表示されているデータは、複数の項目を用いてソートが可能となっている．

問題01難易度1の受験履歴						
初回受験結果		最高点		平均学習回数		
平均点	平均所要時間	平均点	平均所要時間	問題分析		
95.55点	226.74秒	100.00点	224.95秒	1.87回		

学生	初回受験結果				最高点受験結果				解答回数
	点数	学習開始時刻	所要時間	点数	学習開始時刻	所要時間			
14233	0	2015/01/09 12:32:03	0時間0分59秒	100	2015/01/09 12:32:52	0時間0分16秒	3	表示	
14233	100	2014/12/22 11:10:52	0時間1分14秒	100	2014/12/22 11:10:52	0時間1分14秒	1	表示	
14233	100	2015/01/07 15:29:41	0時間1分22秒	100	2015/01/07 15:29:41	0時間1分22秒	1	表示	
14233	100	2014/12/27 23:52:08	0時間1分23秒	100	2014/12/27 23:52:08	0時間1分23秒	1	表示	
14233	100	2014/12/28 14:56:35	0時間1分24秒	100	2014/12/28 14:56:35	0時間1分24秒	1	表示	
14233	100	2014/12/21 16:01:22	0時間1分26秒	100	2014/12/21 16:01:22	0時間1分26秒	1	表示	
14233	100	2015/01/09 12:54:21	0時間1分27秒	100	2015/01/09 12:54:21	0時間1分27秒	2	表示	
14233	100	2014/12/27 23:42:57	0時間1分32秒	100	2014/12/27 23:42:57	0時間1分32秒	1	表示	
14233	100	2015/01/08 10:16:43	0時間1分32秒	100	2015/01/08 10:16:43	0時間1分32秒	1	表示	
14233	100	2015/01/06 11:53:11	0時間1分36秒	100	2015/01/06 11:53:11	0時間1分36秒	1	表示	

図 7 問題毎の受験者一覧表示機能

教員はこの機能を利用することで、問題に解答していない学生を探し出せる。また、この問題を苦手とする学生を把握できる。例えば、学生名でソートし各項目を見ることで解答しているか確認でき、点数でソートすると解答していない学生をまとめて確認できる。また、点数や所要時間でソートすることで、この問題を苦手とする学生を探し出すことができる。初回受験時の結果と最高点時の結果を見比べることで繰り返しの結果どの程度効果があったかを確認できる。この機能で特定した学生が問題に対しどのように解答しているか等、さらに細かく分析したい場合は解答過程や学習履歴の分析で提供する機能を利用する。

問題01難易度1の分析						
step	プログラム	ルーン	step	main	main	main
1	int main() {	a	b	c	d	
2	int a;					
3	int b = 10;					
4	int c[] = {2, 4, 6, 8, 10};					
5	int d[] = {2, 4, 6, 8, 10};					
6	int i;					
7	for(i=0; i<c.size(); i++) {					

初回受験時	最高点時
正答率	正答率
97%	100%
平均所要時間	平均所要時間
21.17	20.97
平均解答回数	平均解答回数
1.24	1.24

解答	人数	正答	所要時間	解答回数
1	28	21.43	30	20.3
2	1	14	0	0

学生	解答	正答	所要時間	解答回数
14233	1	○	7	1
14233	1	○	10	1
14233	1	○	11	2
14233	1	○	9	1
14233	1	○	9	1
14233	1	○	10	1

図 8 問題分析機能

### 5.3 問題分析機能

問題分析機能(図8)では、問題を構成する個々の穴抜きの解答結果を表示する。それぞれの穴抜きを選択すると、その穴抜きの初回受験時と最高点時の「正答率」、「平均所要時間」、「平均解答回数」が表示される。また、各学生の「最終的な解答」や「正誤」、「所要時間」、「解答回数」

も表示し、解答の種類毎に集計した結果も表示する。また、学生によっては解答途中に席を離れるなどし、解答時間が1000秒など異常に長くなってしまう場合がある。そこで、所要時間の範囲を指定し、平均を再計算できるようにしている。表示されているデータは、複数の項目を用いてソートが可能となっている。

教員はこの機能を利用することで、問題を構成する個々の穴抜きの難易度を把握できる。また、個々の学生の解答内容を具体的に把握できる。

## 6. 解答過程や学習履歴の分析機能

### 6.1 学習履歴表示機能

学習履歴表示機能(図9)では、1人の学生が解いた1つの問題に対する学習履歴を表示する。表示する内容は、各受験回の点数と学習開始時刻、所要時間である。表示されているデータは、複数の項目を用いてソートが可能となっている。

問題09難易度1の受験履歴			
受験回数	点数	学習開始時刻	所要時間
1回目	75	2015/01/03 20:31:07	0時間1分53秒
2回目	100	2015/01/03 20:33:08	0時間0分30秒

step	プログラム	ルーン	step	main	main	main	出力
				input_value	input_value	input_num	
main	1	?	?	?	?	?	
main	2	?	?	?	?	?	
main	3	?	?	?	?	9	
main	4	?	?	?	?	?	
main	5	13.5	?	?	?	?	
main	6	?	?	?	?	?	
main	7	?	?	?	?	?	

```

#include <iostream>
using namespace std;
int main() {
    int input_num;
    double input_value;
    cin >> input_num;
    cout << input_num * 2 << endl;
    cin >> input_value;
    cout << input_value / 3.0 << endl;
    return 0;
}
    
```

図 9 学生の問題毎の学習履歴表示機能

教員はこの機能を利用することで、繰り返し解答した結果を確認できる。受験回を追う毎に点数が高くなっている場合や、所要時間が短くなっているような場合は繰り返し学習の効果があると考えられる。一方、初回受験時の所要時間が極端に短い場合、早々に諦めて正解を表示しようとしたことも判別できる。

### 6.2 解答過程の分析機能

解答過程の分析機能(図10)では、1つの学習履歴に対して学生が問題に解答していく過程を表示する。ページ左上部に学生が解答した順番で学生の解答、正答、所要時間を表示する。また、ページ下部には問題が表示されており、解答履歴の表のラジオボタンや戻る/進むボタンで選択した時点での穴埋めの状態が再現され、進めることで解答過程を視覚的に確認できる。ラジオボタンで選択された解答履歴に該当する穴抜きは黄色で表示される。

教員はこの機能を利用することで、学生が苦手としている箇所の把握ができる。例えば、何度も繰り返し解答しな

おしている場合や、所要時間が長い穴抜きは苦手としている箇所と考えられる。また、問題によっては学生の解答過程に違いが出ることがわかっており、点数の良い学生と悪い学生の解答過程を確認することで指導の参考になることが期待される。

The screenshot shows a web interface for analyzing student solutions. At the top, there's a header for '問題09難易度1の解答過程' (Problem 09 Difficulty 1 Solution Process) with a date '2015/01/03 20:31:07' and a '表示' (Display) button. Below this is a table with columns '解答' (Answer), '正答' (Correct Answer), and '所要時間' (Time Taken). The table lists 10 rows of student data. To the right of the table are buttons '戻る' (Back) and '進む' (Next). Below the table is a code editor window titled 'プログラム' (Program) showing C++ code for a program that takes two integers and outputs their sum and average. The code is as follows:

```

#include <iostream>
using namespace std;
int main () {
    int input_num;
    double input_value;
    cin >> input_num;
    cout << input_num * 2 << endl;
    cin >> input_value;
    cout << input_value / 3.0 << endl;
    return 0;
}
    
```

図 10 解答過程の分析機能

## 7. 各機能の実装

pgtracer は広く普及している LMS Moodle[4]のプラグインとして実装されている。開発言語は PHP と JavaScript であり、データベースには MySQL を使用している。以下にログ分析機能の実装における主要な工夫点を示す。

### ● 受験結果の表示

多くの機能では初回受験時と最高点時の受験結果を表示しているが、学習履歴をもとに生成している。学習履歴では学生の受験毎に点数や学習開始時刻と終了時刻を収集している。点数については学習履歴の値を、所要時間については学習開始時刻と終了時刻の差を所要時間として使用している。各機能で初回受験と最高点の場合を表示しているが、初回受験の場合は学習開始時刻が1番小さいレコード、最高点の場合は点数が1番大きく、学習開始時刻が1番小さいレコードを使用している。

### ● ソート機能の実装

多くの機能では表示されている表のソートができるようになっている。これは jQuery のプラグインである tablesorter[6]を使用して実装した。tablesorter プラグインでは表のヘッダー部分をクリックすることで昇順・降順ソートができるようになっている。また、複数の項目を用いた

ソートも、シフトボタンを押しながら列をクリックすることで可能である。

### ● 問題分析機能

問題分析機能では、穴抜き毎に初回受験と最高点時の正解率や平均所要時間、解答の種類毎の集計、各学生の最終的な解答と正誤、所要時間、解答回数を表示する。

最終的な解答や正誤は、学生毎に初回受験と最高点時の学習履歴に対応する最終解答履歴から穴抜きの XPath 式に対応するレコードの学生の解答と正誤を取得しており、合計所要時間と解答回数は、初回受験と最高点時の学習履歴に対応する解答履歴から XPath 式に対応するレコードの所要時間の合計とレコードの個数を集計し使用する。

穴抜きの解答結果を表示している表は、穴抜き毎に用意されており、穴抜きが選択されていない時点では非表示になっている。利用者が穴抜きを選択すると JavaScript が動作し、選択された穴抜きに対応する表を表示する。それ以外の表は非表示のままにする。

### ● 解答過程の分析機能

解答過程の分析機能では、学生が解答した順序で学生の解答、正答、所要時間を表示しているが、これは解答履歴をもとに表示している。解答履歴のレコードは学生が穴を埋めた順で生成されている。そのため、解答過程を分析したい学習履歴に対応する解答履歴のレコードを取得し、レコードの順序に表示していくことで解答した順序となる。

ラジオボタンで選択した解答履歴の時点での状態を再現する処理は jQuery を用いて次のような方法を取っている。

- 1 選択されているラジオボタンの index を取得する。
- 2 取得した index までのラジオボタンについて以下の処理を繰り返す。

#### 2.1 ラジオボタンの行の「学生の解答」を問題内の

該当する穴抜きのテキストボックスに入れる。

これにより学生が同じ穴抜きに何度か解答していても、後から入力した方に上書きされ、図 11 のように選択した時点での穴抜きの状態を再現することができる。

The screenshot shows a table with columns '解答' (Answer), '正答' (Correct Answer), and '所要時間' (Time Taken). The table lists 6 rows of student data. To the right of the table is a code editor window titled 'プログラム' (Program) showing C++ code for a program that takes two integers and outputs their sum and average. The code is as follows:

```

#include<iostream>
using namespace std;
int main(){
    int a, b, tmp;
    cin >> [a];
    cin >> [b];
    while(a > 0){
        tmp = [b%a];
        b = a;
        a = tmp;
    }
    cout << b << endl;
    return 0;
}
    
```

Red arrows point from the table rows to the corresponding code lines in the editor, illustrating how the state of the solution is reproduced.

図 11 穴抜きの状態の再現

## 8. 試用者からのレビューコメント

実装した各機能を、数人に試用してもらったところいく

表 4 コメントと対応策の一覧

	コメント	対応策	対象となる機能
修正済み	受験結果では学習開始時刻と終了時刻ではなく学習時間と所要時間を表示してほしい。	学習終了時刻から開始時刻を引いた値を所要時間として表示する。	学生の利用状況表示 学生毎の受験履歴 問題毎の学習履歴全体の分析 問題毎の受験履歴 問題分析機能
	受験結果がユーザ ID 順になっているため予め名前ですべてソートしてほしい。	データを取得する SQL 文で名前をソートするようにする。	学生の利用状況表示 学生毎の受験履歴 問題毎の学習履歴全体の分析 問題毎の受験履歴 問題分析機能
	プログラムやトレース表が縦長になると分析内容と問題を見比べにくい	プログラムやトレース表にスクロールバーをつける。	問題分析機能 学習履歴の表示 解答過程の分析
	解答過程の分析機能で戻る/進むボタンを追加し、1 ステップずつ確認できるようにしてほしい。	戻る/進むボタンを追加し、選択されているラジオボタンの前/次のラジオボタンが選択された時の処理を行う。	解答過程の分析
	解答過程の分析機能で学生がどこを書き換えた時点の状態なのかわかりにくい。	学生が書き換えた箇所を該当する穴抜きの色を黄色で表示する。	解答過程の分析
未修正	表示している分析内容を CSV 形式 (または Excel 形式) でダウンロードできるようにしてほしい。	CSV 形式か Excel ファイルをダウンロードできるようにする。	全ての機能
	度数分布の棒グラフを表示してほしい。	点数と人数の度数分布を棒グラフで表示するようにする。	問題毎の受験者一覧

つかのコメントを得ることができた。コメントの中には既に修正を加えているものも含む。コメントと対応策、対象となる機能の一覧を表 4 に示す。

CSV 形式のダウンロードは、Excel や Access を用いた分析などに使用する予定である。PHPEXCEL[7]を用いることで、Excel 形式でのファイル生成も可能と考えている。また、グラフ表示は JpGraph [8]を使用して実装する予定である。

## 9. おわりに

本稿では、穴埋め問題を用いたプログラミング教育支援ツール pgtracer で収集するログデータを分析する分析機能の企画と実装について述べた。pgtracer では学生毎の分析、問題毎の分析、解答過程や学習履歴の分析の 3 つの観点から分析する 7 つの機能を提供する。教員はこれらの機能を用いて pgtracer で収集したログデータを分析し、学生の理解度や不得意箇所を把握、問題の難易度の評価を行える。

現在、pgtracer で作成できる各種の穴埋め問題を対象として、問題の難易度に影響を与える要因の分析を進めている[5]。本機能を用いたログデータの分析は、その際にも有用な情報を提供している。

今後の課題としては、様々な穴埋め問題を活用した学生

の理解度評価手法を構築するための系統的なプロセスの確立が挙げられる。また、学生の不得意項目等に基づいた適切な難易度の穴埋め問題を出題できるよう、穴埋め問題に対する定量的な難易度評価を進める予定である。

## 参考文献

- [1] 柳田, 太田, 大月, 掛下, 穴埋め問題を用いたプログラミング教育支援ツール pgtracer の運用実験, 情報教育シンポジウム 2014 論文集.
- [2] Tetsuro Kakeshita, Ryo Yanagita, Kosuke Ohta, Mika Ohtsuki, A Programming Education Support Tool pgtracer utilizing Fill-in-the-Blank Questions : Overview and Student Functions, ERMM2015 (to appear)
- [3] Tetsuro Kakeshita, Kosuke Ohta, Ryo Yanagita, Mika Ohtsuki, A Programming Education Support Tool pgtracer utilizing Fill-in-the-Blank Questions : Teacher Functions, ERMM2015 (to appear)
- [4] Moodle.org, <https://moodle.org/>
- [5] 村田, 柳田, 太田, 大月, 掛下, 穴埋め問題を用いたプログラミング教育支援ツール pgtracer の問題難易度に関する考察, 情報処理学会 第 129 回コンピュータと教育研究会, 2015.
- [6] jQuery plugin: Tablesorter 2.0, <http://tablesorter.com/docs/>
- [7] PHPEXCEL, <https://phpexcel.codeplex.com/>
- [8] アシアル, JpGraph, <http://www.asial.co.jp/jpgraph/>