Vol.2014-CE-124 No.5
2014/3/14

穴埋め問題を用いたプログラミング教育支援ツール pgtracer の概要と学生用機能の実装

太田康介[†1]　柳田峻[†1]　大月美佳[†1]　掛下哲郎[†1]

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 の全体構成と学生用機能を示す．pgtracer は Moodle 上で動作し，プログラムやトレース表に対する穴埋め問題を学生に出題する．同じプログラムでも穴の位置を変更することで様々な難易度の問題を出題できる．学生が穴を埋めるとシステムは自動的に採点し結果を表示するが，結果表示のタイミングは穴を埋めた直後か，全ての穴を埋めて解答を終了した時点のいずれでも，教員が任意に指定できる．学生が入力した答案，時刻，正誤などはログデータとして収集され，教員はログデータを分析することで学生の理解度を把握できる．

A Programming Education Support Tool pgtracer utilizing Fill-in-the-Blank Questions : Overview and Student Functions

KOSUKE OHTA[†1]　RYO YANAGITA[†1]
MIKA OHTSUKI[†1]　TETSURO KAKESHITA[†1]

This paper presents overview and student functions of a programming education support tool pgtracer utilizing fill-in-the-blank questions. This tool runs under Moodle and provides fill-in-the-blank questions of a given computer program and a trace table to the students. The tool can provide questions having various difficulty levels from the same program. This can be realized by changing the position of the blanks of the program and trace table. When a student fills the blanks, the tool automatically evaluates the answer. The tool automatically collects answers, time and evaluation result as log data. A teacher can analyze understanding level of the students by analyzing the log data.

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]．本ツールは大学等で広く普及している e-Learning システム Moodle[2]のプラグインモジュールとして動作し，プログラムやトレース表に対する穴埋め問題を出題する．

pgtracer が出題する問題は，プログラム，トレース表，プログラム用マスク，トレース表マスクを表現する 4 つの XML ファイルで構成されており，教員は Moodle のページ上で簡単に XML ファイルを作成することができる．同じプログラムでもトレース表やプログラム用，トレース表用マスクを変更することで様々な難易度に設定することができ，学生は自分の学力に合った問題を選択し，解答する．学生が入力した答案，時刻，正誤などはログデータとして収集され，教員はログデータを分析することで個々の学生や全体の理解度や不得意箇所を特定し，教育改善に役立てることができる．また，ログデータから学生が解答した順序や，それぞれの穴を解くのに要した時間を求めることもできる．

本ツールには，教員用の機能として 4 種類の XML ファイルの自動生成・編集機能，問題の定義機能，学習履歴の管理機能が，学生用の機能として問題の提示機能，ログデータの収集機能，自動採点機能，採点結果の表示機能が用意されている．

本稿では，本ツールの全体構成と学生用機能について示す．

2. 穴埋めを用いたプログラミング問題

本ツールで使用するプログラミング問題は，プログラムとトレース表の組に対して，いくつかの箇所にマスクをして学生に提示し，穴埋めを行わせる形式で出題される．図 1 は Euclid の互除法のプログラムとトレース表の一部をマスクしたものである．

ステップ	プログラム
	#include<iostream>
	using namespace std;
	void main(){
	int a,b,c;
1	cin >> a; cin >> b;
2	while(a>0){
2.1	c = b [ ] a;
2.2	b = a;
2.3	[ ]
	}
3	cout << b;
	}

ステップ	a	b	c	出力
1	18	27		
2	18	27		
2.1	18	27	9	
2.2	18	[ ]	9	
2.3	9	[ ]	9	
2	9	[ ]	9	
2.1	9	[ ]	0	
2.2	9	9	[ ]	
2.3	0	9	[ ]	
2	0	9	[ ]	
3	0	9	[ ]	[ ]

図 1　穴埋め問題の例

†1 佐賀大学
Saga University

プログラミング能力を向上させるには，学生本人の学習意欲を高く維持する必要がある．学生の理解度に相応しくない難易度であれば，継続性や学習効果は低下する．学生の継続性や学習効果を高めるには，本人の理解状況に応じた難易度の問題を出題する必要がある．同じプログラムとトレース表の組でも，様々なパターンのマスクを定義することで難易度の調整が可能なほか，プログラムの入力データを変えることでトレース表が変化するため，より多くの問題を作成できる．

本ツールで使用する問題は，プログラムとトレース表に対する穴埋め問題だが，直接的にプログラムとトレース表のテキストに穴を空けるのではなく，プログラムとトレース表本体の他に，穴埋め箇所を定義したプログラム用マスクとトレース表用マスクを作成し，これらを組み合わせて問題を構成するアプローチを採用する．プログラムやトレース表本体と穴抜きのマスク情報を分離することで，プログラムやトレース表の再利用，マスクの編集を容易に行うことができる．

マスクする箇所の例として，以下のような候補が考えられる．

- トレース表に示された変数値をマスクすることで，変数値の変化を正しく理解していることを確認する．
- トレース表のステップ番号や変数名をマスクすることで，文の実行順序や対応する変数を正しく理解していることを確認する．
- プログラムの変数名，演算子，予約語等をマスクすることで，初歩のプログラム作成能力を確認する．
- 式，文，複合文，ルーチン等をマスクすることで，より高度なプログラム作成能力を確認できる．

文単体を理解していれば解けるような易しい問題から，制御構造を正しく理解し実行順序が記述できるかを確認する問題，予約語やルーチン呼び出し等の知識が要求される問題，プログラムの記述能力やプログラム全体の理解が必要な問題まで様々な難易度の問題を作成できる．このようにマスクのパターンを変えた様々な問題を解かせることで，学生の理解度や不得意な部分を把握し，適切な難易度の問題を出題できる．また，穴埋め部分をマスクするだけでなく，出題の際にプログラムの一部やトレース表の一部を非表示にすることや，トレース表の列の順番を入れ替えることもできる．

3. 関連研究

本ツールと同じようにプログラミング教育を行う上で，穴埋め問題を出題するといった研究はいくつか行われている．しかし，その多くはプログラムに対し穴抜きをおこなうものであり，本ツールのようにトレース表に対しても穴抜きを行うシステムは知られていない．

3.1 Moodle を基盤としたプログラミング教育のための穴埋め問題生成に関する検討

Moodle の小テストモジュールには穴埋め問題を出題する機能があり，特殊なタグを用いて問題文を記述することで穴埋め問題を作成できる．また，穴埋めの正解として複数の答えを設定することもできる．たとえば，if(a > 0)の中の a > 0 を穴埋めにする場合 a > 0 だけでなく 0 < a も正解となる．この場合特殊なタグを用いた記述を行うと，以下のようになる．

```
if( {1:SHORTANSWER:=a > 0~=0 < a} )
```

新開らは，小テストモジュールを用いた場合，特殊なタグでの記述を行う必要があることと，複数の解答を全て書く必要があることが教員の負担になると考え，プログラム穴埋め問題作成モジュールの開発を行っている[3][4]. プログラム穴埋め問題作成モジュールでは問題となるプログラムのテキストファイルをアップロードし，穴埋め問題作成用のテキストエディタ内に表示する．その後テキスト内で穴抜き箇所にしたい部分を選択し赤文字に変更して保存する．この際テキストエディタ内の文字列の赤字部分を穴埋め用の特殊タグにし，解答が自動挿入されるようになっている．

3.2 Java プログラミングの予約語学習のためのオンライン穴埋め問題作成機能の実装

Java 言語の予約語学習支援を目的とし，伊永らは Java プログラミングの予約語学習のためのオンライン穴埋め問題機能の実装を行っている[5]. 伊永らは，学生が提出した Java のプログラムの自動採点を行う，テスト駆動型開発手法による Java プログラミング教育支援ツールの提案を行っていた．しかし，Java 言語の学習がある程度進んだ段階でのみ利用可能で, Java 初学者を支援するのは困難と考え，Java 初学者が覚える必要がある予約語の学習ができるような機能の実装を行っている．

この機能は Java のソースコードの予約語に対し穴埋め問題を自動生成するツールである．教員は問題に使用する Java のソースコードをいくつかデータベースに登録しておく．問題を作成する際は学習させたい予約語や穴抜きにする割合を選択する．すると選択した予約語が使用されているコードが一覧表示されるためその中から出題したいものを選択する．

4. pgtracer の概要

pgtracer は授業の一環, 低学力の学生を対象とした補習，学生の自習等で活用する予定で，主に本学 1~2 年で学習する C++（構造化プログラミング），Java（オブジェクト指向プログラミング），Z80（アセンブリ言語）の学習を行うためのシステムとして企画しているが，現在は C++プログラミング用の支援ツールがほぼ完成している．本ツールは，

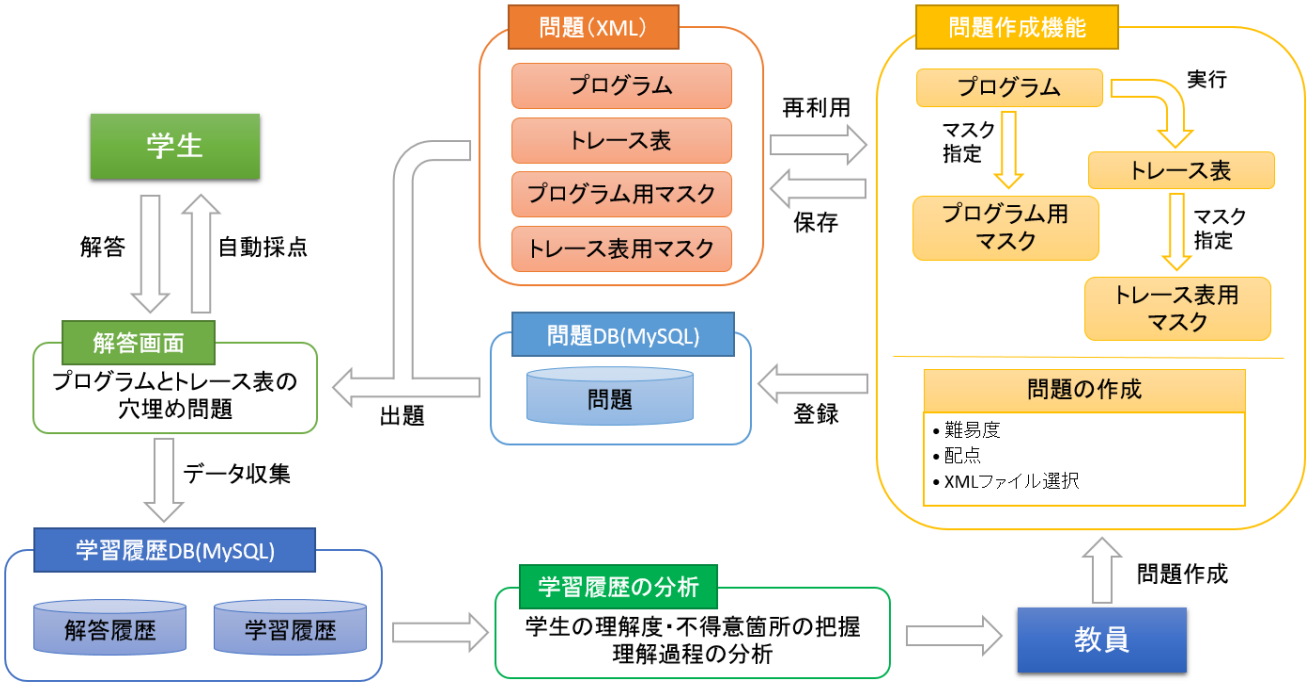

図 2　ツールの概念図

プラグインを用いた機能拡張も容易な Moodle をプラットフォームとし，プラグインモジュールとして開発している．

図 2 は本ツールの概念図である．教員はまず，問題作成機能で出題する問題の作成を行う．本ツールで使用する問題はプログラム，トレース表，プログラム用マスク，トレース表マスクの情報を保持する XML ファイルの組み合わせで構成されるため，ツール上でそれぞれのファイルの作成や，問題の配点や難易度等を設定する．作成された XML ファイルや問題の情報は Moodle のディレクトリ内や MySQL テーブルに保持される．学生には MySQL テーブルの情報をもとに問題が表示される．学生が問題を選択して解答するとツールは自動採点を行い，結果を学生に返す．学生の採点結果は学習履歴として，解答の過程は解答履歴のログデータとして MySQL テーブルに格納される．教員は学習履歴や解答履歴から個々の学生や全体の理解度・不得意箇所を把握し，学生のプログラム理解過程を分析する．その後，分析結果を参考にして新たな問題を作成し，履歴データを収集するといったサイクルを繰り返すことで，学生の不得意箇所に対する重点的な教育や適切な難易度の問題を作成することが可能になる．これにより教育改善や学生の学習意欲の向上が図れると考えている．

これらの機能のうち教員用の機能に関しては著者による別論文[6]で詳しく述べているため，本節では概要のみ述べる．

4.1 XML ファイルの作成・編集機能

本ツールで使用する問題はプログラム，トレース表，プログラム用マスク，トレース表マスクの情報を保持する XML ファイルから構成される．そのため，教員はまず XML ファイルの作成を行う必要がある．XML ファイルの作成は外部の XML エディタ等でも行えるが，手作業で XML ファイルを作成させるのは負担が大きい．そこで本ツールではこれらの XML ファイルを容易に作成できるように XML ファイルの作成機能を実装している．

- プログラム

教員に PC 上からプログラムのソースファイルをアップロードさせ XML ファイルを自動生成する．この際，アップロードしたソースプログラムは自動的にコンパイルされ，コンパイルに成功した場合のみ XML ファイルを出力する．失敗した場合にはエラー文を表示する．

- トレース表

上記の自動生成機能で生成されたプログラムの XML ファイルからトレース表の XML ファイルを自動生成する．トレース表は，プログラムの標準入力やファイル入力によって実行結果が変化するため，入力後の実行結果を一度表示し，教員の確認後にトレース表の XML ファイルを生成する．

- プログラム用マスク

プログラム用マスクでは，プログラムの穴抜き箇所及び非表示箇所を定義する．本ツールでは Moodle のページ上でプログラムを指定し，穴抜きや非表示にしたい箇所を選択させることで，マスク用の XML ファイルの生成・編集機能を提供する．

- トレース表用マスク

トレース表用マスクでは，トレース表の穴抜き箇所及び表示させる行・列の選択を行う．プログラム用マスクの場合と同様に Moodle のページ上でトレース表を表示し，穴

抜きしたい箇所や表示する行および列の選択を行う．

これらの機能を用いて作成された XML ファイルは，ファイルの種類毎に設定されたディレクトリに格納される．

4.2 テーマ・問題の登録・編集機能

pgtracer が出題する問題はテーマ毎に登録する．テーマと作成する際には，テーマの登録・編集機能を用いる．各テーマにはテーマ名，レベル数，自習用/試験用，問題の一覧表示/非表示の設定を行う．テーマ名は何についての学習なのか，誰が作った問題か，などのように自由に設定できる．自習用としてテーマを設定すれば，学生が該当テーマの問題の解答中に穴の採点結果を知ることができる．一方，試験用であれば解答を終了した時点で採点結果が表示される．一覧表示/非表示の設定は，登録された問題を学生が選択できるか否かの設定である．例えば毎週の課題を全て予め非表示で作成しておき，週ごとに設定を表示に変更するといったこともできる．テーマを登録すると，テーマ一覧から問題の登録ができるようになる．

ツールで出題する問題はテーマ，タイトル，難易度，配点，プログラム，トレース表，プログラム用マスク，トレース表マスクのそれぞれの XML ファイルの情報を持つ．教員は問題の登録/編集機能でこれらを設定し 1 つの問題として定義する．難易度に関してはテーマのレベル数で設定した値を上限とし，1 から上限までの間を選択できる．ここで設定した内容は MySQL テーブルに格納される．

4.3 出題および自動採点機能

問題の登録・編集機能で作成された問題は，問題の選択画面（図 3）で MySQL テーブルの情報をもとにテーマ，タイトル，難易度毎に一覧表示されており，学生は自分の学力に合った，もしくは教員に指示された問題を選択する．それぞれの問題は学生が一度も受験したことのない問題ならば「未受験」と表示され，受験したことのある問題ならば「得点/満点」の形式で表示される．これらの文字列は解答画面へのリンクとなっており，解答したい問題のリンクをクリックすれば受験できる．

問題一覧

テーマ	タイトル	難易度			
		1	2	3	4
四則演算	足し算・引き算	未受験	100/100	100/100	90/100
	掛け算・割り算	未受験	100/100	100/100	未登録
奇数・偶数	奇数偶数の判断	未受験	未登録	80/100	未登録
if文	問題が未登録です。				
for文	問題が未登録です。				

図 3　問題一覧画面

解答画面（図 4）では選択された問題に登録された 4 種類の XML ファイルをもとに穴埋め問題を作成・表示する．学生が穴を埋めると自動的に採点し，学生が入力した答案，正誤，時間などがログデータとして保存される．

図 4　解答画面

自習モードの問題の場合は，穴埋めを行った際に正誤を判定し，穴の色で正誤の表現を行う．正解ならば黄緑で表し，不正解ならば赤で表す．不正解の穴にマウスカーソルを合わせると，正解を吹き出しで表示する（図 5）．

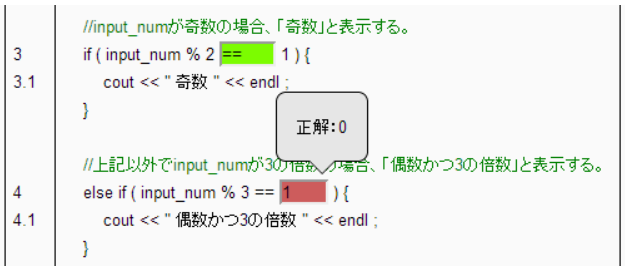

図 5　自動採点及び正しい答えの表示

学生が解答を終え，終了ボタンを押すと採点結果画面（図 6）へ移動する．結果画面では，最終的な解答画面の状態をもとに採点を行い，結果を学生に表示する．結果画面では自習モードの場合と同様に穴埋めした内容が正解であれば黄緑，不正解であれば赤色で表示され，正しい答えが吹き出しで表示される．

図 6　採点結果画面

4.4 学習履歴の分析機能

教員は本ツールを用いて収集した学生の解答のログデータを分析することで，個々の学生やクラス全体の理解度や不得意箇所を把握し，教育の改善に役立てることができる．また，学生の解答過程や問題の難易度を定量的に評価する際にも役立つと期待される．

学生等の理解度や不得意箇所を把握する方法としては，正解率の低い穴や解答所要時間の長い穴を解答履歴のログデータから探す方法が考えられる．また，ログデータを参照することで，学生が教員の指示通りに演習を行ったかを確認することもできる．

プログラムの理解過程については，穴ごとの解答時間と解答順の観点から分析する．分析方法としては，解答履歴に基づいてガントチャートを作成し，それを分析する方法が考えられる．ガントチャートでは横軸を解答時間，縦軸を解答順として作成することで，プログラミング問題の解答過程が可視化できる（図 7）．本末らの研究[5]によると，学生がプログラム全体を見通してから問題を解いている場合には，最初の穴の解答所要時間が長く，後の穴の解答時間が比較的短くなる．それぞれの穴を同じペースで解いている場合は，プログラムを上から順に解いている可能性が高い．オブジェクト指向プログラミングの問題では，ガントチャートにより，クラスの継承関連や集約関連に沿った順序で理解しているかを確認できる．

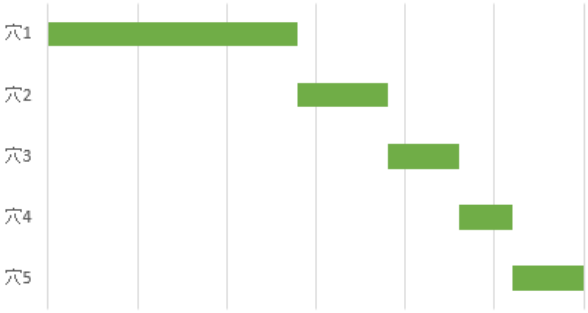

図 7　ガントチャートの例

5. 問題の構成

pgtracer で使用する問題は 4 つの XML ファイルの組み合わせから構成される[6][8]. 本節では穴埋め問題を構成する上で XML ファイルがどのように機能しているか説明する．

プログラムの XML ファイルは問題の基盤となる部分である．穴埋め問題の難易度はプログラム自体の難しさにも依存するが，穴抜きをプログラムやトレース表のどこに行うか，プログラムの入力値つまり実行結果であるトレース表のどの部分を提示するかによっても変わる．つまり 1 つのプログラムに対し，トレース表やプログラム用マスクは複数存在しうる．そこでプログラムの XML では id を属性として定義しておき，トレース表やプログラム用マスクの XML では対象となるプログラムの XML の id を指定することでファイル間の関連を示す．これにより，プログラムの再利用や穴埋め箇所の編集を容易に行えるようにしている．プログラムの XML ファイルではクラスやルーチン，単純な文，複合文，変数の定義部など段階的にノード分けされており，最終的にはトークンにまで分解される．

トレース表の XML ファイルはプログラムの実行結果であるトレース表の情報を持つ．トレース表に関しても穴埋めをどこに行うかで難易度の調整が行えるため id を属性として持っており，トレース表用マスクではトレース表の id を指定する．トレース表の XML ファイルでも同様に表のヘッダーや行，ステップ番号，トレース表の値などにノード分けされる．

プログラム用マスクの XML ファイルはプログラムの穴埋め位置の情報を保持する．穴埋めには question 要素を使用し，穴の位置情報を指定する際には，プログラムの XML に対して定義した XPath 式を利用する．これにより，トークン単体や複数のトークン，文全体などを柔軟に指定することができる．question 要素には穴の重みとして 1 以上の数値を属性として持たせており，採点を行う際には正解した穴の重みの率を集計し，全体の配点に対してかけることで採点を行う．また，非表示を行う hidden 要素も定義されており，プログラム中のコメントなど解答のヒントとなる部分を表示しないといったこともできる．

トレース表用マスクの XML も同様に穴埋めの位置を question 要素で定義する．トレース表用マスクはプログラムとは逆に表示したい行を row 要素で指定する．これによりトレース表の一部を中略して出題することもできる．

問題を作成する際はプログラムとプログラム用マスク，トレース表とトレース表用マスクそれぞれを組み合わせて作成する．プログラムの穴埋めであれば図 8 のように question 要素に記述された XPath 式に対応する部分が穴埋めに，hidden 要素の記述された XPath 式に対応する部分が非表示となる．

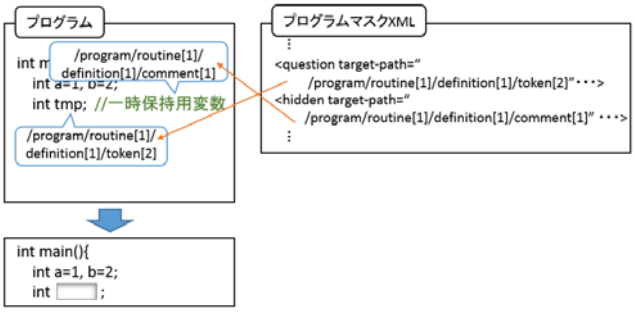

図 8　XML での問題作成処理

6. テーブル設計

Moodle では各種データを MySQL で管理しており，MySQL のレコードの追加，削除，更新，取得を容易に行う関数も用意されている．本ツールでも，問題の情報やロ

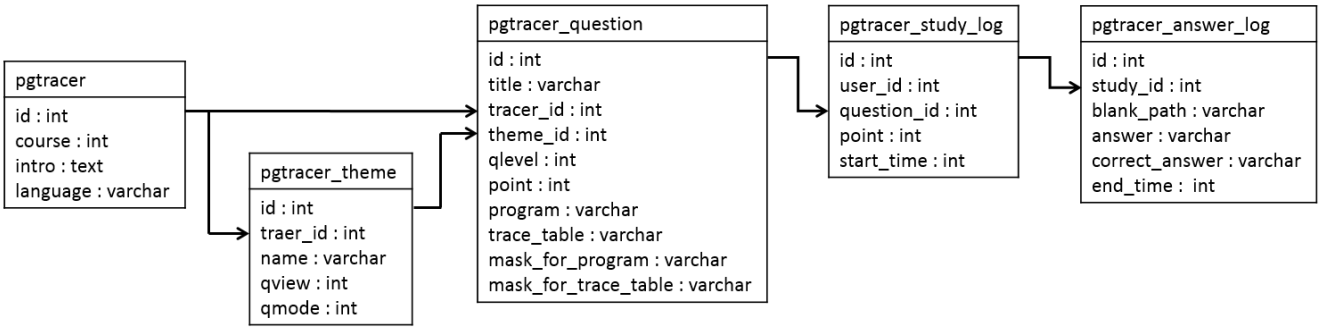

図 9　テーブル間関係図

グデータなどを MySQL のテーブルに保持するようにし，図 9 のテーブル間関係図で示した 5 つのテーブルを使用する．

pgtracer はモジュールの情報を保持しておくテーブルで，どのコースに登録されているかや，どの言語の学習を行うかの情報を保持している．

pgtracer_theme は pgtracer に登録されたテーマのレコードを保持するテーブルである．テーマは，学習内容や誰が作成した問題かのように教員が自由に設定することができ，学生に提示する問題なのかの設定や，試験モードなのか自習モードなのかの設定を行う．

pgtracer_question はツールで使用する問題の情報を保持しているテーブルである．問題のタイトルやどのモジュールのどのテーマに登録されている問題なのかを保持する（表 1）. 問題作成ページで設定された問題の難易度, 配点, プログラム，トレース表，プログラム用マスク，トレース表用マスクのファイル名を保持しておく．問題を表示する際はこのテーブルから該当する問題のレコードを取得し，XML ファイルを読み込み，問題を作成する．

表 1　問題テーブル（pgtracer_question）

主キー	名前	データ型	説明
○	id	int	問題 ID
	title	varchar	問題のタイトル
	tracer_id	int	対応する pgtracer の ID
	theme_id	int	対応する pgtracer_theme の ID
	qlevel	int	難易度
	point	int	配点
	program	varchar	プログラムのファイル名
	trace_table	varchar	トレース表のファイル名
	mask_for_program	varchar	プログラム用マスクのファイル名
	mask_for_trace_table	varchar	トレース表用マスクのファイル名

pgtacer_study_log は学習履歴を保持するテーブル（表 2），pgtracer_ans_log は解答履歴を保持するテーブル（表 3）である．これら 2 つのテーブルは学生の学習のログデータを保持するために使用する．学習履歴を保持するテーブルでは，1 回の学習が 1 レコードに対応し、解答開始の段階でレコードが生成される. 解答開始の段階ではユーザーID や問題 ID 学習開始時間のみレコードが格納され，解答終了した時点で採点結果と学習終了時間が格納される．解答履歴を保持するテーブルは穴を 1 回埋める度に 1 つのレコードが追加される．学生が穴を埋める度に入力された文字列の正誤判定を行い, 穴の位置情報や学生が入力した文字列, 正しい答え，入力した時間等が格納される．

表 2　学習履歴テーブル(pgtracer_study_log)

主キー	名前	データ型	説明
○	id	int	学習履歴 ID
	user_id	int	ユーザーID
	question_id	int	問題 ID
	point	int	採点結果
	start_time	int	学習開始時刻

表 3　解答履歴テーブル(pgtracer_answer_log)

主キー	名前	データ型	説明
○	id	int	解答履歴 ID
	study_id	int	学習履歴 ID
	blank_path	varchar	穴の XPath
	answer	varchar	答案
	correct_answer	varchar	正解の文字列
	end_time	int	解答時刻

7. 学生用機能の実装

本節では学生がツールで使用する機能の実装について述べる．本ツールでは学生用の機能として，問題の選択画面の表示機能，解答画面の表示機能，ログデータの収集機能, 自動採点機能, 結果画面の表示機能が用意されている．

7.1 問題の選択画面の表示

問題一覧画面では，教員が作成した問題がテーマ，タイトル，難易度毎に一覧表示される．MySQL テーブルから問題のレコードを取得したそのままの状態では，問題を登録した順番に取得されてしまう．そのため HTML のテーブルに表示する際にテーマ，タイトル，難易度毎に表示するのが困難になる．そこでテーマ名やタイトルをキーとした多次元配列に一度格納し，その配列を順番に表示することで一覧表示を可能とした．

7.2 解答画面の表示

解答画面では，選択された問題をもとにプログラムとトレース表の穴埋め問題を作成して，学生に出題する．解答開始時に学習履歴のレコードを生成し，ユーザーID，問題の ID，学習開始時間を格納する．問題は MySQL テーブルに XML のファイル名が登録されているため，それをもとにプログラム，トレース表，プログラム用マスク，トレース表マスクの XML ファイルを読み込み，穴埋め問題を作成する．学生に穴埋め問題を出題するためには，プログラムとプログラム用マスク，トレース表とトレース表用マスクのそれぞれの XML ファイルを組み合わせて穴埋め問題を作成する必要がある．以下にプログラムの穴埋め問題を作成するアルゴリズムを示す．

1 プログラムとプログラム用マスクの XML ファイルを読み込む．
2 プログラム用マスクの XML から穴の位置の XPath の情報を全て取得する．
3 穴の位置の情報全てに対し以下の処理を実行する．
  3.1 穴の位置情報に該当するプログラムの XML のノードを取得する．
  3.2 ノードの値を HTML の<input type=”text”…>に置き換える．
4 プログラム用マスクの XML から非表示の位置の XPath の情報を全て取得する．
5 非表示の位置の情報全てに対し以下の処理を実行する．
  5.1 非表示の位置情報に該当するプログラムの XML のノードを取得する．
  5.2 ノードを削除する．

これらの処理を行うとプログラムの XML は教員が作成した穴埋め問題の状態になる．トレース表も同様の処理を行うと穴埋め問題が作成されるため，ノードの値を順番に表示していくことで穴の部分はテキストボックスに変換された図 4 のようなプログラムとトレース表の穴埋め問題が表示される．

7.3 ログデータの収集機能

学生は表示された穴埋め問題に対して解答を行うが，その過程はログデータとして収集されていく．ログデータは MySQL テーブルに格納されるが，データベースへアクセスするためには PHP での処理が必要なため，そのままでは学生が穴を埋めるたびにページをリロードする必要がある．それではあまり見栄えも良くなく，穴を埋めるたびにリロードが発生すると待ち時間が発生し，正しい理解時間の計測が難しくなる．そこで JavaScript の Ajax 通信を用いてページのリロードを行わずに PHP での処理を行うように実装を行った．pgtracer では JavaScript のライブラリの一つである jQuery を様々な個所で使用している．jQuery の中にも Ajax 通信を行える ajax 関数というものが用意されており，今回はその関数を用いて実装している．ログデータの取得は次のような方法で行っている．以下の①～③は図 10 に対応している．

① 解答画面のページで学生が穴を埋めると JavaScirpt が動作し，学生が入力した文字列，穴の位置情報である XPath をログデータ収集用の PHP に転送する．
② ログデータ収集用の PHP では送られてきた XPath に該当するノードを XML から取得し，ノードの値と学生が入力した答えの比較を行う．
③ MySQL のテーブルに穴の XPath，学生が入力した答え，正しい答え，現在の時刻などを格納する．

7.4 自動採点機能

問題が自習モードであれば学生が穴を埋めた時点で採点結果を表示する必要がある．穴の採点自体はログデータを収集する際に行われているため，その結果を解答画面に送ることで実装を行った．自動採点の結果表示は次のように行っている．

④ ログデータ収集用の PHP から解答画面のページに穴の正誤と正しい答えを転送する．
⑤ 問題が自習モードの場合は穴の正誤の表示を行う．

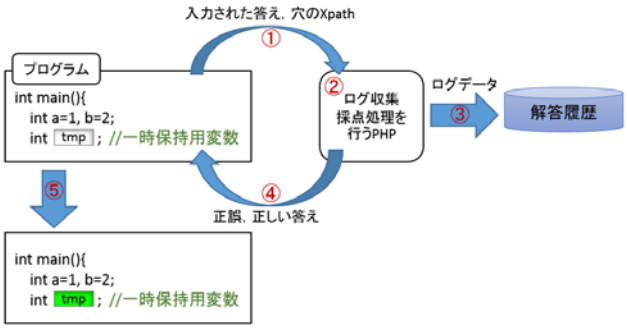

図 10 ログ収集及び自動採点処理

7.5 結果画面の表示

結果画面では最終的な穴埋めの正誤表示と問題の採点結果の表示を行う．それぞれの穴の採点は自動採点機能と同

様に，穴の XPath をもとに XML ファイルからノードの値を取得し，穴に入力された文字列との比較を行う．合計点を出す際は正解した穴の重みの合計を求め，問題全体の重みの合計で割り，問題の配点にかけて求める．例えば問題の配点が 100 点で重みが 1 の穴が 6 個あり，そのうち 5 個が正解だった場合，合計点は 83 点（小数点以下切り捨て）になる．採点後は解答開始時に作成された学習履歴のテーブルの点数を更新する．

8. 学生用機能に関してのレビュー

現在研究メンバー内で試験運用を行っており，学生用機能に関しては次のようなコメントが寄せられた．

- 穴埋めのテキストボックスが見づらい

これは Moodle のページの基本カラーが薄いグレーであり，ブラウザによってはテキストボックスの枠が分かり辛いことが原因である．そのためテキストボックスの枠線の色を黒にすることで対応を行う．

- 穴埋めした内容が正しい答えと完全に一致しなければ正解とならない

現在は学生の答案と正しい答えを比較し正誤判定を行っている．そのため，文全体を穴埋めにしたような場合，学生が記述した答えがプログラム的には間違っていなくとも正解の答えと一致しない場合が考えられる．例えば c=a*b が答えの穴に c=b*a と記述したような場合，学生の答えはプログラム的には間違っていないが正解の答えと一致しないため不正解となってしまう．今後はこのような場合も考慮し，学生が入力したプログラムをコンパイル・実行した上で採点を行う方法を検討する予定である．

- 学生側からは学習履歴が最新のものしかわからない

問題一覧画面では受験結果の点数が解答画面へのリンクとして表示されるが現在は最新の 1 レコードになっている．このままではこれまでの履歴は見ることができない．そこで図 11 のような画面を問題一覧画面と解答画面の間に入れるようにし，こちらの画面から解答を開始するように改善することを検討している．

受験履歴の一覧

1回目	80/100
2回目	75/100
3回目	85/100

解答を開始する

図 11　受験履歴の表示

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 の全体構成と学生用機能について述べた．本ツールは主に本学 1~2 年で学習する C++（構造化プログラミング），Java（オブジェクト指向プログラミング），Z80（アセンブリ言語）の学習が行えるようなツールである．教員が作成した穴埋め問題に対し，学生に穴埋めを行わせることで出題する．学生の解答する過程や採点結果は解答履歴や学習履歴として保存され，教員はその履歴から個々の学生や全体の理解度・不得意箇所の把握，理解過程の分析などを行う．分析結果から問題の新規作成を行い，さらにデータ収集を行うといったサイクルを繰り返すことで不得意箇所の重点的な教育や適切な難易度の問題を作成することができる．これにより教育改善や学生の学習意欲の向上が行えると考える．

今後は，3 月中は 8 節で挙げたような修正を行い，来年度の前期の講義で実際に学生に使用してもらい，ログデータやレビューコメントを収集する計画である．必要があればレビューコメントに沿った修正を行い，来年度後期からは，プログラミングの授業において本格的な運用を行っていく予定である．

参考文献

[1] 掛下，大月，嘉藤，村田，穴埋め問題を用いたプログラミング教育支援ツールの全体構想，平成 25 年度電気関係学会九州支部連合大会 11-2p-01
[2] Moodle.org，https://moodle.org/
[3] 新開，早勢，宮地，Moodle を基盤としたプログラミング教育のための穴埋め問題生成に関する検討, 電子情報通信学会技術研究報告. ET, 教育工学 108(247), 5-10, 2008-10-10
[4] 新開，早勢，宮地，Moodle におけるプログラム穴埋め問題の生成と活用に関する検討，電子情報通信学会技術研究報告. ET, 教育工学 110(263), 7-10, 2010-10-23
[5] 伊永，松島，船曳，中西，天野，Java プログラミングの予約語学習のためのオンライン穴埋め問題機能の実装, 電子情報通信学会技術研究報告. ET, 教育工学 110(453), 125-130, 2011-02-25
[6] 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 における教員用機能の実装，情報処理学会 コンピュータと教育研究会，2014 年 3 月
[7] 本末，掛下，既存プログラムを対象としたソフトウェア設計の理解過程の分析，情報処理学会研究報告. ソフトウェア工学研究会報告 2013-SE-179(8), 1-8, 2013-03-04
[8] 大月，太田，柳田，掛下，XML を用いた穴埋め式プログラミング問題の記述，平成 25 年度電気関係学会九州支部連合大会 11-2p-03