「マルチメディア，分散，協調とモバイル
(DICOMO2013) シンポジウム」 平成25年7月

プログラミング授業において利用可能なMoodleプラグインの試作

市村 哲[1]　川端下 和紀[1]　吉田 匠汰[1]　中村 亮太[1]

概要：プログラミング授業は大学や専門学校等の情報系学科では必須科目となっているが，大学で初めて習う学生の多くはプログラミングに対する苦手意識が強い．一方，近年，Moodle を用いた学習支援システムが注目されている．Moodle はオープンソースによって開発されている学習管理システム（LMS）の１つであるが，Moodle を導入する機関が急増していることが報告されている．今回著者らは，Moodle をプログラミング講義に導入することを想定した場合に，「穴埋め問題を作成する際，Moodle 独自の形式に沿って入力しなければならず扱いが困難である」という問題と，「Moodle の HTML エディタでは，ソースコードを見やすく表示するためのインデントや行番号が表示されない」という問題があることに着目した．そして，Moodle 上でソースコードを適切に表示することができるモジュール，および，穴埋め問題作成エディタを，Moodle プラグインとして開発・実装した．

A proposal of Moodle plugin suitable for programming education

Satoshi Ichimura[1]　Kazki Kawahake[1]　Shota Yoshida[1]　Ryota Nakamura[1]

1. はじめに

プログラミング授業は大学や専門学校等の情報系学科では必須科目となっていることが多い．しかしながら，大学で初めて習う学生の多くはプログラミングに対する苦手意識が強いというのが現状である．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，講師が学生ひとりひとりに対応することが難しいという問題がある．一方近年，インターネットを利用した新しい情報サービスとして WBT(Web-Based Training) などの E-ラーニングが注目されている．いつでもどこでも自分の進捗状況に合わせて学習を進められるなどの利点がある．このような背景から，著者らはプログラミング講義を支援するシステムの研究および開発を以前から継続的に行なってきた．

さらに最近になって，Moodle[1] を用いた学習支援システムが注目されている．Moodle はオープンソースによって開発されている学習管理システム（LMS）であり，PHP，HTML，JavaScript，MySQL 技術を用いて構築されている．Moodle の学習用コンテンツはコースごとに管理されており，コース毎にロール（学生権限，教師権限，TA 権限）の内容を設定可能である．講師も学習者も自分のアカウントで Moodle が提供する Web ページにログインして用いる．

Moodle では，講師は学習用コンテンツ（講義スライド，シラバスなど）を Web 上に掲示する他，小テストやレポート課題を作成して学習者にオンラインで出題したり，学習者の学習状況（学習進捗状況や，小テスト解答の正否，レポート提出の有無など）を閲覧したりすることが可能である．小テスト等の機能を用いることで，学生の理解度を逐次確認しながら授業を進めることができる．小テストやレポート課題には提出締め切り日時を設定することも可能である．一方，学習者は講師が用意したコンテンツを閲覧したり，小テストを受講したりするだけでなく，講師に質問したり，共用 Wiki へ質問や意見を投稿したり，投票したりできる．また，ネット環境さえあれば好きな時間に学習できることも特徴である．

今回著者らは，Moodle をプログラミング講義に導入する際，いくつか問題となる点を発見した．具体的には，穴

[1] 東京工科大学 コンピュータサイエンス学部
Katakura 1401-1, Hachioji, Tokyo 192–0982, Japan

埋め形式の問題を作成する際，Moodle 独自の形式に沿って記述しなければならず，その結果，講師（問題作成者）の手間がかかるという問題がある．また，ソースコードを Moodle 標準の HTML 表示画面に表示する際，不適切にインデントがされてしまい見づらかったり，行番号が挿入できず指摘箇所を示す際にわかりにくいという問題がある．

そこで著者らは，穴埋め問題作成エディタと，Moodle 上でソースコードを適切に表示することができるモジュールとを Moodle プラグインとして開発・実装した．具体的には，Moodle 上でソースコードの穴埋め問題を出題する際，講師が Moodle 独自形式を意識することなく問題作成が容易にできる機能と，ソースコードを見やすく表示するためのインデントや，指摘箇所を示すために必要な行番号を自動的に付与する機能を実装した．本稿では，実装および評価について述べる．

2. 関連研究と本研究の位置づけ

Moodle を用いた LMS が近年注目されている．Moodle はオープンソースによって開発されている学習管理システムであるが，他のシステムとのデータ連携が可能（SCORM 共通化規格に対応），独自プラグインを開発して機能を拡張することが可能，複数サーバーに機能分散・負荷分散することが可能といった特徴がある．Moodle 等の LMS を用いることで，学習コンテンツの再利用性（reusability），アクセス可能性（accessibility），相互互換性（interoperability）が向上すると言われている [3].

近年，Moodle をプログラミング講義で利用する動きが見られる．例えば，筑波大学の情報学群情報科学類における「プログラミング入門 I」では，Moodle の出欠確認機能および課題提出機能を利用していることが報告されている [2]．出欠確認では，特定の文字列を表示専用画面に表示し，それを制限時間内に Moodle に入力させることで，出席の集計を自動化している．ただし，本研究が対象としているような，ソースコードを適切に表示するための機能や，ソースコード穴埋め問題を容易に作成するための機能については言及されていない．

一方，著者らは，2008 年からプログラミング講義を支援するシステムの研究および開発を継続的に行なってきた．支援対象とする受講者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．

著者らは過去において，プログラムの作成過程をなかなか理解することができない学生が多いことに着目し，講師がプログラムの作成過程を動画として自動作成し学生に提示できるソフトウェア [8][9] を構築した．従来のプログラミング講義で使用される講義スライドには完成されたソースコードが提示されることが一般的であるが，プログラミング初学者にとってはソースコードをどのような手順で作成して行けばよいかを読み解くのが難しい．本システムは，講師が普段通りにプログラムを書くだけでプログラムを書く過程を Web 掲載可能な動画として生成することができ，キータイプミス等の冗長場面を容易に削除したり，動画の随所に吹き出しコメントを挿入したりすることが可能となっている．

また，従来の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，エラーの解決方法や，デバッグの方法を学ぶことが難しいという問題がある．そこで著者らは過去において，TA が学生にエラーの解決方法を教えている様子を動画として記録し，学生と TA 間でオンライン共有することができるソフトウェア [10] を構築した．学生が起こしたトラブルを TA が解決している画面の様子を動画収録するクライアントツールと，その動画収録したトラブル解決課程を学生と TA 間で共有できる資料検索サイトから構成されている．

さらに，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，わからないのに手を挙げない学生の発見が難しいことや，講師が想定しないポイントで多くの学生がつまづいている場合にそのことを発見するのが難しいことがわかった．これらの問題に対応するため，著者らは過去において，Web ブラウザ上で動作するプログラミング演習支援システムを構築し，学生の操作ログ・エラーログをサーバー側で収集して解析し，講師や TA にわかりやすく提示できるようにするソフトウェア [11] を構築した．コンパイラーが出力したエラーメッセージや，学生が作成したプログラムの実行結果は，Web サーバーによって取得され，データベースに記録される．エラーメッセージには，エラー個数，エラー名，エラー発生行，エラー発生箇所，エラー原因解説文が含まれる．一方，プログラムの実行結果は，プログラムが標準出力に出力した文字列である．つまずいている学生を早期発見すること，および，多くの学生が共通に抱える問題を発見することを目的としている．

しかしながら，上記過去システムは，講師と学習者が一堂に会する授業スタイルを想定したものであり，講師が宿題を出したり，オンラインで小テストに答えさせるというような機能は提供していなかった．また，標準的な学習管理システム上で動作しておらず，学習コンテンツの再利用性，アクセス可能性，相互互換性に問題があった．

その他，プログラミング講義を支援する研究として，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 [4]，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 [5]，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 [6]，WEB 上でコンパイルが可能なシステム [7] が提案されている．これらのシステムは，講師が宿題を出したり遠隔で協調学習を行ったりするシステムであるが，標準的な学習管理システム上で動作させる

ことを意図しておらず本研究とは方向性が異なる．

3. 本研究の課題

本研究で解決すべき課題は以下の通りである．

課題 1 穴埋め問題を作成する際，Moodle 独自の形式に沿って入力しなければならず扱いが困難である．

課題 2 Moodle の HTML エディタでは，ソースコードを見やすく表示するためのインデントや行番号が表示されない．

上記 2 つの課題が解決されると，本研究の目的が達成されたこととなる．

課題 1 は，Moodle 独自の記述方法を覚える必要があり，かつ，選択肢が多い場合には記述が長く難解になるという問題である．以下は，5 択問題で，正解が 2 つ不正解が 3 つの場合の例である．

記述例 {1:MC: =String args [] #正解です． %0%string [] args #不正解です．string の s は大文字です． %0% string args [] #不正解です．string の s は大文字です． %0% string args #不正解です．args は配列型なので [] が必要です．また string の s は大文字です．%100% String [] args #正解です.})

上記例のように，選択肢・コメント・配点・正否の別を，それぞれの選択肢に記述する必要がある． :%=#*}) などの記号を多用する必要があり，選択肢が多い場合や問題を修正する場合に間違えやすい．

課題 2 は，Moodle 標準の HTML エディタに入力したソースコード中のインデントが授業ページ上で適切に表示されないという問題と，Moodle 標準の HTML エディタには行番号を付与する機能が備わっていないという問題についてである．

Moodle 標準では図 1 のように，タブ文字が消されてインデントが無くなるためソースコードが極めて見辛くなってしまう．

また，実際の講義場面では，講師がソースコードの特定箇所を行番号を示して説明するようなケースが多々存在するが，Moodle 標準ではこのための行番号を付与する機能が備わっていない．このためソースコードを入力する際に，行番号を全て手入力する必要がある．

4. 提案

本研究では，プログラミング授業において利用可能な Moodle プラグインを提案する．提案プラグインは以下の機能を備える．

機能 1 穴埋め問題作成機能

機能 2 インデントおよび行番号付与機能

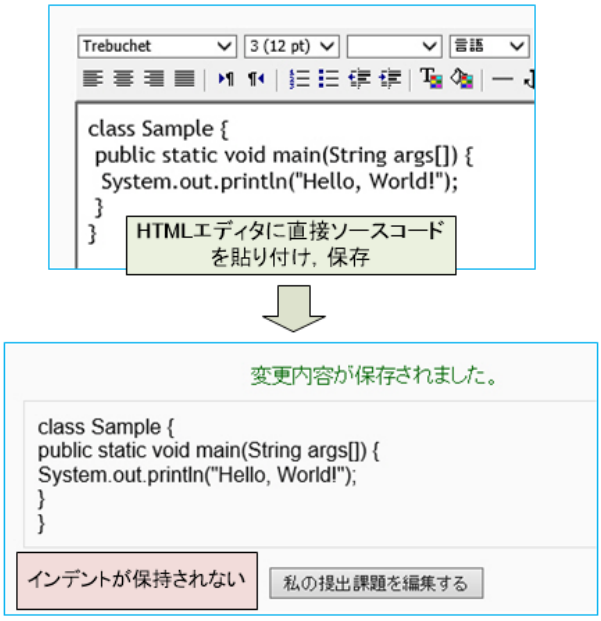

図 1 インデント崩れ
Fig. 1 Collapsed Indent

機能 1 は，Moodle 上でソースコードの穴埋め問題を出題する際，講師が Moodle 独自形式を意識することなく問題作成が容易にできる機能であり，その機能の実現のために，Moodle プラグインとして穴埋め問題作成エディタを実装した．機能 1 は課題 1 に対応している．

機能 1 の動作の概要について図 2 を用いて述べる．まず講師は，予め用意した正常動作するソースコードを穴埋め問題作成エディタに表示し，穴埋め問題にしたいソースコード部分をマウスドラッグで選択する．次に必要に応じて正解時に表示される情報と点数を編集する．そして穴埋め生成ボタンを押すと，マウス選択した部分が Moodle 独自形式に変換されると共に，問題出題時，マウス選択した部分が空欄（入力欄）となって表示される．

機能 2 は，穴埋め問題作成エディタに入力されたソースコード中のタブ文字を所定個数の空白文字に変換することでインデント表示を可能とし，かつ，各ソースコード行の先頭に行番号を自動的に挿入することを可能としている．本機能は，穴埋め問題作成エディタの機能の一部として実装されている．機能 2 は課題 2 に対応している．

5. 実装

開発した Moodle プラグイン（穴埋め問題作成エディタ）の実装について述べる．穴埋め問題作成エディタは，Moodle の記述問題形式と多肢選択問題の両方に対応している．講師が自分で穴埋め問題を作成する典型的なシナリオを想定し，提案機能の実装の概要を述べる．

まず，記述問題形式のシナリオについて述べる．

(1) Moodle の標準 HTML エディタ (図 3) に追加された「エディタ起動ボタン」（図中，最右最上のボタン）を

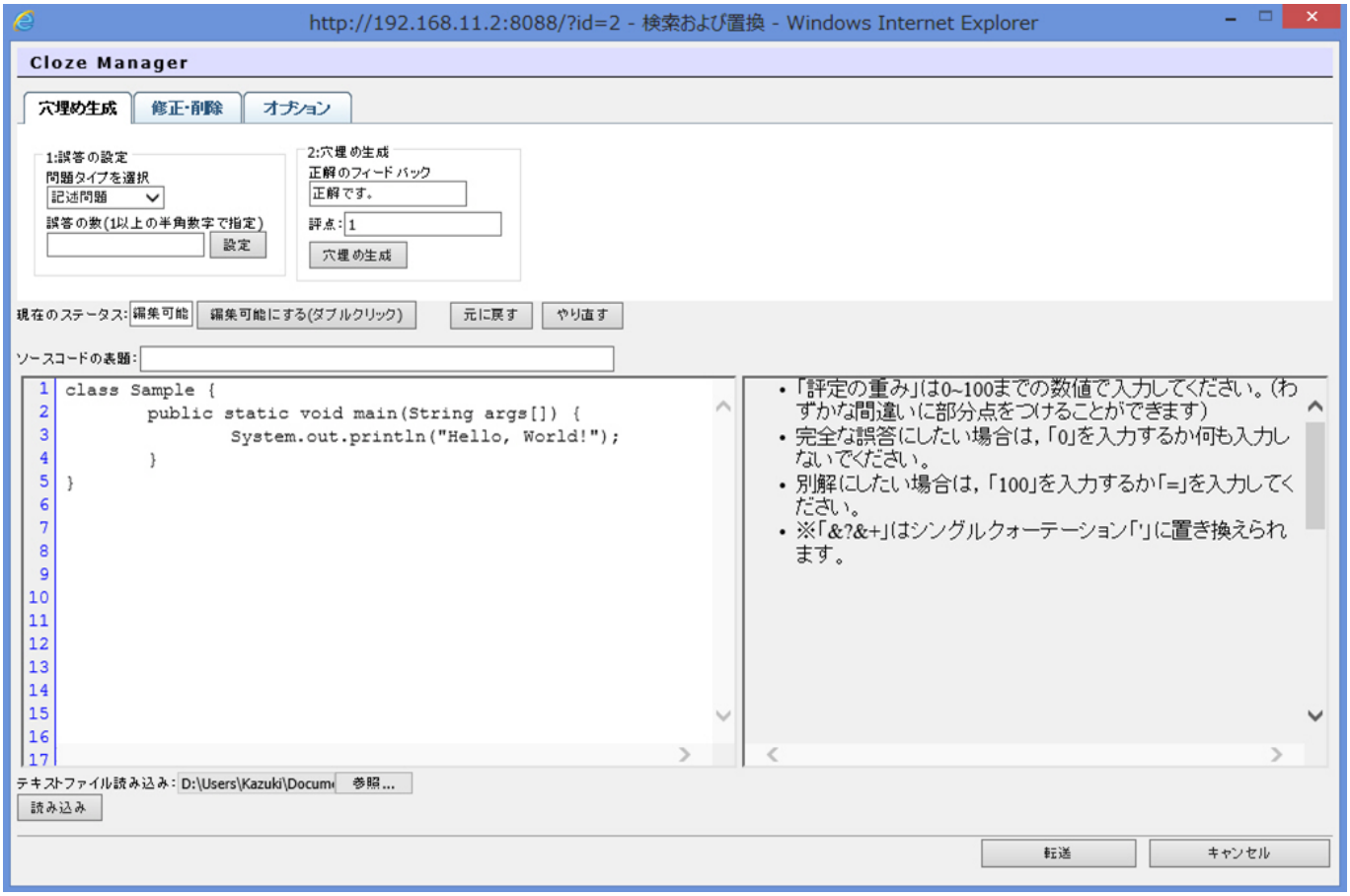

図 4　穴埋め問題作成エディタ
Fig. 4　Fill-in-the-blank creating editor

押すと穴埋め問題作成エディタが起動するので，講師は，予め用意した正常動作するソースコードを，穴埋め問題作成エディタにコピーする (図 4).

(2) 穴埋め問題にしたいソースコード部分（図中，System.out.pintln(“ Hello, world! ”)）をマウスドラッグで選択する (図 5)．問題出題時，ここで選択された部分が空欄（入力欄）となって表示される．マウス選択された文字列の通りに受講者が入力すれば正解と表示され，違う文字列を入力すると不正解と表示されることとなる．

(3) 「穴埋め生成」ボタンを押すと，選択したソースコード部分が Moodle 独自形式に変換され，エディタに表示される (図 6).

(4) 全ての穴埋め問題の作成が完了し「転送」ボタンを押すと，ソースコード中のタブ文字が空白文字列に変換されると共に行頭に行番号が付与され，その結果が標準 HTML エディタに表示される (図 7).

(5) 標準 HTML エディタのプレビュー機能を利用すると，受講者にどのような出題画面が表示されるか確認できる (図 8)．マウス選択した部分が空欄（入力欄）となって表示されている他，インデントが正しく保持され，行番号が付与されていることがわかる．

(6) 受講者が空欄に解答を入力すると，入力した内容や採点結果が Moodle データベースに保存されると共に，例えば正解であった場合には図 9 のようにチェックが表示される（Moodle 標準の動作）.

次に，正解パターンや不正解パターンが複数存在する記述問題の作成について述べる．

正解パターンや不正解パターンを追加したい場合は，穴埋め問題作成エディタ右下のパネルに入力することとなる．図 10 は，正解が 2 パターン（番号 1 と 2），不正解が 3 パターン（番号 3〜5）の場合である．番号 5 のアスタリスクはワイルドカードであり，番号 1〜番号 4 のどれにも当てはまらない場合にマッチし，不正解と表示するようになっている．アスタリスクの他にも，正規表現に類似した記法が使用可能である．

続いて，多肢選択問題形式を作成する際のシナリオについて述べる．まず，図 11 のように問題タイプで「多肢選択問題」を選ぶ．次に，図 12 のように，穴埋め問題作成エディタ右下のパネルに正解選択肢や不正解選択肢を追加する．図 12 は，正解が 1 パターン（番号 1），不正解が 2 パターン（番号 2 と 3）存在する例である．「評定の重さ」の

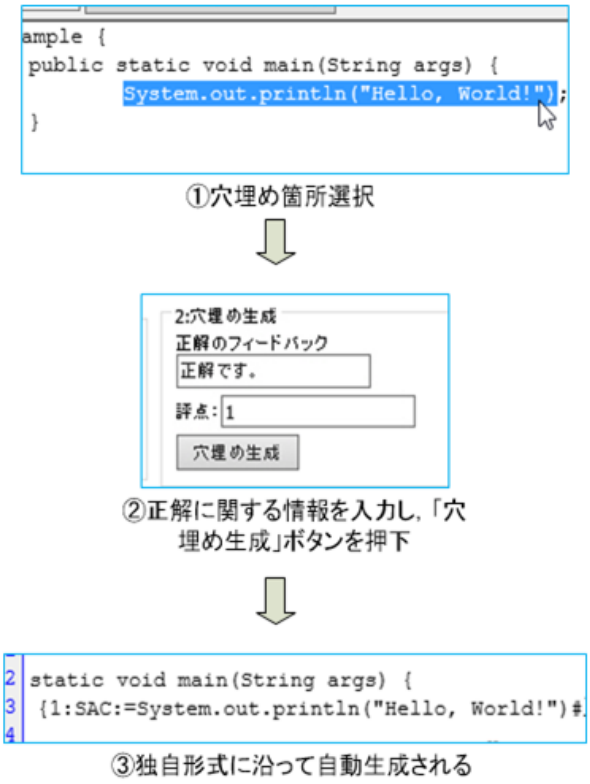

図 2 機能 1 の概要
Fig. 2 Outline of function 1

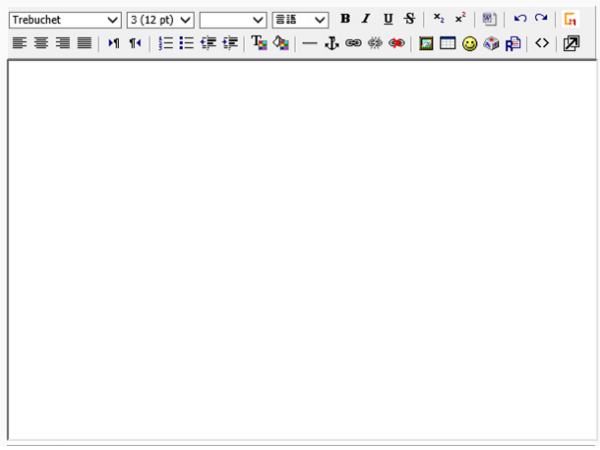

図 3 Moodle の標準 HTML エディタ
Fig. 3 Moodle's standard HTML editor

欄には配点を記述できる．

なお，穴埋め問題作成エディタは，一度定義した問題の削除や編集を行う機能も提供している．

6. 評価

穴埋め問題作成エディタを使用した場合に作業効率が向上するかどうかを検証する．

Moodle 標準の HTML エディタを用いて作業を行った場合 (従来) と本提案の穴埋め問題作成エディタを使用した場合 (提案) とで，ソースコード穴埋め問題の作成にかかる

図 5 穴埋め問題にしたい部分の選択
Fig. 5 Selected part to be Fill-in-the-blank

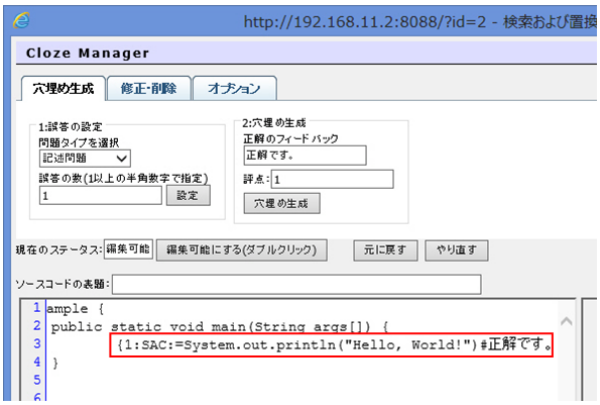

図 6 Moodle 独自形式に変換されたソースコード
Fig. 6 Moodle-script converted from source code

図 7 標準 HTML エディタに表示された問題文
Fig. 7 Questions shown in Moodle's standard HTML editor

作業時間を比較した．機能 1 の穴埋め問題作成機能（課題 1 に対応）と，機能 2 のインデントおよび行番号付与機能（課題 2 に対応）を用いて作業して問題作成時間が短縮したら，本研究の目的が達成されたと言える．

被験者 2 名（大学生）に対し，それぞれ複数回の実験を実施した．実験の公平性を高めるため，実験は，(従来) → (提案) の順と，(提案) → (従来) の順を混ぜて実施した．実験

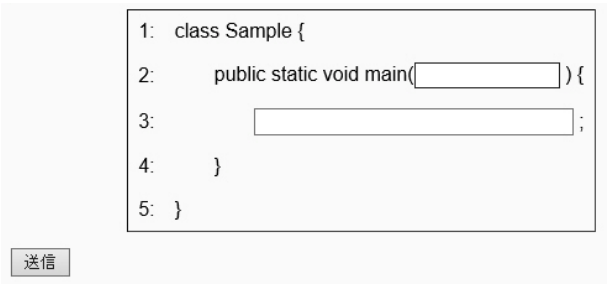

図 8 プレビューモード
Fig. 8 Preview mode

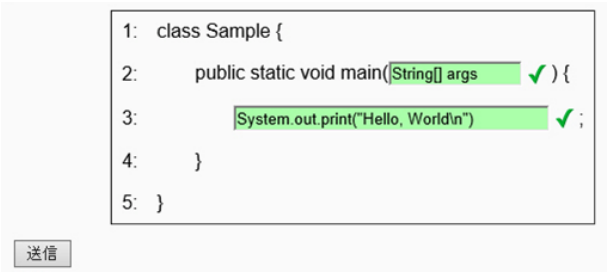

図 9 正解の場合
Fig. 9 In the case of correct answer

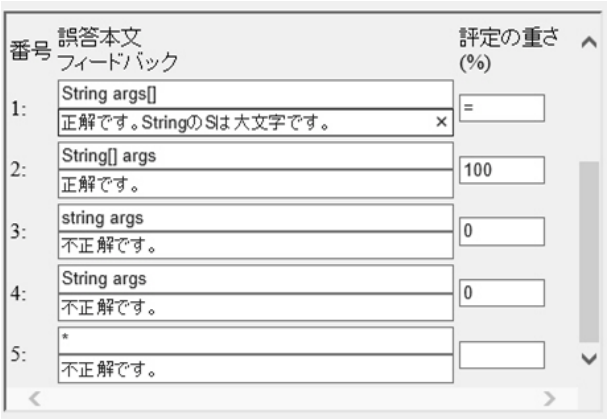

図 10 不正解選択肢の追加
Fig. 10 Added incorrect answers

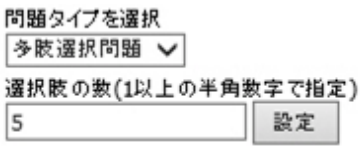

図 11 多肢選択問題の選択
Fig. 11 Selecting choice-type questions

を始める前に，被験者には，Moodle 標準の HTML エディタの使い方と，本提案の穴埋め問題作成エディタの使い方とを数十分間かけて練習させた．そして，実験を開始する際には，正しく動作するソースコードをテキストエディタ上に表示しておき，Moodle 標準の HTML エディタおよび本提案の穴埋め問題作成エディタに即座にコピー＆ペーストできるようにしておいた．

図 12 多肢選択問題の作成
Fig. 12 Creating choice-type questions

被験者には，Moodle 標準の HTML エディタ使用時には，Moodle 独自の記述方法で問題作成すること，行番号を先頭に付与すること，見やすいインデントを空白文字によって作成するように指示した．また，本提案の穴埋め問題作成エディタ使用時には，機能 1 の穴埋め問題作成機能と機能 2 のインデントおよび行番号付与機能を利用するように指示した．

実験結果について述べる．穴埋め問題作成エディタを利用しなかった場合と利用した場合では，利用した場合の方が作業時間が短くなる傾向が確かめられた．

(従来) → (提案) の順で行った際の平均作業時間は以下の通りとなった．

(従来)	(提案)	(提案)/(従来)
平均 7 分 37 秒	平均 5 分 14 秒	平均 68.7 %

一方，(提案) → (従来) の順で行った際の平均作業時間は以下の通りとなった．

(従来)	(提案)	(提案)/(従来)
平均 7 分 19 秒	平均 3 分 34 秒	平均 48.8 %

以上のように，(従来) → (提案) の順で行った実験時には，提案方式は従来方式の 68.7 %の時間で作業が完了する結果となり，また，(提案) → (従来) の順で行った実験時には，提案方式は従来方式の 48.8 %の時間で作業が完了する結果となった．よって，ソースコード穴埋め問題の作成にかかる作業時間が短縮することが示され提案システムの有効性が示唆される結果となった．

また，上記実験後，穴埋め問題作成エディタの使いやすさについて，5 段階（1：使いにくい〜5：使いやすい）でアンケート調査すると共に，理由を被験者にヒアリングした．

アンケートの結果，使いやすさの評価は平均 4.5 となった．また，ヒアリングでは意見が得られた．

- 従来方式では入力に手間がかかる他，プレビューを頻繁に確認しないと正しく入力できているか分からなかった．
- 穴埋め問題作成エディタを利用することで手入力より手間がかからず，自分の入力が間違っているか気にする必要が無くなり良かった．

- 穴埋め問題作成エディタ使用時は，式を直接入力しなくてもよいのが良かった．

7. まとめ

本研究では，「穴埋め問題を作成する際，Moodle 独自の形式に沿って入力しなければならず扱いが困難である」，「Moodle の HTML エディタでは，ソースコードを見やすく表示するためのインデントや行番号が表示されない」という 2 つの課題に対し，穴埋め問題作成機能とインデントおよび行番号付与機能の 2 つの機能を開発した．

評価実験において，穴埋め問題作成機能とインデントおよび行番号付与機能の 2 つの機能によって，ソースコード穴埋め問題の作成にかかる作業時間が短縮することが示され，本提案の有効性が示唆された．

参考文献

[1] William H. Rice IV, 喜多 敏博, 福原 明浩:「Moodle による e ラーニングシステムの構築と運用」，技術評論社 (2009).

[2] 筑波大学「e-Learning Blog」，http://www.els.tsukuba.ac.jp/els/e-learning__blog/2010/10/-imoodle.html. (2010)

[3] SCORM とは: http://satt.jp/dev/scorm.htm (2013).

[4] 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).

[5] 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).

[6] 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)

[7] Truong, N., Bancroft, P., Roe, P.: Learning to Program Through the Web, In Proceedings of the ACM 10th annual SIGCSE conference on Innovation and technology in computer science education, pp.9-13 (2005)

[8] 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).

[9] 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 DICOMO 2011, 6B-2, pp.1035-1042 (2011).

[10] 安田光, 井上亮文, 市村哲：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, 情報処理学会論文誌, Vol.53, No.1, pp. 1-10 (2012).

[11] 市村哲, 梶並知記, 平野洋行：プログラミング演習授業における学習状況の把握と指導支援の試み, 情報処理学会 DICOMO 2012, 5H-3, pp.1410-1416 (2012).