6ZD-7

プログラミング学習支援のための Moodle の拡張

川端下 和紀† 吉田 匠汰† 中村 亮太† 市村 哲†
東京工科大学 コンピュータサイエンス学部‡

1 はじめに

Moodle[1]を用いた学習支援システムが注目されている．著者らは，これ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．まず，穴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点がある．また現状の Moodle には，学習者のプログラミングの技能を培う機能が備わっていないという問題がある．これらの問題を解決するために，穴埋め問題作成エディタ機能およびプログラムコンパイル・実行機能を Moodle に追加した．

2 背景

Moodle は，インターネット上で授業用のページを作成できるという特徴がある．そのページ上で，教師が教材を用意したり，学生が，教師の用意した小テストを受験したりする．

近年，Moodle をプログラミング講義で利用する動きが見られている．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 の課題提出機能等を利用している [2]．

3 問題点

Moodle は講師および学習者に対して，豊富な機能を提供しているが，著者らはこれをプログラミング講義に適用する上で，いくつか問題を発見した．以下に，その問題を列挙する：

1. Moodle の小テストにおける穴埋め問題を作成する際，Moodle 独自の形式（List 1）に沿って入力しなければならない．また，講師がソースコードの特定の箇所を説明する際等に重要となる行番号は，手作業で入力しなければならない．

List 1　Moodle 穴埋めにおける独自形式の例

```
{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#不正解です。}
```

2. ソースコードを既存の HTML エディタに埋め込む際，インデントが正しく保持されない．
3. Moodle の機能のみでは，プログラミング技能（コンパイル，デバッグ技法など）を培うことが難しい．

4 提案

今回の機能拡張は，Moodle 上で穴埋め問題を作成する際の効率性を向上させるためのものである．さらに，著者らが開発した Web ブラウザ上でプログラムをコンパイル・実行可能なシステムと Moodle を，相互に連携させる機能を提案する．

5 実装

5.1 穴埋め問題作成エディタ

穴埋め問題に関する Moodle 独自の形式を意識しなくて済むよう，ＵＩベースでの入力方式を採用した．Moodle に標準で備わっている HTML エディタに対し，新規にボタンを追加し，そのボタンを押してエディタを表示させることで利用できるようにした．講師はあらかじめソースコードを読み込ませる．そして，誤答および解答方式といった情報を入力する（図 1）．その後，ドラッグで穴埋めにしたい箇所を指定し，

Expansion of Moodle for programming learning support
†Kazuki Kawahake, Shota Yoshida, Ryota Nakamura, Satoshi Ichimura
‡School of Computer Science,Tokyo University of Technology

正解時に表示する情報，および点数を入力し，「穴埋め生成」ボタンを押下すると，Moodle 独自の形式に沿った字句に置換される（図 2）．

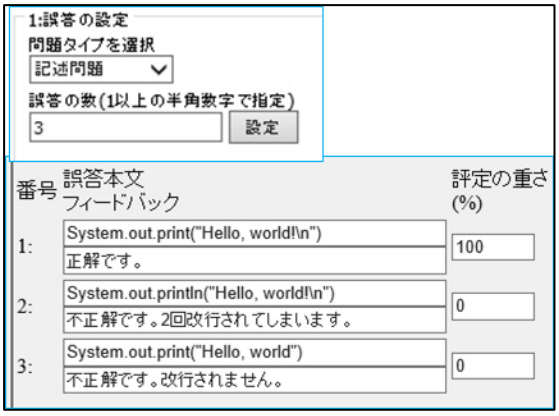

図 1　誤答および解答方式の入力

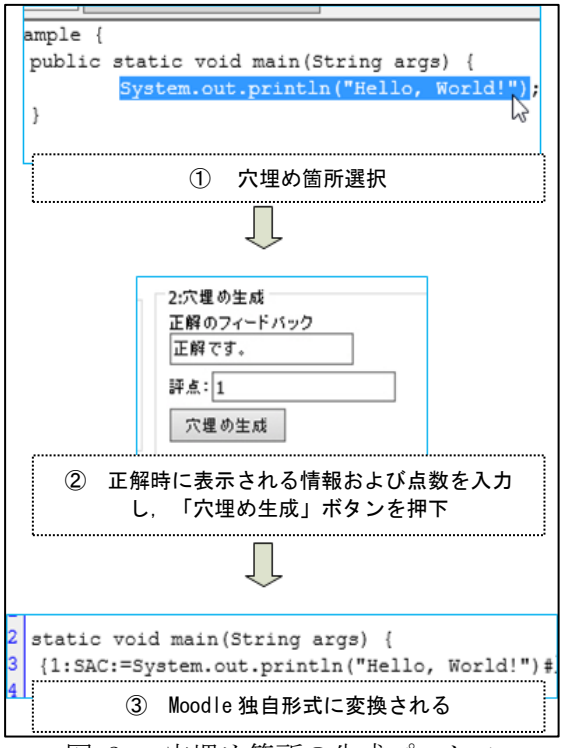

図 2　穴埋め箇所の生成プロセス

最後に，転送ボタンを押下すると，HTML エディタに，行番号が付与された状態でプログラムが転送される．

5.2 既存機能の改善

インデントの問題の原因は，Moodle がタブ文字や改行文字を 1 つの空白文字に置き換えてしまうためである．そこでソースコードを正しく埋め込めるよう，敢えて 1 つの空白文字に置き換えられないよう修正した．

5.3 オンライン上でコンパイル可能なシステムとの連携

著者らの研究室が開発を進めてきた Web ブラウザ上でプログラムをコンパイル・実行できるシステムと Moodle を連携させ，学習者がオンランでコンパイルできるようにした．上記のシステムは，学習者がテキストエリアにプログラムを入力（図 3）し，実行ボタンをクリックすることで，入力したプログラムの実行結果が表示される（図 4）．

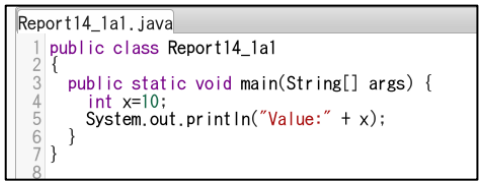

図 3　ソースコードの入力画面

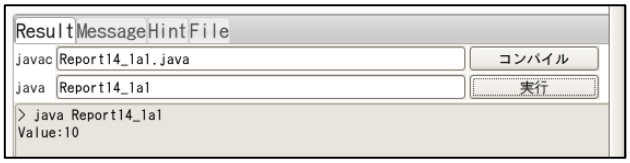

図 4　実行後の画面例

講師は，Moodle を使い課題を入力すると，学習者がプログラムを Web ブラウザ上でコンパイル・実行できるようになる．

6 今後の予定

現状，穴埋め生成に関しては，穴埋めの生成のみをサポートしている．そのため今後は，ソースコード内の穴埋め箇所を検知し，それを UI 上で情報を修正，および削除を行えるようにする．また，過去の穴埋め箇所をデータベースに集積し，その体系に基づいて，穴埋め箇所を自動生成できるようにしたい．

7 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