

Moodle に対応した JavaApplet 作成支援モジュール

松尾大介*, 田中聡, 喜多敏博, 中野裕司 (熊本大学)

Developing a Moodle's Module for supporting Java Applets
Daisuke Matsuo, Satoshi Tanaka, Toshihiro Kita, Hiroshi Nakano

Abstract

We are developing a module for supporting Java Applets on the well-known open source learning management system Moodle. Using the module, Java source code can be edited directory on the web browser and can be compiled and played on the Moodle. It is not necessary for designers to prepare Java developing environments and to manage the source code on their PCs. The module is useful for teachers to develop the interactive contents as scientific simulations, and also useful for students to study Java language directory on the learning management system only with a Web browser.

キーワード: Moodle, モジュール, eラーニング, LMS, Java, Applet
(Moodle, module, e-Learning, LMS, Java, Applet)

1. はじめに

現在,企業での社員教育や大学での授業など,様々なeラーニングの活用事例が広がってきており,時間・場所に束縛されない学習形態が普及しつつある.また,Java アプレットは,インターネット上での学習コンテンツとして,非常に有用である.しかしながら,Java アプレットを作成するためには,コンパイル環境等を整える必要があるため,簡単に作成できるとは言いがたい.

本研究では,オープンソースのLMSであるMoodleに組み込める,Javaアプレットの作成を支援するモジュールの開発を行った.LMS (Learning Management System)とは学習管理システムのことであり,インターネットを利用したeラーニングサイトを構築するためのシステムである.

2. Moodle の概要

<2-1> Moodle の概要 Moodle⁽¹⁾は,Martin Dougiamas氏がPHP言語で開発したeラーニング向けの学習管理システムで,無償で使用できるオープンソースソフトウェアである.GNU General Public Licenseに基づいて配布されており,自由に改造,配布することができる.Moodleの公式サイト⁽¹⁾自身もMoodleを用いて構築されている.日本語言語パッケージの翻訳は,本稿著者の一人である吉田光宏により,2002年11月から現在に至るまで行われている.60以上の言語に対応しており,世界中で利用されているが,最近日本でも急激にユーザ数が増加しており,教育機関をはじめ様々な所で活用されている⁽²⁾.

<2-2> Moodle の機能と特徴 Moodleは,学習管理システムとしての基本的な機能はすべて備えており,商用の学習管理システムと比較しても遜色ない.学生は,教師の作成したコンテンツ閲覧だけでなく,フォーラムに参加して発言したり,小テストを受けたり,課題を提出したりすることができる.コンテンツのページは設定により全ての人が閲覧できるようにすることも,登録した学

生にのみ公開することも可能である.フォーラム等により,Moodle上で学生と議論したり,学生の質問に答えたりすることができ,授業外での教師と学生,あるいは学生間の接触を促進する機能も豊富である.

Moodleはオープンソースであるため,カスタマイズや機能の追加を比較的簡単に行うことが可能で,本研究以外にも様々な組み合わせがある⁽³⁾.また,Moodleサーバの構築に関しても,Vine LinuxやFedora Core用のRPM (Red Hat Package Manager)形式も用意され⁽⁴⁾,インストールが容易になりつつある.

3. JavaApplet 作成支援モジュール

実際に作成したモジュールを用いての,Javaアプレットページの作成方法を示す.

図1に,リソースの追加のプルダウンメニューを示す.作成したモジュールはリソース追加のプルダウンメニューに組み込まれ,ユーザは他の最初から提供されているリソース追加モジュールと全く同じ感覚で使用することが出来る.

「Javaアプレットページの作成」を選択すると,図2のような画面が表示される.従来のリソース作成と同様に,まずユーザは,リソースの名称と概要を入力する.そして,詳細のテキストボックスにJavaアプレットのソースコードを記入した後,各種設定を行い保存する.なお,現在のところは,テキストのフォーマットを「プレイン

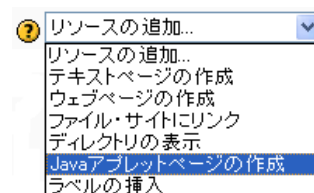


図1 「リソースの追加」の項目として拡張
Fig. 1. Extension as a resource item.

テキストフォーマット」にする必要がある。

リソースを保存し、Java アプレットのコンパイルが適切に行われると、図 3 のような画面が表示される。テキストボックスに入力された Java アプレットのソースコードは、サーバ側でコンパイルされる。そして、作成されたクラスファイルを表示するための HTML ファイル（厳密には HTML を埋め込んでいる）がサーバ側で自動生成され、リソース画面に出力される。サーバ側でコンパイルを行っているため、ユーザ側に Java のコンパイル環境がなくても Java アプレットを使用することができる。また、それを表示するための HTML ファイルを作成したり、それをサーバにアップロードする手間も省ける。さらに、Moodle サーバのバックアップやディスクの2重化等が施されていれば、ユーザが自分のPCにファイルを保管するよりも安全かつ容易にファイルやバージョンの管理が行える。

4. モジュールの追加

リソースを追加するためには、Moodle 本体にファイルの追加またはプログラムの追記をする必要がある。ここでは、その方法について説明する。

〈4-1〉 プルダウンメニューの追加 今回作成したモジュールに関連する部分のディレクトリ階層図を、図 4 に示す。まず、図 1 に示したプルダウンメニューの部分に文字列を追加するため、`/moodle/mod/resource/lib.php` を編集する。今回は、「applet」という名前のリソースタイプを作成した。それに伴い、日本語で表示するために、`/moodle/lang/ja` ディレクトリ中の

`resource.php` を編集し、

```
$string['resourcetypeapplet']  
= 'Java アプレットページの作成';
```

という文を追加する。他の言語に対応するには、その言語のディレクトリの対応部分を編集するだけで可能である。これにより、プルダウンメニューに「Java アプレットページの作成」というメニューが追加される。

ここまでの処理では、ただメニューに表示されるだけなので、次は実際にリソースの追加機能を実装していくことになる。図 4 に示したように、`/moodle/mod/resource/type/` の中に、`applet` というディレクトリを作成する。その中には `applet.html` と `resource.class.php` の二つのファイルが必要となるが、まず、`/moodle/mod/resource/text` ディレクトリの中にある `text.html` と `resource.class.php` を `applet` ディレクトリへコピーし、`text.html` のファイル名を `applet.html` に変更する。そして `resource.class.php` の内容を以下のように書き換える。

```
resource_text → resource_applet  
type/text/text.html → type/applet/applet.html
```

これで、リソースの追加のプルダウンメニューが選択できるようになる。

リソースの追加の機能は `resource.class.php` に記入されているので、現在はテキストページの作成とほぼ同じである。これを、Java アプレットのコンパイルおよび表示ができるように書き換えることになる。

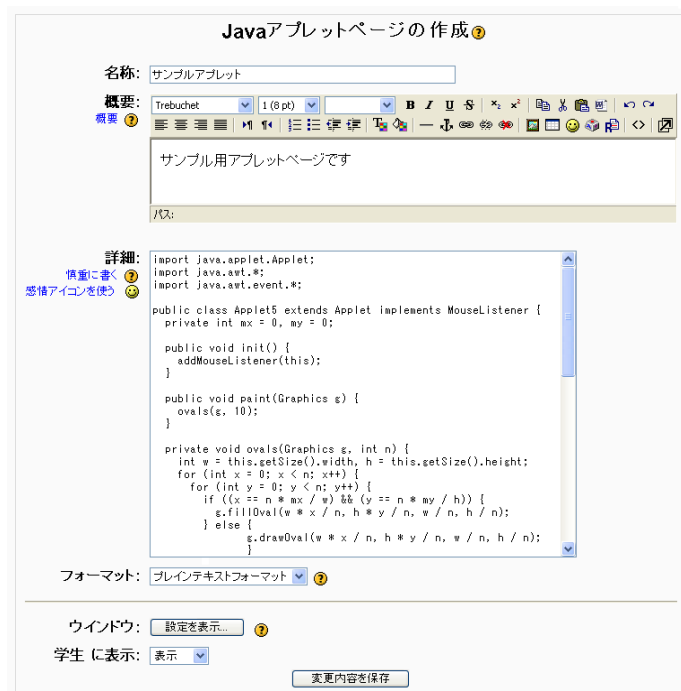


図 2 Java ソースコードの作成及び編集
Fig. 2. Adding and editing Java source code.

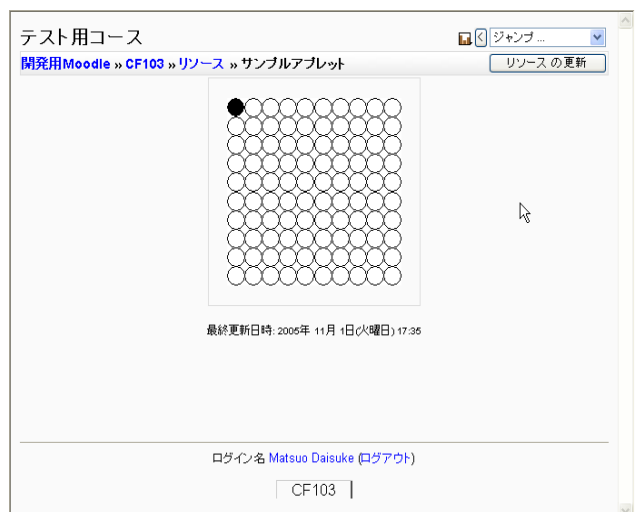


図 3 Moodle 上での Java アプレットの表示(実行)
Fig. 3. Executing Java Applet as a Moodle's content.

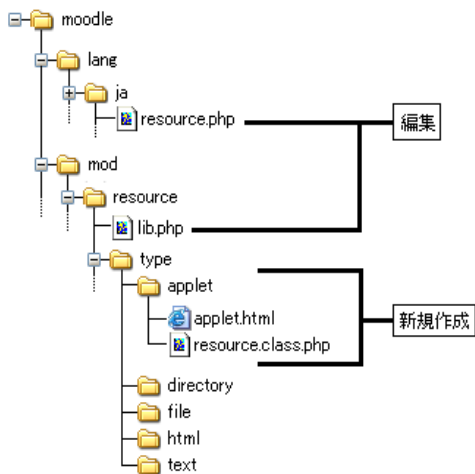


図4 作成したモジュール関連部分の階層図

Fig. 4. Directory structure of the developed module.

<4-2> Java ファイルの作成およびコンパイル テキストボックスに書き込まれたデータは、まず Java ファイルとして保存されなければならない。Moodle では、ユーザからアップロードされたデータ等は専用のディレクトリに格納される。今回は、作成される Java ファイルおよびコンパイルした class ファイルをそのディレクトリに格納することにする。

入力フォームにおいて保存ボタンが押されると、*resource.class.php* はまず Java ファイルの格納場所を決める。データディレクトリの位置は、初期設定時に決定した際に、*\$CFG->dataroot* という関数に格納されている。また、コースやリソースごとに同じ名前のファイルを使用できるよう、データディレクトリの下にコース ID (*\$course->id*) とリソース ID (*\$resource->id*) を名前にしたディレクトリを作成した。その中に *Applet○○.java* (○○はリソース ID) という空のファイルを作成する。その後、テキストボックスに書き込まれた内容を保持した関数 *\$resource->alltext* を Java ファイルに書き込む。

<4-3>コンパイルおよび表示 コンパイルは、サーバ側の *javac* コマンドを PHP から *shell_exec()* コマンドを呼び出すことで行う。ここでは、外部コマンドを実行していることから、セキュリティに最新の注意を払う必要があり、*escapeshellcmd()* 関数による保護等を行っている。

作成された class ファイルは、Java ファイルと同じディレクトリに格納される。表示する。閲覧部分は PHP の内部で生成されるため、HTML ファイルとしては作成されない。class ファイルは実際のディレクトリの位置ではなく、*/moodle/file.php/コース ID/リソース ID/Applet リソース ID.class* という形で参照される。作成した閲覧部分を、Moodle の既存の表示部分 (*print_simple_box*) に引数として渡す。

これらの作業が保存ボタンを押された後に実行され、図3のような形で出力される。

5. 結論

本研究では、Moodle のモジュールとして、Java アプレットの作成支援モジュールを開発した。Moodle 上で、他の既存モジュールと同じ感覚で、アプレットのソースコードを入力、編集だけで Java アプレットの作成及び表示(実行)することができた。

本モジュールを利用することで Moodle サーバ上で殆ど全ての作業が行えるようになるため、ユーザ側にコンパイル環境を構築する必要もなく、ソースファイルの管理も Moodle に任せることができる。さらに、HTML を記述しなくとも、Moodle 上の他の教材コンテンツ同様に表示、実行が可能となった。

6. 今後の課題および展望

現在の問題点および改善点として、以下のものがあげられる。

- テキストフォーマットの指定部分
プレインテキストフォーマットを選択してもらうことを前提としているが、この部分はデフォルトで設定されるようにしたい。
- アプレットのクラス名
現在は Moodle 側で指定しているため、自由度が全くないため、入力フォームの作成あるいはソースファイルから取得する等のプログラムを行い、ユーザ側で任意に設定できるようにしたい。
- 入力フォームの数
必要な class ファイルが複数におよぶ場合、現在の仕様では入力が困難である。class ファイル数を入力するフォームを作成し、それに応じてテキストボックスの数を増やす等の改善を行いたい。
- 毎回のコンパイル
リソースの表示の度にコンパイルを行う仕様なので、マシンに負担がかかってしまう。更新日時によるコンパイル必要性の判定や自動バージョン管理等の対策をとり、最低限のコンパイルですむようにしたい。
- Windows サーバでは動作しない
現在の PHP では、Windows サーバの場合に外部コマンド(今回は *javac*)が動作しないため、この問題の解決を待ちたい。

本モジュールにより,Java アプレットによる教材を LMS である Moodle 上で直接作成登録,利用が可能になることから,インタラクティブなシミュレーション教材等の作成への応用が期待される. また,実際の演習がオンラインで可能な Java 言語学習コンテンツ等への応用も期待される.

今後は,現在わかっている問題点の解決を行った上で,実際にユーザに使用してもらった上で意見を求める等を行い,より使いやすいモジュールとするための改善を続けていきたい.

7 参考文献

- (1) M. Dougiamas and P.C. Taylor, Moodle: Using Learning Communities to Create an Open Source Course Management System, Proc. EDMEDIA 2003 (2003).
Moodle.org: <http://moodle.org/>
- (2) 村嶋, 吉田, 喜多 他, Moodle の市民塾における活用 -くまもとインターネット市民塾-, コンピュータ&エデュケーション, No.19, (2005). (掲載決定済)
- (3) M. Campione, K. Walrath, A. Huml and Tutorial Team, The Java Tutorial Continued: The Rest of the JDK, Addison-Wesley, 1998.
Sun Java: <http://java.sun.com>
- (4) Moodle の RPM パッケージ:
<http://t-kita.net/rpm/moodle/>