

# プログラミング学習を支援する Moodle プラグインと UI の提案と実装

村山 舜<sup>†</sup> 早川 智一<sup>†</sup> 疋田 輝雄<sup>†</sup>  
 明治大学理工学部情報科学科<sup>†</sup>

## 1. はじめに

近年、学習者に e ラーニング環境を提供する学習管理システム (LMS : Learning Management System) が普及してきている。LMS のオープンソースの実装としては、Moodle[5]や Sakai[6]などが有名である。

大学での LMS の利用は、教員と学生の人数の差を補う点で有用であるが、プログラミング教育で LMS を利用する際には、解決すべき課題が残っていると我々は考える。それは、ソースコードの潜在的なバグや可読性の低さなどを検出して学習者にフィードバックする仕組みの提供である。なぜならば、学生の提出するソースコードには、動作はするもののプログラミングの作法としては望ましくない記述 (例：インデントがない) が散見されるためである。

我々は、前述の課題を解決するために、学習者が LMS に提出したソースコードを診断し視覚的にフィードバックを行う汎用的な仕組みを提案し、その実装を本稿で報告する。LMS にはプラグインの開発が容易な Moodle を採用し、専用 UI (User Interface) を Moodle プラグインとして実装した。

本稿の構成は次のとおりである。2 節では先行研究を紹介する。3 節では提案手法を概説する。4 節と 5 節では設計と実装を述べる。6 節では評価結果を報告する。7 節では今後の課題を述べる。

## 2. 先行研究

伊藤ら[7]は、Moodle プラグインと Web サービスを組み合わせた汎用的な課題チェックの仕組みを提案している。これは、授業特有の機能を Web サービスとして用意し、Moodle からその Web サービスを利用するものである。我々の提案は、プログラミング学習を視覚的に支援するための汎用的なフィードバック機能であり、汎用化の対象と粒度が異なる。

## 3. 提案手法

図 1 に、提案手法の概要を示す。処理の流れは次のとおりである：Moodle プラグインは、(1) 学習者からソースコードを受け取り、(2) Web サービスを利用してソースコードを診断し、(3) 診断結果を専用 UI に表示する。図 2 に、提案手法のプロトタイプのスクリンショットを示す。図中では、Java のソースコードをコンパイラや静的チェッカで検査した診断結果を表示している。図より、ソースコード上に診断結果を色付けして表示することで、どこにどのような問題があるかを学習者が視覚的に把握できることが分かる。

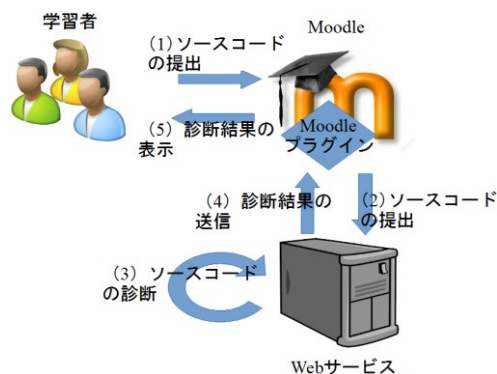


図 1 提案手法の概要

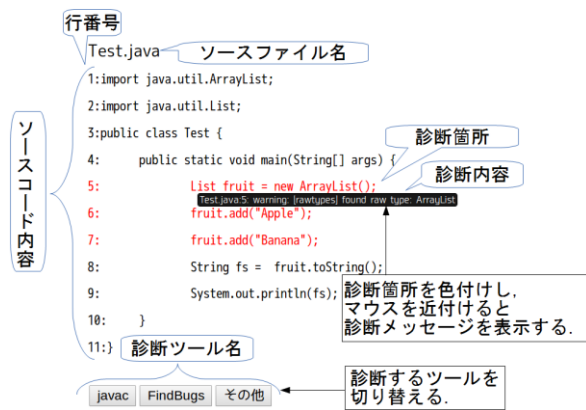


図 2 提出手法のプロトタイプ画面

Proposal and Implementation of Moodle Plugin and UI for Programming Learning  
<sup>†</sup>Syun Murayama, Tomokazu Hayakawa and Teruo Hikita, School of Science and Technology, Meiji University

## 4. 設計

我々は、汎用性を高めるために、提案手法を次のように設計した：(1) Web サービスを用いてソースコードの診断機能を Moodle プラグインから切り離し、(2) Web サービスと Moodle プラグイン間のメッセージ形式を JSON[4]を用いて定式化し、(3) Moodle プラグインは Web サービスから受信した JSON メッセージを専用 UI に表示する。これにより、Moodle の仕様が変更されたりソースコードの診断機能を追加・変更・削除したりする場合でも、対応が容易になることが期待できる。図 3 に、提案手法の概要シーケンス図を示す。図より、Moodle プラグインは Moodle 固有の処理と UI の表示しか行っておらず、ソースコードの診断を Web サービスを用いて汎用的に実現していることが分かる。

### 4.1. Web サービス設計

Web サービスは、コンパイラや静的チェッカを抽象化した REST エンドポイントとして設計した。SOAP よりも REST を選択したのは、軽量性や JSON との相性を考慮したためである。

### 4.2. メッセージ設計

Web サービスと Moodle プラグイン間の JSON メッセージは、次の四つ組の情報を含む：(診断ツール名, ソースコード, 診断メッセージ, 診断メッセージのソースコード上の位置情報)。

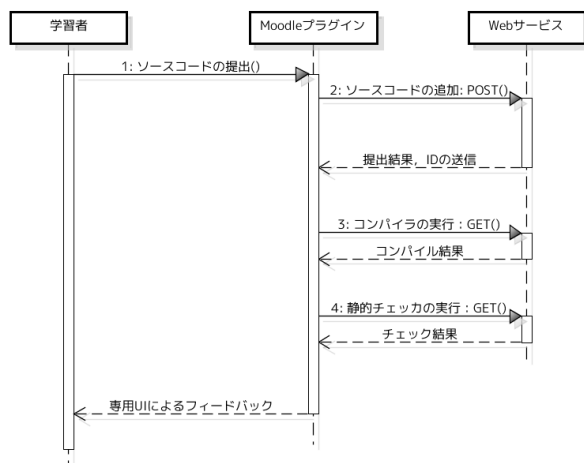


図 3 提案手法の概要シーケンス図

表 1 実装に使用したソフトウェア

| ソフトウェア       | バージョン    | 備考               |
|--------------|----------|------------------|
| Tooltipster  | 2.2.1    | ポップアップの表示        |
| jQuery       | 2.0.3    | クロスブラウザ対応        |
| Moodle       | 2.5.1+   | LMS              |
| Curry        | 1.4.11.1 | REST フレームワーク     |
| PHP          | 5.5.4    | 開発言語             |
| Apache HTTPD | 2.2.22   | Web サーバ          |
| Ubuntu Linux | 13.10    | Operating System |

### 4.3. 専用 UI 設計

専用 UI は図 2 に示したとおりである。専用 UI は、Web サービスからの JSON メッセージに基づいてソースコードに色付けを行い、色付けされた箇所学習者がマウスを近づけると診断メッセージをポップアップ表示する。

## 5. 実装

表 1 に、提案手法の実装に我々が用いたソフトウェアを示す。Moodle プラグインと REST の実装には PHP を使用した。また、REST の実装を容易にするために、フレームワークとして Curry[2]を使用した。専用 UI は、HTML5 と CSS3 と JavaScript で実装した。なお、クロスブラウザ対応を考慮し、JavaScript ライブラリとして jQuery と各種プラグインを使用した。

## 6. 評価

我々は、予備的な評価として、本学の授業で学生が提出した全 9 回分の課題のソースコードを、静的チェッカを用いて診断した。ここで、ソースコードの記述言語は Java であり、静的チェッカには Checkstyle[1]と FindBugs[3]を用いた。結果として、学生が提出したソースコード 1 つあたり平均 14 個の警告が表示され、授業回数が増えてもこの警告数は減らないことが分かった。我々の提案手法を用いれば、これらの警告が課題の提出時に画面に表示されるため、改善後の再提出を学生に促すことができるようになる。

## 7. おわりに

本稿では、LMS 上でプログラミング学習者に視覚的なフィードバックを行う汎用的な仕組みを提案し、その実装について報告した。今後は、提案手法を授業で使用し、学生のソースコードがどう改善されるかを検証するのが課題である。

## 参考文献

- [1] Checkstyle: <http://checkstyle.sourceforge.net/>.
- [2] Curry PHP Framework: <http://www.curryfw.net/>.
- [3] FindBugs: <http://findbugs.sourceforge.net/>.
- [4] JSON: <http://www.json.org/>.
- [5] Moodle: <http://moodle.org/>.
- [6] Sakai Project: <https://sakaiproject.org/>.
- [7] 伊藤恵, 美馬義亮, 大西昭夫: コース管理システムと授業固有の課題チェック機能の Web サービスによる連携, 情報処理学会論文誌, Vol.52, No.12, pp.3121-3134 (2011).