

# プログラミング学習環境の汎用的な構築法の提案と評価

西角 知佳† 早川 智一† 疋田 輝雄†

明治大学理工学部情報科学科†

## 1. はじめに

近年、プログラミング学習者（以下、学習者）のソースコードの品質を向上させる教育法への関心が高まってきている。この方法の1つとして、プログラミング学習を指導する教師（以下、教師）と学習者とが二人一組でプログラミングを行うペアプログラミングがある。これは、ソースコードの改善点の指摘を学習者が即座に受けられる点で有用である。一方で、ペアプログラミングは、学習者と同数の教師を必要とするため常々実施できるとは限らない。

そこで、学習者のソースコードの改善点を教師が不在でも指摘できる学習環境がプログラミング教育の一助になると我々は考え、その実現法を提案してきた[4]。この提案手法は、学習者のソースコードを教師の代わりにコンパイラや静的コード解析ツール（以下、チェッカ）で診断し、その診断結果を Moodle 等の学習管理システム（LMS : Learning Management System）上で学習者に視覚的にフィードバックする専用 UI（User Interface）の構築法に関するものであった。

一方で、従来の提案手法には、教育のレベルや用途に応じた多様な機能や UI の提供が困難であるという課題が残っていた。これは、従来の提案手法が LMS と密結合する形で実現されており、機能や UI の追加・修正に LMS 固有の対応が必要となるためである。

本稿では、前述の課題を解決するために、プログラミング学習環境の汎用的な構築法を提案する。ここで、「汎用的」とは、機能や UI の拡張性や柔軟性を高めることを意味するものとする。本提案手法は、前述の専用 UI を汎用 UI とロジック層とに分割し、LMS との疎結合化を促進することで汎用性の向上を実現する。

本稿の構成は次のとおりである。2 節では関連研究を紹介する。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. 関連研究

井垣ら[3]は、学習者のコーディング過程をオンラインエディタを通じて可視化するシステムを提案している。これは、プログラミング演習における学習者の進捗状況を教師に把握させることで、進捗の遅れている学習者のサポートを支援するものである。井垣らと我々の研究は、プログラミング学習の支援環境をオンラインで提供する点で共通するが、その利用場面や主目的が異なる。井垣らは、プログラミング演習の授業課題に対する支援に焦点を当てており、ソースコードの品質向上を主目的とはしていない。

## 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：(0) 予め LMS に汎用 UI を組み込んでおく；(1) 学習者が LMS にログインすると汎用 UI が表示される；(2) 学習者が汎用 UI 上でプログラミングを行う；(3) 汎用 UI が、プログラミングに必要なコンパイル等の処理をロジック層に依頼する；(4) ロジック層が依頼された処理を実行する；(5) 汎用 UI が、ロジック層から返信された処理結果を画面に反映する。

提案手法の各部位の役割を以下に示す。LMS は、汎用 UI の表示を担当する。ここで、画面描画機能を LMS から汎用 UI に切り離すことで、以前の専用 UI と比較して LMS 固有の処理を低減する。汎用 UI は、学習者がプログラミング学習を行う画面を提供する。汎用 UI は、ロジック層への処理の依頼と処理結果の画面への反映とに専念することで汎用性を実現する。ロジック層は、プログラミング学習に必要な機能を提供する。

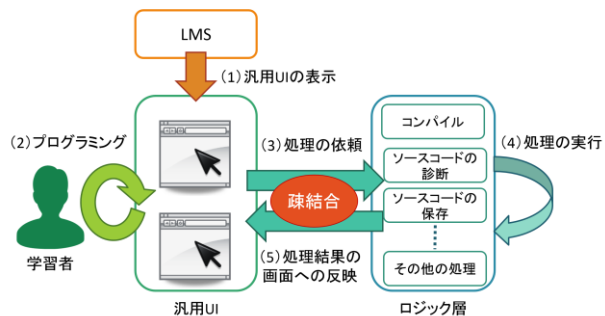


図 1 提案手法の概要

Proposal and evaluation of loosely-coupled design for programming learning environment

† Chika Nishikado, Tomokazu Hayakawa, and Teruo Hikita, School of Science and Technology, Meiji University

図2に、提案手法のプロトタイプ（以下、プロトタイプ）のスクリーンショットを示す。図中では、学習者のソースコードをチェッカで診断した結果を表示している。学習者は、汎用UI上でソースコードの改善点を視覚的に把握できる。ここで、汎用UIの汎用性の例として、チェッカの追加の容易性を挙げる。チェッカの追加は、ロジック層に追加処理を行うだけで自動的にチェッカー一覧に反映される。また、追加したチェッカの診断結果も同様にエディタに反映される。

#### 4. 設計

我々は、我々の提案する学習環境とLMSとの結合を疎にするために、従来LMSと密結合していた処理の大半をLMS外に切り離し、さらにそれをLMSと連携可能な汎用UI（Webアプリケーション）と汎用UIに必要な機能を提供するロジック層（Webサービス）とに分割した。

汎用UIは、再利用可能なUI部品群で構成され、ロジック層が提供する機能を自由に組み合わせることができる。我々は、LMSに容易に組み込めるように、汎用UIをHTML5のSPA（Single-page Application）として設計した。これは、多くのLMSはURLの遷移を自身で管理しているため、URLを遷移させる非SPAのWebアプリケーションを組み込むと正常動作しなくなるためである。

ロジック層は、プログラミング学習に必要な機能を、REST（Representational State Transfer）のエンドポイントとして汎用UIに提供する。新しいエンドポイントをロジック層に追加することで、汎用UIに新機能を提供することができる。エンドポイントは、汎用UIから非同期に呼び出され、処理結果をJSON（JavaScript Object Notation）形式で返信する。

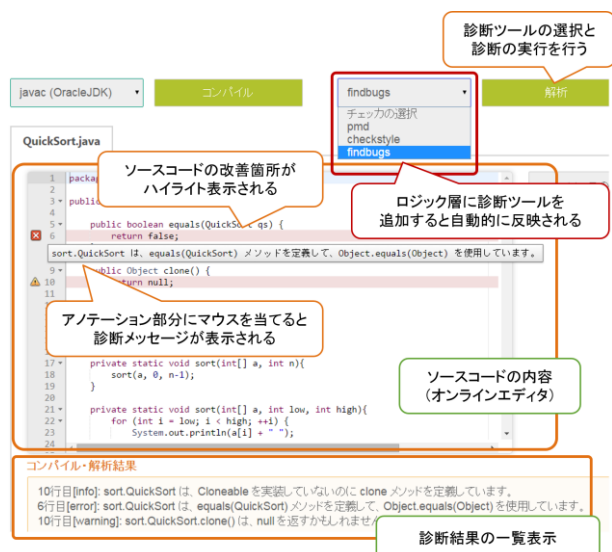


図2 提案手法のプロトタイプ画面

表1 プロトタイプの実装に用いたソフトウェア

ソフトウェア	バージョン	備考
Ace	1.1.8	オンラインエディタ
jQuery UI	1.11.2	jQueryのUIライブラリ
jQuery	2.1.3	JavaScriptライブラリ
Slim	2.4.2	PHPフレームワーク

表2 学習環境への新機能の追加に必要な作業量

	旧手法 (行数)	提案手法 (行数)	削減率 (%)
UI部分	118	42	64.4
ロジック部分	275	194	29.5

#### 5. 実装

表1に、プロトタイプの実装に用いたソフトウェアを示す。汎用UIの実装には、jQueryとjQuery UIを用いた。オンラインエディタにはAce[1]を用いた。ロジック層の実装にはPHPを用いた。また、RESTの実装を容易にするために、PHPフレームワークとしてSlim[2]を用いた。

#### 6. 評価

我々は、提案手法の汎用性を評価するために、チェッカを用いたソースコードの診断機能の学習環境への追加に必要なソースコードの行数を、プロトタイプを用いて以前の提案手法と比較した。表2の結果より、提案手法では、以前の手法と比較してUI部分で約6割強、ロジック部分で約3割弱の行数を削減できることが分かった。UI部分の削減率が顕著な理由は、再利用性の高い汎用UIが、新機能の追加時に必要な作業量を低減するためである。以上より、提案手法が汎用性の向上に有用であるという結論を我々は得た。

#### 7. おわりに

本稿では、プログラミング学習環境の汎用的な構築法を提案した。今後は、プロトタイプを学習者に利用させて、提案手法の学習者に対する有用性を実地検証することが課題である。

#### 参考文献

- [1] Ace: <http://ace.c9.io/>.
- [2] Slim: <http://www.slimframework.com/>.
- [3] 井垣宏, 齊藤俊, 井上亮文, 中村亮太, 楠本真二: プログラミング演習における進捗状況把握のためのコーディング過程可視化システムC3PVの提案, 情報処理学会論文誌, Vol.54, No.1, pp.330-339 (2013).
- [4] 村山舜, 早川智一, 疋田輝雄: 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, 情報処理学会第76回全国大会講演論文集, pp.547-548 (2014).