

Webプログラミング教育環境 SEKENS

Taihei Sugimoto

Electronic and Computer Engineering,
Yokohama National University, JAPAN

Abstract

本稿では、近年、大学などの教育機関で Web プログラミング教育環境が用いられている。しかし、Web プログラミング教育環境は、ソースコードの引用や評価材料がソースコードのみであることにより、学習者の得点差が表れにくい。結果教育者が、学習者のプログラミングスキルに見合った、正当な評価を行い辛い。そこで我々は、そういった問題を解決するための機能を実装した Web プログラミング教育環境 SEKENS(シーケンス, *Social Environment for Konoha Education Networking System*) を提案する。

1 Introduction

(Background) 近年、大学等の教育機関を中心に、Web プログラミング教育環境が普及している。特に、教育者が課題を提示し、学習者がプログラムを作成し提出する形式の授業で用いられることが多い。

一般に Web プログラミング教育環境とは、Web 上(サーバサイド)に教育対象となる言語の実行環境があり、学習者の提出物や提示した課題等をサーバ側で管理するものである。Web プログラミング教育環境を使うメリットとして、学習者が自己の PC にプログラミング環境を構築する必要がない、学習者が他者と自己のソースコードを比較できる、紙媒体やメールでの提出形式と比べ、学習者の提出物等のデータをリスト管理しやすい、といったものが挙げられる。

実際に、クイーンズランド工科大学では、1 年生の Java や C の授業で Environment for Learning to Program (ELP)[1] という Web プログラミング教育環境が使用されている。ELP は Web 上にソースコードの穴埋め問題を出し、学習者が提出したソースコードを独自のプログラム解析フレームワークによって自動で添削する。結果、ソースコードの評価を自動で行えることで、フィードバックを素早く行える等の点で、プログラミング教育に役立っている。

しかし、Web プログラミング環境のように他者のソースコードがオープンに公開されている場合、他の学習者の記述したソースコードを引用すれば、プログラミン

グの不得意な学習者でも他者と同等の得点を取ることが可能である。そこで、ソースコード以外の評価材料を用いる必要が出てきた。評価基準がソースコードのみの場合、引用したソースコードはもちろん、基礎的な演習課題のように、解答がパターン化されているもの等は、どの学習者でも同様のソースコードになりやすく、学習者の得点がどれも同等のものになることが多いからである。

(Problem) 結果、Web プログラミング教育環境では、教育者がプログラミングが得意な学習者・不得意な学習者の区別が正確に行えない。といった、問題が生じる。

(Objective) 教育者が、プログラミングが得意な学習者・不得意な学習者の区別を正確に行えるための Web プログラミング教育環境の構築。

(Proposal) そこで我々は、Web プログラミング教育環境 SEKENS(シーケンス, *Social Environment for Konoha Education Networking System*) を提案する。SEKENS は、ソースコードの引用に制限を加え、ソースコード以外の評価基準を設けることで、教育者がプログラミングの得意な学習者・不得意な学習者の区別を行いやすいプログラミング教育環境である。

(Contribution) SEKENS が実際のプログラミングの授業の現場で実用されることを目指す。

(RoadMap) 本論文は次のように構成される。第 2 章では、既存のプログラミング教育環境で生じる問題について述べる。第 3 章では、SEKENS の設計方針について述べる。第 4 章では、SEKENS の現在の開発状況について、説明する。第 5 章では、SEKENS の評価方法の予定について述べる。第 6 章では、関連研究について紹介する。第 7 章で、本論文を総括する。

2 Problem Statement

Web プログラミング教育環境では、他者のソースコードをコピーし、提出を済ませてしまう学習者がいる可能性がある。そんな中、ソースコード以外で評価する指標が構築されていなければ、ソースコードの内容に違いがない場合、どの学習者も同等の評価をしなければならぬ。そういった場合教育者は、プログラミングが不得意であるため課題を解くことができないために他者のソースコードを引用して提出した学習者に気付くことがで

きない。

結果、教育者は、プログラミングの得意な学習者・不得意な学習者を正確に区別できないため、正当な評価を行い辛い。そのため、学習者のプログラミングスキルに合わせて特別課題の提示や補講等を行う、といったことが辛い。

3 SEKENS

3.1 目的

- 引用されたソースコードとそうでないソースコードの区別が可能
- ソースコード 以外の判断基準を設ける

といった要求を満たすことで、教育者が、プログラミングが得意な学習者・不得意な学習者の区別を正確に行えるための Web プログラミング教育環境の構築をすることにした。

3.2 要求

我々は、SEKENS では以下の 2 つの要求を満たすべきであると考える。

- 引用行為の制限，明示化
- ソースコード 以外の評価方法の導入により，学習者間の得点差を大きくする

3.3 設計

3.3.1 条件付き引用

SEKENS では、ソースコードの引用行為自体は禁止していないが、以下の条件付き引用を認めるようにしている

- 引用の明示化：他者のソースコードをコピー＆ペーストできないようにし、引用の際には、引用元・引用先の両者に引用した事が分かるようにし、そのソースコードが引用物か否かを判別可能となる形式で引用できるようにする。
- 引用の制限：引用物をそのまま提出することは、無条件でコピー＆ペーストすることと変わらないため、引用したソースコードにバグを混入したり、一部欠損させることで、それを防いでいる

SEKENS では、明示的な引用，制限付きで引用することを合わせて fork と呼ぶ。

3.3.2 学習者の評価方法

SEKENS では、ある条件下では引用を許可しているため、ソースコードのみの評価材料で学習者の評価を行った場合、特に fork されたソースコードと fork 元のソースコードの違いが少ない場合、学習者の得点差が表れにくいいため、以下の評価基準を設けている。

- fork の有無：fork は、プログラミングが不得意であり、課題が解けなかった学習者が行うことが多い。そのため、fork の有無を評価基準に含める。
- fork された数：fork された数が多いソースコードは可読性が高く、質の高いソースコードである可能性が高いことを利用し、fork された数が多いソースコードを作成した学習者により高い得点を与えることとする。
- 課題提出までの時間：引用の制限によって、ソースコードを fork してから提出できる状態にするまでに時間をかけさせることができるため、この評価基準を設けることで、fork を行った学習者とそうでない学習者の間に、fork したか否かの評価よりも正確な得点差が生じるようにしている。また、コーディングの速度も十分な評価材料となることも考慮している。

これにより、SEKENS では、課題提出までに要した時間、fork された数も学習者の評価基準とし、より速く提出された物、より多くのユーザに fork された物に高い得点を与えることとする。

また、SEKENS は静的型付けによるスクリプティング言語 Konoha の教育を対象としている。Konoha は、幅広い記述ができるため、学習者の中にプログラミングが苦手な初学者が多い状況でも評価が行い易いため採用した。

3.3.3 その他の設計

- ユーザの情報，ソースコードの情報は登録ユーザなら誰でも閲覧できるようにする。
- ユーザは、Teacher と Student のどちらかとして登録できることとし、Student は Teacher のグループに参加すれば、提示された課題を解く事ができ、Teacher の出す課題は Student 全員に一斉に送信されるようにする。
- Teacher が課題を提示すればそのグループに所属する Student 側に課題提出用のディレクトリとファイルが自動で作成される。
- グループシステム，課題提示システムは Student 側でも自己のグループを持つことにより行え、学習者間で問題を出したり、共同開発に応用することも可能とする。

- 無条件でコピー＆ペーストができないようにするため、他者のソースコードはテキストでなく、画像データとして表示するようにする。
- fork の引用元・引用先が確認できるよう、fork による繋がりが分かるようなツリーを作成し、表示できるようにする。
- fork は他者のソースコードディレクトリから行え、fork したコードは「forked from student A」というように fork したことが分かるようにし、fork 元となるコードには「forked by 2 students」というように何人の学習者に fork されたかを分かるよう表示する。
- ソースコードの評価の際、「fork した否か、fork された数、課題提出までに要した時間」が自動的に得点に加えられるようにする。
- GUI の部分はシンプルなものとし、ページ遷移を伴わないようにする。

4 Implementation Status

SEKENS の実装は以下の方法で行う。

- GUI:Ext Js 4.0 ライブラリを使用して JavaScript
- ユーザ情報の管理:MySQL を konohaScript にバインドしパッケージ化したものを用い、ユーザ名、パスワードや、グループ情報を管理する。
- ユーザのソースコード管理:git のコマンドを利用する .fork ,merge ,add ,push ,pull 等を用い,SEKENS での fork ,ソースコードの Save ,Load ,ユーザ登録時のレポジトリ作成等を行う。
- サーバ側 CGI の記述:KonohaScript を用い,SQL の操作や git の API 使用を行う。また、ソースコードの実行でも KonohaScript の eval 関数を呼び出す。

現在、GUI の部分が概ね実装済みで、ユーザ登録や、ソースコードの Save 等の UI 部分は見掛け上できているが、内部的な処理(ユーザ情報とソースコードの管理等)をするための実装をしている。また、SEKENS の現在の外観を Figure1 に示す。

5 Evaluation & Consideration

6 Related Work

本論文では、教育向け Web プログラミング環境を提案した。本論文と関連するものとしてクイーンズランド工科大学にて実際に 1 年生の Java と C の授業で使われている Environment for Learning to Program (ELP)[1]

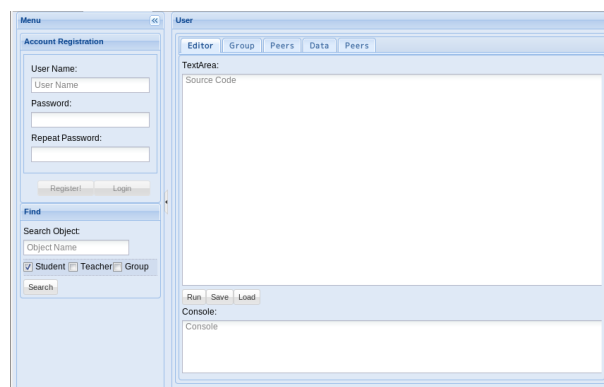


Figure 1. SEKENS

がある。ELP は Web 上にソースコードの穴埋め問題を出し、生徒が解いた物に対して独自のプログラム解析フレームワークによって添削を行い、生徒にフィードバックを行う、インタラクティブな Web ベースのプログラミング環境である。ELP のプログラム解析フレームワークは静的解析と動的解析の 2 種類に分けられ、静的解析ではソースコードの解析を行い、ソフトウェアメトリクスや正解となるソースコードとの類似性といった指標をもとにした解析を、動的解析ではブラックボックステストとホワイトボックステストを行い、それらのテスト結果をもとにした解析を、それぞれ行うことで添削を行っている。

7 Conclusion

SEKENS によって、プログラミング学習者の得点差が各々のスキルに応じて表現されることで、プログラミングの得意な学習者・不得意な学習者の区別ができるようになることが望ましい。また、SEKENS 独自の採点方式の信憑性が証明されれば、既存のソースコード自動採点システムと合わせて用いることで、学習者の評価を全自動で行うことが可能になる。以上のような貢献をすることでプログラミング学習者各々にスキルに応じた指導に繋がられ、プログラミング教育の質を向上させることができれば望ましい。

Acknowledgment

References

- [1] N. Truong, P. Bancroft, and P. Roe. Learning to program through the web. *SIGCSE Bull.*, 37:9–13, June 2005.