

プログラムの実行・評価機構を持つ Web教科書によるソフトウェア開発技能育成

A Web-based Execution and Evaluation Environment for Programs Designed for Honing Software Development Skills

松浦佐江子 芝浦工業大学デザイン工学部

Abstract: Recently, there have been significant advances in information technology, and the industrial world is increasingly expanding the target domain by using information technology. However, there is a shortage of software engineers. Moreover, a serious problem is that there are few excellent software engineers who have been educationally guided from a basic and essential level. In view of the urgent need of the industrial world, universities should continuously produce students with high-quality software development skills that are useful to the industrial world. To develop high-quality software development skills, students need to learn many types of processes for resolving various problems commonly encountered by software developers. Besides, they need to have the skill to judge whether their programs are appropriate from the viewpoint of software quality. The current educational curriculum at universities is insufficient to hone such skills. This paper proposes a Web-based execution and evaluation environment for programs designed for honing software development skills

Keywords: Web-based programming environment, software engineering education, software quality.

1. はじめに

近年IT技術が発展する一方、産業界ではソフトウェア開発技術者不足のみならずソフトウェア開発技術教育を受けていない技術者によるソフトウェアの品質低下という問題も生じている。産業界の要請に答えるために、大学はソフトウェア開発技能をもつ質の高い人材を産業界に輩出する仕組みをもつ必要がある。我々は、文部科学省平成21年度大学教育・学生支援推進事業【テーマA】大学教育・学生支援推進事業に「工学系技術者のソフトウェア開発技能育成」のテーマで採択されたことを契機にIncusphere Project^[1]を立ち上げ、本学SIT総合研究所ソフトウェア開発技術教育研究センター^[2]の活動を行っている。

ソフトウェア開発技能を育成するためには

ソフトウェア開発で遭遇する問題解決プロセスを学習し、自分でソフトウェアの良し悪しを判断できる訓練が必要であるが、大学の学部教育カリキュラムだけでは、学習の機会が不足している。ソフトウェア開発技能を育成するために、多くの大学ではPBL (Project Based Learning) による実践的なソフトウェア開発実習を実施しており、本学でも10年間に亘り半期の実習授業としてPBLを実施している^[3]。しかし、「結局はコーディングで苦労した」「工程を体験することはできたが、目的を達成できたかはわからない」「成果物の品質って何?」といった体験以上の、ソフトウェア開発技術者として必要な技能への理解には至らないことが多い。

また、演習では講義で得た知識を理解し、活用することを目的とするが、単に文法だけを学習しても、プログラムの問題では、「問題そのものを読み解いて解決する方法が判らな

Saeko Matsuura
Shibaura Institute of Technology
E-mail:matsuura@se.shibaura-it.ac.jp

い」「コンパイル・実行はできたが、これで本当に良いのかを判断できない」といった問題が学生側にはある。さらに、教授者側には、「多数のプログラムを機能的要件以外の品質まで詳細に評価することが不可能である」「学生の躓いている問題点（問題意図の理解方法が判らない、アルゴリズムの組み立て方がわからない、文法が判らない）を把握して適切に指導することが困難である」という問題がある。そこで、PBLによる実践的なソフトウェア開発実習において重要な開発工程の目的や成果物の品質を意識できるように、演習だけでは不足するこれらの知識をプログラムの開発を通して事前に学習させる必要がある。

本教育改善では、学部教育カリキュラムに基づく体系的なシナリオベースの教科書とソフトウェアの良し悪しを評価する評価ツールの教材をWebブラウザ上で動作するプログラミング学習環境として開発した。本稿ではこのWebアプリケーションとして実現したシナリオベースの教科書をWeb教科書と呼ぶ。シナリオとはソフトウェア開発者が開発工程において遭遇する具体的な問題とその解決方法のことであり、教科書ではソフトウェア開発プロセスや品質の観点からのシナリオに基づき、説明と問題を構成し、問題解決プロセスの学習を支援する。

さらに、本質的にソフトウェアが提供するサービスや製品の品質の良し悪しを判断できる能力を育成するために、ソフトウェアの品質を評価する評価ツールを用いて、自分でソフトウェアの良し悪しを判断する力を身につけることを支援することにより、ソフトウェア開発技術者に期待される総合的なソフトウェア開発能力の育成を目指す。

2. Web教科書の基本方針

(1) シナリオベースの教科書

ソフトウェア開発では、ある要求を満たすプログラムの実装方法は多数存在する。単に動作させるのみならず、それぞれの違いと、ソフトウェアの品質に与える効果をソフトウェア品質特性（ISO/IEC9126^[4]）のうち、機能性・使用性・効率性・保守性に着目して整理する。要求通りに動作する（機能性を満たす）だけではなく、使いにくい（使用性が低い）、速度が遅い、メモリを無駄遣いしている（効率性が悪い）、無駄があり変更しにくい（保守性が低い）という意味を、事例に基づき説明し、問題により確認する。さらに、段階的に問題解決過程を学習できるように、単一から複数データへの変更、手続きにおける固定からパラメータへの変更、一連の手続きのメソッド化、基本フローと例外フロー、事前条件を定義することで機能の複雑度を減らすといった様々な観点から、要求分析、設計、実装等の開発段階や技術の難易度に応じてシナリオを具現化する。これをストーリーと呼び、Web教科書での学習単位とした。

(2) 評価ツール

(1) 節で述べた品質の観点について具体的にプログラムを動作させ、問題箇所を評価ツールにより指摘する。しかし、評価ツールの評価結果のみでは、なぜそれが悪いのか、どう直せばよいのかはわからない。Web教科書では評価ツールをWeb教科書の問題に組み入れ、要求を満たすプログラムの多様な実現方法の良し悪しをその状況下での評価の意味と修正の指針とともに学習し、自力で評価結果を理解できるようにする。評価内容は、命名規則の適切さ、未使用変数の存在、未使用メソッドの存在、変数のスコープの適切さ、コード行数等のサイズ・メトリクス、アクセス制御の使用状況、クラス内でのフィールドとメソッドの依存関係等である。

評価ツールでプログラムを評価した結果には、絶対評価が可能なものと、相対評価することによって改善につながるものがある。例えば、前者は未使用変数の発見や、依存性の発見等であり、後者は、コード行数や命名規則などがある。後者の場合には、評価結果の値や解答を他の学生の結果と比較することにより、自分のプログラムの評価位置を認識できるようにする。

3. Web教科書による学習

(1) 学習環境の設計と効果

Web教科書は、教材を見ながらプログラミングできるWebアプリケーション環境である。コンパイラおよび実行環境はサーバ側に複数言語に対応して用意し、学生はIDE（統合開発環境）インタフェースを通じてプログラミングすることができる。インストールや設定作業などは初学者にとっての一つの壁であり、これをなくすことで、プログラミングを始めやすくする。また、CおよびJavaをサポートし、一つの教材の中で複数の言語の使い方を対比しながら学習することで、言語によってどこが同じでどこが違うかを学習して、他の言語を学習する際の壁をなくす。

(2) 段階的なプログラムの学習プロセス

教授者は想定するシナリオに基づき、穴埋め、選択、コード問題等の問題と説明を組み合わせるストーリーを作成する。例えば、オブジェクト指向開発における「オブジェクト」の概念を学習する際には、問題の中のオブジェクトを認識し、その属性や操作は何か、どのように作成し、問題を解くためにオブジェクトにどのように仕事をさせるのかを学習する。「複数の長方形の面積を計算するプログラム」を作成するという問題の場合、「幅12で高さが5の長方形の面積は？」といった具体

的に計算してみることを穴埋め問題で行い、オブジェクトの属性や操作を認識することから始める。次に「点」オブジェクトのソースコードを見ながら、図1のように、クラス、フィールド、コンストラクタ、メソッドがどれであるかというオブジェクトの構成要素の確認をドラッグ&ドロップ方式の選択式問題で行う。解答すると自動で正解判定を行い、図1のように、判定結果やコメントが提示される。学生はその場で答えを確認し、コメントを参照しながら正解までの道をたどる。

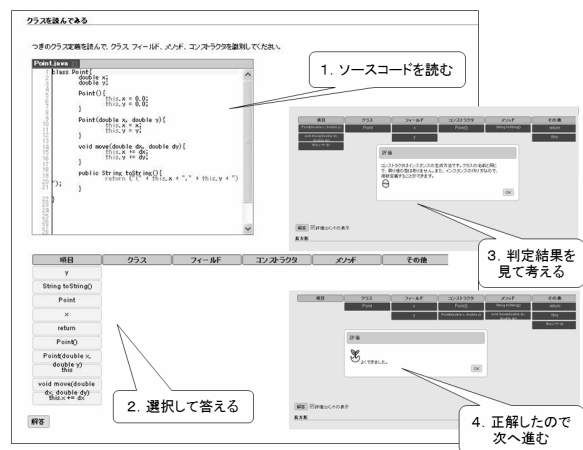


図1 オブジェクト構成の学習例

次に、学生は図2のように、オブジェクトの作り方や振舞いをソースコードの穴を埋めて解答し、その解答結果を実際にIDEを使って動かしながら、期待する結果が得られるまで、コードの修正を繰り返して学習する。穴埋めの解答はそのまま残るので、自分がどこ



図2 プログラムの実行による学習例

を間違えていたのかも確認することができる。
 ソフトウェア開発ではプログラムを定義するまでに要求や設計の問題を分析する段階が重要であり、分析時にはモデリング言語UML(Unified Modeling Language)^[5]を使用することが多くなっている。Web教科書では、問題の分析にUMLのアクティビティ図を用いた図3のような選択問題を提供しており、学生は早い段階からモデルを読むことで、UMLの具体的な利用方法を学習し、問題解決の一つの方法を知ることができる。

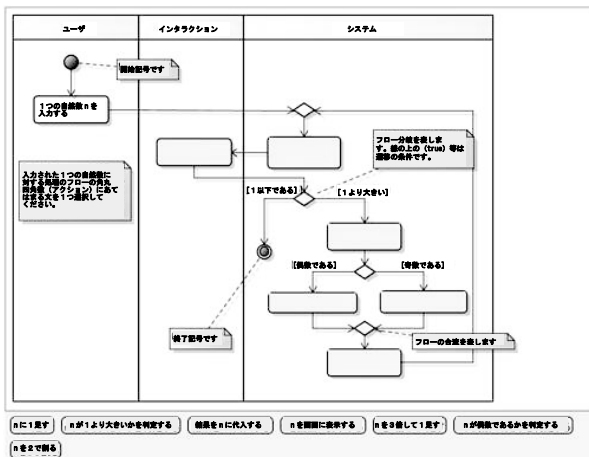


図3 UMLモデルを使用した選択問題

(3) 評価ツールによる学習

プログラムの評価は選択問題などのように正解判定することが困難である。Web教科書では、問題に組み込んだアドオンを動作させることで、コードの品質の評価や、特定のコード生成を行うことができる。これまでに、メトリクス測定による品質評価^[7,9]と、命名の学習^[6,8]の評価ツールをアドオンとして開発した。

メトリクスを用いた品質評価では、2.(2)で述べた観点により、未使用要素がないか、変数のスコープは適切か、アクセス制御は適切か、マジックナンバーはないかといった、コードを読み解き、変更を行う上で妨げとなる項目について学習する。説明とともに、サンプルコードを評価し、学生が問題箇

所を修正することで、問題点の意味を学習する。

プログラムの作成において、識別子の命名は保守性の観点から重要であるが、正解となる名前が常に規定できるわけではないので、絶対評価により学習することが難しい。Web教科書では、同じ課題に対する解答者全員の答えを集計し、他の解答と比べることで、開発者の共通認識が取れる適切な命名とは何かを協調学習する。教授者がクラスの構成要素の理解やデータ構造とアルゴリズムの理解等の学習目標に合わせて、識別子を難読化したソースコードを提供し、学生は実際にそのコードを動作させながら、適切な命名を行う。解答すると、他の学生の解答と合わせて集計され、学生は命名のランキングを見ながら、自分の命名を自己評価し、必要に応じて、答えを変更しながら学習する。教授者は、学生の解答状況を別のストーリーの題材として命名の説明や評価を行うことで、学生の解答へのフィードバックを行うことができる。

(4) 学習状況の把握とフィードバック

学生および教授者は、図4のように学習結果を閲覧する。

学生はストーリー毎の評価点を閲覧するだけでなく、他の学生の点数と比較することが



図4 学習状況の集計

できる。教授者が各問題にソフトウェア開発の過程に必要な分析力・設計力・実装力・テスト力・保守力の項目を設定することで、個別の学習状況のみならず、学生は総合的にソフトウェア開発に必要な技能が身についたかを客観的に知ることもできる。

教授者は、集計アドオンを作成することで、例えば、学年毎、学科毎、授業のクラス毎といった単位で学習状況を集計し、ストーリーの特性（分析力・設計力等、難易度、キーワードによる学習項目）のカバー範囲を分析し、ストーリー作成にフィードバックする。

（5）教材作成の容易性

教材作成の容易性は、継続的に多くの教材を提供するために、重要である。Web教科書におけるストーリーの作成手順は次のとおりである。

プログラミングの問題であることから、サンプルプログラムをリソースとして登録し、コードの例示やコード問題に利用する。コードはリソース用のIDEで適宜編集可能である。

説明・穴埋め問題・選択問題・コード問題といった部品を並べてストーリーを作成する。

各部品に対して、問題作成コマンド(入力欄化、コードの挿入、画像の挿入等)を使って、問題を編集する。コード問題では評価ツールを登録することで、品質評価等を行う。

編集結果をプレビューやシミュレーションにより、学習者の立場から確認する。

ストーリーが完成したら、各問題の評価観点（分析力、設計力・実装力、テスト力、保守力）を設定する。

図5の問題を例として、選択問題の作成方法を説明する。これは、課題の説明文を受け

て、「クラス・フィールド・メソッド」を識別する問題であり、答えは点線枠で囲われたように整理できる。例えば任意のテキストエディタで編集したテキストをコピーして、Web教科書の部品の編集画面に張り付け、「答え」の部分を選択した状態で、問題作成用コマンドから「入力欄化」を選択すると、右側に示すドラック&ドロップ方式の選択問題が生成される。この時点で、正解を判定する仕組みができており、すぐにプレビューまたはシミュレーションで操作と判定結果を確認することができる。

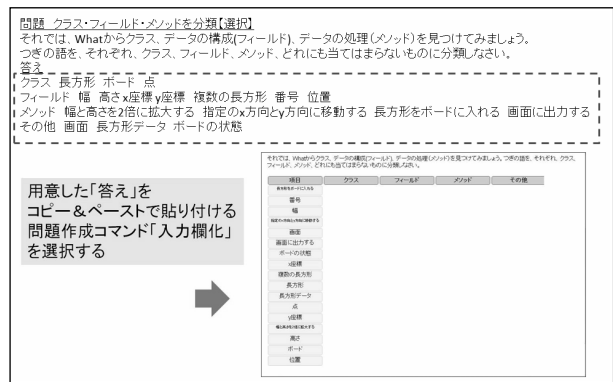


図5 選択問題の作成

4. 教育実践による改善効果

Web教科書は2011年4月に学生モニタによる試験運用を実施し、2011年9月からは本学2学科の学生向けに運用を開始した。現在公開中のストーリーはJava初級コースで14、中級コースで2である。学習している学生数は初級116名、中級43名であり、学習者の延べ人数は376名である。実際に授業等で使用し、応答速度、同時アクセスに関する問題を改善し、使用上の大きな問題はないことを確認している。また使用後の学生アンケートでも「ストーリーにそって学習するのでわかりやすい」「プログラミングは難しいと思っていたが、やれそうな気がした」等、好評であった。いつでも好きなストーリーを学習できるため、授業で指定したストーリー以外にも

学生が自主的にストーリーを選択して学習している様子を見ることができる。

Web教科書は、本来は授業の補助として、学生の自習教材の役割をもつが、授業内で一斉に学習させ、実際にどのくらいの時間で解けるのか、どこで詰まっているのかを把握することにも使用している。通常のプログラミングの演習授業では、学生は与えられた課題に対して、十分に考えずに場当たりのソースコードを書き始めることが多い。段階的に考えていないため失敗することが多いが、演習授業内でのWeb教科書による学習状況を観察した様子では、具体的に考える、コードを読む、構成要素を考える、動かす前にコードを書いてみるといった段階を経ることで、スムーズにコーディングが行え、授業時間内にほとんどの学生が終了に至っている。また、学生がどこで躓いているかもその場でわかるので、個別の指導を行いやすい。

Web教科書で評価の意味を学習した後は、学習効果を高めるために、学生がそれぞれの課題に取り組む際にも、同じ評価ツールを用いてプログラムを改善する訓練を行うことが望まれる。常に品質を意識したプログラミングができるように、同じ評価のエンジンを単独でも利用できるツールとして提供している。

5. まとめ

シナリオベースの教科書の提供形態として、学生が取り組やすいWebベースの教科書を実現した。Web教科書では、教授者が通常の試験や演習問題を作成する要領で、容易に教材を作成できるように工夫している。また、学習状況把握のためのアドオンを用意し、教授者が必要に応じたデータ収集を行うことができる。現状では、試験的な運用であるが、学生が自主的に学習しており、プログラミン

グを学びたい学生は多数いることから、今後も継続して教材の充実と改善を行い、講習会の開催、ホームページによる広報による普及ならびに教材開発者の利用の促進を図りたいと考える。

参考文献および関連URL

- [1] <http://www.sayo.se.shibaura-it.ac.jp/incosphere/> (2012.10.1.参照)
- [2] http://www.shibaura-it.ac.jp/society/advanced_engineering_organization/center07.html (2012.10.1.参照)
- [3] 松浦佐江子: 実践的ソフトウェア開発実習によるソフトウェア工学教育. 情報処理学会論文誌, 48, 8, pp.2578-2595, 2007.
- [4] <http://www.iso.org/iso/> (2012.9.1.参照)
- [5] <http://www.uml.org/> (2012.9.1.参照)
- [6] R.Shikimi and S.Matsuura : Collabo-rative Learning Environment for Identifier Naming in Source Codes, Proc. of CATE 2012, pp.106-111, 2012.
- [7] 若林智徳, 松浦佐江子: オブジェクト指向プログラミング初学者向け評価手法を用いたソースコード改善学習支援. 第74回情報処理学会全国大会, 2012 (学生奨励賞受賞).
- [8] 式見遼, 松浦佐江子: 識別子の難読化と命名の収集による命名学習方法の提案. 第74回情報処理学会全国大会, 2012 (大会奨励賞受賞).
- [9] T. Wakabayashi, S.Ogata and S. Matsuura: Dependency Analysis for Learning Class Structure for Novice Java Programmer, Proc of IEEE ICSESS, pp.532-535, 2011.
- [10] Web Study, <http://www.sayo.se.shibaura-it.ac.jp/incosphere/webStudy.html> (2012.10.1.参照)

本研究は文部科学省平成21年度大学教育・学生支援推進事業【テーマA】「工学系技術者のソフトウェア開発技能育成」の一環として実施した。