

科学研究費助成事業 研究成果報告書

平成 28 年 5 月 30 日現在

機関番号：32407

研究種目：若手研究(B)

研究期間：2013～2015

課題番号：25750090

研究課題名(和文) 細粒度履歴情報をリアルタイムに活用するプログラミング支援環境の研究

研究課題名(英文) A Study for Collecting Fine-Grained History Information to Help with Programming Exercise in Real Time

研究代表者

橋浦 弘明 (HASHIURA, Hiroaki)

日本工業大学・工学部・助教

研究者番号：20597083

交付決定額(研究期間全体)：(直接経費) 2,800,000円

研究成果の概要(和文)：プログラミングの教育は演習を中心に行われており、演習による知識の定着の度合いを評価するために、成果をレポートとして提出させる場合が多い。レポートは演習の最終結果を確認することはできるが、学習者がソフトウェアを完成させていく過程や、その途中でどのような問題に直面し、どのように対処したかを教授者が知ることは難しい。本研究では Eclipse と連携して、学習者のプログラミングの細粒度履歴をリアルタイムに収集し、ネットワークで接続されているサーバに蓄積することにより、ソフトウェア開発演習を支援する環境を構築した。

研究成果の概要(英文)：Programming education is conducted with a focus on exercise, and it is often the case that the result is made submitted as a report, in order to evaluate the degree of fixation for the knowledge by the exercise. By using the report, one can check the final result of the exercise, but it is difficult for the educator to know the process by the learner to complete making the software, or on its way, to face what kind of problems are faced and to cope with them. In this study, in conjunction with Eclipse, a learning environment to support software development exercise is constructed by collecting fine-grained record of learner's programming in real time and by storing it to the server connected to the network.

研究分野：ソフトウェア工学

キーワード：プログラミング ソフトウェア工学教育 細粒度履歴 IDE 統合開発環境

1. 研究開始当初の背景

高度なソフトウェア開発技能を持つ人材とは、単にプログラミングができるだけでなく、ソフトウェアの全体が俯瞰でき、またその本質を捉え、それを必要なときに必要な形で表現できるモデリング(設計)能力を備えていることが必要となる。申請者らはこのような状況に対し、自らが主体的に問題解決を行うソフトウェアを開発演習である Project-Based Learning(PBL) やその支援環境について研究を行ってきた。このような PBL を行うにあたって、学生はソフトウェア開発の要素技術を予め習得しておかなければ十分な効果は期待できない[1]。しかしながら、要素技術の習得に重点を置きすぎると、十分に PBL を行う時間がとれない。このため、要素技術を早期かつ確実に習得させる必要が生じる。

学習者の作成したプログラムの中には、機能的に問題がなくとも、品質的な問題を含んだプログラムが多数存在することが判明している[2]。

ソフトウェア品質特性については ISO/IEC9126-1[3]において 6 種類に大別されているが、初級のプログラミング演習において、課題が一定の複雑さを持つようになると、機能適合性以外の品質特性に対する理解が不十分であるために、受講者自らが気づかないうちに機能適合性以外の品質が低いコードを生産してしまい、そのことが課題完遂への障害となることがある。このような事態を避けるため、教授者は解説の合間に機能適合性以外の品質特性とその低下を防ぐための工夫について解説を行い、常に他の品質特性に気を配るように指導を行っているが、残念なことに、受講者はこれまでに得た知識と、新たな知識を組み合わせながら解答を作成することで精一杯であり、常に機能適合性以外の特性に気を配るような余力はないのが実情である。

このような問題を解決するためには、プログラミングの初期段階から、ソフトウェア工学的観点に則り、段階的かつ継続的にソフトウェア品質についての教育を行うことが必要であると考えられる。しかしながら、機能性以外の品質特性を評価するためには、単にソフトウェアを動作させるだけでなく、実際に作成されたプログラムの内容を詳細に調査する必要がある。さらに、受講者が抱えるコードの品質問題は個人個人で内容が異なっている。よって、このようなソフトウェア品質教育を多人数の受講者に対して行うためには膨大なリソースが必要となる。そのため、これまでプログラミングの初期の段階で品質教育を行うことは難しかった。

<引用文献>

- [1] 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, Vol.48, No.8, pp.2578-25

95, 2007.

- [2] H. Hashiura, S. Matsuura, S. Komiyama, "A Tool for Diagnosing the Quality of Java Program and a Method for its Effective Utilization in Education," Proceedings of the 9th WSEAS International Conference on APPLICATIONS of COMPUTER ENGINEERING (ACE'10), pp.276-282, Mar. 2010.
- [3] "ISO/IEC 9126-1:2001 Software engineering - Product quality - Part 1: Quality model," International Organization for Standardization, 2001.

2. 研究の目的

提案者らは以前からプログラムの品質診断ツールをプログラミング演習に導入し、演習で提出されたプログラムだけでなく、作成中のプログラムについてもリアルタイムに静的解析を行い、ソフトウェアの品質について自動的に学習者にフィードバックを行うという試みを行ってきた[4]。これは学習者にプログラムの品質をプログラミングの早い段階から常に意識させ、悪い癖がつく前に自発的な改善を促すという狙いがあった。

このような試みを進めていく中で、ツールから表示されるフィードバックが大量に表示されることが問題となった。これは、教授者が必要最小限度に絞り込んだと思っているルールであっても、プログラミングのスタイルがまだきちんと定まっていないプログラム初学者にとっては荷が重く、結果的として必要以上のフィードバックが行われることとなったものである。大量のルール違反のフィードバックは、もはや学習者にとって有益ではなく、多くの場合、逆にこれが心理的な負担となって、プログラミングに対するやる気を削ぐ結果をもたらしていた。

このような問題を解決するために、学習者の状態をより詳細に把握し、状況にあった支援がリアルタイムに行える環境の実現を目指すこととした。これにより、学習者、教授者に以下のような効果が期待できる。

学習者への期待される効果

- 自分の作成したソフトウェアの品質特性(どこが悪いのか、どこが良いのか)を認識することができる

教授者への期待される効果

- 躓いている学習者の発見につながる
- 難易度の調整やフォローに役立てることができる

<引用文献>

- [4] 橋浦弘明, 松浦佐江子, 古宮誠一, "Javaプログラムの品質診断ツールとその教育への効果的利用方法", 電子情報通信学会技術報告 教育工学, Vol.109, No193, p.51-56, 2009.

3. 研究の方法

本研究ではソフトウェア統合開発環境 (IDE: Integrated Development Environment) と連動し、学習者の細粒度履歴情報 (プログラムの更新履歴) を随時取得し、得られた情報をリアルタイムに解析することにより、開発中のプログラムの悪所を即座に指摘する環境を構築する。IDE と連携することにより、既存のカリキュラムに影響を与えず、教員の負荷を増やすこともなく、学習者に対して品質に対する教育を行うことが可能となる。

加えて、従来は行われていなかった細粒度履歴情報をリアルタイムに分析するためのツール及びルールセットを開発する。処理により得られた情報を教授者や本人へフィードバックを行い、どのような支援がプログラミング品質の教育に対して有効であるのかを明らかにする。

図 1 にシステム全体の構成と悪所情報フィードバックのイメージを示す。

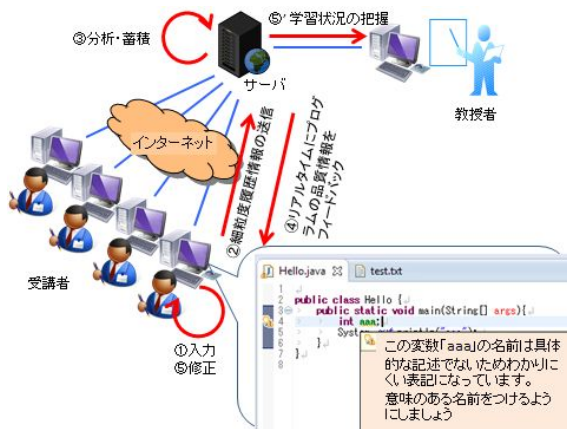


図 1 : システム全体の構成とフィードバックのイメージ

4. 研究成果

(1) 細粒度履歴情報取得環境の構築

本研究ではプログラミングの過程を詳細に把握するために行やファイル単位ではなく、ユーザーの行動単位での履歴 (細粒度履歴) の記録を行う。細粒度履歴とは、IDE における下記の一連の操作の流れとして考えることができる。

- エディタへの文字の入力
- エディタへの文字の削除
- エディタへの文字の変更
- ファイルの保存
- プログラムの実行

本研究ではこれらを自動的に収集するための仕組みを、代表的な統合開発環境である Eclipse にプラグインとして実装し、Verdure と名付けた。加えて、収集した履歴情報を蓄積するための Verdure/Server を

実装した。Verdure/Server は、Verdure から送信された履歴を学習者ごとに管理し、教授者から履歴の取得命令があると、指定された学習者の履歴情報を Verdure に送信する機能を持っている。Verdure の構成を図 2 に示す。

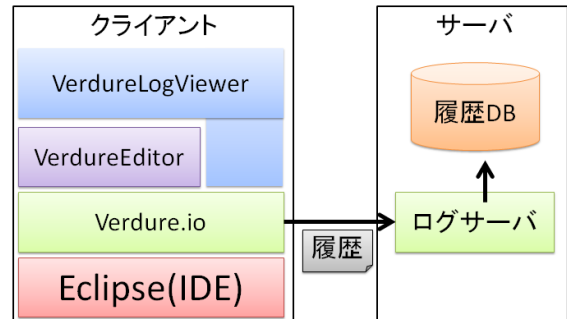


図 2 : Verdure の構成

(2) 細粒度履歴情報分析環境の構築

前章で述べた問題点を解決するために、著者らは以下の 3 つの大まかな方針を立てた。

- 一度に表示される警告の量に上限を設ける
- 複数のルールセットを用意し、学生の進捗状況に応じてルールセットを自動的に切り替える
- 教員の指示や、学生の進捗状況に応じて警告の表示 / 非表示を切り替える

これらの機能を前節で述べた Verdure に追加実装し、Verdure/STEP と名付けた。構成については図 3 に示した。

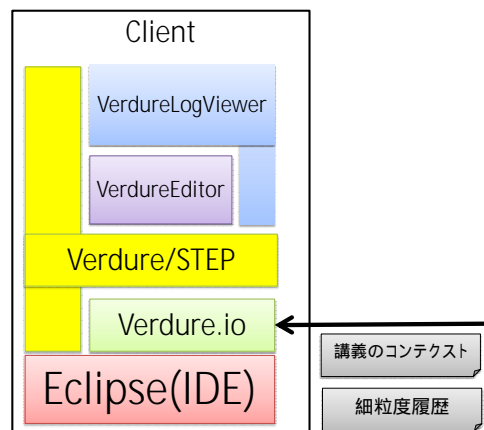


図 3 : Verdure/STEP の構成

ルールセットについては学習者にとって簡単に修正が容易な指摘から始まり、少しずつ高度な知識が求められる指摘となっていくことが望ましい。このような目的を達成するためには、ルールセットにレベルが設定できる必要がある。また、ルールセット間の関係は下位のレベルを包含するように設計す

るものとする。つまり、レベル i のルールセットを R_i とした場合、 $R_1 R_2 R_3 \dots$ を満たすようなルールセットを作成することとした。ルールセットのレベルの切り替えについては、学習者の課題の進捗状況や提出状況など、演習中に学習者の置かれている状況を細粒度履歴から読み取り、これが設定されたある条件を満たした場合に行うこととした。

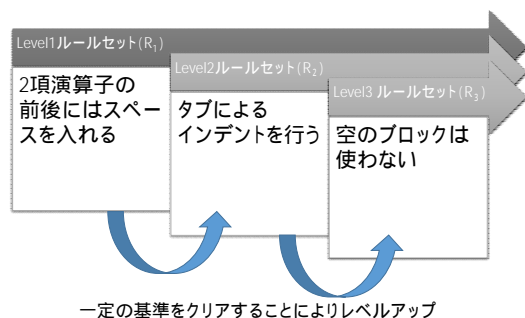


図4：ルールセットの切り替え

さらに、レベルが変化したことについては、学習者にも明示的に通知することで、ゲーム感覚でソフトウェアの品質向上を図ることができるだけでなく、自分のソフトウェアの品質のレベルを自分で認識できるようになった。

現在のシステムの現状ではルールセットの量、質共に満足なものとは言えないため、今後もプログラミング教育への適用を行い、引き続き拡充を図っていく予定である。

5. 主な発表論文等

(研究代表者、研究分担者及び連携研究者には下線)

[雑誌論文](計 7件)

橋浦弘明, 森一樹, 田中昂文, 樫山淳雄, 古宮誠一, 細粒度履歴情報と講義のコンテキストを活用したプログラミング学習支援環境, 電子情報通信学会技術報告, Vol. 115, No. 487, KBSE2015-60, pp. 69-74, 2016年3月, <http://www.ieice.org/ken/paper/20160304obHJ/>, 査読無.
Takafumi Tanaka, Hiroaki Hashiura, Atsuo Hazeyama Seiichi Komiya, Do Learners to Create an Artifact with Good Quality Make a Number of Trials and Errors during the Editing Proceedings?, Proceedings of 3rd International Conference on Applied Computing & Information Technology (ACIT 2015), pp.28-33, Jul. 2015, DOI:10.1109/ACIT-CSI.2015.14, 査読有, .

田中昂文, 橋浦弘明, 樫山淳雄, 古宮誠一, クラス図作成演習における学習者の編集過程の細粒度分析, 電子情報通信学会技術報告, vol. 114, no. 501, KBSE2014-54, pp. 13-18, 2015年3月, <http://www.ieice.org/ken/paper/20150305IBZH/>, 査読無.

www.ieice.org/ken/paper/20150305IBZH/, 査読無.

Takafumi Tanaka, Kazuki Mori, Hiroaki Hashiura, Atsuo Hazeyama, Seiichi Komiya, Proposals of a Method Detecting Learners' Difficult Points in Object Modeling Exercises and a Tool to Support the Method, Proceedings of 12th International Conference on Software Engineering Research, Management and Applications (ACIS SERA 2014), pp.650-655, Aug. 2014, DOI:10.1109/IIAI-AAI.2014.136, 査読有.

田中昂文, 森一樹, 橋浦弘明, 樫山淳雄, 古宮誠一, オブジェクトモデリング演習における学習者にとっての難所の検出手法と支援システムの提案, 情報処理学会研究報告, Vol.2013-SE-183, No.10, pp.1-8, 2014年3月, <http://id.nii.ac.jp/1001/00099398/>, 査読無.

Hiroaki Hashiura, Kazuki Mori, Takafumi Tanaka, Atsuo Hazeyama, Seiichi Komiya, An Environment for Collecting Fine-grained Development Records to Help with Programming Exercise, Proceedings of 12th International Conference on Software Engineering Research, Management and Applications (ACIS SERA 2014), pp.739-744, Aug. 2014, DOI:10.1109/IIAI-AAI.2014.150, 査読有.

Takafumi Tanaka, Kazuki Mori, Hiroaki Hashiura, Atsuo Hazeyama, Seiichi Komiya, Support System for Software Development Exercise that Utilizes Activity Data Collected in Real Time from the Development Environment, Proceedings of The 37th IEEE Conference on Computers, Software, and Applications (COMPSAC 2013), pp.726-727, Jul. 2013, DOI:10.1109/COMPSAC.2013.119, 査読有.

[学会発表](計 1件)

田中昂文, 橋浦弘明, 樫山淳雄, 古宮誠一, 学習者のクラス図作成過程における成果物と正解例との類似度の変遷を用いた進捗状況可視化手法の提案, 情報処理学会 ソフトウェアエンジニアリングシンポジウム(SES) 2015, 2015年9月7日, 慶應義塾大学日吉キャンパス(横浜市・港北区).

[図書](計 0件)

該当なし

[産業財産権]

該当なし

[その他]

該当なし

6 . 研究組織

(1)研究代表者

橋浦 弘明 (HASHIURA, Hiroaki)

日本工業大学・工学部・助教

研究者番号：20597083

(2)研究分担者

該当なし

(3)連携研究者

該当なし