

平成 30 年 6 月 18 日現在

機関番号：13801

研究種目：若手研究(B)

研究期間：2015～2017

課題番号：15K16104

研究課題名(和文) 熟練プログラマの視点に基づく着目範囲・抽象度提案型プログラミング学習支援システム

研究課題名(英文) Programming Learning Support System based on Visualization by Experienced Programmers' Perspective

研究代表者

野口 靖浩 (Noguchi, Yasuhiro)

静岡大学・情報学部・講師

研究者番号：50536919

交付決定額(研究期間全体)：(直接経費) 3,000,000円

研究成果の概要(和文)：プログラミング演習の場において、プログラミング学習者自身が開発中のソースコードを対象として、その状況に合わせて熟練プログラマ(教師)が典型的に用いる視点での視覚化を提供する学習支援システムを開発した。熟練プログラマ(教師)が用いる典型的な視点での可視化、複数の視点の可視化の実行ステップ基準による同期する機能、これらの視点での可視化及び学習者の状況に合わせた表示及びメッセージのカスタマイズを可能にするルール設計及びオーサリングルールの構築を行った。3つの授業(54名/108名/117名)を対象に授業実践を行い、これらの可視化の効果と教師によるカスタマイズ性の評価を実施した。

研究成果の概要(英文)：For the field of programming exercises, I have developed a learning support system that provides visualization with experiences programmers' viewpoints suited for programming learner's own developing source code. I have developed visualization model for typical viewpoints used by experienced programmers, synchronization function for visualizations of multiple viewpoints based on the executing step in learners' developing source code, and customizable rules for these viewpoints and supports for the learners. I have evaluated the system in 3 classes (54 people / 108 people / 117 people) about the effects of visualization and functionality of the customizable rules.

研究分野：学習支援システム

キーワード：学習支援システム プログラミング教育 可視化

1. 研究開始当初の背景

近年、プログラミング技術やプログラミングにおいて鍛えられる論理的思考力が、現代社会の一員としての一般教養として重要であるとの認識が世界的に広まっている。その中で、情報技術の専門課程ほど時間をかけられない学習者に対するより効率的なプログラミング教育カリキュラムや学習支援システムの開発が注目されている。

研究代表者は、アルゴリズム・プログラミングの初級学習者を対象とした学習支援システムを開発し^{①②}、講義等の中で実践評価してきた。その中でカード等を使ってプログラミング実装以前にアルゴリズムを学習する方法を拡張し、より効率的にアルゴリズムの振る舞いを学習できる環境^③を評価した。

その結果、高い抽象度ではアルゴリズムの振る舞いを十分に理解できた学習者であっても、そのアルゴリズムを対象としたコーディング段階の演習となると時間をかけても演習を終了できない者が一定数存在することが分かった。そして、これらの学習者はプログラミング言語の文法的知識が不足しているのではなく、以下の3つの問題からコーディング段階の演習で挫折していることがわかってきた。

- ・ 高い抽象度でのアルゴリズムの振る舞いの理解と、実装レベルでの振る舞いの理解との間に大きなギャップがあり、プログラミング初学者には自力で異なる抽象度間で概念を結びつけることが難しいこと。
- ・ 具体的なデータを対象としたアルゴリズムの振る舞いの理解と、任意のデータに対応してその振る舞いを実現する一般化されたプログラム構造の理解との間に大きなギャップがあり、プログラミング初学者には自力で両者を結びつけることが難しいこと。
- ・ デバッグ知識の不足などにより、バグの原因を特定するために着目すべき範囲を限定することができず、プログラミング初級学者は実装レベルでのトライアンドエラーに終始してしまうこと。そして、既に理解したはずの高い抽象度でのアルゴリズムの振る舞いと、その実装との関係の把握に注力できないこと。

実際の講義では、教師はアルゴリズムやプログラムを仕様レベルで説明し、学習者はそのアルゴリズム・プログラム実装の演習の中で学習することが多い。多くの学習者自身が実装レベルで作っているものと、説明を受けて理解したはずのものとのギャップに苦しんでいる。実装は学習者ごとに異なるため、教師が一律に支援することは難しい。

2. 研究の目的

本研究課題では、演習において学習者自身

が開発しているプログラムを対象に、そのプログラムの状況に合わせて、熟練プログラマー（教師）が典型的に用いる視点での視覚化を提供する学習支援システムを構築する。この学習支援システムを実現するために、以下の3点の課題に取り組む。

(a) 典型的な視点の視覚化モデルの構築

プログラミング学習の現場において、熟練プログラマー（教師）が用いる典型的な視点を抽象度別（ソースコードレベル・データモデル・振る舞いのモデル・メモリの状態などアーキテクチャに依存するレベル・デバッグのモデルなど）にモデル化する。

(b) 異視点間視覚化モデルの構築

学習者の実装したソースコードを静的分析・動的分析し、プログラミング演習課題の知識と (a) のモデルに基づき、異なる抽象レベルの視覚化を行う。これらの異なる各抽象レベルの視覚化に現れる説明概念（例：実行ステップの組、抽象データ構造の図、メモリ上のデータ状況など）間の関係を視覚化するモデルを構築する。

(c) 状況適合型視点提案機能の構築

演習時の学習者の状況を、学習者の開発中のソースコードに対する静的分析・動的分析と、学習者のコーディング・デバッグ活動の中で認識し、状況に適した(a)の視覚化モデルの組み合わせとトレース方法、視覚化モデル中のフォーカスする部分などを提案する機構を構築する。

3. 研究の方法

(1) 一般にアルゴリズム・プログラミング演習で学生が作成するソースコードは、同じ課題であっても学生ごとに異なる。また、ソースコードに含まれる誤理解やバグも千差万別である。従って、担当講師が説明用に作成したスライドに含まれる視覚化の分析に加えて、学生自身が作成したソースコードの分析を行う必要がある。実際の講義・演習で用いる題材を対象として、プログラム初学者が実際に作成したソースコード（誤理解やバグ等を含む）を元に、担当講師及びプログラミング経験を積んだ大学院生が、プログラミング初学者のソースコードに含まれる誤理解の箇所やバグの修正等を行う中で、それを効果的に説明する視覚化を行う。その結果と様子を分析し、プログラマーが用いる典型的な視覚化のモデルを設計する。

(2) 構築した典型的な視点の視覚化モデルを対象として、異なる視点の視覚化において用いられる各説明概念を関係付けて表示する視覚化モデルを構築する。一般に異なる抽象度、異なる側面（ソフトウェアの機能・構造・振る舞い）を視覚化した際に、その上で用いられる説明概念は 1 対 1 で対応する

ことは少なく、担当講師や TA は自然言語の説明を付け加えながら複数のモデル間のトレーサビリティを取った説明をしている。従って、担当講師や TA により構築した複数の視覚化モデルを用いた説明を分析し、各視点間の関係をモデル化する。

(3) 予め教師の用意するサンプルプログラムと異なり、学習者自身の開発するコードは事前にどの行にどの命令が記述されるかを想定することができず、また、予期しないバグも含まれる可能性がある。熟練プログラマの視点を教師が学習者のコードの可視化に際して与えるために、教師が自身の説明意図に基づいて上述の視覚化をカスタマイズできる仕組みを開発し、教育現場での利用を通してフィードバックを受ける。

4. 研究成果

(1) 課題(a)に対応するために、典型的な視点の視覚化のモデルを構築した。その視点での可視化を中心としたプログラミング学習支援システムを構築し、学習現場で利用しての評価を行った。抽象レベルの振る舞いの視覚化モデルを対象として、学会発表⑤⑦にてその成果を報告した。

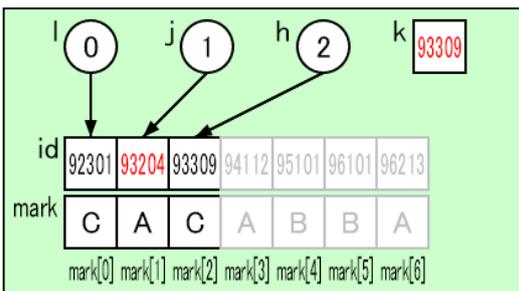


図 1: 抽象レベルの振る舞いの可視化モデル

実行ビュー	変数	値
4203520	const int(*)[7]	data
4203520	int	data[0]
4203524	int	data[1]
4203528	int	data[2]
4203532	int	data[3]
4203536	int	data[4]
4203540	int	data[5]
4203544	int	data[6]
2665552	int	key
2665532	int	low
2665528	int	high
2665524	int	middle

図 2: アーキテクチャ依存のメモリ可視化モデル

(2) 課題(b)に対応するために、課題(a)で構築した各視点の視覚化をソースコードのステップ実行の時点に同期して可視化するシステムを開発した。このシステムは、各視点で視覚化された表示をソースコードのステップ実行の時点に基づいて時間軸上同期することができる。また、各視点で視覚化された表示において学習者の選択したオブジェクトについて、それに対応するオブジェクトが

別の視点で視覚化された表示においても強調表示する機能を持つ。

図 1 図 2 の視覚化に加えて再帰及び多重ループなどを対象として授業実践において評価を行い、その成果を雑誌論文①、学会発表③⑥にて報告した。

(3) 課題(c)に対応するために、教師側が予め準備した(どの行でどのような処理を行っているかを事前に想定可能な)ソースコードではなく、学習者自身が開発中のソースコードを対象として可視化を行えるようにシステムを拡張した(図 3)。

既存の演習課題から典型的な誤り(図 4)を分析し、その講義・演習において教師が設定する教育意図と関係するそれらの誤りについて、適切な可視化と支援メッセージを提示できるように教育意図記述ルールの条件記述並びにその条件の検出機能を拡張した。

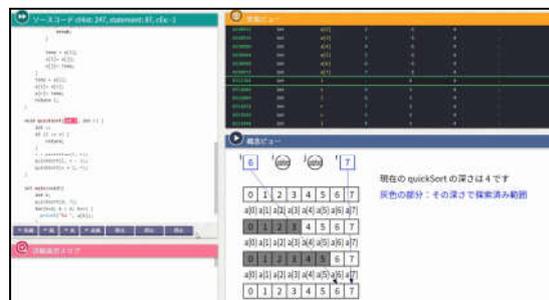


図 3: 学習者コード対応 TEDVIT

拡張したシステムについて、3つの授業(A(初級者向け):54名・B(中級者向け1):108名・C(中級者向け2):117名)を対象とした授業実践を行い、授業実践の中で学習者によって記述されたプログラムの可視化の状況の評価したところ、ほとんどの事例において妥当性の高い(教師の作成したプログラムに対する可視化と同等の可視化)可視化を行えることが確認できた。この成果は学会発表①にて報告した。

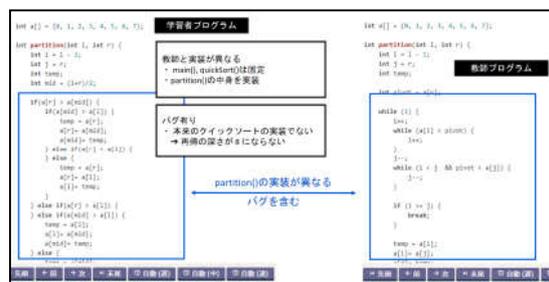


図 4: 授業実践時の学習者コード(含バグ)の例

(4) 課題(3)に対応するにあたって、熟練プログラマの視点を教師が学習者のコードの可視化に際して与える際のコストが無視できないものであることが明らかになった。この問題に対応するため、可視化ルールの記述をサポートするオーサリングツール(図 5)

を開発し、教師に提供した。このツールはルールを GUI 操作から生成できると共に、典型的なルールに関する補完機能を有する。学習現場で利用する可視化ルールの記述作業について、オーサリングツールを利用する場合と利用しない場合で比較評価をした結果、オーサリングツールを利用することで 20% 程度のルール記述効率の向上を確認することができた。この結果は学会発表④にて報告した。

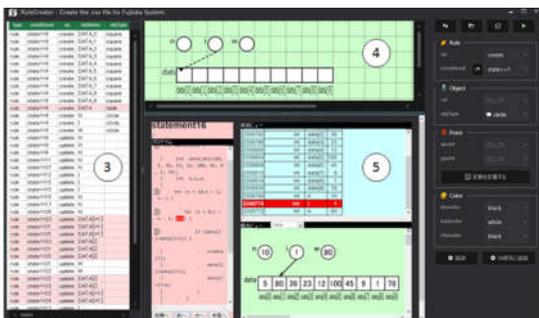


図 5: 可視化ルールオーサリングツール

(5) 上述の(1)-(4)を開発評価するにあたって、“デバッグ知識の不足などにより、バグの原因を特定するために着目すべき範囲を限定することができず、プログラミング初級学者は実装レベルでのトライアンドエラーに終始してしまう”学習者の行き詰まりの原因の一部を把握することができた。そこで、想定するデバッグプロセスに従って、学習者がデバッグプロセス上の自身の状況を確認しながらデバッグを進めることができるワークシートベースの支援システム(図 6)を開発し、評価を行った。この成果は学会発表⑥で報告した。



図 6: デバッグプロセス学習支援の例

<引用文献>

① K. Yamashita, T. Nagao, S. Kogure, Y. Noguchi, T. Konishi and Y. Itoh: “An Educational Practice Using a Code Reading Support Environment for Understanding Nested Loop”, Proceedings of the 22nd International Conference on Computers in Education, 2014.

② 藤岡僚太, 小暮悟, 野口靖造, 山下浩一, 小西達裕, 伊東幸宏: “対象世界におけるプログラムの挙動を視覚化する教材の作成支援環境”, 第 39 回教育システム情報学会全国大会講演論文集, pp. 77-78, 2014.

③ Y. Noguchi, T. Nakahara, T. Konishi, S. Kogure and Y. Itoh: “Construction of a Learning Environment for Algorithm and Programming Where Learners Operate Objects in a Domain World”, International Journal of Knowledge and Web Intelligence, Vol.1, No. 3/4, pp. 273-288, 2010.

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 1 件)

① K. Yamashita, T. Nagao, S. Kogure, Y. Noguchi, T. Konishi, Y. Itoh (2016). Code-Reading Support Environment Visualizing Three Fields and Educational Practice to Understand Nested Loops, Research and Practice in Technology Enhanced Learning (RPTeL), 11:3, pp. 1-22, DOI: 10.1186/s41039-016-0027-3. [査読有]

[学会発表] (計 7 件)

① 手塚大貴, 小西達裕, 小暮悟, 野口靖造, 山下浩一, 伊東幸宏 (2018). 対象世界上で学習者プログラムの挙動を視覚化する学習支援システムの構築, 電子情報通信学会研究報告.

② R. YAMAMOTO, Y. ANZAI, S. KOGURE, Y. Noguchi, K. YAMASHITA, T. KONISHI, Y. ITOH (2017). Learning Environment for Recursive Functions by Visualization of Execution Process, Proceedings of ICCE2017.

③ K. Yamashita, R. Fujioka, S. Kogure, Y. Noguchi, T. Konishi and Y. Itoh (2016.12). Learning Support System for Visualizing Memory Image and Target Domain World, and Classroom Practice for Understanding Pointers, Proceedings of ICCE 2016.

④ D. Tezuka, S. Kogure, Y. Noguchi, K. Yamashita, T. Konishi and Y. Itoh (2016). GUI-Based Environment to Support Writing and Debugging Rules for a Program Visualization Tool, Proceedings of ICCE2016.

⑤ 藤岡僚太, 小暮悟, 野口靖造, 山下浩一, 小西達裕, 伊東幸宏 (2016). プログラムの挙動を教師の説明意図に基づいて視覚化する学習支援環境の構築と実践授業における活用”, 電子情報通

信学会研究報告.

- ⑥ R. Yamamoto, Y. Noguchi, S. Kogure, K. Yamashita, T. Konishi and Y. Itoh (2015). Construction of an environment to support learning systematic debugging process with Worksheets and Synchronized observation tool, Proceedings of ICCE2015.
- ⑦ K. Yamashita, R. Fujioka, S. Kogure, Y. Noguchi, T. Konishi and Y. Itoh (2015). Educational Practice of Algorithm using Learning Support System with Visualization of Program Behavior, Proceedings of ICCE2015.

[図書] (計0件)

[産業財産権]

○出願状況 (計0件)

○取得状況 (計0件)

[その他]

ホームページ等

6. 研究組織

(1) 研究代表者

野口 靖浩 (NOGUCHI, Yasuhiro)

静岡大学・情報学部・講師

研究者番号：50536919

(2) 研究分担者 なし

(3) 連携研究者 なし

(4) 研究協力者

()