

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

An Authoring Tool for Educational Materials to Visualize Behavior of Programs with a Target World

藤岡 僚太^{*1}, 小暮 悟^{*1}, 野口 靖浩^{*2}, 山下 浩一^{*3}, 小西 達裕^{*1}, 伊東 幸宏^{*4}
 Ryota FUJIOKA^{*1}, Satoru KOGURE^{*1}, Yasuhiro NOGUCHI^{*2}, Koichi YAMASHITA^{*3}, Tatsuhiro KONISHI^{*1}, Yukihiro ITOH^{*4}
^{*1}静岡大学大学院情報学研究科 ^{*2}静岡大学情報学部
^{*1}Graduate School of Informatics, Shizuoka University ^{*2}Faculty of Informatics, Shizuoka University
^{*3}浜松大学ビジネスデザイン学部 ^{*4}静岡大学
^{*3}Faculty of Business Design, Hamamatsu University ^{*4}Shizuoka University
 Email: gs14036@s.inf.shizuoka.ac.jp

あらまし：これまでにプログラムの挙動を視覚化するツールが多く開発されているが、教材作成者の説明意図を対象世界の見え方に反映させることができるものは少ない。そこで本稿では、対象世界の見え方を教材作成者が定義できるオーサリングツールと、その定義に基づきプログラムの挙動を視覚化するシステムを構築した。また、開発したオーサリングツールの使用の難易度や所要時間を評価する実験を行った。
 キーワード：プログラミング学習支援, アルゴリズム学習支援, 教材作成支援, 視覚化

1. 研究の背景・目的

アルゴリズム・プログラムの理解において、挙動をイメージすることは重要である。しかし初学者が挙動を正確にイメージすることは難しい。そのためこれまでに Jeliot⁽¹⁾, PROVIT⁽²⁾などプログラムの挙動の視覚化ツールが数多く開発されている。これらの既存ツールはプログラムの挙動を一定の方法で視覚化する。しかし、プログラミング教育の現場で教師が用いる図は説明対象ごとに最適な視覚化方法（木構造の様な抽象データ型の構造の視覚化、処理済み範囲の明確化、処理内容に対するコメントなど）で描かれている。これを教材作成者の説明意図と呼ぶことにする。本研究ではこの説明意図に基づいてプログラムの挙動を視覚化するツール、および説明意図を記述できるオーサリングツールを開発した。

多くの既存ツールではプログラミング言語が提供するデータ構造を直接視覚化する。これに対し我々のツールでは上述の説明意図を反映させた図で視覚化する。我々はそれぞれを実装ビュー、概念ビューと呼ぶ。実装ビューはプログラムのトレース結果に、概念ビューは座学で理解したアルゴリズムの挙動に対応付けやすい。しかし初学者はしばしば両ビューの対応関係の把握に困難を感じる。そこで我々のツールは、実装ビューと概念ビューの両方でプログラムの挙動を提示し、両者の対応関係も視覚化する。

概念ビューを提示する教材を作成する方法として、対象のプログラムから作成する場合、対象プログラムに描画命令を埋め込むなどの煩雑な作業が必要となる。プログラムの挙動を示すスライドや動画として教材を作成することも考えられるが、処理対象データが固定されるという大きな欠点がある。

そこで、我々は説明意図を記述するルールの仕様をデザインし、教材作成者が比較的簡便に概念ビューの見え方を定義できる手法を開発した。

2. 基礎的考察

2.1 要求仕様

開発するシステムの要求仕様は以下の通りである。

- (1) ソースコードからプログラムの挙動を実装ビュー、概念ビュー上で再現できる
- (2) 教材作成者が概念ビューの見え方をルール形式で指定できる

先行システム⁽³⁾においてプログラミング・アルゴリズム学習を支援するために、ソースコード、対象世界、操作系列の対応関係を視覚化する学習支援環境が構築されている。(1)はこのシステムの機能に含まれているためこれを流用する(2.3にて詳述)。

(2)の要求を満たすために概念ビューの見え方の指定に必要な情報を整理すると以下のようになる。

- A. 概念ビュー上にある物（オブジェクト）
- B. オブジェクトの属性に応じた見え方
- C. プログラム実行に伴う概念ビューの変化

A, B を定義するために、静岡大学情報学部のプログラミング、アルゴリズムを対象とする授業の教材からアルゴリズムの説明時に用いられる図の事例を調査し必要なオブジェクトと属性を決定した。結果を表1に示す。特殊なオブジェクトとしてオブジェクト同士の関係を表すものがある。例えばオブジェクト `table` は他のオブジェクトの位置関係を表す。

同記事例研究から C としてはオブジェクトの生成、削除、属性変更を記述できればよいことが分かった。

表 1 必要なオブジェクトと属性

オブジェクトの種類	属性
square(正方形), circle(丸), rectangle(長方形)	対応する変数名, 表示座標, 線の色, 背景色, 文字色
label(文字列)	表示座標, 文字列, 文字色, 表示位置
connector(結線)	繋げるオブジェクト, 線の幅, 線の色
table	表示座標, 属するオブジェクト, オブジェクトのレイアウト
balloon(吹き出し)	対象のオブジェクト, 文字列, 吹き出しの種類
line(線)	座標, 線の色, 線の幅, 線の長さ, 線の角度

2.2 説明意図記述ルール

説明意図記述ルールは以下の形式とする。

[条件][命令][オブジェクト ID]
[オブジェクトの種類][属性]

[命令]は変化を記述するものであり create (生成), delete (削除), update (属性の変更) の 3 種類がある。[条件]は[命令]を実行する条件を記述するものであり、「特定のステートメントが実行された時」、「特定の変数が指定の範囲を取る時」の二つを条件として記述できる。[属性]は表 1 に示した属性の値を記述できる。記述例を以下に示す。

state==3 create OBJ1 circle low X1 Y1 black white black

このルールにより ID が 3 であるステートメントが実行された時、OBJ1 という circle 型のオブジェクトが生成される。OBJ1 の「対応する変数」属性に low という変数名が指定されているので、変数 low の値が OBJ1 の中に表示される。それ以降のパラメータは表示座標、線の色、背景色、文字色を示す。

2.3 概念ビューの生成

概念ビューの生成は以下の方法で行う。まず先行システムのソースコード解析機能を拡張したものをを用いて、ソースコードのステートメントに ID を割り当てるとともに、プログラムの実行ステップ列を生成し、ステップごとの変数の値を抽出する。ステートメント ID はルールの条件記述に用いる。実行ステップごとに表示すべきオブジェクトやその変化をルールから決定し、描画データを作成する。以上より得た変数の値と描画データを基にプログラムの挙動に応じた概念ビューを生成する。生成した概念ビューの例を図 1 に示す。

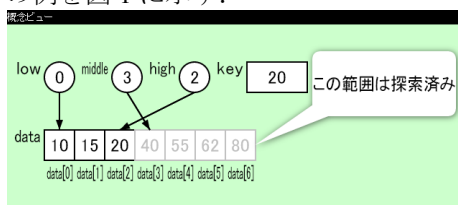


図 1 概念ビュー

3. プログラム挙動視覚化システム

図 2 はプログラム挙動視覚化システムのユーザインタフェースである。<1>はソースコードを表示する領域であり、現在注目しているステートメントが赤色で強調表示される。<2>には再現するステップの進行、後退などの命令ボタンが置かれる。<3>は実装ビューである。<4>は概念ビューである。

4. 評価実験

構築したシステムを用いた教材作成において作業の難易度や時間コストが現実的であるか評価する予備実験を行った。被験者はプログラミング授業における TA 相当の能力を持つ大学院生 1 名 (被験者 a),

大学生 2 名 (被験者 b, c) である。被験者は十分に教材作成に馴れた状態になるまでチュートリアルを受け、選択ソートの概念ビューの説明意図を表現する説明意図記述ルールを記述した。また、対比の為に同等の内容を説明するパワーポイントスライドを作成した。被験者全員がシステムを使用して教材を作成出来た。作成に要した時間を表 2 に示す。

表 2 実験結果

	被験者 a	被験者 b	被験者 c
チュートリアル	56 分	43 分	52 分
システム	32 分	30 分	33 分
パワーポイント	23 分	33 分	29 分

実験結果からシステムを利用することに充分馴れば十分実用的な時間で教材を作成できることが分かる。所要時間はパワーポイントによる教材作成と概ね同程度であるが、システムで作成した教材は処理対象データを変更した場合でもルールを変更せずにプログラムの挙動を再現できるためパワーポイント教材と比較して質が高い教材と言える。

5. むすび

現在のシステムは多重ポイントを含むプログラムに対応していない。対応する文法事項の拡張が必要である。また、現時点ではプログラム挙動可視化システムの学習効果の評価を行っていない。これらは今後の課題である。

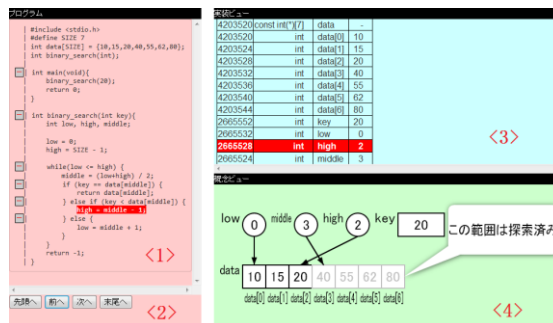


図 2 プログラム挙動可視化システム

参考文献

- (1)A.Moreno, N.Myller, E.Sutinen, and M.Ben-Ari: Visualizing programs with jeliot 3, AVI 04 Proceedings of the working conference on Advanced visual interfaces, pp.373-376 (2004).
- (2)松村和哉, 渡部治朗, 寺内俊, He Aiguo:PROVIT : ソフトウェア可視化手法を用いた初心者向け C 言語教育ツール, 電子情報通信学会技術研究報告. ET, 教育工学,109(268), pp.41-46 (2009).
- (3)S.Kogure, M.Okamoto, K.Yamashita, Y.Noguchi, T. Konishi, and Y.Itoh: Evaluation of an Algorithm and Programming Learning Support Environment based on Classroom Practices, Proceedings of ICCE2013, pp.418-424 (2013).