プログラム挙動視覚化ツール TEDViT における オブジェクトプロパティへの変数値写像機能の構築

Building a Variable-Value Mapping Function for Object Properties of Program Behavior Visualization Tool TEDViT

相馬 洸希[*1], 小暮 悟[*1], 野口 靖浩[*1], 山下 浩一[*2], 山本 頼弥[*3], 小西 達裕[*1], 伊東 幸宏[*1]
Hiroki SOMA[*1], Satoru KOGURE[*1], Yasuhiro NOGUCHI[*1], Koichi YAMASHITA[*2], Raiya YAMAMOTO[*3], Tatsuhiro KONISHI[*1], Yukihiro ITOH[*1]

[*1] 静岡大学 [*2] 常葉大学 [*3] 山陽小野田市立山口東京理科大学
[*1]Shizuoka University [*2]Tokoha University [*3]Sanyo-Onoda City University
Email: soma.hiroki.17@shizuoka.ac.jp

あらまし：我々はこれまでに，教材作成者の説明意図を対象世界の視覚化に反映できるプログラム挙動視覚化システム TEDViT を開発している．本システムは，プログラムの挙動による変数の変化をオブジェクト上の数値に動的に反映できる．しかし変数値をオブジェクトの位置や大きさ，色に対応付けることが非常に困難であった．本研究では，変数値をオブジェクトプロパティに動的に写像する機能を TEDViT に実装した．被験者 10 人による評価実験を行ったので報告する．
キーワード：プログラミング学習支援，プログラム挙動視覚化

1. 背景と目的

アルゴリズム・プログラムを理解する際に，その挙動をイメージできることは重要だとされており，実際，プログラムの挙動を視覚化するシステムが多く開発されている[(1)(2)]．しかし，これらのシステムでは視覚化の方法は開発者が決めた一定の方法であり，教師が視覚化の方法を意図して変更できない．

この問題を解決するため，C 言語プログラムを対象とした挙動視覚化ツール TEDViT[(3)] が開発されている．TEDViT では，教師が挙動の視覚化方法をルールで指定することで，変数に対応付けられたオブジェクトが持つプロパティを設定でき，教師の説明意図に即した描画ができる．オブジェクトに表示される数値や文字は，変数の値が変化すると自動で更新される．これを写像機能と呼ぶことにする．

前述の通り，TEDViT で用いるオブジェクトには視覚化ルール上として教師が指定できるプロパティ（描画色，表示位置など）がある．しかし，TEDViT には，以下の問題点がある．

問題点 1：教師が大きさを指定できるオブジェクトがない

問題点 2：変数の値に応じてオブジェクトのプロパティを変更することが難しい

1 つ目の問題点を解決するために，大きさを指定できるオブジェクトを新規に作成し，教師がルールで生成できるようにする．2 つ目の問題としては，変数の値を条件として個別にルールを作れば記述はできるものの，変数の値に応じてオブジェクトの位置や色を自動で動的に変更することはできなかった．そのため，変数値をオブジェクトの大きさや色，位置などのプロパティに動的に写像できる機構を構築し，教師が視覚化ルールで利用できるようにする．これによって，より詳細に視覚化できるようになり，教師が意図した描画を再現することができる幅が広がる．そして，新しいルールを利用することで，プロパティの変化におけるルールの作成時間を短縮する．この 2 点を実施することが本研究の目的である．

2. 先行研究と関連研究

2.1 挙動視覚化システム TEDViT[(3)]

TEDViT は，ソースコードを表示させるソースコード部，プログラムのメモリイメージを表示させる実装ビュー，プログラムの挙動を視覚化する概念ビューからなる．また，プログラム中の各変数の実行時点の値を自動的にオブジェクト内に表示できる．その際，描画・強調やメッセージなどの表示を行うためには，視覚化ルールを記述する必要がある．視覚化ルールの具体例を以下に示す．

```
rule,j,state==1,create,obj_i,square,
i,x1,y0,blue,white,black
```

この例では，番号 1 のステートメントが実行された際に，座標(x1,y0) に変数 i に対応するオブジェクト obj_i を生成する．このオブジェクトは正方形(square)型であり，輪郭線が青色(blue)，背景が白(white)，文字列が黒色(black)となる．

3. TEDViT における変数値のオブジェクトプロパティへの写像機能の構築

3.1 新規オブジェクトの作成

TEDViT には教師側が大きさを指定できるオブジェクトがない．視覚化に大きさを活用するアルゴリズムの例としては，限られた容量の中で最大の価値

となるものを選択するナップザック問題などが挙げられる．容量の視覚化には大きさを任意に変更できる表現が必要となる．そこで大きさを指定できる新しいオブジェクトを作成することで問題点 1 を解決する．既存のオブジェクトの種類にある矩形のオブジェクトを元として，横，および縦に伸縮する 2 種類のオブジェクトと縦と横に伸縮するオブジェクトの合計 3 種類を作成した．

3.2 変数値を参照する機能の実装

TEDViT では，変数値をプロパティに動的に写像できない．もし変数値の変化に応じてプロパティを変化させようとすると，変化の数だけ追加でルールを記述する必要がある．特定の数値や変数値をプロパティに写像することで視覚化が可能になるものとして，問題点 1 で取り上げたナップサック問題やソートがある．他の視覚化ツール[4][5]では，棒グラフを変化させることでソートの過程を視覚化している．これは変数値によってオブジェクトの「大きさ」を変化させることで表現している．そこで，既存のオブジェクトと問題点 1 で作成したオブジェクトのプロパティに変数値や数値を写像することで，プロパティ値を変数値に自動追随可能にした．これにより，ルールの記述コストを格段に削減し，問題点 2 を解決する．本研究で取り扱う写像機能の対象プロパティは，「大きさ」「色」「座標」である．

【大きさ】: 問題点 1 の解決策として，大きさを指定できるオブジェクトを作成した．このオブジェクトを利用することで，視覚化している変数の値に応じて大きさを動的に変化させられる．またルールに縦と横の変化倍率を数値で指定する項目を追加し，それにより倍率を適用したオブジェクトを描画できるようにした．

【色】: TEDViT のルールでは，オブジェクトの背景色・枠線の色，変数値の文字色を指定することができる．既存の色は HTML のカラーコードを記述することで指定できるが，これに変数値を参照して，色を指定できる機能を 2 つ追加した．1 つ目は，変数値を色のパラメータに一次変換する機能．2 つ目は，条件式によって色を離散的に変化させるものである．条件式と色を指定することで，条件式が真となるときに指定した色に変化するようになっている．現状では条件式は 2 つまで指定することができ，条件式の中に変数値を使用できるように実装している．

【座標】: TEDViT では，x,y 座標系を利用してオブジェクトの位置を指定できる．基準となる座標，参照する変数名，倍率を指定することで基準となる座標から変数値に倍率を掛けた値分だけ座標が移動する機能を実装した．

4. 評価実験

問題点 1 は解決できていることが自明であるため，評価実験を行わないこととする．問題点 2 を解決したことにより，視覚化ルールの記述時間を従来システムより削減できていることを確認するため，ルール作成者を対象とした評価実験を実施した．被験者はプログラミングの講義を担当している大学教員 2 名と TA 経験者の理系大学生 4 名，TA 経験者と同等のプログラミング能力のある理系大学生 4 名の計 10 名である．評価実験では，被験者にルールの定義等を説明後，提示した手本と同じ動作をするように既存ルールと新しいルールの療法を記述してもらい，完成までにかかる時間を測定した．課題として「色」と「座標」を変数の値に従って変化させる視覚化を新旧ルールを利用してそれぞれ 2 問，計 4 問実施した．実験結果を表 1 に示す．

表 1 実験データ

課題分類	色		座標	
使用ルール	旧	新	旧	新
旧→新(mm:ss)	40:35	12:00	31:56	9:22
新→旧(mm:ss)	29:57	25:06	22:32	15:10
平均(mm:ss)	35:16	18:33	27:14	12:16
比率[新/旧](%)	52.6		45.1	

表 1 より，旧ルールと新ルールの作成時間を比較すると，新ルールの方が短くなっていることがわかる．また，実験後のアンケートでも，色と座標それぞれに対して新ルールの方が書きやすいという意見が多かった．この 2 点から，新ルールによって視覚化ルールの作成コストが削減できたと推測できる．

5. まとめ

変数値をオブジェクトのプロパティに動的に写像できるように TEDViT の機能を拡張した．これにより，描画能力の向上と視覚化ルールの作成コストを削減することができた．一方で，変数値を写像したことによる学習効果の測定や写像機能を利用した新たな描画方法の検討が必要である．また，新しい視覚化ルールの記述方法にも改善の余地がある．

参考文献

(1) Moreno, A., Myller, N., Sutinen, E.:"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 他: “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, 教育工学, Vol. 109, No. 268, pp.41-46, (2009).
(3) Yamashita, K. et al.:“Classroom Practice for Understanding Pointers using Learning Support System for Visualizing Memory Image and Target Domain World”, Research and Practice in Technology Enhanced Learning (RPTEL), Vol. 12, No. 17, pp.1-16 (2017).
(4) VisuAlgo (最終閲覧日:2021/5/22) https://visualgo.net/ja
(5) AlgorithmVisualizer (最終閲覧日:2021/5/22) https://algorithm-visualizer.org/