プログラミングにおけるポインタ操作を通した関数学習機能

吉田　英樹†　　金子　敬一‡

東京農工大学大学院工学府†　　東京農工大学大学院共生科学技術研究院‡

1. 序論

一般に，プログラミング初学者は，ポインタ操作をコーディングすることが苦手である[1]．彼らは，どのような操作を行うべきか抽象レベルで理解していても，それに対応するコードを生成することができない．通常の可視化によるプログラミング学習システムは，正しいコードの実行過程を可視化する[2]．しかし，実行過程から正しいコードを提示することはできない．Miura ら[3]は，プログラミング言語 C のサブセットである SmallC を対象に，学習者が可視化されたオブジェクトを直接操作することができる可視化教育システム VIE (Visual Interactive Environment) を実現した．VIE システムは，オブジェクトへの操作に対応するコードの候補リストを提示することができ，学習者はその中から意図したものを選択し，プログラミングを継続することができる．実験の結果，VIE が学習効果を持つことが確認されている．

残念ながら SmallC には関数呼出しの機能がないため，学習者は，ポインタと値渡しの関係を理解することができなかった．そのため，本研究では，VIE システムが関数呼出しを扱うことができるように拡張することを目的とする．

2. VIE システム

VIE は，ソースコード編集領域，可視化領域，メッセージ領域および制御ボタン群で構成される単一のウィンドウからなる(図 1 参照)．学習者は，ソースコード編集領域でソースコードを編集し，制御ボタンの「コンパイル」ボタンを押下して，コンパイルする．コンパイルに成功すると，学習者は，制御ボタンの「ステップ実行」ボタンを使って，1 文ずつ実行しつつ，その実行結果を可視化領域で確認することができる．「ステップ実行」ボタンの効果をキャンセルするための「一つ戻る」ボタンも制御ボタンに含

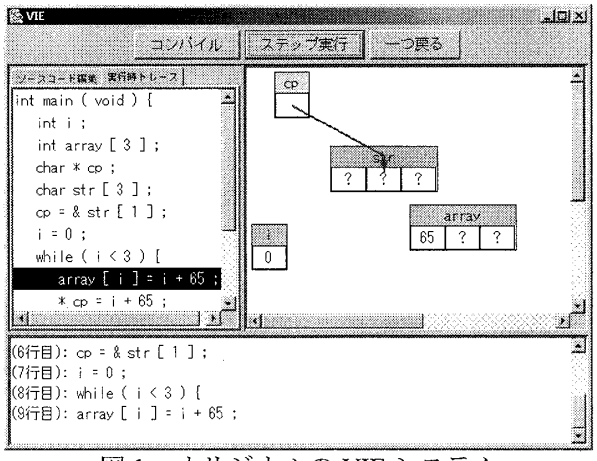

図 1　オリジナルの VIE システム

まれる．可視化領域には，変数とポインタが出現する．変数は矩形のオブジェクトとして出現し，型により色が変化し，変数名，および保持する値が表示される．変数の値が不定の場合は，'?'を表示する．ポインタは，矢印によって表現される．

VIE では，学習者が可視化領域のオブジェクトを操作することができる．許される操作は，変数オブジェクトの移動，変数への値の代入，ポインタの移動である．VIE は，学習者が行ったオブジェクトへの操作に対応するコード候補をメッセージ領域に表示する．このため，学習者が，抽象レベルで理解している操作に対するコードを生成することができない場合，この機能によりコード生成のヒントを得ることができるようになる．

また，図 1 に示す版に拡張を加え，スクラッチからプログラムを記述することができない学習者のために，いくつかのサンプルプログラムも用意されている．具体的には，最新版では制御ボタン群の右に追加されたプルダウンメニューから，サンプルプログラムを読み込んで使用することができるようになっている．

3. VIE の拡張

VIE が対象とする SmallC には，関数呼出しの機能がない．そのため，言語 C の初心者にとってわかりづらいとされる「値渡し」の概念や，

A Learning System for Function Calls and Pointer Operations in Programming
† Graduate School of Eng., Tokyo Univ. of Agric. & Tech.
‡ Institute of Symbiotic Sci. & Tech., Tokyo Univ. of Agric. & Tech.

変数の有効範囲「スコープ」についての学習を支援することができなかった．また，これらの概念はポインタ操作との関連が深い．そこで本研究では，SmallC および VIE を拡張し，関数定義や関数呼出しを扱うことを可能にした．

3.1 関数の視覚表現

図 2 に拡張した VIE を用いて，関数を可視化表現した様子を示す．関数は矩形で表現する．プログラムのステップ実行が関数内に移ると，可視化領域に矩形を出現させ，その内部に関数名を表示する(今後，関数を可視化表現した矩形を関数オブジェクトとよぶ)．関数内で宣言されたローカル変数および引数のオブジェクトは関数オブジェクト内に配置し，その外にはみ出して表示することはできない．また，関数の終了時には関数オブジェクトおよびその内部のオブジェクトを可視化領域から消去する．

関数が入れ子構造になっている場合は，関数をあらわす複数の関数を互いに重ならないように表示する．さらに，再帰関数など同名の関数を複数表示しなければならない場合は、関数名の後に添え字をつけ，矩形内部の塗りつぶし色を異なる種類のものにすることで各実体を区別する．

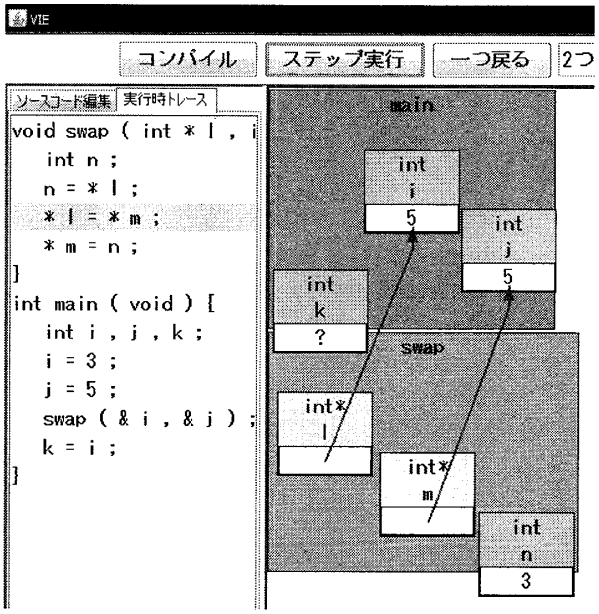

図 2　拡張版 VIE による関数の可視化

3.2 スコープルールの表現

言語 C の初心者は，変数のスコープルールについての理解が不十分なため，すでに解放された自動変数や，文脈から参照できない変数にアクセスする誤りをおかしやすい．

このような誤りに対処するため，拡張 VIE では，可視化領域に対する操作を，スコープルールに基づいて限定する．学習者が，実行中の文脈からはアクセスできない変数に対して操作を行った場合，図 3 のようにダイアログを表示して禁止された操作であることを提示する．

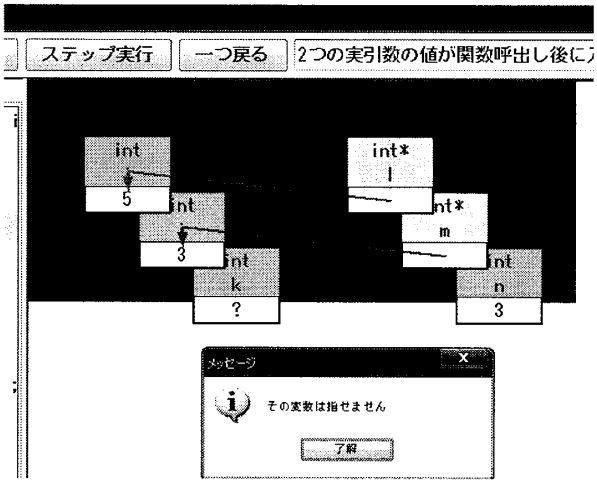

図 3　スコープルールにより禁止された操作

4. まとめ

本研究では，可視化されたオブジェクトを直接操作することができる可視化教育システム VIE を拡張して，関数呼出しが扱えるようにした．拡張版 VIE では，関数のスコープに対応する矩形を表示し，学習者が，スコープルールを破る操作を試みるとダイアログで警告し，その操作をキャンセルする．拡張の結果，学習者は，関数呼出しとポインタ操作の関係を学習することが可能となった．

拡張版 VIE を用いて関数呼び出しとポインタ操作に関する演習を実施し，その教育効果を測定することが今後の課題である．また，システムをさらに拡張して，配列だけでなく，構造体を扱うことができるようにすることも検討している．

参考文献

1) 朝井 淳: ポインタが理解できない理由, 技術評論社, 2002.

2) M. Kolling, B. Quig, A. Patterson and J. Rosenberg: "The BlueJ System and its Pedagogy," J. Computer Science Education, Vol. 13, No. 4, 249–268, 2003.

3) Y. Miura, T. Suzuki, and K. Kaneko: "A Visualized Educational System for Pointer Operations in Programming," Proc. 5th Int'l Conf. Web-based Education, pp. 131−136, 2006.