

## メモリ空間上におけるポインタの挙動を視覚化する学習支援システム

Learning Support System Visualizing the Pointer's Behavior  
in Memory Space

坂田 佳佑<sup>\*1</sup>, 野口 靖浩<sup>\*1</sup>, 山下 浩一<sup>\*2</sup>, 小暮 悟<sup>\*1</sup>, 小西 達裕<sup>\*1</sup>, 伊東 幸宏<sup>\*3</sup>  
 Keisuke SAKATA<sup>\*1</sup>, Yasuhiro NOGUCHI<sup>\*1</sup>, Koichi YAMASHITA<sup>\*2</sup>, Satoru KOGURE<sup>\*1</sup>  
 Tatsuhiro KONISHI<sup>\*1</sup>, Yukihiro ITOH<sup>\*3</sup>

<sup>\*1</sup>静岡大学情報学部, <sup>\*2</sup>常葉大学経営学部

<sup>\*1</sup>Faculty of Informatics, Shizuoka University, <sup>\*2</sup>Faculty of Business Administration, Tokoha University

<sup>\*3</sup>静岡大学

<sup>\*3</sup>Shizuoka University

Email: cs16035@s.inf.shizuoka.ac.jp

あらまし: C 言語には初学者が躓きやすい概念であるポインタが存在する。ポインタを学習するにあたってメモリ空間上の情報を正しくイメージすることが重要である。メモリ空間上の情報を表示する視覚化ツールは開発されているが、初学者がそのようなツールをうまく活用するのは難しい。そこでメモリ空間上の情報を表示するツールに教師の意図を反映させた強調やメッセージを表示させる機構を開発した。複数の評価実験を実施し、システムの有効性を示す結果を得たので報告する。

キーワード: 学習支援システム, ポインタ, メモリ空間, C 言語

## 1. 研究背景と目的

C 言語には学習するにあたって避けて通れず、初学習者が躓きやすい概念であるポインタが存在する。ポインタを正しく理解する鍵は、C プログラムにおいてメモリがどのように構成され、管理されているかを理解し、イメージすることにある。

またプログラミングを学習する際に、プログラムの挙動をイメージすることは重要であるがプログラミング初学者はプログラムの挙動を正しくイメージすることは難しい。この問題に対して多くのプログラム視覚化ツールが開発されている<sup>(1)(2)(3)(4)(5)</sup>。

メモリ空間上の情報を視覚化するツールとして GDB や TEDViT<sup>(3)(4)(5)</sup>の実装ビューなどがある。しかし初学習者は GDB で表示される内容を理解することは難しく、TEDViT の実装ビューではメモリ空間上の情報をただ表示するだけになってしまっている。TEDViT では教師が視覚化ルールを記述することでポインタを概念ビュー上で表現することもできるが肝心のメモリ空間（実装ビュー）での表現が行えていない。よってメモリ空間（実装ビュー）上に教師の意図を反映させた学習に効果的な描画やメッセージに表示を行うことで初学習者の学習による躓きを軽減することができるのではないかと考えた。この考えを実現するにあたり以下のような問題が生じる。

- ▶ 問題 A 実装ビュー上に描画ができない
- ▶ 問題 B 実装ビュー描画の視覚化ルールがない

本研究ではこの2つの問題点についての解決策を模索し、システムの拡張を行うことで、教育者の負担が少なく、メモリ空間（実装ビュー）上に描画を行うことができる拡張 TEDViT の作成を行っていく。

## 2. 先行研究

TEDViT<sup>(3)(4)(5)</sup>はC言語のプログラム挙動視覚化ツールである。TEDViT のメイン画面は大きく分けて4部分に分けられる。ソースコードを表示するソースコード部、詳細な動作を表示する詳細表示部、メモリ空間上の情報を表示する実装ビュー部、プログラムの挙動を視覚化する概念ビュー部からなる。

開発されているツールの多くはツール開発者が決めた一定のルールで視覚化しているため教師の意図を反映させた可視化をすることができない。しかしTEDViT ではこの問題を解決するため教育者が作成したプログラムについて可視化を行うことができる。教育者は可視化ルールと呼ばれるルールを記述することにより、概念ビューに可視化を行うことができる。学習者はTEDViTを使用することでプログラムの挙動を1ステップずつ追うことが可能である。

## 3. 問題の改善策

前述した問題 A, 問題 B について改善案を示す。

### 3.1 問題 A について

実装ビューに描画を行う機能を追加していく。教科書や実際のプログラミング講義の講義資料や演習課題を参考に教師意図を反映させるために最低限必要な4つの機能を絞り込んだ。

4つの描画機能を以下に示す。

- ① ポインタの強調表示  
強調を行いたいポインタ変数を指定することで描画を行う。指定した変数が格納している値が示す番地に矢印を伸ばし囲い強調表示する。
- ② 特定のセルの強調表示  
指定したセル内の値を囲み強調表示するほか文字色、背景色、文字サイズを変更ができる。

- ③ 特定の行要素の強調表示  
行全体を枠で囲むことで強調表示を行う。  
強調したい行の変数を指定すると強調される。
- ④ メッセージの表示  
実装ビューにメッセージを表示する。  
メッセージの表示場所の指定は特定のセルに対して絶対座標と相対座標で指定ができる。



図1 システム概観

### 3.2 問題 B について

問題 B は問題 A の描画機能に必要な情報を教師が指定すべき情報か、指定すべきではない情報かを分け設定することで新たなルールを作成する。

例 : `emprule, state==16, change, header, pointer, , 3, , ,`

上記ルール例ではステートメント番号が 16 の時ポインタ変数 `header` を強調表示する。図 1 内の<1>を描画するルールが上記の視覚化ルールである。

## 4. 評価実験

問題 A に対する評価実験を実験 I、問題 B に対するものを実験 II とする。それぞれの実験では「拡張後の TEDViT を使用することで C 言語のポインタの理解が促進される」という仮説と「教師が現実的な時間で実装ビューの視覚化ルールが記述できる」という仮説を検証する。実験においてはシステムの学習効果と操作性の評価を行う。

### 4.1 実験 I

実験 I は計 2 回行った。1 度目の評価実験の被験者は静岡大学情報学部生 90 名である。学習状況が Java 言語は学習済みであり、C 言語は未学習である。実験の流れは C 言語の授業、プレテスト、システムを用いた演習、ポストテストの順である。テストに関しては以下の 3 種類のポインタに関する理解度を図る問題を出題した。テストの結果（正答率）を表 1 に示す。また表中の時間はシステム使用時間である。

表 1 実験 I（静岡大学情報学部）の結果

学習内容（時間）	プレ	ポスト
基本型のポインタ（10 分）	56.2%	84.4%
配列型のポインタ（10 分）	23.6%	69.2%
構造体型のポインタ（10 分）	38.2%	78.8%

2 度目の評価実験の被験者は静岡大学工学部生 40 名である。学習状況としては授業で C 言語を半年間学習済みである。実験の基本的な流れとしては 1 度目の評価実験と同じである。テストは以下の 3 種類のポインタに関する理解度を図る問題を出題した。テストの結果（正答率）を表 2 に示す。

表 2 実験 I（静岡大学工学部）の結果

学習内容（時間）	プレ	ポスト
基本型のポインタ（10 分）	25.6%	58.1%
ポインタ渡し（20 分）	53.5%	81.4%
構造体型のポインタ（10 分）	16.5%	27.9%

両実験の結果、システムの使用時間が僅か 10 分、20 分程度に対し、すべてのテストで正答率が上がっている。以上によりシステムを使用することで C 言語のポインタに対する理解が促進されたといえる。

### 4.2 実験 II

実験 II の被験者はプログラミング講義担当の教員、TA 経験がある、またはそれに相当するプログラミングの技能がある学生、計 8 名である。問題 2 間に関してルール記述を依頼し、想定している描画を行えているか評価を行い、ルール記述にかかった時間を測定する。問題 1. 配列でのポインタの使用、問題 2. ポインタの関数渡し、の 2 間を利用した。問題 1 に関しては平均 30:18、問題 2 は平均 30:47 であった。

山下ら<sup>6)</sup>が行った概念ビュー描画のルール記述を行う評価実験でのルール記述時間は平均 32 分であり授業準備という観点でも現実的な時間であった。今回追加した実装ビュー描画のルール記述時間は約 30 分程度であり、もし概念ビューと実装ビューの描画ルール作成を同時に行うとするならば、合計時間は 1 時間程度である。授業で TEDViT を使用している教員方からは、許容できる時間であると回答いただいた。実装ビュー視覚化のルールは現実的な時間で記述ができることがわかった。

## 5. むすび

本研究は、実装ビュー上に描画を行いポインタの学習に役立つような TEDViT の開発を行った。評価の結果から描画が C 言語のポインタの学習に効果がありルール記述の時間が現実的であることを確認した。

### 参考文献

- (1) A. Moreno et al. "Visualizing programs with jeliot 3," AVI 04: Proceedings of the working conference on Advanced visual interfaces, pp.373-376, May, 2004.
- (2) 松村和哉ほか : 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, 109(268), pp.41-46, Nov, 2009.
- (3) K.Yamashita et al. "Learning Support System for Visualizing Memory Image and Target Domain World, and Classroom Practice for Understanding Pointers," Proceedings of ICCE2016, pp.521-530, Nov, 2016.
- (4) K.Yamashita et al. "A Learning Support System for Visualizing Behaviors of Students' Programs Based on Teachers' Intents of Instruction," Proceedings of ICCE2018, pp.761-766, Nov, 2018.
- (5) K.Yamashita et al. "Extending Program Visualization System Based on Teacher's Intent of Instruction to Support Learning Dynamic Data Structures," Proceedings of ICCE2019, pp.354-356, Dec, 2019.