

ペンインタフェースを用いた視覚的な Lisp 教育環境

原 謙治* 五十嵐 健夫**

*東京大学理学部情報科学科 **東京大学大学院情報理工学系研究科

**{hara2001, takeo}@ui.is.s.u-tokyo.ac.jp

1. はじめに

従来のテキストエディタにおける Lisp プログラムは多くの括弧を用いて書かれており、プログラムの木構造が見えにくいものとなっている。また、テキストエディタで編集している様子を見ても何をしているのかがわかりにくいという問題も存在する。我々はそうした問題を解決するためにペンインタフェースを用いた視覚的な Lisp プログラミング環境を開発した。このシステムにおいては、ユーザはジェスチャを用いて、可視化された木構造のノードに対して追加、削除、開閉操作を直感的に行うことができ、編集している人の意図の推測しやすさも期待される。

2. 関連研究

ペンを用いたプログラミング環境の研究としては Hyperflow[1]が挙げられる。Hyperflow はペンインタフェース、オブジェクト指向、データフロー型、といった特徴を持った教育用ビジュアルプログラミング言語である。また、ダイアグラム編集におけるペンとマウスの操作性の違いについての実験では、ペンの方がマウスより 2 倍早い操作性を得られるという結果が出ている[3]。ビジュアルプログラミング言語におけるダイアグラムの入力方法については、ペンのみとマウス&キーボードで実験している[2]。この実験においてはモード切替はジェスチャで行うか、入力デバイス選択で行うべきであると結論付けている。また、コピー&ペースト操作の重要性とジェスチャかポップアップメニューを用いることの有効性を主張している。本論文では Lisp 系言語 Scheme のビジュアルプログラミング環境における直感的なインタフェースを提案する。

3. 基本設計

本論文ではペンによるジェスチャを用いて直感的にプログラムを開発できるシステムを提案する。

このシステムではキーボードを用いないペンみでのプログラミングを可能にし、キーボードタイピングに不慣れなユーザにとって扱いやすさ、システムの使い方の覚えやすさ、プログラム編者の意図の推測しやすさを提供する。さらに Lisp プログラムの木構造の視覚的表現はプログラムの理解しやすさを提供している。

以下では木構造の編集におけるノード追加、ノード削除、文字入力、コピー&ペースト、ノードの開閉等のジェスチャによる操作方法を紹介する。

(1) ノード追加

ノードの追加はノードの外から中へ線を引くことで行う。ノードの中へ線を引くと、追加されるノードが青色で表示される。そこでペンを離すと赤色で表示されていたノードが追加される。(図 1)

(2) ノード削除

ノードの削除はノードを繋ぐ枝を横切る線を引くことで行う。枝を横切る線を引くと削除されるノードが赤色で表示される。そこでペンを離すと赤色で表示されていたノードが削除される。(図 2)

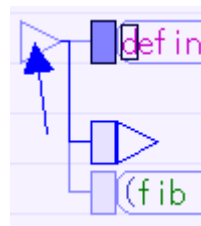


図 1. ノード追加

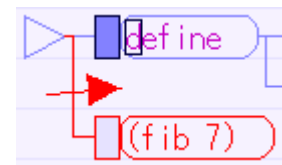


図 2. ノード削除

(3) ノードの開閉

ノードの開閉はノードの先端にあるヘッダ(ノードのすぐ左の部分)から左右にドラッグを開始することによって行う。ドラッグを開始すると縦に紺色の線が表示される。選択されているノード以下にあるノードのうち、その線より左側にあるノードは開かれ、右側にあるノードは閉じられる。(図

3)

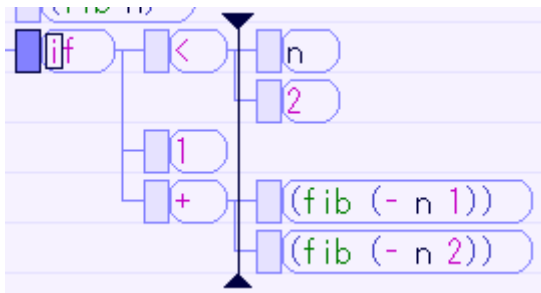


図 3.ノードの開閉

(4)文字入力

文字入力はノードをタップすることで現れる入力スペースに graffiti を用いて入力を行う。(図 4)

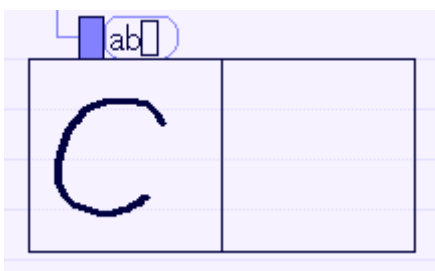


図 4.文字入力('c'を入力)

(5)コピー&ペースト

コピー&ペーストはノードからノードへのドラッグ&ドロップで行う。ノードの中からドラッグを開始するとそのノードの背景が暗く表示される。この状態で別のノードの上へドラッグすると、ポインタがドラッグ&ドロップ可能状態で表示される。この時ペンを離すとドラッグを終了したノードの名前がドラッグを開始したノードの名前になる(コピーされる)。(図 5)

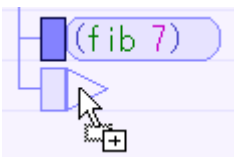


図 5.コピー&ペースト

4. 議論

従来のテキストエディタを用いたプログラミングと比較した時の本手法のメリットとしては 1)キーボードを用いないペンのみでのプログラミングが可能であること、2)プログラム構造が可視化された木構造で表示されているので直感的に理解しやすいこと、3)ペンで操作している様子を見てソフトウェアの使い方が覚えやすいこと、4)プログラムを編

集している様子を見て何をしようとしているかが分かりやすいこと、が挙げられる。

本論文で提案した手法は、プログラミング教育用のインターフェースとしての使用を想定している。従来のテキストエディタを用いたプログラミング教育においては、特にキーボードタイピングに慣れていないユーザは授業以外にキーボード入力も気にしなければならず、また、教師の書いたプログラムを写すだけでは深い理解が得にくいという問題もある。しかし、我々のシステムにおいてはキーボードタイピングに気をを使う必要はなく、教師のプログラム作成過程の分かりやすさ、作成されたプログラムの分かりやすさ、覚えやすさも期待できる。

本手法の問題点としては、編集速度の遅さが挙げられる。キーボード操作の場合、習熟度が増せば非常に高速なプログラム編集を行うことができる。しかし、ペン操作の場合熟練してもキーボードほどの高速な編集は行えない。授業中はこの問題はさして大きくないが、独習中は高速な編集速度が要求されると考えられる。我々はテキストエディタを用いた場合とあまり変わらないキータイプ数で扱えるようにシステムを設計したが、その操作方法をいつ覚えるかという問題も存在する。

今後はシステムの使いやすさ、プログラミングの分かりやすさについて評価実験を行い、それに基づいたさらなる改良を行っていきたいと考えている。

参考文献

- [1] Kimura, T.D. "Hyperflow: A visual Programming Language for Pen Computers," Proceedings of 1992 IEEE Work shop on Visual Languages, Seattle, Washington, September 1992, pp 12.5132.
- [2] Citrin, W., "Requirements for Graphical Front Ends for Visual Languages", IEEE Symposium on Visual Languages, pp. 142-150, Bergen, Norway, August 1993.
- [3] Apte, Ajay and Takayuki Dan Kimura. A Comparison Study of the Pen and the Mouse in Editing Graphic Diagrams. *Proceedings of Visual Languages 1993 Conference*, Bergen, Norway, 1993.