

プログラミング教育用言語処理系 NB2 の設計

4 L - 5

橋本裕[†] 早川栄一[†] 並木美太郎[†] 高橋延匡[‡]

[†]東京農工大学 [‡]拓殖大学

1. はじめに

プログラミング教育では、プログラミングできるようになることはもちろんのこと、学習した言語以外でも応用が可能なさまざまな概念を学習・習得することも重要である。例えば、J97[1] のプログラミング入門の項目でもそれは指摘されている。しかし、これらの諸概念は計算機固有の概念、抽象的な概念が多く存在している。そこで、筆者らは学習者がこのような概念を理解し、プログラムを組むことができるようにするため、プログラミング言語処理系にこのようなプログラミングに必要となる諸概念の学習支援機能を持つ言語処理系を設計している。本稿では主に、基本的な方針の他に、学習者が学習に専念できるようにするための BANB というウィンドウ管理のためのモデルの設計について述べる。

2. 対象

本処理系では大学1年程度の能力を持ったプログラミングの初心者を対象にする。学習対象は、プログラムを組む上で基本となる制御構造やデータ構造、スコープ、BNF やオーダ等である。学習にあたっては、初心者が扱いやすいといわれている言語である BASIC 言語を用いる。具体的には後述するプログラミング言語 NB2 を使用する。これらの学習のためにはプログラムを組む必要があるため、プログラムの編集や実行ができる環境を提供する。

3. プログラミング言語 NB2

プログラミングの入門者が使用するプログラミング言語には、いくつか考えられるが、本処理系では、筆者らが作成したプログラミング言語 NB2 という Full BASIC 言語 [2] を基礎とした BASIC 言語を利用する。

教育用の言語としては、アメリカでは PASCAL が有名だが、(1)日本では BASIC の方が知られており教育しやすい、(2)BASIC は行指向なので学習支援がやりやすい、というような理由から、構造化された制御構造が記述できる Full BASIC を強化した、新しい BASIC 言語を作成した。この BASIC では構造を持つデータ構造を扱うことができ、日本語の識別名を使用できるという特徴を持っている。

4. 設計にあたっての戦略と方向性

本処理系の設計にあたっての戦略と方向性は次に示す通りである。

(1) 編集フェーズと変換フェーズの統合

筆者らは以前 BNF の学習支援機能を持つプログラミング言語処理系 NB を発表した [3]。この処理系では、編集、変換、実行の各フェーズでそれぞれのフェーズ特有の学習支援を行うという方針をとっていた。さてここで、もし編集のフェーズと変換のフェーズが統合されていたらどうなるだろうか(図 4.1)。統合されているので、当然編集時にもプログラムを解析した情報を得られる。したがって、編集時にも、例えばこの変数のスコープはどこまでかといった情報を得ることができ、学習者は編集しているときにすぐにスコープの学習をすることができる。

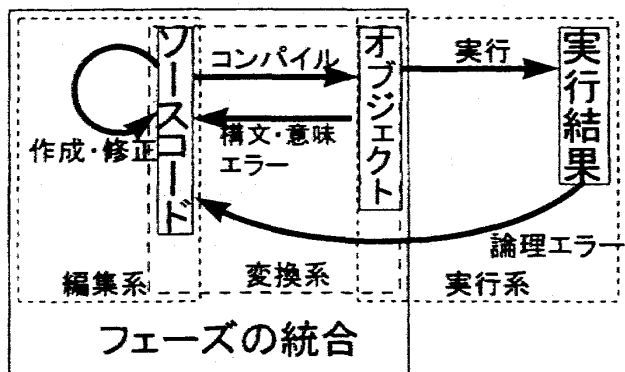


図 4.1 フェーズの統合

(2) 可視化

プログラミングには、計算機固有の概念や抽象的な概念が存在しこれを理解することが求められる。しかし、これらの概念は今までなかったような概念なので、すぐに

Design of a Language Processor 'NB2' for Programming Education

Yutaka Hasimoto[†], Eiichi Hayakawa[†], Mitarou Namiki[†] and Nobumasa Takahashi[‡]

[†]Tokyo University of Agriculture and Technology

[‡]Takushoku University

理解・定着することができないのは当然のことである。そこで、それらを可視化することにより、学習者がそれらを理解・定着しやすくする。

(3) 編集系と実行系の統合的インタフェース

編集時に実行時と同様な機能が提供されるのにも関わらず、実行時と編集時の実行支援機能のインタフェースが異なると、単に同じ機能に別々の使い方を覚えなければならず、学習の本質から離れてしまう。そこで、編集系と実行系では統合的なインタフェースを提供する。

5. 設計

前章で述べた方向と戦略から次のような設計をした。

5.1 ウィンドウの管理方法

利用者に示すべき情報は一つではない。例えば、プログラムを編集するための編集機構や実行機構、システムから利用者に知らせたいメッセージ、利用者がプログラムに関して書いておきたいメモ等、他にもいろいろな情報がある。これらの情報の表示にウィンドウを使用することになると、ウィンドウの管理が必要になる。しかし、単純に各情報を独立したウィンドウにするとその扱いが非常に面倒になる。ではそうかといって、一つのウィンドウにすべての情報を表示するようにすると使い勝手が悪くなる。また、一つのウィンドウで情報を切り替えて表示するようにすると複数の情報を同時に見ることが困難になる。そのような問題を解決するため、本処理系では、本とバインダのメタファを用いる BANB(Books ANd Binders)というモデルを用いてウィンドウを管理する(図 5.1.1)。本に記されたバインダを利用者は計算機から受け取り、利用者はそのバインダの内容を閲覧・編集することができる。ここで、バインダはソースプログラムや関連する情報を管理するプロジェクト、本はプロジェクトの管理を指す。実際のバイ

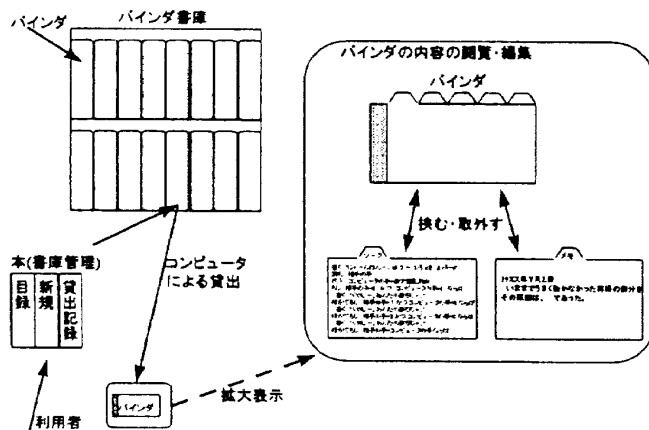


図 5.1.1 BANB モデル

ンダは、紙を自由に挟んだり取り出したりすることができる。このモデルでも同様に、バインダに挟まれた各種情報を自由に別のウィンドウにしたり、再び綴じたりすることができる。バインダにはタブシートがあるので、各種情報を簡単に参照することができる。これらの情報には、ソースプログラム、編集や実行といったページが含まれている。制御構造やデータ構造といった支援は、この編集と実行のページで行われる。

5.2 編集と実行のページ

編集と実行のページでは、基本的には前述した方針に基づき、同一の画面が提供される。これらのページで提供されるのは、ソースプログラム編集機構(編集ページ)、実行可視化機構(実行ページ)、BNF 可視化機構、ローカル変数可視化機構、現在実行位置表示(実行ページ)である。ソースプログラム編集機構/実行可視化機構では、おもに制御構造の可視化が行われる。データ構造は、ローカル変数可視化機構、編集機構と実行可視化機構において、変数は箱、定数は板というような表示を行うことで学習の支援を行う。BNF 可視化機構では、現在のソースプログラムに関する構文木を利用することで BNF 学習支援を行う。

6. おわりに

本稿で述べた方式により、学習者により学習に有益な情報を提供することができるようになり、また、学習者により学習に専念できるウィンドウの管理モデルを設計した。今後の課題は、処理系の実現やその評価である。

謝辞

本研究は、科学研究費基盤研究(A)(2)09358004 により行われた。

参考文献

- [1]大学の理工系学部情報系学科のためのコンピュータサイエンス教育カリキュラム J97, 情報処理学会, 1997.11
- [2]西村恕彦, 植村俊亮, 酒井俊夫, 高田正之編:アメリカ規格 Full BASIC -全訳と解説-, 共立出版, 1990.2
- [3]橋本裕, 早川栄一, 並木美太郎, 高橋延匡: BNFの学習を支援する Full BASIC 言語処理系, 情報処理学会コンピュータと教育研究会, Vol.39, No.3, pp.17-24, 1996