

4U-5 アプローチ指向の教育支援のための ビジュアル言語エディタとシミュレータの統合

中嶋 正夫^{*}, 阪井 和男^{**}, 加藤 浩^{***}, 西垣 通^{**}

^{*}明治大学情報科学センター, ^{**}明治大学法学部, ^{***}日本電気(株)C&C情報研究所

【 1 】 はじめに

我々が研究・開発している情報処理教育を支援する ITS (Intelligent Tutoring System) では、プログラマがある問題を解決する時に適用する様々なアプローチの組み合わせについて着目し、次に挙げるいくつかのアプローチの使い分けのノウハウを学習者への指導方略として採用する予定である¹⁾。

例を挙げると、具体的で詳細な知識を積み上げていく bottom-up アプローチや、処理を機能別にモジュールに分け徐々にブレークダウンしていく top-down アプローチ。そして、与えられた問題に対して問題解決の見通しがある場合には、難しい部分から解決していくという hardest-first アプローチと、見通しが立たない場合に易しい部分から解決していくという easiest-first アプローチである²⁾。また、アルゴリズムを時間的順序を追って構築する時間的アプローチと、プログラムの構成を構造的に把握する空間的アプローチ³⁾などである。そして、これらのアプローチはそれぞれ密接な関係があり、一般に問題解決する過程において、これらのアプローチを適宜に組み合わせ、また使い分けていると思われる。

そこで我々は、上に述べたような、問題解決の過程での様々なアプローチの使い分けについて研究するために、モジュールの階層関係を明示し、top-down、bottom-up のアプローチを柔軟に併用できるエディタとシミュレータの統合環境を構築している。今回は、そのエディタ部について報告する。

従来のプログラミング環境のほとんどでは、プログラマがイメージするアルゴリズムの抽象度に応じたプログラミングが困難であった。本システムによれば、学習者は抽象化のレベルを意識しつつプロトタイピングできる。また、エディタやシミュレータの操作の時間的過程を記録することによって、学習者が問題解決する過程の把握・分析も可能となる。本稿では、教育用プログラミング統合環境の設計方針と機能を明らかにし、初心者がこの環境を用いた時の問題解決の過程の分析について述べる。

【 2 】 システム概要

本システムは、研究中の教育支援システムのユーザインタフェース部分にあたる。このエディタ/シミュレータ上で、学習者はプログラムを作成し、実行させることができる。なお、このシステムは Macintosh の HyperCard 上に実現した(図1)。

このエディタ/シミュレータの特徴は、

- (1) 初心者馴染み易いビジュアル言語
 - (2) 部品 (従来の言語のコマンドやステートメントに相当) のモジュール化を明示的に表現
 - (3) モジュール間の階層構造を視覚的に表現
 - (4) 計算式や条件式を電卓のようなインタフェースで入力
 - (5) 部品間でやりとりされるデータを明確に表現
 - (6) 学習者のエディタ/シミュレータの操作経過を自動的に記録
- などの点である。

学習者は、画面左にある「Parts Box」や「データー一覧」にあるアイコンをマウスでワークエリア上に移動し、線で部品アイコンの実行順序を決定したり、部品アイコンにデーターアイコンを接続することによって、プログラムを構築する。なお、計算式や条件式は、電卓にアナロジーをとった入力インタフェースによって入力する(図2)。また、モジュールを示す部品をクリックすると、下位構造へ移動し、モジュールの内容が開示されて編集が可能になる。

ここで、部品やモジュールで使用する(M1)や(M2)などのデータは、入力データであるのか、出力データであるのかを部品ごとに定義しなくてはならない。これは学習者に、部品間でやり取りするデータを明確に意識させるためである。また、このデータは(M1)や(M2)といった抽象的な名前とは別に、学習者が自由に名前を付けられるため、データの具体的な意味付けができる。

このシステムは、学習者の一連の操作の履歴を時刻とともに記録できる。そして、このデータを分析することによって、例えば今どの部品に着目しているか、どこで迷いが生じているのか、top-down アプローチなのかそれとも bottom-up アプローチで解こうとしているのか、などといった学習者の意図をある程度くみ取ることができる。

Integration of Visual Language Editor and Simulator for Supporting Various Approaches in Learner's Problem Solving
Masao NAKAJIMA^{*}, Kazuo SAKAI^{*}, Hiroshi KATO^{**}, Toru NISHIGAKI^{*}

^{*}Meiji University, 1-9-1 Eifuku, Suginami-ku, Tokyo 168, Japan

^{**}NEC Corp., 4-1-1 Miyazaki, Miyamae-ku, Kawasaki-shi, Kanagawa 216, Japan

【 3 】 まとめ

現在、このエディタ/シミュレータの評価試験として、このシステムを用いてプログラミングの初心者を対象とした実験演習を行っている。

その結果、(1)ビジュアル言語であるため、プログラムの構造の把握が容易である。(2)部品に適切な名前がつけられるので、第三者にもプログラムの内容を理解しやすい。(3)初心者、特に文科系学生にとってネックとなる数式や条件式などの入力を、電卓のようなインターフェースで解決したなどの利点が明らかになった。そして、従来のプログラミング言語と比べ、プログラムの構築が容易となり、労力も低減されることが確かめられた。また、プログラムのモジュール単位での編集やモジュールへの統合・モジュールの分解が簡単にできるため、構造的なプログラミングを学ぶためのツールとしての意義も認められた。

学習者の行動履歴からは、bottom-up や top-down のアプローチのような学習者のプログラミングの意図が、ある程度把握できた。これについては、今後熟練者などを

含めた様々な被験者により実験を行い、「プログラミングの意図」についてさらに研究を進める。そして、現在研究・開発中である教育支援システムの中に学習履歴の解析結果を反映させ、学習者モデルとして結実させていく予定である。

【謝辞】

本研究・開発にあたり、ご協力頂いた日本電気(株)の木村義、佐藤雅男、両氏に感謝する。

【参考文献】

- 1)中嶋 正夫他：“アプローチ指向のアルゴリズム教育支援 I T S”、情報処理学会 第43回全国大会論文集 (1)、p.21-22 1991.
- 2)宮本 勲：“ソフトウェアエンジニアリングの現場と展望”、ソフトウェアリサーチアソシエイツ.
- 3)阪井 和男：“問題解決のアプローチ手法としてのプログラミング”、私立大学等情報処理教育連絡協議会 第5回私情協大会、1991.9.3.

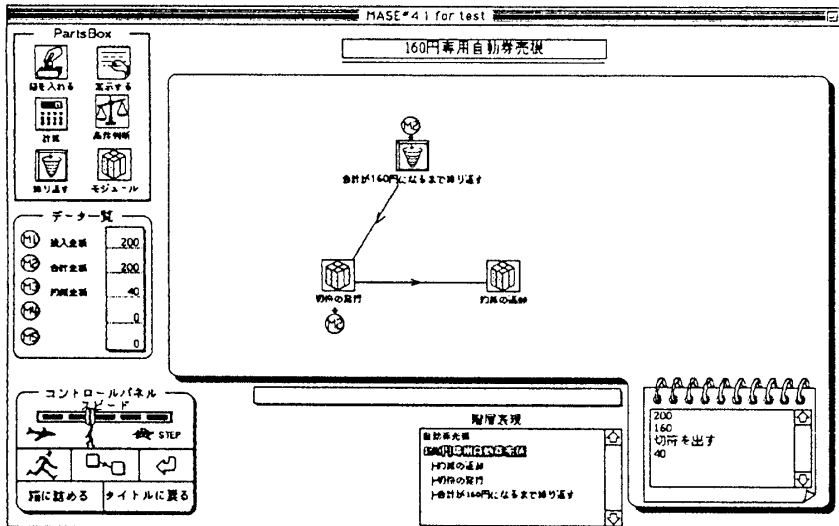


図 1

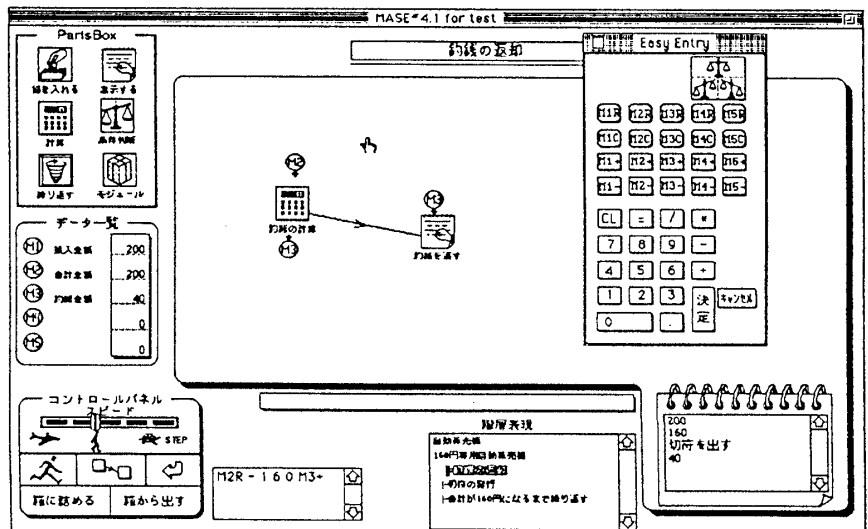


図 2