4F－1

プログラミング演習のためのプログラム及びデータ構造化エディタ

竹澤のり子　秋山憲一　上田賀一

茨城大学工学部情報工学科

1．はじめに

プログラミング演習におけるプログラム作成は、アルゴリズムを構造化チャート図で表現し、その上でプログラミングするといった当然の工程を踏まない方法で行われがちである。これは種々のプログラミング言語の形態にとらわれ、そのためプログラミングそのものを充分に理解できないことを意味する。この問題に対し、本研究ではプログラミング教育用としての構造化チャートエディタを開発する。当然の工程を抵抗なく行えるようにするために、初心者が理解しやすく容易に扱える図式的なプログラム構造やデータ構造を与え、演習段階に応じてプログラム制御構造やデータ型の使用レベルを設定できる構造化チャートエディタについて述べる。

2．設計目標

プログラミング演習において理解すべき最も重要な点はプログラムの構造であり、構造化されたプログラムの理解および構造化プログラムを作成できる能力を身に付けることである。さらに、プログラミングの初心者にとって理解しにくく作成に困るのが、データ構造である。本研究では、プログラムの構造だけでなくデータの構造も容易に取り扱えることを目的とする。具体的には以下の要求を設計目標とした。

・C, Pascal, Fortran 等の標準的な言語に対応すべく、連接・反復・選択の3つの基本要素以外に各種の制御構造を用意する。
・演習段階に応じて、制御構造やデータ構造の使用レベルを設定できる。
・ルーチン（モジュール）の記述上の構造や呼び出し関係の構造を容易に扱える。
・データ構造の型定義を用意された基本的なリスト構造やツリー構造から作成できる。
・全体から部分へのトップダウン的開発を原則的に支援するが、部分からの開発も自由に行える。
・マウスによる直接操作方式のインタフェースを多用する。

3．システム概要

本システムは構造化エディタ pred といい、Fig.1 に示す構成を取る。pred は Unix 上に構築され、グラフィカルユーザインタフェースのために X-window を用いている。取り扱うファイルには以下の5種類がある。

・プログラムファイル
　C, Pascal, Fortran 等の言語で記述されたソースプログラムファイル.

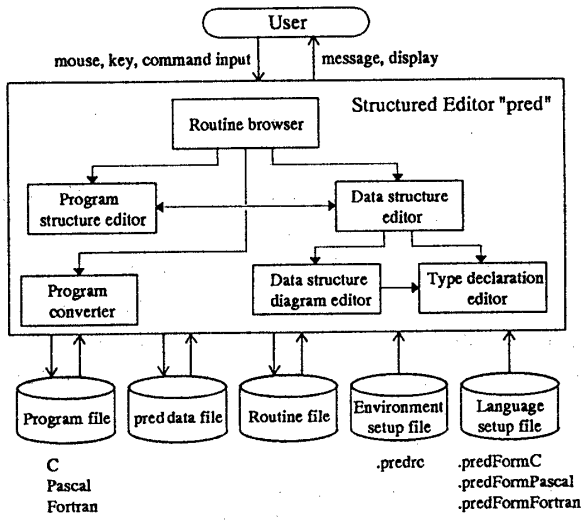

Fig.1 System Components of Structure Editor "pred".

・pred データファイル
　構造化エディタ pred の内部データに対応した書式のデータを保持したファイル。
・ルーチンファイル
　ルーチン群のライブラリとしてのファイル。
・環境設定ファイル
　対象プログラミング言語の種類およびエディタ環境の各種既定値を記述したファイル。
・言語設定ファイル
　各プログラミング言語の制御構造やデータ構造などの特徴を記述したファイル。このファイル中の記述により制御構造やデータ構造の使用レベルを設定できる。

Fig.1 の内部に示すように pred はいくつかの機能部分からなり、以下に各機能を説明する。

○ルーチンブラウザ

これは Fig.2 に示されるように構造化エディタ pred を起動したときに表示される最もベースのウィンドウである。内部は、ルーチンの呼び出し関係を表示する部分、ルーチンの全体の構造を表示する部分、未製作ルーチンを表示する部分、編集作業に用いるルーチン群のカットバッファ部分からなる。中央部分を作業領域として構造を作成・変更することができる。

○プログラム構造エディタ

Structured Editor of Program and Data for Programming Exercise.
Noriko TAKEZAWA, Ken-ichi AKIYAMA, Yoshikazu UEDA
Ibaraki University

これは各ルーチン毎に別々のFig.3のようなウィンドウを持ち、この上でYAC IIをベースとして考案した構造化チャート図SSCを用いてプログラムを作成する。プログラム構造エディタは、入出力引き数、ルーチン内で定義した変数および使用可能な変数を示す部分、そのルーチンで使用可能な定数や型を示す部分、プログラムを作成する部分およびそのためのメニューボタン群の部分からなる。作業を容易にするためカットバッファを持ち、ポップアップメニューによりプログラムの一部分をカット・コピー・ペーストすることができる。また、そのカットバッファの内容を表示したり、対応するデータ構造エディタの起動を行える。

構造化図式表記法には様々なものがあるが、図記号領域とテキスト領域を区別し、文字列長により構造図が分かりにくくなることを防いだチャート図としてYAC IIがある。本研究では、YAC IIの先の特徴を始めとする様々な特徴を取り込み、更に木構造モデルで表された制御線をより簡潔にすべく作業の流れを中心とした構造化チャート図SSC (Structured Stream Chart) を提案し、これを用いる。

○データ構造エディタ

各ルーチン毎に別々のFig.4のようなウィンドウを持ち、ルーチン内で宣言したデータ型を示す部分、そのルーチンで使用可能なデータ型を示す部分、データ構造図エディタとデータ型宣言エディタを呼び出すボタン群の部分からなる。このウィンドウはデータ型一覧表として存在しており、実際の編集はここから呼び出されるデータ構造図エディタとデータ型宣言エディタで行う。また、ここから対応するプログラム構造エディタの起動も行える。

○データ構造図エディタ

Fig.5に示すようなウィンドウを持ち、データ構造図を作成する部分とそのためのメニューボタン群の部分からなる。データ構造を作るにあたり、ユーザがひとつずつ組み上げていくこともできるが、予め用意されたリストやツリーのデータ構造を選び、内部を埋めていくことで容易に作成できる。

○データ型宣言エディタ

データ構造図エディタ中のひとつの構造部分に対する型宣言エディタがFig.6に示すようなウィンドウを持って起動する。ウィンドウ内には、データ型を宣言する部分とそのためのメニューボタン群の部分がある。データ構造図エディタとデータ型宣言エディタは一対で働く。

4．実現

これまでに上述した機能の主要部分を有するシステムはほぼ開発できているが、現在のところプログラムソースファイル入力に対するコンバータが無く、また、各エディタの細かい部分での機能の不備が残っており、開発を進めているところである。

5．まとめ

本研究では、プログラミング演習という初心者を対象とした構造化エディタの開発を試みた。本システムは多くの要求を取り入れ、プログラミング技術の習得が容易になるように設計した。現在は開発中で実現に到っていないが、実現後は実際の演習を通して実用試験を行い、評価を与えたいと考える。

【参考文献】原田賢一編：構造エディタ，共立出版（1987）

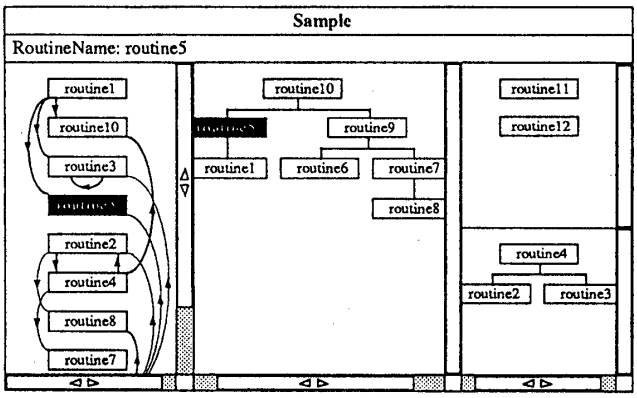

Fig.2 Outline of Routine Browser.

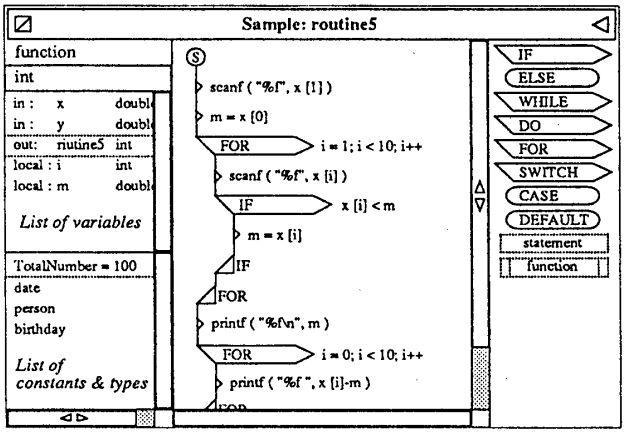

Fig.3 Outline of Program Structure Editor.

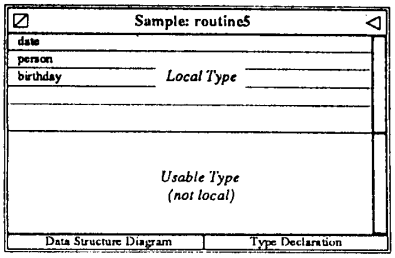

Fig.4 Outline of Data Structure Editor.

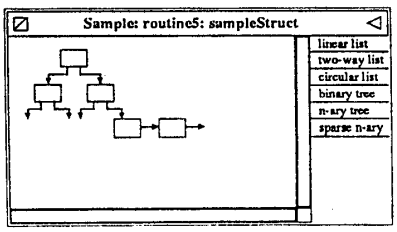

Fig.5 Outline of Data Structure Diagram Editor.

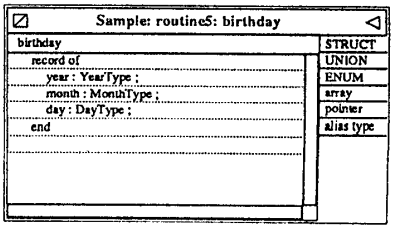

Fig.6 Outline of Data Type Declaration Editor.