図的関数型言語のプログラミング環境の設計と記述

6J－8

森山 宣郎　布川 博士　野口 正一
東北大学電気通信研究所

1.はじめに

我々は図式をシンタックスに持つ関数型言語D-Cresc.[1]を提案している。これはλ計算の図的表記法である。本論文ではD-Cresc.によるプログラミング及びプログラムの実行を行なうためのシステム(D-Cresc.システム)とその設計について述べる。

2.図的関数型言語D-Cresc.[1]

D-Cresc.(Diagramatic Crescendo)は関数型言語Crescendo[1]の図式表現である。D-Cresc.の基本要素はCresc.で書かれた式を1つのモジュールとしたものである。このモジュールに対し変数束縛を表わす矢印を付け、他のモジュールへの適用を表わす線で結び、更に、組み立てられたモジュールに名前を付ける事により新たにモジュールを作成する。図1にモジュールの例を示す。

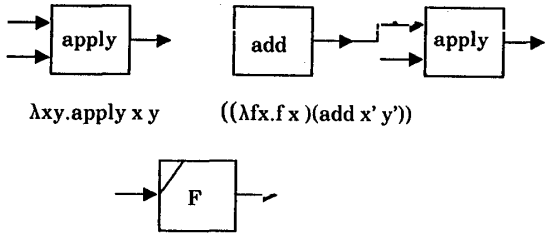

図1 D-Cresc.のモジュール

3.D-Cresc.システム

3.1 D-Cresc.システムの概要

D-Cresc.システムは、モジュールを作成するためのエディタ、その他モジュールの作成や実行のための種々の機能、ツールを全て含めた視覚的なプログラミング環境である。

3.2 D-Cresc.モジュール作成支援環境

モジュールの作成や実行を行なう場合、より抽象度の高いレベルでの操作で行えることが望ましい。D-Cresc.システムでは、モジュールや各種ツールを記入、保存するために、シートをユーザに提供している(図2)。シートは、各種デバッグツール等を記入するシステムシート、基本関数の入ったビルトインシート、ユーザがモジュールを記入するユーザーシートがある。ユーザはシートの中にあるモジュールやツールを自由に参照できる。

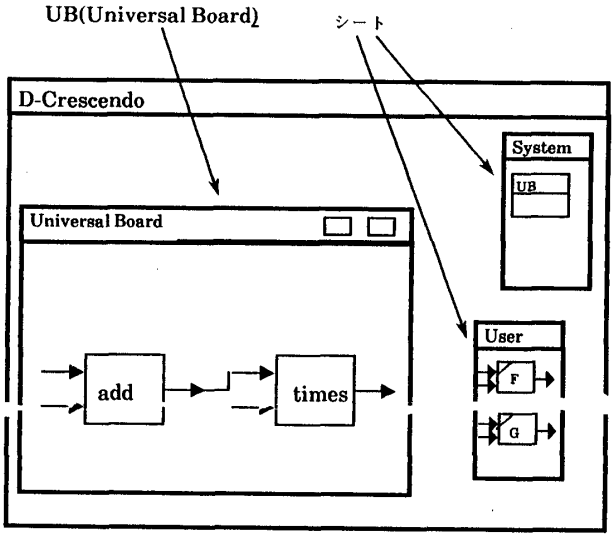

図2 D-Cresc.システムの概観

UB(Universal Board)(図2)は、モジュールの作成や実行及びデバッグを行なうための場である。モジュールの作成は2章で述べた手順で行なう。できあがったモジュールは、ユーザーシートにコピーして保存する。

3.3 D-Cresc.システムの構成

本システムのユーザインタフェースはシーハイム・モデル[2]に基づいて構成した(図3)。図中Pre.UBとは、UBに関するプレゼンテーション部の機能を表す。その他も同様である。シーハイム・モデルでは、ユーザインタフェースを除く機能をアプリケーション部と呼ぶ。D-Cresc.システムでは、以下の機能がこの部分を構成する。

エディタ部

D-Cresc.モジュールの内部表現の作成、編集

Design and Description of Programming Environment of Diagrammatic Functional Programming Language

Noburo Moriyama, Hiroshi Nunokawa, and Shoichi Noguchi
Research Institute of Electrical Commumication, Tohoku Univercity

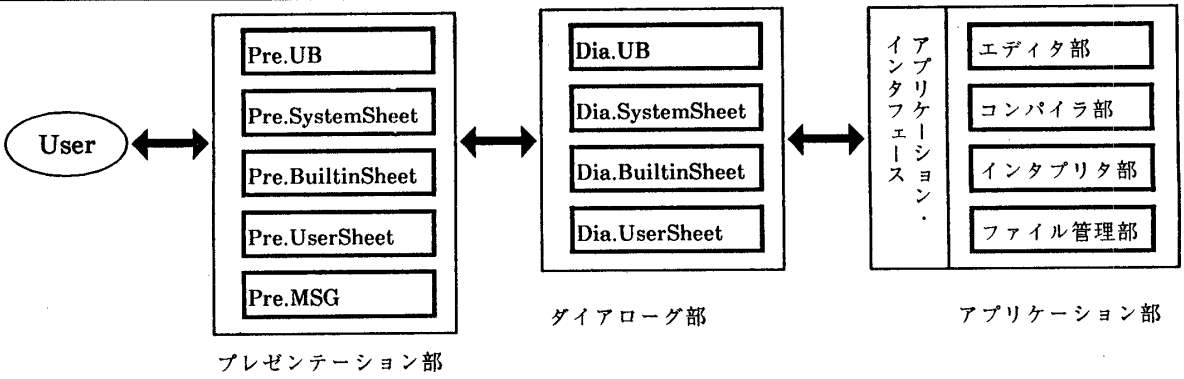

図3 シーハイム・モデルに基づくD-Cresc.システムの構成

コンパイラ部
　D-Cresc.内部表現をCresc.式へ変換
インタプリタ部
　Cresc.インタプリタである。D-Cresc.内部表現はコンパイラ部でCresc.式に変換されたあと、ここで計算される。
ファイル管理部
　D-Cresc.内部表現やツールを保存する部分である。

4.ユーザインタフェース部の設計

4.1 ユーザインタフェース部の構成

ユーザインタフェース部は、アプリケーション部の種々の機能を視覚的に提供したり、ユーザ入力を変換する部分である。構成法は、ユーザインタフェース部をUBや各シートのための機能に分割し、更に分割された各部について(1)表示を行い、又ユーザ入力を直接受ける機能、(2)UBやシート、アプリケーション部の動作の決定を行う機能に分けた(図3)。

4.2 ユーザインタフェース部の動作の記述

ユーザインタフェース部の各部の動作を代数的言語の1つである項書き換え系で記述した。図4は記述の例である。記述は、70個程度の書き換え規則で出来た。ここではプレゼンテーション部とダイアローグ部間のメッセージの受渡しを記述している。itはUB上でユーザが選択したモジュール等の情報をダイアローグ部へ送るためのものである。

例えば、規則(Pre.Rule1)は、プレゼンテーション部が表示しているUBのメニューから、削除が選択された時(イベントを受け取った時)、ダイアローグ部にメッセージ削除を送る、ということを意味する。ダイアローグ部では、メッセージdeleteを受け取ると、削除する対象の有無に従って、送るメッセージを決定する(Dia.Rule1)。

(a) Pre.UBの動作記述(一部)

```
Pre.UB(Menu(削除))▸Dia.UB(delete)                    (Pre.Rule1)
Pre.UB(Menu(移動))▸Dia.UB(move)                      (Pre.Rule2)
Pre.UB(Menu(内部参照))▸Dia.UB(reference)
Pre.UB(Menu(名前付け))▸Dia.UB(naming)
Pre.UB(Menu(実行))▸Dia.UB(run)
```

(b) Dia.UBの動作記述(一部)

```
Dia.UB(delete)▸if it ≠nil then Pre.UB(EditArea (del(it)))
                    else Pre.MSG(どこを消しますか);
                        Pre.UB(EditArea(select(*part*)));
                        Dia.UB(delete)                (Dia.Rule1)

Dia.UB(move)▸if it ≠nil then
        if it(2) ≠then Pre.UB(EditArea(move it(1) it(2)))
                else Pre.MSG(どこへ移動しますか);
                    Pre.UB(EditArea(click(*pos*)));
                    Dia.UB(move)
    else Pre.MSG(どれを消しますか);
        Pre.UB(EditArea(select(*part*));
        Dia.UB(move)                                  (Dia.Rule2)
```

図4 項書き換え系による記述の例

5.まとめ

図的関数型言語D-Cresc.のためのシステムの構成及びシステムのユーザインタフェース部の動作の記述を行なった。今回述べた記述は抽象度が高く、基本的な動作のみ定義されている。システムの実現のためには、更に詳細化を行なう必要がある。特に、デバッグ時の動作の記述をも容易に行えなくてはならない。そのための仕様定義言語の開発も必要であろう。

参考文献

[1] 布川博士, 富樫敦, 野口正一: 図式をシンタックスに持つ関数型言語, コンピュータソフトウェア, Vol.6,, No.2(1989), pp.11-23.
[2] Green, M.: Report on Dialogue Specification Tools, User Interface Management Systems, Pfaff, G. E. (ed.), Springer-Verlag, 1983,