4R-9

開発過程記述用図式エディタの作成

西村　好洋　　飯田　元　　井上　克郎　　鳥居　宏次
大阪大学　基礎工学部

1．はじめに

ソフトウェアの開発過程を形式的に記述する試みがなされている[1],[2]．形式的に記述する事によりあいまいさをなくし，作成されるソフトウェアの品質を保つ事が期待できる．

我々は，関数型言語ＰＤＬ（Process Description Language)とその処理系を作成し[3]，さらに，構造的開発過程記述用言語ＳＰＤＬ（Structured Process Description Language)とその処理系をすでに作成している[4]．今回，ＳＰＤＬによる開発過程の記述（スクリプト）を効率よく作成できるようＳＰＤＬ用図式エディタを作成した（図１）．この図式エディタは画面上でマウス等を利用してＳＰＤＬスクリプトを作成できる．このエディタで作成したスクリプトはトランスレータによってＰＤＬスクリプトに変換され，ＰＤＬインタープリタにより解釈実行される．本稿では，ＳＰＤＬ用図式エディタについて述べる．

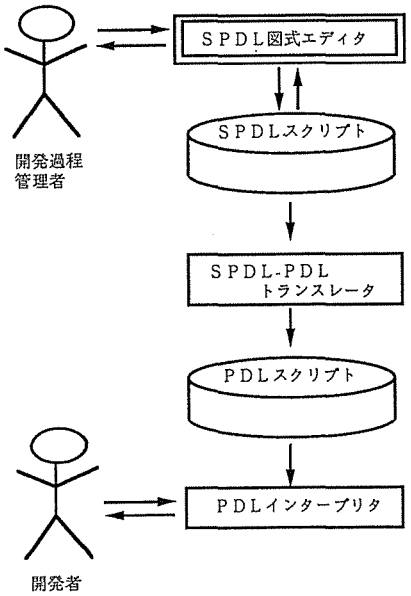

図1　ＰＤＬ,ＳＰＤＬシステム

2．ＰＤＬとそのインタープリタの主な特徴

ＰＤＬは，代数型言語ＡＳＬ[5]と同様の構文及び意味定義を持つ関数型言語である．ＰＤＬの基本関数にはツールの起動，ウィンドウのオープン／クローズなどの操作を行う関数がある．また，ＰＤＬには複数の作業を並列に実行するための記述法が用意されており，マニュアルを参照しながらエディットを行うなどを記述する事ができる．

ＰＤＬインタープリタはＰＤＬで記述された開発過程を解釈，実行する．また抽象的な記述から得られた，未定義関数を含むようなスクリプトでも実行できるよう，実行中に検出された未定義関数の値をユーザが決めたり，あるいはその関数の定義を与えたりする事ができる．

3．開発過程記述言語ＳＰＤＬ

開発過程を形式的に定義するため，種々の作業を表す表現形式（プロセスモデル，単にモデルと呼ぶ）として，我々はWilliams の Behavioral Process Model [2]を基にしたＢＰモデルを用いる．

ＢＰモデルにおいて，プロセスp_i は５字組

＜入力プロダクト，出力プロダクト，前提条件部，作業本体，完了条件部＞で構成される．ここで，

- 入力プロダクトはプロセス中で参照されるファイルやドキュメントなどの集合．
- 出力プロダクトはプロセスによって生成されるファイルやドキュメントなどの集合．
- 作業本体は前提条件部が満たされたときに行なう作業．ツール起動などの具体的な作業か，または，サブプロセス集合S_iである．S_iはプロセスの半順序集合であり，上界:Startと下界:Endがある．S_iの任意の要素p_jに対して，p_iをp_jの親プロセス，p_jをp_iのサブプロセス（子プロセス）と呼ぶ．
  各p_jはS_i以外の他のサブプロセス集合に属してはならない．
- 前提条件部は，<条件式，分岐先>の２字組の集合．プロセスの作業本体を行う前に各条件が調べられる．
  分岐先はp_iの属するサブプロセス集合S_kに属する１つのプロセスで，条件が満たされないとき実行される．

A Graphical Editor for a Process Description Language
Yoshihiro NISHIMURA, Hajimu IIDA, Kasturo INOUE and Koji TORIIFFaof Engineering
Faculty of Engineering Science, Osaka University

・完了条件部はプロセスでの作業を行なった後に判定する条件の集合．前提条件部と同様な２字組の集合である．

このように，ＢＰモデルでは開発作業をプロセスという単位で表現する．プロセスは階層構造を持ち，１つのプロセスを細かいサブプロセスに分解することができる．また，開発過程全体を１つのプロセスとみなすことができる．各プロセスは普通入力プロダクトを用いて，出力プロダクトを生成する．

作業本体で定義したサブプロセス間の順序関係はいわばプロセス間の正規の流れである．それに対して，前提条件部及び完了条件部で定義する分岐先とは，処理の流れの中で何らかの例外が起こり，正常な処理を続行できなかったときにとるべき流れであるといえる．

このＢＰモデルを形式的に記述するためにＳＰＤＬを設けた．ＳＰＤＬでは，各プロセスの５つの構成要素を順次テキストとして記述していく．

４．ＳＰＤＬ用図式エディタ

ＳＰＤＬスクリプトは直接テキストエディタ等を利用して記述する．この際，プロセス間の順序関係，親子関係について確認しながら，注意深く記述を進める必要がある．そこで効率よくスクリプトを作成できるよう図式エディタを作成した．

図式エディタはＸ－Ｗｉｎｄｏｗシステムとマウスを利用することによりスクリプトを作成する上での必要な情報を視覚的な形で表示，入力し，ＳＰＤＬスクリプトの作成支援を行う．

このエディタでは，１つのウィンドウが１つのサブプロセス集合に対応する．さらに個々のサブプロセスがウィンドウの中に箱の形で表示される（図２）．図２では，プロセス　”ｍａｉｎ”に対して，そのサブプロセス　”ｅｄｉｔ”と”ｃｏｍｐｉｌｅ”を示している．それぞれの箱の中にはそのプロセスの名称が書かれている．プロセス間の順序関係および前提／完了条件を満たさない場合の分岐先は，マウスにより入力し，矢印の形で示される．この順序関係の入力に対して不備がある場合には警告が出される．矢印には３種類あり，それぞれ，前提条件を満たさない場合の分岐先（点線の矢印），完了条件を満たさない場合の分岐先（細線の矢印），プロセス間の正規の流れ（太線の矢印）で区別している．図２の例では，”ｅｄｉｔ”の後に”ｃｏｍｐｉｌｅ”を行うことを太線の矢印で示している．”ｃｏｍｐｉｌｅ”の前提条件または完了条件が満たされない場合は，分岐先が”ｅｄｉｔ”であることを細線と点線の矢印で示している．

これらの入力を終えた後に，それぞれのプロセスについて，前提条件及び完了条件，作業本体における具体的な作業を，それぞれ画面上に開かれる３組のエディタからテキストで入力する．

プロセスによっては，前提／完了条件が不要なプロセスもある．それらについては，入力が必要のないことをメッセージで表示し，ユーザに知らせる．また，作業本体が具体的な作業のときはその記述を行う．一方，サブプロセスの系列としてすでに定義されている場合には，作業本体部の記述はそこから自動的に生成する．

このように，ＳＰＤＬ用図式エディタを用いることにより，直接ＳＰＤＬスクリプトを記述するよりも短い時間で正確なスクリプトをつくることが期待できる．

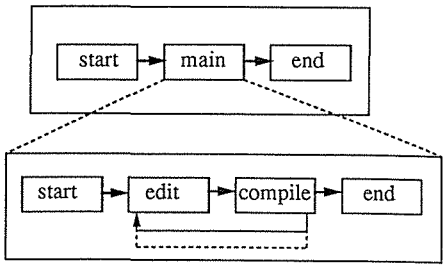

図２　簡単な開発過程の例

５．おわりに

構造的開発過程記述言語ＳＰＤＬによるスクリプト記述を支援するための図式エディタについて，その概要を説明した．現在，この図式エディタはプロトタイプ版であり，今後はエディタとしての機能拡張及びその操作性の向上を図りたい．

［参考文献］

[1] L.Osterweil:"Software processes are software too",Proc.of 9th ICSE,pp.2-13(1987).

[2] L.G.Williams:"Software process modeling : A behavioral approach",Proc.of 10th ICSE, pp.174-186(1988).

[3] 荻原，井上，鳥居:”ソフトウェア開発を支援するツール起動自動制御システム”，信学論　Ｄ－Ｉ採録決定(1989).

[4] 飯田，荻原，井上，新田，鳥居：”ソフトウェアプロセス記述言語ＳＰＤＬとその処理系の設計”，情報処理学会第３８回全国大会，3M-8(1989).

[5] 東野，関，谷口：”代数的仕様から関数型プログラムの導出とその実行”，情報処理，Vol.29, No.8,pp.881-896(1988).