ソフトウェアプロセス記述言語ＳＰＤＬとその処理系の設計

3M-8

飯田　元+　荻原　剛志+　井上　克郎+　新田　稔++　鳥居　宏次+
+大阪大学　基礎工学部　++㈱ＳＲＡ

1．はじめに

ソフトウェア開発過程を形式的に記述する試みがなされている。[1-3]形式的記述を行なうことによってあいまいさなく人々に伝えることができたり、また正しさの検証を行なったりできる。我々は開発過程を関数型言語PDL(Process Description Language)を用いて記述している。[5]また、この記述を解釈実行し、その開発過程の支援環境を提供するPDLインタプリタを作成した。[4] PDLは、記述の自由度が高く制御の流れや状態の変化を細かに記述できるが、開発過程全体を分かりやすい形で記述することが困難であった。本稿では、開発過程をBP(Behavioral Process)記述と呼ぶスタイルで構造的に記述を行なうことのできるSPDL(Software Process Description Language)について述べる。

2．開発過程の記述

プロセスＰは、計算機システムをある状態S_1から別の状態S_2に変える状態遷移関数である（$P(S_1)=S_2$）。いろいろなユーティリティプログラム（ツール）の実行（例えばソーティング、コンパイル等）や人間がツールと対話的に行なう作業（文章や図のエディット、ゲームプログラムの実行等）、人間が直接行なう作業（コマンドを投入する、システムのメッセージに応答する等）はプロセスである。２つのプロセスの合成　$P_1(P_2(S))$、及び並列　$P_1|P_2$もプロセスである。また次の式で定義される関数Ｐもプロセスである(特にBehavioral Process, BPと呼ぶ)。

```
P(S) ≡  if  Precond1(S)  then A1(S)
    else if  Precond2(S)  then A2(S)
          ...
    else if  Precondm(S)  then Am(S)
    else                       P'(Pbody(S))   ;
ただし
       P'(S) ≡ if  Postcond1(S)  then B1(S)
           else if  Postcond2(S)  then B2(S)
                 ...
           else if  Postcondn(S)  then Bn(S)
           else                        S
```

ここで$Precond_1,\cdots,Precond_m,Postcond_1,\cdots,Postcond_n$はシステムの状態からブール値へ射影する関数、$A_i$ ($0\leq i\leq m$) は式$P'(P_{body}(S))$または$P_i(S)$を表わす（P_iは任意のプロセス）。P_{body}は任意のプロセスである。B_j ($0\leq j\leq n$) はSまたは$P_j(S)$を表わす（P_jは任意のプロセス、iとjは同時に0にはならない）。

通常BPは、P_{body}で表わされるプロセスを実行することを目的とする。しかしあるPrecond が成り立つ時は例外として他のプロセスの実行を行なうことができる。P_{body}が実行された後はPostcondで示される条件によって次に行なうプロセスが指定される。

計算機システムの状態中には例えばドキュメントやグラフィックスのファイル、画面上に開かれているウインドやシステムクロック等の情報が含まれている。一般にソフトウェア開発過程を複数のBPに対応させ、そのBP及び補助的に用いるBPの諸定義の記述を開発過程のBP記述という。（合成または並列プロセスはBPの定義中に直接書く。）

3．ソフトウェアプロセス記述言語ＳＰＤＬ

PDL でBP記述を行うことは可能であるが，記述性や可読性に問題がある．そこで我々は，プロセス記述支援用に，BP記述を自然に行えるよう言語SPDLを新たに設計した．SPDLのプログラムは４章に述べる処理系によってPDL プログラムに変換された後，実行される．また，細かい操作を具体的に記述できるよう，PDL の記述を埋め込むことも出来る．

次にSPDLの構文を示す．SPDLプログラムは以下に示すように，データ型定義部，プロダクト宣言部，プロセス定義部，関数定義部より成る．データ型定義部ではプロダクトの型定義を行い，プロダクト宣言部では扱うプロダクトの集合の宣言を行う．動作の定義はプロセス定義部で行う．また，各プロセスの中で使用する関数は，関数定義部で定義される．関数定義部の構文は，PDL と同様である．

```
sort
 データ型定義部
end
product
 プロダクト宣言部
end
プロセス定義部
関数定義部
```

プロセス定義部では，BPの定義の系列を記述する．各BPの定義は，以下に示すように，入出力プロダクトの宣言部，サブプロセス宣言部，変数宣言部，前提条件記述部 Precond，本体記述部，完了条件記述部 Postcond から成っている．

```
process プロセス名
 [ 入力プロダクト列 -> 出力プロダクト列 ] :=
 child サブプロセス名の並び;
```

A Software Process Description Language SPDL and Its Implementation
Hajimu IIDA+, Takeshi OGIHARA+, Katsuro INOUE+, Minoru NITTA*and Koji TORII+
+ Osaka University　　* SRA

```
var 変数宣言部
( 前提条件式 : オペレーション )
[ プロセス系列式 ]
( 完了条件式 : オペレーション )
endproc
```

簡単な例として，C言語を用いた一般的なプログラム開発のプロセスを記述したものを図1に示す．

4．ＳＰＤＬ処理系

ここでは，現在作成中のSPDL処理系の概要について説明する．SPDL処理系は対話形式で，プロセス記述を支援するエディタ／コードジェネレータと，SPDLによる記述をPDLの記述に変換するトランスレータで構成される．トランスレータによって生成されたPDL 記述をPDL インタープリタで実行することで，ソフトウェアプロセスの支援を行うことが出来る（図2）．

エディタ／コードジェネレータは各プロセスにおけるサブプロセスや前提／完了条件，分岐先などを対話的に入力するための視覚的なユーザーインタフェースを持ち，SPDLプログラムの作成支援に用いられる．

```
/* Product Class Definition */
sort
   document is set of text;
   ...
end

/* Input/Output Product Declaration */
product
   Spec     : document;
   ...
end

/* Process Definition */
process main[ Spec -> Source, Exec ] :=
   child edit,compile,link,test;
   ()
   [ init, edit ] /* init is a primitive process */
   ()
endproc

process edit[ Spec -> Source ] :=
   child Viewer,Editor;
   ()
   [ Viewer | Editor ]
   ( true : compile; )
endproc

process compile[ Source -> Object ] :=
   child Compiler;
   ( !Exist( Source ) : edit; )
   [ Compiler ]
   ( status() = 0    : link;
     true            : edit; )
endproc
...
```

図1　プロセスの記述例

トランスレータは，SPDLプログラムをPDL のプログラムに変換する．各プロセスはライブラリを利用してPDL の基本関数や定義関数に置き換えられる．その定義が完全でない関数がある場合は，変換時に警告が出される（しかし，PDL インタープリタに未定義関数処理機能が用意されているので，エラーとはならない）．

5．おわりに

ソフトウェア開発過程を形式的に記述するための枠組BP記述，及び，BP記述モデルに基づくソフトウェアプロセス記述言語SPDLを提案するとともに，SPDL処理系の概要を紹介した．現在，SPDL処理系のプロトタイプの構築と言語仕様の評価，更に，効率的なプロダクト管理方法についての研究を行っている．

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,pp174-186(1988).

[3] 鈴木，片山: "ＨＦＳＰモデルに基づくソフトウェアプロセス記述言語"，日本ソフトウェア科学会第５回大会論文集,B2-4,pp.93-96.

[4] 荻原，飯田，新田，井上，鳥居 :"ソフトウェア開発環境記述用関数型言語の設計と処理系の試作"，信学技報,COMP88-73.

[5] 稲田，荻原，井上，菊野，鳥居 :"ジャクソン開発法の形式的記述の詳細化とその実行",信学技報,COMP88-74.

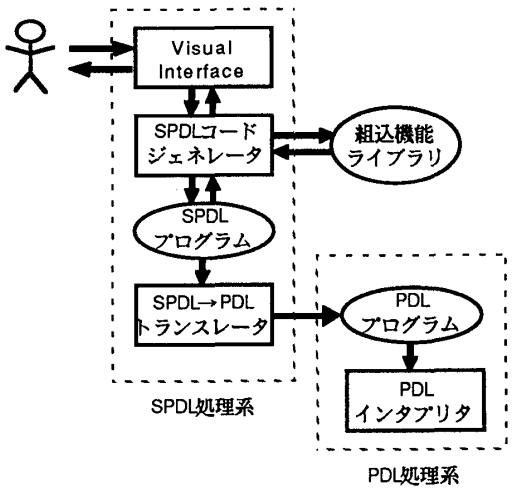

図2　ＳＰＤＬ処理系