OUKA
The University of Osaka
Institutional Knowledge Archive

Title	関数型言語に基づくプロセスプログラミングシステム
Author(s)	飯田, 元; 西村, 好洋; 荻原, 剛志 他
Citation	電子情報通信学会秋季大会講演論文集. 1989, 1989, p. 250-251
Version Type	VoR
URL	https://hdl.handle.net/11094/51104
rights	copyright©1989 IEICE
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

https://ir.library.osaka-u.ac.jp/

The University of Osaka

SD-7-2

関数型言語に基づくプロセスプログラミングシステム

A Process Programming System Based on a Functional Language

飯田元* 西村好洋* 荻原剛志* 新田稔** 井上克郎* 鳥居宏次*
Hajimu IIDA Yoshihiro NISHIMURA Takeshi OGIHARA Minoru NITTA Katsuro INOUE Koji TORII

*大阪大学 基礎工学部
Faculty of Engineering Science, Osaka University

**㈱ＳＲＡ
Software Research Associates, Inc.

1. はじめに

最近，ソフトウェアの開発過程（プロセスと呼ぶ）を厳密かつ形式的に記述し，その記述から開発過程の特性を調べたり，記述を基に開発を支援するシステムを作成する試みが行われている[2]．

我々は，開発過程を記述するための関数型言語ＰＤＬ(Process Description Language)を設計し，ＰＤＬの処理系（ＰＤＬシステム）を試作している[4]．本稿ではＰＤＬの背景，及び，ＰＤＬシステムの構成の概要を述べる．

2. 開発過程記述言語

2.1 開発過程記述言語ＰＤＬ

ソフトウェア開発過程を記述するための言語としては

(1) 意味定義が簡明で，厳密であること
(2) その言語を用いた開発過程の記述に基づいて，開発を行えるような処理系が存在すること

等が望まれる．

我々は関数型言語ＰＤＬを設計し，そのインタプリタを開発した．ＰＤＬは代数的仕様記述言語ＡＳＬ／Ｆ[1]を基礎とした言語である．ＰＤＬは様々な抽象化レベルによる記述が可能であり，開発過程を段階的に詳細化することができる[6]．

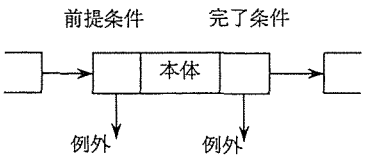
2.2 ＢＰモデルとそれに基づくマクロ言語ＳＰＤＬ

開発過程を表すための方法の１つに，ＢＰ (Behavioral Process)モデルと呼ばれるものがある[3]．

ＢＰモデルでは，１つのプロセスは

①そのプロセスを始めるために満たすべき条件（前提条件）の集合
②複数のサブプロセスの系列（プロセス本体）
③プロセスを終了するために満たすべき条件（完了条件）の集合

で，定義される（図１参照）．



本体：サブプロセスやツール起動のシーケンス

図１ ＢＰモデルのプロセス

我々はＢＰモデルに沿ったＰＤＬによる記述作成を容易にするため，言語ＳＰＤＬ (Structured Process Description Language)を設計した．ＳＰＤＬはＰＤＬへの変換を前提とした言語であり，ＳＰＤＬによる記述のもつ意味は，ＰＤＬへの変換則によって定められている．

3. システムの構成

3.1 全体の構成

図２にＰＤＬシステム全体の構成を示す．開発過程を記述し，その支援システムを作成するには，まず専用の図式エディタまたは通常のテキストエディタを用いてＳＰＤＬで記述する．この記述をトランスレータによってＰＤＬによる記述に変換し，ＰＤＬインタプリタで実行する．以下，ＰＤＬシステムの構成要素について説明する．

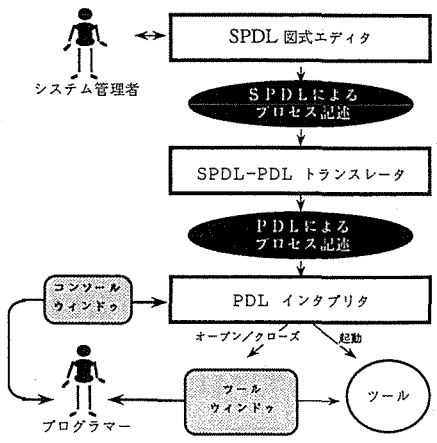


図２ ＰＤＬシステム全体図

3.2 ＳＰＤＬエディタ

ＳＰＤＬエディタはＢＰモデルの各プロセスを箱や矢印等を用いて定義し，対応するＳＰＤＬの記述を出力する図式エディタである（図３参照）．

3.3 ＳＰＤＬ－ＰＤＬトランスレータ

ＳＰＤＬによる開発過程の記述をＰＤＬによる記述に変換する．その際に，プロセスの階層構造の一貫性に矛盾がないか等を調べる．

3.4 PDLインタプリタ

PDLインタプリタはPDLで記述された開発過程を解釈，実行する．そのために，ツール起動関数やウィンドウ関数，並列演算子などを組み込み関数として備えている．また，マクロ定義，スクリプト読み込み機能，デバッグ機能など様々な機能を備えている．

特にデバッグ機能については，未定義関数検出時の実行継続や，視覚的インターフェースによるトレース，ブレークポイントの設定などが可能であり，開発過程記述自体のプロトタイピングなどに有効である．

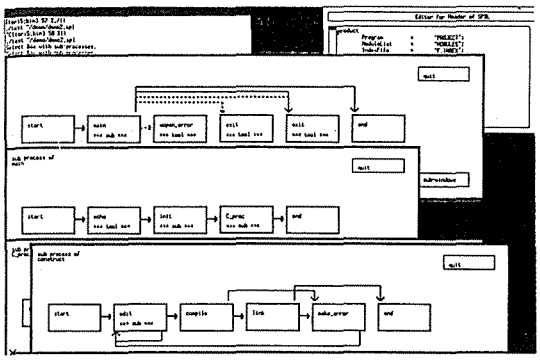


図3　SPDLエディタ画面（部分）

4．開発過程の記述例

ここでは例としてCプログラムを開発する簡単な開発過程の記述（図4）を示す．この例ではプログラムをモジュール単位に分割してモジュールの仕様を決定し，次に各モジュールで用いられる関数をデザインする．その際，デザインに関するドキュメントをそれぞれ対応するファイルに保存する．また，モジュールや関数のリストも作成する．コンパイル，リンクにはmakeユーティリティーが用いられ，更に，各ソースファイルの版の履歴を保存する．

従って，開発過程は大きくエディット，コンパイル，リンクの3つに分かれ，さらにエディットはモジュール設計，関数設計，コーディングの3つに分かれるプロセス構造となっている（図5参照）．また，各プロセスにおけるエディタやmakeユーティリティー，版管理ツールなどの起動を自動化しする記述がなされている．

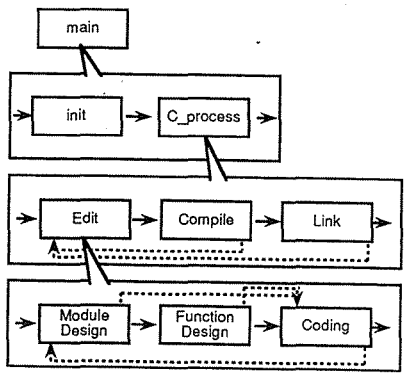


図5　例題におけるプロセス構造

5．おわりに

ソフトウェア開発過程の記述とその記述に基づくソフトウェア開発を支援するためのPDLシステムについて述べた．現在，プロトタイプシステムがUNIX 4．3BSDのXウィンドウシステム上で稼働している．また，プロセスの管理に加えてプロダクト管理機能の組み込みについて研究を行っている．

参考文献

(1) K.Torii, Y.Morisawa, Y.Sugiyama, and T.Kasami: "Functional programming and logical programming for the telegram analysis", Proc. of 7th ICSE, pp.463-472 (1984).

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

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

(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，SS88-35 (1988).

(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，SS88-36 (1988).

(6) K.Inoue, T.Ogihara, T.Kikuno, and K.Torii: "A formal adaptation method for process descriptions", Proc. of 11th ICSE, pp.145-153(1989).

```
//-----------------------------------------------------------------
//      Main Process
//-----------------------------------------------------------------
process main [->]==
child init,C_proc;
( status(sysmod(0,S))=-1 >> exit;
  status(term_tool("xterm",S))=-1 >> exit;
  status(
        wopen("-T ProgrammingExamples -fn cour_b_14 -geometry 80x24+100+100",S
        <>0     >> wopen_error;)
[echo("C program construction kit"),init,C_proc]
( status(wclose(S))<>0  >> wclose_error;)
endproc

//-----------------------------------------------------------------
//      Sub-processes of /main
//-----------------------------------------------------------------
process init [->]==
child new_project,mk_directory,mk_files,dir_error;
()[ new_project,mk_directory,mk_files ]()
endproc

process C_proc [ Spec -> Source, Exec, Document ]==
child view_doc,construct;
()[ view_doc @ construct ]()
endproc
...
```

図4 a　SPDLによるプロセス記述

```
//SPDL:Process main begin
main(S) == prec_main(S);
prec_main(S) == if  ( status(sysmod(0,S))=-1 ) then exit(S)
                else if  (status(term_tool("xterm",S))=-1 ) then exit(S)
                else if  (status(
        wopen("-T ProgrammingExamples -fn cour_b_14 -geometry 80x24+100+100",S
        <>0     ) then wopen_error(S)
                else  posc_main(body_main(S));
body_main(S) == init(echo("C program construction kit", S));
posc_main(S) == if  ( status(wclose(S))<>0       ) then wclose_error(S)
                else  next_main(S);

next_init(S) == C_proc(S);
next_C_proc(S) == S;

//SPDL:Process main end
//SPDL:Process init begin
init(S) == prec_init(S);
prec_init(S) ==  posc_init(body_init(S));
body_init(S) == new_project(S);
posc_init(S) ==  next_init(S);
 ...
```

図4 b　PDLによるプロセス記述