

動的詳細化が可能なプロセス記述の表記法と その開発現場への応用について

石若通利* 元治景朝* 荻原剛志** 井上克郎**

*株式会社 さくらケーシーエス
研究開発部

**大阪大学
基礎工学部情報工学科

神戸市中央区浪花町 27 番地(興和ビル 5 階) 大阪府豊中市待兼山町 1 番 1 号
Tel: (078) 391-5768 Fax: (078) 391-5452 Tel: (06) 850-3063 Fax: (06) 850-3053

あらまし ソフトウェアの開発プロセスを形式的に記述し、開発支援環境を構築する研究が行われている。しかし一方で、詳細なプロセスを記述することの難しさが指摘されている。そこで我々はプロセス記述の詳細化をプロジェクト管理者と開発者の協調作業によって段階的に行うプロジェクト支援機構を検討している。さらに、この機構において、詳細化されたプロセス記述からプロジェクト計画の策定及びプロセスの実行を支援する機能に関して検討を進めている。

本稿では、管理者と開発者の協調作業によって段階的にプロセス記述を詳細化するための機構及びこの機構で用いる表記法を提案する。さらに、本機構の 1 つの機能として詳細化されたプロセス記述よりプロセスの実行優先度を導出し、プロジェクト計画の策定を支援する方法を提案する。

和文キーワード 段階的詳細化, プロジェクト計画, 実行優先度

A Practical Study of Project Supported System Using Stepwise Particularized Process Description

Michitoshi ISHIWAKA*, Kagetomo GENJI*, Takeshi OGIHARA**, Katsuro INOUE**

*Department of Research and Development,
SAKURA KCS Corporation
27, Naniwa-cho, Chuo-ku, Kobe, 650, Japan
tel. :+81-78-391-5768 fax. :+81-78-391-5452

**Department of Information and Computer Sciences,
Faculty of Engineering Science, Osaka University
Toyonaka, Osaka 560, Japan
tel. :+81-6-850-3063 fax. :+81-6-850-3053

Abstract There exist a lot of researches on software process in order to establish process-centered software development environments. It has been pointed out, however, that it is difficult to specify software process in detail. To mitigate the difficulty, we have been studying a project support system which provides cooperative process description and stepwise particularization of software process. Moreover, the system handles the detailed process descriptions to support the project planning for a manager and the process practice for developers. This article proposes the project support system and a software process representation method which the system employs. The article also proposes one of the system functions which obtains the priority of actual processes from the detailed process descriptions.

英文 key words Stepwise Particularization, Project Planning, Process Priority

1 はじめに

大規模なソフトウェアの開発プロジェクトは、多数の開発者の協調作業によって遂行される。このようなプロジェクトでは、管理者は個々の開発者の開発活動を統合的に管理し、円滑に制御する必要がある。近年、この開発活動の管理と制御を目的とするソフトウェアプロセスのモデル化と記述法の研究が盛んである [1, 2, 3, 4, 5]。さらに、プロセス記述に基づくツールの自動実行機能等、開発環境の生成が実現されている [6]。

しかし、一方では詳細なプロセス記述を事前に記述することの困難さが指摘されている [7]。例えば、プロジェクトの開始時点では、プロジェクトの最終目標であるプログラムの種類、数、プログラム間の関係はしばしば不定である。従って、管理者がプログラムに対応するプログラミングプロセスの種類、数、関係を事前に予測し、厳密に記述することは不可能である。また、プロセスは開発者の能力や好みに依存する。このため、総ての開発者に適したプロセス記述は膨大なものとなる。現実的にはこれらを事前に用意することはできない。

そこで我々は、プロジェクトにおける予実管理情報としてプロセス記述を採用し、管理者と開発者の協調によるプロセス記述の詳細化機構を検討している [8]。さらに、動的に詳細化されるプロセス記述を用いたプロジェクト管理支援法を検

討している。この機構では、管理者はプロジェクト計画としてプロジェクト全体のプロセスを記述する。開発者は、個人計画として全体プロセス記述中の担当プロセス(個人プロセス)を能力や好みに応じて詳述する。これらの協調的なプロセス記述活動を支援するために、機構は全体プロセス記述と個人プロセス記述との分解と統合を行う。さらに、詳述された個人計画に基づいて、個々のプロセスの実行優先度を含む詳細なプロジェクト計画を策定する。これにより、管理者は個々の開発者の開発活動を統合的に管理し、円滑かつ効率的にプロジェクトを運営することが可能となる。

本稿では、2章で上述のプロジェクト管理機構を概説する。3章でプロセス記述法と、全体と個人プロセス記述の分解/統合方法を述べる。4章で本機構の1つの機能であるプロセスの実行優先度を含む詳細なプロジェクト計画の策定方法を述べる。5章で簡単なプロジェクトの例を用いて本機構の有効性を示す。6章で、本稿のまとめと今後の課題を述べる。

2 プロジェクト管理機構の概要

プロジェクト管理機構は管理者のためのプロジェクト計画支援機能と、開発者のためのプロセスの実行支援機能で構成される(図1)。以降では、この管理機構の動作概要を説明する。

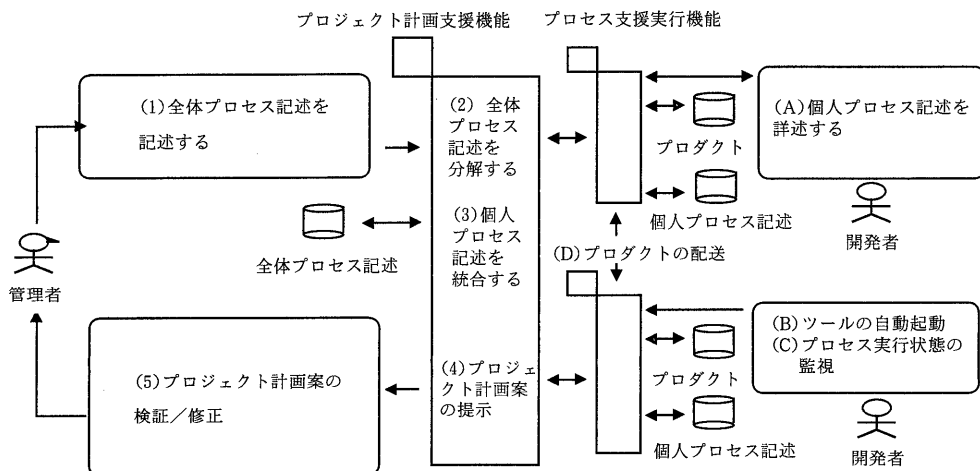


図1: プロジェクト管理機構

2.1 プロジェクト計画支援機能

プロジェクト計画支援機能は、管理者が記述したプロジェクト計画としての全体のプロセス記述を担当開発者毎に分配する。さらに開発者が記述した個人計画としての個人プロセス記述を収集し、全体プロセス記述に統合する。

(1)管理者が、プロジェクト全体のプロセスを記述する。全体プロセス記述は、プロジェクトの開発標準に従ったプロセスのひな型である。

(2)管理機構が、全体プロセス記述を担当開発者毎の個人プロセス記述に分解する。同時に、実行可能なプロセスを検出し、その担当開発者に作業指示として配送する。

(3)管理機構が、各開発者によって詳述された個人プロセス記述を収集し、全体プロセス記述に統合する。これにより、全体プロセス記述の一部は詳細化される。

(4)管理機構が、全体プロセス記述を解析し、以降のプロセスの計画を策定する。プロジェクト計画案は、全体の開発効率を考慮したプロセスの実行優先度と、それに応じた担当開発者の割り当てで構成される。

(5)管理者が、プロジェクト計画案を検証し、必要に応じて修正する。

上記(2)からを繰り返すことにより、全体プロセス記述は詳細化される。さらに、リアルタイムなプロジェクト計画の立案と進捗の管理が可能となる。

2.2 プロセス実行支援機能

詳述された個人プロセス記述に従い、プロセスの実行支援を行う。

(A)開発者が、個人プロセス記述を詳述する。個人プロセス記述は、各開発者の能力や好みに応じて記述される。

(B)管理機構が、詳述された個人プロセス記述に従い開発ツールの起動を行う。

(C)管理機構が、開発者のプロセス実行状態を監視する。

(D)管理機構が、開発者間のプロダクトの受け渡しを行う。

3 プロセス表記法

本機構は、全体プロセス記述と個人プロセス記

述用の2種類の図式表現を採用する。

3.1 全体プロセス記述

全体プロセス記述は、プロセスを表わすノード、プロダクトを表わすノード、その間のアークによって図式表現する。次の6つの要素により定義される(図2)。

A: プロダクトの集合

T: プロセスの集合

I: プロセスと入力プロダクトの関係集合

$I: T \rightarrow A$

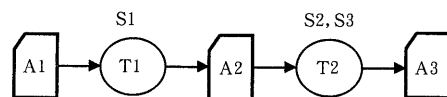
O: プロセスと出力プロダクトの関係集合

$O: T \rightarrow A$

S: 開発者の集合

G: プロセスと開発者の関係集合、

$G: T \rightarrow S$



$A = \{ A1, A2, A3 \}$

$T = \{ T1, T2 \}$

$I(T1) = \{ A1 \}, I(T2) = \{ A2 \}$

$O(T1) = \{ A2 \}, O(T2) = \{ A3 \}$

$S = \{ S1, S2, S3 \}$

$G(T1) = \{ S1 \}, G(T2) = \{ S2, S3 \}$

図2: 全体プロセス記述例

3.2 個人プロセス記述

個人プロセス記述は、詳述されたプロセスを表わすノード、詳述されたプロダクトを表わすノード、その間のアークによって図式表現する。次の9要素によって定義する(図3)。

Ap: 詳述されたプロダクトの集合

Tp: 詳述されたプロセスの集合

Ip: プロセスと入力プロダクトの関係集合

$Ip: Tp \rightarrow Ap$

Op: プロセスと出力プロダクトの関係集合

$Op: Tp \rightarrow Ap$

Sp: 開発者の集合

Gp: プロセスと開発者の関係集合

$Gp: Tp \rightarrow Sp$

Lp: 開発ツールの集合

Np: プロセスと開発ツールの関係集合

$Np: Tp \rightarrow Lp$

Rp: プロダクト間の参照関係集合

Rp: Ap→Ap

図中の fp1, fp2, fp3 は、開発者 S1 が出力プロダクト A21, A22, A23 の参照関係を表現するために採用した仮想のプロセスである。例えば fp1 は、A21 を主たる入力となるプロセスは同時に A22 をも参照する必要があることを意味する。すなわち、開発者 S1 が想定した、A21, A22, A23 を入力とする次プロセスの候補である。

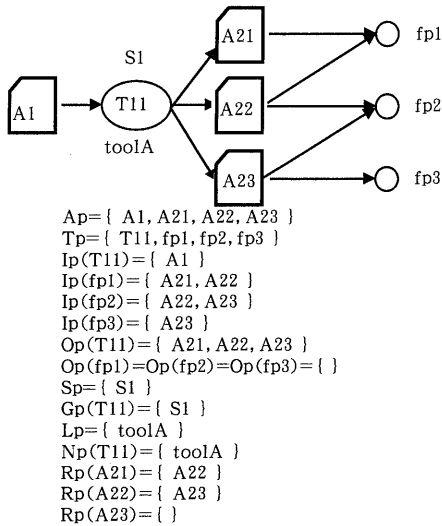


図 3: 個人プロセス記述例

3.3 プロセス記述の分解と統合

全体プロセス記述を担当開発者の個人プロセス記述に分解する。

- (1) 担当開発者毎に個々のプロセスを分類する。
- (2) 分類されたプロセスの集合ごとに隣接するプロダクトを分類する。

この結果、図 4(A) の全体プロセス記述から図 4(B) の個人プロセス記述が導出できる。この時、 $Ap \subseteq A, Tp \subseteq T, Ip \subseteq I, Op \subseteq O, Sp \subseteq S, Gp \subseteq G$ の関係が成立する。

担当開発者によって詳述された個人計画をプロジェクト計画に反映するために、個人プロセス記述を全体プロセス記述に統合する。

- (1) 分解時導出された個人プロセス記述(B)を詳述された個人プロセス記述(C)で置換する。
- (2) 全体プロセス記述のプロセスと個人プロセス記述の仮想プロセスを同定する。但し、同定時仮

想プロセス数に応じて全体プロセス記述のプロセスを複写する。

(3) Lp, Np の情報を棄却する。これらの個人プロセス実行支援のための情報は、管理者がプロジェクト計画を策定することに影響を与えないため統合時に棄却される。

この結果図 4(C) の個人プロセス記述から(D) の全体プロセス記述を導出できる。この時点で、Ap, Tp, Ip, Op, Sp, Gp により A, T, I, O, S, G は再定義される。

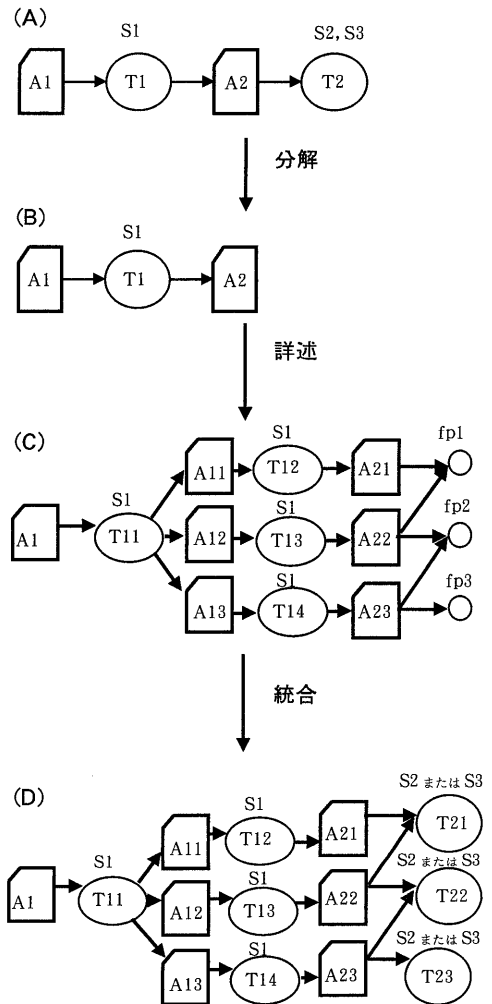


図 4: プロセス記述の分解と統合

4 プロセス実行優先度策定法

図4(D)に示す例を用いてプロセスT11, T12, T13の実行優先度の導出方法を示す。

前提条件

○S1がT11を開始する時点でS2, S3は待ち状態とする。

○T21, T22, T23の何れかが実行可能となった時点でS2, S3の何れかは当該プロセスを即時に開始する。すなわちこの時点でS2, S3の何れかは待ち状態から解放される。

○S2, S3の待ち時間は、T11の開始時点から待ち状態の解放時点までを計算する。

○T11, T12, T13, T14の実行所要時間をそれぞれ1, 1, 2, 1とする。本所要時間は、担当開発者S1からの計画としてプロジェクト管理機構が獲得する。

目標

S2, S3の待ち時間の和を最小とするプロセスT12, T13, T14の実行優先順位を導出する。

導出手順

図5のようにプロセスの実行順序のパターン木を作成しながらS2, S3の待ち時間の和と実行可能なプロセス数の累積を計算する。

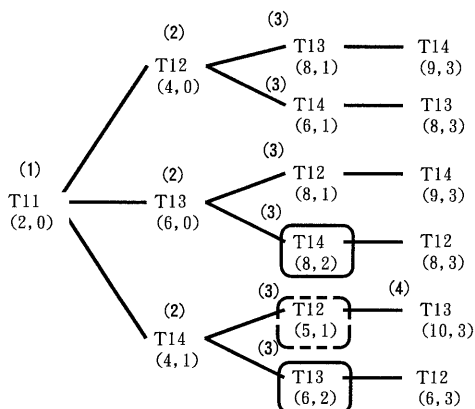


図5: プロセス実行順序木

実行可能プロセス数の累積が待ち状態の担当開発者数以上になった時点で、待ち時間の和が最小となる実行優先順位を獲得する。

(1) T11の実行を完了した時点では待ち時間の和

は2である。実行可能プロセス数の累積は0である。図中の括弧は

(待ち時間の和, 実行可能プロセス数の累積)を表わす。

(2)次にT12, T13, T14のそれぞれを順に実行した場合の待ち時間の和と実行可能プロセス数の累積を計算する。T14を実行した場合には、実行可能プロセス数の累積が1となりS2, S3の何れかは待ち状態から解放される。

(3)(2)のそれぞれの場合において、残る二つのプロセスの何れかを完了した場合の待ち時間の和と実行可能プロセス数の累積を計算する。四角で囲まれたT14の場合にはS2, S3ともに待ち状態から解放される。四角で囲まれたT13の場合も同様である。この時、T11, T13, T14の順で実行された場合の待ち時間の和は8であり、T11, T14, T13の順に実行した場合のそれは6である。さらに同位のほかのノードの待ち時間は、破線で囲まれたT12を実行する場合を除き6以上である。従って機構は本パターン木の探索をT12の場合を除いてここで中止する。T12以降の探索の結果T11, T14, T12, T13の実行した場合の待ち時間の和は10となる。最終的に、S2, S3の待ち時間を最小とするプロセスの実行順序、T11, T14, T13を得る。

5 プロジェクト管理機構の動作例

伝票集計システムの開発プロジェクトを例にプロジェクト管理機構の機能を説明する。このプロジェクトの入力は伝票集計システム設計書である。出力は複数のプログラムである。このプロジェクトの構成員は管理者、開発者A、開発者B、開発者Cである。

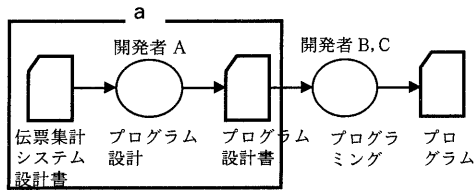
(1)管理者が全体プロセスを記述する。

管理者は、ソフトウェアの品質や保守性等の幾つかの要素を考慮して、中間的プロダクトとしてプログラム設計書を作成することを決定した。その結果、このプロジェクトの開発標準として図6に示す全体プロセス記述を記述した。また、複数のプログラミングプロセスが発生することを予測し、このプロセスに複数の開発者を割り当てた。

(3)本機構が個人プロセス記述へ分解する。

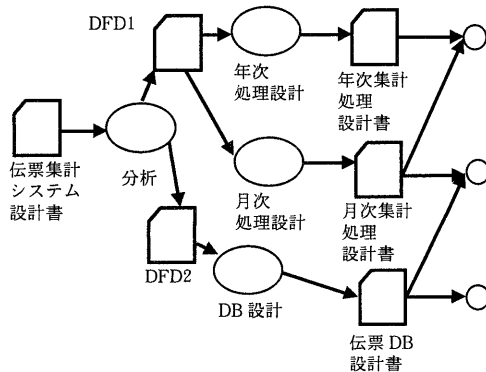
全体プロセス記述の再左端ノードである伝票集計システム設計書は、プロジェクトの入力として与えられている。従って、プログラム設計プロセス

スは実行可能である。本機構は担当開発者 A に管理者からの作業指示として全体プロセス記述の一部(図 7a)を配送する。



S = { 開発者 A, 開発者 B, 開発者 C }
 G(プログラム設計) = { 開発者 A }
 G(プログラミング) = { 開発者 B, 開発者 C }

図 6: 全体プロセス記述



Np(分析) = { DFD エディタ }
 Np(処理設計) = { フローチャートエディタ }
 Np(DB設計) = { ERD エディタ }

図 7: 個人プロセス記述の詳述例

(4) 開発者 A が個人プロセス記述を詳述する。
 開発者 A は、システム設計書をデータフローダイアグラムエディタ (DFD エディタ) を使って分析することにした。フローチャートエディタと ER ダイアグラムエディタ (ERD エディタ) を使って各処理設計と DB 設計をそれぞれ行うこととした。その結果、個人プロセス記述を図 7 のように詳述した。さらに、伝票集計システム設計書の分析を開始し、出力するプロダクトの計画を年次集計処理設計書、月次集計処理設計書、伝票 DB 設計書と決定した。また、各プロセスの作業時間を分析プロセス、年次集計処理設計プロセス、伝票 DB

設計プロセスにそれぞれ 1 時間、月次集計処理設計プロセスに 2 時間と見積り、機構を通じて管理者に計画として報告した。

(5) 機構が個人プロセス記述を全体プロセス記述に統合する。

これにより、詳細化されたプロジェクトの全体プロセス記述が管理者に提示される。

(6) 機構がプロジェクト計画案を提示する。

3 章で述べたプロジェクト計画案策定法に従って、開発者 A のプロセスの実行順序を導出する。その結果、伝票 DB 設計書、月次集計処理設計書、年次集計処理設計書の順で実行すると、開発者 B, C の待ち時間の和が 6 時間となり、プロジェクトの運営上効率的であることが管理者に通知される。

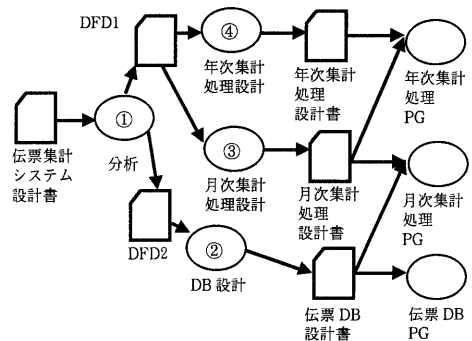


図 8: 詳細化された全体プロセス記述表示例

(7) 管理者が計画案を検証する。

管理者は、計画案を検証し不都合がなければその計画案を承認し、機構を通じて開発者 A にプロセスの実行順序を指示する。同時に開発者 B, C を個々のプログラミングプロセスに割り当てる。

5 おわりに

本稿では、プロセス記述を実開発プロジェクトへ応用する際の問題点を指摘した。そして、この問題を解決するための、管理者と開発者によるプロセス記述の段階的な詳細化機構を提案した。本機構は、管理者と開発者間での予実管理の手段としてプロセス記述を採用する。これにより、プロセスを記述することの余分な作業負担を課すことなく、詳細なプロセス記述を獲得することが可能となる。さらに、詳細なプロセス記述の計算機処理とプロセス記述の実開発プロジェクトへの

適用を可能とする。また、本稿では直観的な理解に易い有向グラフを用いた図式表記法を提案した。

今後の課題としては、(1)本プロジェクト管理機構にプロジェクト全体のスケジュールを取り込み、より有効なプロジェクト管理支援について検討する。(2)繰り返しの発生する複雑なプロセスへの応用を検討する。(3)本プロジェクト管理機構の実装方法について検討することを予定している。

謝辞

この研究の遂行にあたり、貴重な議論及び助言を戴いた奈良先端科学技術大学院大学の鳥居宏次教授、ならびに大阪大学鳥居研究室諸氏に感謝します。また、本研究の機会を与えて下さった株式会社さくらケーシーエスの盛田政敏取締役研究開発部長、ならびに有益な議論をして戴いた研究開発部諸氏に感謝します。

参考文献

- [1] Osterweil, L.: "Software Processes are Software Too", Proc. of 9th ICSE, pp. 2-13 April 1987.
- [2] Inoue, K., Ogihara, T., Kikuno, T., and Torii, K.: "A Formal Adaptation Method for Process Descriptions", Proc. of 11th ICSE, pp. 145-153, 1989.
- [3] Barghouti, N. S. and Kaiser, G. E.: "Multi-Agent Rule-Based Software Development Environments", Proc. of 5th KBSA, pp. 375-385, September 1990.
- [4] Katayama, T.: "A Hierarchical and Functional Software Process Description and its Enaction", Proc. of 11th ICSE, pp. 343-352, 1989.
- [5] Peiwei Mi and Walt Scacchi: "Process Integration in CASE environments", IEEE Software, Vol. 9, No2, pp. 45-53, March 1992.
- [6] 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", 電子情報通信学会論文誌 (D-1), J72-D-I, 10, pp. 742-749,

1989.

[7] Pankaj K. Garg and Sanjay Bhansali: "Process Programming by Hindsight", Proc. of 14th ICSE, pp. 280-293, May, 1992.

[8] 石若, 元治, 荻原, 井上: "プロセス記述を用いた分散開発環境の生成", 第45回情報処理学会全国大会, 講演論文集(5), p277-278, 10月 1992.