

7K-8

## 協調的なプロセス記述支援機構の 開発現場への適用可能性について

石若通利\* 元治景朝\* 荻原剛志\*\* 井上克郎\*\*\*

\*(株) さくらケーシーエス \*\*奈良先端科学技術大学院大学 \*\*\*大阪大学

### 1. はじめに

ソフトウェアの開発活動を分析し、効果的な支援方法を確立するための研究が盛んである[1]。これまでに、多くのプロセスモデルとその記述法が提案されており、それに基づくプロセス駆動型の開発環境が実現されている[2]。しかし、現実の多様なプロセスに対応した詳細なプロセス記述を、事前に獲得することは困難である。そこで、我々はプロジェクトの管理者と開発者による協調的なプロセス記述支援機構を検討してきた[3]。本機構はまず、(1)管理者が記述した全体のプロセス記述を各開発者へ分解し配送する。次に、各開発者はこれに基づき担当プロセスを詳述し、(2)本機構がこれらの詳述されたプロセス記述を収集し、全体のプロセス記述へ統合する。さらに、(3)詳細化された全体のプロセス記述より以降のプロセス計画案を策定する。ここでは、本機構の開発現場への適用可能性を検証するために、あるプロジェクトの事後追跡調査を行ないプロセス記述実験を行なった結果について述べる。本機構の基本的なアイデアと機能は開発現場において有効であることがわかった。

### 2. プロセス記述実験の概要

既に完了したプロジェクトの管理資料と最終プロダクトを基に、プロセスの事後追跡調査を行なった。本プロジェクトはハードウェアの開発と並行する制御系ソフトウェアの開発プロジェクトである。規模としては、最大時約100名の開発者が同時に稼働するものである。このうち一部の機能設計からプログラム設計に至るプロセスを推測し記述した。その結果約400個のプロダクトノードを持つプロセス記述

A Cooperative Process Description Support System  
and its Experimental Adaptation to a Real Project  
Michitoshi Ishiwaka\*, Kagetomo Genji\*, Takeshi  
Ogihara\*\* and Katsuro Inoue\*\*\*

\*SAKURA KCS Corp., \*\*JAIST Nara, \*\*\*Osaka Univ.

を得た。この記述実験から、本機構の他の現実のプロジェクトへの適用可能性を考察する。

### 3. 実験結果からの考察

#### 3.1 段階的詳細化の機能に関する考察

プロジェクト開始当初の分割機能数見積に基づくプロセス記述と、機能設計の一部が進行した時点でのプロセス記述では出力プロダクトの数の減少が観測された(図1)。これは機能設計の担当者によるもので、機能分割の現実的な最適化の結果と考えられる。その結果に従って、以降のプロセス計画も必然的に改訂された。

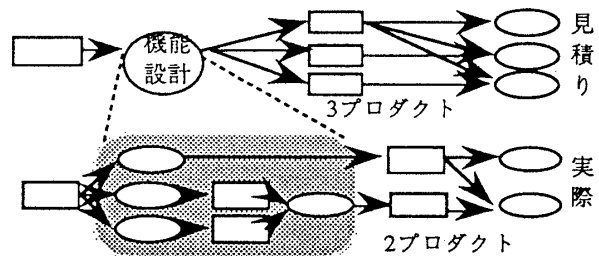


図1：プロセス記述の変更例

この例からわかるように、プロセスの記述の詳細化には柔軟性が必要である。さらに、一部のプロセス記述の変更が全体に与える影響の解析が必要である。本機構では、管理者と開発者によって協調的且つ段階的にプロセスの詳述化を行なうので、柔軟なプロセスの改訂及び詳細化ができる。また、プロジェクト全体を見通すプロセス記述の一元管理によりプロセス記述の改変等による他のプロセスへの影響波及解析が可能である。

#### 3.2 プロセスの実行順序の導出機能に関する考察

プロダクト間の参照関係の複雑化に伴って、どのプロセスから開始すべきかを直観的に判断することは困難である。実験対象プロジェクトでは、プロダクトを相互に参照するプロセスが観測された(図2)。これは、夫々の設計対象プログラムが互いにデータ

を交換する場合で、データ定義が異なるプロダクトに分散する時等に起こり得る。

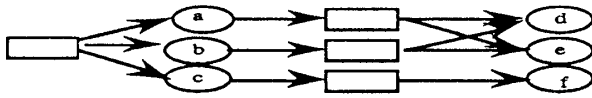


図2：相互参照を含むプロセス記述例

このようなプロダクトの相互参照を含む複雑なプロセス記述では、それらのプロセス(a,b,c)の実行所要時間により、以降のプロセス(d,e,f)担当者の待ち時間を最小とする最適なプロセスの実行順序を直感的に判断することは難しい。これは、本機構のプロセスの実行順序の導出[3]が有効に機能する場面である。

3.3 再実行プロセスの記述支援に関する考察

記述実験の過程で、既知のプロセス記述を繰り返すようなプロセスを多数発見した。このような繰り返しに相当するプロセスの代表的なものは、レビュー及びテスト後のプロセスである。そのレビュー及びテストの結果、評価の対象となるプロダクトを作成してきた一連のプロセスを再び実行することが起こり得る。しかし、我々はこの種のプロセスの再実行は繰り返してではなく、評価結果に従った新たなプロセスの生成であると考え。なぜならば、再実行プロセスでは評価結果のプロダクト、評価対象のプロダクト等が既に存在する。最初の実行時よりも参照可能な情報が増えているため、プロダクトとプロセスの関係は最初のそれよりも複雑となるはずである。また、実験対象プロジェクトでは再実行プロセスの担当者が替わることさえ起こり得た。この担当者の変更により、再実行におけるプロセスは別のプロセスとして記述する必要が生じた。

図3はレビューを含んだプロセス記述の例を表現している。(b)は(a)のレビュー&再実行プロセスが詳細化された一例を示している。この例は、設計書レ

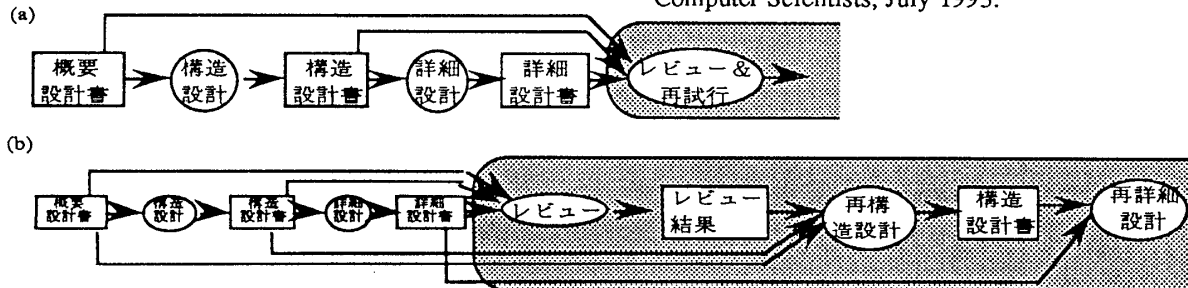


図3：再実行を含むプロセス記述例

ビューの結果、構造設計書の修正が必要である場合に相当する。構造設計書を修正するためには、(1)それを出力とする構造設計プロセスを再実行する必要がある。(2)再実行のためのプロセスは、再実行対象プロセスの出力である構造設計書と直前のプロセスの出力であるレビュー結果を入力とする必要がある。(3)構造設計書が修正されるため、それを入力として実行されたプロセスも再実行する必要がある。今後、既知のプロダクトの入出力関係と再実行時のプロセス記述にある規則性を整理し、プロセス記述の導出を今後検討する必要がある。

4. おわりに

本稿では、完了したプロジェクトのプロセスの記述を行なうことにより協調的なプロセス記述支援機構の有効性を確認した。今後の課題としては、(1)前述の再実行プロセスの記述支援方法の検討、(2)クリティカルパスを考慮したプロジェクトのスケジューリングへの応用方法の検討、(3)本機構の実装及び実プロジェクトへの適用実験等を考えている。

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くだ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に感謝します。

参考文献

[1]Proc. of 2nd Int. Conference on Software Process, Feb. 1993  
 [2]Proc. of 8th Int. Software Process Workshop, March 1993  
 [3]Ishiwaka, Genji, Ogihara and Inoue:"A Project Support System Using Stepwise-Particularized Software Process Descriptions", Proc. of 3rd Int. Conference for Young Computer Scientists, July 1993.