

## 3P-05 ワークフローと構成管理の統合環境を実現した

### システム開発支援ツール

煤孫統一郎 倉谷和彦

株式会社 情報技術コンソーシアム 研究開発部

#### 1. はじめに

分散環境でのソフトウェア開発（システム開発）は、担当者間の確実な情報伝達と、仕様書、プログラムソースなどの整合性確保が困難である。このことに起因するソフトウェアの品質低下が大きな問題となっている。

特に、関係者が大人数で複数の会社に所属し、地理的に分散しているようなソフトウェア開発では、以下の理由で障害が多発する。

- ・配布された変更通知内容が自分に関係するものなのか、判断が難しい
  - ・変更が確実に行われたのか管理者が把握できない
- 分散環境でのソフトウェア開発を支援する研究は、ワークフローによる支援の研究<sup>(1)</sup>と、構成管理による支援の研究<sup>(2)</sup>がある。ワークフローは担当者間の情報伝達を確実に行うために有効である。構成管理は成果物間の整合性を確保するために有効である。従来の研究はどちらか一方の視点から行われていた。上記問題を解決するためには、担当者間の確実な情報伝達と、成果物間の整合性確保を同時に行うことが重要である。本研究では、これらの統合環境を提案する。

#### 2. 事例調査

本研究におけるモデル作成にあたって、実際のソフトウェア開発を調査した。中規模のソフトウェア開発を3事例調査し、(1)成果物の概要（成果物名、目的、記載事項等）、(2)成果物管理の問題点、(3)標準的モデル（ウォータフォール・モデル、スパイラル・モデル）との相違点を検証した。

この結果、(1)一度生成された成果物は何回も改訂される、(2)成果物は作業結果にあわせて変更される、

(3)中間成果物が存在する、ことが観察された。

中間生成物とはヒヤリング・メモ、技術検討メモ、議事録などである（以下これらをまとめて議事録と書く）。標準的モデルでは、議事録の内容は成果物に反映されることを前提とし、議事録を正規の成果物としていない。しかし事例調査では、議事録が作業指示書として使われ、後で参照される事例が多く観察された。このように、実際のソフトウェア開発では議事録が重要な役割を果たしていることを発見した。

#### 3. 統合環境のモデル

事例調査の結果をふまえ、議事録に注目したワークフローと構成管理の統合環境のモデルを構築した。

##### (1) ソフトウェア開発の定義

ソフトウェア開発を、成果物を変更（生成、修正、削除）することと定義する。ソフトウェア開発における作業とは、ある成果物が参照している成果物に変更された場合、整合性を確保するように成果物を変更することと定義する。参照している成果物に変更されるか、議事録の変更（生成）によって成果物の変更を要求されない限り、自分が担当している成果物であっても変更することはできない。

##### (2) レビューと議事録の位置づけ

本モデルでは、議事録をシステム（ツール）に取り込むために、議事録を成果物とするレビューアクティビティを定義する。これはワークフロー・プロセスの任意の位置に起動できるものとする。成果物を変更する場合には、レビューアクティビティを起動してワークフロー・プロセスを実行しなくてはならない。これによりレビューがワークフロー・プロセスのアクティビティとして管理されるので、議事

録をトレースすることが可能になる。

### (3) リリースの定義

本モデルでは、ワークフロー・プロセスのバージョンをリリースと定義する。例を図1に示す、(1)レビューアクティビティが起動する、(2)アクティビティ1 V1.0に変更がおこる。(3)アクティビティ1のバージョンは1.1になる。(4)下流のアクティビティもワークフローの定義にしたがって順番に起動される。こうして新しいバージョンのワークフロー・プロセス(アクティビティ1 V1.1-アクティビティ2 V1.1-アクティビティ3 V1.1)はリリース1.1として管理される。これにより、過去のプロセスの参照が可能になる。

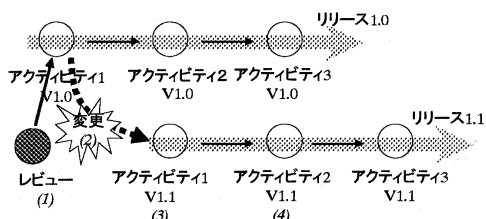


図1. ワークフロー・プロセスのリリース管理

## 4. 統合環境の実装

本モデルをWebベースのシステムとして実装した。サーバ側ではOS上でWebサーバ、アプリケーションサーバ、データベースが常駐し、それらの上でワークフロー&構成管理エンジンが実行される。分散環境での利用を前提として、クライアント側はブラウザのみとした。

本システムの構成概念図を図2に示す。レポジトリ(RDB)にアクティビティ構成情報と成果物構成情報を格納する。ワークフロー&構成管理エンジンは、レポジトリの情報を参照・更新しながら実行される。

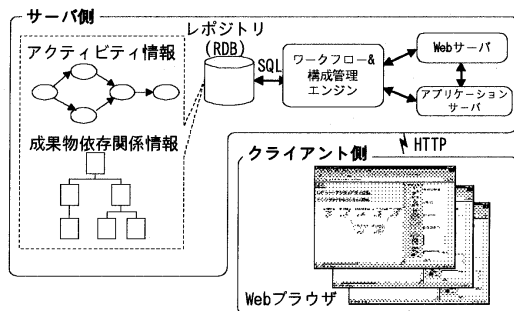


図2. システム構成概念図

## 5. 評価

本システムの評価として、ソフトウェア開発3事例について思考実験を行った。担当者間の確実な情報伝達と成果物間の整合性確保ができなかった例として、ある障害の修正が新たな障害を引き起こした現象(以下副作用と書く)に注目した。これらに本システムを適用した場合の障害の回避可能性を考察した。

たとえば、プログラムの修正が仕様書に反映されなかったために副作用が発生した場合、以下の理由により障害は回避される。

- ・ 成果物を修正するためには、その成果物が参照している成果物を先に修正しなくてはならない
- ・ 成果物への変更要求が、議事録としてシステムに取り込まれ参照可能になるため、配布された変更通知が自分に関係するものなのか判断が容易になる

思考実験の結果を表1に示す。副作用が発生した障害のうち、50%以上が回避可能であった。これらの結果から、担当者間の確実な情報伝達と成果物間の整合性確保に対して、本システムが有効であると考えられる。

表1. 本システムを適用した場合の障害回避率

	事例A	事例B	事例C
A. 障害票総数	768	1003	464
B. 副作用を伴う障害票数	33	54	18
C. 回避可能な障害票数	19	45	13
D. 障害回避率(C/B)	57.6%	83.3%	72.2%

## 謝辞

本研究開発は、(株)情報技術コンソーシアムが情報処理振興事業協会の委託を受け実施したものである。実施にあたって、京都大学、垂水浩幸 助教授のご指導および富士通エフ・アイ・ピー(株)、中電コンピューターサービス(株)、(株)三菱総合研究所のご協力を得たことに感謝致します。

## 参考文献

- (1) 山本里枝子ほか：ソフトウェア開発支援システム SOFTPIE の開発, 情報処理学会研究報告, GW, Vol.20, No.6, pp.31-36(1996).
- (2) 服部真穂ほか：ソフトウェア共同開発におけるバージョン管理支援, 情報処理学会研究報告, SE, Vol.107, No.8, pp.57-64(1996).