

Title	プロセスモデルMonoProcessを用いたソフトウェア開発管理システムのためのオブジェクトベース
Author(s)	湯本, 貴雪; 山本, 哲男; 松下, 誠; 井上, 克郎
Citation	情報処理学会第57回全国大会講演論文集. 1 P.181-P.182
Issue Date	1998-10-06
Text Version	publisher
URL	http://hdl.handle.net/11094/50386
DOI	
rights	ここに掲載した著作物の利用に関する注意 本著作物の著作権は情報処理学会に帰属します。本著作物は著作権者である情報処理学会の許可のもとに掲載するものです。ご利用に当たっては「著作権法」ならびに「情報処理学会倫理綱領」に従うことをお願いいたします。

Osaka University Knowledge Archive : OUKA

<https://ir.library.osaka-u.ac.jp/repo/ouka/all/>

3 J-5 プロセスモデル MonoProcess を用いた ソフトウェア開発管理システムのためのオブジェクトベース

湯本貴雪 山本哲男 松下誠 井上克郎

大阪大学大学院基礎工学研究科

1 はじめに

ソフトウェアの品質や生産性を向上させるため、ソフトウェアの開発過程（ソフトウェアプロセス）を表現するソフトウェアプロセスモデルの研究が広く行われている。また、それらのプロセスモデルに基づく開発管理環境の研究も盛んに行われている。

既存のプロセスモデルは、開発時の作業手順を中心として記述することでプロセスを表現しているため、それらに基づく開発管理環境もまた作業手順に着目している [1]。しかし、実際の開発では作業の流れが動的に変化することが多く、始めから作業全体を記述することは難しい。またあらかじめ記述した作業手順通りに開発を進めていくことは困難である。さらに、オブジェクト指向やコンポーネント指向のソフトウェア開発など、開発時の生成物に着目した開発に対しては、従来のプロセスモデルを用いた記述は適用しにくい。

これらの問題を解決するため、我々は開発時に生成されるプロダクトや開発環境中の様々な資源をオブジェクトとして表現し、そのオブジェクトの状態変化によりプロセスを表現するプロセスモデル MonoProcess の提案を行っている [2]。本研究では、プロセスモデル MonoProcess を用いたソフトウェア開発管理システムのためのオブジェクトベースの試作について述べる。オブジェクトベースは、オブジェクトの情報を蓄えるリポジトリとリポジトリの管理システムから構成される。オブジェクトベースにより、オブジェクトの状態変化の情報を自動的に取得、保存することが可能である。オブジェクトベースを用いることでソフトウェアプロセスの進捗状況を把握するのに必要なオブジェクト情報を効率良く取得できる。

2 MonoProcess

プロセスモデル MonoProcess では、開発環境中にある生成物や資源を表現するオブジェクトを用いた記述を行うことで、開発環境を表現する。開発環境はオブジェクトの集合で表される。オブジェクトは属性とメソッドから構成される。属性はオブジェクトの特性を表し、メソッドはオブジェクトに適用される操作を表す。メソッドの実行により開発作業を行い、オブジェクトの状態を変化させ

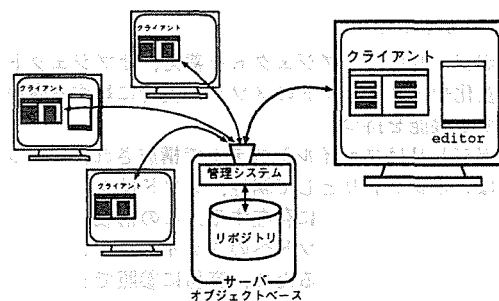


図 1: 開発管理システムの概要

る。MonoProcess ではオブジェクトの状態変化によってソフトウェアプロセスを表現する。また、オブジェクトの状態変化を記録することにより進捗状況を把握する。

3 ソフトウェア開発管理システム

MonoProcess を用いたソフトウェア開発管理システムについて述べる。

本システムは、ソフトウェア開発作業の支援と進捗状況の把握を目的とする。図1に、開発管理システムの概要を示す。開発者はクライアントアプリケーションを用いてサーバアプリケーションに接続して作業を行う。クライアントアプリケーションは、オブジェクトの情報を視覚的に表現したユーザインタフェースを持つ。サーバアプリケーションをオブジェクトベースと呼ぶ。オブジェクトベースは、オブジェクトの情報を蓄えクライアントからの要求を解析し実行する。

本システムでは、ユーザインタフェースを用いてオブジェクトに対する操作を行なえる。例えば、オブジェクトの属性を参照し、現在のオブジェクトの情報や状態を得ることが可能である。また、オブジェクトのメソッドを実行することができる。開発者は、オブジェクトの持つメソッドを実行することによって開発を進める。メソッドを実行することにより属性値の変化やメソッドの実行結果などオブジェクトの状態が変化する。このような開発時に生じるオブジェクトの状態変化、あるいは行った操作を履歴情報としてオブジェクトベースに保存する。この履歴情報をユーザインタフェースを用いて参照することで作業の現状把握が行える。

4 オブジェクトベース

オブジェクトベースは、オブジェクト情報の作成、参照、変更やオブジェクトのメソッドの実行やオブジェクト情報の履歴情報の収集などの機能を持つ。オブジェクトベースは、オブジェクト情報を格納するリポジトリとリポジトリへのアクセスを制御するアクセス管理システムから構成される。

4.1 リポジトリ

リポジトリは、オブジェクトを蓄え、オブジェクトの状態変化やオブジェクトのメソッドの実行に対する履歴を保存する機能を持つ。

リポジトリはファイルシステムで構成され、オブジェクトはディレクトリとして属性、メソッドはファイルとしてファイルシステム上に存在する。この構成によってオブジェクトの属性、メソッドへのファイルシステム上の絶対パスが一つに決定するため、容易に参照できる。

4.2 アクセス管理システム

アクセス管理システムは、クライアントからの要求を解析し、リポジトリ内の情報に対して作成、変更、削除などの操作を行う。

オブジェクトベースにはアクセス権管理機能があり、各々のオブジェクトの属性やメソッドに対し詳細にアクセス権を設定できるようになっている。

アクセス管理システムはマネージャ、ライブラリ、メソッドエンジンから構成される。マネージャはクライアントと通信を行い、クライアントからの様々な要求を解析する。また、マネージャはオブジェクトへのアクセス権を管理する。ライブラリにより各々の要求に対応したりリポジトリへの操作を行う。メソッドエンジンはオブジェクトのメソッドの内容をリポジトリから取り出し、それを実行する。

4.3 動作

図2はメソッド実行時のオブジェクトベース内の制御の流れである。クライアントからの要求をマネージャが解析しメソッドの実行であることを判定しメソッドエンジンを起動する。メソッドエンジンは、ライブラリを使ってリポジトリからメソッドの内容を取り出し実行する。その際メソッド実行の履歴は、ライブラリによってリポジトリ内へ格納される。保存される履歴情報は複数箇所に複製して保存する。これは履歴情報の参照の際、様々な抽出条件に対して高速に取り出すことができるようにするためである。一般的に情報の複製は、情報の変更時に複製全てに対して変更を加えるというオーバーヘッドが心配される[3]が、履歴情報は、内容の変更はなく単に追加される性質を持つため問題とはならない。

4.4 試作

マネージャはC言語、メソッドエンジンはPerlを用いて実装した。ライブラリはC言語とPerlで実装した。リポ

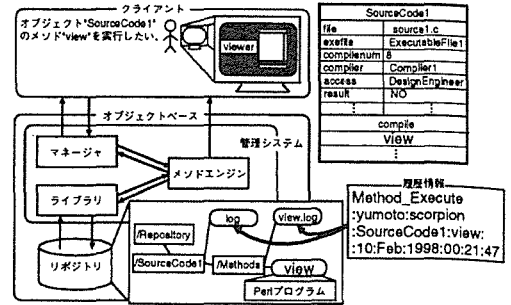


図2: メソッドの実行時におけるオブジェクトベースの動作

ジトリはUNIXのファイルシステムを利用している。

ソフトウェアプロセスモデリングのための例題[4]を本システムに適用することにより、開発環境がオブジェクトの集合としてオブジェクトベースに格納でき、開発時のオブジェクトの履歴情報が取得、保存できることを確認した。

5 まとめ

プロセスモデル MonoProcess を用いたソフトウェア開発管理システムのためのオブジェクトベースの試作を行った。

本研究により、オブジェクトベースを使ってソフトウェアの開発作業が支援できソフトウェア開発の進捗管理に必要なオブジェクトの履歴情報を効率良く取得、保存できる。

今後の課題として同時実行制御の機能追加などのオブジェクトベースの改良がある。その他、開発作業の実験データの取得、さらに他の開発管理システムとの比較や評価を行いたい。

参考文献

- [1] Bandinelli, S.C., Fuggetta, A. and Ghezzi, C.: "Software Process Model Evolution in the SPADE Environment", IEEE Transaction of Software Engineering,
- [2] 松下誠, 大下誠, 飯田元, 井上克郎: "オブジェクトモデルを用いたソフトウェアプロセス記述用言語 MonoProcess", 第14回情報処理学会プログラミング研究会(1997).
- [3] 大堀淳: "オブジェクト指向データベースの形式化", 情報処理, Vol.32, No.5, pp.550-558 (1991).
- [4] Kellner, M.I., Feiler, P.L., Finkelstein, A., Katayama, T., Osterweil, L.J., Penedo, M.H. and Rombach, H.D.: "Software Process Modeling Example Problem", Proceedings of the 6th International Software Process Workshop, pp.19-29 (1990).