

オブジェクト定義に基づく ソフトウェアプロセス管理システムの試作

山本哲男 藤原豊 大下誠 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

近年、ソフトウェアの開発過程(ソフトウェアプロセス又はプロセスと呼ぶ)を明らかにし、その改善を通じて生産性や品質の向上と開発期間の短縮を図る研究が盛んになってきている。

そのなかで、プロセス記述のためのモデルの研究が数多く行なわれている。これらのモデルでは、主に開発の作業手順を記述することでプロセスを表現している。しかし、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では、開発時の生成物に着目した開発が行われている。

そこで本研究では、開発の作業手順ではなく、開発時における生成物や資源を属性とメソッドで構成されたオブジェクトで表現するモデルの提案を行う。また、本モデルに基づくソフトウェアプロセス管理システムの試作を行った[2]。

2 モデル化手法

オブジェクトはソフトウェア開発環境におけるさまざまな生成物や資源を表し、属性とメソッドから構成される。

属性は、オブジェクトについての情報、状態を表すものである。属性名は各オブジェクトにおいて一意に定められ、属性値として属性名が示す内容が表される。

メソッドはオブジェクトに対して許された操作である。メソッド名は各オブジェクトにおいて一意に定められる。メソッド本体に作業の内容を記述する。

オブジェクトの状態を変化させることにより、ソフトウェア開発を進める。このような状態の変化をソフトウェアプロセスと呼ぶ。

3 システム

上記のモデルに基づいたソフトウェアプロセス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の環境を提供する。このシステムを用いてソフトウェア開発を行なうことで、開発作業の進捗状況の把握を容易にし、管理することができる。

本システムは以下のものから構成される。

- リポジトリ
- リポジトリアクセスプログラム
- メソッドエンジン
- HTML トランスレータ
- メソッドセンダ
- メソッドエグゼキュータ

また、この他にも Web ブラウザ、http サーバを使用する。これらは既存のものを用いた。今回の試作は、UNIX(FreeBSD) 上で行った。図 1 にシステムの構成を示す。

3.1 リポジトリ

リポジトリではオブジェクトの集合、メソッドの実行履歴を蓄えておく。このリポジトリがある計算機をサーバ、開発者が作業を行なっている計算機のをクライアントと呼ぶ。リポジトリへのアクセスはすべてリポジトリアクセスプログラムを使用して行う。このようにする事で、リポジトリの構造が変化しても他のリポジトリをアクセスするプログラムは変更する必要がない。今回、リポジトリはUNIXのファイルシステムを用いて実装した。

開発者はオブジェクトを参照することで、現在のオブジェクトの情報・状態を得る。また、開発作業はオブジェクトのメソッドを実行することで行なう。

3.2 ユーザインタフェース

オブジェクトの属性を参照したり、メソッドの実行はすべて Web ブラウザから行なう。各オブジェクトは唯一の URL(Universal Resource Locator) を持つ。Web ブラウザで閲覧したいオブジェクトに

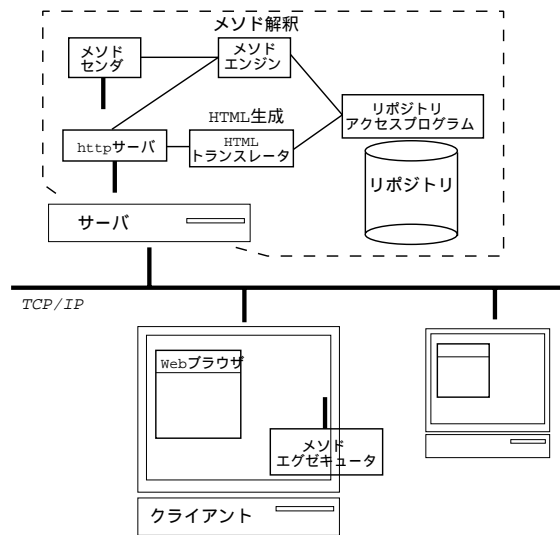


図 1: システムの構成図

に対する URL を指定することで、そのオブジェクトのページが表示される。この時、HTML トランスレータが URL で指定されたオブジェクトの属性、メソッドをリポジトリから取り出し、HTML に変換する。また、HTML トランスレータはユーザ認証も受け持つ。ユーザ認証を行うことで、だれがオブジェクトのメソッドを実行したかを認識できるようになる。

メソッドの実行は Web ブラウザでメソッド名を選ぶことで起動される。この時、メソッドエンジンにクライアントの計算機名、使用者名、オブジェクト名、メソッド名が伝えられ、メソッドエンジンがリポジトリからメソッド本体を取り出し、メソッドの解釈を行う。また、作業履歴の記録も行う。もし、この際にクライアントの計算機でツールの起動等の処理が必要ならば、メソッドセンダに依頼を行う。依頼されたメソッドセンダは、クライアント上のメソッドエグゼキュータに TCP でコネクションを張り、ツールを起動するためのコマンドを送る。そして、コマンドを受け取ったメソッドエグゼキュータがクライアント上にツールの起動を行う。

4 適用

ソフトウェアプロセスモデリングのための例題 [1] を本システムにどのようにして適用したかについて述べる。この例題では、要求の変更から始まり、コードを変更し、そのコードが単体試験をパ

スするまでについての一連の作業を規定している。

まず、この例題のなかで、デザイン、ソースコード、オブジェクトコード、テスト結果等、作業時の生成物をオブジェクトとして定義した。また、作業に携わる人、人の役割、グループもオブジェクトとした。次に、各オブジェクトに対して、属性とメソッドを考える。例えば、「ソースコード」というオブジェクトの場合、属性としてファイルの存在場所、オブジェクトコードのオブジェクト名、編集可能者等の情報、メソッドとして編集を行う、コンパイルする、コード変更の修了を通知する等を定義した。

このようにして、すべてのオブジェクトを定義しリポジトリに保存することで、開発作業の記述を行った。また、そのリポジトリでシステムを動作させた。

この適用の結果、ソフトウェアプロセスをオブジェクトの集合で表せることが分かった。また、開発作業をメソッドで定義でき、そのメソッドを実行することにより開発をすすめることができることも分かった。

5 まとめ

開発の作業手順によらず、開発時における生成物や資源をオブジェクトとして捉え、それを用いたプロセスのモデル化を行なった。また、そのプロセスモデルに基づくソフトウェアプロセス管理システムの試作を行なった。

本システムでは、オブジェクトに対する操作をメソッドという形で限定し、すべての作業を自動的に記録する。それにより、進捗状況の把握が容易になる。また、ツールの自動起動などの開発支援も行えらる。

今後の課題としては、システムの有効性を実験的に評価すること、UNIX 以外のプラットフォームへの対応が挙げられる。

参考文献

- [1] Kellner, M.I., Feiler, P.I., 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).
- [2] 大下誠, 松下誠, 飯田元, 井上克郎: "オブジェクトに着目したソフトウェア開発管理システムの提案", 情報処理学会第 54 回 (平成 9 年前期) 全国大会 4U-10, pp. 1-425 - 1-426 (1996).