



Title	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
Author(s)	大下, 誠; 松下, 誠; 飯田, 元 他
Citation	情報処理学会第54回全国大会講演論文集. 1997, 1, p. 425-426
Version Type	VoR
URL	https://hdl.handle.net/11094/50388
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。
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

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

The University of Osaka

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

4U-10

大下 誠† 松下 誠† 飯田 元† 井上 克郎†
†大阪大学 †奈良先端科学技術大学院大学

1 はじめに

ソフトウェアプロセス(プロセス)とは、ソフトウェアの開発過程のことである。プロセスに関する研究の目的は、ソフトウェアの開発過程を明らかにし、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ことにある。これまでに、現実のプロセスのモデル化と記述、よりよいプロセスの探求、プロセス改善手法、プロセスを中心とする開発支援環境の構築、さらにプロセスを実行する人間の特性やコミュニケーションの問題などについての研究が報告されている[1]。

従来のプロセスモデルの研究では、主に開発作業の手順を記述することでプロセスを表現している。また作業手順がどこまで終了したかで開発作業の進捗状況を把握しようとしている[2]。しかし、最近ではコンポーネント指向のソフトウェア開発、すなわちソフトウェア部品等の生成物に着目した開発作業が行なわれている。このような開発形態においては従来のプロセスモデルは適用が困難であり、また進捗状況の把握も困難である。そこで本研究では開発手順ではなく、開発時におけるさまざまなプロダクトやツール、資源等をオブジェクトとして捉え、その集合としてプロセスを捉えるモデルを提案する。またそれに基づいた開発管理システムの設計を行う。

2 提案するプロセスモデル

オブジェクト O は、属性の集合 A とメソッドの集合 M の2字組 (A, M) である。属性はオブジェクトについての情報であり、メソッドはオブジェクトに対して許された操作である。オブジェクトには一意に認識するための名前が付けられている。

属性 $a \in A$ は属性名 t と属性値 v の2字組 (t, v) である。属性名は一つのオブジェクトにおいて一意に定められる、つまり任意のオブジェクト $O = (A, M)$ において、 $\forall t [\forall v, v' (t, v) \in A \text{ かつ } (t, v') \in A \rightarrow v = v']$ である。

属性名のうち、“elements” という属性名はその属性値

として他のオブジェクトの名前のリスト $\{O_1, O_2, \dots, O_n\}$ を取る。あるオブジェクト $O = (A, M)$ の属性として (“elements”, $\{O_1, O_2, \dots, O_k\}$) $\in A$ があるとき、 $O = \{O_1, O_2, \dots, O_k\}$ と書く。このとき、 O_i ($1 \leq i \leq k$) は O の要素である、あるいは O は O_i を要素として持つと言い、 $O_i \in O$ と書く。更に、オブジェクト O, O' について、 O' が要素として持つ全てのオブジェクトを O が要素として全て持つならば、 O は O' を含む、あるいは O' は O に含まれると言い、 $O' \subseteq O$ と書く、つまり $O' \subseteq O \iff \forall O_i (O_i \in O' \rightarrow O_i \in O)$ である。

全てのオブジェクトの集まりを \mathcal{O} としたとき、メソッド $m \in M$ とは、メソッド名 s と、 \mathcal{O} から \mathcal{O} への写像 f の2字組 (s, f) である。メソッド名は一つのオブジェクトにおいて一意に定められる、つまり任意のオブジェクト $O = (A, M)$ について $\forall s [\forall f, g (s, f) \in M \text{ かつ } (s, g) \in M \rightarrow f = g]$ である。オブジェクトは全て、既に存在するオブジェクトにより作られる。ただし、Root というオブジェクトが初めから存在しているものとする。新しく作られたオブジェクトは作り手となったオブジェクトの要素となる、つまり作り手のオブジェクトの属性 “elements” の値であるオブジェクト名リストに新しいオブジェクトの名前が加えられる。

$O_i = (A_i, M_i) \in O = (A, M)$ のとき、 $\forall t [(t, v) \in A \rightarrow (t, v') \in A_i]$ かつ $\forall s [(s, f) \in M \rightarrow (s, f') \in M_i]$ であるとする。これを O_i が O の属性名とメソッド名を継承すると言う。これにより、あるオブジェクト O からオブジェクト O' が生成されると、 O の属性名とメソッド名を O' は継承することになる。

3 モデルに基づいた開発管理システム

上記のプロセスモデルに基づいた、ソフトウェア開発管理システムについて述べる。

3.1 システムの概略

本システムは主に4つの部品より構成される。

- リポジトリ
- イベントハンドラ
- メソッドエグゼキュータ

Software Development Management System Based on Objects.
Makoto Oshita, Makoto Matsushita, Hajimu Iida, Katsuro Inoue
Osaka University

1-3 Machikaneyama, Toyonaka, Osaka 560, Japan

● ユーザインターフェース

リポジトリは、モデルの O にあたるもので、UNIX のファイルシステムを用いて実装する。まず Repository というディレクトリがある。その下にオブジェクト毎のディレクトリ、更にその下に Attributes, Methods というディレクトリがある。 $(O = (A, M))$ の O がオブジェクト毎のディレクトリ、 A が Attributes, M が Methods にあたる)。Attributes の下に属性名ごとのファイルを作り、属性値を記述する。Methods の下にもメソッド名毎のファイルを作る。これは何らかのプログラム言語で書かれたプログラムであり、写像に対応する。メソッドの実行とはそれに対応したプログラムの実行である。

オブジェクトの実体がどこにあるかはそのオブジェクトの属性により知ることが出来る。よってオブジェクトの実体を各計算機に分散させることができる。

リポジトリは実世界をファイルシステムにマッピングしたものである。リポジトリの中のオブジェクトの属性を参照する事で実世界の情報を得、リポジトリの中のオブジェクトを操作する事で実世界に対する操作を行う。

イベントハンドラ (EH) はリポジトリへのアクセスや、メソッド呼び出し処理などを行う。メソッド呼び出しはまず EH に伝えられ、EH が直接リポジトリを検索してメソッドのプログラムを取得し、後述するメソッドエグゼキュタに実行を依頼する。またメソッド呼び出し履歴の記録も行う。履歴はそれぞれのオブジェクトの属性として、リポジトリに記録される。

イベントハンドラからの依頼を受けてメソッドを実行するのがメソッドエグゼキュタ (MX) である。一般にリポジトリと EH が存在する計算機と、メソッドが実行される計算機は別であるため、EH がメソッドのプログラムを直接には実行できない。メソッドを実行したい計算機上であらかじめ MX を用意しておき、それにメソッドの実行を依頼するという形になる。

システムを利用するためのユーザインターフェース (UI) には Web ブラウザを用いる。オブジェクト毎にページを用意し、属性が参照できるようにする。またメソッド起動用のボタンを配置し、そのボタンが押されるとメソッド呼び出しが起こったことが EH に知らされる。ページは EH がリポジトリを参照しながら生成する。

3.2 システムの動作例

ある C のコード `code.c` をコンパイルするという例で説明する (図 1)。`code.c` が存在する計算機、リポジトリと EH がある計算機、作業をしている計算機は別であるとする。

1. オブジェクト `code.c` に対応した Web ページ上の、メソッド `compile` を起動するためのボタンを作業者が押す。

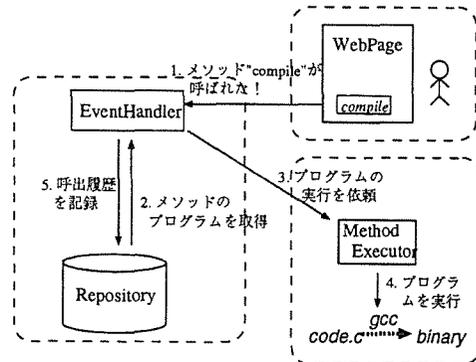


図 1: システムの動作例

するとメソッド `compile` が呼ばれたことが EH に知らされる。

2. EH はリポジトリを検索し、オブジェクト `code.c` のメソッド `compile` の内容を取り出す。このとき、`code.c` が存在するパス名の取得なども行う。ここではメソッド `compile` の内容が `gcc -o binary code.c` だとする。
3. `code.c` が存在する計算機の MX にメソッドの実行を依頼する。
4. MX はメソッドを実行する。即ち、`gcc` を起動して `code.c` から `binary` を生成する。
5. EH はメソッド呼び出し記録を取る。これが作業履歴となり、開発管理者が進捗状況を判断する材料となる。

4 まとめと今後の課題

プロダクトやツールに着目したソフトウェアプロセスモデルを提案した。またそれに基づくシステムの設計を行った。本モデルではオブジェクトの概念を導入することでプロダクトやツール、あるいはそれらの集合といったものを統一的に扱う。またオブジェクトに対する操作をメソッドという形で限定するので、作業履歴が自動的に記録しやすい。

現在、システムの試作を行っている。今後の課題としては、システムを完成させ有効性を評価することが挙げられる。また、リポジトリをオブジェクト指向データベース [3] によって実装することも検討している。

参考文献

- [1] 落水浩一郎: “ソフトウェアプロセスに関する研究の概要”, 情報処理, Vol.36, No.5, pp.379-391(1995).
- [2] Curtis, B., Kellner, M. and Over, J.: “Process Modeling”, *Communication of the ACM*, Vol.35, No.9, pp.75-90(1992).
- [3] 永田元康: “オブジェクト指向データベースシステム”, 森北出版 (1995).