

6Z-03 ソフトウェア開発プロセス設計・適応支援システム

—プロセス計測・評価支援システムの実装—

菅野 亨太 松下 壮一 野中 誠 東 基衛

早稲田大学大学院理工学研究科 経営システム工学専門分野

1 はじめに

ソフトウェアプロセスをメトリクスに基づいて評価・分析し、改善していくためには、開発プロセスの計測が必要になる。

しかしながら、開発プロセスや開発ツールは組織毎、プロジェクト毎に異なるため、計測の自動化および状態の可視化を単純に行うことは難しい。

本研究では、多様な開発プロセスと開発ツールに適応可能なソフトウェアプロセス計測・評価支援環境を、計測・メトリクス適用過程のモデル化を行うことによって実現した。

2 アプローチ

プロセスという概念を中心におき、コンピュータによる支援を目的としたシステムはPSEE(Process-centered Software Engineering Environment)と呼ばれる。SPADE【1】のように、これらのシステム実装はいくつか存在するが、そのほとんどは、プロセスモデルを手続きとして形式言語で記述し、実際にシステム内でプログラムとして実行するという指向で設計されている。しかし、厳密に作業手続きや状態遷移を記述することは困難であり、実際の開発プロセスとの同期が失われた際の対処が困難であるといった問題も発生しうる。

本研究では、プロセスを状態を持つデータオブジェクトとしてのみ捉え、プロセスの状態の計測と分析という目的に適したプロセス計測・評価支援システムを実装する。

3 システム化の課題

システム化の課題は以下に述べる3点である。それぞれについて述べる。

①イベントモデルの導入による、開発ツール、計測システム本体とプロセスモデルの相互独立性の向上

開発において用いられるツールは組織毎に異なる

ため、開発ツールと計測システム本体は疎に結合されていることが望ましい。

②計測に関する概念および実体のコンポーネント化

計測にあたって使用する帳票やメトリクスはソフトウェアプロセスの定義や目標によって異なるため、計測システム内で扱うデータ構造は一意に固定できない。

従って、システム構築を行うには、計測過程ならびに計測・メトリクス適用に関わるオブジェクトを抽象化したコンポーネントフレームワークと、それに基づく基盤サービスの実装が必要である。

③コンポーネントの再利用の支援

②において述べたコンポーネントは作成され、再利用されることが想定されるが、その際、コンポーネント間の依存性を解決し、システムとして一貫性を持たせる必要がある。また、システム構築のプロセスが明らかにされている必要がある。

4 システムアーキテクチャ及び動作

システムは稼動する環境に依存しないフレームの部分と、環境や計測目的に応じて変更されるコンポーネント部分によって構成される。下図にシステムの構造を示す。

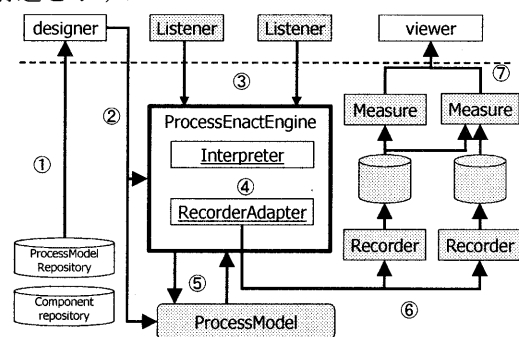


図 1 システム構造図

図中灰色部分がコンポーネントを用いて稼働環境ごとに変更を加える部分に該当する。また、各コンポーネント間の通信にはイベントが用いられる。これらについては5章において示す。

動作概要について説明する。

①～②：計測対象を決定し、必要なコンポーネント

をリポジトリからロードしてシステムを構成する。
 ③～⑥:開発ツールから発生するイベントを Listener を経由して受け取り, パラメータに応じて, ProcessEvent へと変換する. 得られた ProcessEvent に応じて, Engine は ProcessModel の状態を更新する. その後, ProcessEvent はそのイベントを受け取ることが定義されている Recorder へと渡され, Recorder は自身に定義されているフォーマットおよび, 記録方法に従って記録を生成する.
 ⑦:現在のプロセス状態および Measure コンポーネントの定義に基づいて, 計測値およびその表現が提供される.

5 構成コンポーネント及びイベントの解説

本システムの構築にあたって, いくつかのコンポーネントが組織やプロジェクトの計測目的に沿って開発される必要がある. 以下にそのうち, 中心となる Measure, Recorder, ProcessEvent コンポーネントについて述べる. 図2に3つのコンポーネントの基底クラスの図を示す. コンポーネント作成にあたってはこれらのサブクラスを定義することで行う. 以下にそれぞれについて解説する.

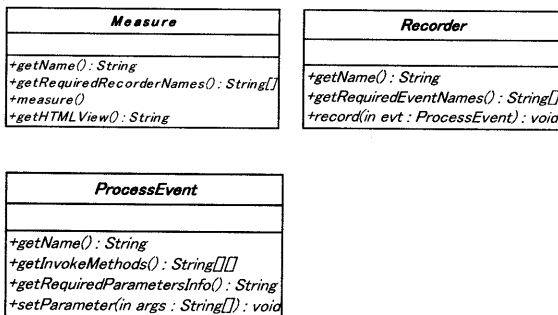


図2 構成コンポーネントの基底クラス概要

5.1 Measure クラス

メトリクス適用及び可視化のための表現について抽象化した基底クラスである. コンポーネント開発の際にはサブクラスを定義し, 名前, 必要な帳票を生成する Recorder コンポーネントの名前, 計測の手続き, 表現形式, を定義するメソッドをオーバーライドする.

5.2 Recorder クラス

計測データを記録した帳票を生成するコンポーネントが Recorder コンポーネントであり, Recorder クラスはその基底クラスである. コンポーネント作成

の際には, Recorder クラスのサブクラスを定義し, 名前, レコードを生成するために必要な ProcessEvent の型, レコード生成・記録の手続き, を定義する各メソッドをオーバーライドする.

5.3 ProcessEvent クラス

ProcessEvent クラスはシステム内での情報伝達手段であるイベントオブジェクトを定義するための基底クラスであり, サブクラスを定義し, 必要に応じて属性を定義し, 名前, プロセスオブジェクトへのび出し操作及び引数, 等を定義するメソッドをオーバーライドすることでシステム中でイベントオブジェクトとして使用することが可能になる.

6 システムの構築プロセス

下図は, メトリクス設定から各コンポーネントの作成, システム設定までの手順を示したものである. また, 新規ツールの追加の際にはリスナを作成しなければならない.

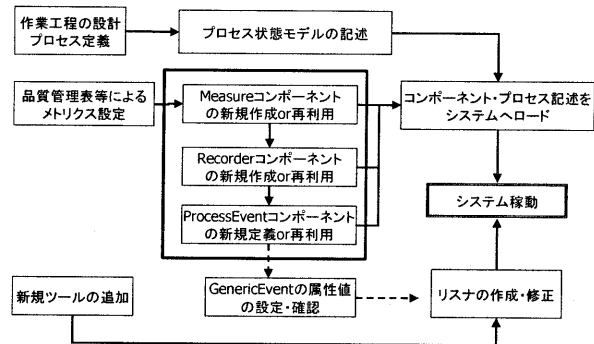


図3 システム構築手順図

7 実装および評価・結論

以上にに基づきプロトタイプを実装した. 言語及び実行環境としては JDK1.3 を用いた. なお, 分散オブジェクト技術として CORBA を用い, ORB には JDK 付属の Transient NameServer を用いた. 実装したクラス数は約 40 個, またコンポーネントのサンプル実装を 10 個程度作成した. 結果, 3 の課題 3 つを満たす実装を確認できた. 一方, 一貫性の確認のため開発ツール-システム間で通信されるパラメータの確認が必要になるという問題が指摘された. 今後は, プロセス記述及び操作の仕様確定が課題となる.

- [1] S.Bandinelli, A. Fuggetta: Computational reflection in software process modeling, The SLANG approach, Proc. Of the 15th International Conference on Software Engineering, IEEE Computer Society Press, pp144-154, 1993