

Title	共同ソフトウェア開発における開発者の依存関係に関する研究
Author(s)	周, 翼
Citation	
Issue Date	2003-09
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/1766
Rights	
Description	Supervisor:落水 浩一郎, 情報科学研究科, 修士

共同ソフトウェア開発における 開発者の依存関係に関する研究

周 翼 (110202)

北陸先端科学技術大学院大学 情報科学研究科

2003 年 8 月 15 日

キーワード: Rational Unified Process(RUP), Coplien の組織パターン (organizational patterns), 役割の管理モデル, 成果物、コミュニケーションパス、プロセスモデル .

1 背景と目的

共同ソフトウェア開発では、開発者は決められた開発プロセスに従い、作業を行わなければならない。ラショナル統一プロセス (Rational Unified Process:RUP) は、業界における様々なプラクティスを反映し、成果物中心の開発方法論であり、開発現場によく使われている。ユースケース駆動、アーキテクチャ中心、反復型開発は、RUP プロセスの特徴として挙げられる。

ソフトウェア開発においては、開発者が従わなければならない開発プロセスに加えて、開発者間のコミュニケーションが重要である。しかし、RUP プロセスは、成果物に注目し、成果物を作成するための作業内容と作業内容の順序を定義しているが、コミュニケーションについてはふれていない。

Coplien による組織パターンはソフトウェア開発を進める上で開発の進め方や組織構造のあり方についてコミュニケーションパスの構成法の観点から記述されたパターンである。そこで、本研究の目的は、RUP プロセスと Coplien の組織パターンを結合したプロセスモデルを定義し、プロセスモデルを用いて成果物中心の活動とコミュニケーションを融合する方式を提案する。

2 RUP と Coplien の組織パターンの結合

RUP プロセスと Coplien の組織パターンを結合するために、Coplien の組織パターン構成の基本 (役割とコミュニケーション) に注目する。J.Coplien は、密接に関係するソフトウェアエンジニアリング活動 (アクティビティ) の集合として役割を定義し、コミュニケーションパスを (役割の要素である) アクティビティの意味的結合として定義する。

RUP プロセスも役割の定義を利用している。RUP プロセスの役割を抽象し、役割のメタモデルを定義する。役割のパッケージには、開発者と開発者が責任を担う作業内容がある。作業内容は、密接に関係するアクティビティの集合である。ゆえに、役割を用いて RUP プロセスと Coplien の組織パターンの結合ができるを考える。

役割を用いて RUP プロセスと Coplien の組織パターンを結合するために、RUP プロセスの各役割をまとめ、役割の管理モデルを定義する。結果として、RUP プロセスの各役割をシステムエンジニア、アーキテクト、システム統合者、ユーザーインターフェイス、ユースケースエンジニア、コンポーネントエンジニアの 6 種類に分類する。役割の成果物および成果物に対する責任を用いて役割の作業内容を表現する。各役割の成果物間の関係に基づき各役割間の実装関係、参照関係、トレース関係の 3 種類に分類する。さらに、RUP プロセスの各役割と Coplien の組織パターンの役割を比較し、同一の役割であると判断すれば、結合を行う。結果として、

- アーキテクトとパターン 13
- ユースケースエンジニアとパターン 11
- コンポーネントエンジニアとパターン 11
- テスト設計者とパターン 19、パターン 18
- テスト担当者とパターン 19
- システム分析者とパターン 11、パターン 25
- システム統合者とパターン 11

を結合できた。

さらに、Coplien による組織パターンのコミュニケーション定義に基づき、以上の各結合を用いて各役割間にコミュニケーションパスを付ける。

3 RUP と Coplien の組織パターンを結合したプロセスモデル

プロセスモデルを定義するために落水によるチーム構造モデルを参照する。プロセスモデルの要件を以下のようにとらえている。

- RUP における個人の責任範囲の定義
- RUP における成果物中心の活動とコミュニケーションの融合

個人の責任範囲を定義するために、役割を用いる。RUP プロセスの各役割に基づきシステム分析者、ユースケース定義者、アーキテクト、システム統合者、ユーザーインターフェイスデザイナー、ユースケースエンジニア、コンポーネントエンジニア、テスト設計

者、テスト担当者、顧客、プロジェクトマネージャを定義する。成果物中心の活動とコミュニケーションの融合を実現するために、役割の責任を成果物関連責任とコミュニケーション責任に分類する。RUP プロセスの各役割の成果物関連活動を用いて成果物関連責任を定義し、前章で発見した役割間のコミュニケーションパスを用いて、コミュニケーション責任を定義する。

コミュニケーションパスオブジェクトは、コミュニケーションを行う役割とコミュニケーションの概要によって構成される。共有中間成果物に対する操作を用いて、コミュニケーションの概要を定義する。

4 有効性の確認

本の具体例を本手法で再生成した結果と本に記述している結果を比較した。成果物を生成する順序はほとんど一致したが、一部、一致しないものもあった。例えば、本手法ではユースケースを洗い出すと同時に、統合テストのテストケースを作成するが、本の具体例では、コードを作成した後にテストケースを作成する。これは、Coplien の組織パターンに基づき役割のコミュニケーション責任が定義されたからである。テストの開発は時間がかかり、システムが完成したからといってすぐに始められるものではない。ゆえに、これは本手法のメリットである。

5 まとめと今後の課題

本論文は、RUP プロセスと Coplien による組織パターンを結合したプロセスモデルを定義し、プロセスモデルを利用して成果物中心の活動とコミュニケーションを融合する方式を提案した。まず、Coplien による組織パターン構成の基本に着目し、RUP の役割の管理モデルを定義し、役割の定義を用いて Coplien の組織パターンと RUP プロセスの結合を実現した。つぎに、落水によるチーム構造モデルに基づきプロセスモデルを定義し、RUP プロセスの成果物中心活動とコミュニケーション関連活動を融合した。この手法を具体例に適用し、有効性を示した。

今後の課題として、開発現場での実証実験による評価を行うことで、プロセスモデルの精度を高める必要がある。また、成果物の詳細情報を表現するために、成果物オブジェクトを導入する必要がある。