

管理方針への適応性を持つ構成管理システムのフレームワーク

藤 枝 和 宏[†]

Eclipse が流行している理由の一つは、さまざまなミドルウェアや開発ツールへの適応性にある。構成管理システムもまた適応性を持つべきである。ここではその理由と、それを実現する構成管理システムのフレームワークについて述べる。

Frameworks of SCM Systems Adaptable to Different Management Policies

KAZUHIRO FUJIEDA[†]

One of the reasons why Eclipse is very popular is its adaptability to different target applications or development tools. Software configuration management (SCM) systems should also have adaptability. In this position paper, I discuss its reasoning and frameworks of SCM systems to realize it.

1. はじめに

ソフトウェア開発ツールや開発環境は、高い適応性を備えていなければならない。開発プロジェクト間はもちろん、同一プロジェクト内においても、ツールや環境が適応すべきさまざまな相違が存在する。これらの相違には、エディタのキー割り当てのような個人の嗜好から、開発に用いるプログラミング言語やミドルウェア、共同作業の形態や方針までが含まれる。単なるオプションやパラメータの設定では適応の幅が限られるため、これらの相違には適応できない。より高い適応性の実現には、専用あるいは汎用の言語によるプログラミングによる適応をサポートする必要がある。

開発環境として Eclipse が流行している理由の一つに、よく設計されたプラットフォーム・アーキテクチャと SDK による高い適応性がある。Java 言語でプラグインを記述して Eclipse に組み込むことで、さまざまなミドルウェアやツールに適応させることができる。たとえば、Eclipse がデフォルトでサポートしている構成管理システムは CVS だが、プラグインを記述するか既存のものを探して組み込むことで、他の構成管理システムも利用できる。

さまざまなツールやミドルウェアをサポートしなければならない統合開発環境では、このような適応性は必須であり、その実現に必要なフレームワークは古く

から研究されてきた。しかし、統合されるツールそのものが持つべき適応性やそのためのフレームワークは、言語処理系やエディタなどを除くと、あまり取り組まれていない。成果物の管理に主要な役割を果たす構成管理システムについても、その管理方針は開発プロジェクトによって異なるので適応性を持つべきである。しかし、そのための取り組みは十分とはいえない。

2. 構成管理システムに求められる適応性

構成管理には成果物の変更管理に関するさまざまな活動が含まれており、その方法や方針は開発プロジェクトごとに異なっている。たとえば、変更時のロックの方法、ベースラインやブランチの利用方針、変更を受理する際の条件やプロセスの相違のほかに、利用するビルドシステム、テストシステムやバグトラッキングシステムの相違がある。構成管理システムは、これらの相違への適応性を持つ必要がある。

多くの構成管理システムは、特定のイベントの発生時に実行されるプログラムを登録可能にしたり、リポジトリにアクセスするプログラムを書くための SDK を提供したりしている。適応性の実現にはこの両者が必要だが、ほとんどのシステムではいずれかしか提供していない。分散ソフトウェア開発では、ネットワークを介して共同作業を行う複数の開発者間で、変更管理の方針を共有する必要がある。この共有を容易にするために、構成管理システムの適応性はサーバ側で提供する必要があるが、クライアント側でしか提供していないシステムが多い。

[†] 北陸先端科学技術大学院大学
Japan Advanced Institute of Science and Technology

変更管理方針の異なる複数のチームによる分散ソフトウェア開発では、方針の衝突とそれにもなう交渉のコストを減らすために、構成管理システムのリポジトリを各チームで分ける場合がある。リポジトリを分けることで、全チームで共有する場合と比べて構成管理システムの可用性を高めることもできる。リポジトリを分けたとしても、各チームは成果物を共有する必要があり、互いの変更も適切に反映されなければならない。このリポジトリの分散・協調形態もまた、開発プロジェクトごとに異なっており、これをサポートする構成管理システムは適応性を提供する必要がある。

既存の構成管理システムでは、特定のモデルに基づいたリポジトリの分散形態のみをサポートしており、適応性はなく、それぞれに可能な分散・協調形態に制約がある。たとえば ClearCase Multisite には、リポジトリ全体を複製して共有するという制約のほかに、リポジトリごとに変更可能なブランチを割り当てなければならない、リポジトリ間の変更の反映がブッシュモデルでしか行えないといった制約がある。

3. 適応性のためのフレームワーク

構成管理システムの適応性に関する取り組みには、文献 1) や文献 2) がある。前者は、構成管理システムのプリミティブを提供し、それをういた新たな構成管理システムの構築を可能にしている。後者は、制約モジュールを組み込み可能な汎用サーバと、アクションモジュールを組み込み可能な汎用クライアントで適応性を実現しようとしている。いずれの文献も、開発プロジェクトごとの構成管理の方針の相違に言及している。しかし、実現しているのはそのための適応性ではなく、まったく新しい構成管理システムを実装するためのフレームワークである。リポジトリの分散モデルについては、前者はプリミティブの下に隠蔽されており適応性がなく、後者は制約モジュールの一つとして実現できるとしているが、既存のシステムの分散モデルを見る限りまったく現実性がない。

現実のソフトウェア開発を見る限り、構成管理システムの提供する管理モデルや支援機能を大きく挿げ替えるような適応性は、もはや必要ではない。既存の構成管理システムの提供する基本的な管理モデルや機能は、多くの開発プロジェクトの必要性を満たしている。構成管理の方針やプロセスに関わる支援機能については、プロジェクトごとに要件が異なるため、これを提供しないシステムが広く受け入れられている。その代わりに、これらのシステムでは、自然言語で記述された方針やプロセスを開発者が手動で履行することにな

る。システムによっては、特定のイベントに対応して実行されるプログラムで、これをある程度支援できる。しかし、プログラムの実行されるイベントやプログラムに渡される情報などに制約があり、必要な支援の記述が難しいことが多い。

著者らは、既存の構成管理システムの基本的な機能は触らずに、支援機能の記述を容易にするオブジェクト指向フレームワークをサーバ側に導入するアプローチを提案している³⁾。これは実際には、CVS のサーバとクライアントの間に、オブジェクト指向フレームワークを持ち拡張可能なプロキシをはさむ形で実現している。リポジトリの分散については、多くの開発プロジェクトの必要性を満たす基本的なモデルは、今のところ存在しない。そこで著者らは、リポジトリ間で成果物と変更履歴を複製・交換するためのプリミティブを提供し、開発プロジェクトにあわせてリポジトリの分散モデルを、このプリミティブと先のサーバ側の適応性を用いて記述できるフレームワークの開発を進めている。

参 考 文 献

- 1) van der Hoek, A., Carzaniga, A., Heimbgner, D. and Wolf, A.: A Testbed for Configuration Management Policy Programming, *IEEE Transactions on Software Engineering*, Vol. 28, No. 1, pp. 79–99 (2002).
- 2) van der Lingen, R. and van der Hoek, A.: Dissecting Configuration Management Policies, *Software Configuration Management: ICSE Workshops SCM 2001 and SCM 2003*, LNCS, No. 2649, Springer Verlag, pp. 177–190 (2003).
- 3) 早坂良, 藤枝和宏, 落水浩一郎: プロセス支援機能を組み込み可能な CVS プロキシ構成法, 電子情報通信学会技術研究報告 SS2002-51, pp. 19–24 (2003).