

1 研究テーマの背景

1.1 分散開発の高まり

現在ソフトウェア開発環境は大きく変わりつつある。つい最近までは、ソフトウェアは企業や団体など1カ所で集中的に開発されていた。

しかし、近年になり、オープンソースウェアの開発のようにネットワーク上で知り合った、顔を知らない多くの技術者たちが共同で1つのソフトウェアを開発するという機会が増えてきている。また、企業に所属する開発者も在宅での作業が可能になった。これは、ネットワークインフラが整備され、個人でも低価格で安定した高速回線接続サービスの恩恵を受けることができるようになったためである。

このように、従来のソフトウェア開発環境に加え、分散開発環境という新しい場が生まれている。

1.2 ソフトウェア開発管理支援システム

従来からソフトウェア開発現場においては、ソースコードの管理などを支援するツールの導入や、開発要員の間での情報のやりとりのためのメーリングリストなどが構築されていた。分散環境における開発においてもこれらのツールは使用されている。また、それに加えてネットワークインタフェースを備えたツールが開発、使用されている。ソースコードなどのバージョン管理を行う CVS^{*1}やバグ管理システムの Bugzilla^{*2}などはその代表的なツールである。近年では、WWW ベースでテキストの共同編集作業を支援する Wiki と呼ばれるツールも、ドキュメントの編纂段階で導入されつつある。

このように、ソフトウェア開発時、特に分散開発時において多くのツール、ソフトウェアが利用されている。以降では、これらのツールを総称して、「ソフトウェア開発管理支援システム」と呼ぶことにする。

1.3 既存ツールの問題点

ソフトウェア開発管理支援システムは、各ツールの完成度（安定性、処理速度）は実用に耐えうるレベルに達している。しかしながら、現状においてこれらのツールの間で連携して動作するように設計されている例は少ない。

また、オープンソースソフトウェアなどの、自

由に技術者が参加できる開発環境の場合は、さらに Web インターフェースの充実した開発管理支援ツールが必要となってくる。この場合、現状のツールにおいてリポジトリ内のファイル閲覧機能を持つものもあるが、あくまで補助的な機能として提供されているもので、Web ページの一つとして考える場合には機能性・閲覧性ともに不十分である。

さらには、現状の支援ツール類は、高機能・多機能な反面、プログラマ・SE 以外のツール類を頻繁に使うことのない開発要員にはわかりにくいソフトウェア構成となっている。場合によっては、この多機能性が新たな開発要員参入の壁となる可能性がある。

2 研究テーマの目的

先に挙げた問題を解決するため、本稿では新しいソフトウェア開発管理支援システムを提案する。具体的には、以下に掲げる3点の機能を備えるシステムの開発を目的とする。

2.1 各ツール間での連携動作の強化

各ツール間での連携をあらかじめ考慮する。これにより、開発者の開発作業以外のコスト（ソースコードのコミット、バグ対応レポートの記入など）を最小限にとどめることができると予測される。

2.2 導入・習得の簡便性

クライアント側のシステムとして、Web や簡単なインターフェースなど、ユーザーが使いやすいインターフェースを容易に構成することのできるシステムを構築する。

2.3 Web での展開

ソフトウェア開発管理支援システムに開発中のソフトウェアを Web にて公開するという視点も取り入れる。すなわち、フリーソフトウェアと同じようにソフトウェアを広く一般に公開し、様々な人にテスターとなってもらえる環境を容易に構築できるようなシステムを提供する。また、プロジェクトの Web サイトを容易に構築できる仕組みを備える。

*1 <http://www.cvshome.org/>

*2 <http://www.bugzilla.org/>

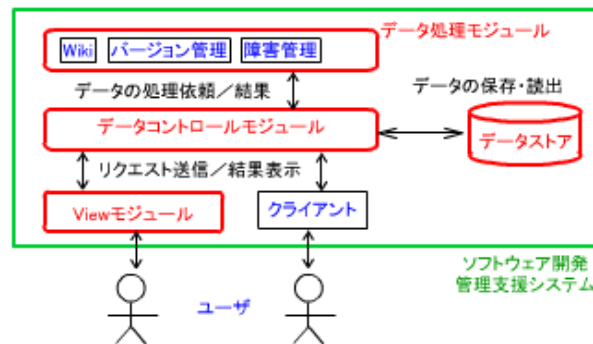


図1 提案するシステムの構成図

3 アプローチ

この章では、本提案を実現するためのアプローチ、構成について述べる。

本提案のシステム構成を図1に示す。このシステムは以下の4つの要素により構成される。

データ処理モジュール バージョン管理や障害管理など、データの処理を行う。

データコントロールモジュール ユーザ側や管理・支援モジュールからの要求に従い、データを各処理モジュールやデータストアに振り分ける。

データストア メタデータや障害管理情報など、開発作業で実際に発生したデータを格納する。

Viewモジュール Web上での閲覧画面を提供する。

各データ処理モジュールは、データストアに格納されているデータにデータコントロールモジュールを介してアクセスすることができる。これにより、各モジュール間で相互にメッセージパッシングを行うことができ連携を高めることができる。例を挙げると、特定のモジュールに関する障害が報告されているとき、そのモジュールをアップデート(ダウンロード)する際に付随情報としてその障害報告も同時に受け取るなどのこともできる。

Web閲覧に関するインターフェースはViewモジュールとして分割する。これにより、実際のデータ処理と閲覧部分を切り離すことができ、Web用インターフェースを容易に構築、提供することが可能となる。

プロジェクトの統計情報などを収集するなど、支援管理の際に新たなモジュールが必要になったときは、システム管理者が管理・支援モジュールとして新たなモジュールを登録する。さらに、Wikiの

ようなWebインターフェースを利用するものは、Viewモジュールにも登録を行う。このように各モジュールを管理することで、保守性を高めることが出来る。また、実際のデータ処理とデータの閲覧部分を分離することで、閲覧のための機能追加を容易に行うことができる。

4 まとめ

本稿ではソフトウェア開発を管理・支援するシステムの提案を行った。このシステムによりソフトウェア開発、特に分散開発における開発管理の見通しがよくなることが期待できる。また、システム利用のための習得コストを低くしわかりやすいインターフェースを提供することで、ソフトウェア開発者以外の開発要員も用意にプロジェクトに参入できる土台を作ることができる。

一方で、今回の提案ではソフトウェアデータの保守性・堅牢性については触れなかった。この点は開発支援を行う際に最も重要な要素である。この点に関しても検討を行う必要があるだろう。

入学した際には、このシステムの開発を通じて、ソフトウェア開発、特に近年盛んとなっている分散開発に適したソフトウェアプロセスのあり方、および開発要員のコミュニケーション・コラボレーションのあり方を模索していきたいと思っている。また、オープンソースソフトウェアの開発などでは議論される機会の少ない、品質管理、ドキュメント作成・管理の効果的かつ体系的な手法についても研究をしていきたいと思っている。