

ソフトウェア開発における過去の開発履歴情報を用いた 保守作業軽減手法の提案

山田 洋明[†] 樋山 淳雄[†]

東京学芸大学 教育学部 情報教育専攻[†]

1. 研究の背景・目標

近年、情報技術の進化やユーザからの多様な要望に伴い、ソフトウェア開発は大規模化、複雑化してきている。しかしその一方、システムの開発期間は短くなっており、テストや保守作業にかけられる時間が短くなっているのが現状である。時間をかけてもすべてのバグを見つけることができない状況であるのに、このような状況では重大なバグでさえも見つけることができない可能性がある。そのような中、現在のソフトウェア開発では、版管理システムを用いることが一般的となっている。版管理システムでは一つのドキュメントに対して、開発履歴を版として残しておくことができる。その情報をうまく利用することができれば保守作業の場面においても効率よく障害情報を発見することができると考えられる。しかし、版管理システムから自分が有益な情報を見つけるためには、どの情報が自分に有益なのかを探し選別しなければならないため、多大なコスト(時間)がかかってしまう。それに加え、版管理システムの情報だけではなくそれに関連するドキュメントや情報がさまざまな場所に散在しており、その情報を見つけ出すことはもっと困難である。

そこで本論文では、ソフトウェア開発の保守の場面に着目し、版管理システムの情報と障害情報とを関連付け、開発者が現在開発中のソースコードから、過去のソースコードに潜んでいた障害情報やそれを解決するための情報を提供できるシステムを構築し、ソフトウェアの保守作業を軽減する手法を提案する。

2. 関連研究

Davor Cubranic らは Hipikat というシステムを構築している[1]。Hipikat では成果物の識別子をシステムに送ることにより、システムが推薦す

るドキュメント(障害情報の要約など)をユーザに提示するとともに、そのドキュメントを推薦した理由と確信度を提示する。Hipikat では、成果物の情報(チェックインコメントなど)をもとにドキュメントを推薦するため、ソースコード自体の内容に関係する情報は発見しづらいと考えられる。

Ye らは CodeBroker というシステムを構築している[5]。CodeBroker は Emacs のバックグラウンドで常時動いており、プログラムの Doc コメント、シグニチャ宣言の情報を解析することによりコンポーネントの必要性を予測し提供するとともに、そのコンポーネントの類似度なども付加する。この研究でも、ソースコード自体の内容に関係している情報は発見しづらいと考えられる。

3. 保守作業軽減手法

3.1 システム構成

本研究では、版管理システムに蓄積されているソースコード情報と、障害情報とを関連付け、開発者が開発中のソースコードをシステムに検索をかけることにより、過去のソースコードに潜んでいた障害情報やそれを解決するための情報を提示する。

システム構成を図 1 に示す。システムは障害情報を登録する Web アプリケーションと、あるソースコードに対して、それに類似する既存のソースコードを検索し開発者に提供する統合開発環境 (IDE: Integrated Development Environment) Eclipse[6] の plug-in から構成される。類似検索の手法は次項で述べる。IDE 内で情報を提供することにより、保守作業の効率化を図ることができる。

3.2 類似検索アプローチ

過去のソースコードと現在開発中のソースコードとの類似度を測るために、Levenshtein 距離の概念を応用したコードクローンの検索手法を用いる[2]。Levenshtein 距離とはある二つの文字列の一方の文字列を挿入、削除、置換を行いも

A Proposal of Maintenance Work Reduction Technique Using Development History Information in Software Development
[†]Hiroaki YAMADA and Atsuo HAZEYAMA (Department of Information Education, School of Education, Tokyo Gakugei University)

う一方の文字列に変形するための最小回数として与えられる。たとえば、“sitting”と“kitten”であれば“s”を“k”に置換し、次に“i”を“e”に置換、最後に“g”を削除すると同一の文字列となり、Levenshtein 距離は 3 となる。

コードクローンの検索手法を利用する理由として以下がある。

- 1) プログラム言語初学者は、プログラムを作成するとき授業の配布資料やインターネット上を探しその情報を参考にするため、類似したソースコードがしやすい
- 2) 近年のソフトウェア開発では、信頼性の観点から、ゼロからソースコードを開発するよりも、既存のソースコードをコピーアンドペーストすることにより再利用することが多い[4]

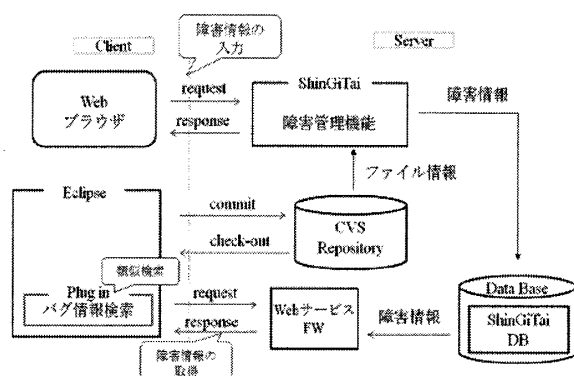


図1 システム構成図

3.3 システムの機能概要

システムは障害情報登録機能と障害情報検索機能からなる。障害情報登録機能では、障害情報を入力してソースコードとの関連付けを行い保存する。障害情報検索機能では、バグを探したいソースコードをシステムに検索をかける。

3.3.1 障害情報登録機能

障害情報登録機能では、著者らの研究室で開発しているソフトウェア開発グループ演習環境「ShinGiTai」[3]の障害管理機能を用いて障害情報を記述するとともに、CVSのリポジトリ内にあるソースコード名とソースコードの版番号を入力することにより障害情報と版管理システムの情報を関連付け、データベースに格納する。障害情報としては、概要、障害の重要度、詳細、原因の分類、原因、対処内容などを記述する。他にもバグ修正中に交わされたコミュニケーションの内容も関連付けることとする。このことにより、障害に関するより詳細な情報を得るこ

とができると考えられる。

3.3.2 障害情報検索機能

障害情報検索機能ではEclipseのplug-inで障害情報検索を行う。Eclipseのplug-inから、開発者が開発中のソースコードを選択する。その後、3.3.1項で登録されたソースコード情報とそれに関連する障害情報をデータベースから取得する。そして、選択されたソースコードと取得したソースコードとの類似度を計算する(その手法は3.2項で述べた通りである)。類似したソースコード情報が発見された場合、そのソースコードに関連のある取得した情報を一覧としてEclipse内に表示させる。一覧表示された情報から自分が欲しいと思われる情報を選択することによりその情報の詳細情報を閲覧することができる。

4. まとめ

本研究では、ソフトウェア開発におけるソースコードとそれに関連する障害情報とを関連付け、その情報を活用することにより、保守作業を軽減させるシステムを構築した。今後は、適用実験を行い、有効性を確認する予定である。

参考文献

- [1] D. Cubranic and G. C. Murphy: Hipikat: recommending pertinent software development artifacts, Proceedings of the 25th International Conference on Software Engineering (ICSE 2003), pp.408-419, 2003.
- [2] 伊藤 寛之: コードクローンを対象としたリファクタリング支援環境の提案と実現, 東京学芸大学大学院教育学研究科総合教育開発専攻情報教育コース修士論文, 2007.
- [3] 三浦 真人, 小林 祐介, 島田 和幸, 高橋 晃一, 清木 進, 樋山 淳雄: 個人とコミュニティの支援を有するソフトウェア開発グループ演習環境の提案, 情報処理学会研究報告グループウェアとネットワークサービス, pp.19-24, 2007.
- [4] 佐々木 亨, 井上 克郎: プログラム変更支援を目的としたコードクローン情報付加手法の提案と実装, 大阪大学基礎工学部情報科学科特別研究報告, 2004.
- [5] Y. Ye and G. Fischer: Information Delivery in Support of Learning Reusable Software Components on Demand," Proceedings of 2002 International Conference on Intelligent User Interfaces (IUI'02), pp.159-166, 2002.
- [6] Eclipse Foundation, <http://www.eclipse.org/>.