

細粒度リポジトリホスティングサービスの開発と展望

上村 恭平^{1,a)} 田中 大樹^{1,b)} 一ノ瀬 智浩^{1,c)} 畑 秀明^{1,d)} 飯田 元^{1,e)} 松本 健一^{1,f)}

概要：細粒度なメソッド・関数単位で記録されたソースコードの変更履歴は、より詳細なソフトウェアの分析を可能にする。我々は先行研究において、MSR 研究及びソフトウェア開発者の支援を目的に、細粒度なソフトウェアリポジトリである Historage のホスティングサービスである Kataribe を提案、開発している。本論では、Kataribe を開発運用するなかで明らかになった課題を解決するため、新たな細粒度なりポジトリホスティングシステムを提案し、今後の展望について議論する。

1. はじめに

マイニングソフトウェアリポジトリ (MSR) はソフトウェア工学における研究トピックの一つである。ソースコード分析によるバグ予測や開発予測、バグ修正パッチの自動修正などが行われている [1], [2], [3]。ソースコードを分析する際、メソッドや関数単位の細粒度な情報を用いることで、より詳細な分析を行うことができるが、一般的なソフトウェアリポジトリからこれらの情報を収集することは容易ではない。そのため、我々は先行研究において細粒度なりポジトリホスティングシステムである Kataribe を開発した [4]。Kataribe は畑らの提案している細粒度な git リポジトリである Historage と、標準の git リポジトリを提供するサービス^{*2}である。細粒度かつ大規模な MSR 研究の支援を目的に、Kataribe では約 600 個のリポジトリを公開している。

本論では、現在運用中の Kataribe における課題を整理し、課題を解決するための新しい細粒度リポジトリホスティングシステムの提案を行う。加えて、細粒度リポジトリホスティングシステムの今後の展望について述べる。

2. 先行研究

本章では先行研究である Historage と Kataribe について紹介する。加えて Kataribe の課題について整理する。

¹ 奈良先端科学技術大学院大学
Nara Institute of Science and Technology, Ikoma, Nara 630-0192, Japan

a) uemura.kyohei.ub9@is.naist.jp

b) tanaka.daiki.sx4@is.naist.jp

c) ichinose.tomohiro.ik1@is.naist.jp

d) hata@is.naist.jp

e) iida@itc.naist.jp

f) matumoto@is.naist.jp

^{*2} <http://kataribe.naist.jp>

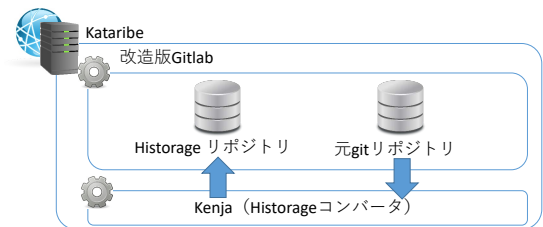


図 1 Kataribe の構成

2.1 Historage とは

Historage は通常の git リポジトリからファイルとディレクトリを変換した git リポジトリである。名前空間やクラスの親子関係をディレクトリツリーで表し、クラスに含まれる関数やメソッドがファイルとして対応するディレクトリに保存されている。このファイルのファイル名はメソッド関数の名前がつけられ、中身には関数の実体が記述されている。そのため、git の変更履歴を参照すると、どのクラス中の、どのメソッドの何行目がどのように変更された、という細粒度な変更情報を容易に得ることができる。Historage の変換には藤原らの開発した変換ツールである Kenja[5] が用いられており、現在は Java, C#, Python, Ruby に対応している。

2.2 Kataribe の構成と課題

現在運用されている Kataribe の構成を図 1 に示す。Gitlab^{*1} はオープンソースの git ホスティングシステムであり、通常 1 つのプロジェクトに対し 1 つのリポジトリを保存する。Kataribe は、Gitlab を改造することで変換前の git リポジトリと Historage の 2 つのリポジトリを提供する機能を実現している。

Kataribe は MSR 研究のために細粒度なりポジトリを提

^{*1} <http://gitlab.com>

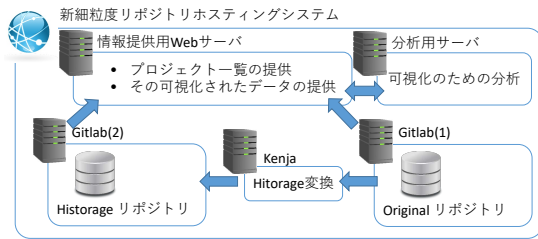


図 2 新細粒度リポジトリホスティングシステムの構成

供するだけでなく、開発者の支援のための細粒度な分析に基づく可視化情報を提供することを目的としている。可視化した情報を提供するためには、Gitlab を改造する必要がある。この際、Gitlab のコード上の変更の必要な箇所や、挙動の変化の影響範囲を調査する必要がある。そのため、可視化のための機能の実装が完了しても迅速に公開できないという問題がある。また、Gitlab は現在も開発中であり、脆弱性対応のためのアップデートなども行われている。しかし、Gitlab の変更、我々の変更と差が大きく、マージが困難になっている。これは Kataribe の保守、機能拡張を行う上での障害となる。

先行研究における細粒度リポジトリホスティングシステムの課題を下記にまとめる。

課題 1: 機能追加時に Gitlab の変更が必要であり、迅速な開発ができない。

課題 2: Gitlab の機能追加や脆弱性対応といったアップデートへの対応が困難である。

これらの課題を解決するため、我々は新規に保守、機能拡張が容易な細粒度リポジトリホスティングシステムを提案、開発する。

3. 提案システム

提案するシステムに求められる要件を整理する。

要件 1: 変換前の git リポジトリと Hitorage の 2 種類のリポジトリを提供する。

要件 2: Hitorage を用いた可視化サービスの基盤となる。

要件 3: 機能の追加・更新に際して迅速に行える。

要件 4: Gitlab のアップデートへの追従が容易である。

上記の要件を満たすため、新規に開発するシステムの構成を図 2 に示す。変換元の git リポジトリと Hitorage の 2 つを管理するため、Gitlab サーバを 2 台内包する。新システムの中核となるのは Gitlab と独立した情報提供用 Web サーバである。情報提供用 Web サーバがそれぞれの Gitlab から情報を収集し、利用者にリポジトリおよび可視化情報を提供する。Gitlab と情報提供用 Web サーバ間の連携は Gitlab の標準 API 及びファイルのやり取りで実現可能であるため、Gitlab を改造する必要はない。また、機能拡張も、情報提供用サーバを変更するだけで実現できる。

4. 細粒度リポジトリホスティングシステムの展望

我々は現在 git リポジトリおよび Hitorage を用いた可視化システム ROCAT を開発している。ROCAT はソースコードの構造を街に見立て、俯瞰的に分析するためのシステムである。ROCAT は単純にソースコードの構造を可視化するだけでなく、その上にさらに情報を重ねることで有用性が増す。例えば、コメント文によりソースコードの書き換えの必要性が示唆される箇所を重畳して表示することで、改修すべきメソッドを選定することに役立てることができる。提案するシステムはこのような可視化情報もあわせて提供することが可能である。

Hitorage 変換に利用している Kenja は Hitorage を用いた高速かつ高精度なリファクタリング検出ツールでもある。従来の git ホスティングシステムで提供されるソースコードの変更履歴に加え、リファクタリング履歴を表示することで、開発者のリファクタリングへの理解を深めることを支援することができるだろう。

細粒度なリポジトリマイニングは未だ発展途上な研究分野である。細粒度な変更履歴を使うことで、バグ予測や進捗予測の精度の向上が期待される。我々の提供する細粒度リポジトリは、開発者の支援になるだけでなく、MSR 研究における大規模なデータ分析にも有用であると考える。

5. おわりに

本論では、保守及び機能追加を考慮した細粒度リポジトリホスティングシステムの構想について述べた。提案システムはオープンソースの git ホスティングシステムである Gitlab を改造することなく、内包することで保守性及び機能拡張性を確保している。今後はシステムの開発・運用を行い、更なる課題の洗い出しを行う。並行して細粒度な可視化について、システム上で提供することを目指す。

参考文献

- [1] F. Zhang, A. Mockus, I. Keivanloo, and Y. Zou. Towards building a universal defect prediction model. MSR 2014, pages 182–191, 2014.
- [2] J. Davies, D. M. German, M. W. Godfrey, and A. Hindle. Software bertillonage. *Empirical Softw. Eng.*, 18(6):1195–1237, Dec. 2013.
- [3] F. S. Ocariza Jr., K. Pattabiraman, and A. Mesbah. Ve-jovis: Suggesting fixes for javascript faults. ICSE 2014, pages 837–847, 2014.
- [4] K. Uemura, Y. Saito, S. Fujiwara, D. Tanaka, K. Fujiwara, H. Hata, H. Iida and K. Matsumoto. A Hosting Service of Multi-Language Hitorage. ICIS 2016, pages 843–848, 2016.
- [5] K. Fujiwara, N. Yoshida, H. Iida. An Approach for Fine-grained Detection of Refactoring Instances using Repository with Syntactic Information(in Japanese). IPSJ Journal, volume 56, number 12, pages 2346–2357 December 2015.