

# リポジトリマイニングに適した プログラムコメントの版管理手法の提案

藤原 賢二<sup>1,a)</sup>

**概要:** 本稿では、プログラムコメントに着目したリポジトリマイニングを支援することを目的に、プログラムコメントの改版履歴を追跡するための版管理システムを提案する。

## 1. はじめに

近年、Git や Subversion に代表される版管理システムが普及し、オープンソースソフトウェアなどが積極的にこれらを採用している。そのため、研究者らはソフトウェアの開発履歴を容易に収集できるようになり、開発者らを支援するために開発履歴を対象にデータマイニングを行うリポジトリマイニングと呼ばれる研究が盛んに行われている。版管理システムを対象にしたリポジトリマイニングは、従来は開発者情報や変更行数などの、分析が容易な情報を対象に行われてきていたが、近年の研究ではソースコードの構造情報の変更などを追跡するような研究が増加してきている。また、ソースコードの構造情報ではなく、ソースコードに記載されたコメントがソフトウェアの品質に影響を与えると考え、その影響を調査している研究がいくつか報告されている [1-3]。しかしながら、これらの研究を支援するためのツールや手法は確立されておらず、研究者らは研究の目的に合わせて都度コメントの抽出やコメントの変更履歴を追跡するためのプログラムを作成しているのが現状である。

そこで本研究では、プログラムコメントに着目したリポジトリマイニングを支援することを目的に、プログラムコメントの改版履歴を追跡するためのリポジトリを提案する。我々は、これまでソースコードの構造情報に着目したリポジトリマイニングを支援するための版管理システム Historage を提案し、そのホスティングサービス Kataribe を開発してきた [4, 5]。本研究では、Historage を拡張することで、プログラムコメントの変更履歴を容易に追跡可能な版管理システムを提案する。

## 2. Historage

Hata らはメソッド、コンストラクタ、フィールドの単位でソースコードの変更を管理するための Historage を提案している [4]。Historage は各コミットにおけるソースコードを構文解析し、得られた構文木を Git リポジトリ上のディレクトリ階層に対応付けて記録する。具体的には、クラスがメソッドを保持することをディレクトリ上の親子関係で表現し、メソッドの本体をファイルとして表現する。このように構文情報を版管理システムに記録することで、通常ファイル変更を追跡するのと同様の操作でメソッド本体に対する変更を追跡できる。

## 3. 提案手法

本研究では、Historage を拡張する形でプログラムコメントの変更履歴を追跡可能な版管理システムを提案する。通常、プログラムコメントはソースコード中の対応するメソッドやフィールド、変数といった、その説明の対象となる要素を持っている。また、コメントが記載されている箇所によっても説明の対象が変わる。リポジトリマイニングを行うにあたっては、コメントが説明の対象とする要素と、説明の目的を知ることが重要であると考え。そこで、ソースコード中のコメントを、メソッドやフィールド、クラスなどのソースコードの構造情報（ディレクトリ階層）に対応する形でファイルとして格納し、ファイル名からコメントの目的を知ることができるように Historage を構築する。

提案手法の概要を図 1 に示す。提案手法では、プログラムコメントを Steidl らによる分類 [1] に従って分類した後、説明対象となる要素の直下にファイルとして格納する形で Historage を構築する。Steidl らは、プログラムコメントをその目的に基づいて表 1 に示す 7 種類に分類し、機

<sup>1</sup> 豊田工業高等専門学校  
NITTC, Toyota, Aichi, 471-8525, Japan  
<sup>a)</sup> fujiwara@toyota-ct.ac.jp

表 1 Steidl らによるコメントの分類

分類名	概要	説明の対象となる要素
コピーライト	著作権表記やライセンスについて説明しているコメント	ファイル
ヘッダ	クラスの説明や、リビジョン番号、作成者などの情報が記載されたコメント	ファイル
メンバ	メソッドやフィールドの役割について説明しているコメント	メソッド、フィールド
インライン	メソッドの実装方法、概要などが記載されたコメント	メソッド
セクション	前後でソースコードを区切り、その意図を説明しているコメント	クラス、メソッド、フィールド
コード	コメントアウトされたソースコード	クラス、メソッド、フィールド
タスク	実装予定の機能や、対応予定のバグに関する情報が記載されたコメント	ファイル、クラス、メソッド、フィールド

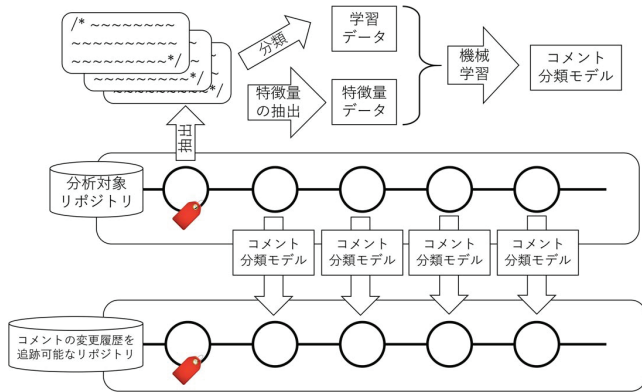


図 1 提案手法の概要

機械学習により半自動的に分類する手法を提案している。

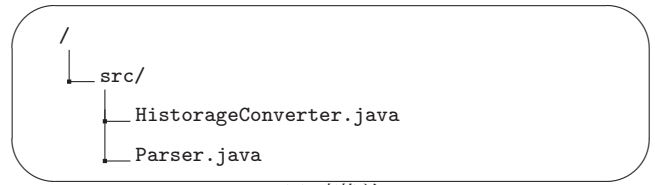
提案手法は、分析対象リポジトリを変換するにあたり、まずコメントの分類を行うためのモデルを機械学習により作成する。モデルの作成には人手による正解集合の作成が必要なため、タグバージョンなどの基準となるコミットを対象に作成する。そして、対応するバージョンのモデルを用いて各コミットのコメントを分類し、Historage中の対応する階層に格納する。提案手法により作成される Historage のディレクトリ構造を図 2 に示す。このようにディレクトリ構造を作成することで、例えば図中の HstorageConverter.convert(Parser) メソッドのメンバコメントの変更履歴を追跡したい場合は、対応するファイル (member-comment) の変更履歴するだけで行えるようになる。また、特定のメソッドにコメントが追加されたコミットの特特定なども容易に行えるようになると思われる。

#### 4. おわりに

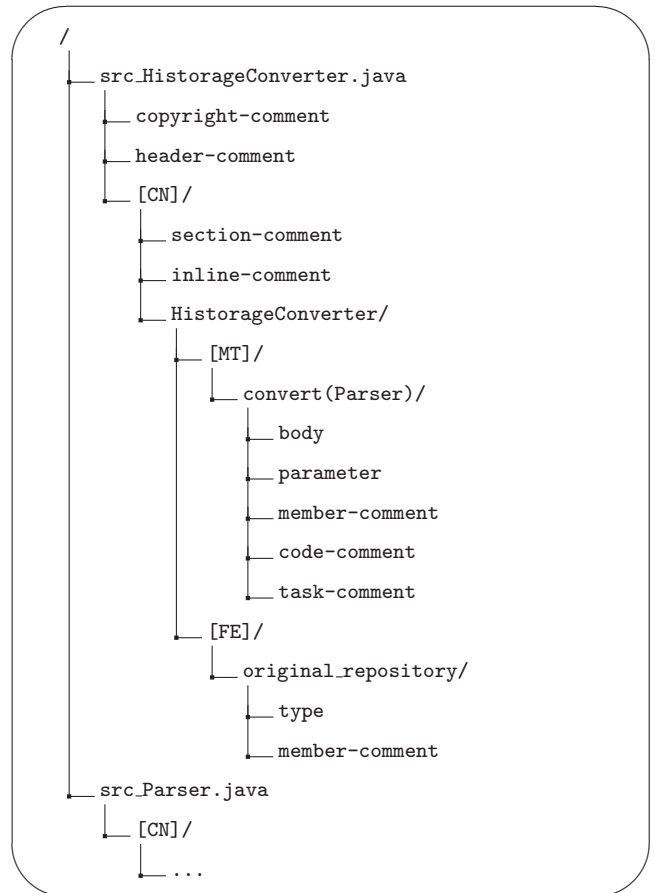
本稿では、Historage を拡張することでプログラムコメントの変更履歴を容易に追跡できる版管理システムの構築手法を提案した。今後は提案手法を実装し、プログラムコメントに着目した既存研究に適用することで手法の有効性を確認する予定である。

#### 参考文献

[1] Steidl, D., Hummel, B. and Juergens, E.: Quality Analysis of Source Code Comments, *In Proc. of ICPC 2013*, pp. 83-92 (2013).  
 [2] 阿萬裕久, 天寄聡介, 佐々木隆志, 川原 稔: 変数名とスコープの長さ及びコメントに着目したフォールト潜在性に



(a) 変換前



(b) 変換後

図 2 提案手法によるディレクトリ構造の変換

関する定量的調査, ソフトウェアエンジニアリングシンポジウム 2015 論文集, pp. 69-76 (2015).  
 [3] Wehaibi, S., Shihab, E. and Guerrouj, L.: Examining the Impact of Self-admitted Technical Debt on Software Quality, *In Proc. of SANER 2016*, pp. 179-188 (2016).  
 [4] Hata, H., Mizuno, O. and Kikuno, T.: Historage: Fine-grained Version Control System for Java, *In Proc. of IWPSE-EVOL 2011*, pp. 96-100 (2011).  
 [5] Fujiwara, K., Hata, H., Makihara, E., Fujihara, Y., Nakayama, N., Iida, H. and Matsumoto, K.: Kataribe: A Hosting Service of Historage Repositories, *In Proc. of MSR 2014*, pp. 380-383 (2014).