変化点検出とトピック分析を用いたリポジトリマイニング手法の提案

久木田　雄亮[†1]　　柏　祐太郎[†1]　　大 平　　雅 雄[†1]

本研究の目的は，大規模ソフトウェア開発における開発状況の変化を早期に発見する手法を構築することである．本研究では，変化点検出アルゴリズムに基づいてリポジトリデータを解析した結果に加え，トピック分析を併用する．開発プロジェクト内で生じる様々な変化を分析者（管理者）がいち早く察知し，開発コンテキストを把握しながら変化の原因を理解するのを助けることが主たる目的である．本ワークショップでは，今後の研究の方向性と適用対象について議論する．

Yusuke Kukita,[†1] Yutaro Kashiwa[†1] and Masao Ohira[†1]

1. はじめに

大規模ソフトウェア開発において高品質なソフトウェアを安定して開発するためには，開発状況を随時把握しておく必要がある．開発状況をリアルタイムに計測するツール（ [1] など）は多数存在しているが，プロジェクトで発生する異変をいち早く検知するための支援は，いまだ十分とは言い難い．

例えば，ソースコードの規模推移を LOC を計測してリアルタイムにモニタリングする状況を考える．実装開始直後は通常，LOC の変化（増加量）は日々比較的大きな値をとり，出荷に向けて LOC の変化は小さな値をとる．テスト工程での欠陥修正による LOC の変化は，実装工程での LOC の変化に比べ小さなものであることが多いため，テスト工程でのコード修正にまつわるなんらかの異常を LOC の変化から見て取ることは困難であると考えられる．すなわち，ソースコードの規模推移がすべて見渡せる環境があるが故に，分析者（プロジェクト管理者）は変化の大きさを相対的に認知してしまい，工程や期間ごとに意味の異なる変化を見落としてしまう可能性が高い．

そこで本研究では，プロジェクトに発生する異変を分析者がいち早く検知するのを支援するために，変化点検出アルゴリズム [2] を用いてリアルタイムに計測するメトリクスから変化点を検出する．また，検出された変化点のコンテキストを把握するのを助けるために，トピックモデルに基づくトピック分析を併用し，分析者が変化が生じた原因を理解するのを支援する．

†1 和歌山大学大学
Wakayama University

2. 変化点検出とトピックモデル

変化点検出とは，データマイニングの分野で用いられている異常検知手法のひとつである．変化点とは，時系列データの振る舞いの急激な変わり目を指す．本研究では，ChangeFinder [2] を用いて変化点検出を行う．ChangeFinder は，AR モデルのオンライン忘却型学習アルゴリズム SDAR を用いた時系列モデルの 2 段階学習に基づいている．

トピックモデルとは，文章から潜在的なトピックを統計的モデルによって推定するためのモデルである．本研究では，トピックモデルの 1 つの手法として Latent Dirichlet Allocation (LDA) [3] を用いてトピックを抽出し，プロジェクト内でどのような議論が行われているかを提示することで，分析者が変化点の前後に何が起きていたかを理解するのを支援する．

3. ケーススタディ

現在構築中の手法を試用し，手法の改善点や適用対象を選定するために Eclipse Platform プロジェクトを対象に行ったケーススタディについて述べる．変化点検出の対象は，修正待ち不具合数，コード行数 (LOC)，サイクロマティック数の平均とした．また，トピック分析の対象は，ニュースフォーラムと不具合管理システムでの開発者らの議論とした．版管理システム Git，ニュースフォーラム Eclipse News Forums，不具合管理システム Bugzilla からデータソースを抽出した（期間は 2002 年 1 月から 2012 年 12 月まで）．

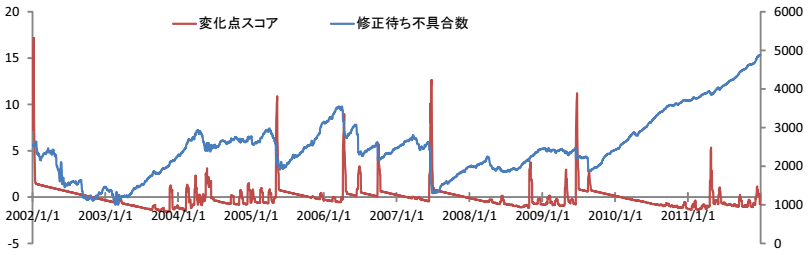

図 1 修正待ち不具合数での ChangeFinder の結果

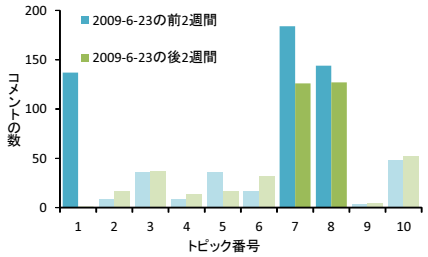

図 2 2009 年 6 月 23 日の前後 2 週間の Bugzilla コメントに対してトピック分析を行った結果

表 1 2009 年 6 月 23 日の前後 2 週間の上位 3 つのトピック

トピック	頻出上位の単語
1	bug update platform equinox addressed mass
7	bug problem duplicate comment marked view
8	created details attachment patch workspace fix

提案手法を用いた分析手順を以下にまとめる．

(1) 対象となるデータソースからメトリクスを抽出する（リビジョン毎など）
(2) 抽出したメトリクスを 1 日毎に集計し時系列データとしてまとめる
(3) メトリクスの時系列データに対して，ChangeFinder を適用し変化点を検出する
(4) 検出された変化点の日時の前後 2 週間の議論に対して LDA を適用する
(5) 変化点の前後のトピックの移り変わりを見ることによってプロジェクトに起きている変化を推測する

4. 結果と考察

図 1 は，修正待ち不具合数に対して ChangeFinder を適用し，変化点スコアを算出した結果である．左軸が変化点スコア（赤色）の数値で，右軸が修正待ち不具合数（青色）である．紙面の都合により，図 2，表 1 にそれぞれ，変化点スコアの大きい 2009 年 6 月 23 日の前後 2 週間のみを対象にしたトピック分析の結果を示す．

図 1 全体を俯瞰した限りでは，2009 年 6 月 23 日の前後に修正待ち不具合数が急激に変化しているようには見えないが，変化点スコアは非常に大きな値を取っている．図 2，表 1 から，2009 年 6 月 23 日の 2 週間前にはトピック番号 1, 7, 8 に関してのコメント数が多いことが分かる．不具合修正に関する議論が行われている様子が見て取れる．一方，これらのトピックに関するコメントは，2009 年 6 月 23 日の 2 週間後には減少している．特に，トピック番号 1 に関してはほぼ言及されていないことが分かる．これらの結果から，Eclipse プロジェクトでは，通常 6 月にプロダクトをリリースするため，2009 年 6 月 23 日を境に不具合修正作業を終了したものと推察できる．

5. おわりに

今後，検出された変化点はどのような原因で発生したのか特定するためのさらなる分析を行う．ワークショップでは，プロジェクトにおける変化とはどのようなものが挙げられるのか，変化点検出によって検出された変化点に対する分析方法はどのようなものがあるのかを議論したい．

謝辞 本研究の一部は，文部科学省科学研究補助金（基盤 (C): 24500041）による助成を受けた．また，独立行政法人情報処理推進機構が実施した「2013 年度ソフトウェア工学分野の先導的研究支援事業」の支援を受けた．

参 考 文 献

1) 大平雅雄，横森励士，阪井　誠，岩村　聡，小野英治，新海　平，横川智教：ソフトウェア開発プロジェクトのリアルタイム管理を目的とした支援システム，電子情報通信学会論文誌 D-I，Vol.J88-D-I, No.2, pp.228–239 (2005).
2) 山西健司：データマイニングによる異常検知，共立出版 (2005).
3) Blei, D.M., Ng, A.Y. and Jordan, M.I.: Latent Dirichlet Allocation, *J. Mach. Learn. Res.*, Vol.3, pp.993–1022 (2003).