

OSSの開発状況理解支援のための可視化手法の提案

伊原 彰 紀^{†1} 大平 雅 雄^{†1}
松本 真 佑^{†1} 松本 健 一^{†1}

Visualization Method for Understanding Developmental Status in Open Source Software

AKINORI IHARA,^{†1} MASAO OHIRA,^{†1}
SHINSUKE MATSUMOTO^{†1} and KEN-ICHI MATSUMOTO^{†1}

1. はじめに

オープンソースソフトウェア (OSS) 開発は分散開発を行っており、世界各地に点在する OSS 開発者の多くはボランティアとして活動しているため、開発プロジェクトへの参加・離脱が自由であることから開発者間の不具合修正に関する情報共有や意思疎通を行うことが難しい。そのため、各プロジェクトでは過去の活動履歴や活動の進捗情報を様々なツールを利用して記録し、不具合の正確な修正に取り組んでいる。例えば、多くのプロジェクトではバージョン管理システムを用いてソースコードを管理している。また、不具合の偏在傾向や修正状況を把握するために、発見された不具合の情報を不具合管理システムで一元管理している。その他にも開発者が開発に関する議論を行うためにメーリングリスト (ML) や掲示板などを利用し、活動の記録として残している。

しかしながら、プロジェクトの運用が長期化し、ソフトウェアが大規模になると、記録される情報は増大し、プロジェクト全体を把握することが困難である。例えば、不具合が混入するソースコードの編集を行う際、他のソースコードに影響を及ぼしていないかを確認する必要がある。ただ、ソフトウェアが大規模になり複雑になると、共通のクラスや関数など依存関係のあるソースコードを把握することが困難になる。また、OSS 開発は企業のソフトウェア開発と異なりソースコードに関与する担当者が決定していないため、様々な開発者が編集を行うことができる。ソースコードの編集を円滑に行うためには、ソースコードの修正を行った開発者を把握する必要がある。

従来研究では円滑な大規模ソフトウェアの実現に向けて、依存関係のあるソースコードの特定や類似する専門性を持つ開発者を見つけ出すために、ソースコード間の関係や開発者間の関係のような同一属性間関係を可視化する研究³⁾²⁾や、ソースコードと開発者の関係のような異なる属性間関係を可視化する研究が行われている¹⁾。不具合を早急に修正するためには、不具合が混入するソースコードやそのソースコードに関係の深い人物を早急に特定し対応する必要がある。

本研究では不具合修正活動の情報を容易に把握するために、開発者、ソースコード、不具合の関係の可視化手法を提案する。OSS 開発者が開発の様子を俯瞰するために、不具合の情報が記載されているバグ票、ソースコード、不具合修正やソースコードに携わる開発者の関係を同時に可視化する手法を考えている。

2. OSS 開発におけるデータ

各 OSS プロジェクトでは開発者やユーザがソフトウェアの品質向上や情報共有を行うことを目的として、過去の活動履歴や活動の進捗情報を管理している。不具合の修正はソースコードのクラス関係などを考えて修正する必要がある。各ソースコードに関与する開発者を把握することで、円滑な修正活動が実現できると考える。つまり、不具合修正には開発者、ソースコード、不具合情報の間に存在する関係が重要であると考えられる。本章では、不具合修正を行うために必要な開発者、ソースコード、不具合の情報のそれぞれに存在する関係について説明する。

ソースコードは複数のバージョンの差分を記録し、修正や機能追加を復元、再修正するためにバージョン管理システムで管理されることが多い。また、ソースコード間の関係を可視化する研究が数多く行われている³⁾。異なるソースコードで同一のクラスや関数を持つ場合、ソースコードを編集することで一方の異なるソースコードに影響を及ぼすことを未然に

^{†1} 奈良先端科学技術大学院大学 情報科学研究科

Graduate School of Information Science, Nara Institute of Science and Technology

防ぐことができる。

ML や掲示板は開発者間の情報交換を行う主要な場として位置づけられている。機能ごとに ML が用意されているプロジェクトも存在するため、頻繁にメッセージの交換を行う開発者は専門分野が類似していると考えられる。開発者の協調作業の様子を理解するために開発者が利用する ML の履歴から開発者間の関係を可視化する研究も数多く行われている²⁾。

不具合情報は修正すべき不具合の管理やソフトウェアの評価などを行うためにバグ管理システムなどを用いて管理している。バグ管理システムでは発見された不具合に関する情報をバグ票に記載している。バグ管理システムには Bugzilla や Trac など様々なツールが存在するが、共通の機能として不具合の修正管理、検索、履歴管理などがある。報告される不具合の中には類似した不具合や一方の不具合を修正しないと修正に取り掛かれないような利害関係ある不具合が存在し、効率的に不具合を修正するために、開発者は関係性を持つバグ票を把握しておく必要があると考える。

3. 開発者、ソースコード、バグ票の関係

不具合を円滑に修正するために、不具合が混入しているソースコードや修正することで影響を及ぼす可能性のあるソースコードや修正すべきソースコードについて詳しく知る開発者を把握することは重要である。本研究では OSS 開発状況を容易に理解するために開発者とソースコードとバグ票の関係に着目する。各属性の関係図を図 1 に示す。

開発者、ソースコード、バグ票において異なる属性の関係は図 1 より 3 パターン（開発者とソースコード、開発者とバグ票、ソースコードとバグ票）が挙げられる。開発者とソースコードの関係は誰によって作成/編集されたソースコードなのかを知ることができるため、不具合が発見された時、ソースコードについてよく知る開発者を容易に特定することができる。しかしながら、誰に修正依頼をするべきか判断することは難しい。そのため、開発者とバグ票の関係を可視化することで、バグを含むソースコードの編集を行った経験のある開発者を容易に特定できると考える。

さらに、ソースコードを修正することで新たな不具合を生み出すことがある。ソースコードとバグ票の関係を可視化することで、不具合を修正するために、ソースコードを編集することで、別のソースコードに影響が出ないかを把握することが可能であると考える。

4. まとめ

本稿では OSS の開発状況を容易に把握するために、開発者、ソースコード、バグ票の関

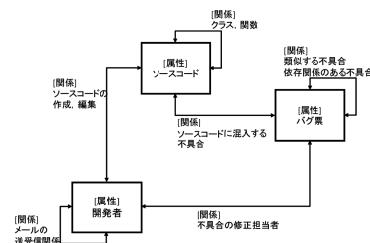


図 1 関係

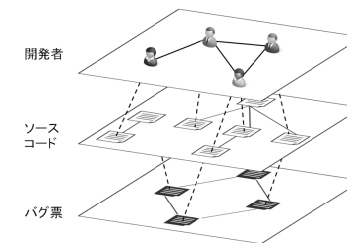


図 2 イメージ図

係図を提案した。開発者、ソースコード、バグ票の関係を可視化することで、プロジェクト運営者にとっては開発、不具合修正の依頼や不具合の偏在傾向の原因を発見するために利用することができると考えている。また、開発者にとっては開発に関する相談相手を見つけるのに利用できると考えている。図 1 の関係図を具体的に可視化したイメージを図 2 に示す。各属性を階層に表わし、3 つの属性全体を俯瞰できるツールの開発を考案中である。

謝辞 本研究の一部は、文部科学省「次世代 IT 基盤構築のための研究開発」の委託に基づいて行われた。また、本研究の一部は、文部科学省科学研究補助費(若手 B: 課題番号 20700028, 21・8995)による助成を受けた。および公益信託マイクロソフト知的財産研究助成基金による助成を受けた。

参考文献

- 1) Froehlich, J. and Dourish, P.: Unifying artifacts and activities in a visual tool for distributed Software Development Teams, *Proceedings of the 26th International Conference on Software Engineering*, pp.387-396 (2004).
- 2) Howison, J., Inoue, K. and Crowston, K.: Social dynamics of free and open source team communications, *In Proceedings of the 2nd International Conference on Open Source Systems (OSS'06)*, pp.319-330 (2006).
- 3) Storey, M.-A. and Muller, H.: Manipulating and documenting software structures using SHriMP views, Vol.0, p.275 (1995).