

## リポジトリマイニングの研究成果の産業界への応用

門田暁人† 伊原彰紀†

本稿では、オープンソースソフトウェア(OSS)を利用しようとする企業が直面している課題に着目し、その解決策について OSS のリポジトリマイニングの研究成果に基づいて議論する。

### Industry Application of Research Findings from Mining Software Repositories

Akito Monden† Akinori Ihara†

This paper focuses on industry concerns in using open source software, and tries to solve them based on results of mining software repositories (MSR) researches.

#### 1. はじめに

今日の多くのソフトウェア関連ビジネスは、オープンソースソフトウェア(Open Source Software; OSS)に依存している。例えば、Android はいまやスマートフォンの主要なOSとなっており、携帯端末の開発企業のみならず、アプリケーション開発企業や個人開発者も盛んに Android 向けアプリを開発している。

OSS を利用することにより、低価格・短納期で顧客にサービスを提供できると期待されるが、その一方で、OSS ならではの障害も多い。独立行政法人情報処理推進機構(IPA)による第 3 回オープンソースソフトウェア活用ビジネス実態調査(有効回答数 700 社以上、2009 年度の調査結果)[1]によると、多くの企業は、テクニカルサポート、品質、ライセンスなど、多岐にわたって OSS 導入時の課題やデメリットを感じている。

本稿では、そのような課題について、リポジトリマイニング(Mining Software Repositories; MSR)の研究成果に基づいた解決策の提案を試みる。多くの OSS プロジェクトでは、構成管理システム(CVS, Subversion, Git など)、バグ管理システム(Bugzilla, Trac, redMine など)などのソフトウェアリポジトリを公開しており、それらのデータを題材として盛んに MSR の研究がなされており、OSS の開発者のみならずユーザーを支援するための成果が得られている。

以降では、企業が OSS を導入する際の課題と、その解決策について議論する。

#### 2. OSS 利用時の課題とその解決策

本章では、IPA の調査[1]に基づき、多くの企業が回答した OSS 導入時の課題について議論する。

##### 2.1. 課題 1: 緊急時の技術的サポートが得にくい

67.3%の企業がこの課題を挙げている [1]。企業において OSS を導入した際に、様々な技術的な問題が発生し得る。しかし、商用ソフトウェアと異なり、OSS にはサポート窓口がないため、自力で解決するか、あるいは、何らかの方法により OSS コミュニティから解決策を得る必要がある。

解決策につながりそうな MSR 研究としては、OSS のオンラインフォーラムから解決策を検索するというアプローチがある[2]。ユーザの多い OSS では、複数のユーザが同じ問題に直面することが少なくないため、オンラインフォーラムにおいて回答済の問題も多い。そのため、与えられた問題に対し、オンラインフォーラムにおける議論の中から既回答の文章を検索することが有効な解決策となる。ただし、キーワードベースの検索では、なかなか回答にたどり着けないことも多い。そこで、Gottipati ら[2]は、自然言語で記述された問題文を入力とするセマンティック検索エンジンフレームワークを提案し、従来のキーワードベースの検索エンジンと比べて大幅に検索精度が改善したことを示している。

##### 2.2. 課題 2: 利用している OSS がいつまで存続するか分からない

58.8%の企業がこの課題を挙げている。一般に、OSS 開発プロジェクトが突然終了する、もしくは、プロダクトのサポートを打ち切ることが起り得る。例えば、かつ

†奈良先端化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

てブラウザ市場で大きなシェアを占めていた Netscape は、シェアを次第に失っていったことが原因で、2008 年 3 月に全てのサポートを打ち切り、プロジェクトを終了した。さらに言えば、多くの OSS は、メジャーリリースを出荷すると、旧バージョンのバージョンアップを打ち切る場合がある。このような打ち切りは、OSS をビジネスに用いている企業の死活問題である。

解決に繋がりそうな MSR 研究の一つは、開発者／ユーザ・コミュニティの活動メトリクス[3]を計測することである。活動が低下してきたら、プロジェクトの終了が近づいている可能性がある。また、別の MSR 研究では、開発者とユーザのコミュニケーション・ネットワークを分析し、コミュニケーションがうまくいっているか否かを評価している[4]。分析事例として、Netscape プロジェクトの終了間際には、ユーザと開発者をつなぐコーディネーターと呼ばれる人物がほとんどいないことが観測されている。一方、活発なプロジェクトの一つである Apache は、多数のコーディネーターがいることが観測されている[4]。

OSS プロジェクトの終了を防ぐことはできないが、開発者／ユーザ・コミュニティの活動状況を把握することで、終了の兆候をつかむことは十分に可能であると考えられる。

### 2.3. 課題 3: バグの改修や顧客からの要請対応に手間がかかる

43.4%の企業がこの課題を挙げている。一般に、OSS のユーザ企業が、OSS に含まれるバグを修正する、あるいは、OSS に機能追加・変更を行うことは容易でない。なぜならば、OSS は自らが開発したものではなく、また、他人が開発したソースコードを変更することは一般に多大な労力を要するためである。そのため、OSS のバグ修正や変更を少しでも容易にする技術が求められている。

MSR 研究からの一つのアプローチは、bug localization と呼ばれる技術を用いることである[5]。この技術は、故障 (failure) の詳細な記述 (例えば、故障を再現させる動作、故障が起こった機能の名称など) を入力として、テキストマイニングにより、その原因となるソースコードを推定する[5]。ただし、現状の推定精度は 30%程度であり、今後の研究の進展が期待される。

もう一つのアプローチは、co-change もしくはロジカルカップリングと呼ばれる技術である[6]。このアプローチでは、ソースコードリポジトリに対する過去のコミットログから、同時に変更されるソースファイル群を特定する。これにより、ソースコードの修正の手助けとなる。

### 2.4. 課題 4: ライセンスが複雑で把握しにくい

34.8%の企業がこの課題を挙げている。OSS には多種多様なライセンスが存在し、それらを正しく理解することは容易でない。OSS ライセンスのリファレンスモデルとして、Open Source Initiative (OSI) は、Open Source Definition (OSD) を定義している。また、OSI は約 70 のライセンスを認定し、それらを分類して整理しており、OSS のユーザがライセンスを理解する一助となり得る。

ライセンスに関する課題のもう一つは、各 OSS プロダクトが、どのライセンスを採用しているのかを把握することがそもそも難しいということである。一般に、一つの OSS プロダクトが複数のライセンスを含むことは少なくない。一つの解決策として、テキストマイニングによる自動ライセンス特定システムが提案されている[7]。

## 3. おわりに

本稿では、企業が OSS を導入する際の課題と、その解決策について議論した。本稿で紹介した以外にも様々な課題が提示されており[1]、これらについて解決策を検討していくことが今後の課題となる。

## 参考文献

- [1] 独立行政法人情報処理推進機構, 第 3 回オープンソースソフトウェア活用ビジネス実態調査, 2010.
- [2] S. Gottipati, D. Lo, J. Jiang. Finding relevant answers in software forums. Proc. ASE'11, pp. 323-332, 2011.
- [3] H. Zhong, Y. Yang, and J. Keung. Assessing the representativeness of open source projects in empirical software engineering studies. Proc. APSEC'12, pp. 808-817, 2012.
- [4] S. Matsumoto, Y. Kamei, M. Ohira, and K. Matsumoto. A comparison study on the coordination between developers and users in FOSS communities. Proc. STC'08, pp.1-9, 2008.
- [5] J. Zhou, H. Zhang, and D. Lo. Where should the bugs be fixed?. Proc. ICSE'12, pp.14-24, 2012.
- [6] D. Beyer. Co-change visualization applied to PostgreSQL and ArgoUML. Proc. MSR'06, pp.165-166, 2006.
- [7] D. M. German, Y. Manabe, and K. Inoue. A Sentence-Matching Method for Automatic License Identification of Source Code Files. Proc. ASE'10, pp. 437-446, 2010.