人工知能学会
第 16 回知識流通ネットワーク研究会
SIG-KSN-016-05

OSSコミュニティサイトを利用した分析手法に関する考察

○ 桑田 喜隆(*1)
(*1) 室蘭工業大学

A study on analysis methods of OSS community sites

Yoshitaka Kuwata(*1)
(*1) Muroran Institute of Technology, Japan

概要

オープンソースソフトウェア(OSS)は商用ソフトウェアと共に，商用システムでも広く使われるようになっている．しかしながら，ソフトウェアごとに状況が異なるため，採用するにあたっては，その継続性，サポート，品質，およびライセンス条件などを考慮することが重要である．筆者らは OSS コミュニティの情報をもとに，採用判断に必要な情報を取り出し，分析をする手法を開発している．本稿では，OSS コミュニティサイトを利用した分析手法について考察する．

Abstract

Recent years, Open Source Software (OSS) is widely used for commercial systems, as well as commercial software. In order to choose one of OSS required for ones' purpose, it is very important to consider various conditions such as continuity, support, quality, and license. We proposed analysis method of OSS based on OSS community model. In this paper, we examined analysis method based on OSS community sites

1. はじめに

ここ数年，ソフトウェア工学の分野を中心にオープンソースソフトウェア（OSS）コミュニティで開発されるソフトウェアやその開発手法，コミュニティの分析が盛んに行われるようになってきている．

従来から OSS の公開レポジトリからソフトウェアのソースコードやその改変履歴等が取得可能であった．またソフトウェア開発に関する議論を行うメーリングリストなどもアーカイブがあり，ソフトウェア開発のアクティビティ等を分析することが可能である．近年，OSS の開発プロセスを分析しやすくするための環境も整備させてきている．例えば，2011 年に公開された GitHub Archive[8]を使うと，GitHub[7]上で発生した開発イベントの履歴を参照して分析することが可能である．これらを活用することで，ソフトウェア生産物およびその生産プロセスの定量的な分析を行うことが可能である．

他方，上記とは別に，OSS コミュニティを人のつながりと捉え，コミュニティの動向からソフトウェア開発手法を分析するアプローチがある．例えば，OSS のコミュニティサイトである advogato.org[11]や Ohloh[2]（2014 年に OpenHub と改名）の分析を行うことで，OSS コミュニティの状態や動向を把握する研究が行われている．

筆者らは，OSS を利用する立場から OSS の選定にあたっての将来性やリスクなどを知ることを目的に検討を行ってきている．これまで，コミュニティとプロジェクト，成果物のモデルを作成し，コミュニティの成熟度評価することを提案してきた．

本稿では，従来のアプローチのメリットデメリットを分析した上で，コミュニティを分析する新たな方法を提案する．

2. OSS コミュニティサイトの分類

本稿では，OSS 関連の情報源の総称を OSS コミュニティサイトと呼ぶこととする．

Storey ら[13]の GitHub 利用者への調査によると，OSS の開発者は複数の情報源（チャネル）を使っている．以下にチャネルを

1 Yoshitaka Kuwata
室蘭工業大学
北海道室蘭市水元町２７−１
kuwata@mmm.muroran-it.ac.jp

あげる．

- ソースコードレポジトリ(例：GitHub, SourceForge)
- 対面コミュニケーション(F2F)
- Q&A サイト(例：Stack Overflow)
- Web サーチ
- マイクロブログ（例：Twitter）
- Private Chat（例：IRC, Skype）
- Blog や RSS のからの最新情報

図 1 に各チャンネルをあげた開発者数を示す．

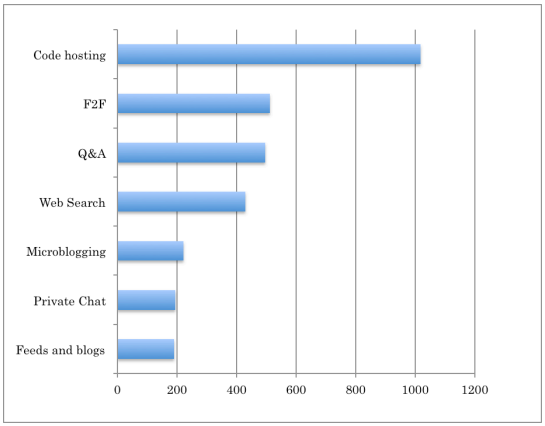

図 1 GitHub 開発者 1516 人の利用しているチャンネルとその度数(Storey より引用)

この調査によると，OSS の開発者は複数のチャネルを同時に利用して活動を実施していることがわかる．ただし，上記調査は GitHub を利用した開発者に電子メールでアンケートをとる方法を取っているため，ソースコードレポジトリの利用が実際より多くカウントされている可能性がある点に注意したい．

一方で，OSS のコミュニティに関する情報を一元的に集めた OSS レポジトリが提供されている．OSS レポジトリの例として Advogato[11]，RepOSS[12]，OpenHub[3]があげられる．これらのサイトは，OSS 組織のプロジェクトの情報を統括的に提供している点が特徴である．複数のプロジェクトを統括している組織が提供するポータルとは区別して扱うこととする．例えば，Linux Foundation は配下に多くのプロジェクトを統括しており，Linux Foundation 配下のプロジェクトのポータルサイトを提供している．これに対して，OpenHub は特定の OSS 組織の情報ではなく，組織を超えて統合的に OSS の情報の提供を行っている．

以上をまとめると，OSS コミュニティサイトには以下が含まれる．

- ソースコードレポジトリ
- OSS 組織のポータルサイト(Wiki, ML などを含む)
- OSS レポジトリ
- Q&A サイト
- OSS に関する個人の Blog

3. これまでの取り組み

近年，IEEE や ACM 主催のソフトウェアレポジトリのマイニングに関するカンファレンスが開催されるなど，ソフトウェア工学の分野で OSS の分析が注目されている．

特に，最も広く使われているソースコードレポジトリである GitHub の分析に関しては多くの論文が発表されている．例えば Tsay[14]らは GitHub 上でソースコードの変更提案(pull-request)に関連する議論の流れについてインタビューを通じて分析を行っている．また，Biazzini[1]らはプログラムコードの fork に着目してプロジェクトの状態を可視化する方法を提案している．

GitHub Archive[8]は 2011 年以降の GitHub 上での 20 種類以上のイベントを時系列列データとして提供しており，GitHub で行われたソフトウェア変更の過程を追いかけることが可能である．

他方，Kalliamvakou [9]らは，GitHub の分析に関する限界を指摘している．例えば，ソースコードレポジトリは必ずしもソフトウェア開発やその議論の場として使われておらず，成果物の格納場所として使われている場合がある．また， GitHub 全体の commit 数の中央値は 6 であり，多くのプロジェクトでは開発成果を頻繁に更新していないことを指摘している．このように，ソースコードレポジトリのマイニングだけからプロジェクトの状態を推定するためには，前提条件などを明確にしておくことが必要になる．

ソースコードレポジトリを解析するアプローチに対して，OSS プロジェクトのメタ情報を集めた OSS レポジトリを解析するアプローチも取られている．例えば，Bruntink は OpenHub の分析ツールを提供[5]すると共に，ソフトウェアの最も基本的なメトリックとして，コードサイズとその変化率を提案[6]している．またそれに先立ち OpenHub のデータの不完全性に対して，クレンジングする方法を提案[4]している．

いずれのアプローチの場合にも，前提条件および解析の限界を明確にしておくことが必要であると考える．

4. 提案手法

これまで，筆者らは OSS コミュニティを組織の成熟度モデルに基づき評価する方法を提案[10]した．公開されている組織の運営状況や関係者へのインタビュー等をもとにコミュニティの状態を特定し，成熟度レベルを決める方法を採用している．

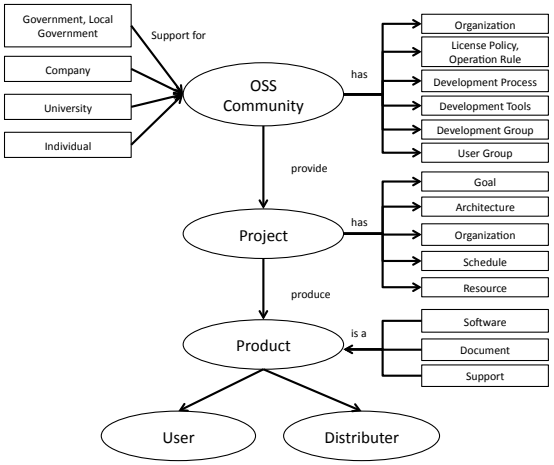

図２ OSS コミュニティの関係モデル

図２に筆者らの提案する OSS コミュニティの関連モデルを示す．本稿では，OSS コミュニティサイトから集めたプロジェクトやプロダクトに関する定量的な情報を活用することで，手動で収集した情報とあわせてコミュニティの状態を分析することで分析の精度を向上させる方法を検討する．

5. 課題

OSS コミュニティの比較分析を行うためには，複数の OSS コミュニティサイトの分析が必須になる．

ソースコードレポジトリの解析に対しては，以下の課題があげられる．

(1) ソースコードレポジトリはプロジェクト単位に管理されているため，OSS コミュニティの単位でプロジェクトを分類し，解析する必要がある．
(2) OSS コミュニティによって採用しているソースコードレポジトリが異なる．複数のソースコードレポジトリを参照することが必要になる．
(3) ソースコードレポジトリの利用方法や取得可能な情報がプロジェクトやソースコードレポジトリの管理方法によって異なるため，その違いを意識した解析が必要である．

これらの課題に対して，OSS リポジトリは複数のソースコードレポジトリからの情報を網羅的に提供しているため課題(2)に関しては解決策となる．一方で，以下の課題が新たに発生する．

(4) ソースコードレポジトリ等から自動的に取得される情報は網羅性が高いと考えられるが，手動で登録される「人」や「OSS コミュニティ」等に関しては，その網羅性について検証が必要である．
(5) 自動取得される情報はソースコードレポジトリのクローリングにより収集される 2 次情報であるため，更新のタイミングが遅れる可能性がある．

また，どちらのレポジトリの解析にも以下の共通の課題が存在する．

(6) 情報の不完全性や誤りが含まれていることを前提にした解析を行うことが必要である．

6. OpenHub の分析

筆者らの論文で分析を実施した OSS コミュニティは以下の 4 個である．

- Linux Foundation
- Free Software Foundation
- Apache Software Foundation
- OpenStack Foundation

この中で，2015 年 3 月時点において OpenHub 上で情報の入手が可能であった Apache Software Foundation (ASF) および OpenStack Foundation に関して分析を行った．

6.1 Apache Software Foundation のケース

(1) プロジェクト数

OpenHub 上では ASF に所属するプロジェクトが 338 個登録されている．ASF のポータルページ[2]ではプロジェクトとして 248 個が登録されており，上記と数が異なっている．これはソフトウェアコードレポジトリ上に製品単位で登録されているものがあるため, OpenHub も実際のプロジェクト数より多くのプロジェクトがあげられている．

OSS 組織の成熟度モデルの基礎情報としてプロジェクト数の情報を OpenHub から取得する場合には精査が必要であることが分かった．

(2) プロジェクトごとのコミッター数

図 3 に OpenHub のデータを基に計算したコミッターの分布を示す．100 人以上のコミッターが関与するプロジェクトは 2 個のみである．また，50 人以上のプロジェクトは 16 個ある．

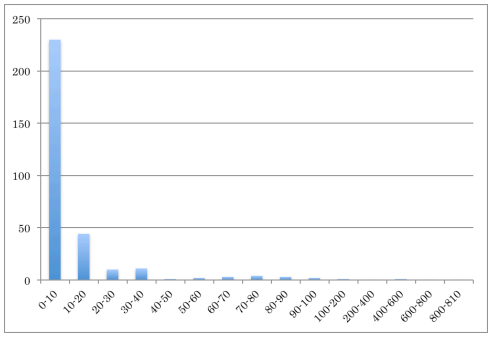

図 3 Apache Foundation 配下の 340 プロジェクトの 12 ヶ月以内に活動のあったコミッター数の分布

10 人以下のプロジェクトが全体の 67%である．少数の大規模プロジェクトと大規模な小プロジェクトが混在していることが分かる．

(3) コミッターの外部比率

OpenHub に登録する際に，提携先としてコミュニティ名を選択することが可能である．プロジェクトにコードを投稿した人のうちで，コミュニティ名を選択したコミッターとそれ以外を選択したコミッターの比率は，プロジェクト外部からの協力者の比率を示していると考えられる．

ASF のケースでは, 30 人以上の中規模から大規模プロジェクトに内部コミッターが多く配置されていることが分かった．

但し，OpenHub の人の所属情報に関して基準が明確でないため，入力されている情報にばらつきが大きいことや，誤りが含まれていることが予想される．

6.2 OpenStack Foundation のケース

(1) プロジェクト数

登録されているプロジェクトは少ないこともあり，OpenStack Foundation の公開しているプロジェクト数と OpenHub 上のプロジェクト数に差がない．

(2) プロジェクトごとのコミッター数

図 3 に OpenHub のデータを基に計算したコミッターの分布を示す．全てが 50 人以上のコミッターが関与するプロジェクトであり，規模が大きなプロジェクトであることが分かる．これは，活発に開発が行われている状況を示しており，筆者らの分析とも一致する．

[2] http://projects.apache.org/indexes/quick.html

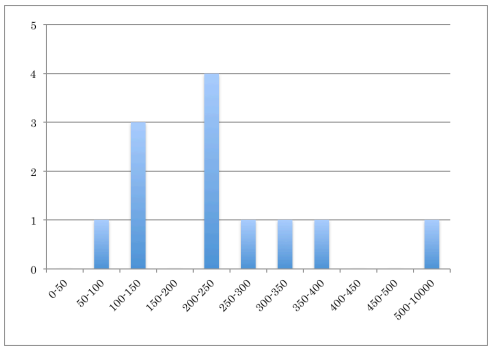

図 4 OpenStack Foundation 配下のプロジェクトの 12 ヶ月以内に活動のあったコミッター数の分布

7. OpenHub の分析から得られた知見

7.1 コミュニティサイトの不完全性

OpehHub は OSS コミュニティサイトであり，情報提供者が自主的に登録する仕組みとなっている．プロジェクトやソフトウェアレポジトリの情報を入力することで，OpenHub が情報の収集を開始する仕組みである．またコミュニティとプロジェクトの関係も手動で定義している．このため，誤りや登録の漏れが生じる．

情報の不完全性の例として，2015 年 3 月現在 OpenHub に Free Software Foundation(FSF)に関する情報がない点があげられる．組織のエントリーは存在するが，内容な未入力である．

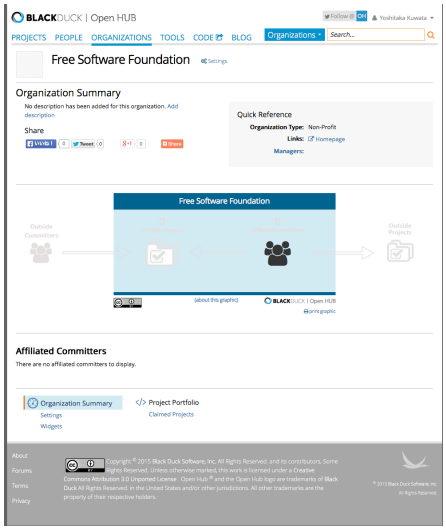

図 5 OpenHub の FSF に関するページ[3]

他方，Gnu Emacs や gcc など FSF 関連の著名なプロジェクト情報は登録されている．また FSF の主催者である Richard Stallman も登録されているが，コミッター ID としては Richard M. Stallman，Richard Stallman の両方があり，統合化されていない状態である．

これは単なる誤りではなく，FSF に関して当事者を含めて積極的に OpenHub サイトに情報を提供する人がいないことが考えられる．OpenHub 自身もまた OSS コミュニティの一部であり，集められた情報の量がコミュニティ同士の関係性に依存する部分があると考えられる．

7.2 情報の欠損およびばらつき

多くのプロジェクトで，データが入力されていないフィールドがある．またプロジェクトとして登録するための基準がないため，誰でもプロジェクトを追加できる．例えば，一人で開発を行う「一人プロジェクト」でも登録することが可能である．GitHub に公開しているエントリーを登録することで，コードに関しても解析が行われる．

一人プロジェクトの登録例として筆者の登録したプロジェクトのページを図 6 に示す．

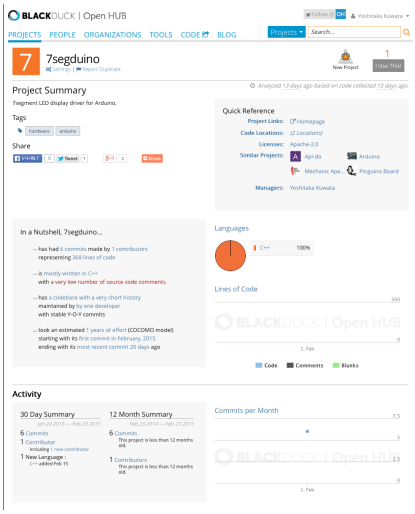

図 6 OpenHub の一人プロジェクトの例[4]

[3] https://www.openhub.net/orgs/311

[4] https://www.openhub.net/p/Sevensegduino

8. まとめと今後の課題

本論文では，OSS コミュニティサイトの情報を分析することで，OSS 組織の成熟度モデルの評価のための情報を取得する方法を検討した．

OSS レポジトリである OpenHub の分析から，コミュニティの推定に利用可能な情報の一部を取得することが可能であることが分かったが，一方で情報の精度や特性など，利用に当たって注意すべき点があることが分かった．

また，OpenHub にはソースコードの解析機能があり，ソフトウェア品質を検証するための情報として利用可能であると考えられる．

既存の OSS レポジトリとソフトウェアコードリポジトリの情報を組み合わせることで，更に効率的に情報を収集し，情報の精度を向上することが今後の課題である．

A. 参考文献

1. Biazzini, M. and Baudry, B. "May the fork be with you": novel metrics to analyze collaboration on GitHub *Proceedings of the 5th International Workshop on Emerging Trends in Software Metrics*, ACM, Hyderabad, India, 2014, 37-43.
2. Black Duck Software Inc. OpenHub, https://openhub.net/, 2015, Mar, 2, 2015 access.
3. Black Duck Software Inc. OpenHub, https://openhub.net/, Mar, 2, 2015 access.
4. Bruntink, M. An Initial Quality Analysis of the Ohloh Software Evolution Data. *Electronic Communications of the EASST*.
5. Bruntink, M. OhlohAnalytics data set and analysis tools, 2013.
6. Bruntink, M. Towards base rates in software analytics Early results and challenges from studying Ohloh. *Science of Computer Programming*, *97*. 135-142.
7. GitHub Inc. GitHub, https://github.com/, Mar, 2, 2015 access.
8. Ilya, G. GitHub Archive, https://githubarchive.org/, Mar, 2, 2015 access.
9. Kalliamvakou, E., Gousios, G., Blincoe, K., Singer, L., German, D.M. and Damian, D. The promises and perils of mining GitHub *Proceedings of the 11th Working Conference on Mining Software Repositories*, ACM, Hyderabad, India, 2014, 92-101.
10. Kuwata, Y., Takeda, K. and Miur, H. A Study on Maturity Model of Open Source Software Community to Estimate the Quality of Products. *Procedia Computer Science*, *35* (0). 1711 - 1717.
11. Rainwater, R.S. Adbogato, https://adbogato.org, Mar, 2, 2015 access.
12. RepOSS Project. RepOSS, http://reposs.org/, Mar, 3, 2015 accesse.
13. Storey, M.-A., Singer, L., Cleary, B., Filho, F.F. and Zagalsky, A. The (R) Evolution of social media in software engineering *Proceedings of the on Future of Software Engineering*, ACM, Hyderabad, India, 2014, 100-116.
14. Tsay, J., Dabbish, L. and Herbsleb, J. Let's talk about it: evaluating contributions through discussion in GitHub *Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering*, ACM, Hong Kong, China, 2014, 144-154.

※ 記載されている会社名，商品名，又はサービス名は，各社の商標又は登録商標です．