

推薦論文

OSSコミュニティにおける 開発者の活動継続性を理解するための Politeness 分析

宮崎 智己^{1,a)} 伊原 彰紀² 大平 雅雄¹ 東 裕之輔¹ 山谷 陽亮¹

受付日 2017年3月6日, 採録日 2017年10月3日

概要: オープンソースソフトウェア (OSS) 開発はオンラインでの非対面コミュニケーションを通じた協調作業を基本とする。OSS コミュニティに参加する開発者が快適に継続的に活動を行うためには、開発者がお互いを配慮するためのコミュニケーション上の工夫 (本研究における Politeness) が必要になると考えられる。本研究では、Politeness を定量化するためのツールを用いて、OSS 開発における膨大な量のコミュニケーションデータから Politeness を数値化し、開発者の Politeness と活動継続性との関係を分析する。Apache HTTP Server および Python プロジェクトを対象とするケーススタディを行った結果、開発者自身の Politeness と活動継続性には一定の関係があることを確認した。本研究で得られた知見は、開発者の離脱を予防・阻止するための方策を立案することに役立てることができる。

キーワード: Politeness, 活動継続性, 開発者間コミュニケーション

An Politeness Analysis for Understanding Continuous Activities of Developers in OSS Communities

TOMOKI MIYAZAKI^{1,a)} AKINORI IHARA² MASAO OHIRA¹
YUNOSUKE HIGASHI¹ YOSUKE YAMATANI¹

Received: March 6, 2017, Accepted: October 3, 2017

Abstract: Developing open source software (OSS) is collaborative in nature through non-face-to-face communication online. Developers joining an OSS community would be necessary to make communicative considerations (Politeness in this paper) to each other, in order to comfortably continue their efforts for long periods of time. In this study, we analyze the relationship between Politeness and developers' continuous activities, using large-scale communication data in OSS development and a tool to measure Politeness quantitatively. As a result of our case study on the Apache HTTP Server and Python projects, we found that there exists a certain relationship between Politeness and developers' continuous activities. We believe that the findings in this study could be useful to plan a step to prevent long-term developers from leaving an OSS project.

Keywords: Politeness, continuous activities, communication between developers

1. はじめに

オープンソースソフトウェア (OSS) は、自由に利用、拡張、再配布できる特徴から、個人ユーザをはじめとす

る多くのボランティア開発者が、新たな機能拡張を提案し、そして、自らが開発に参加している。一方で、多数の開発者が1年未満でOSSプロジェクトを離脱しているため、長期的に参加する開発者の不足がOSSの継続的な開発保守を困難にしている。従来研究では、開発者が継続

¹ 和歌山大学

Wakayama University, Wakayama 640-8510, Japan

² 奈良先端科学技術大学院大学

Nara Institute of Science and Technology, Ikoma, Nara 630-0192, Japan

a) s181052@center.wakayama-u.ac.jp

本論文の内容は2016年7月のマルチメディア、分散、協調とモバイル DICO2016 シンポジウムで報告され、グループウェアとネットワークサービス研究会主査により情報処理学会論文誌ジャーナルへの掲載が推薦された論文である。

的に活動を行う要因を明らかにする研究がされてきている [1], [2], [3], [4], [5]. Zhou らは、長期的にプロジェクトに参加する開発者 (Long Term Contributor) を参加初期に予測するモデルの開発を行い、開発者のプロジェクト参加直後の活動意欲 (コメントや不具合報告など)、および、他の開発者からのサポートが継続的な活動に重要であることを示した [2], [3], [4].

様々な地域に点在する OSS 開発者は、主にメールを用いてコミュニケーションを行っている。メールのような非対面コミュニケーションでは、意思疎通に誤解が生じやすいため、開発者は自身の考えを正確に伝えるために、配慮のあるコミュニケーションを行っていることが考えられる。たとえば、自信のない提案、指摘には曖昧な表現を使用する。一方で、継続的に活動する開発者は曖昧な表現の使用頻度が減少するなどが考えられる。しかしながら、継続的に活動する開発者を対象に、コミュニケーション上の配慮について分析した研究はない。

本論文では、開発者のコミュニケーション上の配慮を分析することで、開発者が継続的に活動することを可能にしている要因を明らかにする。継続して活動を行っている開発者が他の開発者と配慮のあるコミュニケーションを行っていることを明らかにすることで、開発者の早期離脱を阻止するための開発者間での配慮のあるコミュニケーションの方策を示すことができる。

本論文では、開発者のコミュニケーション上の配慮を Politeness [6] を用いて定義し、Apache HTTP Server プロジェクトと Python プロジェクトにおける開発者メーリングリストのコミュニケーションデータを用いたケーススタディを行う。本ケーススタディでは、以下の3つのリサーチクエスチョンに取り組む。

RQ1: 他の開発者の Politeness は開発者の活動継続性に関係するのか？

RQ2: 開発者自身の Politeness は開発者の活動継続性に関係するのか？

RQ3: Politeness の変動が長期開発者の継続性と離脱率に関係するのか？

本研究における「活動継続性」とは、開発者が取り組む活動の継続性を指す。たとえば、オープンソース開発では、発見した不具合を報告する、修正パッチを投稿する、投稿された修正パッチをレビューする (コードレビュー) など様々な活動があり、それらの活動に継続的に取り組む開発者が存在する。本論文では、Politeness を数値化する Stanford Politeness を用いてメーリングリスト上での開発者のコミュニケーションの継続性を計測する手法を提案する。

以降、2章では、開発者の継続性について述べ、本論文の目的を位置付けする。次に、3章で、本論文で用いる Politeness について説明し、4章で、リサーチクエスチョンの動機と分析手法を述べる。5章では、ケーススタディ

の結果を示し、6章で、結果を考察する。最後に7章で本論文をまとめる。

2. OSS 開発者の活動継続性に関する関連研究

OSS 開発では不特定多数の開発者が参加し、高機能、高品質なソフトウェアを実現している。開発者は、時間をかけてプロジェクトに貢献していくことで中心的な役割を担うようになるが、プロジェクトに長期間貢献する開発者はごくわずかである [7]. Bird らは、大規模 OSS (Apache, PostgreSQL, Python) プロジェクトにおける開発者の平均活動期間は約 1.5 年であることを明らかにし、開発者がプロジェクトに参加してから 1.5 年以内に重要な役割を与えることで開発者が長期的に貢献することを示唆している [1]. Steinmacher らは、OSS 開発者を対象にアンケートを行い、新人開発者のプロジェクト参加を妨げる 58 種類の障壁 (技術的障壁/社会的障壁) があることを明らかにしている [8]. 新人開発者の獲得は中心的な開発者の獲得につながるため、プロジェクトはこれらの障壁を取り除く必要がある。

OSS 開発は協調作業であるため、他の開発者との人間関係は継続的に活動を行ううえで重要である。Bird らは、Apache HTTP Server を対象に分析を行い、開発者が活動を行うにつれて他の開発者とコミュニケーションを行う頻度が高くなることを示した [9]. また、Bird らは5つの OSS プロジェクトを対象に分析を行い、プロジェクトが進化するにつれて開発コミュニティ内にサブコミュニティが形成されること、また、同じサブコミュニティに属する開発者間で共通のファイルを変更する傾向があることを示した [10]. Dabbish らは、ソーシャルコーディングサイト Github で活動を行っている開発者に対してアンケートを行い、他の開発者の活動を認識することが自身のモチベーション向上につながることを示した [11]. Qureshi らは、社会的活動を多く行っている開発者はコア開発者になる傾向があることを示した [12].

OSS を長期的に保守するためには、長期的に貢献する開発者を確保することが必要不可欠であり、開発者不足はプロジェクトに対して危険を及ぼす。Apache Open Office プロジェクトでは 2016 年 9 月に開発者不足が原因でソフトウェア保守を終了することを提案している*1。プロジェクトは保守終了を撤回し、現在もソフトウェアがアップデートされているが、開発者不足はプロジェクトの課題となっている。また、特にコア開発者の離脱はプロジェクトの運営維持に深刻な問題を引き起こす。Gimp プロジェクトでは、コア開発者の 1 人がプロジェクトを離脱し、他のコア開発者が代理を務めるまでの 1 年間プロジェクトが停止している [13].

*1 <http://www.mail-archive.com/dev@openoffice.apache.org/msg28186.html>

従来研究では、開発者が継続的に活動を行う要因を明らかにする研究がされてきている。Zhou らは、長期的にプロジェクトに参加する開発者 (Long Term Contributor) を、開発者の参加初期のデータを用いて予測するモデルの開発を行っており、開発者のプロジェクト参加直後の活動意欲 (コメントや不具合報告など)、および、他の開発者からのサポートが継続的な活動に重要であることを示した [2], [3], [4]。Gharehyazie らは、コミュニティに参加してから 3 カ月間に社会的活動を多く行っている開発者は継続して活動を行うことを示した [5]。

様々な地域に点在する OSS 開発者は、主にメールを用いてコミュニケーションを行っている。メールのような非対面コミュニケーションは対面コミュニケーションと比べて誤解が生じやすいため、継続的に活動を行っている開発者はコミュニケーションを行う際に他の開発者と配慮のあるコミュニケーションを行っていることが考えられる。短期間でプロジェクトから離脱した開発者は他の開発者から配慮のないコミュニケーションを行われたか、もしくは他の開発者に対して配慮のないコミュニケーションを行ったために、良好な人間関係を確立できずにプロジェクトから離脱した可能性が考えられる。しかしながら、継続的に活動を行っている開発者のコミュニケーション上の配慮の分析はされておらず、コミュニケーション上の配慮が開発者が継続的に活動を行うことを可能にしている要因であるかは分からない。

3. Politeness

本論文では、Brown らによって定義された Politeness [6] を用いて、長期的に活動する開発者のコミュニケーション上の配慮の分析を行った。

Politeness とは、日本語の「丁寧さ」や「礼儀正しさ」とはニュアンスが異なる概念であり、円滑な人間関係を確立・維持するための相手への配慮を指す。Politeness を適切に表現するためには、他者に認められたい・評価されたいという欲求 (ポジティブフェイス) と、他者から批判されたくない・行動を妨げられたくない欲求 (ネガティブフェイス) のどちらも満たすよう、言葉遣いを工夫する必要がある。たとえば、“このエラーの原因は何?” のような質問は、聞き手に心理的負荷をかけてしまい聞き手のネガティブフェイスを侵害してしまう。一方で、“このエラーの原因を教えてくださいませんか?” のような質問は、回答することでポジティブな効果 (感謝されるなど) があり、たとえ回答できなくても問題ないことを暗黙的に示しているため聞き手への心理的負荷が低くなる。

本論文では、Stanford Politeness [14] *2を用いて、開発者のコミュニケーションデータを対象にコミュニケーション



図 1 Stanford Politeness における Politeness の算出方法
Fig. 1 Calculation of Politeness scores by Stanford Politeness.

ン上の配慮を数値化する。Stanford Politeness とは、言語表現に込められた Politeness を定量化するツールである。Stanford Politeness は、図 1 に示すとおり、2つの処理から構成される。

Politeness Strategy の検出。 文章中の言語表現に込められた Politeness を定量化するために、相手を配慮するための具体的な表現を分類した Politeness Strategy [6] を用いる。Politeness Strategy は、相手のポジティブフェイスとネガティブフェイスを満たす語彙を分類しており、Politeness の値が高くなる語彙群には、ポジティブフェイスを満たす語彙 (Gratitude, Deference など) とネガティブフェイスを満たす語彙 (Please, Hedges など) が含まれる。また、Politeness の値が低くなる語彙群には、ネガティブフェイスを侵害する語彙 (Please start, Direct question など) が含まれる。Politeness Strategy では、このような Politeness に影響する語彙群を 18 種類に分類している。本論文では、入力テキストを Stanford Parser *3により構文解析し、Politeness Strategy に登録される語彙を検出する。

SVM による Politeness 値の予測。 Stanford Politeness は、機械学習アルゴリズムの 1 つであるサポートベクタマシン (SVM) [15] を用いて、入力テキストの Politeness 値を算出する。SVM 構築には、Stanford Politeness Corpus [14] を学習データとして用いている。Stanford Politeness Corpus は、Wikipedia ユーザのトークページ*4から抽出した 4,353 の質問文と Stack Exchange *5から抽出した 6,604 の質問文に対して人手で Politeness 値を付与したデータセットである。Stanford Politeness Corpus を学習データとし、検出した語彙を用いて入力テキストの Politeness 値を 0~1 の値 (値が大きいほど相手を配慮する表現が取り入れられている) で予測する。

表 1 に、OSS 開発者のメーリングリストに投稿されたメールに対して Stanford Politeness を適用した例を示す。たとえば、1 番目の文章の Politeness の値は 0.997 と高い。これは文中に含まれる “thanks” が Politeness Strategy の Gratitude に、“suggestion” が Politeness Strategy の Hedges に該当しているためである。2 番目の文章は、単語数が多く情報量の多い文章ではあるが、Politeness Strategy を含んでおらず、0.507 と中間の値が算出されている。3 番目の文章は、“So, what...” の部分が Politeness Strategy の

*3 <http://nlp.stanford.edu/software/lex-parser.shtml>

*4 http://en.wikipedia.org/wiki/Wikipedia:User_pages

*5 <http://stackexchange.com/about>

*2 <https://github.com/sudhof/politeness>

表 1 Politeness 分析 [14] を適用した結果の例
Table 1 Examples of result of applying Politeness analysis [14].

No.	例文	Politeness 値
1	OK, thanks for your suggestion. I will change that. Regards XXX (XXX は, ファーストネーム).	0.997
2	Oh dear. My question is already slipping down the list into obscurity... I guess I must be the only person using eclipse 3.4 who would like to be able to do this.	0.507
3	So, what was the solution for this problem then??	0.122

Direct start に該当しており, 相手を不愉快にさせる可能性の高い表現であるため, 0.122 と低い値となっている。

4. リサーチクエスチョン

本論文は, 開発者の Politeness が開発者が継続的に活動を行うことを可能にしている要因であるかを明らかにするために 3 つのリサーチクエスチョンに取り組む。

4.1 RQ1: 他の開発者の Politeness は開発者の活動継続性に関係するのか?

動機: 開発者は, 円滑な人間関係を確立・維持するための相手への配慮を適切に表現するために言葉遣いを工夫する必要がある。たとえば, 人間関係が確立していない相手から Politeness の低い表現を受け取ると, 聞き手は不快に感じてしまい相手に対して距離を置いてしまうことが考えられる。また, 継続的に活動を行うためにはすでに確立した人間関係を維持することが必要であるため, つねに Politeness の高い表現を受け取ることが必要であると考えられる。RQ1 では, 開発者の活動期間による, 他の開発者から受信したメールに含まれる Politeness 値の違いを分析する。

アプローチ: RQ1 では, 活動期間の異なる長期開発者と短期開発者が受信したメールの Politeness 値を比較する (長期開発者と短期開発者の定義は 5.2 節で述べる)。ただし, 開発者ごとに受信メール数が異なるため, 各開発者が受信した全メールの Politeness 値の平均値を算出し, 長期開発者と短期開発者の Politeness 値の分布を比較する。長期開発者と短期開発者の受信メールの Politeness に違いがあることを確認することで, 他の開発者の Politeness と開発者の活動継続性に関係があることを明らかにすることができる。

4.2 RQ2: 開発者自身の Politeness は開発者の活動継続性に関係するのか?

動機: RQ1 では開発者の活動期間による他の開発者の Politeness 値の違いを確認したが, 良好な人間関係を確立する過程では開発者自身も Politeness の高い表現をする必要があると考えられる。RQ2 では, 開発者の活動期間による, 開発者自身が送信したメールに含まれる Politeness 値の違いを分析する。

アプローチ: RQ2 では, 長期開発者と短期開発者が送信したメールの Politeness 値を比較する。本論文では, 他の

開発者からのメールに対する返信メールと自らが機能追加などを提案する発信メールをまとめて, 開発者自身が送信したメールとする。また, RQ1 と同様に, 開発者ごとに送信メール数が異なるため, 各開発者が送信した全メールの Politeness 値の平均値を算出し, 長期開発者と短期開発者の Politeness 値の分布を比較する。

4.3 RQ3: Politeness の変動が長期開発者の継続性と離脱率に関係するのか?

動機: RQ1, RQ2 では, 長期開発者と短期開発者が受信, または送信したメールの Politeness 値に違いがあるのかを調査する。一方で RQ3 では, プロジェクトにおける活動を経て, 協調作業を行う他者からの配慮や開発者自身の他者へ対する配慮が変動するか否かを明らかにする。たとえば, プロジェクトで活動する時間が長くなるにつれて, 徐々に他の開発者から尊敬の配慮を含むメールが増加することが考えられる。一方で, 開発者自身は, 慣れにより相手への配慮が低下していくことが考えられる。RQ3 では, 長期開発者が受信, または, 送信したメールに含まれる Politeness 値が変化する過程を分析する。

アプローチ: 長期開発者の中でも送信メール数の最も多い開発者 5 名を対象に, 活動時期による受信, または, 送信したメールの Politeness 値の推移を分析する。送信メール数の最も多い開発者 5 名を対象とする理由は, 活発にコミュニケーションを行う開発者ほど, メール中に含まれる Politeness 値の変動についての詳細な調査が可能になるためである。

5. ケーススタディ

5.1 データセット

本ケーススタディでは, 2 つの大規模 OSS (Apache HTTP Server, Python) プロジェクトを対象とする。表 2 にデータセットの概要を示す。本ケーススタディにおいて両プロジェクトを選択した主な理由は以下のとおりである。

- 長期にわたり存続しているプロジェクトであり, 開発者の活動継続性を分析する対象として適していること
- ボランティア開発者が中心であるため, 開発者の Politeness に対して役職や階級などの社会的地位による影響が少ないこと
- 同じプロジェクトを対象とした多数の先行研究があ

表 2 対象プロジェクト
Table 2 Target projects.

対象プロジェクト	対象期間	対象開発者数		対象メール件数	
		長期開発者	短期開発者	受信メール	送信メール
Apache HTTP Server	1995/09/11~2016/03/30	183	173	82,190	109,685
Python	1999/04/02~2016/10/13	227	183	93,874	106,993

り、ケーススタディにより得られた知見の妥当性を確保しやすいこと

- 開発者メーリングリストを通じてやり取りされたメールデータがすべてアーカイブされており、かつ、容易に取得可能であること

5.2 分析手順

メールデータの取得. 本論文が分析対象とする開発者間のメールデータは、対象プロジェクトがホームページで mbox ファイル形式で公開している*6,*7。当該データセットには、メッセージ ID、プリメッセージ ID、全送信者名、全送信者のメールアドレス、件名、全送信日時、本文が含まれる。本論文では開発者の配慮を表現する語彙を抽出するため、著者らはメールの本文に含まれる署名や引用文 (“On XXX (人名) wrote:” など) を可能な限り除去しているが、開発者によって記述形式が異なるため完全に除去することはできていない。

開発者の抽出. OSS プロジェクトに参加する開発者の中には、複数のメールアドレスを利用している者がいるため、著者らは名前、メールアドレス、署名を目視で確認し、同一人物が利用するメールアドレスを特定し、開発者の抽出を行った。

開発者の活動期間と活動量の算出. 開発者を長期開発者と短期開発者に分類するために、開発者の活動期間を算出する。本論文では、開発者のメーリングリストへの最初の投稿日から、分析対象期間中の最後の投稿日までを活動期間とする。ただし、開発者は必ずしも継続してメールを送信しているとは限らない。活動期間中、数年間メールを送信していない開発者も存在し、このような開発者は継続的に活動しているとは考えにくい。本論文では、メールの送信間隔が 90 日以内であれば継続して活動していると見なし分析対象期間中に連続して送信した日数で最長の期間を開発者の活動期間とする。また、メールの送信数が少ない、または、活動期間が短い開発者の配慮を計測することはできないため、活動期間中に 10 件以上のメールを送信しており、かつ活動期間 90 日以上を開発者を対象とする。

長期開発者と短期開発者の分類. 本論文では、活動期間に基づき長期開発者と短期開発者を分類する。長期開発者と

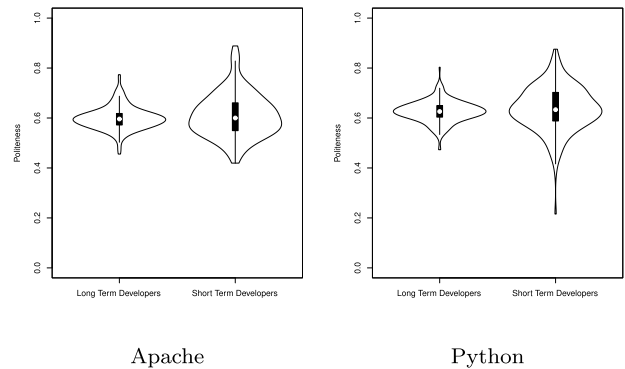


図 2 長期開発者と短期開発者の Politeness 値の分布 (受信メール)
Fig. 2 Violin plots of Politeness scores of long-term and short-term developers (received emails).

表 3 Politeness 値 (受信メール) の統計量と検定結果
Table 3 Statistics of Politeness scores (received emails) and result of Mann-Whitney U test.

プロジェクト	長期開発者			短期開発者			p 値
	中央値	最大値	最小値	中央値	最大値	最小値	
Apache	0.60	0.77	0.46	0.60	0.89	0.42	0.44
Python	0.63	0.80	0.47	0.63	0.88	0.22	0.06

短期開発者のサンプル数に偏りが出ないようにするために、活動期間の閾値を 270 日とし、活動期間が 270 日以上を開発者を長期開発者、活動期間 270 日未満の開発者を短期開発者とする。

送受信メールの Politeness 値の算出と長期開発者と短期開発者の Politeness 値の比較. 長期開発者と短期開発者の送信メールと受信メールを対象に、Stanford Politeness を用いて Politeness 値を算出する。算出した値をもとに長期開発者と短期開発者の Politeness 値の比較を行うが、開発者ごとにメール数に個人差が見られるために、Politeness 値の平均値を用いて比較する。長期開発者と短期開発者の Politeness 値に統計的有意差があるか否かを確認するために、マンホイットニーの U 検定を用いる。

5.3 分析結果

5.3.1 RQ1: 他の開発者の Politeness は開発者の活動継続性に関係するのか?

図 2 は、両対象プロジェクトの長期開発者と短期開発者の受信メールに対する Politeness 値の平均値のバイオリンプロットを示す。表 3 には、長期開発者の Politeness と短期開発者の Politeness の統計量 (中央値, 最大値, 最小値)

*6 Apache Mailing List: http://mail-archives.apache.org/mod_mbox/httpd-dev/

*7 Python Mailing List: <https://mail.python.org/mailman/listinfo/python-dev>

とマンホイットニーの U 検定の結果を示す。

Apache HTTP Server プロジェクト, および, Python プロジェクトにおいて, 開発者が受信したメールの Politeness 値の中央値は, 短期開発者と長期開発者に違いはなく, 長期開発者と短期開発者の間には統計的有意差を確認できなかった。

RQ1 の回答: 長期開発者と短期開発者が受信したメールの Politeness に違いはなく, 他の開発者の Politeness は開発者の活動継続性に関係ない。

5.3.2 RQ2: 開発者自身の Politeness は開発者の活動継続性に関するのか?

図 3 は両対象プロジェクトの長期開発者と短期開発者の送信メールに対する Politeness 値の平均値のバイオリンプロットを示す。表 4 は長期開発者の Politeness と短期開発者の Politeness の統計量 (中央値, 最大値, 最小値) とマンホイットニーの U 検定の結果を示す。

Apache HTTP Server プロジェクト, および, Python プロジェクトにおいて, 開発者が送信したメールの Politeness 値の中央値は, 短期開発者の方が長期開発者に比べて高く, 長期開発者と短期開発者の間に統計的有意差を確認した。

RQ2 の回答: 長期開発者と短期開発者が送信したメールの Politeness に違いがあり, 開発者自身の Politeness と開発者の活動継続性に関係がある。

5.3.3 RQ3: Politeness の変動が長期開発者の継続性と離脱率に関するのか?

図 4 は, 送信メール数の最も多い長期開発者 5 名の, 各月に送信したメール (左図), および, 受信したメール (右

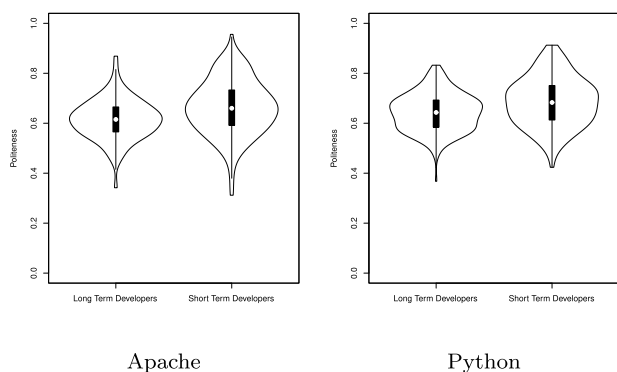


図 3 長期開発者と短期開発者の Politeness 値の分布 (送信メール)
Fig. 3 Violin plots of Politeness scores of long-term and short-term developers (sent emails).

表 4 Politeness 値 (送信メール) の統計量と検定結果

Table 4 Statistics of Politeness scores (sent emails) and result of Mann-Whitney U test.

プロジェクト	長期開発者			短期開発者			p 値
	中央値	最大値	最小値	中央値	最大値	最小値	
Apache	0.62	0.87	0.34	0.66	0.96	0.31	0.00
Python	0.64	0.83	0.37	0.68	0.91	0.42	0.00

図) の平均 Politeness 値の推移を示す。青線は Politeness 値, 赤線はメール送信数, および, メール受信数を示す。なお, メール送信数, 受信数が少ない月は外れ値となり推移を理解することが困難となるため, 各月の Politeness の値, および, メール件数は, 当該月の前後の月を含む 3 カ月間の平均値で示す。また, 点線の赤丸は Politeness 値が急激に変動している箇所を示している。

[離脱した長期開発者] 長期開発者の中で離脱直前に送信メール, および, 受信メールの Politeness 値が急激な変動 (0.15 以上の上昇, または, 下降) をする開発者を確認した。0.15 を閾値とした理由は, 図 2 および図 3 に示したように, 両プロジェクトの受信メールおよび送信メールの Politeness 値の分布における四分位範囲 (interquartile range, IQR) はいずれも 0.15 以下であり, 0.15 以上の上昇・下降は大きなばらつきを意味することから急激な変動があったと見なせると考えたためである。Apache HTTP Server プロジェクトにおける Dev#3 は送信メールの Politeness 値 (図 4(e)) が下降している。また同プロジェクトの Dev#4 は, 送信メールの Politeness 値 (図 4(g)) は上昇し, 受信メールの Politeness 値 (図 4(h)) は下降している。Python プロジェクトの Dev#4 は, 送信メール, 受信メールともに Politeness 値 (図 4(q), 図 4(r)) が上昇している。このような変動は, プロジェクトから開発者が離脱する兆候である可能性がある。実際にどのようなメールによって Politeness 値が変動しているのかを 6.2 節で調査する。

[離脱していない長期開発者] Apache HTTP Server プロジェクトにおける Dev#1, Python プロジェクトにおける Dev#1, Dev#3, Dev#5 は, 長期開発者の中でも特に長く活動し, 分析対象期間において離脱することがなかった。離脱していない開発者のうち, Python の Dev#1 と Dev#3 の送信メールの Politeness 値 (図 4(k), 図 4(o)) は, 開発者の参加開始から現在に至るまで急激な変動は確認されなかった。したがって, Politeness 値が急激に変動しない開発者は継続して活動する可能性がある。一方で, 離脱していない開発者のうち, Apache の Dev#1 と Python の Dev#5 の送信メールの Politeness 値 (図 4(a), 図 4(s)) は, 活動期間中に急激な変動があることが確認されている。これらの開発者は, Politeness 値の急激な変動が生じた時期にコミュニティ内で何らかの問題が発生したが, その問題に対して適切な対処が行えたため離脱せず現在に至るまで継続して活動を行えているのではないかと考えられる。

RQ3 の回答: 長期的に Politeness 値が変動しない開発者は離脱せずに, 継続して活動する可能性がある。一方で, Politeness 値の急激な変動は, プロジェクトから開発者が離脱する兆候である可能性がある。

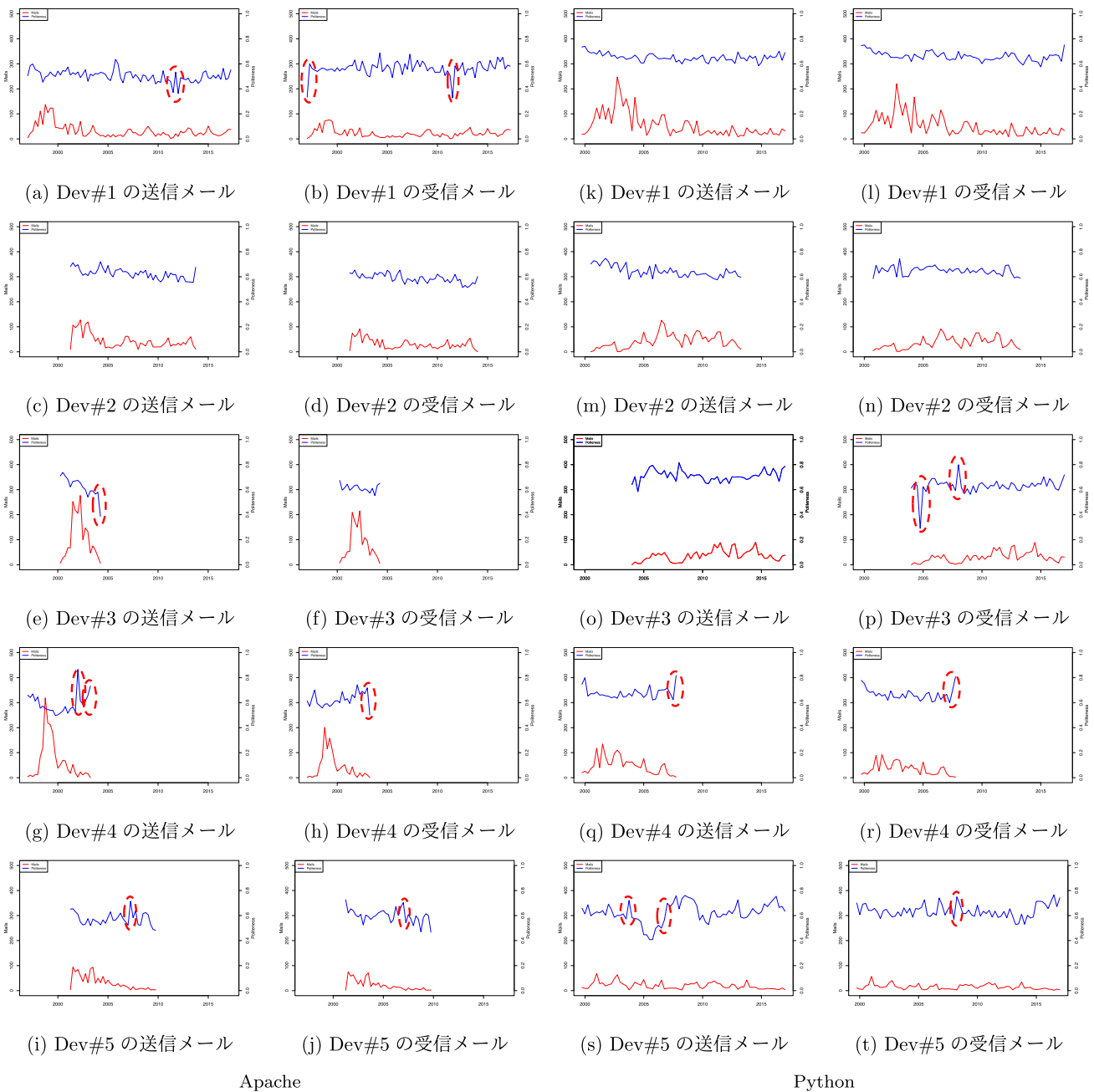


図 4 長期開発者の送信/受信メールの Politeness 値の推移
Fig. 4 Time series transition in Politeness scores (sent and received emails) for long-term developers.

6. 考察

6.1 長期開発者と短期開発者の送信メールに含まれる Politeness

RQ2の結果、短期開発者に比べ長期開発者が送信したメールの Politeness 値は有意に低い値をとることが確認された。しかしながら、中央値の差自体は大きくない。開発者の活動継続性を Politeness 値のみで説明できるほどの大きな差ではないため、開発者の自身の Politeness は活動継続性に影響を与える要因の1つとしてとらえるのが妥当である。継続的に活動する開発者は、良好な人間関係を維

持するために Politeness の高い表現を使うと考えていたため、長期開発者の方が短期開発者よりも自身の Politeness が低いという結果は予想外の結果であった。この結果は、長期開発者と比べてコミュニティ内での議論が不慣れた短期開発者は、曖昧な表現を用いやすいために Politeness の高い表現を行っているからであると考えられる。

実際のメール内容を確認したところ、短期開発者は「私は不具合を見つけたと思います。」*8や「それは問題だと思

*8 http://mail-archives.apache.org/mod_mbox/httpd-dev/200403.mbox/%3C404F6CD3.8080904%40openwave.com%3E

います。]*⁹というような曖昧な表現を使用することが多く、このような曖昧な表現は、Politeness 分析において高い値を出力する。一方で、長期開発者は、開発者間で誤解を引き起こす曖昧な表現を避けることが多く、たとえば「何が不具合の原因になっているのか?」*¹⁰という表現を用いることが多いが、このような表現には Politeness を高める働きがない。結果的に、プロジェクトに長期間参加した開発者自身の Politeness は短期開発者に比べて低い値となったと考えられる。

また、RQ3 の結果から長期開発者が送信するメールの Politeness 値は時期により変動することが分かった。しかしながら、すべての開発者において活動期間が経過するにつれて送信メールの Politeness 値が下降するわけでない。このことから、プロジェクト参加直後から Politeness の低い表現を使っている開発者が継続的に活動を行う傾向があると考えられる。

6.2 開発者の活動時期による送受信メールの Politeness の変動

RQ3 の結果、Politeness 値の変動を計測することにより、継続的にプロジェクトで活動するか否かを判断できる可能性があることが分かった。本節では、Apache HTTP Server プロジェクトの長期的に活動した Dev#1, Dev#3, Dev#4 について、送信、または受信メールに含まれる Politeness Strategy の割合を時系列分析する。図 5, 図 6 には、Apache HTTP Server の長期開発者、短期開発者の、送信、または受信メールに含まれる 18 種類の Politeness Strategy の時系列推移を示す。赤色から黄色で示しているのは Politeness 値を上げる Strategy, 青色で示しているのは Politeness 値を下げる Strategy である。

Dev#1 は、プロジェクトから離脱することなく活動した長期開発者である。プロジェクト参加直後の Dev#1 の受信メールの Politeness 値は低いが、その後は 0.6 前後の Politeness 値を持つメールを受信している。Dev#1 の Politeness Strategy の語彙を分析した結果、“Thanks”, “great”, “Nice” といった Politeness 値を上げる語彙が使用されており、Dev#1 のパッチ投稿などの活動についての感謝のメールを多数受け取っていることが確認された。また、送信メールに関して、全活動期間において時期によって少しの変動はあるものの、Politeness Strategy の割合はほぼ一定である。具体的には、“Thanks” や “Please” を含む Politeness 値を上げる表現を用いつつ、“What”, “But” を含む Politeness 値を下げる表現も用いている。このこと

から、Dev#1 は他の開発者への配慮を行いながら指摘を行いつつ開発の議論を行っているため、プロジェクトから離脱することなく活動を行っていると考えられる。紙面の都合上、掲載していないが、Python プロジェクトの Dev#1, Dev#3, Dev#5 も同様の結果であった。

一方で、プロジェクトを離脱した Dev#4 は、プロジェクトを離脱する前に送信メールの Politeness 値が一度下降した後を上昇している。Dev#4 の Politeness Strategy の語彙を分析した結果、Politeness 値が下降している期間は、“What”, “Why”, “But”, “So” を含む Politeness 値を下げる語彙の使用が増えていること、Politeness 値が上昇した期間は “What”, “Why”, “But”, “So” を含む Politeness 値を下げる語彙の使用が減り、Politeness 値を上げる語彙の使用が増えていることが確認された。

Dev#3 も同様に、プロジェクトを離脱しているが、Dev#4 と異なり、プロジェクトを離脱する直前の送信メールの Politeness 値が下降している。Politeness Strategy の語彙を分析した結果、Politeness 値を下げる語彙が使用されており、具体的には、「私が書いた記述を見直してください」や「いや、これが解決作です。あなたは説明をしていない。」というメールが確認された*¹¹。

短期開発者は、Dev#1 と同様に、Politeness 値を上昇させる語彙、下降させる語彙が同時期に使用されている。しかし、活動期間が短いため、Politeness 値の変動をとらえることは難しい。今後の研究では、継続するか否かを判断するために分析すべき Politeness 値の期間を調査する。

6.3 制約

本論文では、対象プロジェクトにおいて開発者がコミュニケーションをとるために用いているメーリングリストのデータを使用した。しかし、Apache HTTP Server プロジェクトが公開するデータセットにおいて、約 5% の返信メールに各メールが保持するはずの ID が記載がされていなかった。したがって、これらのメールについては、返信メールと送信メールの区別ができていない。また、5 章で述べたとおり、開発者の配慮を正確に計測するために、機械的に生成された文章を排除しているが、少数のメールにのみ含まれる定型文を完全に排除できていない。

また、メーリングリストのコミュニケーションデータのすべての期間を対象期間にしているが、開発が活発な時期や保守作業のみの時期による Politeness 値の違いは考慮していない。開発の活発度によってメールの送信頻度、使用される語彙の違いが発生する可能性があるが、それは今後

*⁹ http://mail-archives.apache.org/mod_mbox/httpd-dev/200104.mbox/%3C3AE0AE2D.A12CEAF1%40omniti.com%3E

*¹⁰ http://mail-archives.apache.org/mod_mbox/httpd-dev/200210.mbox/%3CPine.LNX.4.21.0210122333450.16766-100000%40shell.ntrnet.net%3E

*¹¹ http://mail-archives.apache.org/mod_mbox/httpd-dev/200210.mbox/%3CPine.LNX.4.21.0210122333450.16766-100000%40shell.ntrnet.net%3E
http://mail-archives.apache.org/mod_mbox/httpd-dev/200212.mbox/%3CPine.LNX.4.33L2.0212210949220.18361-100000%40vangogh.rkbloom.net%3E

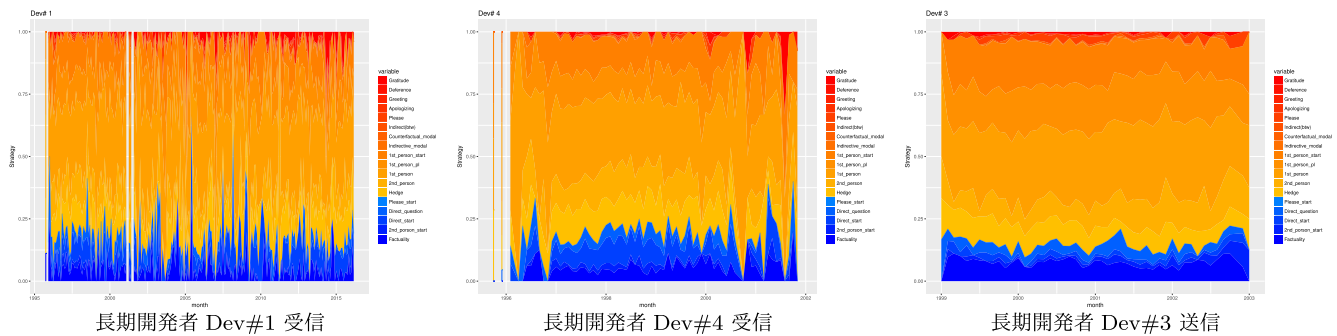


図 5 Apache の長期開発者 Dev#1, Dev#4 の受信メール, Dev#3 の送信メールの Politeness Strategy の時系列推移

Fig. 5 Time series transition in Politeness Strategies for long-term developer Dev#1 and Dev#4 (received emails) and developer Dev#3 (sent emails) in Apache.

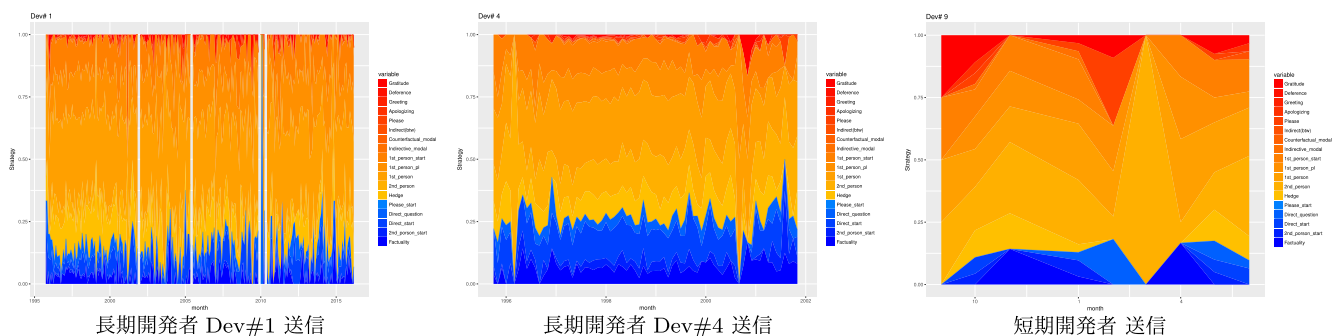


図 6 Apache の長期開発者 Dev#1, Dev#4, 短期開発者の送信メールの Politeness Strategy の時系列推移

Fig. 6 Time series transition in Politeness Strategies for long-term developer Dev#1 and Dev#4 (sent emails) and short-term developers (sent emails) in Apache.

の課題とする。

7. おわりに

本論文では、OSS 開発プロジェクトで交わされるコミュニケーションに着目し、開発者が継続的に活動を行うことを可能にしている要因を明らかにするために、開発者の送受信メールを対象に良好な人間関係を確立・維持するための配慮を表す Politeness を分析した。Apache HTTP Server と Python を対象にしたケーススタディに取り組んだ結果、長期開発者の方が短期開発者よりも送信メールの Politeness 値が有意に低いことが分かった。しかしながら、開発者の活動継続性を Politeness 値のみで説明できるほどの大きな差ではないため、開発者の自身の Politeness は活動継続性に影響を与える要因の 1 つとしてとらえるのが妥当である。

本論文では、開発者間のコミュニケーションに着目したが、今後、開発者間の人間関係、および、パッチ投稿、コードレビューなどの技術的貢献を分析することで、Politeness 値が変動する要因を明らかにし、短期開発者が継続的に貢献するための方法論の確立を目指す。

謝辞 本研究の一部は、文部科学省科学研究補助金（基盤（C）：15K00101）による助成を受けた。

参考文献

- [1] Bird, C., Gourley, A., Devanbu, P., Swaminathan, A. and Hsu, G.: Open Borders? Immigration in Open Source Projects, *Proc. 4th Int'l Workshop on Mining Software Repositories (MSR '07)* (2007).
- [2] Zhou, M. and Mockus, A.: Developer Fluency: Achieving True Mastery in Software Projects, *Proc. 18th ACM SIGSOFT Int'l Symp. Foundations of Soft. Eng. (FSE '10)*, pp.137-146 (2010).
- [3] Zhou, M. and Mockus, A.: What Make Long Term Contributors: Willingness and Opportunity in OSS Community, *Proc. 34th Int'l Conf. Soft. Eng. (ICSE '12)*, pp.518-528 (2012).
- [4] Zhou, M. and Mockus, A.: Does the Initial Environment Impact the Future of Developers?, *Proc. 33rd Int'l Conf. Soft. Eng. (ICSE '11)*, pp.271-280 (2011).
- [5] Gharehyazie, M., Posnett, D. and Filkov, V.: Social Activities Rival Patch Submission For Prediction of Developer Initiation in OSS Projects, *Proc. 2013 IEEE Int'l Conf. Soft. Maintenance (ICSM '13)*, pp.340-349 (2013).
- [6] Brown, P. and Levinson, S.C.: *Politeness: Some universals in language usage*, Cambridge University Press (1987).
- [7] Ye, Y. and Kishida, K.: Toward an Understanding of the Motivation Open Source Software Developers, *Proc. 25th Int'l Conf. Soft. Eng. (ICSE '03)*, pp.419-429 (2003).
- [8] Steinhilber, I., Conte, T., Gerosa, M.A. and Redmiles, D.: Social Barriers Faced by Newcomers Placing Their

- First Contribution in Open Source Software Projects, *Proc. 18th ACM Conf. Computer Supported Cooperative Work & Social Computing (CSCW '15)*, pp.1379-1392 (2015).
- [9] Bird, C., Gourley, A., Devanbu, P., Gertz, M. and Swaminathan, A.: Mining Email Social Networks, *Proc. 2006 Int'l Workshop on Mining Software Repositories (MSR '06)*, pp.137-143 (2006).
- [10] Bird, C., Pattison, D., D'Souza, R., Filkov, V. and Devanbu, P.: Latent Social Structure in Open Source Projects, *Proc. 16th ACM SIGSOFT Int'l Symp. Foundations of Soft. Eng. (FSE '08)*, pp.24-35 (2008).
- [11] Dabbish, L., Stuart, C., Tsay, J. and Herbsleb, J.: Social Coding in GitHub: Transparency and Collaboration in an Open Software Repository, *Proc. ACM 2012 Conf. Computer Supported Cooperative Work (CSCW '12)*, pp.1277-1286 (2012).
- [12] Qureshi, I. and Fang, Y.: Socialization in Open Source Software Projects: A Growth Mixture Modeling Approach, *Organizational Research Methods*, Vol.14, No.1, p.208 (2011).
- [13] Shibuya, B. and Tamai, T.: Understanding the Process of Participating in Open Source Communities, *Proc. 2009 ICSE Workshop on Emerging Trends in Free/Libre/Open Source Software Research and Development (FLOSS '09)*, pp.1-6 (2009).
- [14] Danescu-Niculescu-Mizil, C., Sudhof, M., Jurafsky, D., Leskovec, J. and Potts, C.: A computational approach to politeness with application to social factors, *Proc. 51st Annual Meeting of the Association for Computational Linguistics (ACL '13)* (2013).
- [15] Gunn, S.: Support Vector Machines for Classification and Regression, Technical Report, School of Electronics and Computer Science, University of Southampton (1998).

推薦文

本論文は、コミュニケーションを円滑化し人間関係を維持するための配慮（本論文における Politeness）を定量化する分析手法 Politeness Analysis を用いて、オープンソースソフトウェア開発コミュニティの持続的進化に重要となる開発者同士のコミュニケーションの質的側面について分析したケーススタディをまとめたものである。ケーススタディでは、20年以上の歴史を有する Apache HTTP Sever プロジェクトのトップ開発者 10 名の総計 123,538 件分のコミュニケーションデータに対して Politeness Analysis を適用し、開発者ごとに Politeness 値に個人差があること、プロジェクトの時期によって Politeness 値にも変化があること、Politeness 値の急激な変化と開発者のプロジェクト離脱に関連がある可能性があることなどを示している。研究自体はこれからまだ発展できる余地が残されているものの、大規模コミュニケーションデータに対して Politeness Analysis を導入することで、オープンソースソフトウェア開発コミュニティに限らず様々な組織内コミュニケーションの質的側面を計測できる可能性を示した点についての貢献は大きく、高く評価できる優れた論文であるため、推薦する。

（グループウェアとネットワークサービス研究会主査
市村 哲）



宮崎 智己

平成 29 年和歌山大学システム工学部情報通信システム学科卒業。現在、同大学大学院システム工学研究科博士前期課程在学中。学士（工学）。ソフトウェア工学の研究に従事。



伊原 彰紀（正会員）

平成 19 年龍谷大学理工学部電子情報学科卒業。平成 21 年奈良先端科学技術大学院大学情報科学研究科博士前期課程修了。平成 24 年同大学院同研究科博士後期課程修了。同年より同大学院情報科学研究科助教。博士（工学）。ソフトウェア工学、特にオープンソースソフトウェア開発支援の研究に従事。電子情報通信学会、日本ソフトウェア科学会、IEEE 各会員。



大平 雅雄（正会員）

平成 10 年京都工芸繊維大学工芸学部電子情報工学科卒業、平成 15 年奈良先端科学技術大学院大学情報科学研究科博士後期課程修了。同大学情報科学研究科助教を経て、平成 24 年和歌山大学システム工学部准教授。博士（工学）。ソフトウェア工学、特にマイニングソフトウェアリポジトリの研究に従事。電子情報通信学会、ヒューマンインタフェース学会、ACM 各会員。



東 裕之輔

平成 27 年和歌山大学システム工学部情報通信システム学科卒業。平成 29 年和歌山大学大学院システム工学研究科前期課程修了。修士（工学）。



山谷 陽亮

平成 26 年和歌山大学システム工学部情報通信システム学科卒業。平成 28 年和歌山大学大学院システム工学研究科前期課程修了。修士（工学）。