OSS コミュニティにおけるオープンコラボレーションの理解

Understanding Open Collaboration in OSS Communities

松本 真佑（まつもと しんすけ・Shinsuke Matsumoto）[1]・亀井 靖高（かめい やすたか・Yasutaka Kamei）[2]
大平 雅雄（おおひら まさお・Masao Ohira）[3]・松本 健一（まつもと けんいち・Ken-ichi Matsumoto）[4]

[1,2]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
[3]奈良先端科学技術大学院大学 情報科学研究科　助教
[4]奈良先端科学技術大学院大学 情報科学研究科　教授

［Abstract］
Open Source Software (OSS) communities provide us with high quality and functional software products which are freely available and are equivalent to proprietary software. Some of the products such as Linux and Apache are widely used in our daily life as a part of the social infrastructure. Although many studies have analyzed the success factors of OSS communities and products from a variety of perspectives, it is still unclear how open collaboration among developers and users in an OSS community works. The goal of this study is to provide a clear understanding of open collaboration in an OSS community. In this study, the relationship between the quality of open collaboration and the growth of a community was analyzed. Using a technique of social network analysis, three kinds of the following sub-communities were compared: developer community, user community and coordinator community. As a result, the existence of coordinators who contributed to the intermediation of collaboration between developers and users was discovered to be the key factor for the growth of an OSS community.

［キーワード］
オープンコラボレーション、OSS 開発コミュニティ、Social Network Analysis、中心性、コーディネータ

1.　はじめに

近年、Open Source Software（OSS）コミュニティによるソフトウェアの開発形態が注目を浴びている。Raymond らは、OSS コミュニティの成功例として Linux を取り上げ、その開発形態をバザール方式と名付けた[18]。バザール方式では、バザールでの売買のようにコミュニティの参加者を限定せず、ルールや指揮系統が少ない自由でオープンなコラボレーションによって開発が進められる。Web サーバーソフトウェアである Apache はバザール方式で開発された OSS 開発の成功例である。2008 年 8 月現在、Apache は Web サーバ市場のシェアの約半数の世界一のシェアを誇っており、社会基盤として広く利用されている[15]。

このような OSS を開発したコミュニティが社会的に大きな注目を集める傍らで、大多数の OSS コミュニティは停滞や失敗に終わっている[16]。OSS の開発環境を無料で提供する SourceForge. net[1]では、登録されている 10 万件以上の OSS プロジェクトのうち、開発者が 3 人以下の小規模コミュニティが全体の 9 割を占めている[17]。このような少人数のコミュニティでは、開発者をはじめとする参加者のモチベーションの維持や開発資源（開発時間や開発に必要となる技術や知識など）の確保が困難であり、参加者同士のコラボレーションが難しい。一方、多くの参加者を抱える大規模コミュニティであっても、参加者の持つ知識や意見が互いに伝わりにくいコミュニケーション構造である場合にも円滑なコラボレーション作業は困難である。このように、バザール方式を取り入れただけではコミュニティが活発に活動するとは限らず、コミュニティ内の調整役の存在や正統的周辺参加[11]といったコミュニケーション形態の重要性が指摘されている[18][22]。

[1] http://sourceforge.net/

一般的な OSS 開発では、E-mail や掲示板などのオープンなメディアを用いてユーザがソフトウェアに対する意見や要望を開発者に伝え、そのフィードバックを次の成果物に反映するというサイクリックでかつオープンなコミュニケーション形態がとられる。開発者とユーザの自由でオープンなコラボレーションによる OSS の開発プロセスが、品質や機能といった面で商用ソフトウェアにも引けをとらないソフトウェアの実現を可能にしているといえる。

近年、OSS コミュニティの成功要因を明らかにしようとする研究が盛んに行われているが[8][12][14][22]、OSS コミュニティの参加者同士が、実際にどのようなコミュニケーション構造を築いているかは明らかにされていない。本論文では OSS コミュニティの成長過程において、コミュニティ内におけるオープンコラボレーションの違いがその後のコミュニティの成長にどう関与するかを理解することを目的として、3 つの有名な OSS コミュニティを対象にコミュニティの参加者同士が形成するコミュニケーション構造についての分析を行い、その特徴を観察する。OSS コミュニティにおけるオープンコラボレーションの理解により、社会基盤として重要な役割を占めつつある OSS 開発の更なる発展が期待される。

以降、2 章で本論文におけるオープンコラボレーション分析の観点を提示し、3 章でその分析手法について述べ、4 章で実際の OSS コミュニティを分析したケーススタディについて報告する。最後に 5 章で本論文のまとめを述べる。

２．OSS コミュニティにおけるオープンコラボレーション

２．１ OSS コミュニティの開発形態

Open Source Software（OSS）とは、ソフトウェアの設計図にあたるソースコードを無償で公開し、自由に改良や再配布を行えるようにしたソフトウェアのことを指す。OSS 開発コミュニティによるソフトウェア開発は、一般的なソフトウェア開発企業が取る開発環境と比べて以下のような特徴がある[3]。

（1）開発者が自由に参加／脱退することが可能

OSS コミュニティは不特定多数の開発者が自由に参加／脱退することが可能である。OSS 開発の初期段階ではコミュニティの創立者を中心に成果物（ソフトウェア）が作り出されるが、開発が進むにつれ多数の開発者が参加しコミュニティが拡大する。大規模なコミュニティでは開発者が 200 人以上になることもある[17]。

（2）ボランティアでの参加が基本

参加する開発者への対価として金銭的な報酬がある場合は極稀であり、その参加動機としては世界中の参加者との協調作業の魅力や、開発技術の学習や共有などの個人の知的好奇心の充足による場合が多い[2][7][23]。OSS コミュニティへの参加により得られるこれらの対価は、参加するコミュニティの規模が大きく参加者が密なコミュニケーションを取るほど大きく、モチベーションの維持に繋がる。

（3）厳格な指揮系統が存在しない

開発者が自由に参加するため開発者間での厳格な指揮系統は存在せず、バザールでの売買のような個人中心で自由な環境で開発が行われる[18]。このため開発者のコミュニケーションネットワークは、木構造のような整理された指揮系統ではなく、動的に参加者が増減し相互に絡み合う複雑な構造を取る。

（4）ユーザ参加型コミュニティ

OSS コミュニティではユーザが重要な役割を果たす。ユーザからの反応は外部市場からの評価であり、開発者が社会的なニーズに対応しているという実感を与える[7][18]。つまり、ユーザは成果物の享受者であると同時に、開発者へのモチベーションの提供者でもある。また、ユーザによるソフトウェアの欠陥の発見と報告という行為の重要性から、ユーザを共同開発者と位置付ける場合も多い[18]。このようにユーザは OSS コミュニティ内において極めて重要な役割を果たす。

（5）ネットワークを介した分散開発環境

地理的に離れた世界中の参加者が、ネットワークを介して開発を行う。したがって、コミュニケーションを取

るための手段としては、メーリングリスト（ML）や掲示板などの非対面でかつ非同期な媒体が基本となる。つまり、ML や掲示板などのコミュニケーションメディアが参加者のコミュニケーションネットワークに大きな影響を与える[21]。

２．２ 分析の観点

OSS におけるオープンコラボレーションを理解するために、以下の 4 つの観点から分析を行う。

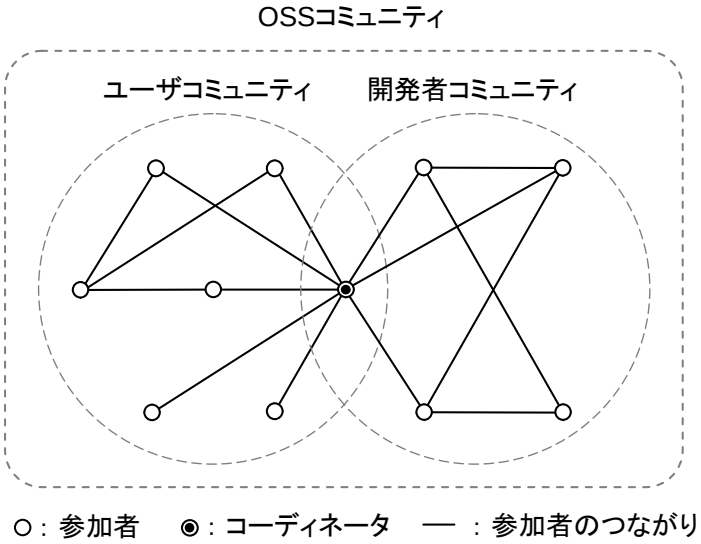


図-1 OSS コミュニティの組織構成

観点 1. 時間の経過に伴うコミュニケーションネットワーク全体の変化

本論文ではコミュニケーションネットワークの時間的な変化について以下のような点について着目し、分析を行う。

- 開発者ネットワークとユーザネットワークの違い。
- リリース時期前後でのネットワークの違い。

分析手段としては社会構造の分析に良く用いられる Social Network Analysis (SNA)を用いる。SNA の詳細については次章で述べる。

観点 2. コーディネータのコミュニケーション活動の変化

バザール方式を用いた OSS コミュニティが活発に活動し続けるためには、全く指揮系統のない状態ではなく、ある程度の調整を務める人物が重要であるとされている[6][10][18]。また、Ye らは OSS コミュニティの新規参加者が参加者同士の学習を通じてコミュニティへの帰属意識を持ち、次第に重要な役割へと変化するという参加形態（正統的周辺参加[11]）が OSS 参加者のモチベーションに繋がるとしている[22]。

本論文では、OSS コミュニティにおけるオープンコラボレーションを支える役割として、開発者とユーザの 2 つの参加者コミュニティを調整し媒介する人物（コーディネータ）の存在について着目する。図 1 は 2 つの参加者コミュニティとコーディネータの関係を表したものである。一般的に大規模な OSS 開発プロジェクトでは、図 1 に示す通り開発者用のメーリングリストとユーザ用のメーリングリストが個別に設けられている場合が多い。これは情報の混乱を避け、開発者（またはユーザ）にとっての知識や情報、議論の選別が容易に行えるというメリットがある一方で、お互いの情報共有や協調作業が困難になるという問題が発生しやすい。例えば、互いのメーリングリストの橋渡しとなる役割（図 1 に示すコーディネータに該当）が一切存在しない場合、ユーザからの意見や要求を開発者に伝える伝達経路が不足するため、協調作業の成果物であるソフトウェアの成長が困難となる。同様にコーディネータが不在のコミュニティでは、ユーザとして参加した新規参加者と高い技術を持つ古参開発者の協調作業が成立せず、コミュニティ内での正統的周辺参加が成り立たない。そのため、コミュニティそ

のもの成長も阻害される可能性がある。

図 1 に示すコーディネータのように、開発者としての深い知識を持ちながら、ユーザとのコミュニケーションを取ることは、コミュニティへのユーザの正統的周辺参加を促し、帰属意識を高めることに繋がる。逆に（開発コミュニティにとって外部の）ユーザコミュニティからの評価を開発者へ伝えることは、開発者のモチベーションの向上や市場が要求するソフトウェアの実現に繋がる。このように、OSS 開発においては開発者とユーザのコミュニティの協創活動が重要であり、それらを媒介するコーディネータがイノベーションの創出を促す重要な役割を占めると考えられる。

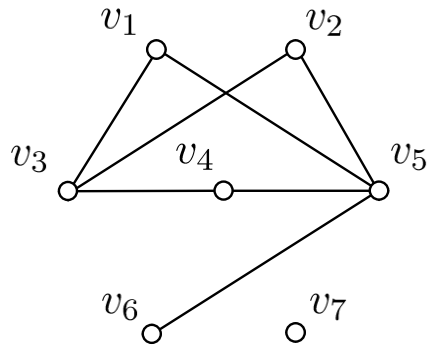
３．分析手法

３．１ ソーシャルネットワーク分析

本論文では人々の社会的関係がどのような特徴を有しているか構造的に分析するソーシャルネットネットワーク分析[19][20]を用いる。ソーシャルネットワークとは社会的な関連性をネットワークモデルとして表現ものである。このネットワークでは一個人や一組織をノードとして表し、個々のノード間の関係や関連性の有無をエッジとして表す。このように社会的な個人間、組織間の関連性をネットワークモデルとして表現することで、複雑な現実社会の現象の特徴の解釈を容易にすることが可能である。このソーシャルネットワークに対して従来研究されてきた数学的なネットワーク分析手法を用いて分析することをソーシャルネットワーク分析と呼ぶ。ソーシャルネットワーク分析を用いることで、その社会的ネットワーク全体の可視化や構造的な特徴の数値化、また個々の actor 同士の関係の強さやネットワーク内での中心人物の特定といった分析が可能となる。

本論文では、ソーシャルネットワーク分析が提供する指標のうち、3 つの中心性[4]に着目し、OSS コミュニティにおける各参加者のコミュニケーションネットワークの特徴を把握する。

以降、本章では分析手法を説明するための各種用語と、分析手法について述べる。



○：ノード　　──：エッジ

３．２ ネットワークの定義

本論文では、各点（ノード）とそれぞれを繋ぐ線（エッジ）の集合をネットワークとする。以下、ネットワーク全般の用語（次数、経路、距離）について図 2 を用いて説明する。

・ 次数

あるノードの持つエッジの数である。例えば、ノードv_1の次数は 2 である。

・ 経路

あるノードから他のノードへ到達するために通るノードの順路である。例えば、v_1からv_6への経路としては、

$v_1 \rightarrow v_5 \rightarrow v_6$ や $v_1 \rightarrow v_3 \rightarrow v_4 \rightarrow v_5 \rightarrow v_6$ などがある。あるノードから他のノードへ到達できる（経路が存在する）経路を到達可能な経路、到達できない（経路が存在しない）場合を到達不可能な経路と呼ぶ。例えば、v_1 から v_6 へは到達可能であり、v_1 から v_7 へは到達不可能である。

・ 距離

あるノードから他のノードへの経路の長さである。例えば、経路 $v_1 \rightarrow v_5 \rightarrow v_6$ の距離は 2 であり、経路 $v_1 \rightarrow v_3 \rightarrow v_4 \rightarrow v_5 \rightarrow v_6$ の距離は 4 である。v_i から v_j への経路のうち、距離が最も短い経路を最短経路、その距離を最短経路長と呼ぶ。例えば、v_1 から v_6 の最短経路は $v_1 \rightarrow v_5 \rightarrow v_6$ で、最短経路長は 2 である。

３．３ OSS コミュニティにおけるコミュニケーションネットワーク

OSS コミュニティはオンライン上の分散開発環境であるため、ML や掲示板を通じた非対面のコミュニケーションを基本とする。対面でのコミュニケーションが大きな役割を占める一般的なソフトウェア開発とは異なり、OSS コミュニティでは、ML や掲示板などのコミュニケーションメディアが参加者のコミュニケーションネットワークの形成に大きな影響を与える[21]。したがって、OSS コミュニティにおけるコミュニケーションネットワークは、ML や掲示板などでのメッセージの送信者と返信者の関係から構成されるとみなすことができる[1][21]。

本論文では、分析対象とするコミュニティでの参加者同士のメッセージの送受信関係をエッジとして OSS コミュニティのコミュニケーションネットワークを定義する。

３．４ ネットワークの中心性

本論文では、参加者が OSS コミュニティ内でどのような役割を果たしているかを観察するために、ソーシャルネットワーク分析が提供する指標のうち、3 つの中心性を用いる[4]。それぞれの中心性は、(a)次数中心性:各ノードがどの程度の情報を発しているか、(b)媒介中心性:各ノードがどの程度の情報を媒介しているか、(c)近接中心性:各ノードがネットワーク全体に対してどの程度効率よく情報を発しているかを表す指標である。中心性の指標は、電子メールなどのデータを基に企業内での従業員の構造的立場を分析するために利用された事例[5]がある。3 つ中心性を用いることにより、ネットワーク全体の構造の特徴や、個々のノードの性質について定量的な計測が可能となる。このように中心性を用いて定量的な計測により、個々のネットワークの構造や個々のノードの性質が時間の経過に従いどのように変化するかについても容易に理解が可能となる。以下ではソーシャルネットワーク分析において一般によく用いられる 3 つの中心性（次数中心性、媒介中心性、近接中心性）について説明する。

・ 次数中心性

ノード v_i の次数中心性 $C_{degree}(v_i)$ は、ネットワーク内でノード v_i が取りうる最大の次数によって、実際の v_i の次数を正規化した値である。次数中心性は、式（1）のように定式化できる。

$$C_{degree}(v_i) = \frac{deg(v_i)}{n-1} \qquad (1)$$

ここで、n はネットワーク内のノード数を、$C_{degree}(v_i)$ は v_i の次数を表す。

$C_{degree}(v_i)$ は最小 0 から最大 1 までの値を取り、値が 1 に近いほど次数中心性が高い。次数中心性が高いノードほど、他のノードと隣接しているエッジの数が多いノードである。

OSS コミュニティにおいては次数中心性が高い参加者を、多くの参加者へ情報を発信／受信している参加者とみなすことができる。

・ 媒介中心性

ノードv_iの媒介中心性$C_{betweenness}(v_i)$は、他の 2 つノードv_jからv_kへの最短経路にノードv_iが含まれる割合である。媒介中心性は、式 (2) のように定式化できる。

$$C_{betweenness}(v_i) = \frac{\sum_{j<k}^{n} \sum_{j<k}^{n} p_{jk}(v_i)}{\sum_{j<k}^{n} \sum_{j<k}^{n} p_{jk}} \qquad (2)$$

ここで、nはネットワーク内のノード数を、p_{jk}はv_jからv_kへの最短経路、$\sum_{j<k}^{n} \sum_{j<k}^{n} p_{jk}$は$v_j$から$v_k$への最短経路の総数を表し、$\sum_{j<k}^{n} \sum_{j<k}^{n} p_{jk}(v_i)$は$v_i$を含む$v_j$から$v_k$への最短経路の総数を表す。

$C_{betweenness}(v_i)$は最小 0 から最大 1 までの値を取り、値が 1 に近いほど媒介中心性が高い。媒介中心性が高いノードほど、他のノードの仲介者としての役割を果たすノードである。

OSS コミュニティにおいて媒介中心性が高い参加者ほど、他の参加者同士を繋ぎ合わせる参加者である。つまり、媒介中心性が高い参加者が突然いなくなると、OSS コミュニティでコミュニケーションが円滑に行われなくなると考えられる。

・ 近接中心性

ノードv_iの近接中心性$C_{closenness}(v_i)$は、ノードv_iから他のノードへの最短経路長の理論上最小となる総和を、実際の最短経路長の総和によって除算した値である。媒介中心性は、式 (3) のように定式化できる。

$$C_{closenness}(v_i) = \frac{n-1}{s(v_i)} \qquad (3)$$

ここで、v_iはノードv_iから自身を除くノードそれぞれへの最短経路長の総和を、nはネットワーク内のノード数を表す。

$C_{closenness}(v_i) = \frac{n-1}{s(v_i)}$は最小 0 から最大 1 までの値を取り、値が 1 に近いほど近接中心性が高い。近接中心性が高いノードほど、他のノードに近い距離で到達可能なノードである。v_iと他のノードv_jが到達不可能な場合、本論文では松尾ら[24]と同様、v_iとv_jの距離をnとする。

OSS コミュニティにおいて近接中心性が高い参加者ほど、界隈に参加者が多い（少ない距離で他の参加者へ到達できる）参加者である。つまり、次数中心性が高い参加者が、(OSS コミュニティ全体ではなく）自身と隣接する参加者を対象とする一方で、近接中心性が高い参加者は、OSS コミュニティ全体に対して効率よく情報を伝達しているとみなすことができる。

３．５ 分析手順

ソーシャルネットワーク分析を用いた各 OSS コミュニティの分析手順を以下に示す。

手順1：オンライン上に公開されている開発者ML及びユーザMLのデータを取得する。
手順2：各MLから開発者ネットワーク及びユーザネットワークを形成する。
手順4：3つのネットワーク（開発者、ユーザ、コーディネータ）それぞれの中心性を計測する。
手順5：手順1～4を全てのコミュニティ（Apache、GIMP、Netscape）に適用し比較・分析する。

4．ケーススタディ

4．1 対象OSSコミュニティ

ケーススタディでは3つのOSSコミュニティを対象とした。これらの対象プロジェクトの選定条件は以下の通りである。

条件1：開発者MLとユーザMLが個別に設けられている
条件2：数多くの開発者とユーザが存在する
条件3：総投稿メッセージが5000以上である
条件4：存続期間が3年以上である
条件5：OSSの中でも特に有名である

条件1は本論文でのキーアイデアとなるコーディネータを分析するために必須であり、分析対象プロジェクトが最低限満たすべき前提条件である。条件2～4は、分析対象コミュニティの規模を平滑化するための条件である。これらの条件1～4を設定することにより、OSSコミュニティの成長過程においてコーディネータを中心とする開発者とユーザのコラボレーションの違いが、以降のコミュニティの成長にどのように影響するかを理解する。条件5は他のOSS開発に対する有用な知識の発見のために設定したものである。上記の選定条件を満たすプロジェクトとして、本論文ではApache、GIMP、Netscapeの3つを選択した。各コミュニティの統計量を表1に示す。また、各コミュニティの開発しているソフトウェアは以下の通りである。

・ Apache HTTP Server[2]

現在、世界一のシェアを誇るWebサーバ（HTTP Server）ソフトウェアである。パフォーマンスの高さや拡張性の高さ、使用が無制限であることなどから広く利用されている。また欠陥の修正パッチを継続的に多数リリースしているため、非常に高品質なソフトウェアとして広く認知されている。

・ GIMP[3]

GIMPとはGNU Image Manipulation Programの略称であり、画像の編集や加工を行うためのソフトウェアである。無料でありながら、有料かつ高額な画像編集ソフトウェアと同等の機能を有しており、多数のユーザから支持されている。

・ Netscape Browser

Webブラウザソフトウェアであり、1996年ごろの最盛期には市場の8割を占めていた。Netscapeコミュニティは、Internet Explorerのシェア拡大に対抗するためRaymondらの研究[18]を受けて1998年にオープンソース化を実施したが、シェアを取り返すことができず衰退していった。現在はMozillaコミュニティがNetscapeの資産を引き継ぎ、Firefoxなどのソフトウェアを開発している[13]。

本論文では各コミュニティの開発したソフトウェアの実績（ダウンロード数やリリース数など）に基づき、ApacheとGIMPを成功コミュニティ、Netscapeを衰退コミュニティとみなす。

4．2 分析対象データ

分析を行うにあたっては、各OSSコミュニティの用意しているコミュニケーションメディアの履歴を元に、コミュニケーションネットワークの構築を行った。

---

[2] http://httpd.apache.org/
[3] http://www.gimp.org/

対象データはApache とGIMP はML、Netscape はニュースグループであり、それぞれ各OSS コミュニティがコミュニケーション方法として利用しているメディアである。分析期間に関しては、履歴が存在する期間の全てを用いたが、GIMP に関しては2003 年9 月以降の24 ヶ月にかけて履歴が存在しないため、2003 年9 月までの履歴を対象とした。また、期間P の間に2 つのコミュニケーションメディア（開発者用とユーザ用）に一度でもメッセージを送信した人物をコーディネータとみなした。

表-1 OSS コミュニティの統計量

	分析対象期間			総リリース数		総参加人数		総メッセージ数	
	開始年月	終了年月	期間(月)	メジャー	マイナー	開発者	ユーザ	開発者	ユーザ
Apache	2001/11	2006/09	59	1	22	1,619	9,818	32,985	68,495
GIMP	1999/10	2003/09	50	2	46	1,120	1,519	15,846	6,638
Netscape	1999/09	2007/02	92	3	11	8,161	8,002	29,417	61,946

４．３ 分析結果

各コミュニティの分析対象期間における統計量（参加人数とメッセージ数）の推移と、3 つのネットワーク中心性（次数中心性、媒介中心性、近接中心性）の平均値の推移を図4 に示す。グラフの横軸は時間を、縦軸は各期間における指標の値であり、縦の破線はメジャーバージョンのリリースが行われた時期を表す。

各中心性の平均値を扱う理由としては、参加者の個々の指標を扱うことは数百、数千ノードというコミュニティの規模の大きさから困難であり、コミュニケーションネットワーク全体の形を捉えるために不適切であるためである。また、エッジを持たないノード（一度しか発言しなかった参加者）が多数存在しており、平均値が0 に近い値を取りネットワークごとの比較が困難になることから、ネットワーク構造の特徴を浮き立たせるために各指標の値の高いものから上位10%の平均値を扱った。以降、2 章で述べた分析の観点に従って分析結果を説明する。なお、文中で用いる（A-1、G-1）などは図4 の各グラフのラベルを指す。

観点1．時間の経過に伴うコミュニケーションネットワークの変化。

まず、分析期間全体についてのネットワーク構造の変化について着目する。Apache とGIMP のコミュニティは参加者（A-1、G-1）が減少傾向にあるが各ネットワークの中心性（A-3,4,5、G-3,4,5）は、分析期間の後半にかけて著しく向上しており、コミュニケーションネットワークが小さく密にまとまっていることが読み取れる。これは時間の経過に伴って、モチベーションの高い参加者同士が密に連絡を取り合うことで活発にコラボレーションを進められていると考えられる。一方でNetscape の参加者数（N-1）では分析期間初期から単調に参加人数が減少傾向にある。特にVer.8 リリース直後にかけて媒介中心性(N-4)、近接中心性(N-5)が大きく下がっており、コミュニケーションネットワークの急激な過疎化が読み取れる。

次に開発者ネットワークとユーザネットワークの違いに着目する。参加者同士の緊密さを現す近接中心性はApache とGIMP ではほぼ全期間に渡り開発者がユーザより高い値を取っている一方で、Netscape ではユーザのほうが高い値をとる傾向にある。さらにNetscape ではユーザが開発者の倍以上参加していることからNetscape コミュニティはユーザの盛り上がりに対して開発者のモチベーションが低く、ユーザからの意見や評価が成果物に反映しきれない状態であると考えられる。

リリース前後でのネットワーク構造の違いについて着目する。いずれのOSS コミュニティでもメジャーバージョンのリリースされた時期に共通する変化がない。例えば、GIMP では2 つのメジャーリリースが行われているが、Ver.1.2.0 のリリース前後では開発者、ユーザにかかわらず、参加者数（G-1）が多く近接中心性（G-5）の低い疎なコミュニケーションネットワークを築いているが、Ver.1.3.0 のリリース前後では逆に小さく密なネットワークを築いている。表1 に示す通り、1 月あたりのマイナーバージョンを含むリリース数は、Apache は平均0.39 件、GIMP は平均0.96 件と3 ヶ月に1 度はリリースを行っている。早めで頻繁なリリースを行い、ユーザからの意見やバグ報告を早めに取り込むことで成果物の品質を向上しようとするバザール方式の考えに習ったものといえる。細かいリリースを頻繁に行うため、メジャーバージョンがリリースされてもネットワーク構造の大きな変化が生じなかったと考えられる。

観点 2. コーディネータのコミュニケーション活動の変化

各 OSS コミュニティのコーディネータの数（A-1、G-1、N-1）は 20～40 人程度と大きな違いはなく、Netscape のような衰退コミュニティであっても数十人のコーディネータが存在している。各 OSS コミュニティのコーディネータの媒介中心性（A-3、G-3、N-3）に着目しても、値の差はほとんどない。原因としては、媒介中心性が表す値が参加者がいずれのコミュニティ(開発者ユーザ)に属しているかを考慮に入れないためであると考えられる。

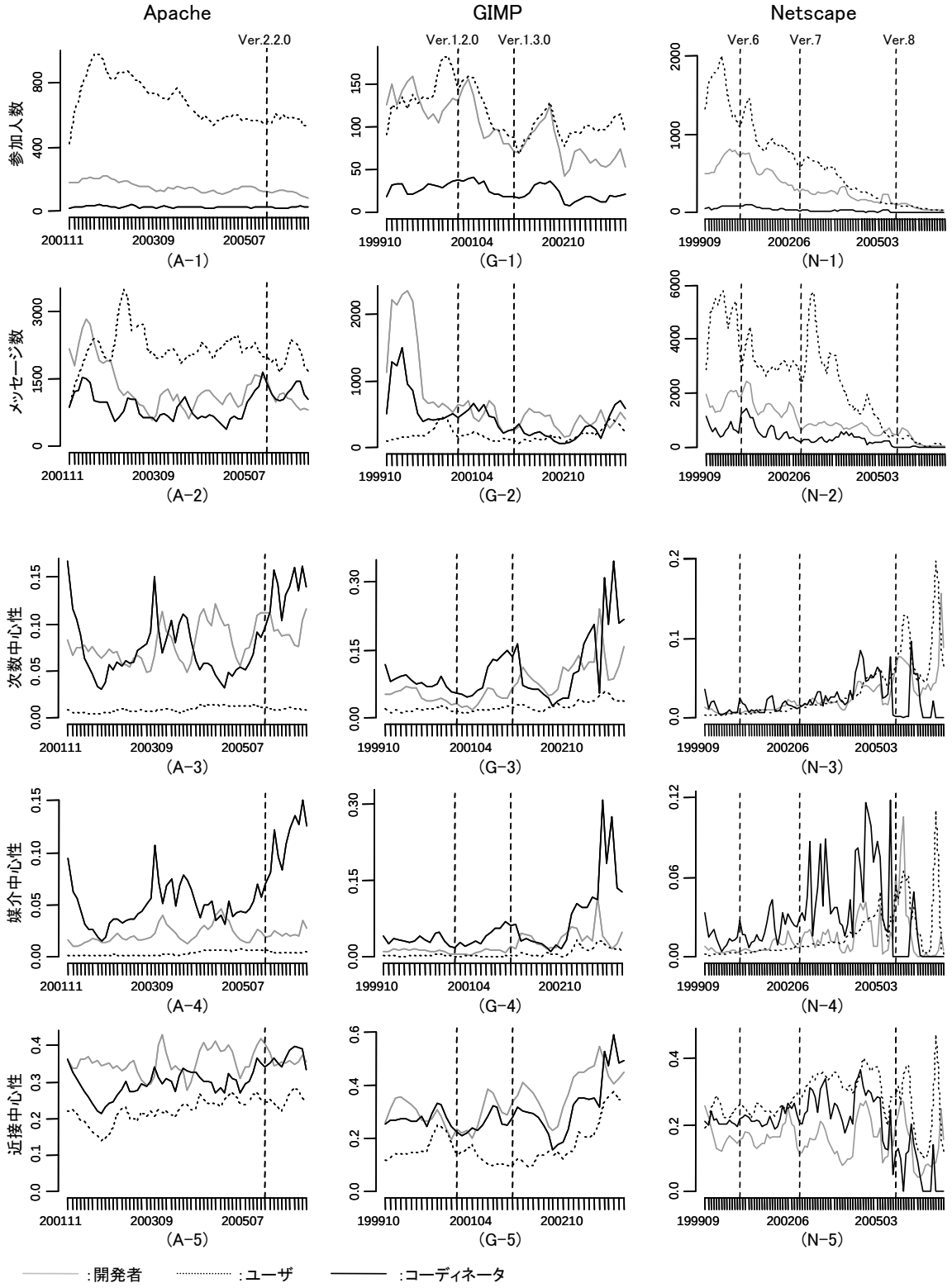


図-4 各 OSS コミュニティの統計量と中心性の推移

表-2 上位コーディネータの次数

		1	2	3	4	5
Apache	開発者との次数	16	30	34	19	28
	ユーザとの次数	89	33	29	21	1
GIMP	開発者との次数	25	5	10	5	11
	ユーザとの次数	9	17	0	3[4]	0
Netscape	開発者との次数	1	0	26	19	20
	ユーザとの次数	375	29	0	1	1

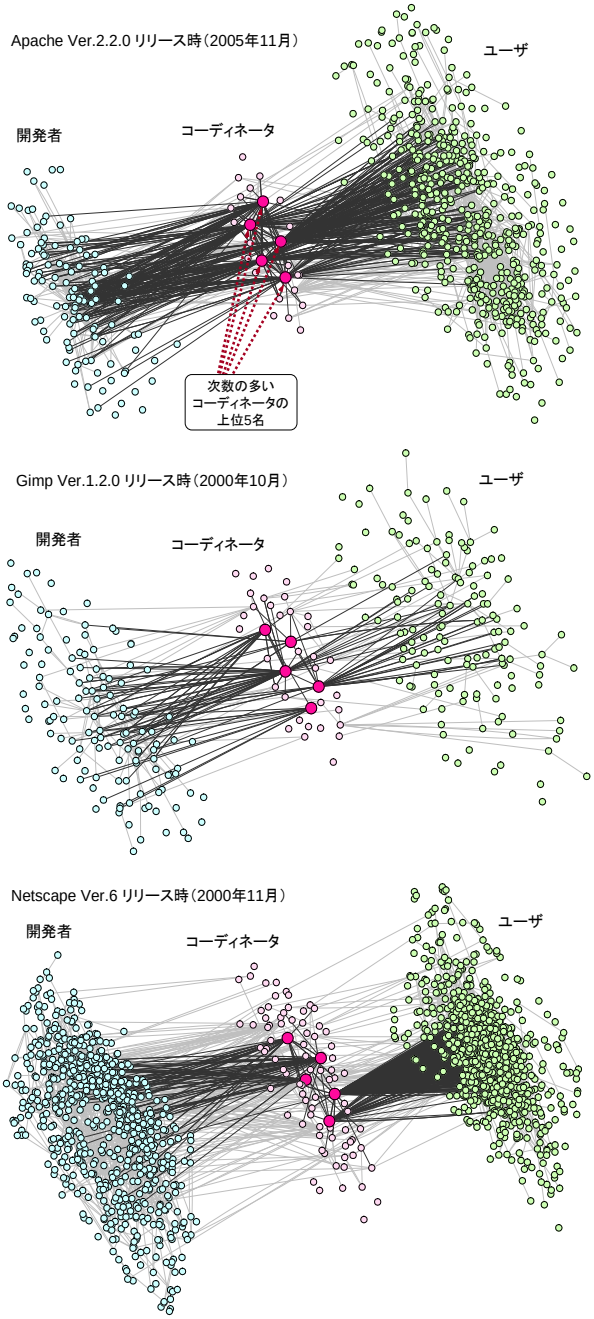


図-5 ある一期間におけるコミュニケーションネットワーク

[4]次数 0 とは発言を行ったが、返信メッセージが 1 つもなかったことを表す。

本論文で示すコーディネータとは、開発者と開発者またはユーザとユーザを媒介する人物ではなく、開発者とユーザを媒介する人物のことであり、媒介中心性では表現することができない。そこで、ある一期間におけるコミュニケーションネットワークに着目し、実際にコーディネータがどのようなコミュニケーションを行っていたかについて詳細に分析を行う。

図5はメジャーバージョンのリリースが行われた一期間におけるコミュニケーションネットワークを可視化したものである。対象時期はそれぞれ、Apache は Ver.2.2.0、GIMP は Ver.1.2.0、Netscape は Ver.6 とした。図の左に位置する集団は開発者コミュニティ、右はユーザコミュニティ、中央はコーディネータを表しており、次数の多い上位5名のコーディネータ（上位コーディネータ）のノードとエッジを強調している。表2に図5の上位コーディネータの開発者コミュニティとの次数と、ユーザコミュニティとの次数を示す。

図5と表2より、各 OSS コミュニティのコーディネータの持つ開発者コミュニティとユーザコミュニティへの次数のバランスの違いが確認できる。Netscape の上位コーディネータは、エッジの少ないほうのコミュニティへの次数が0または1であり、いずれのコーディネータもコミュニティの媒介者としての役割を全く果たせていない。それに対し Apache、GIMP の上位コーディネータはバランスよく2つのコミュニティとコミュニケーションを取っており、2つのコミュニティを媒介するコーディネータとしては理想的な働きをしている。

また、観点2での知見の1つである Netscape コミュニティの開発者とユーザの盛り上がりの差についても、コーディネータのバランスという観点から説明することができる。つまり、Netscape の開発者のモチベーションの低下は、ユーザの盛り上がりを開発者へ伝えるコーディネータの欠如によるものであり、このことからも OSS コミュニティのイノベーション創出のためにはコーディネータの存在が重要であるといえる。

５．おわりに

本論文では、OSS コミュニティのコラボレーションを明らかにするために、2つの参加者コミュニティが形成するコミュニケーションネットワークに対して分析を行った。3つの著名な OSS コミュニティケーススタディにより得られた知見は以下の通りである。

- 成功コミュニティでは時間の経過に伴いネットワーク構造が小さくまとまる傾向にある。
- 開発者コミュニティとユーザコミュニティの活発さの差はコミュニティの衰退を招く。
- OSS では高い頻度でリリースを行うため、リリース前後でのコミュニケーション構造の変化は小さい。
- OSS コミュニティが活発に活動するためにはコーディネータの存在が重要である。
- 成功コミュニティでは2つの参加者コミュニティをバランスよく媒介するコーディネータが重要な役割を占める。
- 媒介中心性は複数の種類のノードを持つネットワークに対しては適用できない場合がある。

本論文では各 OSS コミュニティのコミュニケーション媒体のみを対象として分析を行ったが、コミュニティの開発したソフトウェアの変化(例えばコミット回数やソースコードの行数など)については分析を行っていない。ソフトウェアのリリース時期のみならず、これらのメトリクスと対比して分析を行うことは今後の課題である。

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)、公益信託マイクロソフト知的財産研究助成基金および、特別研究員奨励費（課題番号：20009220）による助成を受けた。

［参考文献］

[1] Bird, C., Gourley, A., Devanbu, P., Gertz, M. and Swaminathan, A.: Mining Email Social Networks, In Proceedings of the 2006 International Workshop on Mining Software Repositories, pp.137-143 (2006).

[2] David, P.A., Waterman, A. and Arora, S.: FLOSS-US: The Free/Libre/Open Source Software Survey for 2003, available from <http://www.stanford.edu/group/floss-us/> (accessed 2007-06-30).

[3] Feller, J. and Fitzgerald, B.：Understanding Open Source Software Development, Addison-Wesley (2002).
[4] Freeman, L.C.：Centrality in Social Networks: Conceptual Clarification, Social Networks, Vol.1, No.3, pp.215-239 (1979).
[5] Hossain, L., Wu, A. and Chung, K. K.S.：Actor Centrality Correlates to Project Based Coordination, In Proceedings of the 2006 Conference on Computer Supported Cooperative Work, pp.363-372 (2006).
[6] 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, pp.319-330 (2006).
[7] International Institute of Infonomics (University of Maastricht) and Berlecon Research GmbH: Free/Libre and Open Source Software: Survey and Study, available from <http://www.infonomics.nl/FLOSS/report/> (accessed 2007-06-30).
[8] Jensen, C. and Scacchi, W.：Role Migration and Advancement Processes in OSSD Projects: A Comparative Case Study, In Proceedings of the 29th International Conference on Software Engineering (ICSE'07), pp.364-374 (2007).
[9] Kakimoto, T., Kamei, Y., Ohira, M. and Matsumoto, K.：Social Network Analysis on Communications for Knowledge Collaboration in OSS Communities, In Proceedings of the International Workshop on Supporting Knowledge Collaboration in Software Development, pp.35-41 (2006).
[10] Lakhani, K.R. and von Hippel, E.：How Open Source Software Works: "Free" User-to-User Assistance, Research Policy, Vol.32, No.6, pp.923-943 (2003).
[11] Lave, J. and Wenger, E.: Situated Learning: Legitimate Peripheral Participation, Cambridge University Press (1991).
[12] Mockus, A., Fielding, R.T. and Herbsleb, J.D.：Two Case Studies of Open Source Software Development: Apache and Mozilla, ACM Transactions on Software Engineering and Methodology, Vol.11, No.3, pp.309-346 (2002).
[13] Mozilla Japan: Firefox と Mozilla 関連組織の歩み (概略), 入手先 <http://foxkeh.jp/downloads/history/history-foxkeh.pdf> (参照 2007-06-30).
[14] Nakakoji, K., Yamamoto, Y., Nishinaka, Y., Kishida, K. and Ye, Y.：Evolution Patterns of Open-Source Software Systems and Communities, In Proceedings of the International Workshop on Principles of Software Evolution, pp.76-85 (2002).
[15] Netcraft Ltd.: Netcraft Web Server Survey, available from <http://www.netcraft.com/Survey/> (accessed 2008-08-29).
[16] Ohira, M., Ohoka, T., Kakimoto, T., Ohsugi, N. and Matsumoto, K.：Supporting Knowledge Collaboration Using Social Networks in A Large-Scale Online Community of Software Development Projects, In Proceedings of the International Workshop on Supporting Knowledge Collaboration in Software Development, pp.835-840 (2005).
[17] Ohira, M., Ohsugi, N., Ohoka, T. and Matsumoto, K.: Accelerating Cross-Project Knowledge Collaboration Using Collaborative Filtering and Social Networks, In Proceedings of the 2005 International Workshop on Mining Software Repositories, pp.111-115 (2005).
[18] Raymond, E.S.: The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary, O'Reilly and Associates (1999).
[19] Scott, J.: Social Network Analysis: A Handbook, SAGE Publications (2000).
[20] Wasserman, S. and Faust, K.：Social Network Analysis: Methods and Applications, Cambridge University Press (1994).
[21] Yamauchi, Y., Yokozawa, M., Shinohara, T. and Ishida, T.: Collaboration with Lean Media: How Open-Source Software Succeeds, In Proceedings of the 2000 Conference on Computer Supported Cooperative Work, pp.329-338 (2000).
[22] Ye, Y. and Kishida, K.：Toward an Understanding of the Motivation Open Source Software Developers, In Proceedings of the 25th International Conference on Software Engineering, pp.419-429 (2003).
[23] 株式会社三菱総合研究所：FLOSS-JP オープンソース/フリーソフトウェア開発者 オンライン調査日本版, 入手先 <http://oss.mri.co.jp/floss-jp/> (参照 2007-06-30).

[24] 松尾 豊，篠田考祐，中島秀之：中心性に着目した合理エージェントのネットワーク形成，人工知能学会論文誌， Vol. 21，No. 1，pp. 122-132（2006）.