

OSS コミュニティにおける共同作業プロセス理解のための中心性分析

大平 雅雄[†] 梶本 真佑[†] 前島 弘敬[†]
 亀井 靖高[†] 松本 健一[†]

本研究の目的は、コーディネータのコミュニケーション構造が OSS コミュニティにおける共同作業プロセスへ与える影響を明らかにすることである。本稿では、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を分析しその特徴を観察する。OSS コミュニティのコミュニケーション構造に対して中心性分析を行った結果、(1) 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こと、(2) 成功コミュニティでは 2 つのサブ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存在すること、(3) 成功コミュニティでは時間の経過に伴いコミュニケーション構造が過密化することが知見として得られた。

Centrality Analysis for Understanding the Collaboration Process in OSS Community

MASAO OHIRA,[†] SHINSUKE MATSUMOTO,[†] HIROTAKA MAESHIMA,[†]
YASUTAKA KAMEI[†] and KEN-ICHI MATSUMOTO[†]

The goal of our research is to construct deeper understandings of the collaboration process in OSS community. In this paper, we observed three famous OSS communities and analyzed how coordinators have effects on the collaboration process. As the results of analysis, we have found the followings: (1) coordinators play an important role for activities of OSS communities; (2) coordinators who are intermediate between sub-communities exist in a successful community; and (3) the communication structure in a successful community is concentrated over time.

1. はじめに

品質や機能などの面で商用ソフトウェアに引けを取らないソフトウェアを無償で利用できるという理由などから、近年、行政機関や教育機関において Open Source Software (OSS) の導入が進みつつある。一般的な OSS 開発は、開発者やユーザからなる参加者が WWW 上に形成した OSS コミュニティにおいて、ルールや指揮系統の少ない個人間のやり取りを通じて自由に行われる⁶⁾。コミュニティ主導型の OSS 開発は何らかの原因により停滞や中止を余儀なくされる場合も多く、エンドユーザが継続的なサポート（バグの修正や機能追加など）を得られなくなることがある。そのため、商用ソフトウェアの代替肢として企業や組織で大規模に OSS を導入しようとする際には極めて慎重な判断が求められる。社会基盤として OSS が普及しつつある現在、OSS コミュニティの継続的な活動に対する見通しを得るための方法を構築することが強く求められている。

このような背景から、OSS コミュニティにおける開発プロセスの実態や 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^{4),5),8)}。厳格なルールや指揮系統を持たないコミュニティ主導型の OSS 開発であっても、ある程度の秩序と規律を保ちながら共同作業を円滑に

進めるためには、開発者とユーザとの間の橋渡しの役割を担うコーディネータの存在が重要であると指摘されている^{3),6)}。しかしながら、コーディネータが開発者とユーザの間で実際にどのようなコミュニケーション構造を築いてコミュニティの共同作業の調整を行っているかは現在のところ明らかにされていない。

本研究の目的は、コーディネータのコミュニケーション構造が共同作業プロセスへ与える影響を明らかにすることである。そのために本稿では、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。以降、2 章で OSS コミュニティにおける共同作業プロセスについての仮説を提示し、3 章で仮説を確かめるための分析手法について述べる。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し、5 章で分析結果についての議論を行う。6 章で本稿をまとめ、今後の課題について述べる。

2. OSS コミュニティにおける共同作業のプロセス

2.1 OSS コミュニティによるソフトウェア開発

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を指す。OSS 開発コミュニティによるソフトウェア開発は、ソフトウェア開発企業の一般的な開発環境と比べて以下のような特徴がある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

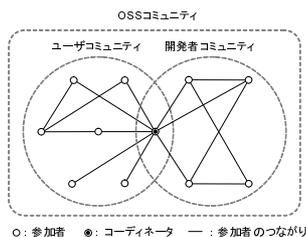


図 1 OSS コミュニティの組織構成
Fig. 1 Organizational structure of an OSS community.

- (1) 開発者が自由に参加/脱退することが可能
- (2) ボランティアでの参加が基本
- (3) 厳格な指揮系統が存在しない
- (4) ユーザ参加型コミュニティ
- (5) ネットワークを介した分散開発環境

これら OSS コミュニティに特徴的な開発環境を考慮に入れた上で、コミュニティが活発に活動するためのコミュニケーション構造についての仮説を提示する。

仮説 1. 成功コミュニティでは、時間の経過に伴いコミュニケーション構造が過密化する。

開発者が自由に参加できるという OSS の特徴 (1) から、成果物が社会的に注目を浴びる時期には参加者が増加する。しかし自由に脱退することも可能であるため、興味本位といったモチベーションの低い参加者が時間の経過とともに脱退する。その結果、モチベーションの高い参加者のみが残りコミュニケーション構造が密になると考えられる。

仮説 2. 開発者同士が十分にコミュニケーションを取っていると、OSS コミュニティは成功する。

OSS の特徴 (2) の通り、開発者のモチベーションを維持するためには、開発者同士の密なコミュニケーションが重要である。さらに特徴 (3) の通り、OSS コミュニティの取るバザール方式では厳格な指揮系統が存在しない。このため、個々の開発者が完全に自由に開発を行うのではなく、開発者同士のコミュニケーションを通じてそれぞれの進捗状況や意思の統一を図る必要があると考えられる。

仮説 3. 2つのコミュニティを媒介するコーディネータが存在すると、OSS コミュニティは成功する。

OSS コミュニティが活発に活動を続けるためには、開発者コミュニティとユーザーコミュニティの共同活動を支えるために、知識や意見の媒介を行う人物 (コーディネータ) の存在が重要であると考えられる。次節では、コーディネータについて述べる。

2.2 OSS コミュニティにおけるコーディネータの役割

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^{3),6)}。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) が OSS 参加者のモチベーションの維持に繋がるとしている⁸⁾。

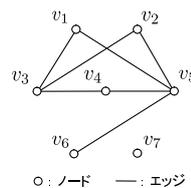


図 2 ネットワークの例
Fig. 2 An example of network.

本稿では、OSS コミュニティの共同作業の円滑化を促進する役割として、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) の存在の重要性を仮説 3 として提示する。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。

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意見や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議論や意見交換が重要であり、それらを媒介するコーディネータが共同作業の円滑化を促す重要な役割を果たしていると考えられる。

3. 分析手法

2.1 節の仮説を確かめることを目的とし、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ワーク分析⁷⁾を用いる。本稿では、ソーシャルネットワーク分析が提供する指標のうち、Freeman の提案する 3 つの中心性²⁾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構造の特徴を把握する。

以下では、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説明する。

3.1 ネットワークの定義

本稿では、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 への経路は、 $v_1 \ v_5 \ v_6$ や $v_1 \ v_3 \ v_4 \ v_5 \ v_6$ などがある。
- 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 $v_1 \ v_5 \ v_6$ の距離は 2 であり、経路 $v_1 \ v_3 \ v_4 \ v_5 \ v_6$ の距離は 4 である。また、 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 $v_1 \ v_5 \ v_6$ で、最短経路長は 2 である。

3.2 OSS コミュニティにおけるコミュニケーション構造

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構造の形成に大きな影響を与える。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構造は、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。

本稿では、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構造を定義する。

3.3 ネットワークの中心性

以下では、Freeman の提案する 3 つの中心性（次数中心性、媒介中心性、近接中心性²⁾）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。

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

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

$C_{degree}(v_i)$ は最小 0 から最大 1 までの値を取る。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きノードである。OSS コミュニティにおいて次数中心性が高い参加者は、多くの参加者へ（から）情報を発信（受信）する人物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り、式 (2) のように定式化できる。

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

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、 $\sum_{j < k} \sum_{j < k} p_{jk}$ は v_j から v_k への最短経路の総数を表し、 $\sum_{j < k} \sum_{j < k} p_{jk}(v_i)$ 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 までの値を取る。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 $C_{closeness}(v_i)$ 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。

$$C_{closeness}(v_i) = \frac{n-1}{s(v_i)} \quad (3)$$

ここで、 $s(v_i)$ 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。

$C_c(v_i)$ は最小 0 から最大 1 までの値を取る。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、 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、界限に参加者が多い（少ない距離で他の参加者へ到達できる）参加者である。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のに対して、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。

3.4 時系列分析

コミュニケーション構造は時間の経過とともに変化するため、すべての期間のデータを用いて各中心性を求めることは不適切である。そのため、ある一定期間ごとにコミュニケーション構造を求め、各指標を算出する必要がある。

本稿では、ある一定期間 P ごとにコミュニケーション構造を求める際に、コミュニケーション構造を求める期間 P の開始時期を一定期間（例えば、 $P/2$ ）ずつスライドさせる Sliding Time Method を用いる。

本稿では、OSS コミュニティにおける開発者の社会的構造を分析した研究³⁾と同様、コミュニケーション構造を求める期間 P を 3ヶ月、スライドさせる期間を 1ヶ月とした。

4. ケーススタディ

本章では、2章で提示した 3 つの仮説の確認を目的として行ったケーススタディについて述べる。

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。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 : 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。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 : 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 : 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⁶⁾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。

本稿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構造の構築を行った。

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。分析対象のメディアの種類に違いはあるが、これらのメディアが各 OSS コミュニティ内でのコミュニケーションの基本媒体であったことや、本研究ではメッセージの送受信関係からコミュニケーション構造を構築していることから、メディアの違いによる分析結果への影響は小さいものと考えている。

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。

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 3 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人単位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構造全体の形を捉えるために不適切であるためである。さらに、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 章で提示した仮説の確認を行う。

仮説 1. 成功コミュニティでは、時間の経過に伴いコミュニケーション構造が過密化する。

予測: 成功コミュニティでは時間の経過に伴って、モチベーションの低い参加者が脱退しモチベーションの高い参加者が残ることで、参加者数は減少するが 3 つの中心性が上昇する。

結果: Apache と GIMP のコミュニティは参加者 (A-1、

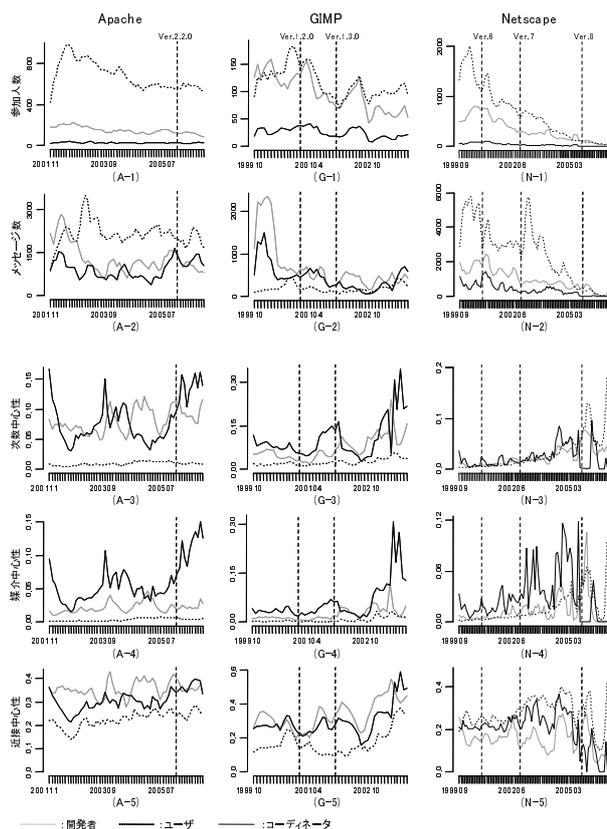


図 3 各 OSS コミュニティの統計量と中心性の推移

Fig. 3 Changes of statistic values and centralities of each OSS community.

G-1) が減少傾向にある。しかし各ネットワークの中心性 (A-3,4,5, G-3,4,5) は、若干の波はあるものの全体的に右肩上がりの傾向を示している。特に、どちらの OSS コミュニティも分析期間の後半にかけて全ての中心性が著しく向上しており、コミュニケーション構造が小さくまとまっている。Netscape の参加者数 (N-1) も同様に減少傾向にあるが、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構造の急激な過疎化が読み取れる。これらのことから、仮説の正しさを確認できる。

仮説 2. 開発者同士が十分にコミュニケーションを取っていると、OSS コミュニティは成功する。

予測: 成功コミュニティと衰退コミュニティを比較すると、成功コミュニティの開発者コミュニティは活発に活動を行い相互にコミュニケーションを取るため、参加者同士の近さや親密さを表す近接中心性が高い。

結果: Apache と GIMP の開発者コミュニティは近接中心性 (A-5, G-5) が高いが、Netscape の近接中心性 (N-5) は低く開発者間でのコミュニケーションが不足していると推測できる。さらに、Netscape のユーザコミュニティは多くの参加者数 (N-1) を抱えながらも近接中心性 (N-5) が高いことから、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

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)

表 1 各 OSS コミュニティの統計量
Table 1 Statistic values of each OSS community.

	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

価が成果物に反映しきれない状態であるといえる。これらのことから、この仮説の正しさを確かめることができる。

リリース前には開発者コミュニティが活発に活動を行うため、参加者同士の親密さを表す近接中心性が向上する。リリース後にはユーザコミュニティの近接中心性が向上する。

仮説 3. 2つのコミュニティを媒介するコーディネータが存在すると、OSS コミュニティは成功する。

予測：成功コミュニティのコーディネータは、衰退コミュニティに比べて数が多く、同時にコミュニティ内の媒介能力を表す媒介中心性が高い。

結果：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、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。Netscape は他の OSS コミュニティより多くの参加者 (N-1) を抱えており、総参加者数に対してコーディネータの数の不足ということも考えられるが、この仮説を十分に支持できる結果であるとはいいい切れない。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。ただし、Netscape は Ver.8 リリース時期にコーディネータの数が 0、つまり 2つの参加者コミュニティが分断された状態であり、コミュニティの崩壊を表しているともみなすことも可能であるが、中心性に基づいた分析だけではこの仮説を十分に確かめることはできない。

5. 議論

本章では、4章で十分に確認することができなかった仮説 3 についての再検討と、本稿で用いた分析方法の妥当性について議論を行う。

5.1 コーディネータの媒介能力の重要性

仮説 3 では、コミュニティの成功/衰退にかかわらずコーディネータの媒介中心性にほとんど差異はなかった。その原因としては、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。本稿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、媒介中心性ではこの違いを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構造に着目し、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 4 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構造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

表 2 上位コーディネータの次数
Table 2 Degree of top 5 coordinators.

コミュニティ		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	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

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 4 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 4 と表 2 より、各 OSS コミュニティごと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。これは、仮説 1 の正しさを支持する根拠であるといえる。

また、仮説 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 OSS コミュニティの共同作業の円滑化にはコーディネータの存在が重要であるといえる。

5.2 中心性分析の妥当性

中心性に基づいた分析の結果、Raymond らにより指摘されている以下の点⁶⁾ について確かめることができた。

- 多少不完全でも頻繁にリリースを行い、ユーザの意見を多く取り込むことがバザール方式の重要な要素の 1つである。
- 一部の開発者によってある程度動作するものができ上がったから、バザール方式により開発者コミュニティが活発になる。

前者は仮説 3 から導かれるものであり、後者は GIMP の開発者コミュニティに着目することで以下のように確認できる。GIMP の開発者コミュニティでは、開発初期には極端に多くのメッセージ (G-2) がやり取りされているにも関わらず、

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。

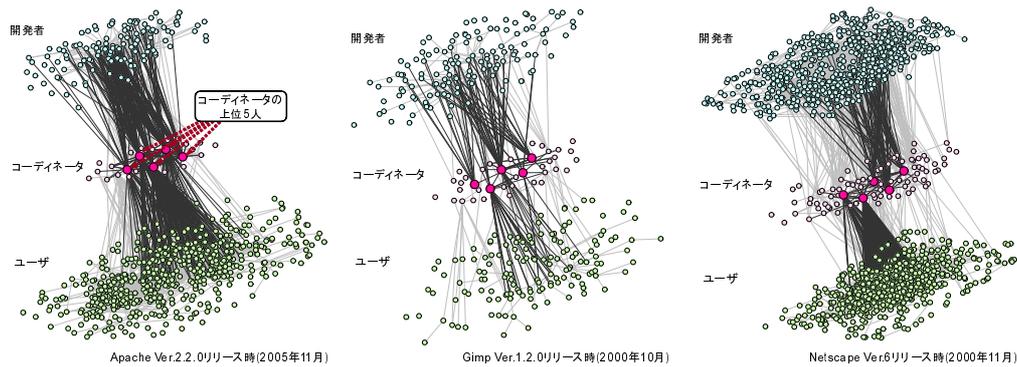


図 4 ある一期間におけるコミュニケーション構造
 Fig. 4 Communication network of during one period of time.

開発者の数 (G-1) と各ネットワーク中心性 (G-3,4,5) に大きな変化がない。これは、ネットワーク構造の変化は小さいものの、初期のネットワークのみエッジの重みが極端に高い状態であることを表している。つまり、開発初期に一部の開発者同士で大量のメッセージのやり取りが行われており、開発の負担が偏っている状態の表れである。開発中期以降ではメッセージ数は減少するものの各中心性が増加しており、複数の開発者同士がコミュニケーションを取ることで負荷分散やコミュニティの活性化が生じていると考えられる。

OSS の開発形態であるバザール方式についての従来研究の指摘を確認できたことは、OSS コミュニティの共同作業プロセスを明らかにするための手段としての中心性に基づく分析の妥当性を支持する 1 つの根拠といえる。

6. おわりに

本稿では、OSS コミュニティの共同作業のプロセス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構造に対して中心性分析を行った。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。

- 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存在する。
- 成功コミュニティは時間の経過に伴いコミュニケーション構造が過密化する。

また、従来の OSS についての研究により示唆されている以下の点についても、実際のコミュニケーション履歴により確認することができた。

- 多少不完全でも頻繁にリリースを行い、ユーザの意見を多く取り込むことがバザール方式の重要な要素の 1 つである。
- 最初からバザール方式で始めるのは不適切であり、一部の開発者によってある程度動くものができ上がってから開発者コミュニティが活発になる。

また、コーディネータの重要性を調べる方法としては、3 つの中心性の指標のみでは不十分であることが分かった。本

稿ではコーディネータの次数を確認することでその重要性を示したが、コーディネータの媒介能力を示すことのできるモデル及び指標の構築が今後の課題である。

謝辞 本研究の一部は、文部科学省「e-Society 基盤ソフトウェアの総合開発」の委託に基づいて行われたものである。また、本研究の一部は、文部科学省科学研究補助費（基盤研究 B：課題番号 17300007，若手 B：課題番号 17700111）による助成を受けた。

参考文献

- 1) Feller, J. and Fitzgerald, B.: *Understanding Open Source Software Development*, Addison-Wesley (2002).
- 2) Freeman, L.C.: Centrality in Social Networks: Conceptual Clarification, *Social Networks*, Vol. 1, No.3, pp.215-239 (1979).
- 3) 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).
- 4) 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).
- 5) 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).
- 6) Raymond, E.S.: *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*, O'Reilly and Associates (1999).
- 7) Wasserman, S. and Faust, K.: *Social Network Analysis: Methods and Applications*, Cambridge University Press (1994).
- 8) 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 (ICSE'03)*, pp.419-429 (2003).