

論文

**OSS コミュニティにおけるイノベーション
プロセスの理解：中心性分析の適用**

**Understanding Innovation Process in
OSS Communitites Using Centrality
Analysis**

杢本 真佑 (奈良先端科学技術大学院大学 情報科学研究科)
[学生会員#200604271]

亀井 靖高 (奈良先端科学技術大学院大学 情報科学研究科)
[学生会員#200327596]

前島 弘敬 (奈良先端科学技術大学院大学 情報科学研究科)
[学生会員#200602832]

大西 洋司 (奈良先端科学技術大学院大学 情報科学研究科)
[学生会員#200602361]

大平 雅雄 (奈良先端科学技術大学院大学 情報科学研究科)
[正会員#200403115]

松本 健一 (奈良先端科学技術大学院大学 情報科学研究科)
[正会員#198527448]

連絡先

杢本 真佑

〒 630-0192 奈良県生駒市高山町 8916-5

奈良先端科学技術大学院大学 情報科学研究科

TEL: (0743)72-5318 FAX: (0743)72-5319

email: shinsuke-m@is.naist.jp

概要

Linux や Apache に代表される Open Source Software (OSS) コミュニティは、商用ソフトウェアに引けを取らない品質と機能を備え、かつ、社会基盤としても広く利用されるソフトウェアを実現しているという点で、ソフトウェアの開発形態にイノベーションをもたらしたといえる。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 コミュニティのイノベーションプロセスが、OSS コミュニティ参加者のどのようなコミュニケーション構造によって支えられているのかについては明らかにされていない。本研究の目的は、OSS コミュニティのイノベーションのプロセスがどのように生じるのかを明らかにすることである。本論文では、3つの著名な 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中心性の観点から分析しその特徴を観察する。分析の結果、OSS コミュニティが活発に活動するためにはユーザと開発者をバランスよく仲介するコーディネータの存在が重要であること、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ことが明らかになった。

Abstract

Open Source Software (OSS) communities such as Linux and Apache community bring innovative insights into existing software development, in that they provide us high quality and functional software which is favorably comparable with proprietary software and widely used in our society. Although many studies have been conducted to explore the success key of OSS community, it is not clear that how communication structure of participants in OSS communities sustains the innovation process of OSS community. The goal of our study is to identify how the innovation process in OSS community arises. In this paper, we analyze communication structure of participants in three OSS communities using centrality analysis. As the results, we found that coordinators who mediate between users and developers play an important role in encouraging activities of OSS communities and the success community makes the communication structure dense over time.

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個人間のやり取りを通じて開発が行われる。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 の成功例であり、2007 年 6 月現在、Web サーバ市場のシェアの約 7 割と世界一のシェアを誇る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^{*}では、登録されている 10 万件以上のコミュニティのうち、開発者が 3 人以下の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、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な技術知識など）の確保が困難であり、コミュニティ内でのイノベーションの創出が難しい。一方、多くの参加者が存在するコミュニティであっても、参加者が持つ知識や意見が伝わりにくいコミュニケーション構造である場合はイノベーションの創出が困難である。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⁾の重要性が指摘されている^{18),22)}。

一般的な OSS 開発は、開発者が成果物であるソフトウェアをリリースし、ユーザが成果物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な開発プロセスがとられる。開発者とユーザのコミュニケーションを通じた共同作業による 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。本論文では、Apache や Linux のような成功した OSS コミュニティが用いる開発サイクルを、コミュニティ内におけるイノベーションのプロセスと考える。

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

^{*} <http://sourceforge.net/>

われているが^{8),12),14),22)}, OSS コミュニティの参加者が、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。本論文は、OSS コミュニティ内でどのようにイノベーションのプロセスが発生しているかを明らかにすることを目的とする。そのため、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イノベーション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する。5章で分析結果についての議論を行い、6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イノベーションのプロセス

2.1 OSS コミュニティ

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 イノベーションプロセスに関する仮説

OSS コミュニティの特殊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ネットワーク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ネットワークが過密化する。

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残りコミュニケーションネットワーク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

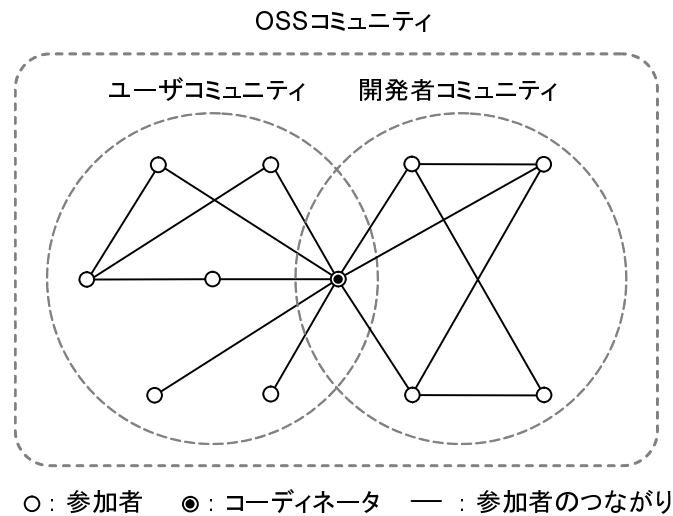


図1 OSSコミュニティの組織構成

Fig.1 Organizational structure of an OSS community.

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. 成功コミュニティでは、リリース前後で活発に活動するコミュニティが異なる。

OSSの特徴(4)の通り、OSSコミュニティ内では開発者コミュニティのみではなく、ユーザコミュニティの存在が重要である。図1にOSSコミュニティ内の2つの参加者コミュニティを示す。2つのコミュニティは相互のコミュニケーションによりモチベーションを保つが、活発に活動する時期は異なると考えられる。

特に、開発者が現在のバージョンに改良や修正を加えたバージョンを公開（リリース）する直前には、開発者コミュニティが活発に活動すると考えられる。また、リリース後にはバージョンに対する質問やバグ報告、機能追加要求などの意見が多数報告されるためユーザコミュニティが活発になると推測できる。

仮説4. 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するためには、開発者コミュニティとユーザコミュニティの協創活動を支え、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。コーディネータについての細説は次節

で行う。

2.3 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^{6),10),18)}。一方、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 コミュニティのイノベーションの創出を促す役割として、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4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、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

2.2 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^{19),20)}を用いる。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、Freeman の提案する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。

3.1 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。

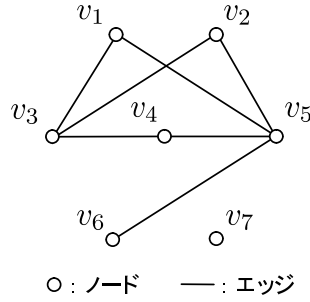


図2 ネットワークの例

Fig.2 An example of network.

- 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．

3.2 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。

3.3 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、Freeman の提案する 3 つの中心性を用いる⁴⁾。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。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）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ためである。

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。

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

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、 $\sum_{j < k}^n \sum_{j < k}^n p_{jk}$ 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 $C_{closeness}(v_i)$ 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。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。

3.4 時系列分析

コミュニケーションネットワーク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ネットワークを求め、各指標を算出する必要がある。

本論文では、ある一定期間 P ごとにコミュニケーションネットワークを求める際に、コミュニケーションネットワークを求める期間 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⁹⁾。ある一定期間におけるコミュニケーションネットワークの求め方を図 3 に示す。あ

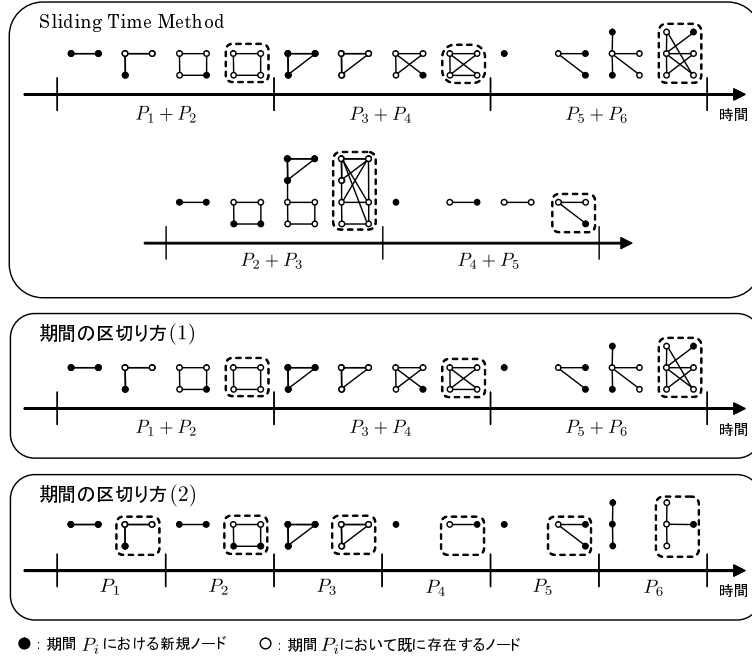


図3 ネットワーク構造の時系列分析方法⁹⁾

Fig. 3 Methods for time-series analysis of network structures.

る一定期間のコミュニケーションネットワークの各指標はその期間の終了時期のネットワークから算出される。

本論文で用いる Sliding Time Method は、分析の期間が重複しない図3の(1)や(2)のような期間の区切り方より、コミュニケーションネットワークの変化を捉えやすい。例えば、図3に示すように $P_2 + P_3$ の期間で、Sliding Time Method によるコミュニケーションネットワークでは、コミュニケーションの活発さが反映されているが、(1)、(2)ではともに反映できていない。

本論文では、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⁶⁾と同様、コミュニケーションネットワークを求める期間 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4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に市場の 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。

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。

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

[☆] <http://httpd.apache.org/>

^{☆☆} <http://www.gimp.org/>

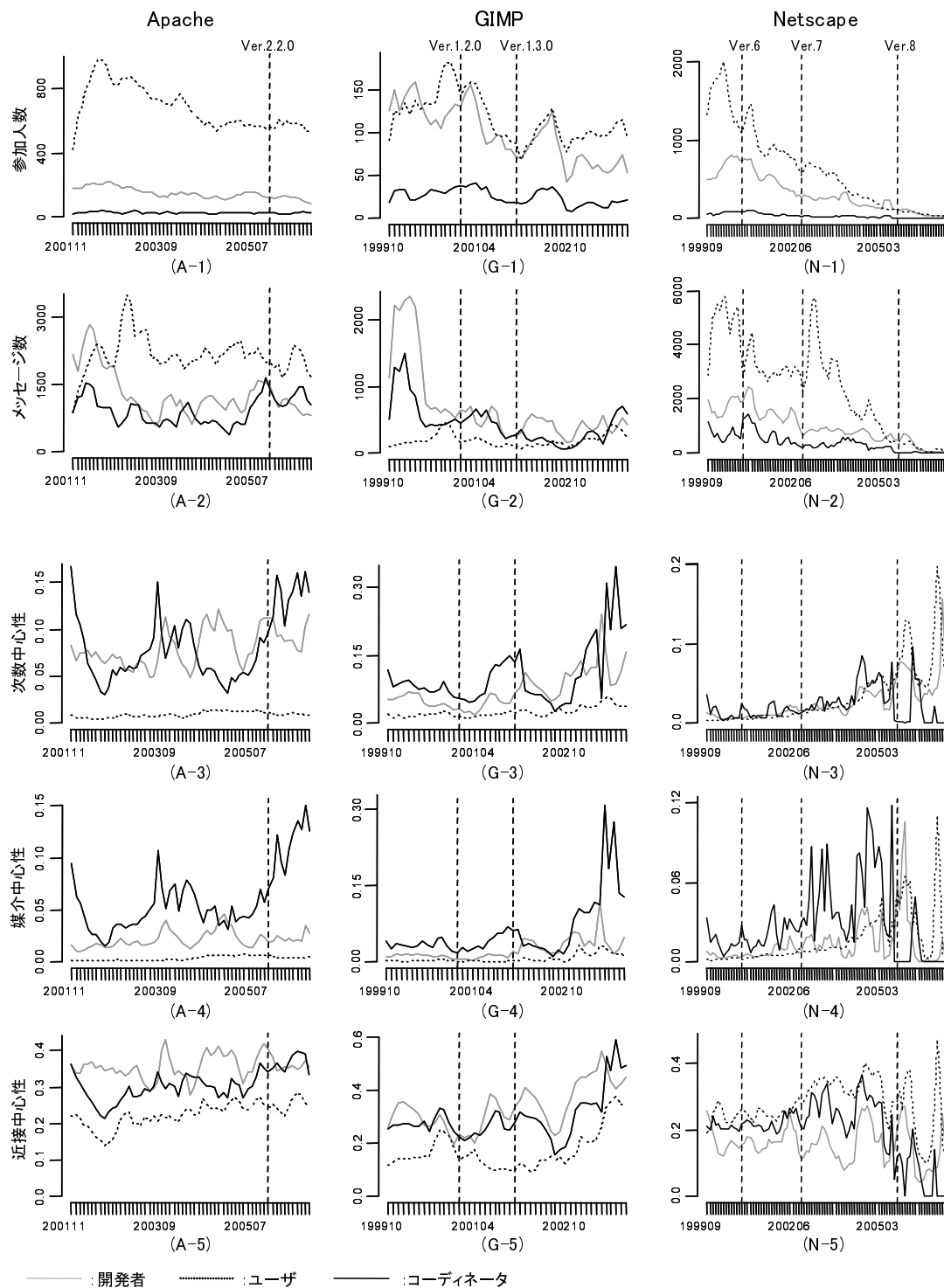


図4 各OSSコミュニティの統計量と中心性の推移

Fig. 4 Changes of statistic values and centralities of each OSS community.

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

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。ただし、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。なお、次項の文中で用いる（A-1、G-1）などは図4の各グラフのラベルである。

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ネットワークが過密化する。

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、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ネットワーク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ネットワーク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。

仮説 3. 成功コミュニティでは、リリース前後で活発に活動するコミュニティが異なる。

予測：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。

結果：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く、仮説を指示できる結果はない。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いと考えられる。この仮説の正しさ確認することはできない。

仮説 4. 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で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 4 についての再検討と、本論文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 4 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

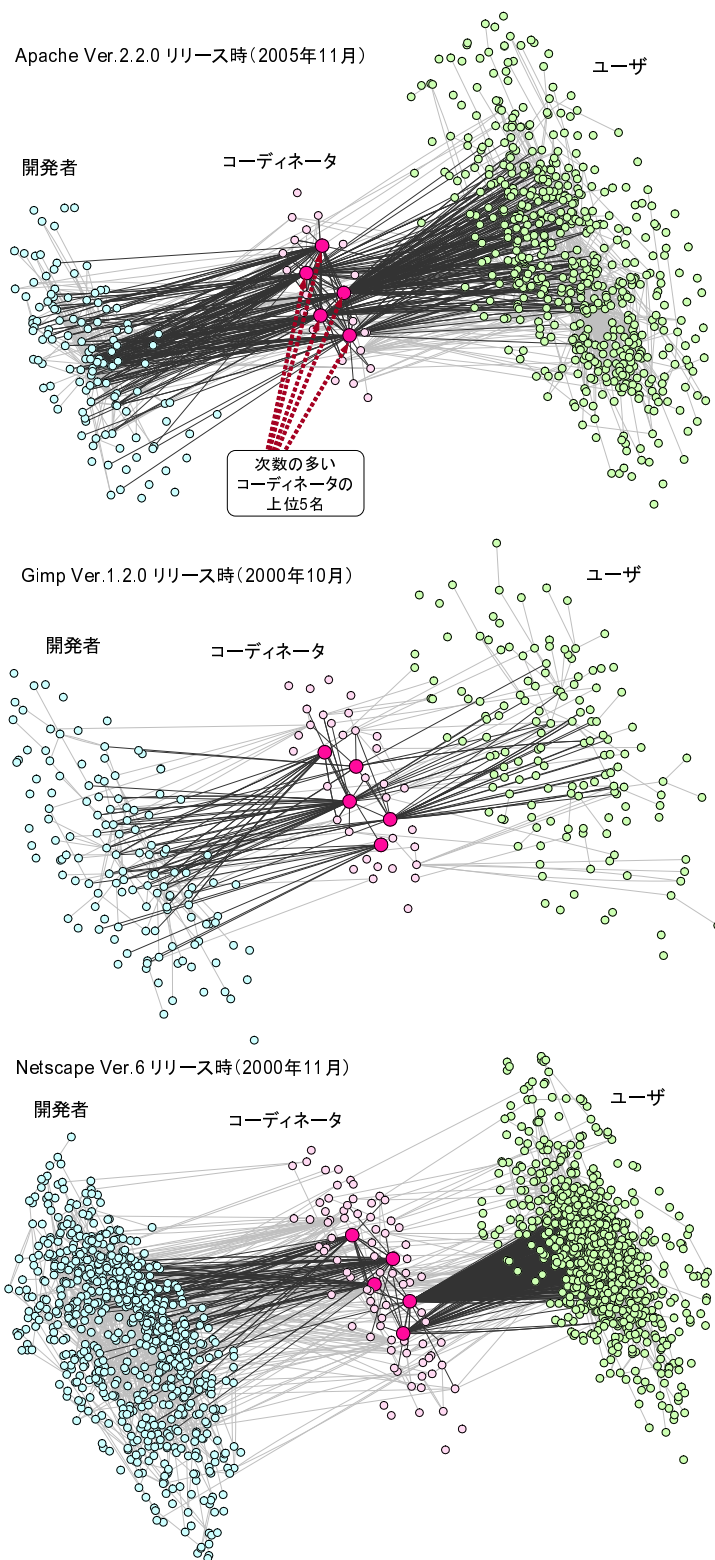


図5 ある一期間におけるコミュニケーションネットワーク

Fig. 5 Communication network of during one period of time.

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

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 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。

5.2 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¹⁸⁾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てから、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、開発者の数 (G-1) と各ネットワーク

[☆] 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。

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の分散やコミュニティの活性化が発生し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 コミュニティのイノベーションの創出過程を明らかにするための手段として、中心性に基づいた分析の妥当性の支持の 1 つといえる。

6. おわりに

本論文では、OSS コミュニティのイノベーション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ネットワーク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。本論文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。

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 (MSR'06)*, 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 (CSCW'06)*, 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 (OSS'06)*, 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 (KCSD'06)*, 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 (IWPSE’02)*, pp.76–85 (2002).
 - 15) Netcraft Ltd.: Netcraft Web Server Survey, available from <http://www.netcraft.com/Survey/> (accessed 2007-06-30).
 - 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 (KCSD’05)*, 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 (MSR’05)*, 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 (CSCW'00)*, 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 (ICSE'03)*, 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).

謝辞 本論文を執筆するにあたって有益なアドバイスを下さったソニー（株）の岡本圭司氏に深く感謝します。なお、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。