

# 複数のサブコミュニティを有する OSS コミュニティにおけるコーディネータの分析

伊原 彰 紀<sup>†1</sup> 前 島 弘 敬<sup>†1</sup> 松 本 真 佑<sup>†1</sup>  
亀 井 靖 高<sup>†1</sup> 大 平 雅 雄<sup>†1</sup> 松 本 健 一<sup>†1</sup>

Open Source Software (OSS) 開発は開発者やユーザなどの参加者同士の協調作業によって成り立つことから、開発者とユーザ間の情報伝達や調整行動を務めるコーディネータが重要な役割を果たしている。先行研究では、ユーザと開発者を橋渡しするコーディネータについて分析を行ったが、信頼性の高いソフトウェアを実現するために重要となるバグ報告者とコーディネータとの関係については明らかになっていない。本稿では、開発者、ユーザ、バグ報告者の3つのサブコミュニティを媒介するコーディネータについて分析を行う。分析の結果、PostgreSQL コミュニティでは、3つのコミュニティを媒介するコーディネータの人数がコミュニティの成長につれて増加する傾向にあることがわかった。

## Analysis of Coordinators in an OSS Community with Multiple Sub Communities

AKINORI IHARA,<sup>†1</sup> HIROTAKA MAESHIMA,<sup>†1</sup>  
SHINSUKE MATSUMOTO,<sup>†1</sup> YASUTAKA KAMEI,<sup>†1</sup> OHIRA MASAO<sup>†1</sup>  
and MATSUMOTO KEN-ICHI<sup>†1</sup>

Since Open Source Software (OSS) development heavily depends on collaborations among participants such as developers and users, one of the most important aspects for OSS development is the existence of coordinators who play the roles of sharing information and arranging communications among developers and users. Although previous studies have explored coordinators in OSS communities, it is not clear about the relationship between coordinators and bug reporters who are also important to realize software with high reliability. In this paper, we analyzed coordinators who are intermediate among three sub communities (developers, users, and bug reporters). As the results of analysis, we have found that the coordinators who are intermediate among three sub communities were on the increase as the PostgreSQL community progressed.

### 1. はじめに

Linux や Apache に代表されるように、近年、Open Source Software (OSS) は商用ソフトウェアに劣らない品質と機能を備えるようになっており広く普及が進んでいる。例えば、Apache は、2007 年 6 月現在、Web サーバソフトウェアの市場シェアの約 7 割と大多数を占めており<sup>8)</sup>、情報化社会における社会基盤を支えているといっても過言ではない。

一般的な OSS の開発は、自由参加のボランティアが WWW 上に形成するオンラインコミュニティ (OSS コミュ

ニティ) によって行われている。OSS の社会的・経済的影響が大きくなってきていることから、近年、Linux や Apache などの OSS コミュニティを分析対象として、その成功要因を明らかにするための研究が現在盛んに行われている<sup>4),6),7),11)</sup>。

我々の先行研究<sup>12)</sup> では、3つの著名な OSS コミュニティ (Apache, Gimp, Netscape) を対象として、各コミュニティ内のサブコミュニティ (開発者コミュニティとユーザコミュニティ) 間で形成されるコミュニケーション構造について分析を行った。分析の結果、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間の橋渡し役を務め、開発者とユーザの協調作業を支えるコーディネータの存在が重要であることが明らかとなった。

先行研究は、サブコミュニティ間の協調作業を円滑化するコーディネータの存在を同定することを第一目的とした

<sup>†1</sup> 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

ため、OSS コミュニティのサブコミュニティのうち開発者コミュニティとユーザコミュニティのみに着目し分析を行った。しかし、一般的な OSS コミュニティの参加者は開発者とユーザに限らない。特に、欠陥が少なく信頼性の高いソフトウェアを実現するためには、バグを発見し開発者へ伝えるバグ報告者サブコミュニティが重要な役割を担う<sup>9),11)</sup>。

そこで、本稿では PostgreSQL コミュニティを対象として、3つのサブコミュニティ（開発者コミュニティ、ユーザコミュニティ、バグ報告者コミュニティ）に存在するコーディネータに着目して、OSS コミュニティ成立初期からのサブコミュニティの成長の様子を観察したケーススタディについて報告を行う。ケーススタディの対象として PostgreSQL コミュニティを選んだ理由は、PostgreSQL は関係データベース管理システムであり高い信頼性を確保する必要があるという特徴があるため、特にバグ報告者が重要な役割を担っているコミュニティであると考えられるためである。

以降 2 章では、関連研究について述べ本研究の立場を明らかにする。3 章では、本研究で想定している OSS コミュニティにおける参加者間のコミュニケーション構造とコーディネータについて述べ概念定義を行う。4 章では、ケーススタディで行う分析の方法について説明し、5 章で PostgreSQL を対象としたケーススタディについて報告する。6 章では、ケーススタディの結果を考察し、7 章で本稿のまとめと今後の課題を述べる。

## 2. 関連研究

OSS コミュニティの参加者の内、協調作業を支えるコーディネータ（コーディネータの定義については次章で述べる。）の分析を行った研究として、Bird らの研究<sup>1)</sup> や大平らの研究<sup>12)</sup> がある。

Bird らは、Apache のサブコミュニティである開発者コミュニティを対象として、開発者コミュニティの内での協調作業を支えるコーディネータの分析を行っている。開発者同士のつながりを表すコミュニケーション構造（コミュニケーション構造については次章で述べる。）から求めた開発者の交流人数と、ソースコードの更新履歴から求めた開発者の活動の度合との相関を調べることで協調作業の実態を明らかにした。この研究では、多くの開発者と交流している開発者ほど、活動の程度が高くサブコミュニティ内での協調作業に寄与していることを示唆する結果を得ている。

本稿は、開発者コミュニティという単一のサブコミュニティ内のコーディネータに着目した Bird らの研究に対して、OSS コミュニティ内に存在するサブコミュニティ間の協調作業を支えるコーディネータに着目している点が異なる。

大平らは、OSS コミュニティはユーザ参加型コミュニティ

であると指摘する Raymond らの研究<sup>9)</sup> に着目して、3 つ OSS コミュニティ（Apache, Gimp, Netscape）を対象に、開発者とユーザの 2 つのサブコミュニティ間の協調作業を支えるコーディネータの分析を行っている。2 つのサブコミュニティに属する人物をコーディネータとして捉え、それぞれのサブコミュニティに存在するコーディネータの交流人数と、OSS コミュニティの活動の活発さとの関係から協調作業の実態を明らかにした。この研究では、活動が活発な OSS コミュニティのコーディネータほど、2 つのサブコミュニティに対して偏りなく交流する傾向にあることを確認している。

本稿では、バグ報告者を分析の対象に加えることで、OSS コミュニティにおける開発者、ユーザ、バグ報告者の 3 者の協調作業の実態をコーディネータの視点から観察する。

## 3. OSS コミュニティにおけるコミュニケーション構造とコーディネータ

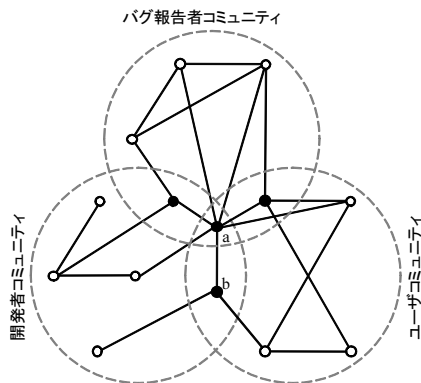
### 3.1 コミュニケーション構造

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、Mailing List (ML) や掲示板などのオンラインメディアを通じた非対面のコミュニケーションを基本とする。また、OSS コミュニティの参加者はオンラインメディアを利用し、役職や年功による序列にほとんど縛られることなく自由に対等な立場で情報交換や議論を行うことができる。ただし、参加者の役割（開発者やユーザなど）によって議論したい話題の種類（新機能についての話題やソフトウェアの使い方についての話題など）が大きく異なることを考慮して、多くの OSS コミュニティでは、複数の ML や掲示板を用意することでサブコミュニティ（開発者コミュニティやユーザコミュニティ）が形成されている。

このような特徴を有する OSS コミュニティ参加者らのコミュニケーション構造は図 1 のように表現することができる。各参加者はそれぞれの役割に応じてサブコミュニティ内で他の参加者とコミュニケーションを行う。次節で説明するコーディネータは複数のサブコミュニティに参加している参加者である。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間のコミュニケーション構造に大きな影響を与える<sup>10)</sup> と考えられるため、本稿では、分析対象とするサブ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（参加者同士のつながり）と捉え、参加者（ノード）とエッジの関係によって OSS コミュニティのコミュニケーション構造を定義する。

### 3.2 コーディネータ

開発者コミュニティとユーザコミュニティといった複数のサブコミュニティを調整し媒介する人物をコーディネータと呼ぶ<sup>2)9)</sup>。本稿では、図 1 に示す参加者  $a$  や  $b$  のよう



●: コーディネータ ○: 参加者 ー: 参加者のつながり

図1 OSSコミュニティのコミュニケーション構造

に、複数のサブコミュニティと送受信関係のある参加者は何かしらの調整行動をはかっていると考え、コーディネータとみなす。図1の参加者aは、開発者、ユーザ、バグ報告者、それぞれのサブコミュニティの参加者とメッセージの送受信関係があり、開発者、ユーザ、バグ報告者、3つのサブコミュニティを結ぶコーディネータ（開発者&ユーザ&バグ報告者）である。参加者bは、開発者コミュニティとユーザコミュニティの参加者それぞれとメッセージの送受信関係があり、開発者コミュニティとユーザコミュニティを結ぶコーディネータ（開発者&ユーザ）である。このようなコーディネータは3つのサブコミュニティを対象とした場合、以下の4種類が挙げられる。

- コーディネータ（開発者&ユーザ）
- コーディネータ（開発者&バグ報告者）
- コーディネータ（ユーザ&バグ報告者）
- コーディネータ（開発者&ユーザ&バグ報告者）

ケーススタディでは、PostgreSQLにおける4種類のコーディネータと3つのサブコミュニティについて分析を行う。

#### 4. 分析方法

##### 4.1 コミュニケーション構造分析

メッセージの送信者と返信者の関係からコミュニケーション構造を構成し、コミュニティ成立初期からのコミュニティの成長の様子を観察するために、コーディネータの総数といった統計量を用いてその構造の特徴を分析する。本稿では、以下の統計量を用いる。

- サブコミュニティごとの参加者の総数

参加者の総数が増減するということは、コミュニティの規模がより大きく（小さく）なるということであり、コミュニティが成長（衰退）しているとみなすことができる。

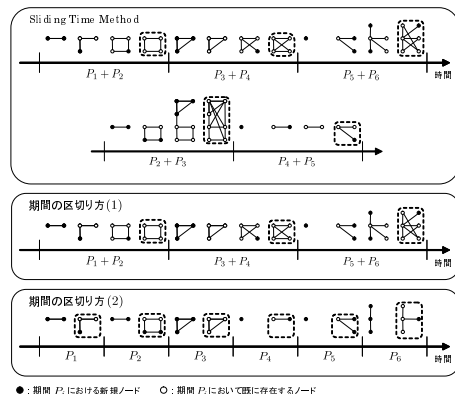


図2 ネットワーク構造の時系列分析方法<sup>5)</sup>

- サブコミュニティごとのコーディネータの総数  
コーディネータの総数が増減するということは、コミュニティがより円滑に機能する（しない）ということであり、コミュニティが成長（衰退）しているとみなすことができる。
- サブコミュニティごとのコーディネータの割合  
コーディネータの総数に加えて、サブコミュニティ内でのコーディネータの割合を調べることでより正確にコミュニティが円滑に機能しているかどうかを調べることができる。例えば、コーディネータの総数は一定で参加者が一時的に増加した場合、コーディネータの割合は小さくなるため以前に比べてコミュニティの運営は困難（コーディネータの不足状態）になっているものとみなすことができる。

##### 4.2 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ネットワーク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る必要がある。

本稿では、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<sup>5)</sup>。ある一定期間におけるコミュニケーション構造の求め方を図2に示す。ある一定期間のコミュニケーション構造はその期間の終了時期のものである。

本稿で用いる Sliding Time Method は、分析の期間が重複しない図2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構造の変化を捉えやすい。例えば、図2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構造では、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。

本稿では、OSSコミュニティにおける開発者の社会的構

表 1 各 ML の設置開始時期

| ML            | 開始年月        |
|---------------|-------------|
| pgsql-hackers | 1996 年 1 月  |
| pgsql-patches | 1997 年 6 月  |
| pgsql-general | 1998 年 5 月  |
| pgsql-bug     | 1996 年 12 月 |

造を分析した研究<sup>3)</sup>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。

## 5. ケーススタディ

本章では、PostgreSQL コミュニティにおける開発者、ユーザ、バグ報告者の 3 者の協調作業の実態をコーディネータの視点から観察したケーススタディについて述べる。

### 5.1 PostgreSQL

PostgreSQL は BSD ライセンスに基づき配布されているフリーの関係データベース管理システム (RDBMS) である。商用 / 非商用を問わず無償で利用することができ、企業システムへの採用も増えていることから、商用の RDBMS に劣らないソフトウェアとして広く認知されている。また、データベース管理システムであるという点から、特に高い信頼性が求められるソフトウェアであるといえる。

PostgreSQL は、1986 年に POSTGRES プロジェクトとして発足し開発されてきたものが、1995 年にオープンソース版 (Postgres95) として Web 上でリリースされた後、1996 年終わりに名称を PostgreSQL に変更して ver.6.0 としてリリースされ現在まで開発が続いている。ver.6.0 以降は、ver.7.0, ver.8.0 がリリースされている。ver.8.0 ではテスト版として ver.8.0 $\beta$  がリリースされている。本稿では、PostgreSQL の開発が開始された ver.6.0 のリリース時期から ver.7.0 のリリース時期にかけてを前期、ver.7.0 のリリース時期から ver.8.0 のリリース時期にかけてを中期、ver.8.0 のリリース時期以降を後期として分析を行った。

### 5.2 分析対象データ

分析対象データは各コミュニティの議論の場として主に用いられる ML とする。本稿では、各 ML の利用者の役割に応じてサブコミュニティを定義した。以下では各サブコミュニティが利用している ML の目的について述べる。

#### ● 開発者コミュニティ

pgsql-hackers : PostgreSQL の開発に関する話題を取り扱う ML であり、開発中のバグや新規機能についての議論が主として行われる。

pgsql-patches : PostgreSQL の開発に関する話題を取り扱う ML であり、リリース後のバグの修正や新規機能についての議論が主として行われる。

#### ● ユーザコミュニティ

pgsql-general : PostgreSQL のユーザが利用する ML

であり、主として導入時のトラブルや SQL のバージョン問題など、開発者を交えて議論が行われる。

#### ● バグ報告者コミュニティ

pgsql-bug : PostgreSQL のバグ報告専用の ML であり、主としてバグの修正についての議論が行われる。また、報告されたバグの修正が行われた場合は pgsql-patches へメッセージを送ることになっている。

分析期間としては、ML への投稿履歴が存在する全ての期間を利用したが、ML ごとにその設置時期が異なるため、ML が存在しない期間については参加人数を 0 人として扱った。表 1 に各 ML の設置時期を示す。

### 5.3 分析結果

各サブコミュニティに属する人数の推移を図 3 に、コーディネータの人数の推移を図 4 に示す。また、各コミュニティのコーディネータの割合 (他のコミュニティと兼任している割合) を図 5 に示す。グラフの横軸は時間を表し、縦軸について、図 3 と図 4 は人数を、図 5 は割合を表す。また、縦の破線はリリース時期を示す。ただし、図 5 の初期 (1996 年 ~ 1998 年) 頃は激しく変動している。これは ML が設置され始めたばかりで、全てのコミュニティの参加人数が少なかったためである。

以下では、各グラフの分析期間を前期・中期・後期に分け、3 つのサブコミュニティにおけるコーディネータに着目してコミュニティの成長の様子を分析する。

前期: 図 3 よりユーザ ML 設置以来、ユーザコミュニティの参加人数が増加し、ver.7.0 のリリース後、ユーザコミュニティの参加人数が最大となる。このユーザコミュニティの参加人数が増加する時期に、開発者のコーディネータの兼任率も増加する傾向がある。ここで、図 4 を分析すると、開発者コミュニティに関するコーディネータのうち、最も増加しているのはコーディネータ (開発者 & ユーザ) であることが確認できる。

中期: 図 4 より、ver.7.0 のリリース以降、コーディネータ (開発者 & ユーザ) の人数は減少しているが、中期の期間、コーディネータ (開発者 & ユーザ & バグ報告者) の人数は緩やかに増加していることが確認できる。

後期: 図 5 において ver.8.0 $\beta$  のリリースから ver.8.0 のリリースにかけてバグ報告者コミュニティ内のコーディネータの割合が急に減少し、一方で、図 3 からバグ報告者コミュニティ全体の参加人数が増加していることが確認できる。

## 6. 考 察

本章では、前期、中期、後期において得られた分析結果をそれぞれ考察する。

前期: ver.7.0 のリリース前後にユーザコミュニティの参加人数が増加し、同時期にコーディネータ (開発者 & ユー

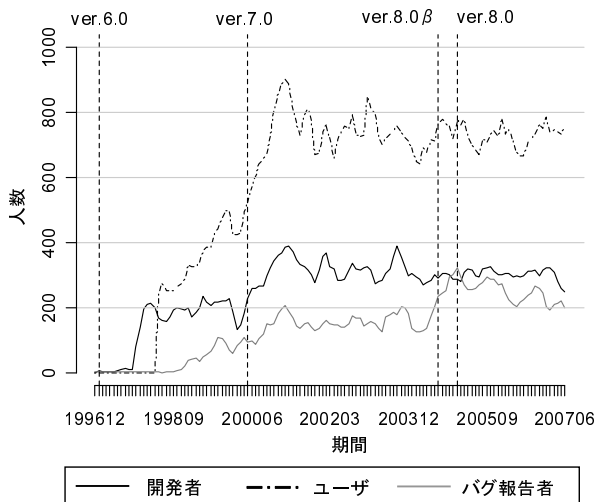


図 3 各コミュニティの総参加人数の推移

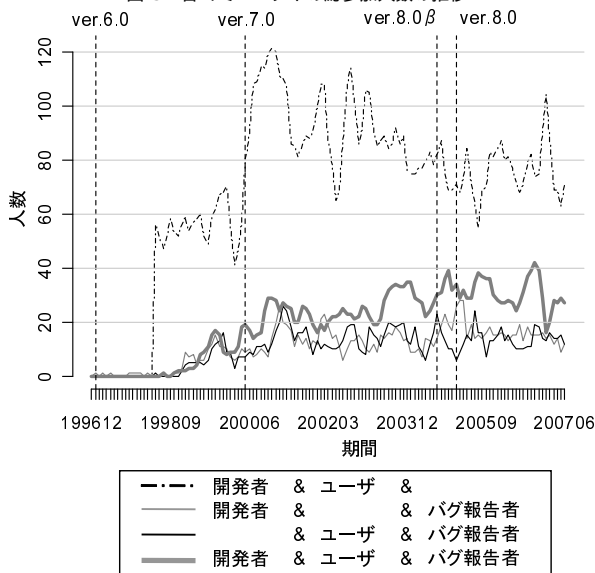


図 4 コーディネータの人数の推移

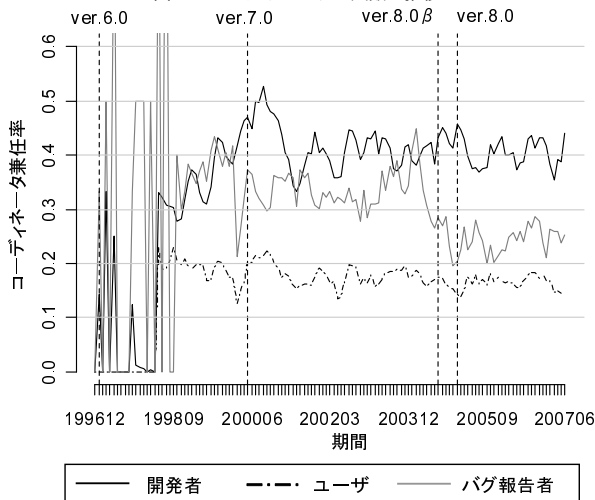


図 5 各コミュニティのコーディネータの割合の推移

ザ)の人数が増加していることを確認できる。これは解釈の一つとして、ユーザコミュニティの参加人数の増加によりユーザからの要望などが増え、それに応えるために多くの開発者がユーザコミュニティにも参加したためと考えられる。ユーザコミュニティでは導入時のトラブルといった議論が開発者を交えて行われており、ver.7.0のリリース時期ということもあり、開発者がユーザのサポートを行った結果と思われる。

中期: ver.7.0をリリースして以降、コーディネータ(開発者&ユーザ)の人数は減少しており、コーディネータ(開発者&ユーザ&バグ報告者)の人数は増加している。さらに分析を行うと、図3より、3つのサブコミュニティの参加人数はほとんど変動しておらず、また、図4より、コーディネータ(開発者&バグ報告者)とコーディネータ(ユーザ&バグ報告者)はほとんど変動していないことが確認できる。よって、コーディネータ(開発者&ユーザ)がバグ報告者コミュニティにも参加し始めたことで、コーディネータ(開発者&ユーザ)の人数が減少し、コーディネータ(開発者&ユーザ&バグ報告者)の人数が増加したと考えられる。

バグ報告者コミュニティでは、バグの修正についての議論などが行われているが、バグ報告者はバグを報告するのみであり、実際にバグを修正するのは開発者であるため、コーディネータ(開発者&ユーザ)がバグ報告者コミュニティにも参加したのと考えられる。この傾向は、コーディネータ(開発者&ユーザ&バグ報告者)により3つのサブコミュニティが協調作業を行い始めているものとして捉える事ができる。

後期: ver.8.0β, ver.8.0のリリース頃において、バグ報告者コミュニティにおけるコーディネータの割合が減少すると同時にバグ報告者コミュニティの参加人数が増加している。つまり、バグ報告者コミュニティにのみ参加している個人が増加していると考えられることから、新規のバグ報告者が増加し始めたと考えられる。

ver.8.0はテスト版としてver.8.0βがリリースされることも考えると、バグ報告者が急増しバグ報告される量が増加したことが考えられる。このようにバグ報告が増加すれば、開発者がバグ報告に応えていくことが必要となり、開発者であるコーディネータ(開発者&バグ報告者)やコーディネータ(開発者&ユーザ&バグ報告者)の必要性が高くなるものと考えられることができる。

## 7. おわりに

本稿ではOSSコミュニティにおける開発者コミュニティ、ユーザコミュニティ、バグ報告者コミュニティという3つのサブコミュニティに存在するコーディネータに着目し、コミュニティ成立初期からのOSSコミュニティの成長の様子

を分析した。PostgreSQL を対象として行ったケーススタディから得られた知見を以下に示す。

- 分析期間通して、3つのサブコミュニティにおける4種類のコーディネータを確認できたが、コーディネータによってコーディネータの人数は大きく異なる。
- (初期)ユーザ数の変動に対応して開発者コミュニティにおけるコーディネータの割合が増加し、特にコーディネータ(開発者&ユーザ)の人数が増加している。
- (中期)3つのサブコミュニティの協調作業を行うコーディネータ(開発者&ユーザ&バグ報告者)が増加していく傾向にある。
- (後期)バグ報告者コミュニティの参加人数が増加している。

本稿では人数の推移のみで分析を行った。しかし、本稿で行った分析だけでは、参加者がどのサブコミュニティの個人とどれくらいの頻度でコミュニケーションをとっているのか(メッセージの送信, 返信回数)まで知ることはできない。また、コーディネータが複数のサブコミュニティに参加していく様子を分析することは難しい。よって、今後は以下の3点を課題とする。

- 参加者間のコミュニケーションの頻度をメッセージの送信, 返信回数から分析する。
- サブコミュニティの参加者とコーディネータ間における情報伝達の方向を分析する。
- コーディネータが他のサブコミュニティに参加することで引き起こる人数の変動について分析する。

以上の課題をコミュニケーション構造より分析し、さらに他のOSSコミュニティと比較しながらOSSコミュニティの成長していく過程を分析し、OSS開発の成功要因を明らかにしていきたい。

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) 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) 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).
- 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) 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 (KCSW'06)*, pp. 35-41 (2006).
- 6) 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).
- 7) 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).
- 8) Netcraft Ltd.: Netcraft Web Server Survey, available from (<http://www.netcraft.com/Survey/>) (accessed 2007-06-30).
- 9) Raymond, E. S.: *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*, O'Reilly and Associates (1999).
- 10) 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).
- 11) 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).
- 12) 大平雅雄, 榎本真佑, 前島弘敬, 亀井靖高, 松本健一: 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, グループウェアとネットワークサービスワークショップ2007, (to appear) (2007).