社団法人 電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE

メールスレッドのクラスター分析による OSS プロジェクトのアクティビティ予測手法

大蔵 君治† 大西 洋司† 川口 真司† 大平 雅雄† 飯田 元† 松本 健一†

†奈良先端科学技術大学院大学 情報科学研究科 〒630-0192 奈良県生駒市高山町 8916 番地の 5

E-mail: †{kimiha-o, yoji-o, kawaguti, masao, matumoto}@is.naist.jp, iida@itc.naist.jp

あらまし オープンソースソフトウェア(OSS)開発において，プロジェクトのアクティビティ（開発の活発度）は成果物の品質に影響を及ぼす重要な要因の一つである．OSS 開発では，一般的に詳細な開発ドキュメントを作成することは少ないため，開発ツールのログデータを用いた分析手法が多く用いられてきた．しかしながら，開発ツールはプロジェクトによって多種多様なものが使用されるため，異なるプロジェクトに対して同じ分析手法を用いることは困難である．本稿では，多くの OSS プロジェクトにおいて開発に用いられるメーリングリスト(ML)に着目し，開発者の共起性から OSS プロジェクトのアクティビティを予測する手法を提案する．我々は，実際の OSS プロジェクトに対して提案手法の適用実験を行い，開発者の共起性がプロジェクトのアクティビティに影響を与えることを確認した．

キーワード OSS，クラスター分析，メーリングリスト，アクティビティ，プロジェクト分析

A Method for Activity Prediction using Cluster Analysis on Email Threads

Kimiharu OHKURA† Yoji ONISHI† Shinji KAWAGUCHI† Masao OHIRA†
Hajimu IIDA† Ken-ichi MATSUMOTO†

†Graduate School of Information Science, Nara Institute of Science Technology
8916-5 Takayama-cho, Ikoma-shi, Nara, 630-0192 Japan

E-mail: †{kimiha-o, yoji-o, kawaguti, masao, matumoto}@is.naist.jp, iida@itc.naist.jp

Abstract In open-source software (OSS) development, *activity* is one of important factors which affect quality of software products. Various development tools' logs have been used to analyze certain OSS projects because of the lack of development documents in OSS projects. However, it is difficult to analyze OSS projects in common since tools used in software development vary from project to project. In this paper, we focus on mailing lists (MLs) that are used in most OSS projects, and propose a method for activity prediction based on in co-occurrence of developers in MLs. From the result of our experiment, we confirmed that co-occurrence of developers affected the activity in OSS projects.

Keyword OSS, Cluster Analysis, mailing list (ML), Activity, Project Analysis

1. はじめに

近年，ソフトウェア開発の形態としてオープンソースソフトウェア(OSS)開発が注目されている．OSS 開発は，主として Web 等のネットワークを介して行われ，開発者，あるいはユーザ間のコミュニケーション手段としてメーリングリスト(以降，ML)が広く用いられている．OSS プロジェクトは企業における一般的なソフトウェア開発と異なり，開発ドキュメントや作業記録の作成，ソースコードの秘匿といった義務は無く，プロジェクトへの参加及び脱退も自由であることが多い．そのため，開発者に課せられる責務は企業におけるソフトウェア開発と比べ軽微である．しかしながら，開発者にとって制約が少ない利点がある反面，プロジェクトの進捗状況の把握や事後分析に必要となる情報が不足しがちである．さらに，分析対象となるデータのフォーマットが統一されていないため，ますます分析が困難なものとなる．

特に OSS 開発では，プロジェクトのアクティビティ（開発の活発度）は成果物の品質に影響を及ぼす重要な要因の一つであるため，プロジェクト管理者やプロジェクトの成果物を利用する組織がアクティビティを正確に把握できることが望ましいが，前述の理由から現状では限定的にしかプロジェクトのアクティビティを把握することができない．そこで我々は，多くの OSS プロジェクトで使用されているコミュニケーションツールである ML に着目し，ML アーカイブのみを用いてプロジェクトのアクティビティを分析する手法を提

案する．

本研究におけるアクティビティとは，そのプロジェクトにおける開発の活発度であると定義する．「アクティビティが高い」状態とは，プロジェクトが活発に活動しており，開発が停滞無く進んでいる状態を言う．逆に「アクティビティが低い」状態とは，開発が滞っておりプロジェクトの活動が衰退傾向を示している状態を指す．我々はOSSプロジェクトのアクティビティに対して，次のような仮説を立てた．

仮説：MLのスレッド内における開発者の共起性が高いほど，そのプロジェクトはアクティビティが低い

これは，多くの開発者が参加している話題が一定期間に多く存在するほど議論がまとまっておらず，プロジェクトが混乱している状態にあり開発の進捗が遅れていると考えられるからである．本稿では，MLのスレッドにおける開発者の共起性をクラスター分析によって計測し仮説を検証する．本稿の構成は以下の通りである．まず，2章において関連する研究を紹介し，3章では，本研究で提案する分析手法の詳細について説明する．4章で適用実験による仮説の検証を行い，5章にて，まとめと今後の課題について述べる．

2. 関連研究

MLを対象としたソフトウェア開発プロジェクトの分析手法は，過去にも様々なものが提案されてきた．Birdらは MLアーカイブに対してソーシャルネットワーク分析手法を適用し，OSSプロジェクトにおけるソフトウェア開発のアクティビティと ML内でのアクティビティに相関があることや，開発者と非開発者とのアクティビティの違い等を明らかにしている [1]．また，Rigbyらは OSSプロジェクトにおける開発者の言語的手がかりを用いて，メジャーリリース前後における開発者の言動の変化や，優れた開発者に見られる言語的特徴を計量心理学の観点から分析している[3]．また，Hanakawaらは開発における ML内での議論の集中度に着目し，OSSプロジェクトのバグトラッキングシステムを対象とした分析手法を提案している[2].

これらの研究は，ある時点におけるプロジェクトの状態を把握することを可能にするが，分析において事前知識を多く必要とするほか，分析対象のフォーマットが制限されている（例えば，本文は英語でフォーマルな会話でなければならない等）ため，手法を一般化することは困難である．本研究では，分析対象として ML アーカイブのみを用いるため，開発者同士のコミュニケーション手段に MLを用いている OSSプロジェクト全てに手法を適用可能であり，分析は一部の処理を除きプログラムによって機械的に行われるため，分析者が手法について事前知識を備えておく必要はない．

	開発者A	開発者B	開発者C	開発者D	開発者E
スレッド1	0	2	1	1	0
スレッド2	0	0	0	1	1
スレッド3	2	1	1	0	0
スレッド4	0	1	1	0	0
スレッド5	0	1	0	1	1

図 1. 各ユーザの投稿頻度によるスレッドのベクトル化例

3. 分析手法

我々は，1章で述べた仮説に基づいて開発者の共起性を MLアーカイブから分析する．本研究で提案する手法は，次の4ステップからなる．

1. ユーザの抽出：MLアーカイブから参加ユーザを洗い出す．
2. スレッドの抽出：ヘッダ情報から MLのスレッド構造を構築する．
3. スレッドの符号化：参加ユーザの投稿頻度を特徴量として，スレッドをベクトル化する．
4. クラスタリング：ベクトル化されたスレッドを，クラスタリングアルゴリズムに基づいて幾つかのクラスターに分類する．

以降では，各手順の詳細について述べる．

3.1. ユーザの抽出

始めに，MLアーカイブから MLに参加しているユーザを抽出する．ユーザはメールヘッダに含まれる From行から抽出するが，同一ユーザ（以降ユニークユーザと呼ぶ）の識別を行うために，以下の前処理を行う．

3.1.1 名前部とアドレス部の分離

OSSプロジェクトで用いられる MLの多くは，From行が名前部とクォート記号で囲まれたアドレス部から構成されている．以下にその例を示す．

Kimiharu Ohkura <kimiha-@is.naist.jp>

From行をそのまま識別子として用いた場合，アドレスは同じだが名前部が異なっているときに，それらが異なるユーザとして扱われてしまう．我々は，全てのメールについて From 行から名前とアドレスの分離を

行い，アドレスが同じであれば同一ユーザとみなすようにした．

3.1.2　空白文字と記号の除去

我々はまた，分離させた名前部とアドレス部の各々について，空白文字と記号の削除を行った．From 行の名前部は，ダブルクォーテーションで囲われていることや，前後に余計な空白が付与されているケースがしばしば存在する．例えば「”Ohkura”」と「Ohkura」は異なるユーザとみなされてしまうため，名前部に含まれる全ての記号と空白文字を除去した．空白文字にはスペース，タブ，改行が含まれる．

3.2. スレッドの抽出

次にスレッドを構築するため，全てのメールについて Message-Id ヘッダ，及び In-Reply-To ヘッダを参照し，プログラム上でスレッドのツリーを形成した．本来であれば Reference ヘッダも考慮に入れるべきであるが，今回用いたデータセットでは Reference ヘッダはほとんど使用されておらず，また In-Reply-To のみでスレッドを構築可能であったため扱わなかった．

以上の前処理は，mbox 形式の ML アーカイブを対象とした Perl プログラムを用いて行った．また，重複している Message-Id や，本文の行頭に From が含まれていて抽出に失敗した一部のメールについては手動で処理を行った．

3.3. スレッドの符号化

次に，抽出した全スレッドとユニークユーザのリストを基に各スレッドを符号化する．符号化は，スレッド毎に各ユーザの投稿頻度を記録しベクトルを生成することで行う（**図 1** に例を示す）．全てのスレッドをベクトル化した後，ベクトル空間モデル[4] に基づいて各スレッド間の距離を計算する．

本研究ではベクトル間の距離としてコサイン類似度を用いた．コサイン類似度は二つのベクトルが織りなす角によって類似性を計測する距離尺度である．ベクトル間の距離としてよく用いられる他の尺度としてユークリッド距離がある．しかし，全く共起が発生していないスレッド間(積集合が空であるベクトル同士)でも距離が近くなる場合があるため，ユークリッド距離は共起の強さを表す尺度として適切ではない．

コサイン類似度はベクトル間の内積から計算されるため，共起が全く発生していないスレッドは類似度が 0 となる．これにより，全く共起が発生していないスレッド間で距離が近く，すなわち類似度が高くなることは無くなり，共起性を計る尺度としてより適切な結果を得ることができる．本研究ではベクトルの特徴量として開発者の投稿頻度を用いているため，“類似度の高いスレッド”は“同じようなユーザが参加しているスレッド”と考えることができる．

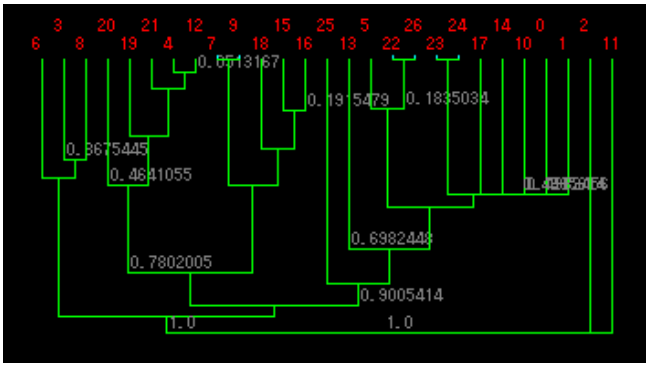

図 2. Kaffe プロジェクト 2005 年 10 月の樹形図

3.4. クラスタリング

次に，クラスタリングアルゴリズムに基づいて各スレッドを分類する．本研究では，クラスタリングアルゴリズムに階層型クラスタリング手法の 1 つである群平均法を用いた．

群平均法は，クラスターに含まれる要素間の平均距離をクラスター間の距離と定義し，距離の近いクラスター同士を結合させていく方法である．一般的に精度が高いと言われている他のクラスタリングアルゴリズムとしては Ward 法があるが，先に述べた理由により距離関数としてコサイン類似度を用いるため，ユークリッド距離を前提としている Ward 法を用いることはできない．この理由から，我々は他のアルゴリズムにおいて最も適していると考えられる群平均法を用いてクラスター分析を行った．

4. 実験

1 章で述べた仮説を検証するため，実際の OSS プロジェクトに対して提案手法の適用実験を行った．

4.1. データセット

実験は OSS プロジェクト Kaffe [5] を対象に行った．Kaffe プロジェクトは 1995 年から開始したオープンソースの Java 仮想マシン開発プロジェクトであり，1996 年以降の CVS リポジトリと ML アーカイブが公開されている．公開されている ML アーカイブに対して提案手法を適用し，CVS のコミット数との比較を通して提案手法の妥当性を検討する．

4.2. 実験方法

我々は，対象プロジェクトの ML アーカイブを月単位に分割し，各月ごとの ML アーカイブに対して提案手法を適用した．そして，得られたクラスターに関する幾つかの統計値と，CVS リポジトリから得られた月毎のコミット総数とを比較した．実験は 2004 年 1 月～

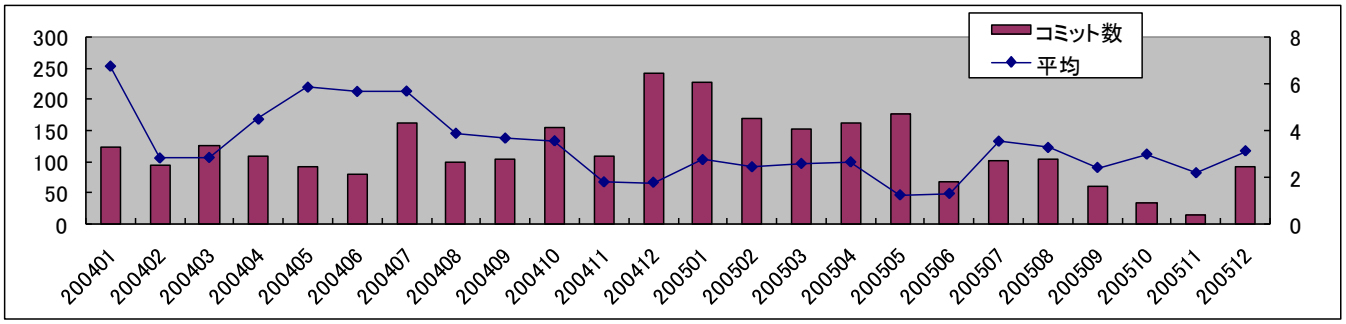

図 3. クラスター要素数の平均とコミット数の推移

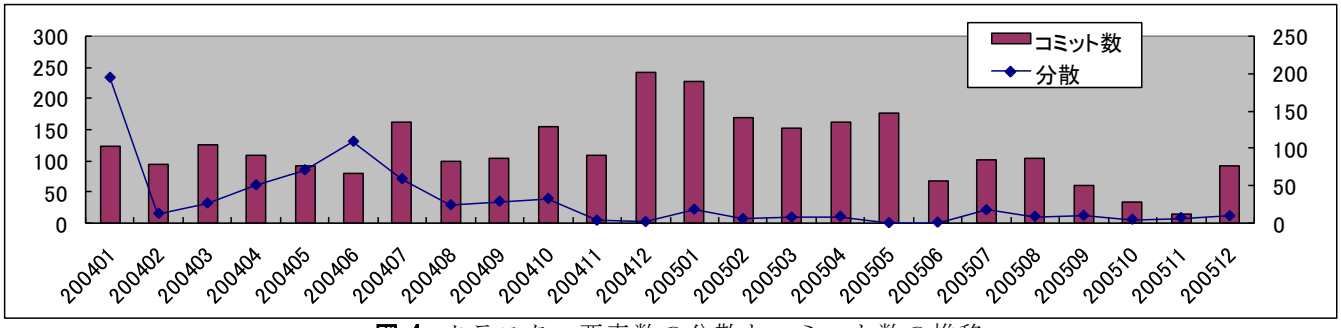

図 4. クラスター要素数の分散とコミット数の推移

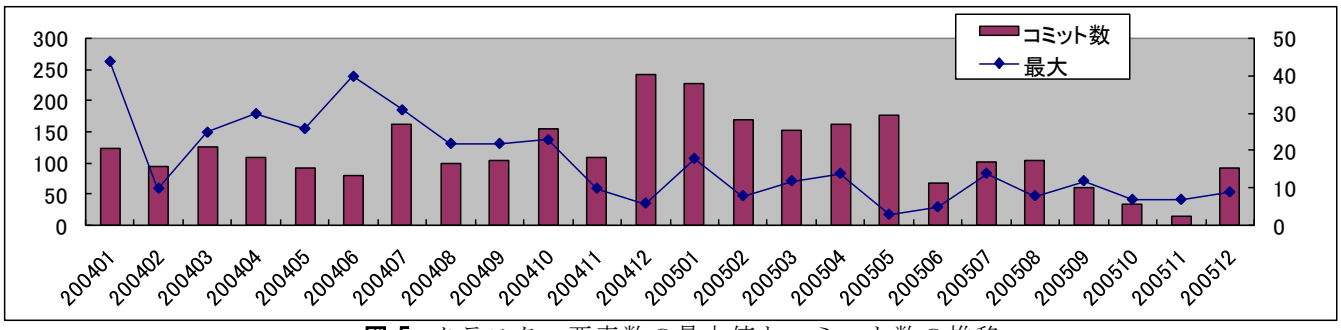

図 5. クラスター要素数の最大値とコミット数の推移

2005 年 12 月の 24 か月分の ML アーカイブに対して行った.

本手法ではクラスタリングアルゴリズムとして階層型クラスタリングを用いているが，これは距離の近いクラスターを順に結合し，最終的に 1 つのクラスターに併合されるというアルゴリズムであるため，適切な閾値を定めてクラスターを分割する必要がある.

対象プロジェクトをクラスタリングした際の樹形図の例を**図 2** に示す．図は 2005 年 10 月のものを用いた．樹形図作成プログラムの都合上，図では上階層ほど数値が 1.0 に近づいているが，コサイン類似度では通常 1.0 に近づくほど距離が近いと定義される．本稿においても，以降は「類似度が高い」と言った場合は「ベクトル間の距離が近い」と同義であるとする.我々は 2004 年から 2005 年における月毎の樹形図から，十分に共起が発生していると考えられる類似度の閾値を経験的に 0.5 と設定した.

4.3. 結果・考察

我々は，3 章で述べた手法に基づいて形成されたクラスターに関する統計値として，月毎にみたクラスター要素数の平均・分散・最大値とクラスター数そのものを用い，コミット数と比較した.

最初に，クラスター要素数に関する統計値についてその結果を考察する.クラスター要素数の平均,分散,最大について，月毎のコミット数と比較したグラフを**図 3**，**図 4**，**図 5** に示す．グラフ縦軸の左の目盛がコミット数で，右側の値がそれぞれの統計値である.

平均（図 3）と最大値（図 5）については，2004 年に激しく変動し，2005 年後半にかけて変動が緩やかになるという傾向が見られた．しかし，分散（図 4）は各月における値の幅が大きく，その推移について有意な傾向は認められなかった.

クラスター要素数の平均（図 3）は，その月における話題の平均規模を表す．話題の平均規模が大きくなればなるほど，多くの開発者が色々な話題に加わって

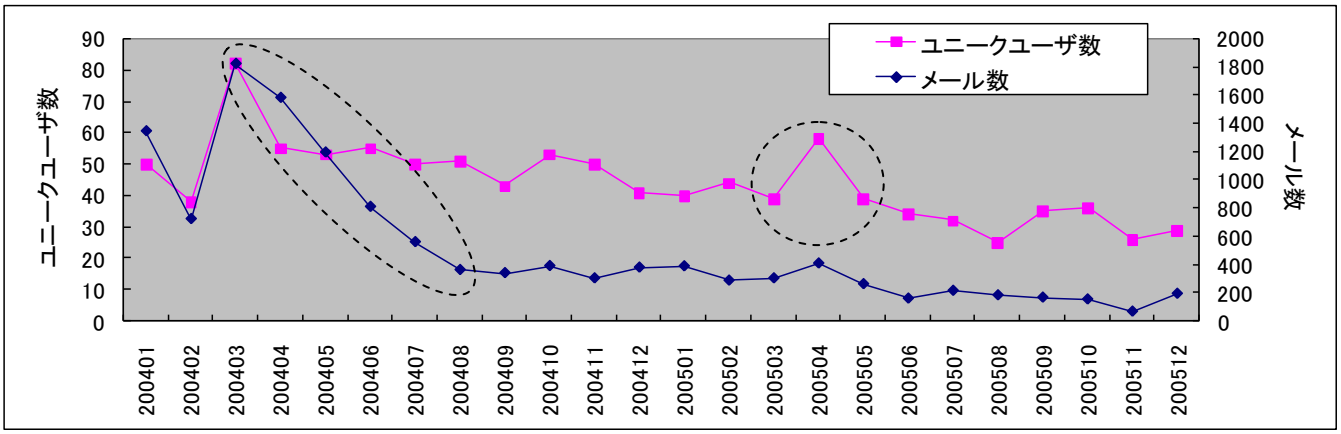

図 6. 各月のメール件数とユニークユーザ数の推移

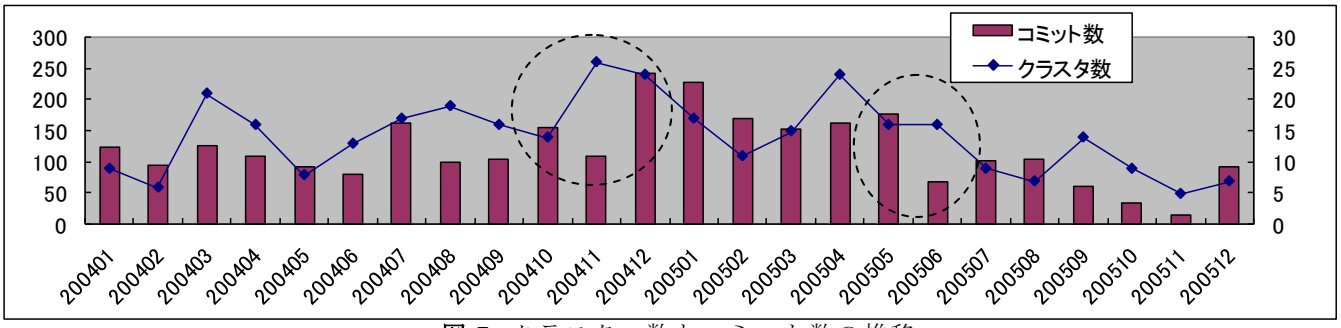

図 7. クラスター数とコミット数の推移

いる,即ち開発者の共起が発生していると考えられる.したがって，我々の仮説が正しいとすれば，平均値が高いときほど開発が停滞していると考えられる．グラフを見ると，平均値の低い 2004 年末から 2005 年 6 月の間，コミット数も同じく高い値を記録しており仮説とは矛盾しない．しかし，クラスターの要素は ML スレッドであるため，その平均値は各月に送信されたメールの件数に大きく影響を受ける．

我々は，実験対象期間におけるメール送信数の推移を調べた．**図 6** 中の折れ線の一つにその結果を示す．図 6 から，対象プロジェクトは 2004 年 3 月から 2004 年 8 月にかけて大きくメール件数が減少していることがわかる.要素数の平均も 2004 年 8 月以降停滞傾向を示しているため，図 3 のみからはクラスター要素数の平均とプロジェクトのアクティビティの間に直接的な相関があると言うことはできない．

クラスター要素数の最大値（図 5）は，その月における最も大きな話題の規模を示す．開発者全員に対するアンケートや，今後の開発方針といった巨大な話題が発生した際に高い値を記録し，小規模な話題しか発生していない月は値が低くなる．しかし，今回の結果からは平均値と同じような推移を見せており最大値を用いたプロジェクトの分析は困難であると考えられる．

以上の結果をまとめると，クラスターの要素に関する統計値は，メールの件数そのものに影響を受けるため，メールの件数にばらつきがあるプロジェクトのアクティビティを計測することはできないと言える．

最後に，クラスターの要素ではなくクラスターそのものの数について，コミット数との比較を行った．1 つのクラスターは，同じユーザ（開発者）が参加しがちな 1 つのまとまった話題を表す．本手法では，どのスレッドとも結びついていない要素数 1 のクラスターもクラスター数として数えている．よって，クラスターの数が少ないほどそのプロジェクトにおいて開発者の共起が発生していると言える．月毎のクラスター数の推移を**図 7** に示す．仮説に従えば，クラスター数を示す折れ線が高い位置にある月ほど共起性が低く，アクティビティが高いことを示す．

我々は，2004 年 11 月から 12 月にかけてのコミット数の急激な上昇,及び 2005 年 5 月から 6 月にかけてのコミット数の急激な減少に着目した．クラスター数の推移を見ると，2004 年 11 月はクラスター数が前月から急上昇している．つまり，プロジェクトのアクティビティが高くなっており，開発が軌道に乗った状態であると考えられる．次月に以降におけるコミット数の増加は，アクティビティ上昇の結果であると言える．しかし,その後 2005 年 2 月に向けてグラフは下降傾向を示している.その後 2005 年 4 月に再び急上昇するが，コミット数に急激な変化のある 2005 年 6 月からは二ヶ月もずれており，アクティビティの予想に用いることはできない．

我々はこの結果に対し，クラスター数に影響を与え

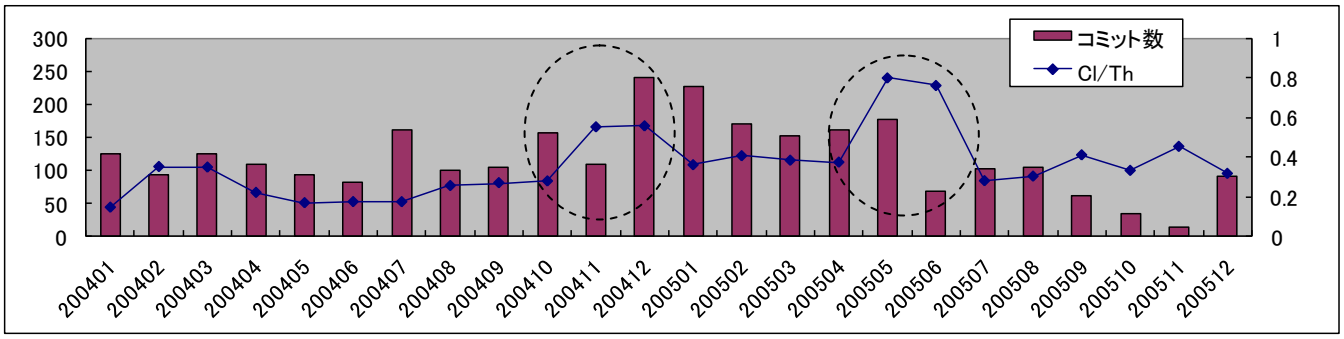

図 8. クラスター形成率とコミット数の推移

ると考えられる月毎のユニークユーザ数（メール送信者数）に着目した．ユニークユーザが多い場合，各スレッドを表すベクトルの次元数が増えることとなり，より細かくクラスターが分類され易くなる．対象プロジェクトの 2004 年から 2005 年にかけての，ユニークユーザ数の推移を図 6 中の折れ線に示す．2005 年 4 月にユニークユーザ数が高い値を示しており，図 7 の 2005 年 4 月におけるグラフ変動の原因になっていると考えられる．我々は，月毎のユニークユーザ数の違いがクラスター数に与える影響を抑えるために，その月におけるクラスター数をその月のスレッド総数で割り，正規化を行った．本稿では正規化後の値をクラスター形成率と呼ぶ．結果を**図 8** に示す．

単純にクラスター数を見た図 7 と比べて，ユニークユーザ数の影響が抑えられており，より適切に開発者の共起性を表していると言える．このグラフにおいて，急変動のあった 2004 年 12 月，及び 2005 年 6 月の前月を見てみると，いずれもクラスター形成率が急激に上昇していることがわかる．とりわけ，2005 年 4 月から 7 月にかけての推移は特徴的であり，プロジェクトになんらかの変化が起こっていると考えられる．該当期間について対象プロジェクトを精査したところ，2005 年 6 月以降に Kaffe プロジェクトの開発メンバーの多くが他プロジェクトに移動していたことが明らかになった．

5. まとめと今後の課題

我々は，OSS プロジェクトのアクティビティを計る手法として，開発者の共起に基づくクラスター分析法を提案した．また，適用実験を行い，開発者の共起性がプロジェクトに与える影響を計測した．実験結果のまとめは以下の通りである．

- クラスターの要素数に関する統計値は，メールの件数に影響を受ける為，アクティビティの予測に用いることはできない．
- クラスターの数は計測期間におけるユニークユーザ数に影響を受けるため，そのままアクティビティの予測に用いることはできない．
- クラスター形成率は，プロジェクトになんらかの変化が起こる際に急激な上昇を見せる．

以上の結果から，クラスター形成率により計測された開発者の共起性が，プロジェクトのアクティビティに影響を及ぼすことが確認できた．しかしながら，今回行った分析は巨大なデータセットの一部分についてのみであり，全ての期間において分析を行ったわけではない．また，対象としたプロジェクトも 1 つの OSS プロジェクトのみである．今後は分析の対象，及び対象期間を広げた上で，統計的手法によって各値の有意差を検定し，定量的な分析を行っていくことが必要である．

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] Christian Bird, Alex Gourley, Prem Devanbu, Michael Gertz, Anand Swaminathan, "Mining email social networks", Proceedings of the 2006 international workshop on Mining software repositories, pp.137-143, May 2006.

[2] Noriko Hanakawa, Kimiharu Okura, "A project management support tool using communication for agile software development", Proceedings of the 11th Asia-Pacific Software Engineering Conference, pp.316-323, Dec 2004.

[3] Peter C. Rigby, Ahmed E. Hassan, "What Can OSS Mailing Lists Tell Us? A Preliminary Psychometric Text Analysis of the Apache Developer Mailing List", Proceedings of the 2007 international workshop on Mining software repositories, pp.23-26, May 2007.

[4] G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing", Communications of the ACM, 18(11), pp.613-620, 1975.

[5] Kaffe Project http://www.kaffe.org/