

Latent Social Structure in Open Source Projects

オープンソースプロジェクトにおける潜在的な社会構造

Christian Bird, David Pattison, Raissa D'Souza, Vladimir Filkov and Premkumar Devanbu
Dept. of Computer Science, Kemper Hall, University of California, Davis, CA, USA,
SIGSOFT 2008/FSE 16, November 9–15, Atlanta, Georgia, USA

背景

大規模なOSSプロジェクトはエンジニアリングを妨げるような組織的なハードルにどのように対処するのか？

Apach Webserver, Apache Ant, Python, Perl, PostgreSQL
プロジェクトの参加者コミュニティを研究

開発における議論はすべて開発者のメーリングリストで行われ、メーリングリストはタスクに重点を置かれている

システム全体についてではなく**トピック**に分かれて議論される

これを分析して組織構造を発見する

背景

図1はコミュニティ構造を持つネットワークの例で、
これとモジュール性を用いてコミュニティの強度を定量化する
開発者は自発的にサブコミュニティを形成し、
そのグループの内部の人々と集中的に通信する

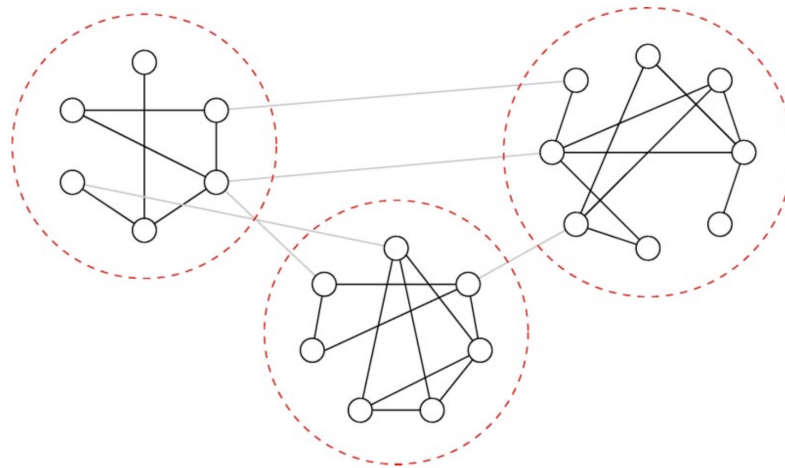


Figure 1: A network with strong community structure. *Modularity*, the measure of strength of community structure, which ranges from 0 to 1, has a value of 0.493 for the given division of nodes in this graph.

背景

**H1：参加者のサブコミュニティは大規模なOSSプロジェクトの
電子メールネットワークに形成されて、モジュール性のレベルは
統計的に重要になる**

統計的に重要というのは、OSSで観察されたモジュール性は、
グループの参加者が同じように社交的な人をえらんで
ネットワークに追加しているから

ランダムに選んだならそのようなモジュール性は現れないと考えられる

背景

H2：製品関連のディスカッションから構築されたソーシャルネットワークは、非製品関連のディスカッションまたはすべてのディスカッションに関連するものよりモジュール化される

メーリングリストでの議論には2種類があり、

「開発プロセスに関して」か「製品の開発アクティビティに関して」か

主に実際の開発（関数インターフェース, API, バグ修正, 機能実装など）の議論がグループ化の目的である

背景

**H3：同じサブコミュニティ内の開発者ペアには、
異なるサブコミュニティの開発者ペアより多くの共通ファイルがある**

サブコミュニティは、ソフトウェアエンジニアリング活動に関連付けられるはずである

サブコミュニティはメインのコミュニティより開発アクティビティを重要視したコミュニティになるため、直接関係するような同じ領域で作業する可能性が高いため

背景

**H4：同じサブコミュニティの開発者によってコミットされた
ファイル間の平均ディレクトリ距離は、異なるサブコミュニティを
描いた同じサイズの開発者グループよりも短くなる**

サブコミュニティは一貫したタスクと何らかの形で結びついているはず

特定の目標のために形成されたサブコミュニティの議論や開発に
焦点を当てる指標として、**平均ディレクトリ距離**を用いる

これらの仮説に答えるための方法とプロセスを提示し、
分析の結果を示し結果を示す

目的

- ・ OSSプロジェクトの社会構造は、**実際には組織化されておらず、すべての人にとって自由なものなのか？**
- ・ 従来の商用ソフトウェア開発の社会構造とは**対照的なのか？**
- ・ OSSプロジェクトには**独自の潜在的な社会構造があるのか？**
- ・ **自発的に形成および進化するサブグループはあるのか？**

OSSプロジェクトの潜在的な社会構造の実証的研究を行い、
これらの問題だけを議論する

研究方法と分析

Step1 : 対象とする**プロジェクト**を選択

Step2 : 各プロジェクトの開発者メーリングリストアーカイブと
ソースコードリポジトリを**マイニング**

Step3 : メーリングリストメッセージをフィルタリングし,
3か月の間隔で参加者のソーシャルネットワークを作成

Step4 : 各ソーシャルネットワークの**コミュニティ構造を計算**

Step5 : マイニングされたソースコード開発データを使用して,
参加者の部門の関連性を定量的・定性的に評価

研究方法と分析

Step1：プロジェクトの選択

選んだ5つのプロジェクトすべてが大規模で複雑なため、
1人の人間がシステムのすべての部分を担当することは困難で、
サブコミュニティの形成を促進すると考えられる

それぞれ異なったガバナンス体制であることから、
多様性があり外部妥当性が高められる

Name	Apache	Ant	Python	Perl	PostgreSQL
Begin Date	1995-02-27	2000-01-12	1999-04-21	1999-03-01	1998-01-03
End Date	2005-07-13	2006-08-31	2006-07-27	2007-06-20	2007-03-01
Messages	101250	73157	66541	112514	132698
List Participants	2017	1960	1329	3621	3607
Files	1092	7682	4290	13308	6083
Developers	57	40	92	25	29
Commits	28517	58254	48318	92502	111847

Table 1: Information on the data gathered for the projects studied.

研究方法と分析

Step2：データのマイニング

公開メールアーカイブから電子メールの日付, 本文, 送信者の名前とメールアドレス, メッセージIDヘッダー, 返信先ヘッダーを抽出

2者間の通信を示し, ソーシャルネットワークで2者の間にリンクを作成

プロジェクトソースコードレポジトリから作成者, コミット時間, ファイル名, 各ファイルの内容などのコード情報を抽出

どの開発者がどのファイルで共同作業をしていたかを確認

研究方法と分析

Step3：コミュニティ構造の発見

目標は、ネットワークをノードのグループに分割し、グループ内の接続が密集し、グループ間の接続が疎になるようにすること

Newmanアルゴリズムを用いてモジュール性の尺度を算出し、ネットワークのモジュール性を最大化

Step4：コミュニティ構造の検証

ランダムグラフ理論を用いて、ネットワーク内の人々が同じようにアクティブであることを想定してランダム化したグラフと、同じ次数分布を持つランダムグラフとを比較し、有意性を判断

研究方法と分析

Step5：メッセージのフィルタリング

各プロジェクトにおける中心的な人物（プロジェクトリーダーや創設者や初期メンバー）は、様々なサブコミュニティ間で活動しているため、その**中心人物達は対象から削除**

また、ソースコード分析から、各メッセージを「**プロセスに関してのもの**」か「**製品に関してのもの**」かを分類した

研究方法と分析

Step5：コミュニケーションと連携

同じサブコミュニティに所属する開発者同士は、異なるサブコミュニティ間より深い関わりをもつことを確認するため、サブコミュニティ内およびサブコミュニティ間の開発者の繋がりの平均レベルを調べる

2サンプルのウィルコクソン検定を使用して**母集団間の平均の差**を検定する

研究方法と分析

Step5：サブコミュニティのタスクフォーカス

同じサブコミュニティ内の開発者が同じ領域で作業する可能性が高いことを検証するため、アクティビティの範囲を定量化

各サブコミュニティ内の開発者がコミットしたファイルペア間のディレクトリツリーの平均距離を調べる

また、参加者のサブコミュニティのコミュニケーションと開発活動を検査し、まとまったタスクか集中的なタスクがあるかどうかを評価

研究結果

様々なプロジェクトのモジュール性測定値Qの値

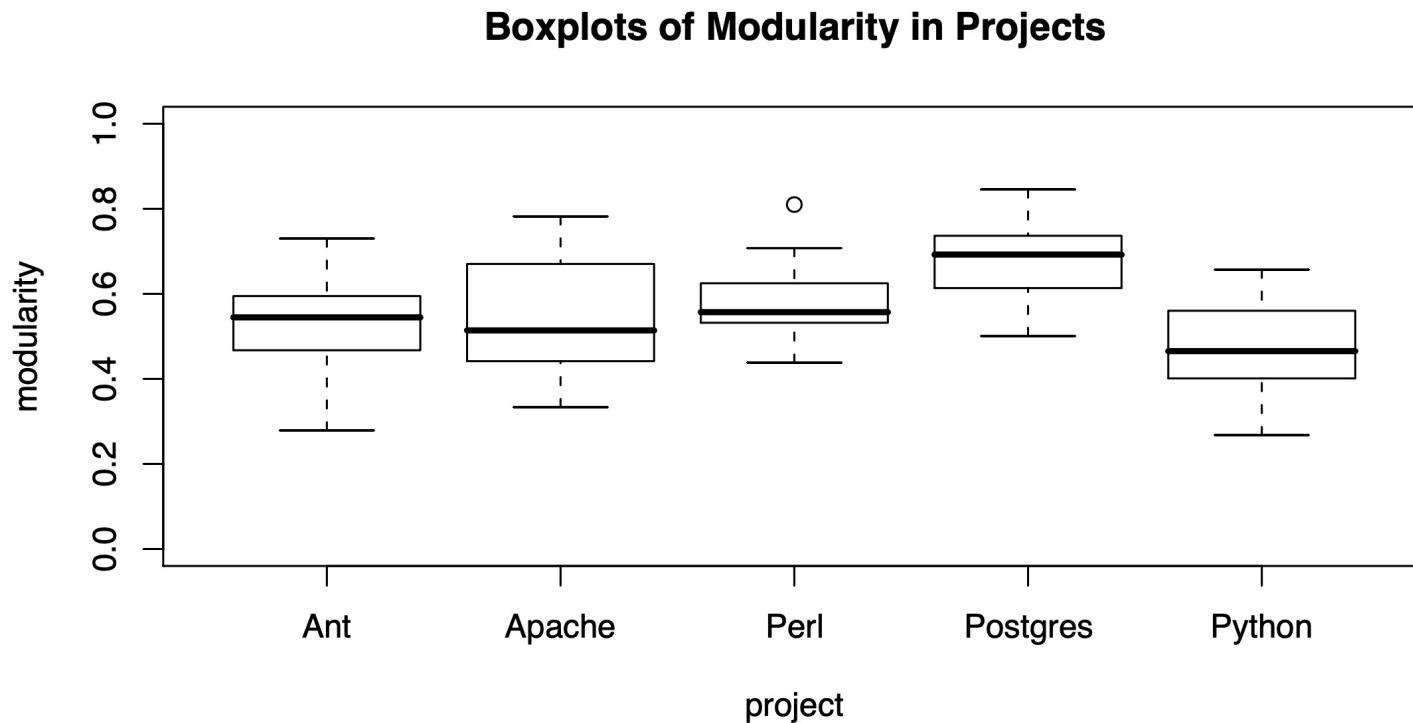


Figure 2: Boxplots of the strength of community structure for the various projects studied.

研究結果

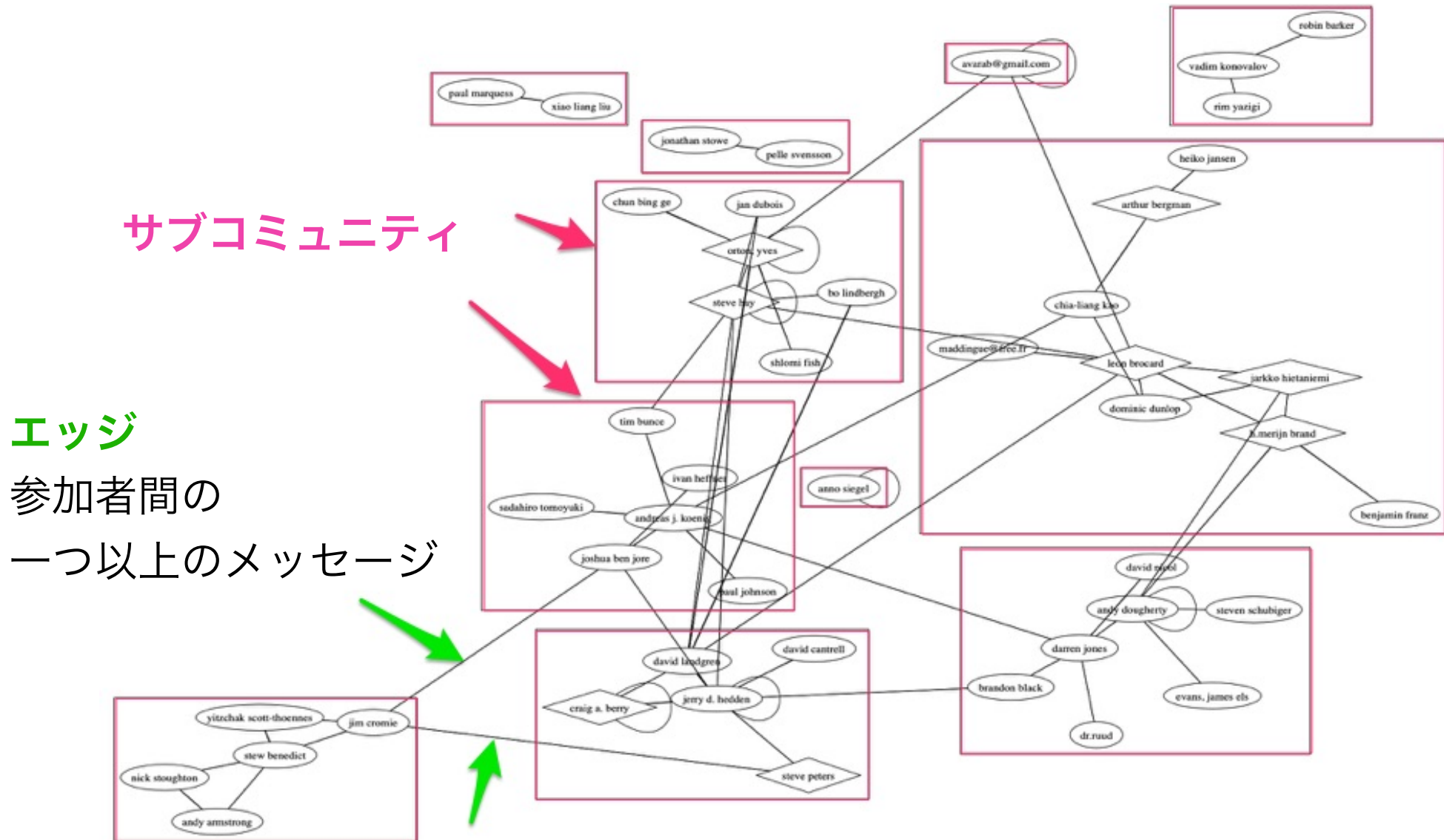


Figure 3: The community structure of Perl from April to June 2007. Diamonds indicate actual developers. Edge weights are not depicted.

研究結果

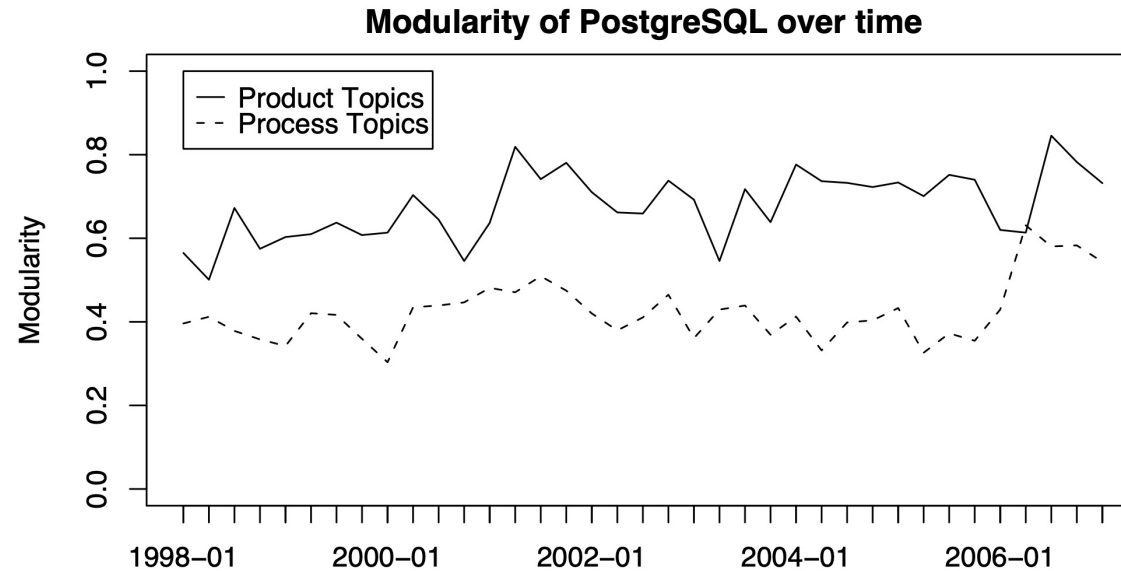


Figure 5: The difference in strength of community structure in the PostgreSQL project over time when filtering on messages that include product-related terms.

自然発生的なネットワークでは、
0.3以上のモジュール性の値が強力なコミュニティ構造を示している

メッセージをフィルタリングする前と後の両方で
この範囲の値が見つかった

研究結果

H1：参加者のサブコミュニティは大規模なOSSプロジェクトの電子メールネットワークに形成されて、モジュール性のレベルは統計的に重要になる

Ant, April to June of 2006

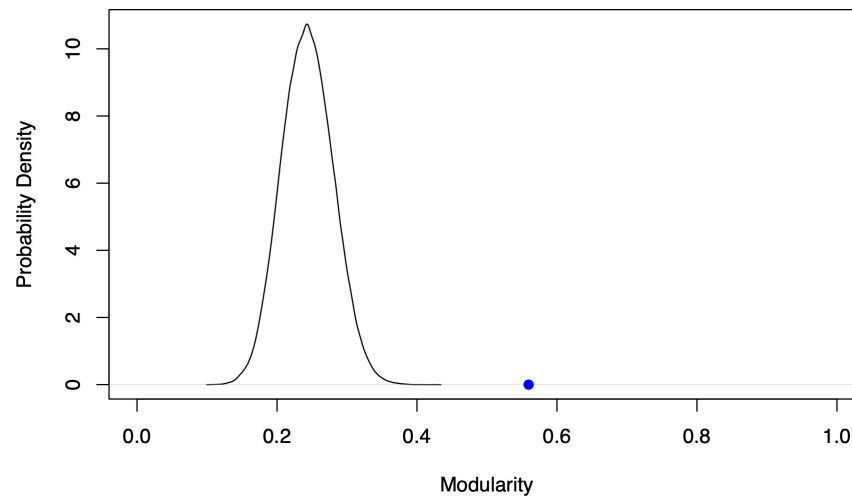


Figure 4: The distribution of modularity values for 100,000 random graphs with the same degree distribution as the observed network. The point represents the actual observed value.

観察されたモジュール性の重要性について

曲線：ランダムネットワークから得られたモジュール性の値の分布

点：観測されたネットワーク

研究結果

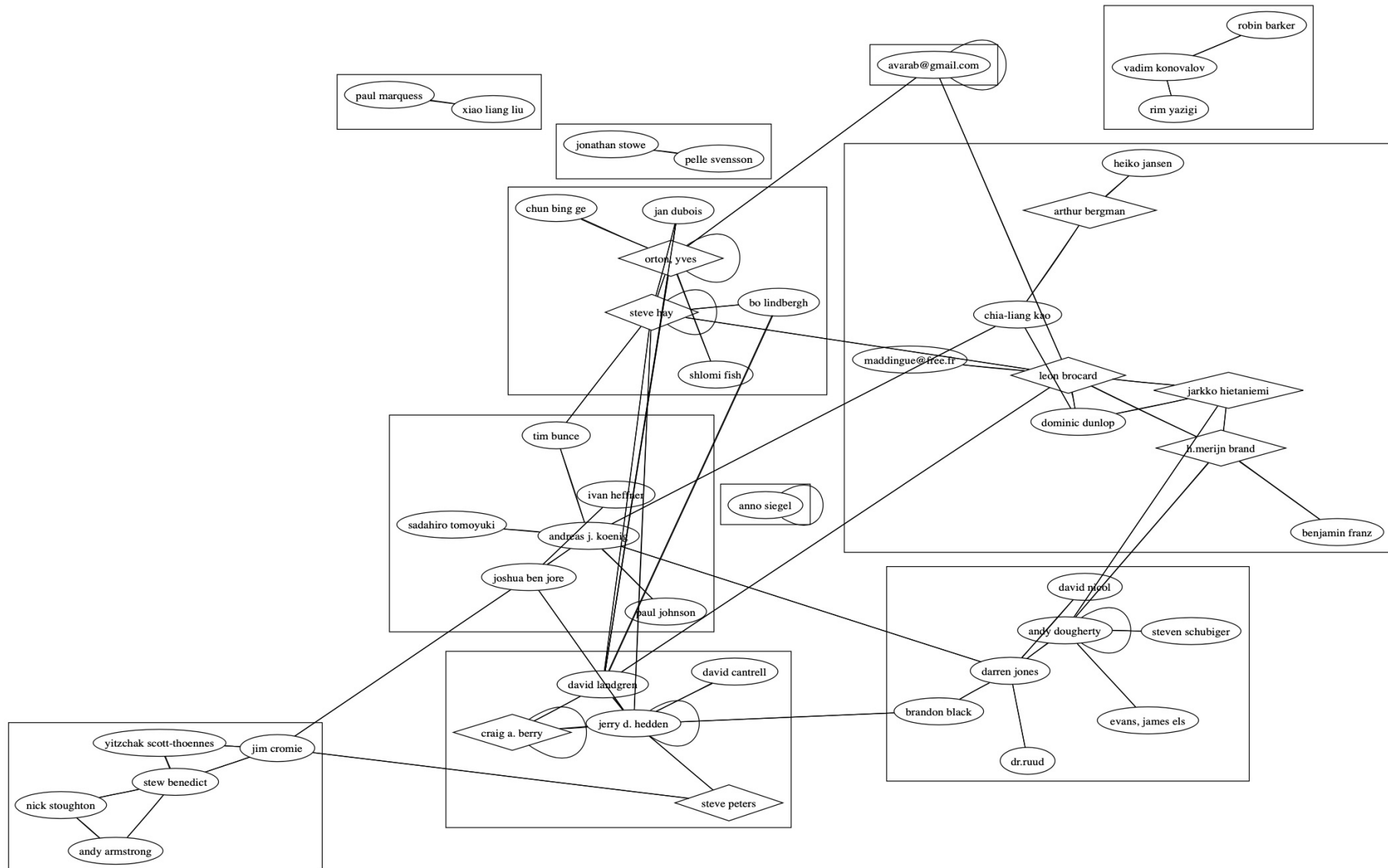


Figure 3: The community structure of Perl from April to June 2007. Diamonds indicate actual developers. Edge weights are not depicted.

研究結果

H2：製品関連のディスカッションから構築されたソーシャルネットワークは、非製品関連のディスカッションまたはすべてのディスカッションに関連するものよりモジュール化される

Name	Apache	Ant	Python	Perl	Postgres
<i>Product</i>	0.548	0.534	0.473	0.567	0.679
<i>Process</i>	0.337	0.485	0.312	0.423	0.425
All	0.325	0.459	0.293	0.400	0.420
P-val	0.001	0.033	0.001	0.001	0.001
% <i>Product</i>	27.6	64.3	50.0	29.9	26.4

Table 2: Means of the modularity when examining only *product* emails, only *process* emails, or all emails. P-val represents the statistical significance of a paired Wilcoxon test of *product* and *process* populations per project. The bottom row is the proportion of messages (as a %) that relate to product topics

製品トピックの電子メールのみを含めた場合のモジュール性の平均的な増加

結果は統計的に有意であり、すべてのケースでp値が0.05(5%)未満

研究結果

H3 : 同じサブコミュニティ内の開発者ペアには,
異なるサブコミュニティの開発者ペアよりも多くの共通ファイルがある

Project	Ant	Apache	Perl	Postgres	Python
Wilcoxon					
P-val	0.000	0.052	0.502	0.000	0.000

Table 3: Probability values for non-parametric tests of difference in means and difference in distributions of co-commits of developers between subcommunities and within subcommunities corrected for multiple hypothesis testing.

サブコミュニティ間および内の開発者間の連携度合いの検討

開発者が共有しているファイルの平均数を測定

開発者は、平均して他の人々よりも頻繁に
自分のサブコミュニティの人々と作業していることがわかった

研究結果

H4：同じサブコミュニティの開発者によってコミットされた
ファイル間の平均ディレクトリ距離は、異なるサブコミュニティを
描いた同じサイズの開発者グループよりも短くなる

ディレクトリ距離分析を行ったが、**帰無仮説は棄却できなかった**

開発者のサブコミュニティからの**ファイル平均距離**について、
全体的に傾向は一貫しなかった

サブコミュニティの開発者の議論とコミットの振る舞いのト
ピックを理解しようとする事例研究を実施した

研究結果

サブコミュニティは3つのタイプに分類できた

1. Apacheのサブコミュニティ
2. Pythonのサブコミュニティ
3. その他のサブコミュニティ

研究結果

1. Apacheのサブコミュニティ

議論と開発がコードベースの1つに焦点を当てていた

参加者のグループは**単一のトピックまたはタスク**に焦点を合わせていた

研究結果

2. Pythonのサブコミュニティ

開発動作の議論と内容の両方に明確な焦点を置いていたが、
変更されたファイルの場所はディレクトリ構造を横断していた

機能が横断的であれば広範なファイルに影響を与えるが、
それを取り巻く議論はまとまりがあった

研究結果

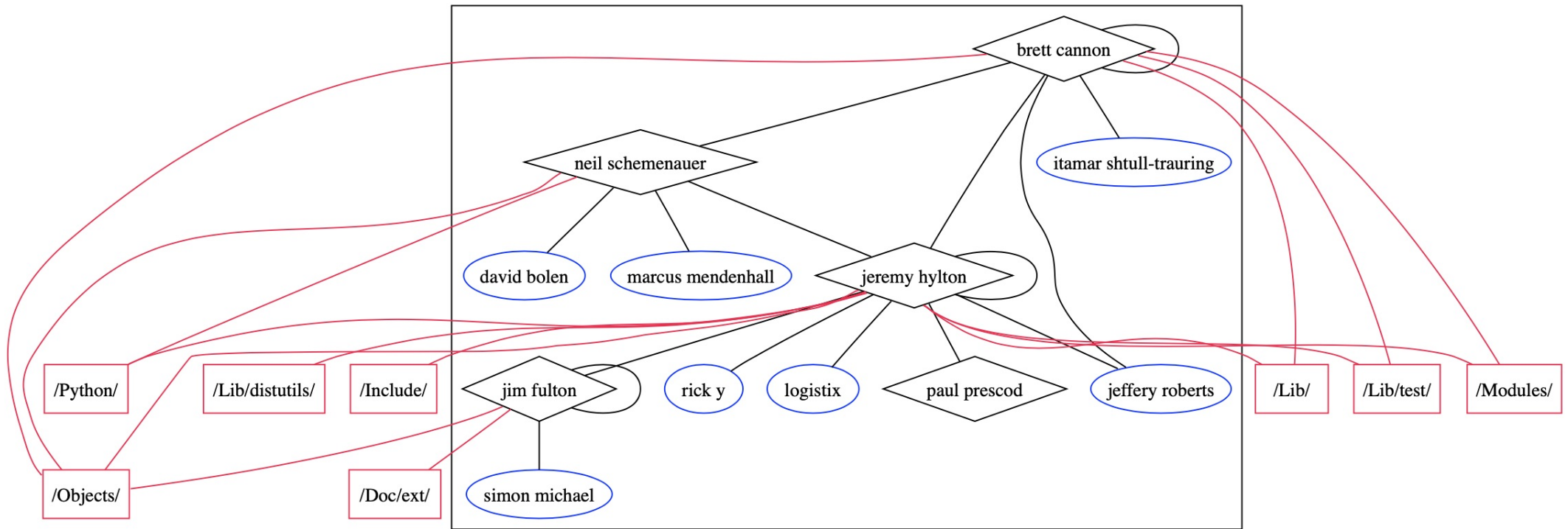


Figure 6: One subcommunity of participants in the python community from April to June 2003. Diamonds are developers, ovals are participants, and rectangles are directories committed to (in lieu of the large number of files committed to).

研究結果

3. その他のサブコミュニティ

議論または開発中の特定のトピックまたはタスクが1つ以上あった

コードベースの2つ以上の異なる領域で作業し, 2つのコミュニティを結び付けた

明確なトピックを区別できないいくつかのケースもあった

研究結果

H4：同じサブコミュニティの開発者によってコミットされた
ファイル間の平均ディレクトリ距離は、異なるサブコミュニティを
描いた同じサイズの開発者グループよりも短くなる

結論として、結合タスクに関するサブコミュニティの焦点に関する
仮説4は複雑

焦点タスクが分野を横断していて、**自動的にマイニングすることが
難しいため、現在調査中**

有効性への脅威

外部の有効性に対しての脅威

OSSプロジェクトから**ランダムに抽出して行った研究ではないため**, 抽出基準に沿っていないプロジェクトで発生する結果と似ているかもしれないが, それを確かにする証拠はない

また, 同じ基準に適合する他のプロジェクトに関しても一般化しない場合がある

結論

5つの大規模なOSSプロジェクトをマイニングして分析した結果、
すべてのケースにおいて参加者のコミュニケーションパターンに
強力なコミュニティ構造が見られた

ソースコードの成果物に直接焦点を合わせた場合、
コミュニティ構造はよりモジュール化されていることがわかった

一部のケーススタディでは、サブグループのタスクフォーカスは多くの
場合に存在したが、定量的分析においては決定的なものにならなかった