Abstract

In this paper, we analyze the mechanism of “open-source development”, in which anybody can join and leave to the collaboration to develop complex software system, as implausible phenomena. Here, we take the case of “Linux”, which is an operating system developed as open-source software. In existing studies, the management of the community is often discussed, but we focus on how the community was emerged. In this paper, we apply social system theory proposed by Niklas Luhmann, especially the concept of “double contingency”, “the nexus of communication”, “Openness and Closeness” of a system, “structural coupling” of systems, and “communication media”. In order to clarify the nexus of communication for developing Linux”, we analyze the logged text of mailing lists and newsgroups.

［キーワード］ オープンソース、Linux、社会システム理論、二重の偶発性、構造的カップリング

オープン・コラボレーションのメカニズム: オープンソース開発再考

Social System Analysis of Open Collaboration: Reconsidering Open Source Development

井庭　崇／*Takashi Iba*（慶應義塾大学総合政策学部 専任講師）

1. はじめに

1990年夏、フィンランドにあるヘルシンキ大学の大学院生リーナス・トーヴァルズは、新しいOS（Operating System）の中核部分（カーネル）を開発し始めた。後に「Linux」と呼ばれるそのOSは、インターネット上で多くの人が連携して開発を行うという、オープンなコラボレーションへと発展していった。Linuxの開発は、自発的な参加によって成り立っており、現在までに数千人が参加したといわれている[1]。この事例が興味深いのは、参加している開発者たちが、ふつうのソフトウェア開発のように同じ組織に所属しているというわけではなく、会ったこともないままコラボレーションを行ってきたということである。開発者コミュニティの境界は、組織の境界としてあらかじめ規定されているのではなく、実際に開発に携わった段階で事実上メンバーになっていくというかたちで規定されていく（図1）。つまり、開発のコミュニケーションが連鎖しながら、絶えず開発コミュニティというものを再生産し続けているのである。一見脆弱に見えるこの組織化原理のもと、Linuxの開発は着実に前へ前へと進んでいき、複雑なソフトウェア[1)]をつくるということに成功しているのである。

このような絶妙なバランスの上に成り立つ動的な秩序は、誰かによって綿密な計画のもとデザイン・管理されたものではない。この秩序は、自生的に生み出されたものであり、その意味で創発的な秩序なのである。本来であれば成り立たちそうにないこの秩序は、いかにして形成さてきたのか――それが本論文の主題である。本論文では、これまで多くの先行研究[2)]が取り上げてきたような、安定的に運営されている段階での「運営管理方法」に注目するのではなく、その秩序形成の最初期に焦点をあて、それが成立するに至った過程を分析する。その際、分析の枠組として、ニクラス・ルーマンの社会システム理論を援

用する。この理論は、コミュニケーションに着目して組織化原理を捉えるため、まさに本論文の目的に合致している。その理論のなかでも、特に「二重の偶発性」、「コミュニケーションの連鎖」、「システムの閉鎖性と開放性」、「構造的カップリング」、「コミュニケーションのためのメディア」の概念を用いて、オープン・コラボレーションの原理に迫っていきたい。さらに、Linux開発に関するメーリングリストとニュースグループのログから、実際にどのようなコミュニケーションの連鎖が起きていたのかを明らかにする。

2．二重の偶発性と社会形成

オープンソース開発は、通常、金銭的な報酬はなく、権力によるコントロールもない状況において、個人の自発的な参加によって進められている。人々が労力をかけてそのような開発作業を行うということは、本来なかなか起こりそうもないことである。Linuxの開発が、これだけ多くの開発者を巻き込むオープン・コラボレーションになろうとは、トーヴァルズ自身も予測していなかったという。しかし、それは実際に実現されているのである。いったいそれはどのようなメカニズムで実現されたのだろうか。この問いに答えるためには、「二重の偶発性」（ダブル・コンティンジェンシー）の問題について考える必要がある。

二重の偶発性は、ありそうにないことがどうしてあり得るようになったのかという問題、すなわち、「社会的な秩序はいかにして可能なのか」という問題と関係している［4］。二重の偶発性がある状況とは、相互行為の場面において、自分の行為が相手の行為に依存しているため、お互いに自らの行為を規定できないという状況のことをいう（図2）。この両すくみの状況は、論理的に決着不可能な循環構造になっており、それを打ち砕く解決策がなければ、行為をするということ自体が成立しないことになってしまう。この問題に取り組んだ社会学者タルコット・パーソンズは、この問題の解決に「共有された価値」というものを持ちだした[3)]。ある価値が共有されていれば、相手の行為の可能性を縮減することができ、行為を予測することが可能になる。しかし、そのような価値がどこから来るのか、いかに形成されるのかについては不明なままであり、これでは秩序形成の問題についてきちんと答えていないということになる。

これに対し、この問題に引き続き取り組んだ社会学者ニクラス・ルーマンは、その対称性を打ち破る「わずかな動き」に注目した。人は社会的に合意を取ってから初めて行動するのではなく、絶えず何らかの「動き」を行っている。その動きによって相手もなんらかの反応をし、そして自分もそこから影響を受ける。このように、ちょっとしたきっかけで対称的な相互依存の状況が、非対称の状況に変わっていくのである。もちろん、相互依存の状況から動き始めたとしても、依然としてお互いは不透明なままであり、完全に理解・予測することはできない。しかし、相手の行為・反応は、自分にとっての情報となり、それを期待形成の判断材料にすることができる。このようにして、相手の行動を「期待」することができるようになるのであり、逆に相手にとっても同様である。

かくして二重の偶発性の状況を超え、関係性が築かれるようになると、今度は「自分の行為が相手にどのように捉えられるのか」を想定するようになる。「相手があるコミュニケーションを受け取るのかそれとも拒否するのだろうか、あるいは行為に還元していえば、ある行為がこの相手には有益であるのかそれとも不利益になるのであろうか」［4］ということを考えることで、次の相手の行動に結びつきやすくなっていくのである。このような仕組みで、社会レベルの秩序形成がなされていくのである。

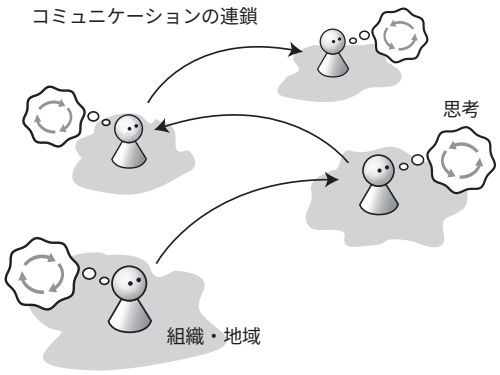

図-1　コミュニケーションの連鎖によるオープンソース開発

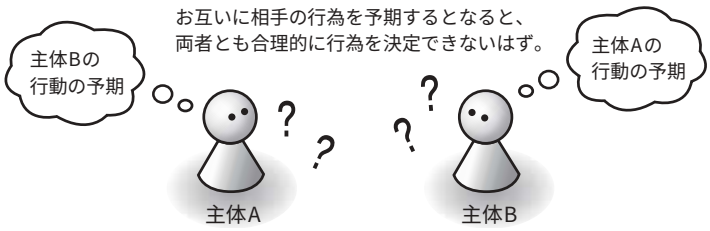

図-2　二重の偶発性（ダブル・コンティンジェンシー）の状況

3．Linux開発の最初期における二重の偶発性の克服

Linuxの事例でいうならば、二重の偶発性における主体は、トーヴァルズを含む潜在的開発者である。当時、Unixマシンは個人で買うには高価であり、教材として開発されたOS「Minix」も拡張についての制約があった。そのような状況において、実際に自分で新しいOSをつくるという大変な作業に着手したり、呼びかけたりするという人はほとんどいなかった。ところがあるとき、その対称性を打ち破る「わずかな動き」が起きた。リーナス・トーヴァルズが、自分のためにOSの中核部分（カーネル）をつくり始めたのである[4)]。

3.1　最初の動き

Linuxの開発は、社会的な意義や必要性からではなく、純粋に個人的な楽しみから始まっている。OSの開発の面白さを、トーヴァルズは次のように語る。「OSはコンピュータの中で起こるすべての基本原理となる。だから、OSを作るのは、最高にやり甲斐のあることだ。OSを作るというのは、世界を作ることだ」[6]という。しかし実際には、それは容易なことではない。OSを作るということは、200〜300あるシステムコールの一つ一つをプログラミングしていかなければならないからである。トーヴァルズは、マニュアルや本でシステムコールの機能を調べてはプログラムを書いていったのであるが、なかには何ヶ月もかかったものもあるという。「この作業には、まったくイライラさせられた。その理由は、何も起こらないからだ。なんの進歩も目にすることができないからだ」[6]といい、「やがて、ずらっと並んだシステムコールを読むのなんかやめて、こんな方法は投げ捨ててしまいたくなる」こともあったという。しかし地道に続けて行くと、なんとか形になりそうだと思うになったのである。

3.2　公開と反応

1991年7月3日に、トーヴァルズは、USENETのニュースグループcomp.os.minixに、UNIXのシステムコールの規格である「POSIX」（Portable Operating System Interface）について質問を投げかけた。残念ながらこの質問に直接答えてくれる返事は来なかったのだが、代わりに「素敵な返事」[6]をもらった。それは、ヘルシンキ工科大学の教育助手アリ・レムケが、FTPサイト（ftp.funet.fi）にLinux用のスペースをつくってくれたというのである[5)]（ただし、この時点ではトーヴァルズはまだコードを公開していなかったので、実際にこのサイトが使われたのは数ヶ月後のことである）。

1991年8月25日、トーヴァルズは再びcomp.os.minixに、「Minixに最も強く望むことは何ですか？」という投稿をした[6]。開発中のOSにどのような機能を追加すべきかを調査したかったからである。また、「bash1.08とGCC1.40を移植しましたが、うまく動いているようです。したがって、あと数ヶ月で実用に耐えうるものができるでしょう」ということも書き添えた。すると数時間後には、フィンランドのジルキ・クォパラから「もっと教えて！・・・移植に関して障害はありますか？」という反応が返ってきた。また、オーストラリアのピーター・ホルツァーからも、「僕はこのOSにとても興味を持っています。僕は以前に自分だけのOSを書こうと考えたことがあります。だけど、最初からすべてのものを書くだけの時間がないと考えました。しかし、赤ちゃんOSの成長を助ける時間は持てると思います:-)」という反応があった。

その後1991年9月17日、トーヴァルズはLinuxの「バージョン0.01」のソースとバイナリを、レムケの用意してくれたFTPサイトにアップした。「大々的に公表はしなかったが、ほんの一部の人には――全部で5人から10人くらいの人には、個人的なメールでftpサイトにアップしてあるとだけ知らせた。ミニックスでは有名なブルース・エバンズやアリ・レムケほか数人だ」[6]という。トーヴァルズは、まだニュースグループに公表できる段階ではないと考えていたのだが、なぜこの段階でアップしたのかということについては、「僕がこのサイトを持っていたので、何かをアップロードする必要があると感じていたんだ」[7]と当時の心境を語っている。この公開の後、「これができたら素敵ですね」というコメントや、「かっこいいけど、わたしのコンピュータではまったく動きません」というような報告も含めて、肯定的な反応がいくつか寄せられたという[6]。

1991年10月5日、comp.os.minixニュースグループに、Linuxのバージョン0.02のリリースと、それをダウンロードできるFTPサイトの情報を公開した。ここで初めて、Linuxが一般に公開されたことになる。この投稿は広告のような書き出しで始まっている。「人びとがだまって自分のデバイスドライバを書いていた時代、あのMINIX-1.1の楽しかった頃が恋しいですか？ 楽しいプロジェクトがなくて、ちょうど自分の必要に応じて修正できるOSの経験を積みたいと思っていませんか？ すべてがMINIX上で動いているとき、挫折というものを味わっていませんか？ 徹夜仕事してまで動かしたいぐらい、素晴らしいプログラムがないって？ この投稿はまさに君のためのものです:-)」。また、次のようなことも書かれていた。「このOSは、ある1人のハッカーによるハッカーたちのためのプログラムです。僕はこのことを楽しんでいるし、これを見て楽しんでいたり、自分のために修正していたりする人たちがいるでしょう。このOSは、理解したり、使ったり、修正するのにちょうどよいぐらいにまだ小さいです。僕はみなさんのコメントを楽しみにしています」。このとき、実際にどのくらいの人がダウンロードしたかはわからないが、トーヴァルズは「10～20名ぐらいの規模だと思う」[7]と推測している。

3.3 テイクオフ

トーヴァルズがコードを公開したのは、オープンソース開発を仕掛けたかったというわけではなく、口だけでなく、自分が本当にやったということを示したかったからだという[6]。しかも、この頃、トーヴァルズはこの開発を長く続けるとは考えていなかったという。「あのままなら、たぶん1991年の末にはやめてしまっていただろう。面白そうだと思えることは、たくさんやってしまっていた。何もかもが申し分なく動いたわけじゃないが、ソフトウェアってやつは、根本的な問題を解決してしまうと、急速に興味を失いやすくするものだとわかった。そのとき、ぼくにもそれが起こっていた。ソフトのデバッグはそんなに面白いものじゃない」[6]からである。

しかし、トーヴァルズが直面したいくつかの出来事によって、続けざるを得なくなったのである。それは、「一つは、うっかりしてミニックスの入っているパーティションを壊してしまったことだ。いま一つは、みんなが感想を送り続けてくれたことだ」[6]という。考えた結果、トーヴァルズはLinuxを開発し続けると

いう道を選んだ。そしてこの直後、トーヴァルズは他の人の期待に応えてある機能を開発し、喜ばれるということを経験する。それは実に「誇らしい気分だった」[6]という。これらの追加・修正を行ったバージョン0.12が、1991年1月5日にリリースされた。このとき追加した機能のおかげで、より多くの人がLinuxに興味を持ち始めることになる。1992年の「1月中に、リナックスのユーザーは5人から10人、20人になり(そのあたりまでが、ぼくがメールできる、名前のわかっている人たちだ)、さらには名前も知らぬ数百人へと増えていった。ぼくはリナックスを使っている全員を知っているわけじゃなかった。これはなかなか愉快だった」[6]という。1992年1月中旬には、Linux-Activistsメーリングリストには196名のメンバーが登録されており、多くの人が興味を持ち出したことから、もはや自分が飽きたからといってやめられるような状況ではなくなっていた。以上が、トーヴァルズの周辺でおきた、Linux開発の最初期における二重の偶発性の克服の物語である。

4．Linux開発におけるコミュニケーションの連鎖

4.1　メーリングリストとニュースグループにおけるコミュニケーションの連鎖

Linuxの開発が始まった1991年末、Linuxに関する議論・やりとりは、「Linux-Activists メーリングリスト」と「comp.os.minixニュースグループ」という2つの場で行われた。基本的には、Linux-Activists メーリングリストでは、実際に開発を行っている人たちがやりとりをしており、Minixユーザーが集うcomp.os.minixではLinuxの機能やMinixとの違いなどが議論された。これらのメーリングリストとニュースグループにおいて、1991年10月～1992年2月に投稿されたメール数／記事数の推移を表したのが、図3である[7]。

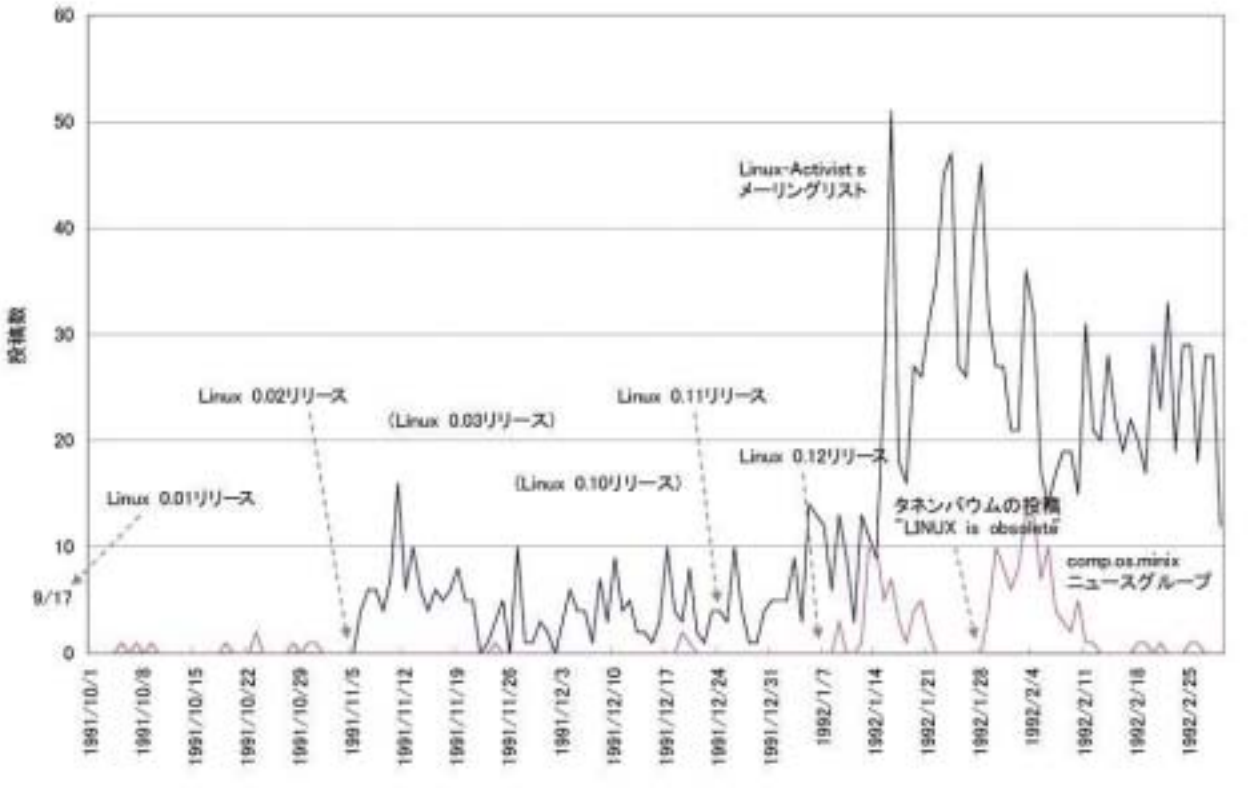

図-3　Linux-Activists メーリングリストとcomp.os.minixニュースグループにおけるLinux関連の投稿数の推移（1991年10月～1992年2月）

このメーリングリストの投稿者の人数や内容から、徐々にコミュニティが形成されていったということがうかがえる。また、Linux-Activistsメーリングリストに投稿されたメールの連鎖関係を可視化したものが、図4から図7である。ここでは、1991年11月6〜13日のやりとりを、2日ごとに分けて可視化している。横軸が時系列であり、縦に投稿者が並んでいる。メールのサブジェクト（件名）が返信になっている場合（「Re:〜」）、および本文で以前の内容を引用している場合に、それらの投稿同士を線で結んである[8]。これらの図をみると、実際に開発のコミュニケーションが連鎖している様子を理解することができる。

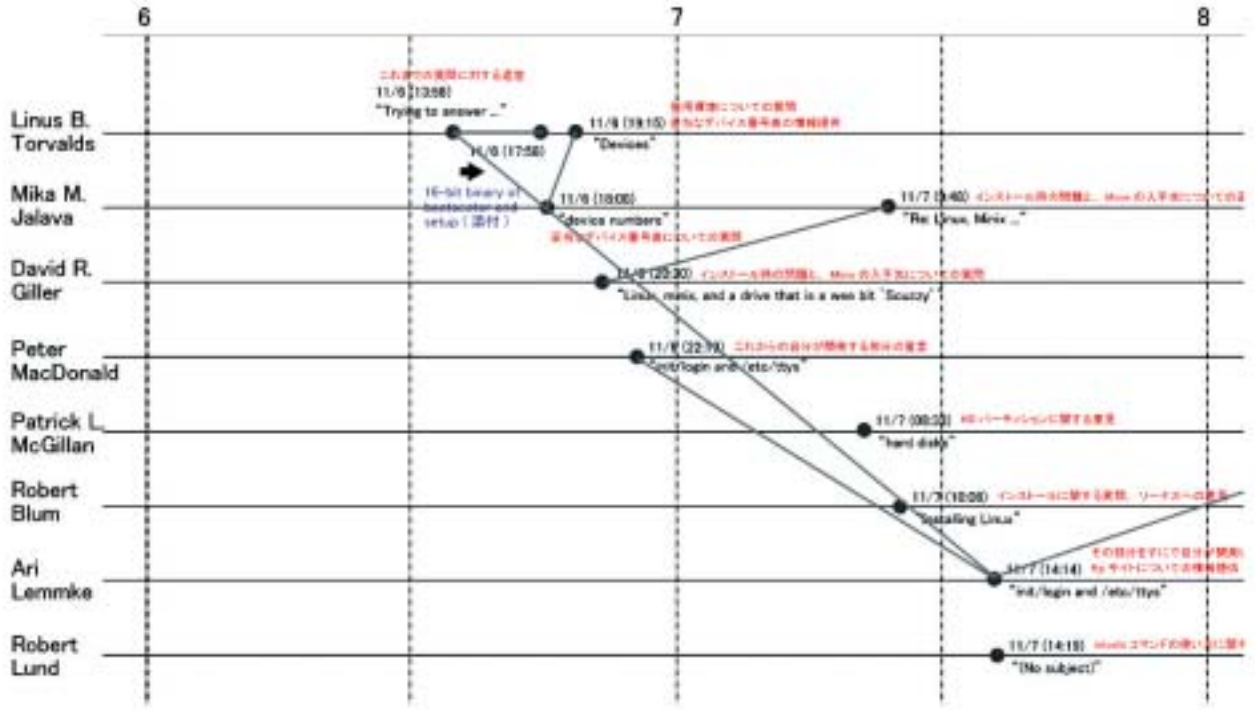

図-4　Linux-Activists メーリングリストにおけるコミュニケーションの連鎖（1991年11月6、7日）

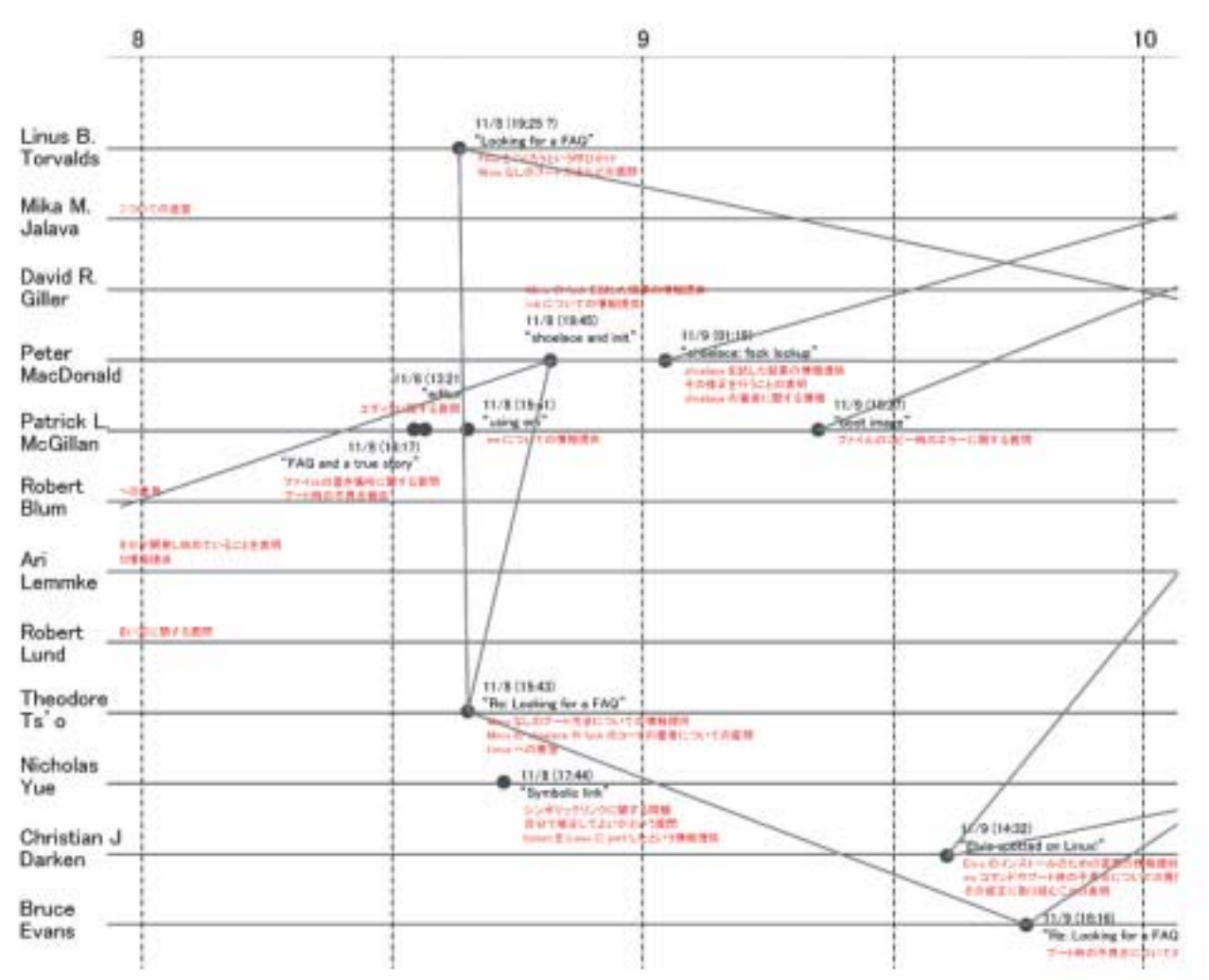

図-5　Linux-Activists メーリングリストにおけるコミュニケーションの連鎖（1991年11月8、9日）

図8は、Linux-Activists メーリングリストの投稿者とその投稿数をグラフ化したものである。縦軸は投稿者別の投稿数であり、横軸は投稿数が多い順に投稿者を並べている。これらのグラフをみると、各人の投稿数にかなりの偏りが生じていることがわかる。図9は、各月の投稿者別の投稿数とその順位の分布を両対数グラフに描いたものである。両対数グラフにおいて直線で近似できることから、べき乗分布であることが示唆される。投稿数がべき乗分布になるのは、ヘッド部分にいる人があらゆるレベルの話題に反応しているからだと思われる。つまり、ロングテール部分にいる人たちの質問にこまめに答えているということである。1991年12月は他の月よりも投稿数が少なく、べき乗分布から乖離していることから、メーリングリスト上でのこのコミュニティの組織化の力が弱かったと判断できる。

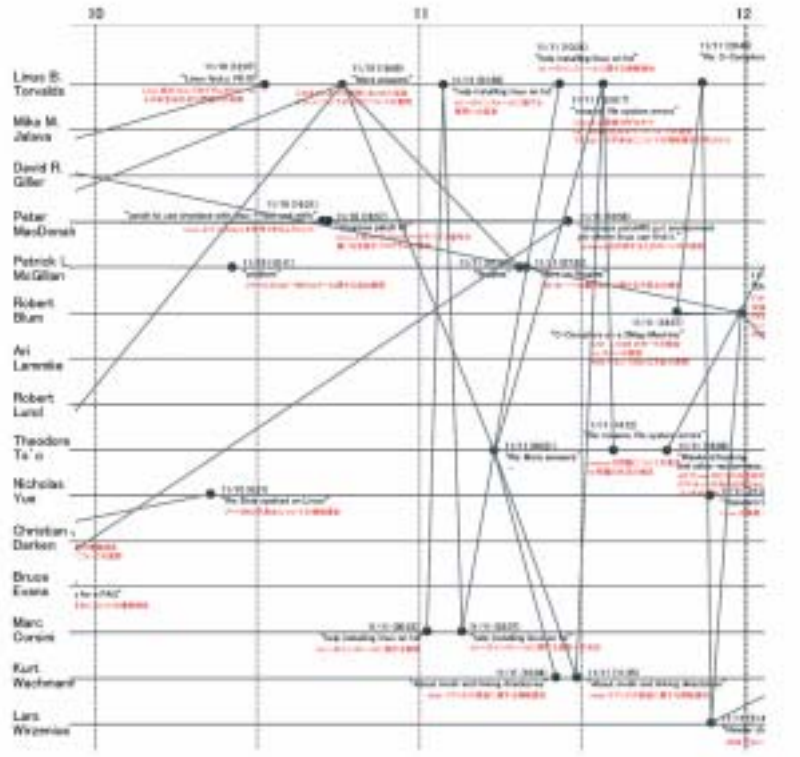

図-6　Linux-Activists メーリングリストにおけるコミュニケーションの連鎖（1991年11月10、11日）

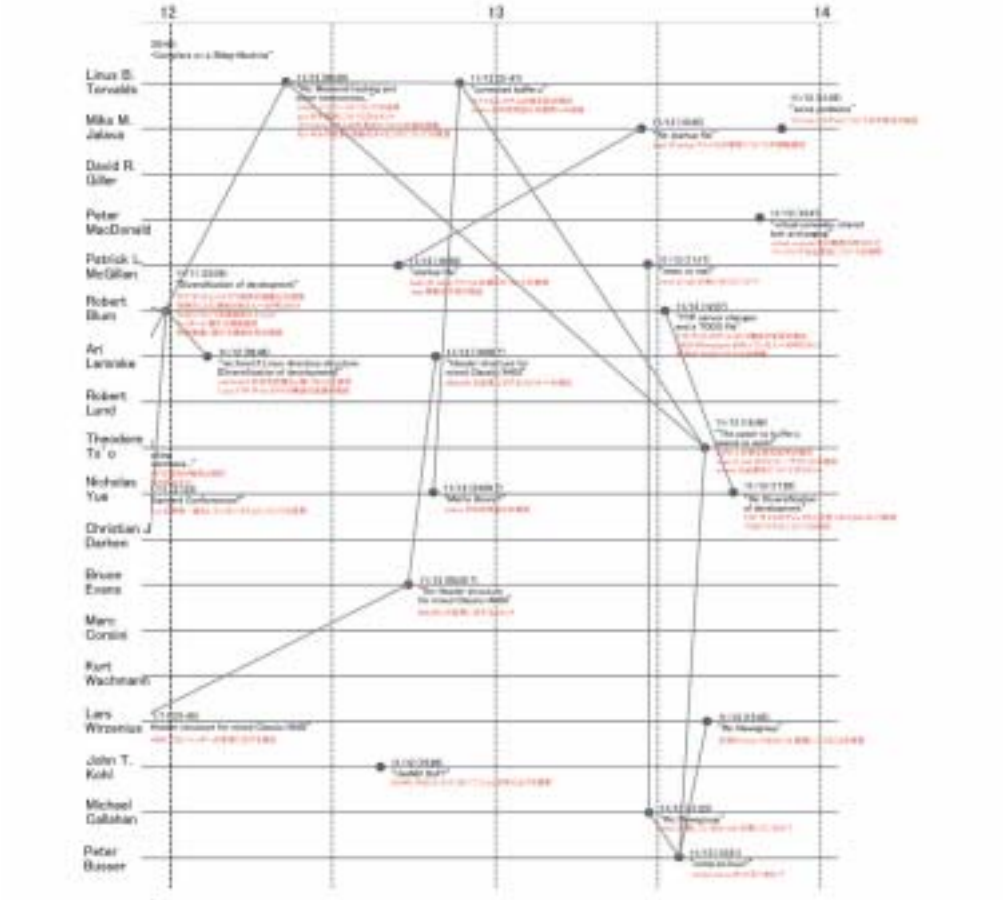

図-7　Linux-Activists メーリングリストにおけるコミュニケーションの連鎖（1991年11月12、13日）

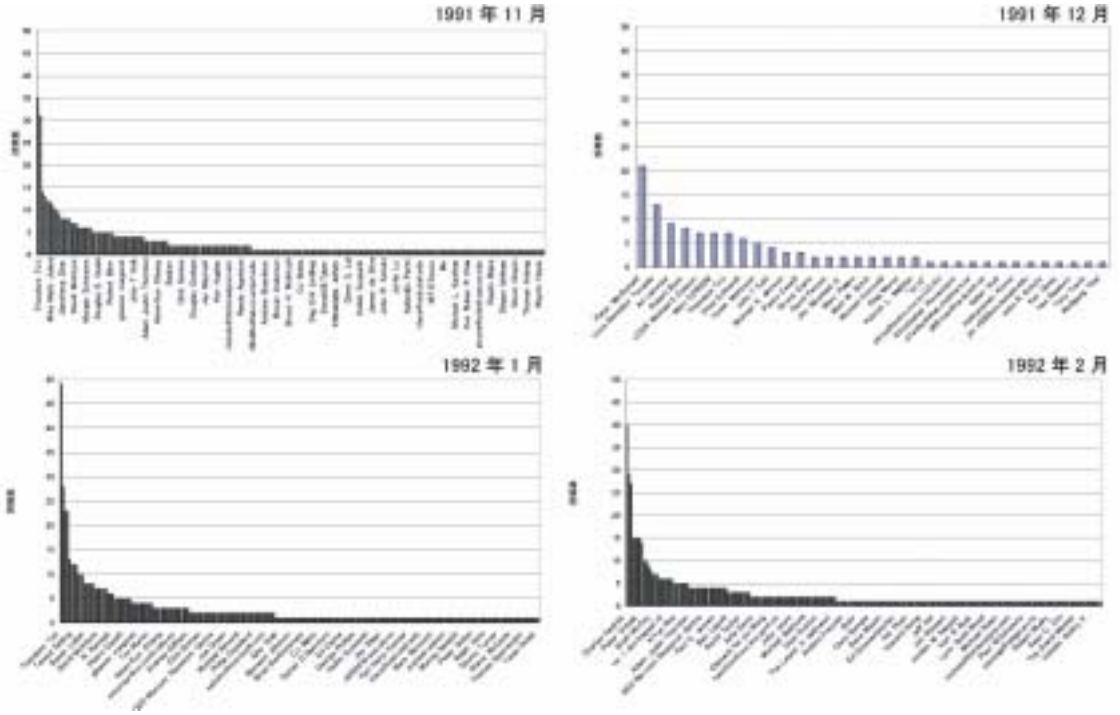

図-8　Linux-Activists メーリングリストにおける投稿者別の投稿数

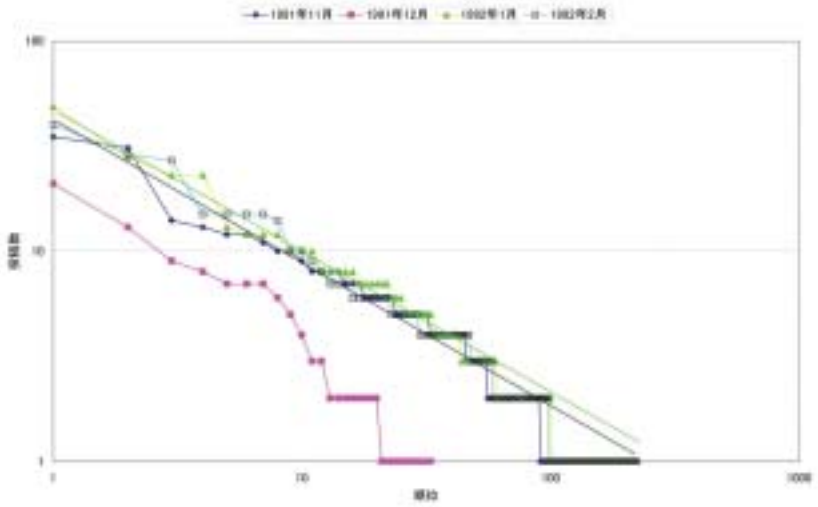

図-9　Linux-Activists メーリングリストにおける投稿者別の投稿数の分布（1991年11月～1992年2月）

そして、1991年8月、同年10月、1992年1月にcomp.os.minixニュースグループに投稿された記事の連鎖関係を可視化すると、図10、図11、図12のようになる。横軸が時系列、縦に投稿者が並んでおり、投稿記事の内容に関係があるもの同士を線で結んである。これらの図をみると、当初は一部の人だけが投稿・反応していたLinuxの話題に対し、多くの人が参加するようになったことがわかる。また、comp.os.minixニュースグループへのLinux関連記事の投稿者とその投稿数をグラフ化したものが、図13である。縦軸は投稿数であり、横軸はその投稿数が多い順に並べている。この投稿者別の投稿数と順位の分布をまとめると図14のようになる。縦軸は投稿数であり、横軸はその投稿数が多い順に並べた両対数グラフである。べき乗分布もしくは対数正規分布のようにみえるが、データ量が少ないこともあり、これだけの情報では分布についての判断は困難である。

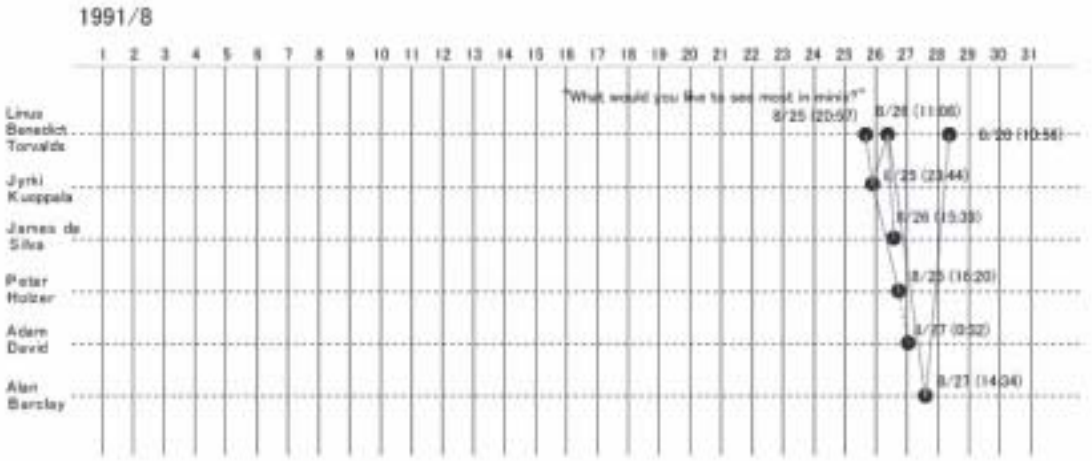

図-10　comp.os.minixニュースグループにおけるコミュニケーションの連鎖（1991年8月）

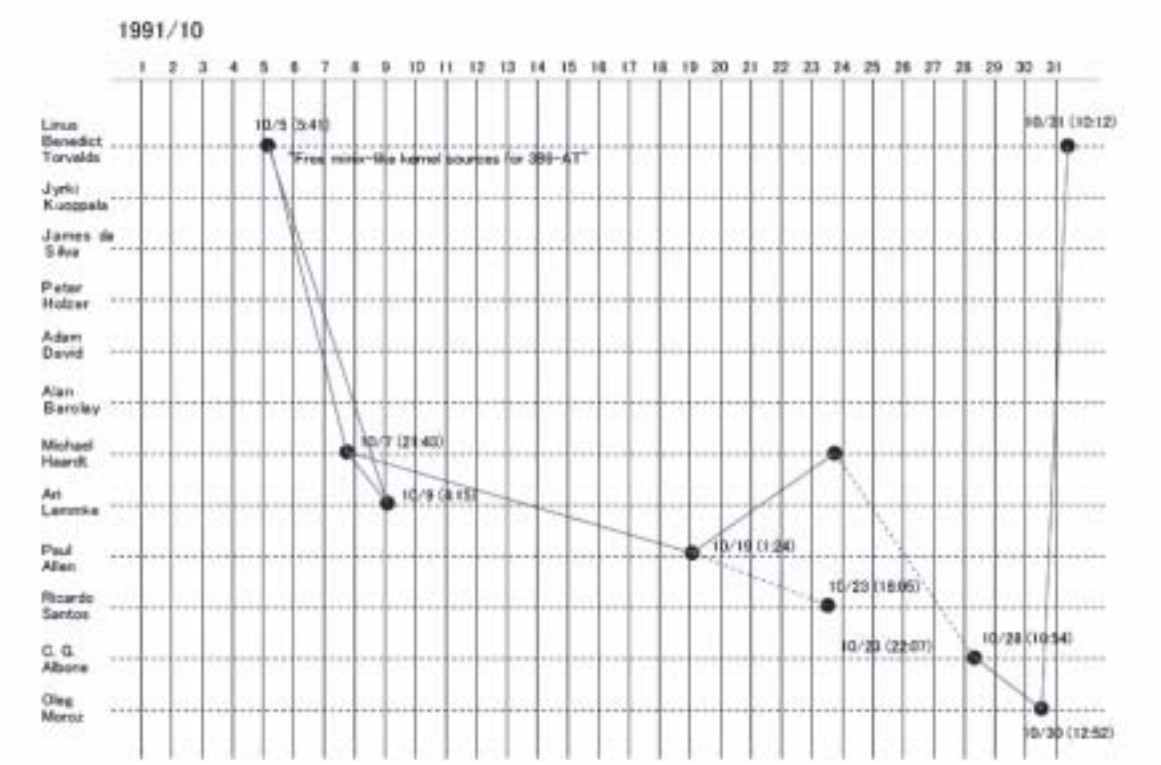

図-11　comp.os.minixニュースグループにおけるコミュニケーションの連鎖（1991年10月）

4.2 「質問」、「呼びかけ」、「不具合報告」によるコミュニケーションの誘発

本論文でこれまで示してきたように、Linux-Activistsメーリングリストとcomp.os.minixニュースグループでは、それぞれLinuxの開発にまつわるコミュニケーションの連鎖が起こっていた。これらのメール／記事の内容を具体的に読んで行くと、コミュニケーションの連鎖を誘発する要因がビルトインされていることがわかってくる。その要因とは、「質問」と「呼びかけ」、そして「不具合報告」である。

1つめの要因は、「質問」のコミュニケーションである。すでに述べたように、オープンソース開発によってつくられ、無料で手に入るLinuxは、二重の意味で人びとの出入りについて開放的である。一つは、誰でも開発者になれるということ、もう一つは誰でもユーザーになれるということである。このような開放

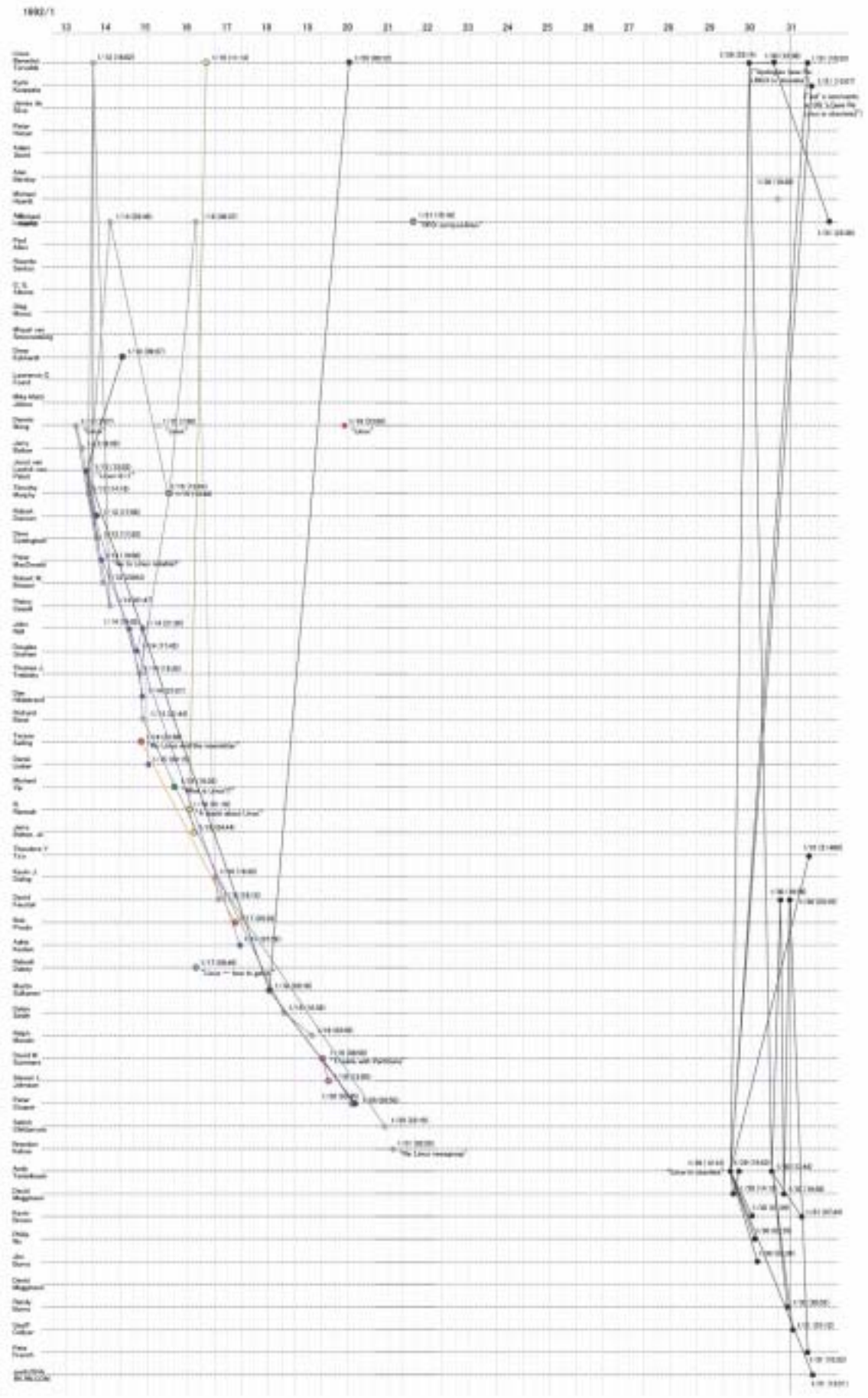

図-12　comp.os.minixニュースグループにおけるコミュニケーションの連鎖（1991年1月）

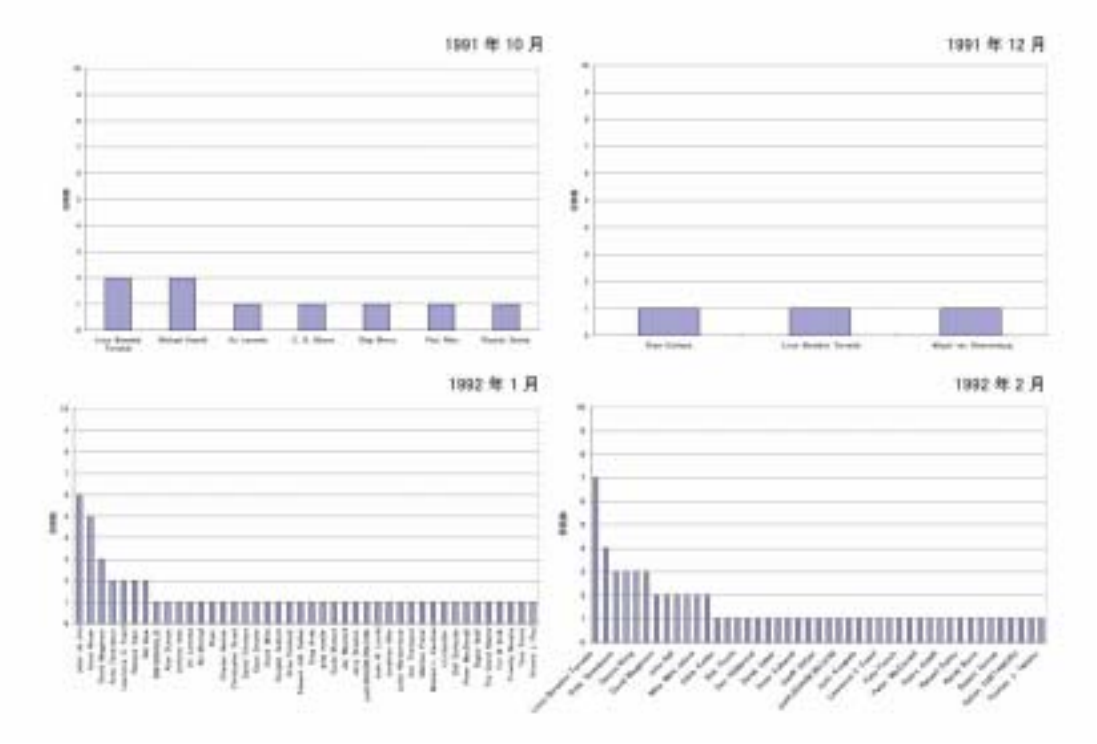

図-13　comp.os.minixニュースグループにおけるLinux関連の投稿者別の投稿数

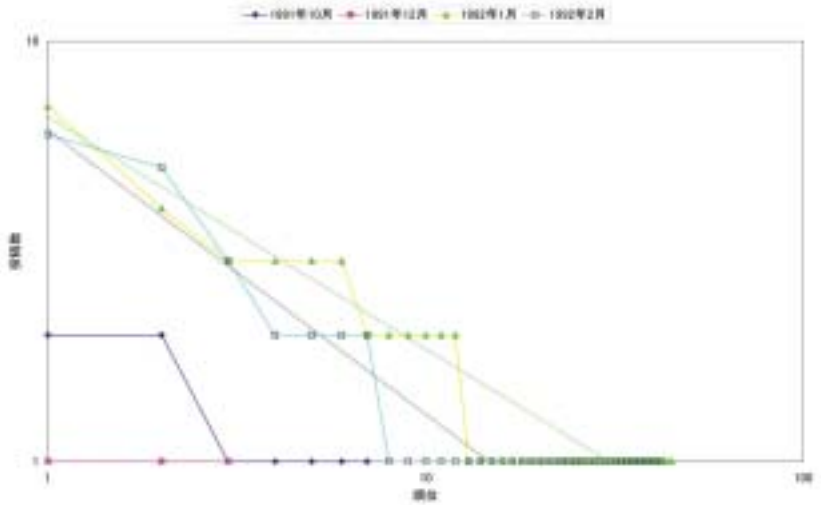

図-14　comp.os.minixニュースグループにおけるLinux関連の投稿者別の投稿数の分布

性から、ニュースグループやメーリングリストには新しい人がどんどん加わることになる。その結果、comp.os.minixニュースグループでは、「Linuxとは何か？」や「Linuxはどこで手に入れることができるのか？」といった質問が絶えず投げかけられている。このような質問に対し、すでにそのコミュニティにいた人たちが親切に答えていく。また、Linux-Activistsメーリングリストにおいても、技術的な意見を求めるような質問が多く投げかけられる。このようにして、「質問」のコミュニケーションは、次のコミュニケーションを誘発するのである。これは、まさにインターネットの特徴でもあり、「ネットワークの中では、『ぼくは知らない、ぼくはできないから』ということを言うことで物事が起こる」[9]ということに他ならない。

2つめの要因は、「呼びかけ」のコミュニケーションである。ここでいう「呼びかけ」というのは、「僕にはお手上げなので、誰かやってください」というような発言や、「こういうものがまだ実装されていないので、それを次のリリースに盛り込みたいと思います」という計画の表明などである。このような投稿がなされると、それについての意見を述べたり、それを実現したコードを提供するコミュニケーションなどを誘発する。オープンソース開発では、そのプロジェクトが「終わった」(＝完成した)という印象を与えないようにすることが非常に重要であると、エリック・レイモンドは指摘している[2]。「そんな印象ができたら、潜在的な貢献者は、自分が必要とされていないような気分になるかもしれない」からである。そうではなく、「自慢はコード自身にさせておいて、口では『ああちくしょう、このソフトはまだXもYもZもできない、まだぜんぜんダメだ』。と言えば、X、Y、Zのパッチはすぐに出てくることが多い」[2]のである。

3つめの要因は、「不具合報告」のコミュニケーションである。例えば、「少し古い機種にインストールしたところ、エラーが出た」というものや「ディレクトリのコピーをしようとしたら、うまくできなかった」というような不具合の報告である。この不具合には、そもそも仕様上できないというものから、いわゆる「バグ」までいろいろな原因がある。前者であれば、その機能の必要性や実現方法についての議論につながり、後者であればそれを修正して報告・提出するコミュニケーションが誘発される。トーヴァルズも「ぼくを奮い立たせるのは、バグ・レポートだ」[6]と語っている。「ぼくは、いつもベッドからごろんと出ると、すぐにメールをチェックした・・・何かの問題が解決されたかどうかを確かめるためだった。『今日はどんな新しいエキサイティングな問題が持ち上がってるかな？』とか『例の問題、誰が解決したかな？』」[6]というチェックをするためだという。

このように、メーリングリストやニュースグループでは、「質問」、「呼びかけ」、「不具合報告」のコミュニケーションがさらなるコミュニケーションを生み出す契機となって、コミュニケーションの持続的な連鎖を実現したのである。

4.3　論争による思考とコミュニケーションの加速、および深化

Linuxの事例においては、comp.os.minixニュースグループにおけるライバルとの論争が、思考やコミュニケーションの連鎖を加速し、深化させたことは注目に値する。1992年1月28日、Minixの開発者であるアムステルダム自由大学教授アンドリュー・タネンバウムは「Linuxは時代遅れだ」という投稿をし、Linuxの問題点を指摘・批判した。これに対し、トーヴァルズも荒々しい言葉を交えて応戦した。この論争は、最終的には多くの人々を巻き込んで展開することになった[9)]。comp.os.minixニュースグループにおける投稿数の推移を表した図3からも、タネンバウムの投稿後、Linux関連の投稿が増加していることがわかる。
結果として、この激しい論争は、LinuxとMinixのそれぞれの特徴を際立たせ、多くの人々をLinuxに惹きつけることになった[1]。そして、ニュースグループの記事を読んでいた多くの人たちが、論争のコミュニケーションを通じてLinuxについて理解を深めることになったのである。LinuxやMinixの欠点や実現できていない点が次々に明らかにされていくので、多くの人が、Linuxの何が課題なのか、次に開発すべきものは何なのか、ということを理解することができたのである。こうして、後にLinux開発のキーパーソンとなる開発者たちが、この論争をきっかけにLinuxの開発に参入することになったのである[1]。

もちろん、この論争はトーヴァルズにとっても必死な戦いであった。Linuxについて突かれるところは、すべて答えなければならなかったため、設計に関する様々なことを考えざるを得ない状況に追い込まれたのである。そのときの状況について、トーヴァルズは、「ぼくはリナックスとぼくの男を賭けて弁論をおこなわざるをえなくなった」[6]と語っている。トーヴァルズが攻撃的に応戦したのは、「ぼくはあそこのみんなに対する社会的な立場ってものをいつも気にかけていたんだけど、教授はそこを攻撃してきた」[6]

からであり、「自分の名誉を守らなければいけないと思った」[6]というわけである。労力を要する論争のコミュニケーションは、このような心的な要因も影響して生成され、加速的に連鎖していくことになったのである。

5．構造的カップリングとコミュニケーション・メディア

本論文では、メーリングリストとニュースグループにおける「コミュニケーションの連鎖」を詳細にみてきたが、その全体的な見取り図をまとめると図15のようになる。開発の中心的な舞台となるLinux-Activistsメーリングリストとcomp.os.minixニュースグループでは、それぞれコミュニケーションの連鎖が起こっている。これらの連鎖は、社会システム理論の用語でいうならば、それぞれが「コミュニケーション・システム」（社会システム）であるといえるだろう。そこでのコミュニケーションの連鎖は、Linuxの開発に関わるコミュニケーションのみから構成されるという意味で「作動上の閉鎖性」がある。しかし、世界中から誰でも参加できるという意味では「開放的」である。このような視点でみると、オープンソース開発は、閉鎖性と開放性をもつコミュニケーションの連鎖によって、自らの境界を形成し続けるコミュニケーション・システムとして捉えることができる。

メーリングリストとニュースグループにおけるコミュニケーションの連鎖は、それぞれメール投稿と記事投稿という異なる作動（操作・手段）によるコミュニケーションで構成されており、作動上は閉じている。しかし、これらのコミュニケーション・システムは構造的にカップリングされており、お互いに刺激し合っている[11]。つまり、これらのシステムは融合することはないが、相手側の内容を情報として取り上げることができ、かつプログラム（ソースコード／バイナリ）や人というリソースも共有している[10)]。このようなコミュニケーション・システムの環境には、各開発者の心的システムが存在している。心的システムにおいては、意識の連鎖が起こっており、それが思考となる。コミュニケーションの連鎖（社会システム）と意識の連鎖（心的システム）にも「構造的カップリング」の関係性がある[12]。それらのシステムはお互いの要素をインプット／アウトプットする関係にないという意味で閉じているが、相互に刺激し合っているのである。

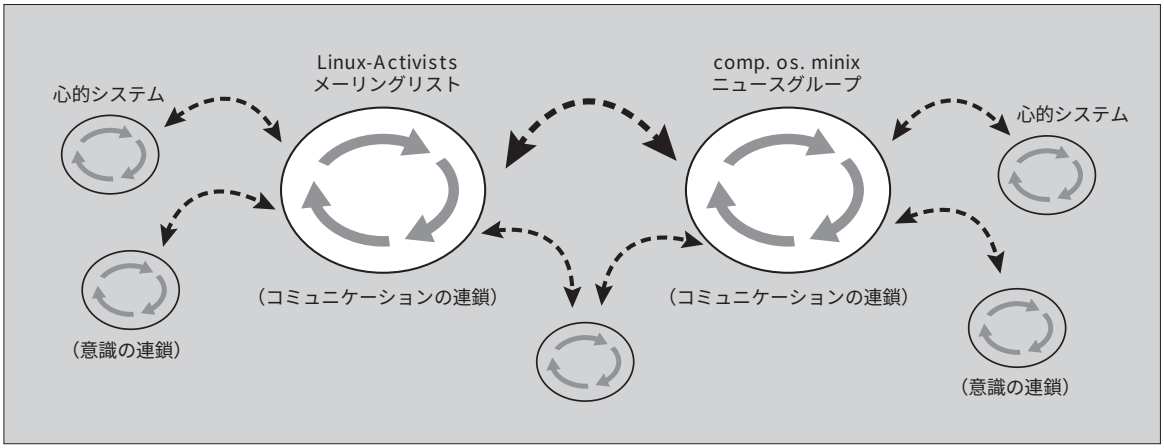

図-15　Linux開発の最初期における2つのコミュニケーション・システムと心的システムの構造的カップリング

最後に、オープンソース開発が持続的に行われる（開発のコミュニケーションが連鎖していく）ために必要であったメディアについて触れておくことにしたい。コミュニケーション・システムにおいては、「コミュニケーションを受容する用意が整えられ、コミュニケーションが企てられてコミュニケーションが見込みのないものとして取りやめられない」[13]ための仕組みが必要となる。このような不確実性を確実性に変換させる機能をもつものを、社会システム理論では「メディア」という[4]。メディアは、本来成立しにくいコミュニケーションを成立させやすくする。ひとつひとつのコミュニケーションは瞬時に立ち消えてしまうが、メディアが存在することで、それに続くコミュニケーションを絶えず生成・連鎖させることが可能になるのである。

社会システム理論では、コミュニケーションには三つの不確実性がつきまとうと考え、それぞれ対応するメディアを考える。すなわち、(1) 他者理解の不確実性、(2) 到達の不確実性、(3) コミュニケーション成果の不確実性に対して、(1) 言語、(2) 拡充メディア（コミュニケーションを拡充するメディア）、(3) コミュニケーション・メディア（象徴的に一般化したコミュニケーション・メディア）というメディアが対応すると考えるのである。

オープンソース開発において、「自然言語」（英語など）や「人工言語」（プログラミング言語など）が、メーリングリストとニュースグループ上でのコミュニケーションを実現するのに用いられていることは明らかである。そして、世界中から不特定多数の開発者が参加できるプラットフォームとして、「インターネット」という伝達メディアが不可欠であった。メーリングリストやニュースグループ、FTPサイトが特に重要な役割を担っていたのは、すでにみてきた通りである。さらに、ソフトウェアが「もの」ではなく「情報」であることから、伝達メディアを通じて多くの複製をつくることができた点も見逃せない点であり、そのことが発散（多様性の生成）と収束（選択）による進化的プロセスを早め、コミュニケーションが立ち消えなかったポイントになっていると思われる。最後に、「オープンソース開発であるということ」が、開発者を次々につなげていくコミュニケーション・メディアになっていたと考えられる。多くの人々がここまで開発に熱中したのも、それが「フリー」なOSを目指した活動であったからだろう。このような多重のメディアが存在することによって、開発コミュニケーションが連鎖していくことが可能になったのである。

6．おわりに

本論文で扱った事例は、専門的な知識・技術をもったコンピュータ技術者たちによる活動であり、特殊なものに見えるかもしれない。たしかに、そのまま他の事例に適用できるような一般性を主張することはできないだろう。その意味で、本論文で得られた知見は、マートンのいう「中範囲理論」であるといえる。しかし、不特定多数の人たちが出入りしながら協働的に付加価値を生み出していくという「オープン・コラボレーション」は、技術の世界だけでなく、いろいろな分野で応用可能であると私は考えている。そのとき問題となるのは、いかにしてそのようなオープン・コラボレーションのムーブメントをテイクオフさせるのか、という問題である。本論文におけるLinux開発の最初期の分析は、そのような問いに答えるための第一歩であるといえる。社会システム理論をベースとした抽象的なレベルでの把握を行ったのも、「ハッカー倫理」[5]というようなコンピュータ依存の理解から離れ、社会レベルの組織化原理に迫るためであった。この試みがどの程度成功しているかは現時点ではわからないが、他の事例においても理解し得る枠組みとなるかどうか、今後のさらなる研究をもって判断していくことにしたい。

[注]

1）Linuxは、そのOSの中核部分（カーネル）だけでも、250万行強のソースコードがあるといわれている。

2）このような先行研究には、たとえば『伽藍とバザール』[2]や『オープンソースの成功』[3]などがある。

3）このような観点でオープンソース開発を捉えたものに、ペッカ・ヒマネンの『リナックスの革命』[6]がある。ヒマネン自身はこのような表現を用いてはいないが、ハッカー倫理が二重の偶発性を解決したと指摘されている。

4）Linuxの最初期の開発については、『それがぼくには楽しかったから』[6]『ソースコードの反逆』[7]等で紹介されている。本論文ではこれらの文献と、メーリングリストやニュースグループのログを参考にした。

5）このFTPサイトは、ソースコードの共有という実際面でも重要な役割を果たした。グリン・ムーディは、このFTPサイトの存在が「Linuxが発展する上で決定的に重要であった」[7]と述べている。また、トーヴァルズ自身も「僕がLinuxに下した最善の決定は、FTPでLinuxを入手できるようにしたことだ」[7]と振り返っている。

6）この時点でのcomp.os.minixニュースグループの参加者数はわからないが、1987年の作成時には「1か月で40,000名もの参加があった」[7]といわれており、OSに興味がある人の相当数の人が見ていたと思われる。

7）本論文では、Linux-Activistsメーリングリストについては、Kansas City Linux User Groupによるアーカイブ（http://www.kclug.org/old_archives/linux-activists/）、comp.os.minixニュースグループについては、LISTSERV.NODAK.EDUによるアーカイブ（http://listserv.nodak.edu/archives/minix-l.html）を参照した。なお、解析期間をこの時期に限定したのは、これ以降はcomp.os.linux（当初はalt.os.linux）という別のニュースグループが立ち上がるためであり、ここまでの区間をLinux開発の最初期と定義し、分析を行った。

8）このようなコメントチェインの分析を行っている先行研究に、金子らの研究[8]がある。

9）この論争は、アーカイブに記録が残っているほか、『オープンソースソフトウェア』[10]にも掲載されている。

10）メーリングリストとニュースグループでは、投稿者の重なりはそれほど多くはない。例えば、1992年1月では、メーリングリストへの投稿者は223人、ニュースグループへの投稿者は45人にいるが、そのうち両方に投稿している人は18人である。1992年2月では、メーリングリストへの投稿者は220人、ニュースグループへの投稿者は42人にいるが、そのうち両方に投稿している人は7人のみである。

[参考文献]

[1] Peter Wayner, *Free for All: How Linux and the Free Software Movement Undercut the High-Tech Titans*, Diane Pub Co, 2000.（ピーター・ウェイナー,『なぜ、Linuxなのか？』, 星 睦（訳）, アスキー, 2001.）

[2] Eric S. Raymond, *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*, Oreilly, 1999（エリック・スティーブン・レイモンド,『伽藍とバザール』山形 浩生（訳）, 光芒社, 1999.）

[3] Steven Weber , *The Success Of Open Source*, Harvard University Press, 2004（スティーブン・ウェバー,『オープンソースの成功』, 山形浩生, 守岡桜（訳）, 毎日コミュニケーションズ, 2007.）

[4] Niklas Luhmann, *Soziale Systeme: Grundrißeiner allgemeinen Theorie*, Suhrkamp Verlag, Frankfult am Main, 1984.（ニクラス・ルーマン,『社会システム理論』, 上下巻, 佐藤勉（監訳）, 恒星社厚生閣, 1993.）

[5] Pekka Himanen, Linus Torvalds, Manuel Castells, *The hacker ethic and the spirit of the information age*, Random House, 2001.（ペッカ・ヒマネンほか,『リナックスの革命』, 安原 和見, 山形 浩生（訳）河出書房新社, 2001）

[6] Linus Torvalds, David Diamond, *Just for Fun*, Harperbusiness. 2001.（リーナス・トーバルズ, デイビッド・ダイヤモンド,『それがぼくには楽しかったから』, 風見 潤（訳）, 小学館プロダクション, 2001.）

[7] Glyn Moody, *Rebel code. 2nd ed.*, Penguin Books, 2001.（グリン・ムーディ,『ソースコードの反逆―Linux開発の軌跡とオープンソース革命』, 小山 裕司（訳）, アスキー, 2002.）

[8] 金子郁容、VCOM編集チーム 編著『「つながり」の大研究－電子ネットワーカーたちの阪神淡路大震災』日本放送出版協会、1996

［9］ 松岡正剛、金子郁容、吉村伸『インターネットストラテジー：遊牧する経済圏』ダイヤモンド社, 1995
［10］Chris Dibona, Mark Stone, Sam Ockman（eds）, *Opensources: Voices from the Open Source Revolution*, Oreilly & Associates Inc, 1999.（クリス・ディボナ, マーク・ストーン, サム・オックマン,『オープンソースソフトウェア―彼らはいかにしてビジネススタンダードになったのか』, 倉骨 彰（訳）, オライリー・ジャパン, 1999.）
［11］Niklas Luhmann, *Das Recht der Gesellschaft*, Suhrkamp Verlag, Frankfurt am Main, 1993（ニクラス・ルーマン,『社会の法』, 第1巻, 第2巻, 馬場靖雄, 上村隆広, 江口厚仁（訳）, 法政大学出版局, 2003.）
［12］ Niklas Luhmann, DIE KUNST DER GESELLSCHAFT, Suhrkamp Verlag, Frankfurt am Main, 1995（ニクラス・ルーマン,『社会の芸術』, 東京大学出版会, 2004）
［13］Niklas Luhmann, *Liebe als Passion: Zur Codierung von Intimitat*, Suhrkamp Verlag, 1982.（ニクラス・ルーマン,『情熱としての愛：親密さのコード化』, 佐藤勉, 村中知子（訳）, 木鐸社, 2005.）