December 2002

THE OPEN SOURCE SOFTWARE DEVELOPMENT PHENOMENON: AN ANALYSIS BASED ON SOCIAL NETWORK THEORY

Gregory Madey
University of Notre Dame

Vincent Freeh
University of Notre Dame

Renee Tynan
University of Notre Dame

THE OPEN SOURCE SOFTWARE DEVELOPMENT PHENOMENON: AN ANALYSIS BASED ON SOCIAL NETWORK THEORY

Greg Madey
Computer Science & Engineering
University of Notre Dame
gmadey@nd.edu

Vincent Freeh
Computer Science & Engineering
University of Notre Dame
vin@nd.edu

Renee Tynan
Department of Management
University of Notre Dame
rtynan@nd.edu

Abstract

The OSS movement is a phenomenon that challenges many traditional theories in economics, software engineering, business strategy, and IT management. Thousands of software programmers are spending tremendous amounts of time and effort writing and debugging software, most often with no direct monetary compensation. The programs, some of which are extremely large and complex, are written without the benefit of traditional project management, change tracking, or error checking techniques. Since the programmers are working outside of a traditional organizational reward structure, accountability is an issue as well. A significant portion of internet e-commerce runs on OSS, and thus many firms have little choice but to trust mission-critical e-commerce systems to run on such software, requiring IT management to deal with new types of socio-technical problems. A better understanding of how the OSS community functions may help IT planners make more informed decisions and develop more effective strategies for using OSS software. We hypothesize that open source software development can be modeled as self-organizing, collaboration, social networks. We analyze structural data on over 39,000 open source projects hosted at SourceForge.net involving over 33,000 developers. We define two software developers to be connected — part of a collaboration social network — if they are members of the same project, or are connected by a chain of connected developers. Project sizes, developer project participation, and clusters of connected developers are analyzed. We find evidence to support our hypothesis, primarily in the presence of power-law relationships on project sizes (number of developers per project), project membership (number of projects joined by a developer), and cluster sizes. Potential implications for IT researchers, IT managers, and governmental policy makers are discussed.

Keywords: Open source software, social networks

Introduction

The global E-Commerce infrastructure relies heavily on open source software (OSS). Examples include web-servers (Apache, iPlanet/Netscape), e-mail servers (Sendmail), languages (Perl, Java, Python, GCC, Tk/TCL), and operating systems (Linux, BSD Unix). Corporate IT managers and decision makers have traditionally depended on either internally developed systems or commercially purchased systems that have been mostly closed source and proprietary. Open source software is written and supported by programmers, many coming from the "hacker culture". This development culture includes hundreds of thousands of distributed programmers voluntarily producing, sharing, and supporting their software with no monetary compensation for their efforts. This presents those corporate IT managers and decision makers with a new development process and culture to learn and

interact with. Although the open source development model has been used for over 25 years, its significance to mainstream business IT grew with the Internet and the emergence of E-Commerce. These OSS developers collaborate from around the world, self-organize, and rarely meet face-to-face. Questions about their motivations, about the quality and ethics of this OSS phenomenon, new intellectual property legal questions, and whether there is a role for new government policy in this area can be asked. For example, should governments widely adopt open source software, should governments take an active role in promoting standards with the OSS process, and what role does OSS have in promoting or diminishing innovation? A better understanding of the process of open source development is needed. In this paper, we report the results of an empirical investigation of the social and collaborative networks present in the open source movement.

Open Source Software Development

Open source software is by definition software for which users have access to the source code. This distinguishes it from the recent common practice by commercial software publishers of only releasing the binary executable versions of the software. Most open source software is also distributed at no cost with limited restrictions on how it can be used; hence the term "free" when used to describe open source carries two meanings: 1) free of cost and 2) free to do with the software as you wish (i.e., most importantly — free to read the code).

Case studies documenting the open source software development model, albeit often sympathetic to that model, point to potential lessons and benefits that may be of value to corporate IT (O'Reilly, 1999; Wu, 2001). It is claimed that open source development produces more bug-free code, faster, than closed proprietary developed code, although this has yet to be conclusively demonstrated (Jorgensen, 2001; Koch, 2002; O'Reilly, 1999; Raymond, 1999; Sharma, 2002; Stamelos, 2002; Wang, 2001; Wu, 2001). Open source software development teams, are generally comprised of volunteers working not for monetary return, but for the enjoyment and pride of being part of a successful virtual software development project. Team members often come from around the world and rarely meet one another face-to-face. The open source projects are self-organized, employ extremely rapid code evolution, massive peer code review, and rapid releases of prototype code. Many of these practices are counter intuitive and the opposite of what conventional software engineering holds as the correct processes for the production of high quality code (Bollinger, 1999; Charles, 1998; Edwards, 1998; Fielding, 1999; Hecker, 1999; Lawrence, 1998; O'Reilly, 1999; Ousterhout, 1999; Payne, 2002; Raymond, 1999; Sanders, 1998; Torvalds, 1999)

The Open Source Software movement is a prototypical example of a decentralized self-organizing process. There is no central control or central planning. It challenges conventional economic assumptions, it turns conventional software engineering and project management principles inside out, it threatens traditional proprietary software business strategies, and it presents new legal and government policy questions regarding software licensing and intellectual property. Moreover, OSS is a major component of the IT infrastructure enabling global e-Commerce. Open source software including BIND, sendmail, Apache, Linux, INN, GNU utilities, MySQL, PostgreSQL, and Perl are critical components of the Internet. They enable major services hosted on the Internet, e.g., e-mail, WWW, e-Commerce, domain name lookup. The *Netcraft.com* survey of 36.6 million web servers worldwide reports an over 60% market share for the open-source web-server Apache (Netcraft, 2002).

Open source software is free and open; this generates questions about its impact on domestic digital divides, the third-world digital-disadvantaged, the OSS influence on innovation, and national and corporate security from cyber-terrorism. For example, while free (i.e., no cost) software may help reduce the digital divide, will it do so by weakening the software industry's business-model? Does OSS promote innovation, or does it ride on the coattails of closed source software development?

Several important studies on the OSS phenomenon have been conducted and provide important foundations for this research. Feller and Fitzgerald (Feller, 2000) developed a research framework and analyzed the OSS phenomenon. Hars and Ou (Hars, 2001) surveyed OSS developers and reported on their motivations for participation in OSS projects. Scacchi (Scacchi, 2002) has an extensive ongoing study of the socio-technical processes associated with OSS development work practices. Wolf, et al recently released the results of a survey of 526 OSS developers from SourceForge, and 134 participants on the Linux kernel mailing list, reporting on developer motivations and attitudes (Wolf, 2002).

Social and Collaborative Network Theory

Several research streams converge to provide us with a number of tools and models for analyzing the open source software movement: social network theory, small world phenomenon, power-laws, self-organization, and graph theory. Social network

theory models persons as nodes of a graph and their relationships as edges of the graph (Jin, 2001; Wasserman, 1999; Watts, 1999; Watts, 1998). Thus two persons are directly connected if they have a relationship (e.g., friendship) with each other; they then are one link away from one another. More distant relationships are modeled as paths through the graph; a "friend of a friend" is two links away. Several studies reveal an interesting phenomenon present in many of these social networks; most persons are very few links from any other person – the Small World Phenomenon (Watts, 1999; Watts, 1998). This idea was popularized in the play (and movie) *Six Degrees of Separation* (Guare, 1990) which claims that all persons in the world are at most six friendship links away.

Collaborative networks are variations of social networks, where the relationships are collaborations, e.g., actors in movies (Tjaden, 1996; Watts, 1999), or co-authors on research papers (Barabasi, 2001; Newman, 2001). Often entire populations are connected into one large cluster with characteristic cluster coefficients (Watts, 1999). Highly prolific actors or authors are linchpins in collaborative networks. Linchpin actors or researchers play key roles in bridging disparate groups into one large cluster by being the only joint participant in two different projects.

Social networks, collaborative networks, and other self-organizing systems (e.g., the Internet, WWW pages, U.S. firm sizes, cities, economic systems, word usage in languages, ecosystems) often have another interesting property; they have highly skewed distributions, which under a log-log transformation results in a linear relationship. This is called a power-law relationship. Power-law relationships have been reported for the Internet (Albert, 1999; Barabasi, 1999; Barabasi, 2000; Faloutsos, 1999; Huberman, 1999), sizes of U.S. firms (Axtell, 2001), city size distributions (Pumain, 1997), ecosystems (Jorgensen, 1998), word rank in languages and writing (Schroeder, 1991) and many others.

Why such systems have power-law relationships is an open research question. Some speculate that self-organizing processes, when modeled as growing networks, display non-random attachment of nodes (sometimes called preferential attachment) (Barabasi, 1999; Callaway, 2001; Newman, 2001).

We analyze the open source movement by modeling it as a collaborative social network. The developers are nodes of a graph and joint membership on an open source project is a collaborative link between the developers. The open source software development movement is highly decentralized and is a volunteer effort where developers freely join projects that they find appealing – all attributes of typical self-organizing systems. We hypothesize that the open source movement displays power-law relationships in its structure. Our empirical analysis of structural data collected from SourceForge suggests that this is the case. If this is supported by more detailed investigations, and as additional general theories are developed about social and collaborative networks (e.g., distributions in networks with non-random growth), that theory may then be applied to the open source software development process.

Data Collection and Analysis

We gathered data monthly over the 14 month period from January 2001 through March 2002 at SourceForge, a web-based project support site sponsored by VA Software. SourceForge provides project management tools, bug tracking, mail list services, discussion forums, version control software for over 33,000 open source developers, participating on over 39,000 projects, as of February 2002 (SourceForge, 2002; Wu, 2001). We note that not all open source projects are registered with SourceForge; many high profile projects maintain their own developer sites, e.g., Apache, Perl, sendmail, Linux. But some large projects have moved to SourceForge (e.g., Samba) and we speculate that there are many smaller projects that have not joined SourceForge. Our assumption is that the projects at SourceForge are representative of the overall open source movement, in part because of its popularity and the large number of projects and developers registered there.

The primary data required for this research is a table consisting of records with two fields: project number and developer ID. Because projects can have many developers and developers can be on many projects, neither field is unique primary key. Thus the composite key composed of both attributes serves as a primary key. Each project in SourceForge has a unique project number. Additionally, each developer is assigned a unique ID when registering with SourceForge.

A web crawler traversed the SourceForge web server to collect the necessary data. All project home pages in SourceForge have a similar top-level design. Many of these pages are dynamically generated from a database. In particular, the developers belonging to a project are found by issuing the following request:

http://sourceforge.net/project/memberlist.php?group_id=projnum

A simple shell script fetches each project's developer page, then parses the HTML, extracting the names of the developers. A python program parses the HTML source. It outputs one line for each developer, which contains the project number and the developers ID.

The above script (and auxiliary programs) creates a file of project numbers and developer IDs. Below is an extract of this file:

8001|dev378
8001|dev8975
8001|dev9972
8002|dev27650
8005|dev31351
8006|dev12509
8007|dev19395
8007|dev4622
8007|dev35611
8008|dev7698

The fields, delimited by "|", represent the SourceForge project ID and the developers userid, respectively. We parsed the data into a database, and used SQL queries to extract statistics and to data-mine for relationships and cluster properties. Actual developer userids have been anonymized.

Identification of clusters — connected groups of developers — is implemented as a modified spanning tree algorithm. We use Oracle PL/SQL and Perl as our analysis, programming and data-mining environments.

Results

The structural data was collected at SourceForge.net, the largest Open Source Foundry (SourceForge, 2002). SourceForge is a free hosting service for Open Source projects which offers, among other things, web site hosting, mailing lists, bug tracking, message forums, and task management software.

We model the OSS developers and projects as a network in two complementary ways. First, each developer is a node in the network; an edge exists between nodes if both developers are on the same project as shown in Figure 1. In that figure we observe two linchpin developers, dev[58] and dev[46], who tie 5 separate projects into a cluster of 24 developers. This representation is analogous to movie actors as nodes and movies as links, or research paper authors as nodes and joint authorship as a link in the collaboration networks discussed above. The second way uses projects as nodes. Our initial analysis of the structural data shows that the developer collaboration network at SourceForge fits a power-law model, as determined by ordinary least squares (OLS) regression in log-log coordinates. As shown in Figure 2, both the project-size (number of developers on the project) and the number of projects per developer (total number of projects-joined by a developer) have power-law distributions. The solid line is the OLS regression line though the data, with an adjusted $R^2 = .93$ for the project-size data, and an adjusted $R^2 = .97$ for the projects-joined data. This power-law distribution is often a property of such self-organizing systems.

Cluster analysis identified the presence of one large cluster, consisting of 6,862 developers. The next largest cluster was of size 55, with sizes ranging down to one, i.e., those developers on single-member projects. Figure 1 displays a small cluster of 5 projects, connected by two linchpin developers. The log-log plot of the distribution of cluster sizes is shown in Figure 3a. The one large cluster is an outlier, relative to the rest of the data points, yielding a poor fit. Removing the one large cluster, and fitting the log-log transformation to a linear relationship (displayed in Figure 3b) yields a good fit, with an adjusted $R^2 = .97$. We thus observe a good fit to the power-law relationship on project sizes, developer participation, and clusters sizes for all but the one large cluster.

In the actor-movie network, over 90% of the actors are in one large cluster (Watts, 1999), while the single large cluster in the open source network is only 25% of the total number of developers at SourceForge. This suggests that the open source developers may not be as well connected as other social or collaborative networks. This difference may be explained by the age and maturity of the movie industry — SourceForge was about two years old when we collected the data — or it may be explained by the transient nature of the links in the OSS development community discussed in the next paragraph.

An alternative explanation may be the relative diversity of project types on Sourceforge. Perl programmers, for example, may

be less likely to develop connections to Fortran programmers than with C++ programmers. In addition, the goal to be included in high budget Hollywood productions is the goal of many if not most actors, which may result in more strategic forging of links to well-connected others, in some cases with the help of others such as agents. While the database does not include agents, they can play a behind-the-scenes role in terms of making connections.

The small projects have not had time to attract linchpin developers to tie the many projects into a large cluster linking a large number of registered developers. Alternatively, the SourceForge site is serving a critical role, by linking developers that might not normally be connected. This suggests that a longitudinal study of the growth of the open source network is needed to follow the attachment, detachment, and evolution of that network.

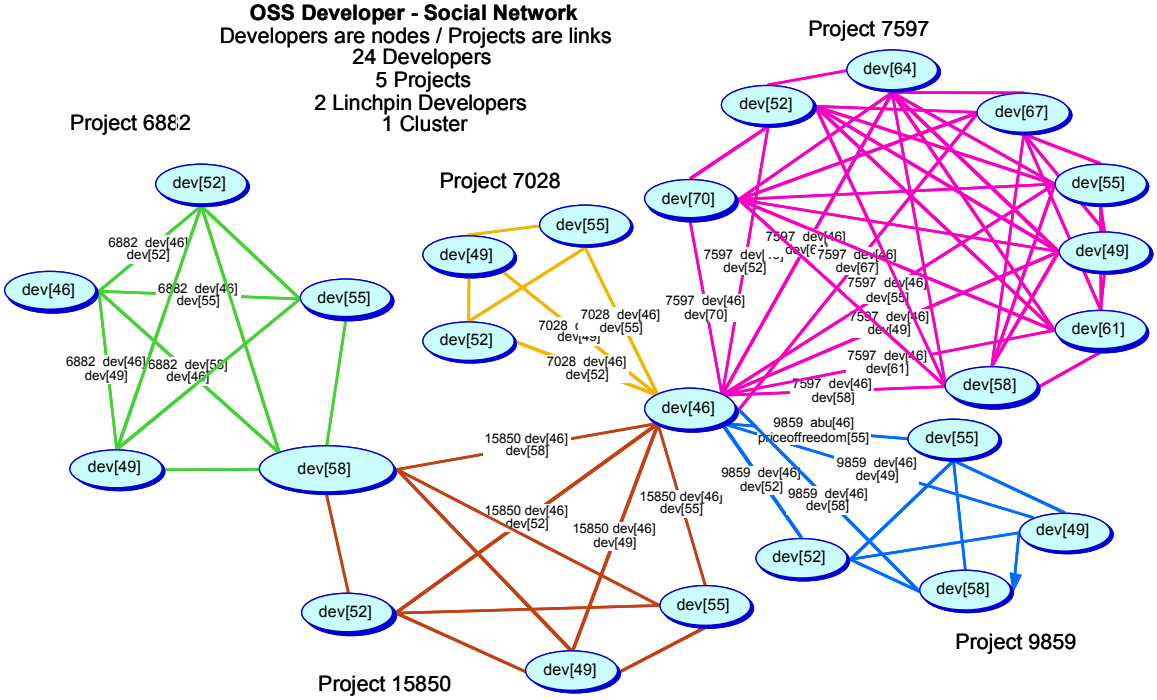

Figure 1. Developer Social-Network, Linked by Joint Project Membership, Cluster of Size 24

In most collaboration networks that have been studied, links are persistent. For example, actors are linked if they appeared together in a movie. This link is never removed, and it does not require any additional effort by the actor to maintain. But in the developer network, the link exists only while the developer is a member of a project. Thus the links in this network are transient, and node degree (number of edges) of developers is much smaller. Over the 14 months of our initial survey, the "busiest" developer (on the most projects) has fluctuated between membership on a low of 17 and a high of 27 projects, while the number of developers belonging to exactly one project has remained a nearly constant 80%.

SourceForge rates projects by activity, such as downloads, updates, and page hits. This allows one to find distinguishing characteristic of top projects. Looking at the top 10 projects for the month of February 2002, we find that there is an average of 29 developers per project, 20 times the SourceForge average. Also these developers belong to 25% more projects than the overall population. Finally, 70% of the developers on top projects belong to exactly one project (versus 80% overall), but the maximum is only 12.

Discussion

The results support our speculation that the open source movement is not a random graph (i.e., new nodes attach to existing nodes with uniform probabilities), but a graph displaying preferential attachment of new nodes (i.e., some nodes have higher probability of attachment that others). This typically happens under situations of positive feedback or increasing returns and is sometimes call the “rich-get-richer” effect or the “band-wagon” effect. In the open source movement, initial success breeds more success because developers prefer to be part of a successful project. Since the developers are free to self-organize and select the projects

they choose to join, it is reasonable to expect that some projects will be more visible or more attractive than others, hence some projects will grow disproportionately larger that expected under random growth. Networks that have heavily skewed distributions and display power-law relationships, are often associated with preferentially connected networks (Barabasi, 2000; Callaway, 2001; Newman, 2001). That we observe the same for the open source software projects at SourceForge supports our above speculation. We can also speculate on the reasons that may induce preferential attachment. Additional data collection and analysis will be needed to resolve this question. Should open source networks be determined to have these power-law relationships, then by inheritance the new results of the ongoing research on similar networks (research authors, actors, the Internet, Web pages, etc.) would also apply to the open source movement. This would enable better modeling of the processes associated with this development strategy, hence supporting research in this area and possibly enabling better implementation of such open source development strategies by IT organizations. For example, we observe the importance of the linchpin nodes in growing larger clusters. In the case of open source development, these developers play a similar role to the "gatekeepers" in organizational studies on technology diffusion. These linchpin developers may need to be identified, nurtured, and supported in their role of facilitating the diffusion of ideas and technology between disparate development groups.

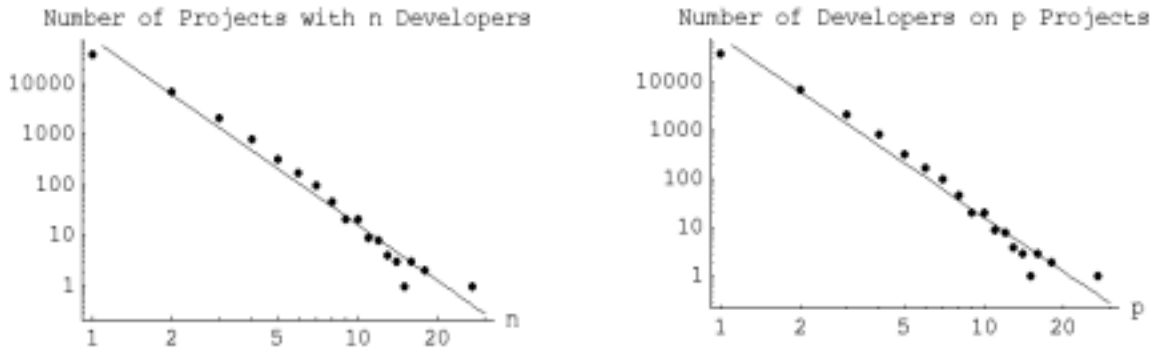

Figure 2. Power Law Relationships: OSS Project Size and Developer Project Membership

The ability to model OSS with power law relationships raises important research questions. What is the mechanism of the rich-get-richer phenomenon in this community? Are there limits to the growth of projects and clusters? The experience of Linux, with its extremely large size and continued explosive growth suggests that it is possible for large OSS projects to continue to be effective over time. The Linux example also may also show that if the site is well-organized and well-run, the benefits of more error checking, opinions, etc. outweigh the process losses which accrue when individuals working together lose some of their potential joint productivity due to ineffectiveness in the work process. If this is the case, it would be extremely useful to determine how the process losses of many other work groups are avoided in well-functioning OSS sites.

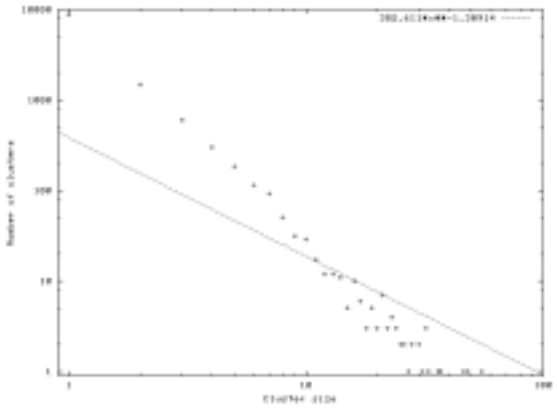

Figure 3a. Power Law—Clusters with Outlier

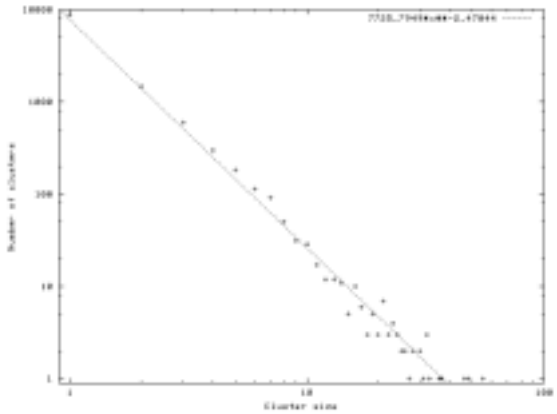

Figure 3b. Power Law—Clusters w/o Outlier

We consider the single large cluster that we labeled an outlier in Figure 3a. In the actor-movie network, over 90% of the actors are in one large cluster, while the single large cluster in the open source network is only 25% of the total number of developers at SourceForge. This suggests that the open source developers may not be as well connected as other social or collaborative networks. This difference may be explained by the age and maturity of the movie industry while SourceForge was just a little over one year old when we collected our first data. The singleton projects just haven't had time to attract linchpin developers to tie the many projects into a large cluster linking a large number of registered developers. Alternatively, the SourceForge site is serving a critical role, by linking developers that might not normally be connected. This suggests that a longitudinal study of the growth of the open source network is needed to follow the attachment, detachment, and evolution of that network.

Summary and Conclusion

We describe an empirical study of the open source projects registered at SourceForge. We believe they are representative of open source movement worldwide. Those projects were modeled as a collaborative social network, with developers as nodes and joint membership in projects as links between the nodes. Analysis of the data displays a heavily skewed distribution, which has a good fit to a power-law relationship. Previous studies of social and collaborative networks with similar properties are believed to grow not as random networks, but as preferentially connected networks. Our study suggests that the same may be true of the open source movement. If this observation is true, then the active research on other such social networks may produce insights that may be applied to further research on open source software.

Several assumptions and limitations are present in the study. We assume that the projects at SourceForge are representative of open source projects in general. This needs to be confirmed. Although we have collected monthly data over 14 months, our analysis only looked at a monthly "snapshot" of the open source network at SourceForge. Once several years of data are collected, a longitudinal and dynamic analysis may provide better understanding of how node attach and detach from the network. Data on developers who dropped off of projects was not analyzed. We consider only the linking relationship of joint project membership; many of the developers are linked through other relationships, e.g., shared subscriptions to newsletters, listserv's, or reading common web pages. The effect of those other linking relationships, along with the effect of SourceForge itself, should be further investigated. Is the open source movement highly fragmented with SourceForge helping to link those fragments together into a larger connected collaborative cluster?

Additional graph theoretic properties can be computed to provide insight on the nature of the open source network, such as cluster coefficients, "degrees of separation", network diameter, etc. This study does not include that analysis, but could be extended as such. Other analytical approaches could include agent-based modeling and simulation as reported elsewhere (Madey, 2002).

References

Albert, R., Jeong, H., Barabasi, A. L. "Diameter of the World Wide Web," *Nature* (401), 1999, pp. 130-131.

Axtell, R.L. "Zipf Distribution of U.S. Firm Sizes," *Science* (293:5536), 2001, pp. 1818-1820.

Barabasi, A.L., Albert, R. "Emergence of Scaling in Random Networks," *Science* (286), 1999, pp. 509-512.

Barabasi, A.L., Albert, R., Jeong, H "Scale-free Characteristics of Random Networks: The Topology of the World Wide Web," *Physica A*, 2000, pp. 69-77.

Barabasi, A.L., Jeong, H., Neda, Z., Ravasz, E., Schubert, A., Viscek, T. "Evolution of the Social Network of Scientific Collaborations," (xxx.lanl.gov/arXiv:cond-mat/0104162v1:April 10, 2001), 2001

Bollinger, T. "Linux and Open-Source Success: Interview with Eric. S. Raymond," *IEEE Computer*, 1999, pp. 85-89.

Callaway, D.S., Hopcroft, J. E., Kleinberg, J. M., Newman, M. E. J., Strogatz, S. H. "Are Randomly Grown Graphs Really Random," Santa Fe:01-05-025, 2001, pp. 1-8.

Charles, J. "Open Source: Netscape Pops the Hood," *IEEE Software*, 1998, pp. 79-82.

Edwards, J. "The Changing Face of Freeware," *Computer*, 1998, pp. 11-13.

Faloutsos, M., Faloutsos, P., Faloutsos, C. "On Power-Law Relationships of the Internet Topology," *Proceedings of the SIGCOMM'99*, Cambridge, MA, 1999, pp. 251-262.

Feller, J.F., B. "A Framework Analysis of the Open Source Software Development Paradigm," *Proceedings of the ICIS 2000*, Brisbane, Australia, 2000, pp. 58-69.

Fielding, R.T. "Shared Leadership in the Apache Project," *Communications of the ACM* (42:4), 1999, pp. 42-43.

Guare, J. *Six Degrees of Separation*, Vintage Books, New York, 1990.

Hars, A., Ou, S. "Working for free? Motivations of participating in Open Source Projects," *Proceedings of the Hawaii International Conference on Systems Sciences*, 2001.

Hecker, F. "Setting up Shop: The Business of Open-Source Software," *IEEE Software*, 1999, pp. 45-51.
Huberman, B.A., Adamic, L. A "Growth Dynamics of the World Wide Web," *Nature* (401), 1999, pp. 131.
Jin, E.M., Girvan, M., Newman, M. E. J. "The Structure of Growing Social Networks," Santa Fe:01-06-032, 2001, pp. 9.
Jorgensen, N. "Putting it all in the Trunk: Incremental Software Development in the FreeBSD Open Source Project," *Information Systems Journal* (11:4), 2001, pp. 321-336.
Jorgensen, S.E., Mejer, H., Nielsen, S. N. "Ecosystem as Self_organizing Critical Systems," *Ecological Modeling*, 1998, pp. 261-268.
Koch, S., Schneider, G. "Effort, Co-operation and Co-ordination in an Open Source Software Project: GNOME," *Information Systems Journal* (12:1), 2002, pp. 27-42.
Lawrence, D. "Internetnews Server: Inside an Open Source Project," *IEEE Internet Computing*, 1998, pp. 49-52.
Madey, G., Freeh, V., Tynan, R. "Agent-Based Modeling of Open Source using Swarm," *AMCIS2002*, Dallas, TX, 2002.
Netcraft "Netcraft.com Web Server survey," (2002:Feb. 16, 2002), 2002
Newman, M.E.J. "Clustering and Preferential Attachment in Growing Networks," Santa Fe:01-03-020), 2001, pp. 1-13.
O'Reilly, T. "Lessons from Open-Source Software Development," *Communications of the ACM* (42:4), 1999, pp. 33-37.
Ousterhout, J. "Free Software Needs Profit," *Communications of the ACM* (42:4), 1999, pp. 44-45.
Payne, C. "On the Security of Open Source Software," *Information Systems Journal* (12:1), 2002, pp. 61-78.
Pumain, D., Moriconi-Ebrard, F. "City Size Distributions and Metropolisation," *GeoJournal* (43:4), 1997, pp. 307-314.
Raymond, E.S. *The Cathedral & the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*, O'Reilly, Sebastopol, CA, 1999.
Sanders, J. "Linux, Open Source, and Software's Future," *IEEE Software*, 1998, pp. 88-91.
Scacchi, W. "Understanding the Requirements for Developing Open Source Software Systems," *IEE Proceedings - Software* (In press), 2002
Schroeder, M.R. *Fractals, Chaos, Power Laws*, W. H. Freeman and Company, New York, 1991.
Sharma, S., Sugumaram, V., Rajagopalan, B. "A Framework for Creating Hybrid-Open Source Software Communities," *Information Systems Journal* (12:1), 2002, pp. 7-25.
SourceForge "SourceForge Home," (2002:February), 2002
Stamelos, I., Angelis, L., Oikonomou, A., Bleris, G. "Code Quality Analysis in Open Source Software Development," *Information Systems Journal* (12:1), 2002, pp. 43-60.
Tjaden, B. "The Kevin Bacon Game," (2001:July), 1996
Torvalds, L. "The Linux Edge," *Communications of the ACM* (42:4), 1999, pp. 38-39.
Wang, H., Whang, C. "Open Source Software: A Status Report," *IEEE Software*, 2001, pp. 90-95.
Wasserman, S., K. Faust *Social Network Analysis: Methods and Applications*, Cambridge University Press, Cambridge, UK, 1999.
Watts, D. *Small Worlds*, Princeton University Press, Princeton, 1999.
Watts, D., Strogatz, S. H. "Collective Dynamics of Small-World Networks," *Nature* (393), 1998, pp. 440-442.
Wolf, B., Karim, R.L., Bates, J. "Hacker Survey," Boston Consulting Group, 2002.
Wu, M.W., Lin, Y. D "Open Source Development: An Overview," *IEEE Computer*, 2001, pp. 33-38.