# How Firm Initiation and Control of Projects Affects Open-Source Development

Anthony Long; Undergraduate Student, Class of 2003 (along@post.harvard.edu)
Final Essay for Sociology 91r:  Supervised Reading and Research
Department of Sociology, Harvard College, Cambridge, MA
Professor Siobhan O'Mahony; Harvard Business School, Research Supervisor
May 12, 2003

## Abstract

After witnessing the success of open-source projects such as Linux and Apache, firms have sought to appropriate the open-source development model and integrate it into their own projects. Firms face a dilemma, however, since their initiation and control of open-source projects affects the traditional open-source development model in significant ways. Once a firm takes the role of starting and guiding an open-source project, the open-source development model that attracted the firm is forever altered. This paper examines the effects commercial firms have on the open-source development model when they initiate and control open-source projects.

## Introduction

...software developed by a corporation is generally created to fill a demand that the company perceives in the marketplace....The needs of the company's customers may be completely different from the needs that drive open source developers to work on the well-known projects that exist in the open source world....As we all know, there are many open source projects that were initiated by corporations that never gained the critical mass their founders had hoped for. Mozilla and StarOffice are two good examples....But for a piece of software to truly be considered a viable open source project, it must have more than just the label of an open source license. It needs to have a thriving developer community constantly working on developing and improving it.....Open Source is not just about licenses. It's about community. And companies that want to truly leverage the open source development model are going to have to devote a substantial amount of resources to building community as well as to enhancing their own code base.
--Adam Goodman, President & Publisher of Linux Magazine (June 2001)

***

Groups of programmers connected to the Internet from around the world created what we today call open-source projects. They came together to share and modify the software they had written. It was not a novel idea, since it had been the modus operandi of programmers before the beginning of the era of the personal computer in the early 1980s. But in the late 1970s, things in the computer industry changed. Corporations began to restrict the copying, modification, and sharing of the software they had produced (Stallman, 1999: 54). They used the legal system to create a series of licenses and end-user agreements to ensure that they held tight control over the intellectual property they had produced. Angered by this change to the traditional way of doing things, Richard Stallman responded with his own set of legal licenses and agreements to uphold the sharing, copying, and modification of software, catalyzing a social movement in the process.

The movement didn't really grow in popularity until the early 1990s, when Linus Torvalds, a student at the University of Helsinki developed a kernel and integrated it with Stallman's other software tools (Rosenberg, 12). Today the integrated system he developed is called Linux, and thousands of programmers contribute to its codebase.

Attracted by the wild success of Linux and other open-source projects such as Apache, corporations began to get further involved in their development. The critical turning point for complete corporate immersion in the open-source development model came in 1998, when Netscape Communications announced its intentions to release the source code of its browser software in order for it to be developed in the same ways as Linux and other open-source projects. According to Eric Raymond, a programmer who played a major role in Netscape's decision, it was the first time, "a Fortune 500 darling of Wall Street had bet its future on the belief that *our way was right*" (Raymond, 1999: 210). The question left unanswered, though, was whether or not Netscape and the companies that later followed, could successfully merge the "programmer way" with the "firm way". Could Netscape make "our way" its way? Organizational theory teaches us that firms with material incentives and communities with intangible incentives are likely to have difficulty interacting successfully with each other. Theory can help us understand why firm initiation and control of projects might affect the outcomes of the open-source development model.

**Theoretical Context**

Peter Clark and James Wilson have argued that organizations are best understood from their incentives: "much of the internal and external activity of organizations may be explained by understanding their incentive systems" (Clark 130). They regard the incentive system as the principal variable affecting organizational behavior. In their analysis of incentives, they derive three primary motives: material, solidary, and purposive.

Material incentives refer to tangible rewards; rewards that can be exchanged for monetary value or easily converted into monetary value (Clark 134). The most explicit examples are salary and wages. Most firms have this incentive structure. Solidary incentives are described as motivations of a social nature. With these incentives, there are very few tangible rewards that can be exchanged for monetary value. Most of the rewards for working in an organization ruled by solidary incentives come from “socializing, congeniality, the sense of group membership and identification, the status resulting from membership, fun and conviviality” (Clark 134-5). Eric Raymond depicts hacker culture as a “gift culture” with reputation as the key incentive (Raymond, 2001: 80). Open-source projects fit into this solidary incentive group. The last incentive Clark and Wilson identify is purposive. Purposive incentives are also intangible, but the benefits are derived mostly from the stated goals of the group instead of from reputation (Clark 135). According to Clark and Wilson, the members of groups with purposive incentives come together to “seek some change in the status quo, not simply to enjoy one another’s presence” (Clark 136). Richard Stallman’s Free Software Foundation, which seeks to advance a political philosophy of software development beyond simple sharing of code fits well into this last incentive category.

From these three primary incentives, Clark and Wilson derive three primary organizational forms: utilitarian, solidary, and purposive. Utilitarian organizations rely mostly on material incentives. Firms are the primary occupants of this category. Solidary organizations rely primarily on social incentives. The organizations that fit into this category include serviceoriented voluntary associations (Clark 141). Open-source projects have the characteristics of solidary organizations.

Utilitarian and Solidary organizations have different incentive structures which cause them to behave in different ways. According to Clark and Wilson, organizations cooperate best when their incentives match. Since firms and open-source communities have different primary incentives, we should expect to see problems arise when firms appropriate open-source communities and integrate their methods into their own corporate strategies.

This study found several problems when firms initiate and control open-source projects. The primary problems include the categories of authority structure, property rights, and incentives. Ultimately, the challenges firms face in initiating and controlling open-source projects have the effect of detaching volunteers from projects. Firms must take cautious steps to ensure that the property rights of volunteers are respected and that they involve volunteers in the project decision-making process as much as possible to make sure that firms' efforts to benefit from the open-source method of development succeed.

## Methods

The study began with the research question: How does firm initiation and control of projects affect volunteer participation? Specifically, what effects do firms have over project authority, property rights, and community solidarity?

After examining the literature on the initiation and organization of community managed open-source projects, I formulated a hypothesis on the likely effects of firm initiation and control of open source projects. Based on my understanding of community managed open-source projects, I reasoned that firms would most significantly impact project structure, volunteer incentives, and intellectual property rights. After forming my initial hypotheses, I examined data sources to confirm or deny them. My data sources included data freely available from projects themselves (especially mailing and discussion lists), data available from the larger OS/FS community (including contributor personal web pages), an international developer survey/report commissioned by the European Union, an interview with a project leader, and observation at the Free Software Foundation. I collected project data from three commercial open-source projects and two community projects. I compared project mission, structure, intellectual property rights, and volunteer recognition systems across the five projects (Table 1). The three firm-initiated open-source projects included Sun's OpenOffice.org, Apple's Darwin, and Netscape's Mozilla. I compared my observations of these commercial open-source projects with observations of non-commercial, community open-source projects Apache and Gnome. My guiding hypothesis was that the mismatch of primary incentives between firms and communities would cause a higher degree of volunteer detachment and tension than would be found in community-initiated open-source projects.

## Community-initiated Open-Source Projects

Community-initiated open-source projects are groups of developers who collaborate via the Internet to program software applications. They consist of a core group of code repository owners who decide which contributions from volunteers located around the world get placed into the next stable release of the product. Volunteers join projects to learn new skills (FLOSS survey, Table 3), because they are passionate about software programming (Himanen), and because of the prestige that comes from fixing problems in a community setting (Raymond).

## Firm-Initiated Open-Source Projects

Commercially-initiated open-source projects are firm controlled projects involving firm employees and volunteer contributors who work through the Internet on a core technology that a firm integrates into a branded product. Firms maintain control over the code repository, project website, and discussion and mailing lists. Firms initiate open-source projects to benefit from rapid code development, peer review, and broad distribution of software.

## Comparing Firm-Initiated and Community Initiated Projects

### *Authority Structure*

The project authority structure is one of the first elements that change when firms initiate projects. Traditional, community-initiated open-source projects usually begin as the efforts of one or a small group of programmers who grow the project contributor base over time. Ultimate authority usually rests with the volunteer group that initiated the project. Since most community-initiated open-source projects have detailed, explicit, and transparent plans for volunteer inclusion in decision-making for the overall project, there seems to be little tension between competing interests. In firm-initiated open-source projects, authority and chief decision-making power ultimately rests with the firm. A firm can shut down an open-source project as easily as it can start one. Firms maintain control over the key and essential elements of open-source projects, including the main website where projects are hosted, the CVS (code) repository, and the mailing and discussion lists. Although foundations that maintain open-source projects have the same means of control available to them, there seems to be less of an issue among contributors since volunteers can be almost certain their work will not be taken and applied to non-open-source products the foundation can sell for profit. Volunteers and foundations, since they both seek no real material incentives, work well together. But in firm-initiated open-source projects, volunteers seek intangible incentives while firms seek material incentives, so there is a mismatch of motives that causes a significant degree of caution among volunteers to commit to such projects.

It is clear that authority rests with firms from the cases in this study. OpenOffice.org has a Collabnet paid employee as its overall project leader. Sun Microsystems pays for site hosting through its partnership with Collabnet. The Darwin project is controlled by Apple, which maintains the "official" code repository, requires volunteer registration, and commits paid employees to the overall project. Furthermore, OpenDarwin.org a project spin-off to foster volunteer community, has Apple paid employees as its project leaders. Finally, Netscape commits paid employees to manage the Mozilla project. This is different from the two community-initiated projects in our study, which both have foundations as the central means of authority (Table 2).

*Property*

Firms leave themselves room to make contributions proprietary and build alternative licenses to facilitate this process. OpenOffice.org uses a combination of a Sun open-source license and a Free Software Foundation license which both allow it to add proprietary software code to the open-source code it redistributes in its branded version. Apple uses its own Apple Public Source License, which attracted criticism and ignited a battle among key open-source luminaries (further detailed in the Challenges section). Netscape originally employed a Netscape Public License for the Mozilla project, but later altered it in response to criticism it received from the open-source community. It is clear from the three case studies that firms have attracted criticism and have ignited tensions among volunteers for their open-source licenses. Neither of the two community-initiated open-source projects studied had to alter their original licensing terms.

The copyright license is the key element in the open-source milieu that facilitates the cooperative development of software through its mechanisms for sharing, copying, and redistribution. It also serves as a means of guarding against free riders (O'Mahony 2003). Firms cannot afford the restrictions of licenses such as the GPL since it might require them to forego a market opportunity that called for the bundling of non-open-source code with open-source code. Foundations don't have this problem since their incentive structure is less complex. They seek to foster community and build robust products. Firms' material motives often lead them to build safeguards into their property structures that prevent them from being tied down to restrictive licenses. The consequences include a lower ability to attract and keep a community based largely on non-material incentives engaged.

*Incentives*

Firms initiate open-source projects mostly for material incentives. The constant motivation seems to be faster product development and broader distribution. Sun initiated OpenOffice.org to build an office suite base into its Open Information Architecture (Sun Microsystems, 4) with the intention of rolling it into a network-services business model:

> Sun believes in open standards and in expanding the market for product implementations based on open standards in which Sun can sell hardware and related services. Since the initial acquisition of the StarOffice software, Sun has had two main objectives for this technology:
>
> 1. To move personal productivity applications to network services
> 2. To move from a software-for-sale business model to a services-driven revenue model
>
> By engaging the energy and creativity of developers worldwide, we will accelerate the addition of innovative features and improved integration with other products. Making the source code available also enables the StarOffice software functionality to be ported to a wider range of systems (http://wwws.sun.com/software/star/openoffice/faqs.html#1q0).

Apple's Darwin project has cited similar motives:

> We believe the open source model is the most effective form of development for certain types of software. By pooling expertise with the open source development community, we expect to improve the quality, performance and feature set of our software.
>
> Secondly, we realize many developers enjoy working with open source software, and we want to provide them the opportunity to use that kind of environment while delivering solutions for Apple customers (http://developer.apple.com/darwin/ps-faq.html).

Firms get involved with open-source projects to benefit from key features of the development model such as collaborative peer review, rapid bug recognition and fixes, and broad distribution. Community foundations form for similar technical reasons, but there are not additional long-term corporate strategies motivated by material incentives involved. Foundations primarily seek to form a structure to solidify the community they have built around open-source projects (O'Mahony 2002) .

## Challenges to Firm Initiated open-source projects

### *Authority & Control*

As stated above, the first and most significant change firms make to open-source projects is to the authority structure. Open-source websites tend to serve as code banks for software and generally have the purpose of acting as the central point of contact and development for open-source projects. When firms initiate open-source projects, they control this critical piece of the puzzle. They have the final word on who gets to register for mailing lists, which code gets incorporated into the core product, and who gets to lead subprojects. Recently on the OpenOffice.org marketing email list, there was discussion about what would happen if another firm acquired Sun Microsystems. In the words of OpenOffice.org's project leader, "Sun supports not only the developers but the site hosting. Were that support to be withdrawn, for whatever reason, and not be picked up by some other megacorporation, then OpenOffice.org as such would likely dissolve" (Marketing email list 5.12.03). Volunteers who have devoted hours of their time to the project would have much to lose if the project were to suddenly disappear as a result of market forces. Sun and the volunteers at OpenOffice.org have tried to form an non-profit entity separate from Sun to guard against an event like the one depicted above, but have faced problems because of Sun legal constraints. Since Sun bases a branded, commercial product on the OpenOffice.org codebase, it is prevented from founding a nonprofit organization like other community initiated open-source projects (Exhibit A).

In his study of commitment and detachment in voluntary associations, David Knoke found a strong correlation between the extent to which volunteers are involved in the decision-making process of organizations and their commitment to such organizations. Specifically, the more volunteers are included in the decision-making process, the more committed they are to the organization (Knoke 153-4). On the other hand, "The less conducive the structural opportunities are for exercising control, the less supportive and more cut off the members feel" (Knoke, 154). Considering these findings, we should expect volunteers to feel detached from open-source projects the more they are not included in the decision making process.

All three firm-initiated open-source projects experienced some degree of friction between community and firm decision-making. In a letter to Sun Microsystems, several volunteers expressed concerns over the lack of a community council, which Sun promised to found in the initial stages of the project (please see Exhibit A). For Apple's Darwin project, a separate website and organization was formed by a partnership between the community, Apple, and the Internet Software Consortium with the mission of providing a platform for developers to exert more control over the code and release process. According to its website, OpenDarwin was formed to "further increase the collaboration between Apple and the open source community" (OpenDarwin.org). Fundamentally, OpenDarwin.org allows developers to exert control over their own website, bug tracking, and code repository in order to facilitate "increased participation by the community" (OpenDarwin.org). The changes made at OpenDarwin.org are fed back to the main Apple code repository at Apple's discretion. Although the OpenDarwin project leader declined to respond to an email requesting information on the motivations and events that led up to the creation of OpenDarwin, one can assume there must have been some agreement that it was in the best interest of the Apple open-source initiative to allow more developer control over the code repository and development process. The tension created by firm authority over projects has been pointed to as one factor involved in volunteer detachment from firm-initiated open-source projects. Mozilla's first project leader expressed his frustration in getting a critical number of outside volunteers to contribute to the project, pointing to the belief that many people were turned away from contributing because they saw Netscape as controlling the project. In his words:

> I've spent much of my time striving to convince people that mozilla.org is not netscape.com. I've told people again and again that the mozilla.org organization does not serve only the desires of the Netscape client engineering group, but rather, serves the desires of *all* contributors to the Mozilla project, no matter who they are. And that's certainly true. But the fact is, there has been very little contribution from people who don't work for Netscape (Exhibit B).

This tension between firm authority/control and community authority/control is in line with what we would expect from David Knoke and his theory of volunteer detachment. As volunteers become detached from firm-initiated open-source projects, efforts are made to bring them into a position of decision-making authority or volunteers become more detached and cease to contribute. For both Darwin and OpenOffice.org, efforts were made by firms to place volunteers in roles that allowed them to have some voice in the overall decision-making process after firms found such steps necessary to get volunteers to commit to working on projects. This is evident from the founding of OpenDarwin.org and stepped up efforts to finally implement the Community Council at OpenOffice.org. It is unclear what efforts have been made by Netscape to facilitate broad volunteer involvement in Mozilla's overall decision-making process.

These types of challenges and changes to project authority and control are not evident in community initiated open-source projects. Both Apache and Gnome have relatively open and transparent decision-making structures where volunteers can be added to foundation boards by election or other means. Firms have taken the approach of maintaining essential control of key decision-making authority by forming separate councils and organizations that contain a high percentage of firm employees. Firms seem to be reluctant to release final decision-making control to community volunteers, but are willing to hear their voices via community councils and organizations in which firm employees and volunteers cooperate. Firms might gain more volunteer commitment the more they involve volunteers in the decision-making process of the project.

### *Property Rights*

Another challenge firms face is to property rights. In community-initiated open-source projects, copyright is usually retained by contributors or jointly assigned to the contributor and the community foundation. Most community projects use their own license or adopt licenses from the Free Software Foundation. For example, Apache uses the Apache Software License, which actually allows less restrictions than the Free Software Foundation's GPL. Gnome uses the Free Software Foundation's GPL. Firms generally use a combination of their own licenses and Free Software Foundation licenses, or only their own licenses. OpenOffice.org uses a combination of Sun's SISSL license and the Free Software Foundation's LGPL and requires contributors to sign a joint copyright assignment form before contributions can be accepted. Sun's SISSL license and the LGPL allow Sun to bundle proprietary software code into their branded distribution of OpenOffice.org. Apple uses its own Apple Public Source License and requires volunteers to register their agreement to its terms before they contribute code to the main Apple code repository. Like Sun's SISSL, Apple's license allows it to bundle proprietary code with code developed through the open-source process. Mozilla uses a combination of a Mozilla license, a Netscape license, and licenses from the Free Software Foundation. While it does not require volunteers to formally assign copyright to the Mozilla.org organization, Netscape's license ensures that contributions can be incorporated to non-open proprietary distributions of Mozilla.

All three firm-initiated projects have faced criticism for their licensing strategies, and some have even changed their licenses or copyright assignment frameworks in response. Sun Microsystems used a copyright assignment form in the first stages of the OpenOffice.org project. This form forced volunteers to assign the copyright of their contributions over to Sun Microsystems. This agreement received wide criticism both from the OpenOffice.org community and outsiders, including the open-source press. Sun responded to the pressure and criticism by changing the copyright assignment to a joint copyright assignment, allowing contributors to retain rights to their work in partnership with Sun. According to Danese Cooper, Sun's Open Source Programs Manager in an interview with OpenOffice.org's project leader:

> Community member's concerns over the required assignment of copyright on community contributions to Sun convinced us to research and eventually author the Joint Copyright Assignment, which is a big improvement over the old agreement because it allows contributors to retain original rights to their work and only asks them to confer a copy of those rights to Sun.

Members of the community seemed satisfied with this change in copyright assignment. According to an article in *The Register* written by an OpenOffice.org contributor, the joint copyright assignment was " 'exactly what the volunteer community has asked for and shows a significant commitment from Sun Microsystems to the volunteer community." Cooper hoped the change to the joint copyright assignment in addition to efforts to move the community council idea forward, would help to improve volunteer involvement with OpenOffice.org:

> I think the JCA should clear up developer concerns over giving contributions to Sun-sponsored projects. I'm hoping that the JCA, together with the establishment of a Community Council (which is moving forward) and the Release Council will help those contributors who want to be more involved in the evolution of the project find entry points to deepen their involvement.

Apple Computer's licensing issues caused a divide among key players in the Open Source Initiative (OSI), an organization tasked with the mission of conferring "open-source" status onto licenses and projects. OSI co-founder Eric Raymond initially gave Apple the right to claim its Apple Public Source License as an open-source license, but was quickly criticized by co-founder Bruce Perens and others. The major problem they cited was Apple's stipulation that developers notify Apple of any modifications they made. Perens and others found issue with this clause because they believed it would hamper redistribution of software if Apple were to go out of business. The license was further criticized by Free Software Foundation founder Richard Stallman, who described it as "unacceptable" (see Exhibit C). In addition to his criticism, Bruce Perens gave a dark outlook for the future of firm initiation and involvement with open-source projects and communities:

> The needs of corporations are not necessarily those of the free software community, and it may even be the case that the twain will never meet. Open Source seems to be splitting into something I'd call "Corporate Source", semi-free programs with disclosed source but less than the full set of rights we are used to, and true Free Software as represented by the GPL, LGPL, X/BSD, and other licenses. Public discussion of this fact is essential. We may eventually have to accept that it will never be possible for corporate participation in the free software community to be as full as we would like. Contributions like the MacOS X source may end up being useless to the free software community as far as code reuse is concerned….

Mozilla was another firm-initiated open-source project that came under fire from the community. It was criticized along similar lines as Sun and Apple. Community members were concerned about the provisions of the Netscape Public License which made it possible for

Netscape to make any contributed material proprietary and then issue it under other licenses. Reacting to this criticism, Netscape set up a "more neutral scheme" using the Mozilla Public License as an alternative (Rosenberg 217).

It is clear that property rights in terms of copyright and licensing restrictions present a challenge to firms when they initiate and control open-source projects. Although Apache uses a copyright assignment that is as restrictive as Sun's original copyright assignment, it has not received any notable criticism from the open-source community. This is most likely due to the fact that the Apache Foundation seeks no material incentives that would motivate it to transfer volunteer contributions to proprietary products. This is similar to volunteers, who also seek no material incentives. It may also be due to the fact that the Apache Foundation has a transparent, clear-cut mechanism for involving volunteers in the decision-making process of the foundation.

### *Building Community*

Another tough area for firms is fostering community solidarity. Most community-initiated open-source projects are started by programmers who develop a volunteer community. Several open-source contributors have criticized firms for neglecting this important aspect of open-source projects. The publisher of Linux Magazine, in an editorial wrote:

> Open Source is not just about licenses. It's about community. And companies that want to truly leverage the open source development model are going to have to devote a substantial amount of resources to building community as well as to enhancing their own code base.

Others have also made this claim, including Richard Stallman, who, in his criticism of Apple's open-source licensing wrote:

> Overall, I think that Apple's action is an example of the effects of the year-old "open source" movement: of its plan to appeal to business with the purely materialistic goal of faster development, while putting aside the deeper issues of freedom, community, cooperation, and what kind of society we want to live in.

> Apple has grasped perfectly the concept with which "open source" is promoted, which is "show users the source and they will help you fix bugs". What Apple has not grasped--or has dismissed--is the spirit of free software, which is that we form a community to cooperate on the commons of software.

The criticisms of firms' negligence in building community raise some important points. Having a commuted and involved community seems to be an important ingredient in a successful open-source project, since without one, the essential open-source benefits of peer recognition and easy bug recognition and repair are not possible.

The implications of firm initiation and control of open-source projects are that they ultimately affect the type of work that gets accomplished. Instead of volunteer contribution to a project's core features, volunteers mostly report software bugs, work on porting software to other platforms, and localize code to native languages. Firms would benefit by focusing attention on fostering community solidarity among volunteer contributors and involving them more in the decision-making process of the firm. The benefits to firms would be more volunteer attachment and commitment.

Table 1: Project Characteristics

| **Project Attributes** | **Apache/Apache Software Foundation (Community)** | **Gnome/Gnome Foundation (Community)** | **OpenOffice.org (Commercial)** | **Darwin/OpenDarwin.org (Commercial)** | **Mozilla (Commercial)** |
|---|---|---|---|---|---|
| **Mission** | Provide a platform and legal entity for collaborative software development projects | Create "an entirely free desktop environment for free systems" | Create the leading office suite that runs on all major platforms through open file formats | Primary is to Support Mac OS X. Secondary is expectation that Darwin will become platform for stand-alone OS distribution | Provide central point of contact and community for those interested in using or improving mozilla code |
| **Year Founded** | 1995 (by the Apache Group) | 1997*(http://primates.ximian.com/~miguel/gnome-history.html) | 2000 | 1999 | 1998 |
| **License used** | ASL (Apache Software License) | GPL/LGPL | SISSL/LGPL | APSL | MPL/NPL/GPL/LGPL |
| **Initiator/Owner** | Apache Software Foundation (Present Owner) | GNOME Foundation (Present Owner) | Sun Microsystems (Initiator/Owner) | Apple Computer (Initiator/Owner) | Netscape/AOL TimeWarner (Initiator/Owner) |
| **Main Project Leader Primary Affiliation/ Employer** | Collabnet*(http://www.lyra.org/greg/ -Greg stein is the major project leader) | Gnome/Ximian *(Gnome website) | Collabnet*(resume) | Apple Computer *(even for OpenDarwin.org) | Mozilla.org |
| **Main Hosting site Owner** | Tribal Knowledge Group/Covalent.net (admin contact/domain name server from Whois search) | Red Hat/Simplemente.net (whois information) | Collabnet | Apple/Synack Communications | Meer.net |
| **Volunteer Reward/ Credit System** | Contributor page with biographies and tasks completed linked from project main pages | Credits page with developer name and key contributions linked from main page | Difficult-to-find contributor name list updated 1 year ago | OpenDarwin.org Committers list linked from main page. No findable Apple-hosted credits | Difficult-to-find alphabetical name list not linked to main page |

Table 2: Authority, Property, and Incentives

| Project Attributes | OpenOffice.org (Commercial) | Darwin/OpenDarwin.org (Commercial) | Mozilla (Commercial) | Apache/Apache Foundation (Community) | Gnome/Gnome Foundation (Community) |
|---|---|---|---|---|---|
| **Authority** | Sun/Collabnet Employee | Apple Computer | Mozilla.org/ Netscape | Foundation Board Members | Foundation Board Members |
| **Property** | Jointly held, formal joint copyright agreement | Jointly held/must register before contributing | Jointly held, no formal agreement | Formal Contributor Assignment to Apache | Jointly held, no formal agreement |
| **Initiator/ Owner Incentives** | Support open systems architecture to speed industry migration to software as services revenue model*(white paper/faqs) | Improve quality, performance, and feature set of Apple's software. Allow developers to work in a way they enjoy on products for Apple's customers | Benefit from collaborative open-source development model to further the market penetration of browser software | Form a community of developers and code repository for the Apache projects | Create user-friendly free desktop environment for free systems and provide space for implementation of Free Software user rights |
| **Volunteer Contributor Incentives** | Education, Prestige, Solutions to own problems | Education, Prestige, Solutions to own problems | Education, Prestige, Solutions to own problems | Education, Prestige, Solutions to own problems | Education, Prestige, Solutions to own problems |

Table 3: FLOSS Survey

**Free/Libre/Open Source Software: Survey and Study**

**7_1. Remembering the time you started developing and/or distributing OS/FS, what was the reason for this? (maximal four answers)**

| | | |
|---|---|---|
| to participate in new forms of cooperation | 842 | 34.54% |
| to learn and develop new skills | 1923 | 78.88% |
| to share my knowledge and skills | 1215 | 49.84% |
| to participate in the OS/FS scene | 745 | 30.56% |
| to improve my job opportunities | 582 | 23.87% |
| to improve OS/FS products of other developers | 821 | 33.68% |
| to get a reputation in the OS/FS developers' scene | 221 | 9.06% |
| to distribute not marketable software | 217 | 8.9% |
| to get help in realizing a good idea for a software product | 581 | 23.83% |
| to solve a problem that could not be done by proprietary software | 724 | 29.7% |
| to limit the power of large software companies | 463 | 18.99% |
| because I think that software should not be a proprietary product | 735 | 30.15% |
| to make money | 107 | 4.39% |
| I do not know | 46 | 1.89% |

Source: http://floss1.infonomics.nl/stats.php?id=7_1

# References

Clark, Peter B. and James Q. Wilson. *Incentive Systems: A Theory of Organizations.* Administrative Science Quarterly, Volume 6, Issue 2, 1961: 129-166.

DiBona, Chris, Sam Ockman, and Mark Stone. Open Sources: Voices from the Open Source Revolution. Sebastopol, CA: O'Reilly & Associates, Inc., 1999.

Feller, Joseph and Brian Fitzgerald. Understanding Open Source Software Development. London: Pearson Education Limited, 2002.

Fielding, Roy and James Herbsleb. *Two Case Studies of Open Source Software Development: Apache and Mozilla*. Research paper available at: www.research.avayalabs.com/techreport/ ALR-2002-003-paper.pdf. Cited 14 May 2003.

Gerth, H.H. and C. Wright Mills. From Max Weber: Essays in Sociology. New York, NY: Oxford University Press, 1958.

Himanen, Pekka. The Hacker Ethic and the Spirit of the Information Age. New York, NY: Random House, Inc., 2001.

Jones, Gareth R. *Transaction Costs, Property Rights, and Organizational Culture: An Exchange Perspective.* Administrative Science Quarterly, Volume 28, Issue 3: 1983: 454-467.

Kidwell, Roland E. Jr, and Nathan Bennett. *Employee Propensity to Withhold Effort: A Conceptual Model to Intersect Three Avenues of Research.* The Academy of Management Review, Volume 18, Issue 3: 1993: 429-456.

Knoke, David. *Commitment and Detachment in Voluntary Associations.* American Sociological Review, Volume 46, Issue 2, 1981: 141-158.

Kunda, Gideon. Engineering Culture: Control and Commitment in a High-Tech Corporation. Philadelphia, PA: Temple University Press.

Lerner, Josh and Jean Tirole. *Some Simple Economics of Open Source.* The Journal of Industrial Economics, Volume L, No. 2: June 2002: 197-234.

Lofland, John and Lyn H. Lofland. Analyzing Social Settings: A Guide to Qualitative Observation and Analysis. Belmont, CA: Wadsworth Publishing Company, 1995.

Moody, Glyn. Rebel Code: The Inside Story of Linux and the Open Source Revolution. New York, NY: Perseus Books Group, 2001.

O'Mahony, Siobhan. *Guarding the Commons: How Community Managed Software Projects Protect their Work*. Research Policy. Forthcoming, 2003.

O'Mahony, Siobhan. The Emergence of a New Commercial Actor: Community Managed Software Projects. Stanford PhD Dissertation: June 2002.

Raymond, Eric S. The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary. Sebastopol, CA: O'Reilly & Associates, Inc., 1999.

Rosenberg, Donald. Open Sources: The Unauthorized White Papers. Foster City, CA: M&T Books, 2000.

Sorensen, Aage B. "Firms, Wages, and Incentives." In The Handbook of Economic Sociology. Princeton, NJ: Princeton University Press, 1994.

Spradley, James P. The Ethnographic Interview. Orlando, FL: Holt, Rinehart and Winston, Inc., 1979.

Sun Microsystems. The OpenOffice.org Project: Foundations of Office Productivity in a Networked Age. White Paper available online. Palo Alto, CA, 2000.

Tucker, Robert C. The Marx-Engels Reader. New York, NY: W.W. Norton & Company, 1978.
Weber, Max. The Protestant Ethic and the Spirit of Capitalism. New York, NY: Routledge, 1992.

Wilson, John and Marc Music. *Who Cares? Toward an Integrated Theory of Volunteer Work.* American Sociological Review, Volume 62, Issue 5: 1997: 694-713.

Young, Robert. Under the Radar: How Red Hat Changed the Software Business—and Took Microsoft by Surprise. Scottsdale, AZ: The Coriolis Group, 1999.

## Exhibit A: Sun's Response to Community Concerns

Date: Fri, 26 Apr 2002 13:25:57 -0700
From: Danese Cooper <Danese.Cooper@Sun.COM>
Content-type: text/plain; charset=us-ascii; format=flowed
Subject: [discuss] Reponse to - On Community, Communications and Copyright letter

Dear Gianluca, Martijn, Josh, Kevin and Scott,

First of all, I'd like to thank you five for taking the time to put together a thoughtfully worded statement of your concerns with respect to copyright and communication between Sun and the OpenOffice.org community. I have been following the debate on the discussion and marketing lists and working with members of Sun's legal and OpenOffice.org teams to address as many of your concerns as possible. I'm pleased to be able to offer this communication to you today in hopes that it will clear up some of the controversies and contribute to further dialog and successful interaction between the OpenOffice.org community and Sun.

As I'm sure you all can appreciate, the de facto rules of engagement for open source projects are a special challenge to large traditional corporations. Open source is a "trial and error, learn as you go" sort of undertaking. Sun has attempted to model its policies on common practices within the larger Open Source community, but we continue to learn and welcome this opportunity to adjust to the needs of this community. My thanks as well for the professionalism of your approach to communication with Sun, which has been truly constructive.

To address your specific points:

1. The Contributor Assignment Agreement:

You seem to have a clear understanding that Sun has undertaken the collection of copyright assignments for code donations with the primary intent of making the LGPL/SISSL dual-licensing model possible. Our original contributor assignment agreement language was patterned off the best examples we could find, namely the forms used by the Free Software Foundation and Apache Software Foundation which required a simple copyright assignment to the owner of the code repository. We have been working on a new version of our standard contributor agreement to take EU laws into account and to reassure donors that we meant to inflict no hardship in collecting copyright assignments.

In the near future we plan to rollout a new Contributor Assignment to all the Sun-sponsored Open Source communities. The new Contributor Assignment requests a grant of joint ownership with respect to copyright and a limited power of attorney for the purpose of perfecting the copyright, so the donor retains all of his or her rights and merely shares them with Sun. It is our sincere hope that this new model of sharing ownership will calm everyone's concerns about copyright assignment.. To cover existing donors (who have a signed Contributor Assignments on file) Sun will re-execute any such agreements on demand.

On the issue of rejected contributions. In order for any Contributor Assignment to be valid, the contribution must first be accepted into OpenOffice.org. Rejection of a contribution means the Contributor Assignment was never perfected (or completed) with respect to the rejected contribution and so ownership reverts to the donor. This has always been the case and we will clarify this issue in the FAQs covering licensing issues.

2. Contributions other than software code:

Sun agrees that non-code donations should not require execution of the Contributor Assignment, however if non-code donations are to be hosted on a Sun-sponsored site, they must be licensed in a way that allows them to potentially be included in whatever package(s) are made available through Sun's distribution channels. We have been working to provide an Open Source Documentation License that we feel is suitable to meet the needs of non-code contributions to projects sponsored by Sun. We have drafted a new license, designed to cover Open Source documentation which will be co-created by many contributors (such as help files or FAQs). Review of this license made obvious the need for a second license designed to cover Open Source documentation which requires moderated editing (such as technical White Papers). We are in the process of drafting this second license. It is Sun's hope that these documentation licenses, when they are made available, will clear up the question of what terms apply to non-code contributions by the OpenOffice.org community.

Contributions in the form of participation on mail lists, submission of bug reports and other ephemera are covered under the terms of use and privacy policy for the website, as they have always been. These terms of use allow Sun the right to use, display and delete the submissions under a license. Such terms of use allow mail lists to exist on the website and are standard for all Sun-sponsored open source websites.

Sun is pleased to welcome community contributions and every attempt has been made to recognize them in the project. For instance, the copyright notice in the compiled binaries for version 1.0 will reference a list of individual contributors to the project at www.openoffice.org/welcome/credits.html .

3. Code under licenses other than the LGPL/SISSL:

In my experience, this is a very common question for all Open Source projects (i.e.,whether or not code licensed under a different open source license than the one used by the main code base should be accepted) and it exposes a problem within the Open Source community. Not all of the various open source licenses allow combination of code under all other open source licenses. Sun has long considered this question and for many reasons has decided to stick with a pattern of only accepting code licensed under the same license as the main code base. This practice simplifies considerations about what can or can not be included in distributions of the code base and what licensing terms apply to that end product. Separate licenses for contributions would require developers and subsequent implementors to track whatever specialized requirements flow from each license. For these reasons, we will continue to request that code donations intended for inclusion in the OpenOffice.org codebase maintained by us be submitted under LGPL/SISSL.

It is a commitment of the OpenOffice.org project that the code always remain available to both the Free Software /Open Source community AND the proprietary software community.  From a licensing perspective, the common link between those two communities is the LGPL, which allows dynamic linking of open source  code with proprietary code.  Acceptance of code licensed using the GNU General Public License ("GPL") would be in conflict with the goal of maintaining the code for all communities. We recognize that there may at times be compelling reasons to include already compiled binaries or applications with OpenOffice.org that are not licensed under the LGPL/SISSL. These situations need to be reviewed as they arise. If accepted, such binary code would need to be hosted in a special section on the OpenOffice.org web site and would need to be linked in to OpenOffice.org at runtime, not statically linked or compiled with the codebase.

4. Community Council:

As we stated in our original communications about the formation of OpenOffice.org, our first objective after launching the project was to get the StarOffice 6.0 version completed, for which completing OpenOffice.org version 1.0 is a necessary precursor.  That focus has necessitated that we move more slowly on some of the community issues (and frankly the overwhelming response of assistance from the non-developer community caught us by surprise). Sun agrees that it is time for the OpenOffice.org community to gain a greater voice in decisions about the direction and future of OpenOffice.org.  As a first step, we plan after the release of OpenOffice.org version 1.0 to establish an Advisory Release Council to manage further patches to the 1.0 codebase, discuss acceptance criteria for point releases and messaging around point releases.  For future versions of OpenOffice.org beyond 1.x, we further intend to establish a Community Council to work with Sun to assist in determining feature sets and process going forward.  The details of participation in the Community Council are still to be determined, but the intention is to bring together Sun and a group of OpenOffice.org community members who have the desire and expertise to positively contribute to the process of driving OpenOffice.org forward.

A word about Foundations:  At the inception of OpenOffice.org, Sun announced its intention to potentially establish a foundation to drive the future of OpenOffice.org. Since that time we've done quite a lot of research regarding open source foundations.  It turns out that successfully establishing such a foundation is problematic for a traditional company planning to ship a branded product based on a codebase which it also hosts for the benefit of the Open Source community. (eg., Mozilla.org is still legally part of Netscape).  For this reason, Sun will not establish a foundation for OpenOffice.org.  It is our hope that by responding to the concerns of the community, we will end up with more trust and productivity in the OpenOffice.org community.

It is our hope that by responding to the concerns of the community with a modified Contributor Assignment, new Open Source Documentation

Licenses and establishing an Advisory Release Council and Community Council we will end up with more trust and productivity in the OpenOffice.org community.

In conclusion I'd like to again thank you gentlemen for your well-considered communication with me, and for the opportunity to expose some of the work we've been doing at Sun to evolve OpenOffice.org to meet community needs. I don't want to leave you with the impression that I'm the one doing this work alone. Zaheda, Louis, Stefan Taxhet, Martin Hollmichel and the entire OpenOffice.org team at Sun and CollabNet have been involved in pushing this agenda through Sun, and the work didn't start with your letter but has been proceeding since we foresaw the need. As I mentioned earlier, Open Source projects are a special challenge to large traditional corporations. Working to meet that challenge, although occasionally frustrating, has been worthwhile and rewarding to me as an Open Source advocate, and I hope to you also as capable spokespeople for the OpenOffice.org community.

Sincerely,

Danese Cooper
Sun Open Source Programs Office

26 April 2002

Exhibit B: Resignation of Mozilla's Project Leader

# resignation and postmortem.

---

April 1st, 1999 will be my last day as an employee of the Netscape Communications division of America Online, and my last day working for mozilla.org.

Netscape has been a great disappointment to me for quite some time. When we started this company, we were out to change the world. And we did that. Without us, the change probably would have happened anyway, maybe six months or a year later, and who-knows-what would have played out differently. But we were the ones who actually did it. When you see URLs on grocery bags, on billboards, on the sides of trucks, at the end of movie credits just after the studio logos -- that was us, we did that. We put the Internet in the hands of normal people. We kick-started a new communications medium. We changed the world.

But we did that in 1994 and 1995. What we did from 1996 through 1999 was coast along, riding the wave caused by what we did before.

Why? Because the company stopped innovating. The company got big, and big companies just aren't creative. There exist counterexamples to this, but in general, great things are accomplished by small groups of people who are *driven*, who have unity of purpose. The more people involved, the slower and stupider their union is.

And there's another factor involved, which is that you can divide our industry into two kinds of people: those who want to go work for a company to make it successful, and those who want to go work for a successful company. Netscape's early success and rapid growth caused us to stop getting the former and start getting the latter.

In January 1998, Netscape hit one of of its blackest periods -- the first round of layoffs. It was quite a wake-up call. Netscape, darling of the computer industry, the fastest-growing company in the world, was not invincible.

More concretely, this was when we realized that we had finally lost the so called ``browser war.'' Microsoft had succeeded in destroying that market. It was no longer possible for anyone to sell web browsers for money. Our first product, our flagship product, was heading quickly toward irrelevance.

And then the unexpected happened: the executive staff decided to release the source code. I won't re-hash the history of the creation of the mozilla.org project, but suffice it to say that, coming as it did only two weeks after the layoffs, it was a beacon of hope to me. Here was

Netscape doing something daring again: here was the company making the kind of change in strategy that I never thought they'd be able to make again. An act of desperation? Perhaps, but still a very interesting and unexpected one. It was so crazy, it just might work. I took my cue and ran with it, registering the domain that night, designing the structure of the organization, writing the first version of the web site, and, along with my co-conspirators, explaining to room after room of Netscape employees and managers how free software worked, and what we had to do to make it work.

At this point, I strongly believed that Netscape was no longer capable of shipping products. Netscape's engineering department had lost the single-minded focus we once had, on shipping something useful and doing it fast. That was no longer happening. Netscape was shipping garbage, and shipping it late.

And daring move or no, this was not going to change: Netscape no longer had the talent, either in engineering or management, to ship quality products. The magic was gone, as the magicians had either moved on to more compelling companies, or were having their voices lost in the din of the crowd, swamped by the mediocrity around them.

The Netscape I cared about was dead.

But I saw mozilla.org as a chance to jettison an escape pod -- to give the code we had all worked so hard on a chance to live on beyond the death of Netscape, and chance to continue to have some relevance to the world.

Beyond that, I saw it as a chance for the code to actually *prosper.* By making it not be a Netscape project, but rather, be a public project to which Netscape was merely a contributor, the fact that Netscape was no longer capable of building products wouldn't matter: the outsiders would show Netscape how it's done. By putting control of the web browser into the hands of anyone who cared to step up to the task, we would ensure that those people would keep it going, out of their own self-interest.

But that didn't happen. For whatever reason, the project was not adopted by the outside. It remained a Netscape project. Now, this was still a positive change -- it meant that Netscape was developing this project out in the open, in full view of the world, and the world was giving important and effective feedback. Netscape made better decisions as a result.

But it wasn't enough.

The truth is that, by virtue of the fact that the contributors to the Mozilla project included about a hundred full-time Netscape developers, and about thirty part-time outsiders, the project still belonged wholly to Netscape -- because only those who write the code truly control the project.

And here we are, a year later. And we haven't even shipped a beta yet.

In my humble but correct opinion, we should have shipped Netscape Navigator 5.0 no later than six months after the source code was released. But we (the mozilla.org group) couldn't figure out a way to make that happen. I accept my share of responsibility for this, and consider this a personal failure. However, I don't know what I could have done differently.

I can come up with a litany of excuses and explanations for why we are so late (heaven knows I've been making these excuses to the media for half the lifetime of the project.) Some of them are:

**Excuse #1:**

It's a really large project, and it takes a long time for a new developer to dive in and start contributing.

**Excuse #1a:**

Because of this, what happens is, someone will try to make a small change, find that it's taking them longer than a few hours, and will give up and do something else instead.

**Excuse #2:**

People only really contribute when they get something out of it. When someone is first beginning to contribute, they especially need to see some kind of payback, some kind of positive reinforcement, right away. For example, if someone were running a web browser, then stopped, added a simple new command to the source, recompiled, and had that same web browser *plus their addition*, they would be motivated to do this again, and possibly to tackle even larger projects.

We never got there. We never distributed the source code to a working web browser, more importantly, to *the web browser that people were actually using.* We didn't release the source code to the most-previous-release of Netscape Navigator: instead, we released what we had at the time, which had a number of incomplete features, and lots and lots of bugs. And of course we weren't able to release any Java or crypto code at all.

What we released was a large pile of interesting code, but it didn't much resemble something you could actually use.

**Excuse #3:**

The code was just too complicated and crufty and hard to modify, which is why people didn't contribute. This was a believable excuse for a while, which is why, six months ago, we switched from the old layout engine to the new layout engine (Gecko/Raptor). By being a cleaner, newly-designed code base, so the theory went, it was going to be easier for people to understand and contribute. And this did get us more contributors. But it also constituted an almost-total rewrite of the browser, throwing us back six to ten months. Now we had to rewrite the entire user interface from scratch before anyone could even browse the web, or add a bookmark.

**Excuse #4:**

It didn't contain a mail reader. There is surely a large class of users who would be interested in working on Communicator that are less interested in Navigator, but we never really found that out, since we never shipped the source code to communicator (for a number of reasons, none very good, some downright pathetic.) Now, as a result of the Gecko/Raptor rewrite, the mail/news reader is being rewritten as well. Maybe it will even ship someday.

**Excuse #5:**

> Netscape failed to follow through on their own plans. During 1998, Netscape sunk a huge amount of engineering effort into doing the 4.5 release: working on a dead-end proprietary code base, the source of which would never be released to the world, and would never benefit from open source development. This was a huge blow to the Mozilla project, since for the first half of the year, we weren't even getting full-time participation from Netscape.
>
> This isn't even so much an excuse as a stupid, terrible mistake, considering we should have learned our lessons about doing parallel development like this in the past, with the abortive ``Javagator'' project.

The worst part about all this is, for the last year, I've spent much of my time striving to convince people that mozilla.org is not netscape.com. I've told people again and again that the mozilla.org organization does not serve only the desires of the Netscape client engineering group, but rather, serves the desires of *all* contributors to the Mozilla project, no matter who they are. And that's certainly true. But the fact is, there has been very little contribution from people who don't work for Netscape, making the distinction somewhat academic.

Now, to be fair, in this first year, we did do some very good things:

- We showed the world how to operate a large software project out in the open. Whatever else happened, we did maintain a high level of communication between geographically and organizationally separate contributors and other interested parties. We transitioned from a secretive and proprietary development model to a very public one. We showed that it can be done.
- Though we didn't get a whole lot of participation in the form of source code, we did get a lot of feedback about the directions the software was going. And the right feedback at the right time can easily be far more valuable than source code. By doing development out in the open, and ``living in a fishbowl,'' I believe that Netscape made better decisions about the directions of development than would have been made otherwise.
- We released the source code to a number of ancillary tools, such as our bug system, source-control interface, and build tools. These are very good (and complete!) tools in their own right. Though they were critical to us in the development of Mozilla, and we created them in support of Mozilla, they are not tied to Mozilla, and others are finding them useful with their own non-Mozilla-related projects. These tools, and the development model they represent, are a valuable contribution in their own right.
- And merely by being who we are and doing what we did, we played a big part in bringing the whole open source development model to the attention of the world at large. We didn't start the mainstream media interest in open source (Linux did that, mostly), but I think we did legitimize it in the eyes of a lot of people, and we did tell the story very well. Lending the Netscape name to this software development strategy brought it to the attention of people who might otherwise have dismissed it.

But despite all this, in the last year, we did not accomplish the goals that I wanted to accomplish. We did not take the Mozilla project and turn it into a network-collaborative project in which

Netscape was but one of many contributors; and we did not ship end-user software. For me, *shipping* is the thing.

Perhaps my goals were unreasonable; perhaps it should have been obvious to me when we set out on this project that it would take much longer than a year to reach these goals, if we ever did. But, it wasn't obvious to me then, or now. These are the goals I was aiming for, and they have not yet been met.

And so I'm giving up.

The Mozilla project has become too depressing, and too painful, for me to continue working on. I wanted Mozilla to become something that it has not, and I am tired of fighting and waiting to make it so. I have felt very ineffectual, and that's just not a good feeling.

For those of you who choose to continue, I wish you all the best of luck.

I must say, though, that it feels good to be resigning from AOL instead of resigning from Netscape. It doesn't really feel like quitting at all. I was the 20th person hired at Mosaic Communications Corporation (*All Praise the Company*), and of those twenty, only five remain. The company I helped build has been gone for quite some time. We, Netscape, did some extraordinary things. But we could have done so much more. I feel like we had a shot at greatness, and missed.

My biggest fear, and part of the reason I stuck it out as long as I have, is that people will look at the failures of mozilla.org as emblematic of open source in general. Let me assure you that whatever problems the Mozilla project is having are not because open source doesn't work. Open source does work, but it is most definitely not a panacea. If there's a cautionary tale here, it is that you can't take a dying project, sprinkle it with the magic pixie dust of ``open source,'' and have everything magically work out. Software is hard. The issues aren't that simple.

Jamie Zawinski, 31-Mar-1999

## Exhibit C: Richard Stallman's Criticism of Apple's Open-source License

- *To*: bcollins@debian.org
- *Subject*: Re: Apple and Open Source
- *From*: Richard Stallman <rms@gnu.org>
- *Date*: Sat, 20 Mar 1999 03:01:16 -0700 (MST)
- *CC*: jmlb2@hermes.cam.ac.uk, spi-general@lists.spi-inc.org, debian-private@lists.debian.org, bruce@hams.com
- *In-reply-to*: <19990317075826.A12008@visi.net> (message from Ben Collins onWed, 17 Mar 1999 07:58:26 -0500)
- *References*: <Pine.SOL.3.95q.990317085506.11499E-100000@red.csi.cam.ac.uk> <19990317075826.A12008@visi.net>
- *Reply-to*: rms@gnu.org

---

After studying Apple's new source code license, the APSL, I have concluded that it falls short of being a free software license. It has three fatal flaws, any of which would be sufficient to make the software less than free.

* Disrespect for privacy.

  The APSL does not allow you to make a modified version and use it for your own private purposes, without publishing your changes.

* Central control.

  Anyone who releases (or even uses, other than for R&D) a modified version is required to notify one specific organization, which happens to be Apple.

* Possibly of revocation at any time.

  The termination clause says that Apple can revoke this license, and forbid you to keep using all or some part of the software, any time someone makes an accusation of patent or copyright infringement.

  In this way, if Apple declines to fight a questionable patent (or one whose applicability to the code at hand is questionable), you will not be able to have your own day in court to fight it, because you would have to fight Apple's copyright as well.

  Such a termination clause is especially bad for users outside the US, since it makes them indirectly vulnerable to the insane US patent system and the incompetent US patent office, which ordinarily could not touch them in their own countries.

Any one of these flaws makes a license unacceptable.

If these three flaws were solved, the APSL would be a free software license with three major practical problems, reminiscent of the NPL:

* It is not a true copyleft, because it allows linking with other

files which may be entirely proprietary.

* It is unfair, since it requires you to give Apple rights to your changes which Apple will not give you for its code.

* It is incompatible with the GNU GPL.

Of course, the major difference between the NPL and the APSL is that the NPL *is* a free software license. These problems are significant in the case of the NPL because the NPL has no fatal flaws. Would that the same were true of the APSL.

At a fundamental level, the APSL makes a claim that, if it became accepted, would stretch copyright powers in a dangerous way: it claims to be able to set conditions for simply *running* the software. As I understand it, copyright law in the US does not permit this, except when encryption or a license manager is used to enforce the conditions. It would be terribly ironic if a failed attempt at making a free software license resulted in an effective extension of the range of copyright power.

Aside from this, we must remember that only part of MacOS is being released under the APSL. Even if the fatal flaws and practical problems of the APSL were fixed, even if it were changed into a very good free software license, that would do no good for the other parts of MacOS whose source code is not being released at all. We must not judge all of a company by just part of what they do.

Overall, I think that Apple's action is an example of the effects of the year-old "open source" movement: of its plan to appeal to business with the purely materialistic goal of faster development, while putting aside the deeper issues of freedom, community, cooperation, and what kind of society we want to live in.

Apple has grasped perfectly the concept with which "open source" is promoted, which is "show users the source and they will help you fix bugs". What Apple has not grasped--or has dismissed--is the spirit of free software, which is that we form a community to cooperate on the commons of software.