AN ABSTRACT OF THE THESIS OF

Claire E. Gilbertson for the degree of Master of Science in Computer Science presented on August 30, 2011.

Title: Gender and Participation in Early Stages of the Free/Open Source Software Joining Process

Abstract approved: _____
Carlos Jensen

Free/Open Source Software (FOSS) is a powerful development paradigm for creating software. Increasingly more FOSS projects, like Firefox and Android, are integrated into mainstream technology. It is important that FOSS projects serve its diverse user base well. Several surveys have found that existing FOSS communities are very homogenous populations and made up of mostly men. There are significantly fewer women participating in FOSS when compared to the percentage of women in computing in general.

FOSS communities have a large amount of turnover and must have a continual influx of new developers to keep the project alive and thriving. When a new developer wants to learn more about a project, report a bug or has a question about the project, he or she typically posts a message on a mailing list. Mailing lists are the primary form of communication with FOSS communities, and are the first place where new users interact with the existing community.

Building on previous research, we examined one of the first steps of joining a FOSS project, subscribing to a mailing list, and studied posting statistics of females during the early stages of the process. In particular, we explored 6 FOSS projects:

Buildroot, Busybox, Jaws, Parrot, Uclibc, and Yum. We found that 8.27% of FOSS list subscribers are women and that significantly fewer posters (6.63%) are women. Women lurked on a list slightly less than men before replying, and remained subscribed for slightly less time to a mailing list.

©Copyright by Claire E. Gilbertson
August 30, 2011
All Rights Reserved

Gender and Participation in Early Stages of Free/Open Source Software Joining Process

by

Claire E. Gilbertson

A THESIS

Submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented August 30, 2011
Commencement June 2012

Master of Science thesis of Claire E. Gilbertson presented on August 30, 2011

APPROVED:

Major Professor, representing Computer Science

Director of the School of Electrical Engineering and Computer Science

Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Claire E. Gilbertson, Author

ACKNOWLEDGEMENTS

I am very grateful for those who have helped me throughout my graduate career. I want to profusely thank the following people.

This thesis would not have been possible without the academic guidance from Dr. Carlos Jensen. I am very thankful for the time and patience he spent working with me on this project.

I would also like to thank my committee members Paul Paulson, Dr. Ron Metoyer and Dr. Branscum for kindly participating.

Dr. Jon Dorbolo and the rest of TAC have supplied a flexible working environment that has allowed me to keep up with my graduate studies. In particular, Jon and Weiwei have offered excellent support throughout the thesis-writing process. I have learned so much from them and will greatly miss their company, advice and kindness.

None of this would be possible without the assistance from the OSL and the project communities who graciously offered data about their FOSS projects. The HCI research group has offered great ideas and a welcoming learning environment. I really appreciate their support throughout this project.

Most importantly, I want to thank my friends and family, especially my Mom, Aunt Nellie and Uncle Kent and Mike for their emotional support, feeding me, babysitting my dog, playing Thursday night pinochle and encouraging me to excel academically.

Thank you!

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

Page

LIST OF FIGURES

Figure Page

LIST OF TABLES

Table Page

Gender and Participation in Early Stages of the
Free/Open Source Software Joining Process

1. Introduction

About the same number of women and men receive bachelor degrees in the science and engineering fields (NSF 07-315, 2007). However, a divide is seen in the area of Computer Science, where only 25% of IT workers are female, and females earn 18% of bachelor's degrees (NWCIT, 2009). In Free and Open-Source Software, or FOSS communities, an even smaller percentage, approximately 2%, of participants are females (FLOSSPOLLS, 2006). It is important to understand the reasons behind the gender differences in FOSS in order to create policies and strategies to encourage greater diversity in FOSS communities. As more companies employ FOSS software, and as FOSS projects diversify to serve a broader variety of needs and populations it is necessary to move from homogenous design to one that supports more types of users. Fostering greater diversity in FOSS communities is one way of potentially supporting a more diverse user group.

FOSS is a powerful development paradigm that can create software to rival proprietary or closed-source software solutions. FOSS software touts freely available source code that is available for redistribution and modification for customized uses. Asynchronous communities of users from around the world typically work together on FOSS projects to meet a common goal. The Open Source Initiative (OSI) describes FOSS as a transparent, powerful peer review process. "The promise of open source is better quality, higher reliability, more flexibility, lower cost, and an end to predatory vendor lock-in." (OSI, 2011) For a FOSS project to thrive it needs to maintain an

active user-base and encourage more people to contribute to the project. FOSS developers are rarely under contract or obligation to work, and there is a high turnover rate of FOSS developers (Ye and Kishida, 2003). One of the topics examined in this thesis is the process new users go through in order to join a FOSS community. When a user is interested in checking out a community, the standard process is to start by lurking, or observing the community without contributing or posting, and examining documentation about the project. Many times users read mailing list archives or logs to get a sense of the culture and expected methods of communication. We focus on mailing list interaction because this is where the majority of communication occurs and especially because this is where newbies, or new users, and the community first interact.

Multiple surveys have found that FOSS communities are not very diverse: About 98% of the community is male. The majority of contributors are between the ages of 18 and 25, and from North America and Western Europe (FLOSSPOLLS, 2006, David et al., 2003, Ghosh et al., 2002, Lakhani and Wolf, 2002).

> "Many FOSS communities and advocacy groups have taken steps to address some of the perceived inequalities and have started experimenting with different programs aimed at recruiting and retaining underrepresented groups. While many groups acknowledge the importance of different types of diversity (primarily gender and cultural diversity), most efforts today are directed at recruiting and the retention of female contributors." (BSD, OSI policy)

Why do so few women participate in FOSS projects, and where do these differences emerge, with regard to the joining process? Some have surveyed contributors and analyzed possible reasons for the low number of female participants.

Scott King's thesis explored mailing list posters' introductory messages and responses. Almost 80% of newbies received a positive reply to their first post, and those who received a timely response were more likely to continue participating on the mailing list (King, 2010). King investigated the effect of different poster demographics, including gender, and found that messages from men and women were treated similarly (tone, helpfulness, percent receiving replies), yet significantly fewer women posted (2.68%) when compared to men. King did not look at the time users lurked on lists before posting, which may be a factor related to the demographics of a user. Nonnecke and Preece examined the lurking period that users needed in order to feel comfortable contributing to a technical discussion and found they needed anywhere from weeks to months, but they did not study this in conjunction with the gender of the user (Nonnecke and Preece, 2000). Building on King's research, this thesis takes a look at the first step in the joining process, examining data for subscribers to FOSS mailing lists –both those who post messages and those who do not. We will look at an additional data point, subscription logs, which add another dimension to Open Source community statistics. In this manner, we can examine the differences between posters and non-posters, and examine whether we are losing women between the step of deciding to learn about a project (joining a mailing list) and deciding to become active (posting a message). This attempts to close part of the gap of demographic data known about FOSS project members and those considering joining a project. Specifically this thesis will investigate the following research questions, in relation to FOSS mailing lists:

1. Given that women participate at a disproportionally low rate, even when considering the lower participation of women in computing, can we determine how early in the FOSS joining process these differences emerge?

2. Once subscribed to a FOSS mailing list, are women as likely to participate (post) as men?

3. Do females post (participate) as frequently as males on these lists?

4. Do women lurk (silently observe the community) longer than men before posting?

5. Do men and women participate (subscribe) for equal amounts of time?

The following sections will discuss background and related work (Section 2) and provide a description of the methodology used to collect and analyze data (Section 3). Next, we describe our findings and results in section 4. Finally, in section 5, we discuss our research questions in relation to our results and in the conclusion (Section 6), we review our major findings and present future research suggestions.

2. Literature Review

2.1. What is Free/Open Source Software

FOSS is software whose source code is freely available, freely redistributable and modifiable, and complies with certain license agreements. In addition, FOSS licenses, "must not discriminate against any person or group of persons" or hold restrictions against "specific fields of endeavor." The Open Source Initiative categorizes software as Open Source if and only if it meets the following criteria:

1. Free Redistribution
The license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license shall not require a royalty or other fee for such sale.

2. Source Code
The program must include source code, and must allow distribution in source code as well as compiled form. Where some form of a product is not distributed with source code, there must be a well-publicized means of obtaining the source code for no more than a reasonable reproduction cost preferably, downloading via the Internet without charge. The source code must be the preferred form in which a programmer would modify the program. Deliberately obfuscated source code is not allowed. Intermediate forms such as the output of a preprocessor or translator are not allowed.

3. Derived Works
The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software.

4. Integrity of The Author's Source Code
The license may restrict source-code from being distributed in modified form *only* if the license allows the distribution of "patch files" with the source code for the purpose of modifying the program at build time. The license must explicitly permit distribution of software built from modified source code. The license may require derived works to carry a different name or version number from the original software.

5. No Discrimination Against Persons or Groups
The license must not discriminate against any person or group of persons.

6. No Discrimination Against Fields of Endeavor
The license must not restrict anyone from making use of the program in a specific field of endeavor. For example, it may not restrict the program from being used in a business, or from being used for genetic research.

7. Distribution of License
The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.

8. License Must Not Be Specific to a Product
The rights attached to the program must not depend on the program's being part of a particular software distribution. If the program is extracted from that distribution and used or distributed within the terms of the program's license, all parties to whom the program is redistributed should have the same rights as those that are granted in conjunction with the original software distribution.

9. License Must Not Restrict Other Software
The license must not place restrictions on other software that is distributed along with the licensed software. For example, the license must not insist that all other programs distributed on the same medium must be open-source software.

10. License Must Be Technology-Neutral
No provision of the license may be predicated on any individual technology or style of interface.

(OSI, 2011)

FOSS software is created by online, asynchronous communities. Typically these groups are unpaid, only 30% of developers are paid by companies who use, promote or have some other interest in the software (David et al., 2003, Lakhani and Wolf, 2002). Many community members are both users and contributors. There is great variety in projects, including highly technical projects such as the Linux Kernel and

end-user applications like the Android OS, Wikipedia, and rivals to expensive proprietary software such as Open Office and GIMP.

2.2. Joining FOSS projects

2.2.1. Motivation

Contributors cite many reasons for joining a FOSS project. Some are interested in expanding their experience coding or other skills, some are paid to work on a project, and some are interested in pursuing FOSS ideals (Lakhani and Wolf, 2002, Robles et al.,2001, Ghosh et al., 2002, David et al., 2003). Often a passionate user starts a project because he or she has a particular interest or need that could be met by a FOSS application (Raymond, 1992). In Dr. Tim Budd's Open Source Software class at Oregon State University, he suggests that students start by finding a project that is interesting, "Just find whatever interests you... Select a category that interests you... There's lots of stuff available." (Budd, 2009) Humanitarian projects are an expanding FOSS category that attracts developers with a passion to help others around the world, such as through disaster relief projects or medical records system (Hfoss, 2007).

A majority of FOSS surveys have found that contributors were interested in improving their programming skills. Working on FOSS projects allows users of any age, education, or experience level to gain valuable experience working on very large systems within a structured team environment. For example, students who want experience to fill a resume may be interested in a FOSS *job*. Or a current employee in

the IT field may want to change roles, but needs experience first in order to apply for or make an employment change.

Another reason FOSS developers cite for motivating their work is the ideology of FOSS itself. Some are very passionate about offering alternatives to proprietary software (Krogh, et. al., 2003, David et al., 2003).

Most projects involve a social aspect that draws and helps retain members. Mailing lists and IRC channels specific to a project are mixed with off-topic conversations and close-knit groups work together.

2.2.2. Roles in FOSS projects

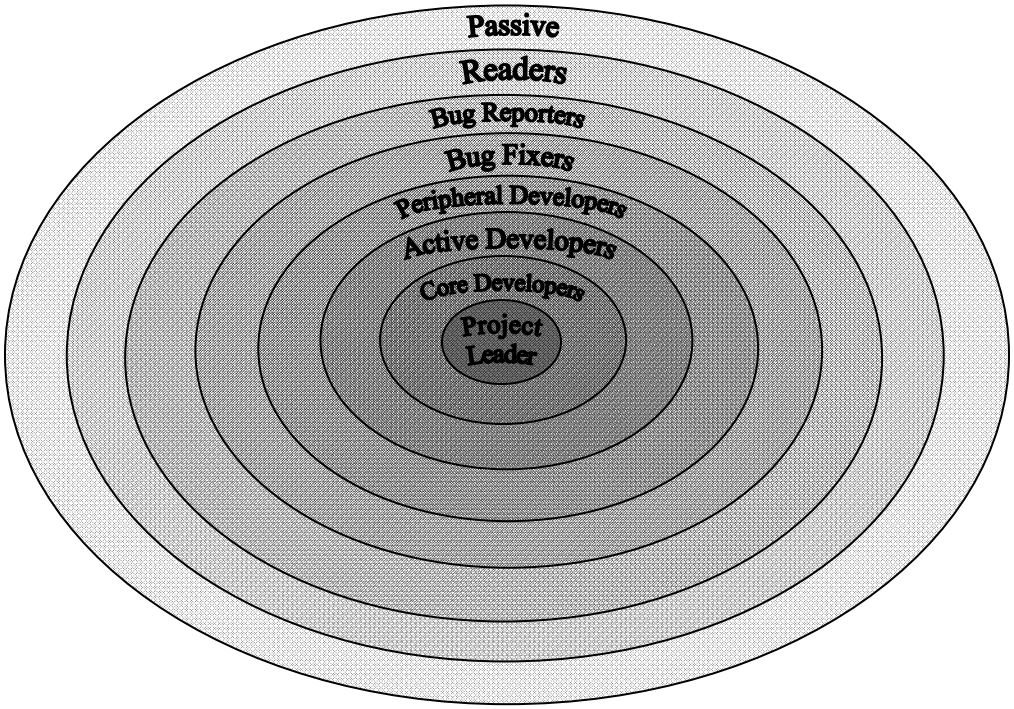
FOSS communities are hierarchical in nature and the role migration and user distribution of its members is described using the "onion" model developed by Ye and Kidhida (Ye & Kishida, 2003). The Onion model shown in Figure 1 illustrates the various role titles of users in FOSS projects. FOSS members typically start on the outer edges of the circle and move closer to the center as they become more involved.

Newbies, or new users, often begin on the outer layers of the onion as passive users or lurkers not yet participating in the community. They may have downloaded the software and registered on a mailing list or IRC channel without contributing anything. This layer makes up the majority of FOSS users (Ye & Kishida, 2003).

The next layer includes active users known as readers who try to learn about the system by reading through documentation, source code and are generally learning the

model used by the particular project. This study focuses on the new users who are passive and may or may not move on to other levels.

Figure 1: Onion model of FLOSS joining process



From readers, users move on to become bug reporters who test the project and locate bugs. However, these users do not fix bugs, that is left up to bug fixers.

Bug fixers solve issues reported by others or fix bugs they find themselves. The difference between bug reporters and fixers is that bug fixers must understand at least a portion of the source code in order to fix the issue.

The next role is peripheral developer . Such users add new features, but contribute intermittently for short periods. As these users contribute more frequently they

become active developers and regularly contribute new functionalities, fix bugs, and help maintain or develop the project significantly.

Closest to the inner core in this onion model are core members. They have typically been involved with the project for a long time and are highly responsible for coordinating the project and guiding other users involved. Core members make significant contributions to the project. Sometimes this group evolves to take over the duties of the project leader . Often the project leader originally created the project and is involved with the overall development path. A famous project leader is Linus Torvalds, the chief architect of the Linux kernel (Wikipedia, 2011).

There are many ways to transcend these roles and Jensen and Scacchi studied three large FOSS project -Mozilla, Apache, and NetBeans- and found that there are numerous paths to take within a FOSS project. This study found that these projects had a clear hierarchal scheme and that the, “presence of corporate and non-profit organizations as the core of OSS organizations, employing project members in a full time fashion has become common.” (Jensen and Sacchi, 2007) It is possible that volunteers are not eligible for all positions; some are for paid developers only, and some reserved for the project founders or people with very specific qualifications. Compensated employees may take a more direct path to the core developer role.

2.3. FOSS demographics

A well-known survey led by David et al. in 2003 asked a large number of developers about their involvement with FOSS projects. From this survey, we can

assemble some basic FOSS demographic statistics, such as gender, occupation, education, nationality, and other interesting facts. It is important to acknowledge the usefulness of this data set as well as its shortcomings. This survey asked developers questions, which does not include lurkers or other users not self-selecting as in a developer role. In addition, this survey was widely publicized in popular FOSS circles online in both English and German, which may skew the type and number of responders. That said, 1588 developers from 65 countries took this survey, predominantly English-speaking developers from Western Europe (52.7%) and North America (27.1%) participated in the study. Nearly all of the participants were males (98.4%) and half were between the ages of 23 and 33, but had a large range of 11 years to 69 years old. Developers were highly educated: 36.7% held graduate degrees, 6.2% had professional levels of education and 36.3 had undergraduate degrees. Nearly 1/5 or 19.4% of developers had only a high school diploma. Most (67.8%) developers were employed and 28.8% were students.

Participants stated many reasons for joining FOSS projects. 77.8% of respondents considered it “very important” or “important” to give back to a community after using a FOSS system. 57.2% “wanted to interact with like-minded programmers” and the majority were interesting in promoting FOSS development, ideals and providing alternatives to proprietary software. 68.7% wanted to become a “better programmer” and only 7.2% joined because their employers wanted them to “collaborate in open source development.” Many other reasons including reputation, having a hobby,

opposing closed-source or proprietary software, social reasons and affordability were also cited as reasons for joining a FOSS project. (David, et. al., 2003)

Scott King focused on new users instead of developers. (King, 2010) King found that (out of all users –both developers and newbies) that 26.5% were non-US users and 73.5% were from the US or a non-specified nationality. King found that 2.68% of posters were female. He studied the first posts made by new users and how they were first accepted (or replied to) by existing members. After examining the newbie's tone, nationality and gender, he found statistically that they received equally prompt replies. King found an alarming number of flaming or ill-willed posts and made several suggestions about the negative effect on minorities or women lurking on the mailing lists. "Thus, while OSS participants were generally polite to newbies, it is possible that newbie expectations and perceptions of politeness could be colored by how the regulars engage with each other." (King, 2010)

2.4. FOSS communication

Mailing lists are the primary way that FOSS users and developers interact with one another (Gutwin, 2004). This method of communication allows everyone to participate asynchronously, keep up with new developments or changes, bounce ideas back and forth, encourage like-minded social groups and discussions about the FOSS project or other topics. "There is a strong culture of 'making it public' where developers are willing to answer questions, discuss their plans, report on their actions, and argue design details, all on the mailing list." (Gutwin, 2004)

Mailing lists are a simple technology, and does not require special technical expertise or application-specific knowledge. They also keep an exact record of public discussions and document the development of a project or community. “Mailing lists allow people to find out who the experts are, simply by initiating a discussion: because the messages go to the entire group, the ‘right people’ will identify themselves by joining the conversation.” (Gutwin, 2004). These characteristics make mailing lists a great source of information about FOSS groups since the communication is all in one “place. For this thesis, we have gathered subscription logs for FOSS mailing lists in order to see not only who is using the list, but also who is lurking. Krogh determined, “developers-to-be would quite often lurk around (observe) the project and its mailing list(s) before actively participating. This lurking period would last anywhere on the order of weeks to months and was needed before they felt comfortable contributing to a technical discussion.” (Krogh, 2003) Lurking is useful for learning about a project. Most communication is public to all subscribers and so many practices, expectations, or methods of operation can be assimilated through lurking. Scott King hypothesized that lurkers may be drawn to a project or pushed away based solely on observation of mailing list interactions.

There are other text-based forms of communication that are popular with FOSS projects such as IRC channels, blogs, wiki, forums, bug tracking systems, social media and others (Cedeno, 2010). These other text-based forms tend to augment mailing lists communication and serve other purposes like documentation, social get-togethers and project FAQs. They are also more difficult to study in terms of collecting data about

lurkers since subscription logs may not apply to these situations or do not have as clear of boundaries regarding user's joining data.

3. Methodology

3.1. Data Collection

The Open Source Lab (OSL) hosts over one hundred and fifty FOSS projects (OSL, 2011). According to their website they "collaborate with contributors and distribute software to millions of users globally" (OSL, 2011) The OSL graciously acted as agents for us in requesting FOSS project's permission to use various mailing lists along with corresponding user subscription logs hosted by them. The MBOX or mailing list files are available via an online interface. However, the subscription logs contain sensitive user data, which is not generally published. We studied the available lists and asked an administrator at the OSL to pass on requeststo selected projects. Predominantly we looked at the more active projects that had many users and messages. Project maintainers or groups self-selected to allow the OSL to share their subscription logs with us. Nearly all of the projects we approached gave permission to use their lists, and only a few did not respond to the request or declined use.

We also asked numerous other communities (not hosted at the OSL) for access to their MBOX files and subscription logs. We received a wide variety of responses; many communities were willing to share their data but did not use mailing lists, did not keep subscription logs or did not track the data. Some were wary about sharing their user's information, even though we pledged to secure individual's data, and others outright attacked us for asking for the data. Without a complete set of message files and corresponding logs, we could not use any other project data offered to us or use it for comparison.

In general the lists we collected are all from mature, highly-technical projects. They span at least 500 days and have a range of number of subscribers between 73 and 944 per mailing list.

3.2. Project Descriptions

We used data from six FOSS projects. Of these six we selected eleven mailing lists and subscription logs: *Buildroot*, *Busybox*, *Jaws*, *Jaws-announces*, *Jaws-bugs*, *Jaws-commits*, *Jaws-developers*, *Jaws-general*, *Uclibc*, *Yum*, *Yum-devel*, and *Parrot*. Table 1 shows the length of each subscription log.

Table 1: The time period of each mailing list subscription log

Number of Users & Subscription time period studied					
List		Number of subscribers	Beginning	End	Time Span
Buildroot		944	Nov 20 09:00:01 2008	Oct 12 15:30:22 2010	16589.51 hours (~691 days)
Busybox		695	Nov 20 09:00:01 2008	May 18 13:15:14 2010	13059.25 hours (~544 days)
Jaws	announces	73	Nov 12 17:07:29 2007	Nov 02 15:16:07 2010	26061.14 hours (~1085 days)
	bugs		Nov 02 17:00:35 2007	Aug 30 16:59:18 2010	25511.98 hours (~1063 days)
	developers		Nov 12 17:08:33 2007	Oct 07 03:37:23 2010	25425.48 hours (~1059 days)
	general		Nov 12 17:08:53 2007	Jul 02 15:18:36 2010	26079.15 hours (~1086 days)
	commits		Nov 12 17:08:09 2007	Nov 03 09:17:43 2010	26079.16 hours (~1087 days)
Parrot		698	Jul 30 20:36:41 2008	May 16 15:09:08 2010	23754.54 hours (~989 days)
Uclibc		428	Dec 04 09:00:02 2008	May 18 09:00:01 2010	12719 hours (~529 days)
Yum		360	Sep 26 10:02:55 2008	May 13 20:04:18 2010	14246.71 hours (~594 days)
Yum-devel		112	Sep 26 02:07:57 2008	May 18 00:45:33 2010	14374.63 hours (~599 days)
All lists combined		Total: 3310	N/A	N/A	Average: 14374.63 hours (~599 days)

3.2.1. Buildroot

This project helps users install Linux on an embedded system. Specifically Buildroot gives users a set of tools necessary to compile code for a particular system. “Buildroot is a set of Makefiles and patches that makes it easy to generate a complete embedded Linux system. Buildroot can generate any or all of a cross-compilation toolchain, a root filesystem, a kernel image and a bootloader image. Buildroot is useful mainly for people working with small or embedded systems, using various CPU architectures (x86, ARM, MIPS, PowerPC, etc.): it automates the building process of your embedded system and eases the cross-compilation process.“ (Buildroot, 2011)

Buildroot only has one mailing list and it is very active with a mix of activity including commits, questions, bug reports and patches. They advertise the #uclibc Freenode IRC channel as a place for help too. Buildroot uses bugzilla to track bugs. It is a very technical project.

3.2.2. Busybox

This project merges numerous UNIX utilities commonly found in GNU fileutils, shellutils and others (OSL, 2011). These utilities are combined into one small executable file and it is modular, which allows for customization of a project. The project aims to minimize the size of these utilities so they can be used on small or embedded systems (Busybox, 2011).

This is a highly technical project with only one mailing list, *busybox*, and it is the main source of communication and suggested tool for communicating with the community.

3.2.3. Jaws

Jaws is a Framework and content Management System (CMS) written in PHP and using the "model, View, and Controller" (MVC) design pattern. It allows developers to write their own modules, called "gadgets" to customize their website. It seems that this project is much smaller than the others examined and recently has lost much of its activity. We included this project since it is smaller than others are and appeared to cater to a wider user base. Also the data for this project was readily available and the maintainer was willing to share the subscription lists. During the time period we examined the logs the project was more active. This project has multiple mailing lists, and for our purposes, we combined all of the messages and subscription logs in order to make better comparisons. We looked at *Jaws*, *Jaws-announces*, *Jaws-bugs, Jaws-commits*, *Jaws-developers*, and *Jaws-general.*

This project has numerous mailing lists whereas the other projects only have a few to encompass the same topics, which Jaws has broken into numerous lists. They are also using trac to manage bugs and have a Freenode IRC channel #jaws. It is still a technical project since it encourages users to develop their own modules, but it is less so than the other projects we examine. Anyone with web authoring skills would find it possible to install this tool and use its interface.

3.2.4. Parrot

Parrot is a virtual machine application built to compile byte code for dynamic languages. Currently it supports or is working to implement translators for Tcl, Javascript, Ruby, Lua, Scheme, PHP, Python, Perl 6, APL, and .NET languages. Virtual machines are helpful for creating applications that work on a variety of computer setups. This project serves a user-base of developers and is highly technical.

Parrot's website directs users to the parrot-dev mailing list for development and discussion, and other documentation sends new users to parrot-users, but this list is practically unused. Another posted communication channel is an IRC channel (#parrot, hosted at irc.parrot.org) (Parrot, 2011).

3.2.5. Uclibc

This project's name actually starts with μ (the greek letter "mu") but for our purposes we substitute the letter u. μClibc stands for "The microcontroller C library." (Uclibc, 2011) It is a smaller alternate to the GNU C Libraryl and almost all applications supported by glibc are compatible. Therefore it is an appropriate alternative to use on tablets or embedded systems. It saves space by refactoring the code to eliminate redundancy, decreasing performance and at some level sacrificing features. (Uclibc, 2011)

Uclibc is related to Busybox and Buildroot and most of their websites include links and/or share IRC channels. Their target user base is probably very similar and their tools interact well with each other. The Busybox and Uclibc tools are used on

small, embedded linux-based systems where space requirements are important and Buildroot offers the ability to compile both on the same system. They can be used separately of course too. Uclibc has two mailing list –one for discussion and development (Uclibc) and another for source commits (Uclibc-cvs) which is dedicated to diff files for bug patches and other code changes. We chose to examine the list for discussion and development as this is where new users are most likely to first post and interact with the community.

3.2.6. Yum and Yum-devel

Yum or "yellowdog updater modified" is a package management system or a "collection of software tools to automate the process of installing, upgrading, configuring, and removing software packages for a computer's operating system in a consistent manner" (Wikipedia, 2011). Yum specifically works with RPM-based systems such as RedHat Enterprise, Fedora and CentOS Linux distributions. Yum automatically finds dependencies and determines what needs to happen in order to install packages (Yum, 2011). Their website touts that their software, "makes it easier to maintain groups of machines without having to manually update each one using rpm.

This is a highly technical project that has four mailing lists: rpm-metadata, yum, yum-commits, and yum-devel. They also have a live chat area on IRC channel #yum on irc.freenode.org. We chose to examine yum-devel and yum since these mailing lists are more active and include activity from a variety of users, including newbies. The

commits list mostly has code modifications and updates with few questions or other communication. The rpm-metadata list was not available at the time.

3.3. Data Parsing: Subscriber Attributes

Using the MBOX documentation from QMAIL (Qmail Documentation, 1998) we created a Java program to parse the MBOX files and corresponding subscription logs into csv files with the following data (if available):

- Email address
- First name
- Last name
- Subscribe date (if any)
- Unsubscribe date (if any)
- Time on mailing list or (Unsubscribe – Subscribe) date
- Number of posts
- Gender
- Date of first post
- Time spent on the list before first post or (First post – subscribe) date
- Last Post
- Frequency of posts or [(Last Post-First Post) / (number of Posts)]
- List (used when combining data from multiple lists)

The program iterated over users on the subscription logs for each MBOX file and counted the number of posts made by that person. When the users signed up for the list they could choose to add a first and/or last name in addition the required email address. We needed this information in order to determine the gender of the participant. When available we parsed this information from the logs.

When this was not available, we attempted to extract a name using the email address as a guide. We used pattern matching to find possible names using the following patterns:

- First.last@ …
- First_last@....
- First-last@...

If these schemes did not match then we added the entire username portion of the email address to the first name field. We needed all possible first names listed in order to determine the gender of each user. If users had the exact same name, but a different email address, we combined their information into one data point.

We recorded the date the user subscribed and unsubscribed to the mailing list using the subscription log. Some users subscribed and unsubscribed to a list multiple times, and in these cases, we treated the first subscription as the primary join date and the very last un-subscription as the primary unsubscribe date. We did not consider the intermediate subscription activity. Some users did not have a subscribe or unsubscribe date in the subscription log. For these cases we assigned a default join or leave date, based on the earliest and latest activity on the subscription list. Since we only have data for approximately two years for each list, some long-time subscribers did not join or leave during that time period. For this study, we examined newbie behavior and lurking statistics within the joining process so these experienced subscribers were of less interest and therefore the default join or leave date designation is appropriate. We used the difference between the subscription data points to calculate the total time spent on the list or (*unsubscribe date – subscribe date*) in terms of hours.

For each user on the subscription list we examined the corresponding MBOX file and counted how many posts, if any, he or she contributed. We did not thread the posts or group them in any way; each post whether a reply or a new topic was counted as its own entity. For each poster, we recorded the date of their first and last post. From these data points, we can determine the amount of time each lurked before posting, and also their posting frequency (if he or she posted more than once).

Using data from the US Census, we matched names to lists of the most common female and male names. We identified 666 users using this process. Some names are used for both women and men, such as Alex, Robin, or Morgan. In these cases, we looked at the frequency of use for each gender for each name. If there was a disproportionate use in one gender, we assigned the user to that gender. For example, Alex is ranked as the 63rd most common name for males in the U.S., and 990th for females. Therefore, all Alex's were assumed to be male. In cases where the rankings were close, we left the user in an "unknown" category. Next, we manually looked at names to filter out obvious "not a name" such as gobeavs2003@yahoo.com or identify possible names like zhangweiwei@onid.orst.edu that may not make it on to the list of most common US names, or names that did not follow the pattern of (first, last). The names not marked as "not a name" were shown to other raters, researchers and international students, via a webpage that displayed email addresses, and possible first and last names. The raters could choose a gender, mark the email address as "not a name", label it as an "unknown" for ambiguous names or skip the name. We asked these raters to only assign a gender in cases where they were 100% certain, and skip

the name otherwise. Most of the names initially unidentified were of Asian or Middle Eastern origins. In the end, we identified 1594 users as either male or female, and were left with 975 unidentified users. In total, adding unknown and "not a name" together, we find that 41.66% of subscribers were unidentifiable. While this is unfortunate, we believe this represents a good effort and a significant and representative sample of the overall community.

Within this dataset are a lot of extreme values; many users contribute little and few users contribute a lot. For each mailing list, we ordered the users by number of posts and if there was a difference greater than ten times the previous user's number of posts, we excluded the user from our set. We did this to prevent a handful of very frequent posters from skewing our statistics. Table 2 shows the number of users excluded from each mailing list.

Table 2: Number of outliers excluded from each data set.

List	**Number of excluded users**			
	Males	**Females**	**Unknowns**	**Total Excluded**
Buildroot	0	1	0	1
Busybox	0	0	0	0
Jaws	2	0	2	4
Parrot	0	1	0	1
Uclibc	0	1	0	1
Yum	0	0	4	4
Yum-devel	0	0	4	4
All lists combined	2	3	10	15

In addition to treating each list separately, we also combined all of the data in order to perform statistical analysis and compare data across FOSS mailing lists. As

some of the projects are commonly used jointly (Busybox, Buildroot, and Uclibc) or the lists serve the same project (Yum-devel and Yum for example) it is possible that users are involved in more than one list. For comparison purposes, we did not combine these user statistics into one data point and instead left each user's role on a list as a separate user

4. Results

This section explores our research questions and displays our findings in relationship to them. Each section shows the data parsed by list, gender and relevant summary columns. First, we begin by looking at the gender of the subscribers. Secondly, we examine the time subscribers spend lurking before their first post. Next, we study the gender of subscribers who make at least one post and of those who post more than once, the frequency of their posts. Finally, we examine the amount of time users subscribe to the mailing lists.

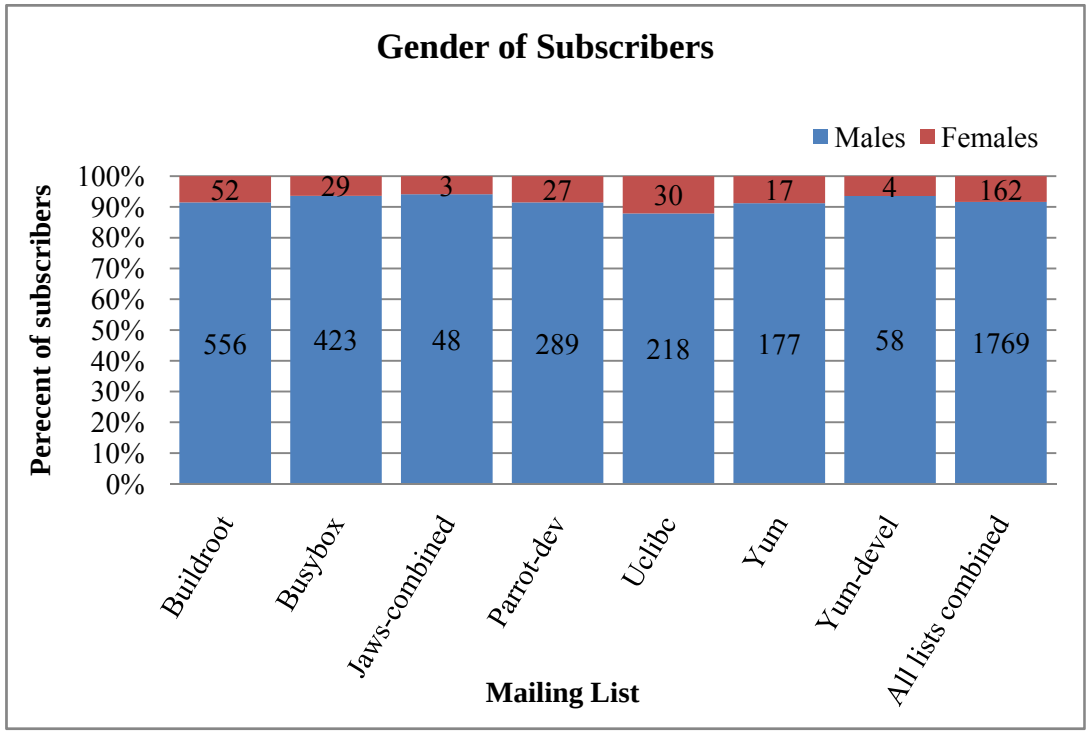
4.1. Research Question 1: Gender of subscribers

Given that women participate at a disproportionally low rate, even when considering the lower participation of women in computing, can we determine how early in the FOSS joining process these differences emerge? We know from the work of King that by the time we get to posting, only approximately 3% of posters are women. One of the earliest, traceable parts of joining a FOSS project is subscribing to a mailing list. In order to answer this question, we counted the number of women and men who subscribed to each mailing list. When we combine the users from all of the lists, we find 1769 males and 162 females subscribed to the mailing lists. 91.73% of all subscribers are male and 8.27% of subscribers are female, a more than 50% drop when compared to the expected number (the 20% rate for women in IT). Table 3 and Figure 2 present these results.

Table 3: The number of females, males and unknowns who subscribed to each mailing list. It also shows the total number of subscribers and genders found.

Gender of Subscribers				
List	Number of males	Number of females	Number of unknowns	All subscribers
Buildroot	556	52	336	944
Busybox	423	29	243	695
Jaws	48	3	22	73
Parrot	289	27	382	698
Uclibc	218	30	180	428
Yum	177	17	166	360
Yum-devel	58	4	50	112
All lists combined	1769	162	1379	3310

Figure 2: Graphical representation of the data as described in Table 2. This figure shows the relative percentages of women and men subscribed to each mailing list.



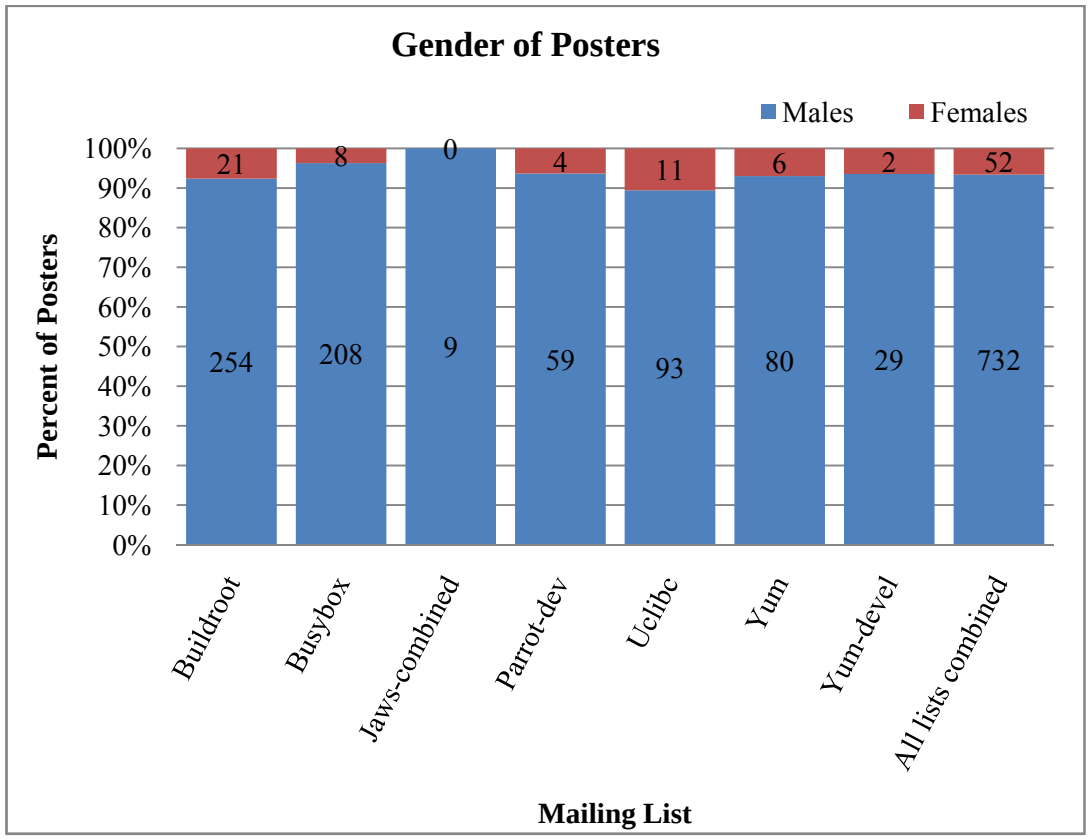
4.2. Research Question 2: Gender of Posters

Once subscribed to a FOSS mailing list, are women as likely to participate (post) as men? We counted the number of posts made by females and males. Table 4 and Figure 3 display the number of women, men, and persons of unknown gender per list and for the entire set of mailing lists.

Table 4: This table shows the gender of posters for each list and also the raw numbers or men and women across all lists.

Gender of Posters				
List	Number of males	Number of females	Number of unknowns	Average subscribers
Buildroot	254	21	157	432
Busybox	208	8	115	331
Jaws	9	0	1	10
Parrot	58	4	47	109
Uclibc	93	11	83	187
Yum	80	6	73	159
Yum-devel	29	2	23	54
All lists combined	731	52	499	1282

Figure 3: This figure corresponds to the data described in Table 3. It graphically represents the percentage of posters who are male and female for each list. Also labeled are the raw numbers of males and females on each list.



The percentage of women posters ranges from 0% to 10.58% of the total number of identified subscribers. We found on average that 6.63% of posters are females. This is a statistically significant decrease from the expected value of 8.37% of subscribers ($\chi 2 = 5.30$, p-value = .0213). 110 or 67.90% of women never posted after joining a mailing list. In comparison, 1065 or 59.30 % of men never posted after joining a mailing list.

4.3. Research Question 3: Posting frequency

Do females post (participate) as frequently as males on these lists? In order to determine the number of hours between posts, we examined the time between a user's first and last post and divided this number by the number of posts for that particular user (Equation 1). We looked at 563 users who posted at least twice in order to calculate a frequency. Table 5 and Figure 4 show the average posting frequency for each list, and for the combined data set.

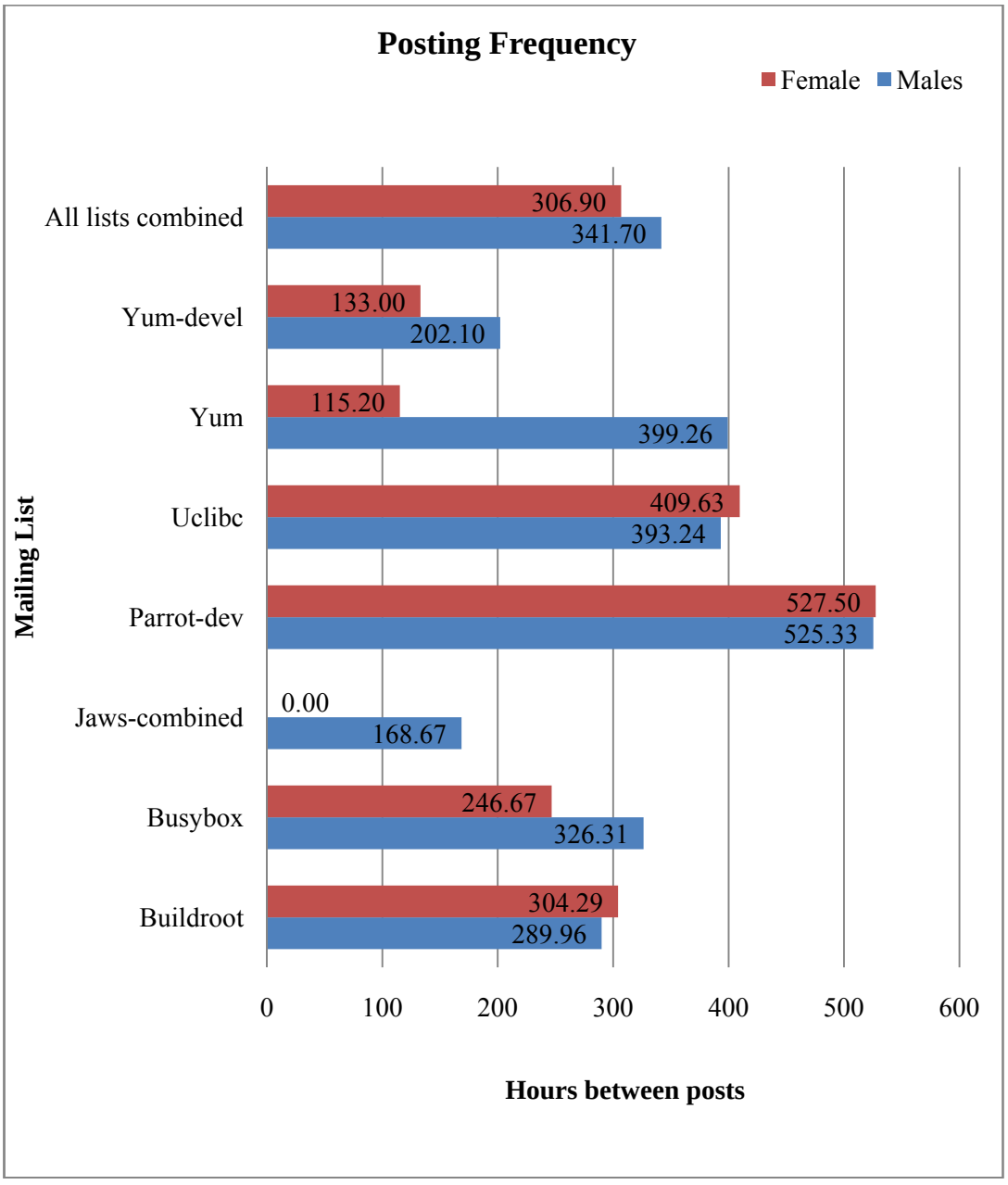
Equation 1: The equation used to calculate the average number of hours between posts.

$$\frac{1}{n}\left[\sum_{user=1}^{user=n} \frac{(Last\ Post_{user} - First\ Post_{user})\ hours}{Number\ of\ Posts_{user}}\right]$$

Table 5: This table describes the average posting frequency of males and females, per mailing list.

Average Posting Frequency					
List	Males (hours/post)	Females (hours/ post)	Unknowns (hours/post)	Δ *M-F*	Average (hours/post)
Buildroot	289.96	304.29	320.73	-14.32	301.03
Busybox	326.31	246.67	321.74	79.64	322.76
Jaws	168.67	N/A	N/A	N/A	168.67
Parrot	525.33	527.50	490.61	-2.17	512.47
Uclibc	393.24	409.63	825.59	-16.39	565.80
Yum	399.26	115.20	749.36	284.06	541.60
Yum-devel	202.10	133.00	722.53	69.10	419.83
All combined	341.70	306.90	495.17	34.81	395.91

Figure 4: The average posting frequency for males and females for each mailing list. This figure also shows the average posting frequency for all of the mailing lists combined.



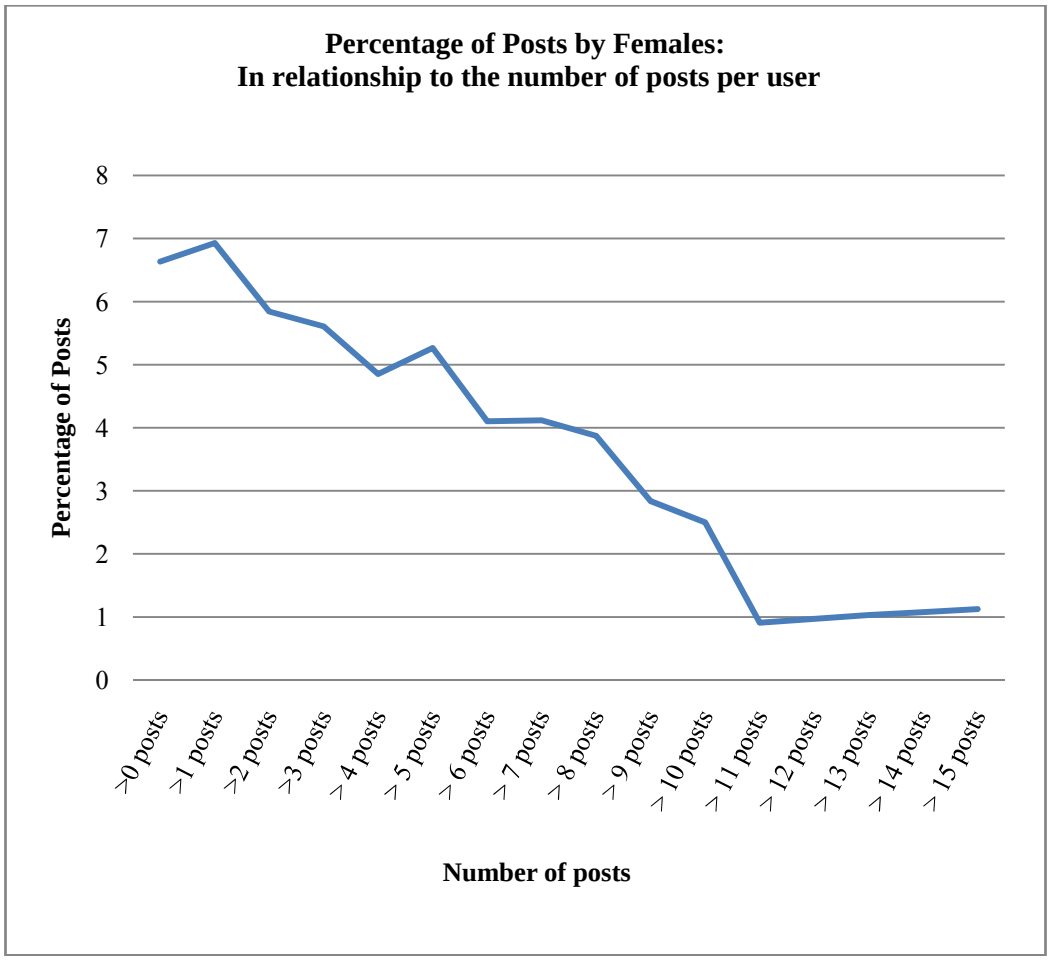
The majority of our lists (excluding Jaws and Yum) have women and men posting about as often. Statistically we did not find any significant difference in the data collected. We broke this set of data down into more categories by looking at users who posted at least once, more than once, more than twice, etc. Our findings are outlined in Table 6**.**

Table 6: This table describes the number of users who posted at least X number of times, where X varies from 0 to 15 posts. We also show the percentage of males and females in each category.

Quantity of Posts by Gender						
Number of Posts	**Number & percent of gender**				**Percentage of all posters**	
	Males		**Females**		**Males**	**Females**
>0 posts	732	100.00	52	100.00	93.37	6.63
>1 posts	524	71.58	39	75.00	93.07	6.93
>2 posts	403	55.05	25	48.08	94.16	5.84
>3 posts	320	43.72	19	36.54	94.40	5.60
> 4 posts	255	34.84	13	25.00	95.15	4.85
> 5 posts	216	29.51	12	23.08	94.74	5.26
> 6 posts	187	25.55	8	15.38	95.90	4.10
> 7 posts	163	22.27	7	13.46	95.88	4.12
> 8 posts	149	20.36	6	11.54	96.13	3.87
> 9 posts	137	18.72	4	7.69	97.16	2.84
> 10 posts	117	15.98	3	5.77	97.50	2.50
> 11 posts	109	14.89	1	1.92	99.09	0.91
> 12 posts	102	13.93	1	1.92	99.03	0.97
> 13 posts	96	13.11	1	1.92	98.97	1.03
> 14 posts	92	12.57	1	1.92	98.92	1.08
> 15 posts	88	12.02	1	1.92	98.88	1.12

We see that females make up about 6% of posters who submit between 1 and 3 posts, about 4% who submit between 4 and 8 times, about 2% who submit 9 or 10 times and 1% submit more than 10 times. Figure 5 shows a graphical representation of the same data. It is interesting to see a linear trend in the data, which suggests the percentage of each gender who post *X* number of times does not differ much. Statistical tests show that the proportion of genders who post X number of times is statistically significant ($\chi 2 = 30.346$, p-value: 0.0107).

Figure 5: This figure categorizes users who post a certain number of times (between 1 and 15 or more posts). It shows the percentage of posts made by males and females in each category.



4.4. Research Question 4: Lurkers

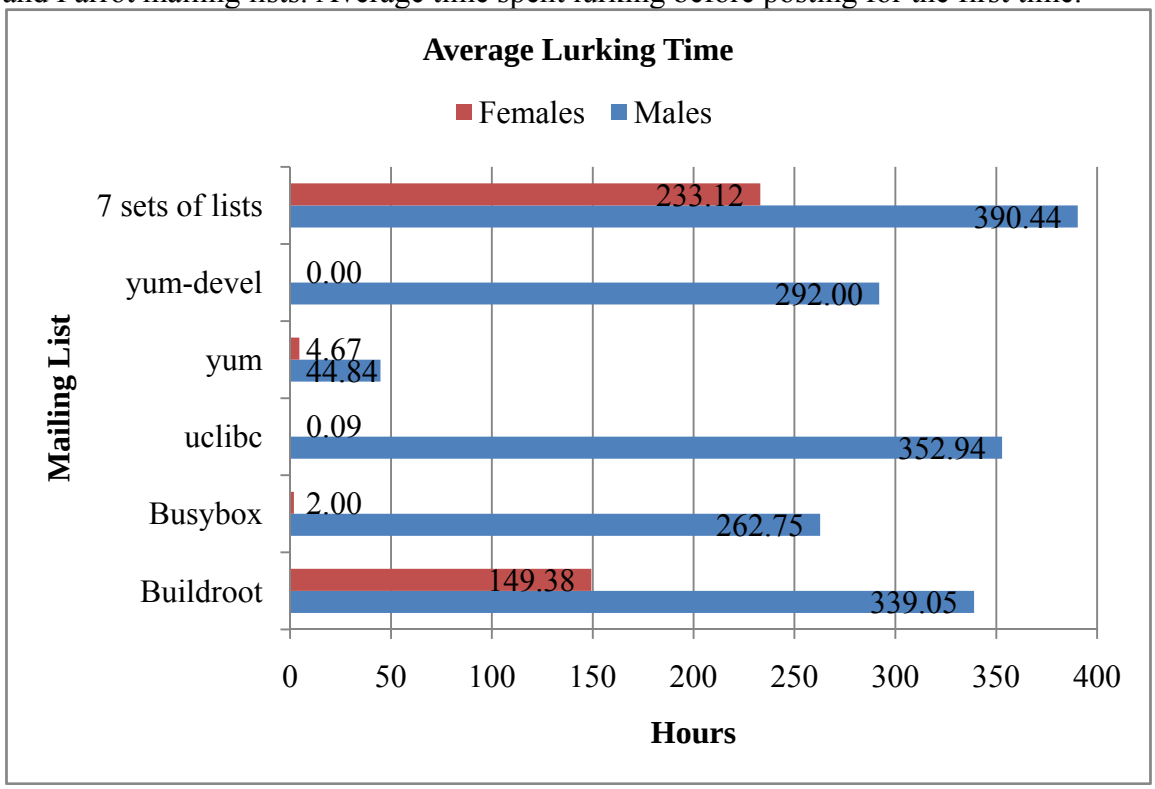
The time spent observing a group before contributing can be a formative experience and possibly encourage or discourage a user from joining or asking a question. We recorded the time difference between a user's first post and when he or she subscribed to the mailing list. Table 7 and Figure 6 describe the time spent lurking for each mailing list and the combined lists.

Table 7: Average time males, females and unknowns spent lurking before posting to the mailing lists. Also shows averages of the combined data sets.

Average time spent lurking before first post					
List	**Males (hours)**	**Females (hours)**	**Unknowns (hours)**	**Δ= *M-F* (hours)**	**Total subscribers**
Buildroot	339.05	149.38	322.08	189.67	323.66
Busybox	262.75	2.00	230.85	260.75	245.37
Jaws	2097.75*	N/A*	90.00*	N/A*	1874.67
Parrot	1406.47*	2235.00*	2196.33*	-828.53*	1770.21
Uclibc	352.94	0.09	193.45	352.84	261.39
Yum	44.84	4.67	7.52	40.17	26.19
Yum-devel	292.00	0.00	58.78	292.00	181.85
All lists combined	390.44	233.12	393.82	157.33	385.37

*Note: Parrot and Jaws average lurking times are more than 2 standard deviations from the rest of the dataset, and these aren't included in figure 6.

Figure 6: Graphical representation of the data described in Table 7, excluding Jaws and Parrot mailing lists. Average time spent lurking before posting for the first time.



In most cases, for our data, women lurked less than men. Notably, no women were identified in the Jaws data set and for the Parrot mailing list, women lurked 2,235.00 hours, which is more than 50% longer than men who lurked 1,406.47 hours, however this is an extreme outlier from the rest of the dataset as the overall time spent lurking on the Parrot list is more than 2 standard deviations longer than the other lists. A t-test analysis of the two populations (excluding the Jaws and Parrot datasets) shows no statistically significant difference in the means of the lurking time (p-value = 0.72).

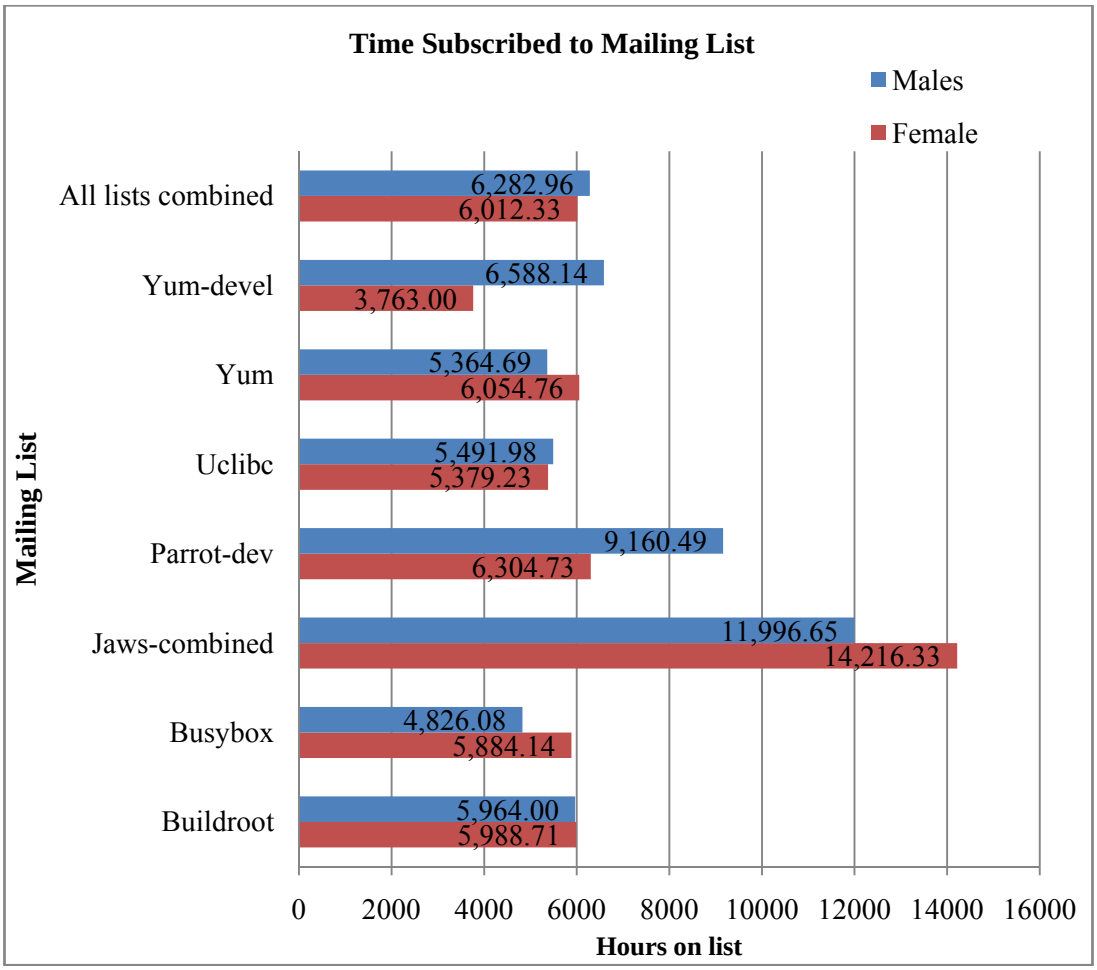
4.5. Research Question 5: Subscription Length

Do men and women participate (subscribe) for equal amounts of time? We recorded the date each user subscribed and unsubscribed from a mailing list in order to calculate the time he or she spent on the list. Table 8 and Figure 7 show the average time users spent on each list, and the combined averages.

Table 8: This table shows the average time males and females subscribed to each mailing list. It shows the calculated averages for the time all users subscribed to a list. It also shows the average values for all of the males and females across all lists.

Average time subscribed on a list				
List	Males (hours)	Females (hours)	Unknowns (hours)	All (hours)
Buildroot	5964.00	5988.71	6204.11	6050.85
Busybox	4826.08	5884.14	5004.44	4933.05
Jaws	11996.65	14216.33	9654.55	11382.02
Parrot	9160.49	6304.73	9333.80	9148.95
Uclibc	5491.98	5379.23	5455.16	5468.65
Yum	5364.69	6054.76	5776.45	5587.14
Yum-devel	6588.14	3763.00	7111.61	6717.41
All lists combined	6282.96	6012.33	6800.54	6485.42

Figure 7: This figure shows the average time females and males subscribed to a list (unsubscribe date – subscribe date) in hours. It also shows the averages of all the males from each list combined and all the females from each list combined.

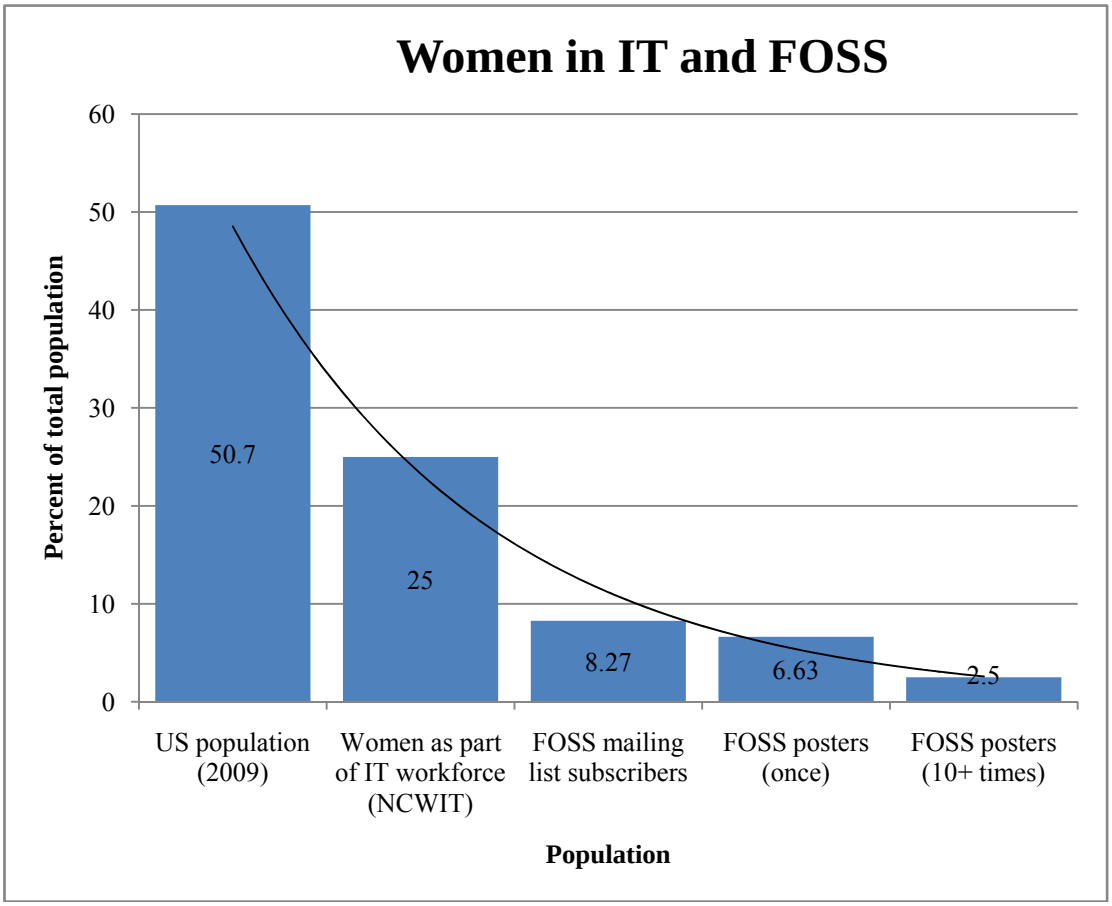


Interestingly, within the same list, women and men subscribed to a list for about the same amount of time (except Yum-devel and Parrot), but there was no statistically significant differences.

5. Discussion

In our study, we found that 8.39% of the FOSS mailing list subscribers were women. This is significantly lower than the 20% of women in computing in general. Figure 8 describes the decreasing trend of participation by women from the general population to FOSS communities.

Figure 8: The percentage of Women in IT and in FOSS. This graph shows an exponential decay of the percentage of women who make up the general population, those in the IT workforce, those subscribed to FOSS lists, those who post at least once and those who post more than 10 times.



Where and why do these gender participation rates emerge? Joining a mailing list is the first step in the FOSS joining process where we can collect information about potential FOSS contributors. There are many steps potential FOSS participants may follow before joining a mailing list: exploring a project's website, documentation, downloading the code, chatting on an IRC channel and exploring forums, messages boards or wikis. Perhaps there is one point in even earlier processes where females are turned away from a FOSS project. Of those who subscribe to a mailing list, we found that 110 or 67.90% of Women never post, and 1065 or 59.30% of men who subscribe to a list never post. Statistically we found that 6.63% of posters are women. There are many fewer women who are posting in comparison to men posting, and women are more likely to never post.

Scott King found about 3% of posters were women, however his focus was on a variety of attributes and gender was not the key focus. As a consequence his data had a much larger number of unknown gender participants, which may have skewed the data. We spent more time manually identifying users as male, female or unknown. Other studies have found that between 1.5% - 2% of code-contributors are female. These studies surveyed FOSS developers who self-selected to participate, whereas our study did not exclude any user from a FOSS mailing list. We know that the joining process is complex and the time and experience it takes to move into a developer role excludes newbies who most likely start in another role. In addition, we found that number of women who post at least a certain number of times (between 1 and 10 or more posts) declines to just over 1% of postings as the number of posts increases,

which is in agreement with the number of female developers found by other studies. In the last ten years there has been a push to increase the amount of diversity, specifically women involved in CS, and perhaps this has influenced more women to participate in FOSS projects. The major FOSS studies were performed between 2001 and 2006 and perhaps the number of women in FOSS has since increased due to these recruiting measures.

In our study, on average, males posted every 341.70 hours and women 306.90 hours. This encourages us to believe that women are not universally forced away from FOSS conversations. There are many situations in FOSS where women have felt uncomfortable, excluded or specifically targeted in a demeaning way (Fisher and Margolis, 2002). If all women were being forced away, we would expect women to post less frequently than their male counterparts. What is interesting is that along every step of the joining process, we seem to be losing a disproportionate number of women. However, given the small sample of women, it is difficult to draw statistical conclusions about the causes.

The subscription logs all varied in the length of time covered. In particular, Jaws and Parrot covered about a thousand days, and the other lists covered less than 700 days. We found that, on average, users on these two lists subscribed longer than users on other lists. On nearly all of the lists women subscribed for slightly less time than males, however we did not find any statistically significant correlation between gender and time spent. An issue that we did not examine was the time between a user's last post and his or hers unsubscribe timestamp. Perhaps this would lend more information

about a user's messages on a list. Since some lists did not have exact subscribe or unsubscribe dates, the time estimates may be less accurate than we wished.

After examining posting statistics, we find that only the average number of posts by women is statistically significant decrease. We also see that the number of women who keep posting declines in a significant manner when compared with the number of men. We did not examine the type of messages posted and it is possible that many users were not interested in joining the FOSS project, but rather asked one-time questions. It would be interesting to add a message-type category to this line of investigation. In addition, the projects examined are highly technical and a mix or comparison with less-technical projects may yield other results.

6. Conclusion

FOSS projects need a persistent influx of new members in order to thrive and replace members as they transition out of the community. It is important to encourage a diverse population of new contributors in order to better FOSS projects, increasingly aimed at serving a wider audience. The proportion of women who participate in FOSS is very low in comparison with the number of women in computing in general. Other studies found that women make up approximately 2% of FOSS contributors, and 20% of the general computing population. Why do so few women join FOSS projects and is there a certain point at which they make the decision not to join? In an attempt to answer this question we examined the joining process in more detail and specifically focused on the first step of joining a FOSS community –subscribing to a mailing list.

We studied eleven mailing lists and corresponding subscription logs from five FOSS projects, with a combined 3,310 users, of which 1,769 were males, 162 females and 1,379 unknown. We found 8.39% females subscribed to these mailing lists, which when compared to the percent of women in computing, is less than half of the expected 20% number. Only 6.63% of posters were women. The only significant difference we found, in terms of behavior of men and women, was in the average number of posts made. Also significant was the proportion of women who made frequent posts. The percentage of women who posted at least *N* number of times decreased to about 1% as *N* reached 13 posts.

On average males lurked slightly longer (390.44 hours) than females (233.12 hours) before posting to the mailing list for the first time. Also, males subscribed

about 270 hours longer to a mailing list than women (Males subscribed 6282.96 hours and females subscribed 6012.33 hours). However, we did not find any statistically significant values in these averages.

This study used data from very technical projects, which are all hosted in the same location. In the future it would be interesting to mix in some consumer or corporate-oriented projects hosted from a variety of locations. Also these projects had a similar number of users contributing, and incorporating some smaller and some larger projects may yield different results. A system to categorize posts from newbies might add insight about a user's intention to join a FOSS community or otherwise.

We intended to learn more about the joining process to determine if there is a significant point at which the number of women participating decreases. We found that fewer women subscribe to mailing lists than men, and that 67.90% of these women never post. However, the percentage of men who never post is lower, 59.30%. These findings support other research that finds FOSS user retention is very low, but it appears that the time women spend lurking on a list is not a significant point at which more women are turned away in comparison with men. Obviously, general measures to retain users who subscribe to a mailing list would increase general participation from men and women on FOSS projects. In order to encourage more women to join FOSS project mailing lists (and therefore take the first step to becoming a contributor) we must study other aspects of the projects, such as the projects' documentation or social areas.

7. Bibliography

"About : Fact Sheet." *NCWIT*. Web. 23 Aug. 2011. http://www.ncwit.org/about.factsheet.html>.

"About OSUOSL." *OSUOSL*. Web. 21 Aug. 2011. <http://osuosl.org/about-osuosl>.

"About Us." *The Humanitarian FOSS Project*. National Science Foundation. Web. 22 Aug. 2011. <http://www.hfoss.org/index.php/about-us>.

Andersen, Erik. *BusyBox*. Web. 21 Aug. 2011. <http://busybox.net/>.

Andersen, Erik. *UClibc*. Web. 21 Aug. 2011. <http://uclibc.org>.

Budd, Tim. "How Do I." *CS 419/519*. Oregon State University / College of Engineering, 27 Mar. 2009. Web. 22 Aug. 2011. <https://secure.engr.oregonstate.edu/classes/eecs/spring2009/cs419/index.php/Main/HowDoI>.

"Buildroot." *Buildroot*. Web. 21 Aug. 2011. <http://buildroot.uclibc.org/>.

Cedeno, Jose D. *Programmer Motivation Through Karma Systems*. Thesis. Oregon State University, 2010. Print.

Chalmers, Rachel. "Women Don't Have the Same Passion for Open Source Men Do? Really?? [LWN.net]." LWN.net, 27 Aug. 2009. Web. 21 Aug. 2011. <http://lwn.net/Articles/349234/>.

Chapman, Daniel. "Mailing Lists." *The Linux Foundation*. 12 May 2011. Web. 21 Aug. 2011. <http://www.linuxfoundation.org/content/26-mailing-lists>.

David, P., A. Waterman, and S. Arora. "The Free/Libre/Open Source Software Survey for 2003." *Stanford University*. 2003. Web. 21 Aug. 2011. <http://www.stanford.edu/group/floss-us/>.

Fisher, A., and J. Margolis. "Unlocking the Clubhouse: the Carnegie Mellon Experience." *ACM SIGCSE Bulletin* 34 (2002): 79-83. Print.

"Genealogy." *Census Bureau Home Page*. Web. 21 Aug. 2011. <http://www.census.gov/genealogy/names/names_files.html>.

Gutwin, R., C. Penner, and K. Chneider. "Group Awareness in Distributed Software Development." *Proc. of the 2004 ACM Conf. on Computer Supported*

Cooperative Work (2004): 72-88. Print.

Hemetsberger, A. "Learning and Knowledge-building in Open-source Communities: A Social-experiential Approach." *Management Learning* 37.2 (2006): 187-214. Print.

"HOWTO Encourage Women in Linux." *The Linux Documentation Project*. Ibiblio, 29 Oct. 2002. Web. 21 Aug. 2011. <http://tldp.org/HOWTO/Encourage-Women-Linux-HOWTO/>.

Jaws Project Jaws. 2008. Web. 21 Aug. 2011. <http://www.jaws-project.com/>.

Jensen, Carlos, Scott King, and Victor Kuechler. "Joining Free/Open Source Software Communities: An Analysis of Newbies' First Interactions on Project Mailing Lists." *Proc. of the 44th Hawaii Int'l Conf. on System Sciences* (2011): 1-10. Print.

Jensen, Chris, and Walt Scacchi. "Role Migration and Advancement Processes in OSSD Projects: A Comparative Case Study." *Proceedings of the 29th International Conference on Software Engineering* (2007): 364-74. Print.

King, Scott T. "Joining Open Source Software Communities: An Analysis of Newbies' First Interactions on Project Mailing Lists." Thesis. Oregon State University, 2010. Print.

Krieger, Bernhard, and James Leach. "Gender: Integrated Report of Findings." *Flosspols*. Mar. 2006. Web. 20 Aug. 2011.

Krogh, G., S. Spaeth, and K. Lakhani. "Community, Joining, and Specialization in Open Source Software Innovation: a Case Study." *Research Policy* 32.7 (2003): 1217-241. Print.

Levesque, Michelle, and Greg Wilson. "Women in Software Open Source, Cold Shoulder." Software Development, Nov. 2004. Web. 20 Aug. 2011.

Lin, Yuwei. "Gender Dimensions of Floss Development." *Mute Magazine | PRINT / ONLINE / SUBSCRIBE*. Mute, 23 Nov. 2005. Web. 21 Aug. 2011. <http://www.metamute.org/en/node/5596>.

"Linus Torvalds." *Wikipedia, the Free Encyclopedia*. 15 Aug. 2011. Web. 22 Aug. 2011. <http://en.wikipedia.org/wiki/Linus_Torvalds>.

"MBOX Documentation." *Qmail Mirror Selection*. 1998. Web. 21 Aug. 2011. <http://www.qmail.org/man/man5/mbox.html>.

McDonald, John H. "Handbook of Biological Statistics: Chi-squared Test of Independence." *University of Delaware*. 11 Sept. 2009. Web. 21 Aug. 2011. <http://udel.edu/~mcdonald/statchiind.html>.

Nafus, D., J. Leach, and B. Krieger. "Free/Libre and Open Source Software: Policy Support." (2006). Print.

Nonnee, Blair, and Jenny Preeee. "Lurker Demographics: Counting the Silent." *CHI '00 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2000): 73-80. Print.

"The Open Source Definition." *Open Source Initiative*. 1998. Web. 21 Aug. 2011. <http://www.opensource.org/docs/osd>.

"Package Management System." *Wikipedia, the Free Encyclopedia*. Web. 21 Aug. 2011. <http://en.wikipedia.org/wiki/Package_management_system>.

Raymond, Eric S. *The Cathedral & the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. Beijing: O'Reilly, 1999. Print.

Robles, Gregorio, Hendrik Scheider, Ingo Tretkowski, and Niels Weber. "Who Is Doing It? A Research on Libre Software Developers." BerliOS, 27 Aug. 2001. Web. 21 Aug. 2011. <http://widi.berlios.de/paper/study.html>.

Edgewall Software. "Yum Package Manager." *Yum*. Web. 21 Aug. 2011. <http://yum.baseurl.org/>.

"Users & Documentation." *Parrot VM*. Web. 21 Aug. 2011. <http://parrot.org/dev/docs/user>.

Women, Minorities, and Persons with Disabilities in Science and Engineering: 2007. Rep. no. NSF 07-315. National Science Foundation, Division of Science Resources Statistics,, Feb. 2007. Web. 21 Aug. 2011. <http://www.nsf.gov/statistics/wmpd/>.

Ye, Y., and K. Kishida. "Toward an Understanding of the Motivation Open Source Software Developers." *Proceedings of the 25th International Conference on Software Engineering, ICSE* (2003): 419-29. Print.