# Are FLOSS developers committing to CVS/SVN as much as they are talking in mailing lists? Challenges for Integrating *d*ata from Multiple Repositories *

Sulayman K. Sowe
Department of Informatics
Aristotle University
54124 Thessaloniki, Greece
sksowe@csd.auth.gr

Ioannis Samoladas
Department of Informatics
Aristotle University
54124 Thessaloniki, Greece
ioansam@csd.auth.gr

Ioannis Stamelos
Department of Informatics
Aristotle University
54124 Thessaloniki, Greece
stamelos@csd.auth.gr

Lefteris Angelis
Department of Informatics
Aristotle University
54124 Thessaloniki, Greece
lef@csd.auth.gr

## ABSTRACT

This paper puts forward a framework for investigating Free and Open Source Software (F/OSS) developers activities in both source code and mailing lists repositories. We used data dumps of fourteen projects from the FLOSSMetrics (FM) retrieval system. Our intentions are (i) to present a possible methodology, its advantages and disadvantages which can benefit future researchers using some aspects of the FM retrieval system's data dumps, and (ii) discuss our initial research results on the contributions developers make to both coding and lists activities.

## Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; D.2.8 [**Software Engineering**]: Metrics—*complexity measures, performance measures*

## General Terms

Open Source Software

## Keywords

Free/Open Source Software Development, Software Repositories, Concurrent Versions System, Mailing Lists

## 1. BACKGROUND

F/OSS developers are not bound to a single project. Even where they are contracted to work in corporate (eg JBoss), or foundation (Apache) projects, they still have the freedom to participate in other projects or communities of interest. They code, take part in discussions in various mailing lists and forums, and occasionally participate in agile-styled project's sprints. Along the way they leave a trail of experience, wealth of knowledge and skills associated with their art. This may be in the form of large and small bits of code, coding ethics and guidelines, documentations, etc. Participants in various F/OSS projects use tools (Versioning Systems (CVS/SVN), mailing lists, Bug tracking systems, etc.) to enable the distributed and collaborative software development process to proceed. These tools serve as repositories which can be data mined to understand *who* is involved, who is talking to *whom*, *what* is talked about, *how much* some one contributes in terms of code commits or email postings. Such information provides insights into the nature of collaboration in the projects concerned. Repositories of some F/OSS projects have extensively been used to better understand the contribution of developers [12, 6], trends and inequality in posting and replying activities in Apache and Mozilla [10], KDE [8], Debian developer and non-developer lists [15], FreeBSD [4], to mention a few. Empirical research in these areas is aided by the fact that F/OSS data is widely and freely available in various repositories in various formats [11, 7]. However, there are problems associated with aggregation and extraction of F/OSS data [2, 14]. A trend in F/OSS research is the use of data stored in a repository of repositories or *RoRs* (e.g. FLOSSMole and FLOSSMetrics)[1], in which data from many and varied projects are brought under one umbrella so that researchers can have easy access [14]. One major advantage of RoRs is that they offer an aggregated mixture of metadata in various formats, allowing researchers to concentrate more on there analysis than data scouting. This research benefits from such RoRs; the FM retrieval system (`http://fm3.libresoft.es/retrieval_system/`) of the FLOSSMetrics project.

*This research is partially sponsored by the FLOSSMetrics Project (Ref. No. FP6-IST5-033547), `http://flossmetrics.org/` and SQO-OSS project (Ref. No. FP6-IST-5-033331),`http://www.sqo-oss.eu/`

[1]http://ossmole.sourceforge.net/ and http://flossmetrics.org/

F/OSS projects are different in their organizational structures as well as their unique way of doing things. However, certain aspects are fundamental to all projects. Source configuration management (SCM), of which CVS or SVN is a part, is a *de facto* tool used to coordinate and view the coding activities of software developers, manage software builds and releases, and other software development related activities. Mailing lists, on the other hand, are the main communication channels [15]. Many important aspects of a project are negotiated in developer lists: software configuration details, the way forward and how to deal with future requests, how tasks are distributed, issues concerning package dependencies, scheduling online and off-line meetings, etc. For a developer to keep abreast with developments in a project, committing code to SVN alone is not sufficient. S/he needs to participate in the respective lists, communicate his ideas, and engage with colleagues. However, due to the volunteering nature of F/OSS development [9], developers are free to choose what to work on, and where to contribute and channel their efforts. Thus, a comprehensive investigation of developers must not only concentrate on their code contribution but also revisit their activities in other project's media (e.g. developer mailing lists) and compare and contrast their quantitative and qualitative contributions.

## 1.1 The purpose of our research

F/OSS researchers study and report developers coding activities in CVS [10, 8, 4] or change logs [1] separately from their mailing lists activities. Important as these studies and their findings are, we conjecture that not all the developers who commit or make changes to a project's source repository also participate in developer mailing lists. In order to *fill this gap in F/OSS research*, this study investigates the concurrence or simultaneous occurrence of F/OSS developers in both SVN and developer mailing lists. That is, we find out if F/OSS developers are coding through commits in SVN as much as they are "talking" in developer mailing lists.

Our understanding of the F/OSS development process informs us that in many projects, a small number of talented core developers or "cod gods" are busily tinkering with code to produce good and usable software for the rest of the community. We also know the contribution these code gods make to discussions in mailing lists; they interact with other software developers and users, the keep abreast with project activities and monitor the what goes on in there projects. Little or no research has attempted to correlate developers commits activities with their corresponding mailing lists activities, either within the same project or across projects. Active mailing lists is a proxy of project success [5, 3]. However, involvement in mailing lists should not only be limited to non-developers. The presence of project's leads, core and active developers in mailing lists has a profound effect on the way individuals within and outside the project see the commitment of the most influential members in the project. For software companies and private enterprises, their presence in lists may indicate that software support activities are not only available from ordinary users, but also comes form individuals behind the software and project. This argument leads to the formulation of the following hypothesis:

*H0 (null)*: FLOSS developers contribute equally to code repository and mailing lists, with alternative:

*H1*: FLOSS developers contribute more to code repository than mailing lists.

In what follows, we investigate the hypothesis and draw conclusions. Our data comes from SVN commits and mailing lists archives of *fourteen* F/OSS projects from the FM retrieval system. First, in Section 2, we present the methodology used in this research, followed by an analysis of our preliminary results in Section 3. Implications of our findings, limitations, and work in progress concludes this paper in Section 4. The observations we have made so far may provide some salient issues to be discussed and fine-tuned with workshop participants.

# 2. RESEARCH METHODOLOGY

Our research methodology aims to overcome challenges associated with investigating the simultaneous occurrence of developers in SVN and mailing lists. This has to do with difficulty in (*identification*) making sure that the developer making SVN commits in a project is the same individual posting to the developer mailing list(s) of the same project. An outline of the methodology (figure **1**) shows the FM retrieval system as our data set choice. In addition to mailing lists and CVS/SVN data dumps, the system provides many attributes of a project including project's description and website, SCM and mailing lists urls. One important use of the SCM urls is to allow researchers to locally checkout and browse projects' repositories and view and analyze change log data. Researchers can freely download and use available SVN and mailing lists data dumps. Bug databases of FM projects are currently being processed. For a detailed specification, design, and description of the FM database, refer to the FLOSSMetrics project's work-package 3 deliverable (3.1, 3.2)[2]. Figure **2** shows three tables from the FM database that we used to obtain SVN and mailing lists data.

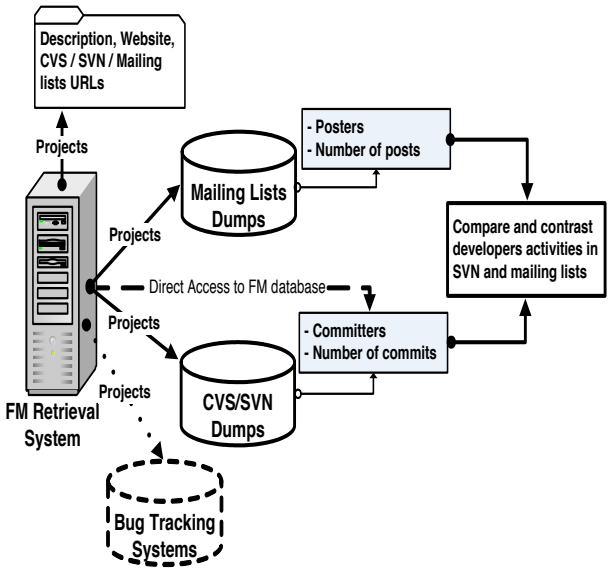



**Figure 1: Outline of research methodology**

[2] `http://flossmetrics.org/sections/deliverables/WP3`

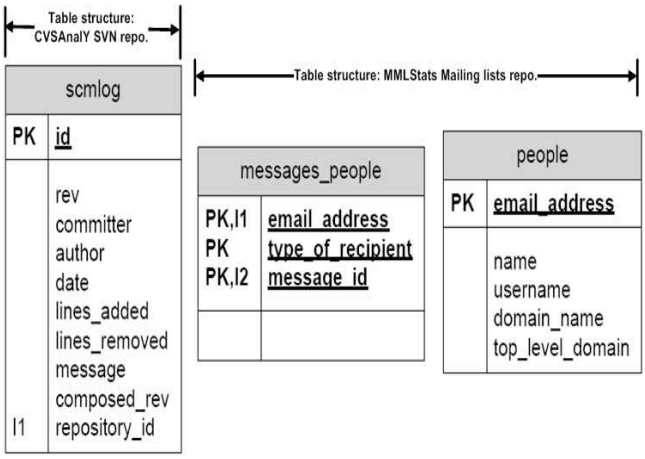



**Figure 2: FM's SVN and Mailing list tables schema**

As of 31st May, 2008, the FM beta version contained 60 projects with both analyzed data from source code management and from mailing lists repositories. From these we randomly selected 14 projects. Our choice of projects was guided by two selection criteria:
- projects should cover as many domains as possible, and
- developer contribution in terms of commits and postings should vary as much as possible.

With this criteria we hoped avoid selecting and studying only "successful" projects with large developer contribution and being bias towards one domain.

## 2.1 SVN Data

The CVSAnalY[3] database dumps provided by the FM retrieval system contains tables with SVN actions and information on committers. However, instead of downloading and using the SVN dumps, we used a python script to access the FM database, as a test and an alternative (see figure **1**). Each project in the FM database can have one or more SVN dumps. For each project, we extracted the SVN committers and the number of commits they made. These two values act as fields in an mysql table (*commits*) for committer identification in each project (see figure **3**). Table **1** shows descriptive statistics of the SVN data in the fourteen projects studied. For each project the total number of SVN

**Table 1: Projects by Commits before analysis**

| Project | Nĉ | Mean | Median | Std. Dev. | Skew. | Max. Commits | Total commits |
|---|---|---|---|---|---|---|---|
| activemq | 13 | 153.54 | 23.00 | 287.974 | 2.085 | 854 | 1996 |
| ant | 62 | 349.19 | 95.50 | 607.249 | 2.821 | 2908 | 21650 |
| apr | 132 | 191.56 | 55.00 | 318.549 | 2.230 | 1277 | 25286 |
| beehive | 13 | 330.54 | 258.00 | 331.582 | 1.475 | 1015 | 4297 |
| ekiga | 3 | 1010.33 | 459.00 | 1295.219 | 1.568 | 2490 | 3031 |
| felix | 12 | 58.92 | 30.00 | 85.896 | 1.758 | 260 | 707 |
| gdm | 24 | 92.67 | 20.50 | 203.638 | 3.265 | 894 | 2224 |
| gedit | 18 | 77.78 | 4.50 | 154.890 | 2.564 | 587 | 1400 |
| ibatis | 5 | 170.40 | 157.00 | 172.575 | 1.504 | 458 | 852 |
| libsoup | 4 | 170.25 | 171.00 | 190.241 | .000 | 335 | 681 |
| nautilus | 171 | 94.01 | 13.00 | 210.142 | 3.112 | 1202 | 16075 |
| openejb | 4 | 583.75 | 339.50 | 767.955 | 1.243 | 1647 | 2335 |
| Spamassassin | 14 | 1106.07 | 212.50 | 1278.926 | .741 | 3120 | 15485 |
| turbine | 24 | 107.92 | 77.50 | 128.307 | 1.786 | 540 | 2590 |

committers (Nĉ), the mean commit per committer, the total number of commits made by Nĉ developers, and other relevant statistics, are shown.

[3]http://cvsanaly.tigris.org/

## 2.2 Mailing Lists Data

The MLStats[4] database data dumps provided by the FM retrieval system contains one or more mailing lists archives of a particular project. Structurally, the data is a dump of the retrieval system's database ("*fm3_activemq_mls*"). We downloaded **.sql* files dumps of each project and extracted data contained in two tables:
- *messages_people* table (**email_address**, type_of_recipient ('**From**','To','Cc'),...), and
- *people* table (**email_address**, **name**, **username**,...).

This information acts as fields in two mysql tables for mailing lists posters identification in each project (see figure **3**). Table **2** shows, for each project, the total number of posters

**Table 2: Projects by Posts before analysis**

| Project | Np̂ | Mean | Median | Std. Dev. | Skew. | Max. Post | Total posts |
|---|---|---|---|---|---|---|---|
| activemq | 1548 | 8.03 | 2.00 | 65.898 | 34.813 | 2487 | 12430 |
| ant | 7301 | 8.67 | 3.00 | 52.725 | 32.967 | 2704 | 63312 |
| apr | 1060 | 17.47 | 2.50 | 70.120 | 10.842 | 1433 | 18514 |
| beehive | 272 | 6.57 | 2.00 | 16.970 | 5.874 | 152 | 1787 |
| ekiga | 525 | 7.93 | 2.00 | 47.647 | 19.774 | 1040 | 4163 |
| felix | 319 | 21.52 | 3.00 | 102.587 | 10.947 | 1283 | 6866 |
| gdm | 640 | 3.56 | 1.00 | 21.666 | 18.982 | 484 | 2281 |
| gedit | 471 | 4.11 | 1.00 | 15.543 | 12.085 | 266 | 1935 |
| ibatis | 1406 | 9.13 | 3.00 | 40.500 | 15.507 | 784 | 12830 |
| libsoup | 26 | 6.23 | 2.50 | 8.262 | 2.672 | 37 | 162 |
| nautilus | 2091 | 16.81 | 4.00 | 117.340 | 32.373 | 4719 | 35150 |
| openejb | 77 | 13.88 | 4.00 | 34.549 | 5.897 | 271 | 1069 |
| spamassassin | 4658 | 28.91 | 6.00 | 148.806 | 16.883 | 4248 | 134649 |
| turbine | 1874 | 21.01 | 6.00 | 70.111 | 10.700 | 1182 | 39377 |

(Np̂), the mean post per poster, the total number of posts made to the project's mailing list by Np̂ developers, and other relevant statistics, are shown.

## 2.3 Identifying Developers

In CVS or SVN and change logs, an individual is simply identified as a "Committer" or an "Author" of one or more commits. Mailing lists participants, on the other hand, can be identified by means of message identifiers like "From:" in email headers [13].

After extracting data of SVN committers and mailing lists posters, we proceeded with identifying developers as shown in figure **3**.

We queried the three tables to obtain data for our analysis. With this method, we were able have fairly accurate identity of individuals and their SVN and mailing lists contribution.

# 3. INITIAL RESULTS & ANALYSIS

Table **3** shows the number of developers (N) who made SVN commits and posted information to the project's mailing lists. Comparing tables **3** and **1**, in eight out of the fourteen (57.14%) projects studied, all the SVN committers also participated in mailing lists discussion. In four projects over 90% and in two projects over 80% of the committers participated in lists. As shown by the sum of commits and posts in tables **3**, developers in each project, with the exception of the *ibatis* and *turbine* projects, made more commits than posts. The mean commit per developer in each project, with

[4]`http://tools.libresoft.es/mailing_list_stats`

**Table 3: Statistics on Developers who made both commits and posts**

| Project | Commits | | | | | | Posts | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | N | Sum | Maximum | Mean | Median | Std. Deviation | N | Sum | Maximum | Mean | Median | Std. Deviation |
| activemq | 13 | 1996 | 854 | 153.54 | 23.00 | 287.974 | 13 | 807 | 380 | 62.08 | 5.00 | 109.439 |
| ant | 62 | 21650 | 2908 | 349.19 | 95.50 | 607.249 | 62 | 10026 | 2704 | 161.71 | 21.00 | 422.599 |
| apr | 129 | 24857 | 1277 | 192.69 | 56.00 | 320.824 | 129 | 11319 | 1433 | 87.74 | 22.00 | 173.654 |
| beehive | 12 | 3282 | 1015 | 273.50 | 239.50 | 271.666 | 12 | 202 | 104 | 16.83 | 6.50 | 28.565 |
| ekiga | 3 | 3031 | 2490 | 1010.33 | 459.00 | 1295.219 | 3 | 1228 | 1040 | 409.33 | 184.00 | 553.539 |
| felix | 12 | 707 | 260 | 58.92 | 30.00 | 85.896 | 12 | 457 | 223 | 38.08 | 15.50 | 61.745 |
| gdm | 20 | 2063 | 894 | 103.15 | 16.00 | 222.023 | 20 | 467 | 227 | 23.35 | 2.00 | 53.664 |
| gedit | 17 | 1399 | 587 | 82.29 | 5.00 | 158.431 | 17 | 334 | 266 | 19.65 | 2.00 | 63.659 |
| ibatis | 4 | 695 | 458 | 173.75 | 114.50 | 199.085 | 4 | 750 | 740 | 187.50 | 4.50 | 368.343 |
| libsoup | 4 | 681 | 335 | 170.25 | 171.00 | 190.241 | 4 | 68 | 37 | 17.00 | 13.50 | 15.684 |
| nautilus | 168 | 15968 | 1202 | 95.05 | 13.00 | 211.803 | 168 | 12891 | 4719 | 76.73 | 10.00 | 384.816 |
| openejb | 4 | 2335 | 1647 | 583.75 | 339.50 | 767.955 | 4 | 84 | 74 | 21.00 | 4.00 | 35.384 |
| spamassassin | 14 | 15485 | 3120 | 1106.07 | 212.50 | 1278.926 | 14 | 7235 | 2834 | 516.79 | 23.00 | 960.613 |
| turbine | 24 | 2590 | 540 | 107.92 | 77.50 | 128.307 | 24 | 15390 | 2364 | 641.25 | 224.00 | 825.384 |
| **Total** | 486 | 96739 | 3120 | 199.05 | 29.50 | 434.670 | 486 | 61258 | 4719 | 126.05 | 14.00 | 402.856 |

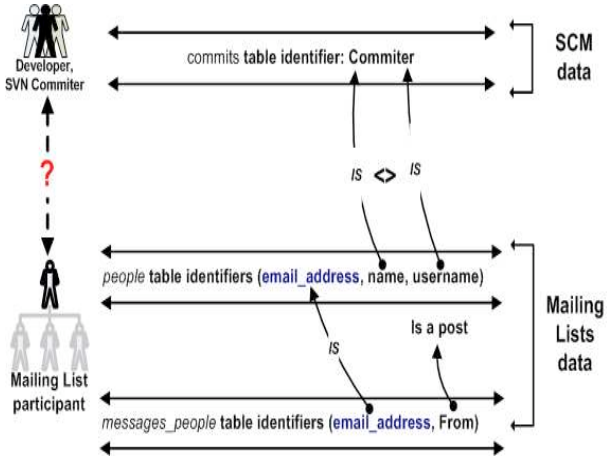



**Figure 3: Identifying developers from multiple repositories (CVS and Mailing Lists)**

the exception of the *turbine* project, are also greater than the mean post per developer.

## 3.1 Distribution of Commits and Posts

The box plots in figure **4** show the distributions of commits and posts for every project. The domination of commits over posts is evident in most of the projects. A comparison

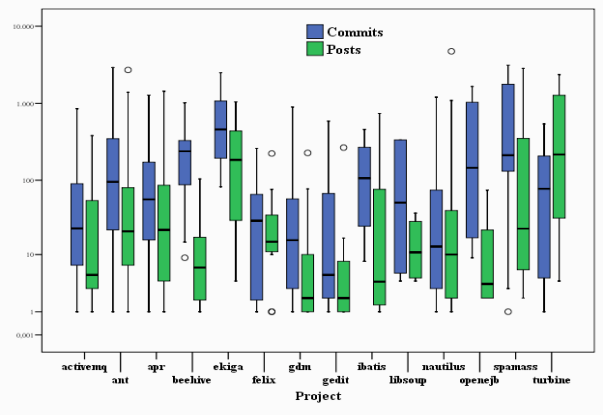



**Figure 4: Distributions of commits and posts for every project. *Y-axis in a logarithmic scale***

of the distributions of commits and posts for all developers together is shown in Figure **5**. The domination of SVN commits, with larger means of commit per developer, over mailing lists posts is evident.

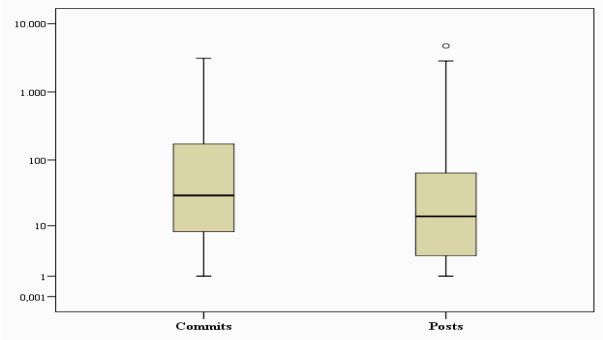



**Figure 5: Distributions of commits and posts for all developers. *Y-axis in a logarithmic scale***

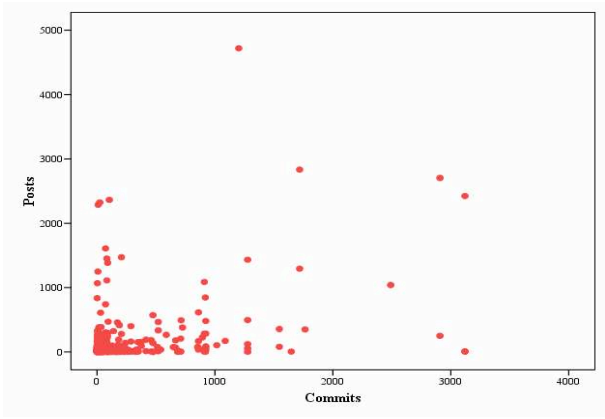
## 3.2 Relationship Between Commits and Posts

We used correlation between commits and posts to study how developers activities in SVN and mailing lists are related. The scatter plots in figure **6** shows the correlation between Commits and Posts in all projects, in two different scaling dimensions.

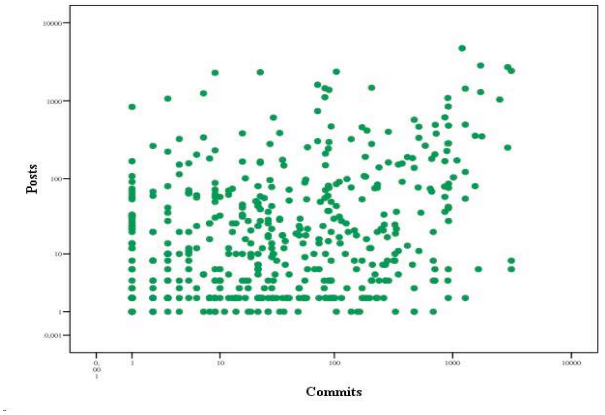
Since the distributions of both variables (commits and posts) are highly skewed and different from normal (kurtosis commits=20.150, std. dev.=434.670; kurtosis posts=47.720, std. dev=402.856), we use the nonparametric coefficient of Spearman which is based on ranks. For each project separately and for all projects together, table **4** gives the Spearman's coefficient ($\rho$) and their significance ($p$-value). Note that the significance is dependent on the size of the sample. Projects showing statistical significance ($p<0.05$) are are marked with an asterisk(*).

## 3.3 Developers Contribution

In order to measure how much developers contributed in terms of commits and posts, we tested the difference between commits and posts for each project and for all the fourteen projects together. We use the Wilcoxon signed ranks test for two related samples. For each project separately, and for all projects together, table **5** shows the significance ($p$-value) of the test. Statistically significant differences are




(a) for all projects in linear scale




(b) for all projects in logarithmic scale

**Figure 6: Correlations between commits and posts**

**Table 4: Commits-Posts Correlation measures**

| Project | N | Spearman's $\rho$ | Significance ($p$-value) |
|---|---|---|---|
| activemq | 13 | **0.612*** | **0.026** |
| ant | 62 | **0.294*** | **0.020** |
| apr | 129 | **0.283*** | **0.001** |
| beehive | 12 | **0.317** | **0.315** |
| ekiga | 3 | **1.000*** | **0.000** |
| felix | 12 | **0.173** | **0.591** |
| gdm | 20 | **0.350** | **0.130** |
| gedit | 17 | **0.528*** | **0.029** |
| ibatis | 4 | **0.000** | **1.000** |
| libsoup | 4 | **0.949** | **0.051** |
| nautilus | 168 | **0.123** | **0.113** |
| openejb | 4 | **0.738** | **0.262** |
| spamassassin | 14 | **0.171** | **0.559** |
| turbine | 24 | **-0.197** | **0.357** |
| All projects Together | 486 | **0.266*** | **0.000** |

considered those having $p<0.05$. In case of significant difference, from the mean ranks we can understand whether commits are generally more than posts. This is denoted by "*commits>posts*", otherwise we write "*commits<posts*". In projects where there is no statistical difference ($p>0.05$) we write "*commits=posts*".

**Table 5: Difference in developers' contributions**

| Project | N | Significance (p-value) | Type of difference |
|---|---|---|---|
| activemq | 13 | **0.124** | **commits=posts** |
| ant | 62 | **0.000** | **commits>posts** |
| apr | 129 | **0.000** | **commits>posts** |
| beehive | 12 | **0.002** | **commits>posts** |
| ekiga | 3 | **0.109** | **commits=posts** |
| felix | 12 | **0.424** | **commits=posts** |
| gdm | 20 | **0.006** | **commits>posts** |
| gedit | 17 | **0.007** | **commits>posts** |
| ibatis | 4 | **0.715** | **commits=posts** |
| libsoup | 4 | **0.109** | **commits=posts** |
| nautilus | 168 | **0.086** | **commits=posts** |
| openejb | 4 | **0.068** | **commits=posts** |
| spamassassin | 14 | **0.035** | **commits>posts** |
| turbine | 24 | **0.012** | **commits<posts** |
| All projects Together | 486 | **0.000** | **commits>posts** |

# 4. DISCUSSION AND CONCLUSIONS

In this paper we have investigated whether F/OSS developers are committing more to SVN than they are posting to mailing lists. This involves tracking their activities in two or multiple repositories and studying their quantitative contribution. This kind of research is made difficult because of the problem associated with the identification of developers in both repositories. The FM retrieval system's table schema has fields attributes which enabled us to overcome this research obstacle so that we are able to study the simultaneous occurrence of developers and measure their contributions in multiple repositories (SVN and mailing lists). The methodology presented in this paper is a possible means to leverage problems associated with empirical F/OSS research in this area.

The conclusion of our research supports our hypothesis.*H0 (null)*, FLOSS developers contribute equally to code repository and mailing lists. The null hypothesis is that the number of commits and the number of posts have the same distribution ($\rho$=0.000 for all the 486 committers who are also posters in the fourteen projects). Alternatively (*H1*), FLOSS developers contribute more to code repositories than mailing lists. The distributions of the two variables (commits and posts) are different for each project (with commits dominating) as shown by the various plots (see figures 4- 6). Furthermore, looking closer at the cases where the difference is not significant, we can see that the number of commits is greater. For example, the *nautilus* project has $p$=0.086 providing some evidence of difference at the 0.10 level and, domination of commits. The only exception was observed in two projects (*turbine* and *ibatis*), where the number of posts are greater than the number of commits. One explanation for this deviation could be that in both projects the prolific committer who is also the most prolific poster had exceptionally high commits (540 commits for the turbine project and 458 commits for the ibatis project) and posts (2364 posts for the turbine project and 740 posts for the ibatis project)relative to the rest of the community. But while the contribution of these prolific developers might skew the commits and posts in the projects, there could be other explanations. For example, both (ibatis, turbine) are Apache projects. Perhaps there is an underlying cause due to the structure or nature of Apache projects?

However, a number of factors or questions remains unanswered in this conclusion that we have not yet investigated. For example: **At commit level**: - What *kinds* of commits are the developers making? are they modifications, dele-

tions, additions? - What *types* of commits are the developers making? are the commits source code related, documents? - Time-stamp, when did a developer make those commits? **At mailing lists level**: - Is there a way we can we consider postings to developers lists only from the FM mailing lists dumps? - What category of information are the developers posting? - What about 'replies'? are there developers who are just posting project or software package information without replying to postings others made to the lists? Answers to some of these questions may help us to have a qualitative insight into the developers contribution to both repositories. And support observations we have made so far.

## 4.1 Few things we want to do

Since we have studied only those developers who are in both repositories, there is a tendency that we have missed some high prolific committers who may play a vital role in the project. We intend to look at other contributors with substantial commits and find out why they have no postings in their project's developer list(s). In our methodology, manual scan was necessary to remove some duplicates. The results of our SQL query sometimes yields results which require manual intervention. We are investigating how this methodology can be automated, perhaps by extending the functionality of CVSAnaY and MLStats data extraction and analysis tools.

## 4.2 Ongoing & future work

1. As the FM retrieval system database continues to grow, we are extracting data from a number of projects which have SVN and mailing lists dumps available to further investigate the trends we have observed.

2. Using information available at the FM retrieval system we can extend the analysis of developers participation in multiple repositories by including data from each project's change log. We have observed that the top "commits by author" in the change logs of each of the projects studied happens to be in top committers and posters in our data.

# 5. ACKNOWLEDGMENTS

We wish to extend sincere gratitude to all FLOSSMetrics and SQO_OSS partners who offered valuable comments and suggestions. In particular; Santiago Dueñas(FLOSSMetrics), Georgios Gousios(sqo-oss), etc.

# 6. REFERENCES

[1] K. Chen, S. R. Schach, L. Yu, J. Offutt, and G. Z. Heller. Open-source change logs. *Empirical Softw. Engg.*, 9(3):197–210, 2004.

[2] M. S. Conklin. Beyond low-hanging fruit: Seeking the next generation in floss data mining. In E. Damiani, B. Fitzerald, W. Scacchi, M. Scott, and G. Succi, editors, *IFIP International Federation for Information Processing, Open Source Systems.*, volume 23, pages 261–266. IFIP, Boston: Springer., 2006.

[3] K. Crowston, A. Hala, and J. Howison. Defining open source software project success. In *Proc. of International Conference on Information Systems, ICIS 2003*, 2003.

[4] T. T. Dinh-Trong and J. M. Bieman. The freebsd project: A replication case study of open source development. *IEEE Transactions on Software Engineering*, 31(6):481–494, 2005.

[5] K. Fogel. *Producing Open Source Software: How to Run a Successful Free Software Project.* CreativeCommons, 2005.

[6] G. Gousios, E. Kalliamvakou, and D. Spinellis. Measuring developer contribution from software repository data. In *MSR '08: Proceedings of the 2008 international workshop on Mining software repositories*, pages 129–132. ACM, 2008.

[7] A. E. Hassan. *Mining Software Repositories to Assist Developers and Support Managers.* PhD thesis, School of Computer Science, Faculty of Mathematics, University of Waterloo, Ontario, Canada., 2004.

[8] G. Kuk. Strategic interaction and knowledge sharing in the kde developer mailing list. *MANAGEMENT SCIENCE 2006 52: 1031-1042.*, 52:1031–1042, 2006.

[9] M. Michlmayr. *Quality Improvement in Volunteer Free and Open Source Software Projects: Exploring the Impact of Release Management.* PhD thesis, University of Cambridge, 2007.

[10] A. Mockus, R. Fielding, and J. Herbsleb. Two case studies of open source software development: Apache and mozilla. *Transactions on Software Engineering and Methodology.*, 11(3):1–38, 2002.

[11] G. Robles. *Empirical Software Engineering Research on Libre Software: Data Sources, Methodologies and Results.* PhD thesis, Dept. of Informatics. Universidad Rey Juan Carlos, Madrid, Spain., 2005.

[12] G. Robles and J. Gonzalez-Barahona. Contributor turnover in libre software projects. In E. Damiani, B. Fitzgerald, W. Scacchi, M. Scotto, and G. Succi, editors, *IFIP International Federation for Information Processing, Open Source Systems*, volume 203, pages 273–286. Springer,Boston, 2006.

[13] S. K. Sowe, L. Angelis, and I. Stamelos. Identifying knowledge brokers that yield software engineering knowledge in oss projects. *Information and Software Technology*, 48:1025–1033., 2006.

[14] S. K. Sowe, L. Angelis, I. Stamelos, and Y. Manolopoulos. Using repository of repositories (rors) to study the growth of f/oss projects: A meta-analysis research approach. In *Open Source Development, Adoption and Innovation*, volume 234/2007 of *IFIP International Federation for Information Processing*, pages 147–160. Springer Boston, August 2007.

[15] S. K. Sowe, I. Stamelos, and A. Lefteris. Understanding knowledge sharing activities in free/open source software projects: An empirical study. *Journal of Systems and Software*, 81(3):431–446., 2008.