

Using Repository of Repositories (RoRs) to Study the Growth of F/OSS Projects: A Meta-Analysis Research Approach

Sulayman K. Sowe, L. Angelis, I. Stamelos, and Y. Manolopoulos

Department of Informatics, Aristotle University, 54124 Thessaloniki, Greece.

Tel: +30-2310-991927 Fax: +30-2310-998419

* `sksowe,lef,stamelos,manolopo@csd.auth.gr`

Abstract. Free/Open Source Software (F/OSS) repositories contain valuable data and their usefulness in studying software development and community activities continues to attract a lot of research attention. A trend in F/OSS studies is the use of metadata stored in a repository of repositories or RoRs. This paper utilizes data obtained from such RoRs -FLOSSmole- to study the types of projects being developed by the F/OSS community. We downloaded projects by topics data in five areas (Database, Internet, Software Development, Communications, and Games/Entertainment) from Flossmole's raw and summary data of the sourceforge repository. Time series analysis show the numbers of projects in the five topics are growing linearly. Further analysis supports our hypothesis that F/OSS development is moving "up the stack" from developer tools and infrastructure support to end-user applications such as Databases. The findings have implications for the interpretation of the F/OSS landscape, the utilization and adoption of open source databases, and problems researchers might face in obtaining and using data from RoRs.

Key words: Open Source Software Repositories, Metadata, Time Series Analysis, Missing Values Estimation, Projects Growth, Linear Trend, Open Source Databases.

1 Introduction

The user of Free and Open Source Software (F/OSS), having access to the source code, is free to study what the program does, modify it to suit his/her needs, distribute copies to other people and publish improved versions so that the whole F/OSS community can benefit. The licenses agreement (e.g. the General Public License or GPL) under which the source code is distributed defines exactly the rights the user has over the product. The Bazaar model [19] of developing F/OSS represents a significant shift in the way we develop and maintain traditional or closed-source software (CSS). As [27] pointed out, F/OSS development

* Correspondence author: Sulayman K Sowe; Email: sksowe@csd.auth.gr

differs in many ways from that of CSS, where it is common to assume centralized software development and administrative authority that controls and manages the resources and schedules for software development and maintenance. The model has produced a number of successful applications in the area of operating systems (Linux), emailing and web services (Gmail, Apache), databases (MySQL, PostgreSQL), to mention a few. Participants in F/OSS project rely on extensive peer collaboration through the Internet, using project's mailing lists, *de facto* versioning systems such as Concurrent Versions System (CVS) or Subversion (SVN), bug-tracking systems (BTS) and bug databases (e.g. Bugzilla), Internet Relay Chats (IRC), discussion forums, etc. These tools not only enable participants to collaborate in the software development process but also act as repositories to store the communication activities of the participants.

With the coming of F/OSS, sprang various portals to provide hosting services for projects of all kinds and flavours. Among the largest and most popular is sourceforge. Freshmeat and Savannah also continue to attract a lot of attention. These portals are hosts to small and large, successful [7] and unsuccessful projects. Yet, many portals are also graveyards strewn with abandoned projects. The plethora of projects or applications available throughout the Internet is an indication of a growing interest in F/OSS and the fact that an increasing number of skilled programmers are willing to transform their [tacit] knowledge and skills into tangible products [24]. While corporations posting projects to sourceforge view their strategies as important in accelerating adoption and migration of their products and services [3], the F/OSS landscape is biased towards certain kinds of projects and software. What is more interesting about the landscape is that applications developed by the community are not uniformly distributed across all domains. Projects cover wide ranging topics, with development being dominated by infrastructure support or Internet based projects [22]. These products may fall under operating systems (Linux, FreeBSD), server and Internet applications (Apache, Sendmail, BIND), and software development tools (GCC, Perl, Python). Many of these projects resulted from an individual (usually a skilled programmer) scratching his own itch [19]. Ideally, if the required software is not freely available, one can either develop it on his own or contribute the initial code-base and release it to the F/OSS community for collaboration. Bezroukov, [4] noted that F/OSS projects are more successful in areas that are directly or indirectly interesting to developers themselves. However, itch-based F/OSS may not succeed in improving ease of use for those users, such as novices, whose background differs from professional developers [16]. Even though there are a growing number of applications targeting end-users (the KDE and GNOME desktops, Firefox, Thunderbird), our aim in this study is to discuss trends in F/OSS research and provide a quantitative analysis of the F/OSS landscape to test one hypothesis:

Hypothesis: *Is there evidence to support that F/OSS development is moving from focusing only on developer tools and infrastructure support to end-user applications such as Databases?*

Many researchers (e.g. [5, 28]) obtain data directly from repositories such as sourceforge. Despite having firsthand access to the data source, harvesting or crawling sourceforge could be a daunting task [12]. Alternatively, researchers may utilize subsidiary meta-data provided in a *repository of repositories* or **RoRs** such as FLOSSmole [11, 12]. FLOSSmole [18] may be described as RoRs or meta-repository of projects hosted at Freshmeat, sourceforge, <http://rubyforge.org/>, and <http://www.objectweb.org/>. From the FLOSSmole repository, we extracted *projects by topic* in five different topics- Database, Internet, Software Development, Communications, and Games/Entertainment- and developed time series analysis to study how projects in these topics grew from January 1st, 2005 to August 31st, 2006. We then compared the growth of open source database projects with projects in the other four topics. The rest of the paper is organized as follows. Section 2 presents trends in the use of RoRs in F/OSS research. In section 3 we present our research methodology and discuss our data collection and extraction. In Section 4 we present the results, discuss our findings, and list the validity threats to our research. Our conclusion and future work is presented in section 5.

2 Trends in F/OSS Research

Compared to traditional research practices under proprietary software, F/OSS development provides researchers with an unprecedented abundance of easily accessible data for research and analysis. A huge amount of data is available to study community participation in F/OSS projects [2, 10, 1, 9, 13] and developer and user involvement in projects mailing list [21, 25, 14, 15]. Web sites which host F/OSS projects also provide each project with repositories or tools to enable the collaborative software development process to proceed. The largest site which has generated a lot of research interest is sourceforge. Data from this site has been used to study many aspects of F/OSS. For instance, the geographical location of developers [20], topological analysis of developer communities [28], Knowledge collaboration across projects [17], patterns of software development [26, 8], percentage distribution of projects [22], etc. The traditional way of obtaining data for most of these kinds of research is spidering or crawling of sourceforge using Perl or Python scripts. However, instead of direct access to the sourceforge repository, researchers may also benefit from reusing data other researchers obtained from sourceforge. The University of Notre Dame maintains a data dump from sourceforge [5, 8] and other researchers may request and reuse the data in their studies [20]. In another study, [29] reused data from [12] to study the self-organizing patterns in wasp and F/OSS communities.

It is becoming increasingly evident that collecting and analyzing F/OSS data has become a problem of abundance and reliability in terms of storage, sharing, aggregation, and filtering [6]. Some of the problems researchers may face in obtaining and using data in their research can be summarized thus:

- **Convergence of data:** There is no standardized way of defining or a naming convention for variables in a repository. This may pose problems for researchers when it comes to harmonizing data across different repositories.
- **Without Notice!** The data structure of a repository is held in a back-tier (database). And because many researchers just interact with the front-end of the repository, researchers can face a daunting task when a site or a repository maintainer changes the structure of the data or schema. [12].
- **Confidentiality:** Due to the sensitive nature of some aspects of the data (e.g. private emails), some projects might be reluctant to release some of their data at a time when the researcher actually needs it.
- **A friend of a friend (FOAF):** A researcher not having direct access to the data he needs for his research may send a request to the project, either through mailing lists or to the repository maintainer. Experience shows that sometimes processing such a request is like waiting for rain in the desert. In this case, knowing someone who has obtained the data the research wants or knowing some members of the core team helps.

These difficulties significantly impede F/OSS research. As a result, many researchers see the need for the establishment and use of **Repositories of Repositories** or RoRs. The whole concept of RoRs is an attempt to pull data from many and varied repositories and bring the data under one umbrella so that researchers can have easy access to data, reports, tools, and scripts used in F/OSS research. As [12] noted in their schematic analysis, current F/OSS research is "one-way traffic"; non-cyclical and non-collaborative. Once researchers obtain, analyze and publish their data, the product of their research is never put back to the community from which they obtained their data. Existing structures of software repositories do little to ameliorate this situation. The aim of RoRs should be to close this loop by encouraging researchers to contribute their data and any scripts and tools they used in their research to the RoRs from which they obtained the original data. Our view of how the RoRs concept should work is illustrated in Figure 1. From the diagram, note that there is a continuous feedback between the research community and the RoRs.

The first kind of RoRs available to researchers is FLOSSmole. For a detailed description of the purpose, design, and requirements of Flossmole, see [12]. Another RoRs in progress is the EU funded FLOSSMetrics (<http://www.flossmetrics.org/>) project. The FLOSSMetrics or Free/Libre Open Source Software Metrics project aims to construct, publish and analyze a large scale database with information and metrics about F/OSS development. Using existing methodologies and tools already developed the project will house data coming from several thousands of software projects. The project will also provide a public platform for validation and industrial exploitation of results. Some of the targets of the project are summarized:

- Identify, evaluate sources of data, and develop a database structure.
- Build and maintain an updated empirical database.
- Disseminate the results, including data, methods and software.

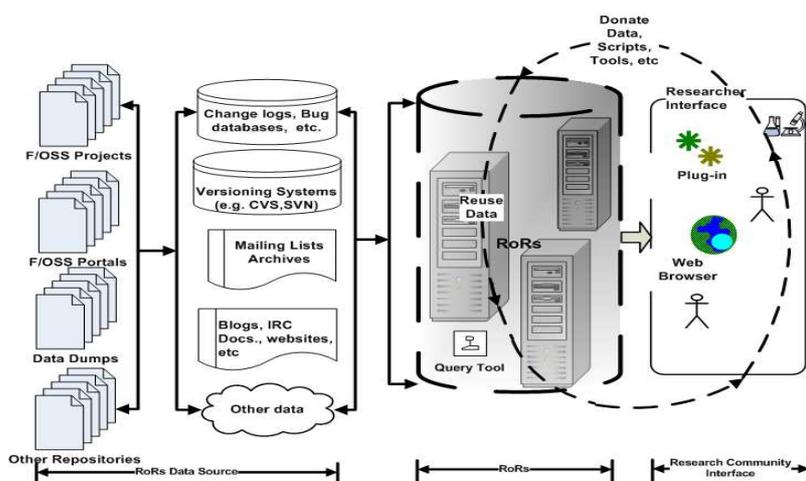


Fig. 1. Conceptual Framework of RoRs

The project will work with other projects such as FLOSSmole, the Software Quality Observatory for Open Source Software or SQO-OSS (<http://www.sqo-oss.eu/>) and QUALity of Open Source Software or QUALOSS (<http://www.qualoss.org/>). Thus, it is becoming increasingly feasible to use data from RoRs for quality F/OSS research.

3 Methodology

The research methodology employed in this paper is schematically shown in Figure 2. Where a similar methodology may be applicable is shown in dotted cones. The "Donate" arrows show a researcher contributing his Python script and the results of his analysis to the RoRs - FLOSSmole.

Data: The FLOSSmole repository has data dumps collected from repositories in text and excel files format. For our study, we downloaded raw and summary data of FLOSSmole's archives of sourceforge data during twenty months period, from January 1st, 2005 to August 31st, 2006. We chose five major projects by topic; Database, Internet, Software Development, Communications, and Games/Entertainment. We selected these five topics because, first, the last four show a dominant position in the sourceforge's "software Map" (Table 1) and will be quite representative of the types of projects being developed by the F/OSS community. Second, all the five topics or categories have been used in a previous study [22], and it will be interesting to compare and contrast our results with that study. Third, to validate our hypothesis whether there is evidence to support that F/OSS development is moving "up the stack" from developer tools and infrastructure support to end-user applications such as Databases, we will

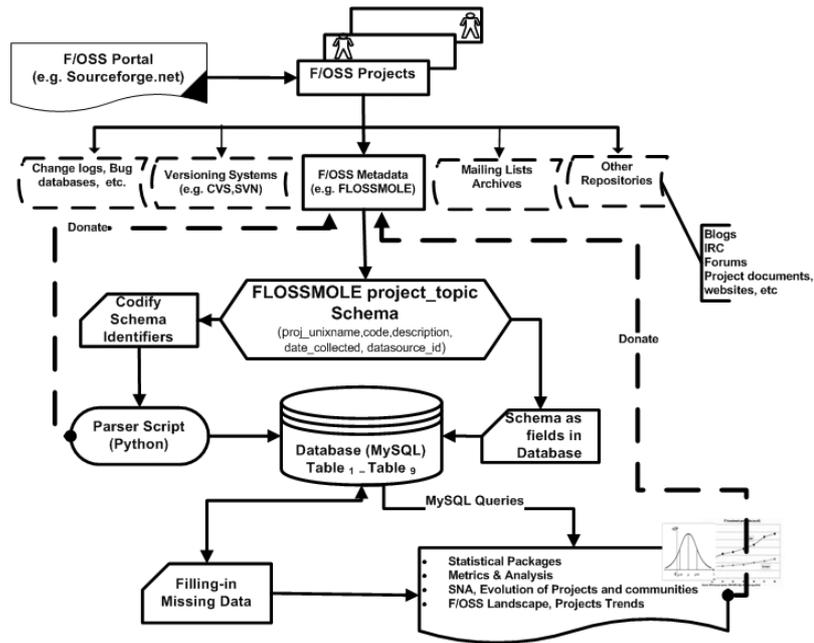


Fig. 2. Methodological outline to extract data from FLOSSmole.

investigate how projects in the database topic scale against other projects in the other four topics.

Table 1. Number of projects in descending order in 9 out of 19 topics. Extracted from sourceforge’s ”software map” on 17/01/2007. Asterisks beside topics in our study.

Topic	Total Projects	Example
Internet*	26505	FileZilla
Software Development*	25840	Gaim
System	21524	phpMyAdmin
Communications*	17115	Gaim
Games/Entertainment*	15894	FreeCol
Multimedia	14426	MediaPortal
Scientific/Engineering	13542	K-3D
Office/Business	8802	Openbravo ERP
Database*	6509	phpMyAdmin
.....

The FLOSSmole files we used include raw ”Project Topic” data for each project. However, not every project in sourceforge lists this information [6]. We codified the ”project_topic” schema obtained from the text files into a Python script and implemented it as fields in our MySQL database containing nine tables,

one for each available data. A text file of each month's data was parsed into the database for subsequent analysis. The database was queried for projects belonging to a given topic and the dates they were hosted at sourceforge.

Extracted data and Missing Values Estimation: Our original raw data had unequal time gaps. We had gaps of two months for the first year (2005) and gaps of one month for the second year (2006). For example, Figure 3 shows the dates and total numbers of projects extracted from the Database and Software Development topics.

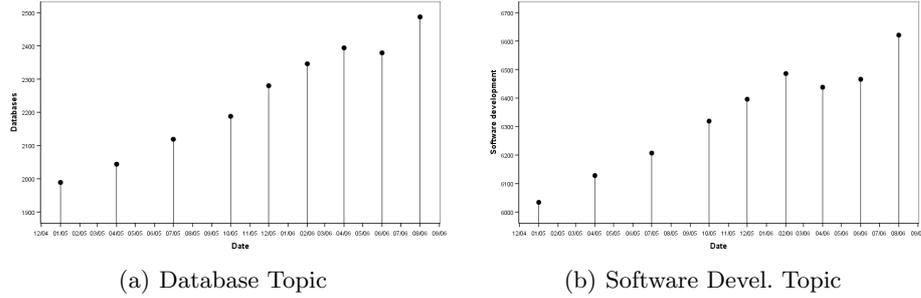


Fig. 3. Scatter plots showing gaps in the original data.

We exploited the linear trend in our data to fill the gaps with estimations. The "linear trend at point" method [23] was used to replace the missing data with the linear trend for that point. First, a linear regression line is fitted to the existing observations with respect to an index variable scaled from 1 to n (total number of observations) and then the missing values are replaced with their predicted values. The new data shown in Tables 2 have real or estimated values, in parentheses, for every month.

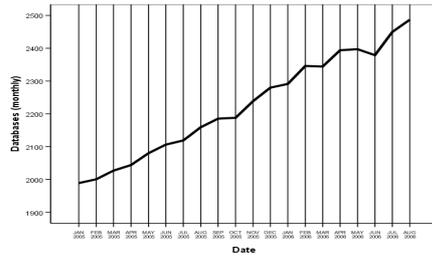
4 Results and Discussions

4.1 Overview of F/OSS Projects' Landscape

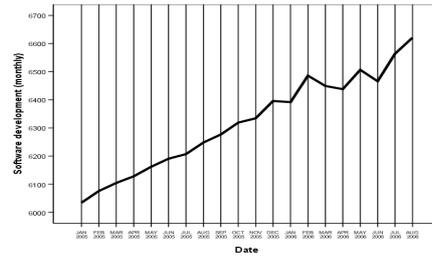
The representation of our data as time series with observations taken (or estimated) in equally spaced time intervals, gives us the opportunity to use time series analysis in order to construct predictive models or to study the behavior of our data in time. Time series plots for the Database and Software Development topics in figure 4 show that the number of projects in the F/OSS landscape is not at all stationary. Instead, projects in all the five topics exhibit a linear growth. Thus, we can use this behaviour to model and forecast the growth of the projects in all the topics.

Table 2. Number of F/OSS Projects by Topics.

Month	Database	Internet	Software Devel.	Communications	Games/Entert.
Jan. 05	1989	2465	6034	1990	2435
Feb. 05	(2001)	(2500)	(6076)	(2019)	(2444)
Mar. 05	(2027)	(2530)	(6104)	(2045)	(2479)
Apr. 05	2044	2546	6128	2064	2505
May 05	(2080)	(2590)	(6162)	(2097)	(2548)
Jun. 05	(2106)	(2621)	(6191)	(2123)	(2583)
Jul. 05	2119	2643	6207	2139	2602
Aug. 05	(2159)	(2681)	(6248)	(2176)	(2652)
Sep. 05	(2186)	(2711)	(6277)	(2202)	(2687)
Oct. 05	2188	2738	6319	2211	2700
Nov. 05	(2239)	(2772)	(6334)	(2254)	(2756)
Dec. 05	2280	2817	6396	2290	2778
Jan. 06	(2291)	(2832)	(6392)	(2306)	(2826)
Feb. 06	2346	2928	6486	2398	2887
Mar. 06	(2344)	(2893)	(6449)	(2359)	(2895)
Apr. 06	2394	2942	6438	2412	2961
May 06	(2397)	(2953)	(6507)	(2411)	(2965)
Jun. 06	2379	2916	6466	2377	2939
Jul. 06	(2450)	(3014)	(6564)	(2463)	(3034)
Aug. 06	2487	3043	6621	2483	3104



(a) Database Topic



(b) Software Devel. Topic

Fig. 4. Linear growth of Projects by Topic.

Forecasting Growth: We applied exponential smoothers to model the behavior of our data and provide forecasts for the next three months, until the end of the year, 2006.future months. The smoothing curves in Figure 5 are almost straight lines due to the strong linear trend.

4.2 Are Database Projects Popular with the F/OSS Community?

End-user applications such as databases are not a panacea or unknown quantities amongst the F/OSS community. Software developers, vendors, and database

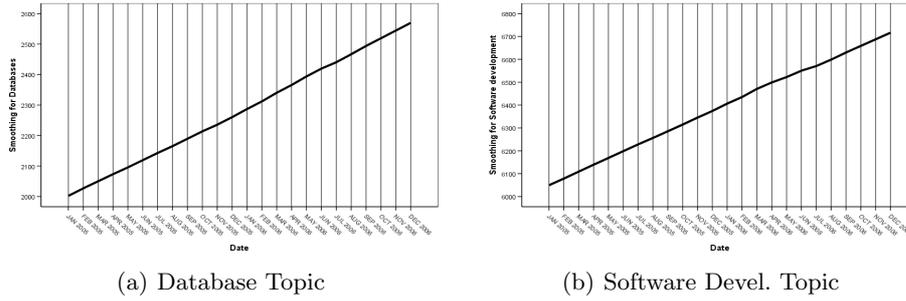


Fig. 5. Sequence plots showing smoothing curves with forecasts until the end of 2006.

users are already familiar with major players. MySQL, PostgreSQL, Berkeley DB, Firebird, and many others, continue to attract attention and their user base is growing. Projects in the database topic are doing well in the F/OSS landscape. Figure 6 shows that there is a significant linear correlation in the growth of Database projects and Internet ($R^2=0.996$) and Software Development projects ($R^2=0.984$). The growth trend is similar in the other topics.

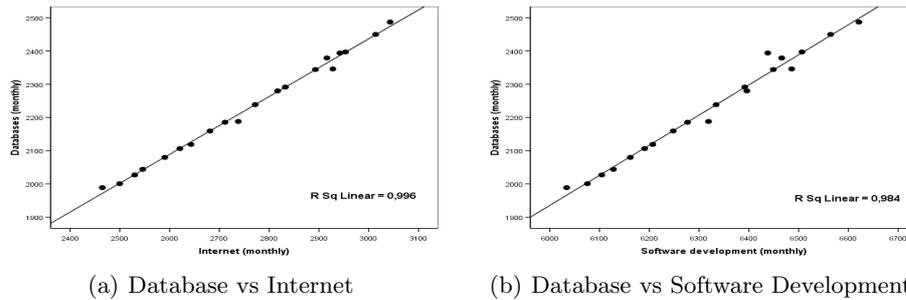


Fig. 6. Trend in the growth of Database projects versus Internet and Software Development projects.

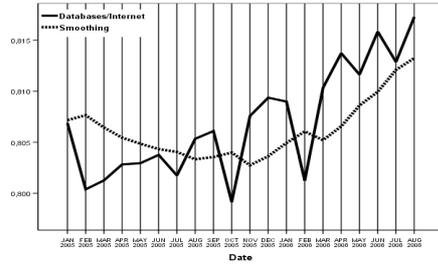
The Pearson correlation coefficients given in Table 3 shows that the linear correlation among all topics is highly significant as all are very close to 1. The corresponding hypothesis tests show that these correlations are significant at the 0.01 level.

In order to compare projects in the Database topic with projects in other topics, it is interesting to investigate the ratios obtained by dividing the number of database projects at each time period by the corresponding numbers of projects in other topics. Some of the sequence plots obtained by plotting these ratios are shown in Figure 7.

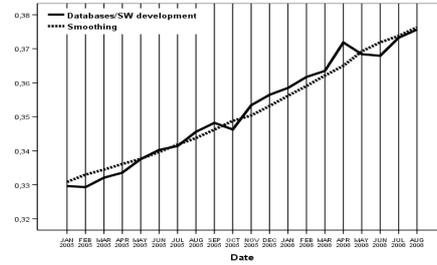
For a better understanding of the comparison, we added in each plot a smoothing curve showing more clearly the overall behavior. Plot (a) shows

Table 3. Correlation between projects in topics.

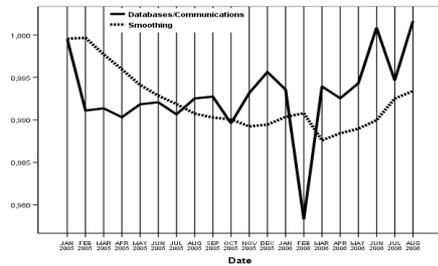
	Internet	Software devel.	Communications	Games_Entert.
Databases	.998	.992	.997	.999
Internet		.995	.999	.996
Software devel.			.993	.990
Communications				.995



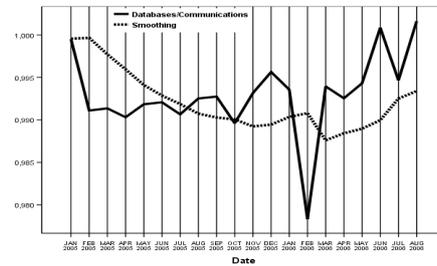
(a)



(b)



(c)



(d)

Fig. 7. Sequence plots showing ratios of database projects by each of the other topics.

that the number of database projects is approximately 81% of the internet projects but there seems to be a growing trend for this ratio from November 2005. Plot (b) shows that the database projects are around 35% of the software development projects but this ratio has a clear linear growth. Plot (c) shows that the number of database projects is almost the same as that of communications, since their ratio is very close to one. There is initially a descending trend but later the trend is ascending. Finally, plot (d) shows that database projects vary by, approximately, 81% of Games/Entertainment projects, but this ratio has an almost linear descending trend.

4.3 Hypothesis Validation

Hypothesis: *Is there evidence to support that F/OSS development is moving from focusing only on developer tools and infrastructure support to end-user applications such as Databases?*

It follows from the discussion presented in section 4.2 that F/OSS development is moving "up the stack" from developer tools and infrastructure support to end-user applications such as open source databases. The analysis shows a steady growth of not only end-user projects such as database but also growth in major areas such as Internet, Software Development, Communications, and Games and Entertainments. Thus, our hypothesis is supported within the limit of our analysis.

4.4 Validity Threats and Considerations

Any claim to map the ecology of the types of projects in the F/OSS landscape should be treated with caution for the reason that:

- *Ex-ante analysis* of our data shows that some projects belong to more than one topic (see Table 1). For example, phpMyAdmin is classified under the Database as well as Systems topic. Each topic also contains 'sub-topics', for example the Database topic also contained 'database Engines/Servers' and 'Front-Ends' as sub-topics. Thus, there is the inevitable consequence of counting some projects more than once, thus inflating the numbers.
- Even though sourceforge is the largest repository of F/OSS projects, there are other repositories (e.g. Freshmeat, Savannah.gnu) which are equally important.
- Not all F/OSS projects are hosted at sourceforge. In fact most of the 'successful' F/OSS projects are hosted outside sourceforge. Others only maintain a link with the portal.
- The quality of projects in sourceforge vary tremendously. A more plausible option would have been to define criteria for the types of projects to study. For instance projects by topic for projects with a certain number of programmers, downloads, sourceforge rating, etc)
- Small dataset. We based our discussion on a dataset obtained during 20 months. Perhaps 2-5 years data would have revealed a different and clearer trend than the one we reported.

The research methodology we employed in this paper may serve as an impetus for researchers faced with the inevitable consequence of missing data. We have also highlighted the importance and benefits of using RoRs in F/OSS empirical studies. However, important questions about RoRs need addressing:

Infrastructural/Technical: What are the requirements for implementing the basic infrastructure required to setup and link the repositories? What are the major problem associated with integrating the schemata of individual and/or

heterogeneous databases [30]? What are the required communication protocols (OAI)? How to deal with the issue of missing data? What are the lessons learnt from the technical challenges from the FLOSSMole?

Data Quality: How will obtaining data from many and different repositories (employing different schemas) affect the quality of the data? How to deal with issues of missing data?

Motivational/Social: Are researchers prepared to 'give back' their fine-tuned data, scripts and research tools to RoRs? How to create a partnership between RoRs and their parent repositories so that RoRs maintainers will be well informed should the structure of the parent repository change.

Economical: Many F/OSS projects are voluntary in nature and depend on benevolent donations from individuals to function. The establishment and maintenance of RoRs need financial funding. How will RoRs be funded? One initiative in this regard is the EU funded FLOSSMetrics project.

5 Conclusion

In this paper metadata from FLOSSmole, which is a repository of repositories (RoRs), was used to discuss the F/OSS landscape in terms of the projects being developed by the F/OSS community. We encountered gaps in our data and used the 'linear trend at point' method to fill the gaps with estimations. Various statistical methods were employed to investigate the F/OSS projects' landscape and we found out that projects in all the topics studied are growing linearly. Exponential smoothing curves produced almost straight lines due to the strong linear trend. Three months forecast showed that the number of projects in these topics continued to grow beyond our study. Comparing the trend in the growth of the number of projects in the database topic against the other four, revealed a high correlation between databases and all the other four projects' topics. These findings show that applications developed by the F/OSS community are not limited to infrastructural or Internet based components only, but also to end-user products such as Databases. The ration of database to Internet, software development, and Games/Entertainment projects showed a linear growth. Furthermore, projects in the database topic are growing almost at the same rate as those in the communications topic.

Future Work: We are spidering sourceforge to obtain data spanning many years so that we can build better prediction models to study the F/OSS landscape. We are also collecting data on researchers who are using the same dataset. We intend to develop social networks where researchers form nodes on a network/graph and two or more nodes are linked if they share the same dataset. Collaborative networks of this nature can reveal a great deal about the social structure of the F/OSS research community, such as the presence knowledge brokers [25].

Acknowledgement. This work is partially supported by the Free/Libre/Open Source Software Metrics and Benchmarking Study (FLOSSMetrics), Project No: 033982, (<http://flossmetrics.org>). Sincere thanks to the maintainers and participants of Flossmole. Script, data, and analysis used in this paper can be found at: <http://ossmole.sourceforge.net/>

References

1. Hahsler M, Koch S., Discussion of a Large-Scale Open Source Data Collection Methodology, Proceedings of the 38th Hawaii International Conference on System Sciences (IEEE, HICSS '05-Track 7), Jan 03-06, Big Island, Hawaii, 2005, page 197b.
2. Barahona, J. et al., Analysing the Anatomy of GNU/Linux Distributions: Methodology and Case Studies (Red Hat and Debian), In Koch, S. (Ed.), Free/Open Source Software Development, Idea Group Inc., 2005, pp: 27-58.
3. govtech.net, SourceForge.net Surpasses 100,000 Open Source Projects, <http://www.govtech.net/news/news.php?id=94043>, May, 2005.
4. Nikolai Bezroukov, Open Source Software Development as a Special Type of Academic Research (Critique of Vulgar Raymondism), Firstmonday, 1999, vol. 4(10).
5. Scott Christley and Greg Madey, Collection of Activity Data for SourceForge Projects, Technical Report: TR-2005-15, University of Notre Dame, 2005.
6. Megan S Conklin, Beyond Low-Hanging Fruit: Seeking the Next Generation in FLOSS Data Mining, In Damiani, E., Fitzgerald, B., Scacchi, W., Scott, M., Succi, G (Eds.), IFIP International Federation for Information Processing, Open Source Systems, Boston: Springer, Vol. 203, 2006, pp: 261-266.
7. Kevin Crowston, Hala Annabi, and James Howison, Defining Open Source Software Project Success, Proc. of International Conference on Information Systems, ICIS 2003, 2003.
8. G. Madey, V. Freeh, and R. Tynan, The open source software development phenomenon: An analysis based on social network theory, In Americas conf. on Information Systems (AMCIS2002), 2002, pp: 18061813.
9. German, D. and Mockus, A., Automating the Measurement of Open Source Projects, ICSE '03 Workshop on Open Source Software Engineering, Portland, Oregon, May 3-10, 2003.
10. Ghosh, A.R., Clustering and dependencies in free/open source software development: Methodology and tools, Firstmonday, Vol. 8(4), 2004.
11. Howison, J., Conklin, M., and Crowston, K., FLOSSmole: A collaborative repository for FLOSS research data and analyses, International Journal of Information Technology and Web Engineering, Vol. 1(3), 2006, pp: 17-26.
12. J. Howison, and K. Crowston, The Perils and pitfalls of mining SourceForge, 26th International Conference on Software Engineering, Edinburgh, Scotland, 2004.
13. Koch S, Schneider G., Effort, cooperation and coordination in an open source software project: Gnome., Information Systems Journal, Vol. 12(1), 2002, pp: 27-42.
14. Krogh G, Spaeth S, and Lakhani, K., Community, joining, and specialisation in open source software innovation: a case study, Research Policy, Vol. 32, 2003, pp: 1217-1241.

15. Lakhani K, Hippel von E., How open source software works: "free" user-to-user assistance, *Research Policy*, Vol. 32, 2003, pp: 923-943.
16. David M. Nichols and Michael B. Twidale, Usability processes in open source projects, *Software Process: Improvement and Practice*, Vol. 11(2), 2006, pp: 149-162.
17. Masao Ohira, Naoki Ohsugi, Tetsuya Ohoka, and Ken-ichi Matsumoto, Accelerating cross-project knowledge collaboration using collaborative filtering and social networks, *MSR '05: Proceedings of the 2005 international workshop on Mining software repositories*, ACM Press, 2005, pp: 1-5.
18. FLOSSmole Project, FLOSSmole Project (2004-2006) Sourceforge, <http://ossmole.sourceforge.net>.
19. Raymond, E.S., *The Cathedral and the Bazaar. Musings on Linux and Open Source by an Accidental Revolutionary*, OReilly, Sebastopol, USA., 1999.
20. Gregorio Robles and Jesus M. Gonzalez-Barahona, Geographic location of developers at SourceForge, In *MSR '06: Proceedings of the 2006 international workshop on Mining software repositories*, ACM Press, 2006, pp: 144-150.
21. Sulayman K. Sowe and Ioannis. Stamelos, Identification of Knowledge Brokers in F/OSS Projects through Social and Collaborative Networks, In *Proc. of 10th Panhellenic Conference on Informatics*, Volos, Greece, 2005, pp.285-303.
22. Sulayman K. Sowe, Ioannis Samoladas, and Ioannis Stamelos, Trends in Open Source Database Management Systems. In *Laura C. Rivero, Jorge H. Doorn, and Viviana E. Ferraggine (Eds.) Encyclopedia of Database Technologies and Applications*, Idea Group, Inc., 2005, pp: 457-462.
23. David J. Sheskin, *Handbook of Parametric and Nonparametric Statistical Procedures*, Chapman and Halucrc, 2004.
24. Sulayman K. Sowe, Ioannis Stamelos, and Karoulis, Anastesiou, A constructivist View on Knowledge Management in Open Source Virtual Communities. In *Figureiredo, D. A, and Paula, A. (Eds.), Managing Learning in Virtual Settings: The Role of Context*, Idea Group, Inc., 2005, pp: 290-308.
25. Sulayman K. Sowe, Ioannis Stamelos, Lefteris Angelis, Identifying knowledge brokers that yield software engineering knowledge in OSS projects, *Information and Software Technology*, 2006, vol. 48, pp: 1025-1033.
26. Katherine J. Stewart, David P. Darcy, and Sherae L. Daniel, Observations on patterns of development in open source software projects, *5-WOSSE: Proceedings of the fifth workshop on Open source software engineering*, St. Louis, Missouri, ACM Press, 2005, pp: 1-5.
27. Walt Scacchi, Joseph Feller, Brian Fitzgerald, Scott A. Hissam, and Karim Lakhani, Understanding Free/Open Source Software Development Processes, *Software Process: Improvement and Practice*, 2006, vol. 11(2), pp: 95-105.
28. Xu J, Gao Y, Christley, S, Madey S., A topological Analysis of the Open Source Software Development Community, *IEEE Proceedings of the 38th Hawaii International Conference on System Sciences*, (IEEE, HICSS '05-Track 7), Jan 03-06, Big Island,Hawaii., 2005, page 198a.
29. Valverde, S., Theraulaz, G., Gautrais, J., Fourcassie, V., and Sole, R. V. Self-Organization Patterns in Wasp and Open Source Communities. *IEEE Intelligent Systems*, 2006; **21**(2): 36-40.
30. N. Bassiliades, I. Vlahavas, A. Elmagarmid, E. Houstis, Interbase-KB: A Knowledge-based Multidatabase Systeem for Data Warehousing, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 15, No. 5, pp. 1188-1205, 2003.