Business models in FLOSS-based companies[1]

Carlo Daffara[1]
1   Conecta srl, research division
cdaffara@conecta.it

Abstract. The sustainability of Free/Libre/Open Source Software (FLOSS) has recently become an important aspect of the overall open source phenomenon, and as such has received a significant research attention. Since 1999, several articles addressed the role of licenses, communities and business models in FLOSS-based companies, but mostly from a theoretical point of view. In this article, we present the result of a detailed analysis of 80 FLOSS-based companies and business models, using publicly available data on service offers, mailing list archives, online articles and when available public financial data. The result is a simplified and effective classification that can be used as a guide to companies willing to offer services in the same area, and an analysis on how the community process is influenced by the choice of licensing and business model.

1   Introduction

Business models are abstract, conceptual models that represents the business and money earning logic of a company in a structured way [1,2]. The traditional business models for software are challenged by the redistributability of open source software, and the fact that the user/customer does have several additional rights not usually allowed in the traditional shrink-wrapped market, including that of creating derivative works and redistributing them freely. Looking at these challenges, within the context of the FLOSSMETRICS project we are performing a study on the business models adopted by companies that are leveraging FLOSS source code, and how the model changes with respect of licenses and commercialization approaches.

2 Methodology

An initial list of 120 companies was prepared during the first two month of 2007 using some popular open source news websites as source[2]; this list was further refined by eliminating companies that were not really adopting FLOSS, even using a

---

[2]   Among them: FreshMeat, Slashdot.org, OSNews, LinuxToday, NewsForge and some blog sites devoted to FLOSS business models like those of Roberto Galoppini, Matt Asay, Fabrizio Capobianco. Additional information was retrieved from Google searches.

very relaxed definition. In the specific, any company that allowed source code access only to non-commercial users, or that did not allowed for redistribution was dropped from the list; also, companies for which no information was available, or for which no clear product or service was identifiable was equally eliminated. One of the companies included (Sourceforge, from the OSTG group) is not open source in itself[3], but represents an example of an "ancillary" model, as the site itself hosts more than 100000 open source projects and provides supporting services like mailing lists, source code versioning systems and file distribution. Also, companies that have a significant OSS contribution, but for which FLOSS is not the core business model were not included[4].

An initial set of variables were selected, including: choice of licenses, product offering (whether a single version or multiple version of a software system are offered), services offered (divided into installation support, integration, training, consultancy, legal and technical certifications), type of contracts offered (subscriptions, licensing or per-incident) and metering form. Additionally, literature from each company's website was retrieved to find references to the business model adopted and how the model impacts the value proposition of the firm. Mailing lists and search engine searches were performed to obtain indicative references of the relationship of the company with the development community, and if there is an external, non-company based support activity in the form of websites, wikis and knowledge bases.

The collected data was then tabulated, eliminating non-significant variables; for example, coupling together installation, training, support and consulting that were found to be part of the offering of most of the companies that offered support services (and coupled in a single Installation/Training/Support/consulting variable, ITSC). The significant variables left are main revenue generation (the service or contractual offer that provides the main revenue to the company) and licensing model. The first is further subdivided into Selection services (finding appropriate FLOSS packages for a need), ITSC, subscription (a recurring license) and one-time licensing. The licensing model is obtained by looking at the licensing scheme adopted by the company and whether the company services were covering a single software project or a set of projects. By performing a simple cluster analysis on the results, it was possible to identify 6 main models and a remainder group that was separately analyzed.

3 Results

The final result is:

---

[3] The original code for the SourceForge collaborative development environment was open source, and from its change of license several "forks" appeared, including Gforge.

[4] This for example includes IBM, HP and Sun; all of which are important FLOSS contributors, but for which open source software is just one of the overall revenue streams (along hardware, IT services and more).

		Main Licensing model					Main revenue generation			
	Company	twin licensing	OSS and commercial versions	Badgeware	Pure OSS	multiple packages covered	selection	ITSC	Subscription	licensing
twin lic.	Funambol	•						•		•
	Lustre	•						•		
	MuleSource	•		•					•	•
	Mysql	•							•	•
	OpenClovis	•							•	
	Pentaho	•						•		•
	sleepycatdb	•								•
Split OSS/commercial releases	Adaptive Planning		•							•
	Alterpoint		•					•		•
	Altinity		•					•		•
	Codeweaver (WINE)		•							•
	Coupa		•							•
	Digium (Asterisk)		•						•	
	Enormalism		•							•
	EnterpriseDB		•							•
	GreenPlum		•							•
	GroundWork		•						•	
	Hyperic		•						•	
	JasperSoft		•							•
	KnowledgeTree		•	•						
	OpenCountry		•							•
	Open-Xchange		•							
	NoMachine NX		•							•
	rPath		•						•	
	Scalix		•							•
	Sendmail		•							•
	Smoothwall		•						•	
	Sourcefire (SNORT)		•						•	
	Splunk		•						•	
	SSLExplorer		•							•
	SugarCRM		•	•						•
	TenderSystem		•	•						•
	VirtualBox		•							•
	Vyatta		•					•	•	
	XenSource (Xen)		•					•		
	Zend (PHP)		•							•
	ZIMBRA	•	•							•
Badgeware	1bizcom			•				•		
	CATS applicant tracking			•					•	
	EmuSoftware/Netdirector			•				•	•	
	Jbilling			•				•		
	OpenBravo			•				•		
	OpenEMM			•				•		
	OpenTerracotta			•					•	
	SocialText			•						•
product specialists	Alfresco				•			•	•	
	Babel				•			•		
	CentraView				•			•		
	CleverSafe				•			•		
	Compiere				•			•	•	
	Exadel				•			•		
	Jitterbit				•			•	•	
	Mergere				•			•		
	Mindquarry				•			•		
	Mirth				•			•		
	OfBIZ				•			•		
	Qlusters (OpenQRM)				•			•		
	Symbiot/OpenSIMS				•			•		
	Talend				•			•		
	UltimateEMR				•				•	
	VISTA				•			•		
	vTiger				•			•		
	Zenoss				•				•	
platf. Provid.	Jboss				•	•			•	
	RedHat linux				•	•			•	
	Sourcelabs				•	•	•		•	
	SpikeSource				•	•	•		•	
	SUSE Linux				•	•			•	
	WSO2				•	•		•		
selection/consulting	ayamon					•	•	•		
	Enomaly					•	•	•		
	navica					•	•			
	openlogic					•	•			
	Optaros				•	•	•			
	x-tend					•	•	•		
Other	CiviCRM				•					
	Eclipse				•					
	Mozilla				•					
	OSAF Chandler				•					
	Sourceforge									

The 6 main clusters identified are:

- Twin licensing: the same software code distributed under the GPL[5] and a commercial license. This model is mainly used by producers of developer-oriented tools and software, and works thanks to the strong coupling clause of the GPL, that requires derivative works or software directly linked to be covered under the same license. Companies not willing to release their own software under the GPL can buy a commercial license that is in a sense an exception to the binding clause; by those that value the "free as in speech" idea of free/libre software this is seen as a good compromise between helping those that abide to the GPL and receive the software for free (and make their software available as FLOSS) and benefiting through the commercial license for those that want to maintain the code proprietary. The downside of twin licensing is that external contributors must accept the same licensing regime, and this has been shown to reduce the volume of external contributions (that becomes mainly limited to bug fixes and small additions).
- Split OSS/commercial products: this model distinguish between a basic FLOSS software and a commercial version, based on the libre one but with the addition of proprietary plugins. Most companies adopt as license the Mozilla Public License, as it allows explicitly this form of intermixing, and allows for much greater participation from external contributions, as no acceptance of double licensing is required. The model has the intrinsic downside that the FLOSS product must be valuable to be attractive for the users, but must also be not complete enough to prevent competition with the commercial one. This balance is difficult to achieve and maintain over time; also, if the software is of large interest, developers may try to complete the missing functionality in a purely open source way, thus reducing the attractiveness of the commercial version.
- Badgeware: a recent reinvention/extension of a previous license constraint[6], that is usually based on the Mozilla Public License with the addition of a "visibility constraint", the non-removability of visible trademarks or elements from a user interface. This allows the company to leverage trademark protection, and allows the original developers to receive recognition even if the software is resold through independent resellers.

---

[5] An exception is MuleSource, that uses a MPL+Attribution license similar to the "badgeware" license described later. As the MuleSource CEO mentions, "So, if you use Mule in your software product and sell it commercially, then you are required to either make a licensing deal with us or keep the "powered by Mule" logo visible." It is still debated by the community and experts if "badgeware" licenses are really open source; some of those have been submitted to the Open Source Initiative for evaluation.

[6] The original BSD license introduced the "advertising claim", that required the licensee to maintain in the advertising material mentioning feature or use of the software the wording "This product includes software developed by the University of California, Berkeley and its contributors".

- Product specialists: companies that created, or maintain a specific software project, and use a pure FLOSS license to distribute it. The main revenues are provided from services like training and consulting (the "ITSC" class) and follow the original "best code here" and "best knowledge here" of the original EUWG classification [5]. It leverages the assumption, commonly held, that the most knowledgeable experts on a software are those that have developed it, and this way can provide services with a limited marketing effort, by leveraging the free redistribution of the code. The downside of the model is that there is a limited barrier of entry for potential competitors, as the only investment that is needed is in the acquisition of specific skills and expertise on the software itself.
- Platform providers: companies that provide selection, support, integration and services on a set of projects, collectively forming a tested and verified platform. In this sense, even linux distributions were classified as platforms; the interesting observation is that those distributions are licensed for a significant part under pure FLOSS licenses to maximize external contributions, and leverage copyright protection to prevent outright copying but not "cloning" (the removal of copyrighted material like logos and trademark to create a new product)[7]. The main value proposition comes in the form of guaranteed quality, stability and reliability, and the certainty of support for business critical applications.
- Selection/consulting companies: companies in this class are not strictly developers, but provide consulting and selection/evaluation services on a wide range of project, in a way that is close to the analyst role. These companies tend to have very limited impact on the FLOSS communities, as the evaluation results and the evaluation process are usually a proprietary asset.

The remaining companies are in too limited number to allow for any extrapolation, but do show that non-trivial business model may be found on ancillary markets. For example, the Mozilla foundation obtains a non trivial amount of money from a search engine partnership with Google (an estimated 72M$ in 2006), while SourceForge/OSTG receives the majority of revenues from ecommerce sales of the affiliate ThinkGeek site; it is possible to classify those as "public funding" and "indirect funding" following the EUWG classification [5].

4 Conclusions

In this article a simplified classification was presented, based on a cluster analysis of 80 FLOSS companies for which public data is available. We believe that this may help companies that are interested in entering the FLOSS market, by providing a first guidance into the potentially applicable models and their repercussions to revenue

---

[7]   Examples of RedHat clones are CentOS and Oracle Linux.

channels and licensing models. We expect to continue the evaluation process, first by extending the list of companies, and then by measuring additional variables extracted from the FLOSSMETRICS database; this way we plan to investigate the relation between the business model, the simplified company-community categorization (Corporate, Voluntary and Hybrid) and what is the impact of the production model on group forming networks [15,16].

1   Bibliography

[1]Osterwalder A, *The business model ontology - a proposition in a design science approach ,* PhD Dissertation, University of Lausanne, Switzerland*,* 2004

[2]Osterwalder A, Pigneur Y., Tucci C. *Clarifying Business Models: Origins, present and Future of the Concept*, CAIS, 2005

[3]Osterwalder A, Pigneur Y., *E-business models and disruptive behaviours,* http://www.businessmodeldesign.com/presentations/PACIS05.ppt

[4]Daffara, C. *Sustainability of FLOSS-based business models,* II Open Source World Conference, Malaga 2007

[5]Daffara, C. Barahona, J.B. *Free Software/Open Source: Information Society Opportunities for Europe?* working paper, http://eu.conecta.it

[6]Dalle, J.-M., et al., *Advancing Economic Research on the Free and Open Source Software Mode of Production*, in *Building Our Digital Future: Future Economic, Social & Cultural Scenarios Based On Open Standards*, M. Wynants and J. Cornelis, Editors. 2005, Vrjie Universiteit Brussels (VUB) Press: Brussels

[7]Fogel, K., *Producing Open Source Software: How to Run a Successful Free Software Project*. 2005: O'Reilly

[8]von Hippel, E. and G. von Krogh, *Open Source Software and the "Private-Collective" Innovation Model: Issues for Organizational Science.* Organization Science, 2003. **14**(2): p. 209-223

[9]Raymond, E.S., *The Cathedral and the Bazaar*, in *The Cathedral and the Bazaar*. 2000

[10]Spiller, D. and T. Wichmann, *FLOSS 3: Basics of Open Source Software Markets and Business Models*, in *FLOSS - Free/Libre Open Source Software: Survey and Study*. 2002, Berlecon Research: Berlin

[11]Schiff, A. *The Economics of Open Source Software: A Survey of the Early Literature*. Review of Network Economics, 1 (1), March 2002

[12]Fink, M. *The Business and Economics of Linux and Open Source*. Prentice Hall, 2003

[13]Ghosh, R.A.B, Cooking pot markets: an economic model for the trade in free goods and services on the Internet. *First Monday*, 3(3), March 1998

[14]Hawkins R.E., The economics of open source software for a competitive firm - why give it away for free? *NETNOMICS: Economic Research and Electronic Networking*, 6(2), 2004

[15]Garzarelli G., Galoppini R. *Capability coordination in modular organization: Voluntary FS/OSS p roduction and the case of Debian GNU/Linux.* Economics Working Paper Archive at WUST, number 0312005, 2004
[16]Reed, D.P. *That Sneaky Exponential—Beyond Metcalfe's Law to the Power of Community Building* available at http://www.reed.com/Papers/GFN/reedslaw.html