A Framework Analysis of Business Models for Open Source Software

Jesper Holck[1], Roberto Zicari[2], and Volker Mahnke[1]

[1]Copenhagen Business School, Department of Informatics, Howitzvej 60, DK-2000
Frederiksberg, Denmark {jeh.inf, vm.inf}@cbs.dk

[2]Johann Wolfgang Goethe University, Database and Information Systems (DBIS), Computer
Science Institute, D-60486 Frankfurt, Germany zicari@cs.uni-frankfurt.de

Abstract The aim of this research is to identify the key elements in the success of business models for companies producing open source software. We have defined a new framework that can be used to evaluate existing and new business models in this area. Based on our analysis, we have identified two classes of business models for open source which in our opinion have the potential for success: *Commodity Products for Horizontal Markets (CPHM)*, and *Innovative Products for Niche Markets (IPNM)*.

1 Introduction

We have recently seen a growing interest in business models based on open source software, and in this paper we focus on so-called OSSg2 companies (Watson et al., 2008), where each OSSg2 company exploits several business models in relation to one or more open source software products.

We will illustrate our discussions with examples from MySQL AB and db4objects Inc., two companies selling dual-licensed open source database software; MySQL's main product is a standard relational database (Codd, 1970), whereas db4objects produces an object-oriented database (Atkinson, 1989; Grehan, 2005) for the embedded market (Olofson, 2005). Both products are examples of "commercial open source … software that a for-profit entity owns and develops" (Riehle, 2007). As dual licensing requires sole copyright of the product, all source code from external developers must be donated to the company (Välimäki, 2003).

The rest of the paper is structured as follows: Section 2 presents a framework analysis for open source business models, and section 3 describes two classes of business models which in our opinion have the potential for success: Commodity Products For Horizontal Markets and Innovative Products For Niche Markets.

2 A Framework to Analyse Open Source Business Models

For analysing the potential success of open source business models we will in this section suggest a framework, composed of several elements that will be evaluated in detail. The methodology used for conducting this research comprises periodic in-depth interviews, a market analysis, and a review of existing literature.

2.1 Direct Revenue Generation

The number one challenge for any business model, including companies selling open source software products, is revenue generation. *Dual Licensing* is an option for companies with sole copyright of the product's entire source code, allowing them to distribute identical software products under two licenses:

- An open source license, allowing free access to product and source code. The license will typically be "viral" (in the style of GPL, General Public License), specifying that possible product derivatives must be distributed under the same license. As a consequence, other companies cannot sell proprietary licenses to their own products if these include parts of the GPL-licensed product.
- A proprietary license, providing the customer the right to sell its own software products with the original product embedded. The economic incentive of *redistribution* makes dual licensing an important element in a successful business model for software products often embedded in other products. We will refer to this type of customers as *embedders*.

Embedders are not the only paying customers of open source software:

- *Subscribers* to technical support, a classical model for end-users. MySQL AB offers, e.g., the "MySQL Enterprise" subscription in which some tiers include 24x7 access to the MySQL support team[1] (Mickos, 2006), and db4objects offers "dDN Enterprise" subscription, including expert support through live pairing sessions[2].
- *Buyers of extra features*, an option similar to product versioning (Shapiro & Varian, 1998). As part of "MySQL Enterprise", MySQL AB offers a "Network Monitoring and Advisory Service", a component not part of the open source distribution, and as part of "dDN Enterprise", db4objects offers exclusive access to "Stable releases"[3].

[1] See http://www.mysql.com/products/enterprise/faq.html

[2] See http://www.db4o.com/about/productinformation/packages

[3] See http://www.db4o.com/about/productinformation/packages

There may also be customers of “second-level support, training, and development services” (Riemer, 2007), but for the present discussion we will focus on three categories of paying customers: *embedders*, *subscribers*, and *buyers of extra features*.

2.2 Indirect Revenue Generation

For MySQL, the paying customers are certainly a minority: With 50,000 product downloads per day (Kawasaki, 2006) and 11 million active installations (Watson *et al.*, 2008), only 10% of the downloads cause the installed base to grow, and from this only around 0.1% end up as paying customers (Mickos, 2006).

But even non-paying users of dual-licensed software may have positive effects on revenue generation. For the company, the cost of a download is very close to zero, and may help in both *direct qualification* of the user into a potential buyer, and in *indirect qualification*, where the user recommends the product to a potential buyer, e.g. his employer. “Because of the dual-license model, developers can work with the software immediately, without going through management and purchasing, at least for the prototyping stage. Then, developers can go to management with a working, demonstrable project. Now, there's really no decision left for the CIO to make – it's a done deal” (Petreley, 2007). Due to this, MySQL has spent less than 10% of total revenues on sales and marketing, compared with 45-50% for most public software companies (Asay, 2006).

Non-paying users also have an important role in establishing a large and active user community, crucial for the success of commercial open source software. At virtually no cost for the vendor, an active community will provide user-to-user support, bug-finding, and suggestions for improvements, increasing the software's value, and it will be an important promotion channel, increasing the software's market share.

As described by Mickos: “When we release a new version, within 24 hours 35,000 people have downloaded and tested it … Not even Microsoft has 35,000 QA engineers … There are enough people out there who certainly test all relevant features without our specific instruction” (Burgelman *et al.*, 2005).

Even if users don't actively promote the software or engage in user communities, they expand the program’s installed base, thereby making it more attractive for other companies to offer supplementary services, compatibility interfaces etc., and not least reducing competitors’ market shares.

2.3 Commodity or innovative products

An important distinction affecting the success of a business model is between *commodity* and *innovative* products. In our definition, *commodity products* have functionalities and benefits already well understood and accepted by a large share of the target entry market, and can attract large user communities; *innovative products*, on the other hand, have less well understood and accepted benefits and functionalities, and only attract small communities. So an important question is whether an open source company should be a technology innovator, or rather focus on a commodity product. On this point, venture capitalist Danny Rimer expressed a strong view:

> "Open-source companies absolutely can't have a new, innovative technology. They have to be smarter approaches to existing technology. … Open source is about coming up with an alternative that's cheaper, not going after a new area." (Anonymous, 2005)

Rimer identifies "three Cs" as the criteria for a successful business model based on open source (Anonymous, 2005): *community*, *commodity*, and *cushion*. This last criterion is explained as "The open source alternative should be considerably cheaper than the closed software alternative and still allow for a price cushion so that over time, open source companies can charge more while still remaining interesting to customers."

Rimer's description fits nicely for MySQL that succeeded in achieving a strong foothold in a well-established database software market, offering a *commodity* product and only incremental innovations compared to existing, proprietary products (Holck *et al.*, 2008). But db4objects' object-oriented database, db4o , was new and innovative technology at the time of its introduction. As explained by Christof Wittig: "db4o was driven by users who felt there was an urgent need for an object persistence solution that was more efficient and better performing than incumbent solutions based on relational paradigms or flat file/serialization" (Petreley, 2007). In contrast to Rimer's prediction, db4objects has been rather successful, with 15,000 registered developers, 1,000,000 downloads, and a growth rate of more than 100% per year (Petreley, 2007).

2.4 Price and product

Regarding pricing and marketing, we expect the companies' considerations to depend on the product category. For companies offering a *commodity product*, the market must be price sensitive and a considerable percentage of users must be dissatisfied with direct competitors' proprietary products. *Innovative products* must solve specific problems not well handled by mainstream competitors, or address a new market segment with a new value proposition. Here too, the market needs are the key to success. Price differentiation does play a role as well, but not as strong as in the case of a commodity product. On the issue of innovation Christof Wittig[4] told us (Wittig, 2006):

> "It is certainly easier/faster to 'commoditise' a well understood mass-market with a low-cost product. But the economics also say: it is cheaper to build and distribute software. If

[4] CEO of db4objects, a commercial open-source software company using dual-licensing

> it then happens to innovate, it can only be better. It just takes longer, because we have two adoption hurdles to overcome: open source in general (which is less and less a hurdle, but it has been five years ago) and [the] product category itself…. But the economic fundamentals are on your side, and that matters to win a business."

Two important concepts are relevant in the context of pricing and competition for open source software products:

- *Switching costs*. When Rimer says that "the open source alternative should be considerably cheaper", he may consider users being sceptical towards open source software in general, and therefore needing "persuasion" in the form of low prices. But more important are the costs following from switching from one set of software to another. For widely-used commodity products, switching costs are high, and as a result, the open source software alternative must be very cheap. For innovative products, the switching costs may be smaller, and more easily outweighed by the benefits of the new product.
- *Usability*. According to (Sen, 2007), in a market with strong network effects, it is important for an open source software product to have high usability, if it is to successfully compete with existing proprietary software. On the other hand, the product should not be *too* usable, as this will partly remove demand for the additional services.

2.5 Entry Market

We next consider the *entry market* as an important discriminating factor for a successful business model, distinguishing between (Cusumano, 2004):

- *Niche (*or *vertical)* market: market within a specific domain, e.g. specific industry or technical platform, with a narrow set of customers. The customer group is smaller, but also more clearly identifiable.
- *Horizontal* market: market across industry or functional specialisation, broad set of customers. The customer group is large, but it may be difficult to target and deliver your product to everyone.

For companies operating in a *niche* entry market to be successful, we suggest that the products must solve a specific set of problems, not satisfactorily handled by mainstream competitors, must be optimised for a specific market with no existing dominant product, and must have competitive advantage over competitors' products in neighbouring markets. If these conditions are satisfied for an innovative product, operating in a niche market reduces risks and increases chances of success.

For companies to be successful in a *horizontal* entry market, we suggest that they should offer low-priced commodity products, satisfying the market's needs and comparing favourably with competing products. A considerable percentage of users should be dissatisfied with existing products (e.g. price or technical issues), and therefore willing to try out alternatives. For these companies, a large user group is essential: “even if open-source is of equivalent quality and is free of charge, it must still gain a ‘critical mass’ of the market share from those *dissatisfied* with the product of the market leader” (Khalak, 2000). If the above conditions are satisfied, a company offering a commodity product in a horizontal market has a chance of gaining a significant market share.

2.6 Ecosystem

For a company offering an open source software product, synergy with other open source software companies and products in the same or related markets increases the chances of adoption and hence greater revenues. This is e.g. the case for MySQL and the so-called LAMP (Linux, Apache, MySQL, and Python/Perl/PHP) stack. The synergy is not a technical compatibility issue only: successful open source software companies often depend on *open source service networks* (Feller *et al.*, 2008).

A successful open source software business model requires active support from a community, having members with various roles, including core members (managers), active and peripheral developers, bug reporters, and passive users (Ye & Kishida, 2003). For both MySQL and db4objects it has been crucial to create a *coherent* and *active* community out of this broad group of members, motivating them to actively collaborate on supporting and improving the product. According to Wittig and Fiddler of db4objects: “Contracted and paid developers are as much part of the community as any community member is part of our company. There is not an 'us' and a 'them'; there is only a 'we'. … The user community is an integral part of product development and support” (Petreley, 2007).

We can take an even broader perspective and consider the *ecosystem* encompassing an open source software company (Watson *et al.*, 2008), including not only the community, but all entities gaining from the company’s presence in the market, e.g. also support services, authors, publishers, educators, and partners. A project’s chance of success will depend on a large and well-functioning community and ecosystem.

3 Two Business Models Based On Dual Licensing

Several business models have been proposed for open source (McKelvey, 2001; Khalak, 2000; Krishnamurthy, 2003; Onetti & Verma, 2008), but little attention has been paid to dual licensing, the focus of this paper. We have identified two classes of open source software business models which in our opinion have the potential for success: *Commodity Products for Horizontal Markets (CPHM)* and *Innovative Products for Niche Markets (IPNM).*

Both models share many common properties: They are based on dual-licensed, open source software products, with several sources of revenue (licenses, support, proprietary supplements). Spending is focused on R&D, as communities help in testing, marketing, and expanding the customer base. The organisation is virtual with a flat hierarchy structure, developers and community collaborate using the Internet, with little need for physical infrastructure. The Internet is also the primary marketing, sales, and distribution channel, enabling a "bottom-up" sales approach to attract new customers.

The two business models are, however, differentiated as follows: The *CPHM* strategy is based on commodity products in horizontal markets, with large communities; the strategy is to consolidate or even expand the horizontal market. The *IPNM* strategy is based on innovative products in niche markets, with smaller communities; the strategy is to aggressively expand from niche to horizontal market, when market conditions make this possible. Interestingly, the same firm may use both business models for different products. E.g., the regular MySQL product can be seen as a highly popular commodity, but MySQL Cluster and other special versions are technology and innovation leaders (Mickos, 2006).

The first results of applying our framework to a real case study can be found in (Holck *et al.*, 2008), where we discovered that MySQL, using a combination of series of rapid incremental innovation, stack-based complement strategies, an open-source product testing approach, and a dual licensing strategy succeeded in the increasingly commoditized database software markets.

4 Conclusions

In this paper we have discussed existing and new business models based on open source software products, and have identified two strategies which in our opinion have the potential for success for open source software: *Commodity Products For Horizontal Markets* and *Innovative Products For Niche Markets.*

We will use this framework for further work, analyzing and comparing real-life examples of OSSg2 companies with dual-licensed, open source software products. We hope others may do the same, providing both better empirical data and better theoretical understanding of this interesting area.

Mårten Mickos, former CEO at MySQL, Christof Wittig, CEO db4objects, and Ioanna Constantiou, CBS, provided useful feedback on earlier versions of the paper.

References

Anonymous (2005): Rimer's Rules for Open Source, BusinessWeek online, http://www.businessweek.com/technology/content/oct2005/ tc2005103_5298_tc_218.htm.

Asay, M.N. (2006): Open Source and the Commodity Urge: Disruptive Models for a Disruptive Development Process, in DiBona, C., Cooper, D., and Stone, M. (eds.): *open sources 2.0*, O'Reilly.

Atkinson, M., Bancilhon, F. et al. (1989): The Object-Oriented Database System Manifesto, *Proceedings of the First International Conference on Deductive and Object-Oriented Databases*, Kyoto, Japan.

Burgelman, R. A., Inkinen, S., and Wittig, C. (2005): MySQL Open Source Database in 2004, in Burgelman, R. A., Grove, A. S., and Meza, P. E. (eds.): *Strategic Dynamics: Concepts and Cases*.

Codd, E. F. (1970): A Relational Model of Data for large Shared Data Banks, *Communications of the ACM*, 13(6): pp. 377-387.

Cusumano, M. A. (2004): *The Business of Software: What every manager, programmer, and entrepreneur must know to thrive and survive in good times and bad*, New York: Free Press.

Feller, J., Finnegan, P. et al. (2008): Bazaar by Design: Managing Interfirm Exchanges in an Open Source Service Network, in Barret, M., Davidson, E. et al. (eds.): *Information Technology in the Service Economy: Challenges and Possibilities for the 21st Century*, Boston: Springer, pp. 173-188

Grehan, R. (2005), Introduction to ODBMS: http://www.odbms.org/introduction.html

Holck, J., Mahnke, V. and Zicari, R. (2008): Winning Through Incremental Innovation: The Case of MySQL, *Proc. IRIS31 – The 31st Information Systems Research Seminar in Scandinavia*, Åre, Sweden.

Kawasaki, G. (2006): Ten Questions with Marten Mickos, CEO of MySQL, http://blog.guykawasaki.com/2006/08/ten_questions_w_2.html

Khalak, A. (2000): *Economic Model for Impact of Open Source Software*, MIT, Department of Mechanical Engineering: Boston.

Krishnamurthy, S. (2003): An Analysis of Open Source Business Models, in J. Feller, et al. (eds.): *Perspectives on Free and Open Source Software*. MIT Press: Cambridge, Massachusetts. p. 279-296.

McKelvey, M. (2001): The Economic Dynamics of Software: Three Competing Business Models Exemplified through Microsoft, Netscape and Linux, *Economics of Innovation and New Technology*, 11: p.127-164.

Mickos, M. (2006): Personal communications with the authors, November 28, 2006.

Olofson, C. W. (2005): Embedded Databases: The Invisible Engine That Could, http://www.db4o.com/about/company/backgrounder/IDC_419_web.pdf

Onetti A, Verma S. (2008): *Licensing and Business Models*, report University of Insubria, 2008/6.

Petreley, N. (2007): *Interview with Christof Wittig and Jerry Fiddler of db4objects*, Linux Journal 154.

Riehle, D. (2007): The Economic Motivation of Open Source Software: Stakeholder Perspectives, *Computer*, 40(4): 25-33.

Sen, R. (2007): A Strategic Analysis of Competition Between Open Source and Proprietary Software, *Journal of Management Information Systems*, 24(1), pp. 233-257.

Shapiro, C. and Varian, H. (1998): Versioning: The Smart Way to Sell Information, *Harvard Business Review*, 76(6):106-114.

Välimäki, M. (2003): Dual Licensing in Open Source Software Industry, *Systemes d'Information et Management*, 8(1): pp. 63-75.

Watson, R. T., Boudreau, M.-C., York, P. T, Greiner, M. E., and Wynn Jr., D. (2008): The Business of Open Source: Tracking the changing competitive conditions of the software industry, *Communications of the ACM*, 51(4), pp. 41-46.

Wittig, C. (2006): Personal communications with the authors, Dec. 1.

Ye, Y., and Kishida K. (2003): Toward an Understanding of the Motivation of Open Source Software Developers, *Proc. International Conference on Software Engineering (ICSE 2003)*, Portland, Oregon, USA.