Journal of Strategic Information Systems 21 (2012) 274–294

Contents lists available at SciVerse ScienceDirect

Journal of Strategic Information Systems

journal homepage: www.elsevier.com/locate/jsis

Strategies for software-based hybrid business models

Swanand J. Deodhar [a,*], K.B.C. Saxena [b,1], Rajen K. Gupta [c,2], Mikko Ruohonen [d,3]

[a] 405, FPM Block, Management Development Institute, Mehrauli Road, Sukhrali, Gurgaon 122 001, India
[b] Fortune Institute of International Business, New Delhi 100057, India
[c] 14, B-Block, Management Development Institute, Mehrauli Road, Sukhrali, Gurgaon 122 001, India
[d] CIRCMI, Department of Computer Sciences, University of Tampere, Kalevantie 4, 33014 Tampere, Finland

ARTICLE INFO

Article history:
Received 25 October 2010
Received in revised form 1 June 2012
Accepted 1 June 2012
Available online 2 July 2012

Keywords:
Open source software
Software strategies
Software business model
Case study research

ABSTRACT

The open source approach to software development has been used by software organizations in tandem with their existing business models, which are based on proprietary software licensing. This led to the creation of hybrid business models that merge open source and proprietary paradigms. This paper explores the practices used by software product vendors using hybrid business models and proposes strategies emerging out of these practices using interpretive, single case study research design.

1. Introduction

Open source software (OSS) is a software licensing and development paradigm (Lerner and Tirole, 2002, 2005). Its conceptual foundation can be traced back to the Free Software Foundation (FSF). In 1985, FSF was founded by Richard Stallman to counter the growing influence proprietary organizations had over the software that was being developed at institutes like MIT. Founded to counter the proprietary model, the Free Software movement had several distinguishing features, compared to the typical proprietary model of software development. Salient features of OSS include product *licensing* that mandates the source code of the software to be publicly available and modifiable (Lerner and Tirole, 2005), as opposed to proprietary licensing that did not allow users to either see or modify the source code, and *highly collaborative software development* involving user communities (Lakhani et al., 2003; von Hippel, 2005), as opposed to the guarded, in-house software development in the proprietary model.

These practices were later carried forward from the Free Software movement to the OSS paradigm and have since led to a creation of widely-used software products, such as the Linux operating system and Apache web server. From the research perspective, OSS has received considerable research attention, leading to research issues, such as motivation for participation and the competitive dynamics of OSS versus proprietary software (von Krogh and Spaeth, 2007).

For commercial software organizations, the open source approach to software development offered advantages like faster product development and faster product distribution (Vitari and Ravarini, 2009), and perhaps to leverage these advantages, commercial software organizations use the open source approach as a part of their business model. Business models that

* Corresponding author. Tel.: +91 9711396807.
E-mail addresses: deodhar.swanand@gmail.com (S.J. Deodhar), kbcsaxena@gmail.com (K.B.C. Saxena), rgupta@mdi.ac.in (R.K. Gupta), mr@cs.uta.fi (M. Ruohonen).

[1] Tel.: +91 9818787645.

[2] Tel.: +91 9868558258.

[3] Tel.: +358 0442704440.

0963-8687/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.jsis.2012.06.001

integrate the open source approach with the traditional proprietary software business model are called hybrid business models in the literature (Bonaccorsi et al., 2006).

Since its conceptualization by Mahadevan (2000), the idea of the business model has been examined through numerous frameworks (Morris et al., 2005). Although there is no well-accepted definition of the term business model (Hedman and Kalling, 2003), it is understood to represent a *map of activities that link value creation, value delivery, and revenue generation.* Thus, in order to understand the business model of a firm, it is important to understand these interrelated activities. Organizational research is full of concepts that talk about an organization's operations. For example, organizational process refers to an interrelated set of activities that are coordinated to achieve a certain goal. However, business models are highly contextualized to particular organizational settings because, in addition to activities, business models interface with the people and other resources. Hence, it is important to view the underlying activities by keeping the context of their execution intact.

Organizational practice is one of the useful theoretical perspectives that can link the activities or actions of a business to its existence (Kostova and Roth, 2002). Organizational practices are defined as *an organization's routine use of knowledge for conducting a particular function that has evolved over time under the influence of the organization's history, people, interests, and actions.* Perceived from the practice perspective, a general business model and a hybrid business model can be viewed as configurations of organizational practices. Consequently, it can be argued that in order to employ a hybrid business model, an organization must employ and configure the organizational practices that aid in hybridization. However, what these practices are is not known. Hence, there is a need for further exploration and elaboration (West, 2003; Fitzgerald, 2006).

In this study, our goal is to partially address this gap through a single case study research design. The product chosen for the case study was a successful OSS product created by a commercial software organization that has a *hybrid business model*. The success of an OSS product implies community participation and software usage. This is in line with the existing literature on OSS product success (Grewal et al., 2006; Stewart et al., 2006; Subramanian et al., 2009; Lee et al., 2009; Comino et al., 2007). Through our case-based research, we report five management practices and three emergent strategies used in formulating the organization's hybrid business model. The authors believed that the study provides a fresh conceptualization of the phenomenon at hand, and it has theoretical and practical implications for undertaking and using OSS-based hybrid business models.

The paper is structured as follows: In the next section, we elaborate on the evolution of OSS from its ideological origins in Free Software to an approach for software development that can be merged with the proprietary business model. We end the section by highlighting the research gap. In the next section, we provide an explanation of the research methodology, including data collection and data analysis processes. Next, we outline the inferred management practices and strategies. The paper concludes by highlighting the contributions and limitations of the study along with future research directions.

2. Open source software: from an ideology to a business model

2.1. Origins of open source software

OSS originated from a Free Software foundation established by Richard Stallman in mid-1980s in order to counter the growing influence of proprietary software organizations. The Free Software foundation grew from the need to share software code, which was a common practice amongst the scientific community and hackers (von Hippel and von Krogh, 2003). However, Stallman feared that commercial organizations would free-ride on the work of the developers who wanted to release their software in a public domain under the banner of Free Software. In order to counter this, he created a licensing scheme commonly known as a General Public License that prevented the code and any derived modifications from being made proprietary. This feature is known commonly in OSS vernacular as the viral effect.

Stallman's idea of *free software* did not sit well with commercial software organizations; hence, Free Software could not immediately become a mainstream software development model and remained largely in the confines of the research laboratories (von Hippel and von Krogh, 2003). In 1998 Eric Raymond and Bruce Perens coined the term *open source* to move away from the ideologically rigid foundation and offer a more accessible concept to the software organizations (von Hippel and von Krogh, 2003). Open source, indeed, retained a majority of the practices that were prevalent during the Free Software movement, including *copyleft* licensing and community-driven software development.

2.2. Private collective model of innovation

The private collective model of innovation is perhaps one of the earliest attempts to conceptualize the OSS phenomenon. Proposed by von Hippel and von Krogh (2003), the underlying argument of this model is that OSS does not coincide exactly with the private-investment model, where a manufacturer produces goods and hopes to reap benefits from them privately, or the collective-action model, where a public good is created by innovators in the condition of market failure. Thus, von Hippel and von Krogh (2003) concluded that OSS belongs to a third category of innovation model. The authors have described this model more formally as follows: "the developers contribute their private resources toward the development of an innovation and then release it freely. In doing so, they still get private rewards, such as enjoyment, fame, learning, and sense of control over the created innovation" (von Hippel and von Krogh, 2003; p: 216). Fig. 1 graphically depicts the private-collective model of innovation.

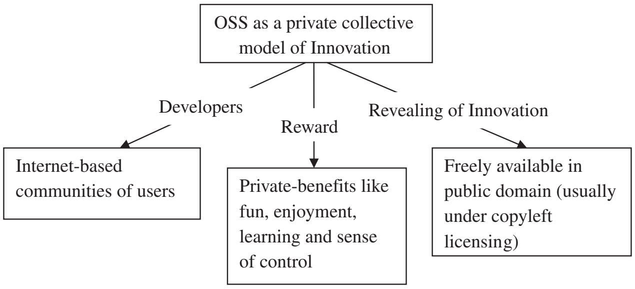
OSS as a private collective model of Innovation
Developers
Reward
Revealing of Innovation
Internet-based communities of users
Private-benefits like fun, enjoyment, learning and sense of control
Freely available in public domain (usually under copyleft licensing)

Fig. 1. OSS as a private-collective model of innovation.

This conceptualization has led to significant theoretical developments in the OSS domain and has been the (implicitly or explicitly) underlying theoretical foundation for many OSS studies, especially those focusing on the motivation for contributors' participation in OSS products (Hertel et al., 2003; Lakhani and Wolf, 2005).

2.3. Development of hybrid business models for software products

The open source approach offers several advantages over the traditional model of software development. These benefits include a faster rate of software development and faster rate of product distribution (Ågerfalk and Fitzgerald, 2008; Vitari and Ravarini, 2009). In order to leverage these advantages, proprietary software organizations are increasingly participating in OSS. However, such participation mandates an inherent conflict where the parties involved have convergent as well as divergent interests, and it is critical to manage both for the success of the *movement* (O'Mahony and Bechky, 2008). The question, therefore, is how such conflicts can be tackled using hybrid business models. Table 1 summarizes existing efforts towards answering this question.

The literature outlined above can be summarized along three dimensions. The first dimension pertains to the motivations and benefits and consequent nature of firms' *interface* with OSS. Research concurs different firms have different nature of interface. For example, Dahlander and Magnusson (2005) outlined three levels of interfacing. According to the authors, organizations can display *symbiotic* (benefitting both the organization and the OSS projects), *altruistic* (benefitting the OSS projects), or *parasitic* (benefitting the organization) approach to interface. However, the authors do not claim that an organization undergoes these phases sequentially. For example, since its inception, one of the studied organizations has adopted a symbiotic approach and continued to use the symbiotic approach while another moved from symbiotic to parasitic. Similar findings were reported by Grand et al. (2004), where the authors developed a four-stage model of firm interface with OSS, which takes an organization from a peripheral presence in OSS (basic level) to an OSS-driven business model (highest level).

Fitzgerald (2006) provided an evolutionary account of the changing nature of OSS and rationalizes the interfacing from commercial software vendors. According to Fitzgerald (2006), organizations participate in OSS to fasten the speed of development. Other benefits include access to a wider work-force. Ågerfalk and Fitzgerald (2008) examined the interface as an extension to the outsourcing phenomenon. According to the authors, community can be treated as a vendor of the

Table 1
Summary of literature on organization's participation in OSS.

Dimension	Research issue	References
Interface	Business rationale for software vendors to interface with OSS. Includes researches on organizational motivation for participation in OSS and possible benefits for software vendors	Bonaccorsi and Rossi (2003), Bitzer (2004), Grand et al. (2004), Dahlander and Magnusson (2005), Feller et al. (2006), Fitzgerald (2006), Ågerfalk and Fitzgerald (2008), Haruvy et al. (2008), Pykäläinen (2008), Lakka et al. (2009), Stam (2009), Hauge et al. (2010)
Involvement	Mechanisms for software vendors to participate in OSS. Includes studies that explain mechanisms/taxonomy of mechanisms for organizations to participate and eventually benefit from OSS	Sharma et al. (2002), West (2003), Kim et al. (2006), West and Gallagher (2006), Dahlander and Wallin, 2006, Dahlander and Magnusson (2008), O'Mahony and Bechky (2008), West and O'Mahony (2008), Shaikh and Cornford (2009), Campbell-Kelly and Garcia-Swartz (2010), Kemp (2009), Morgan et al., 2011
Institutionalization	Development, structure and challenges of OSS-driven business model. Studies under this category look to answer the elusive research question: how OSS-driven business models are structured, get adopted and eventually achieve success? Also, studies under this class explore challenges faced by organizations in exploiting such business models	Bonaccorsi et al. (2006), Feller et al. (2006), Henkel (2006), Rajala et al. (2006), O'Mahony and Ferraro (2007), Sohn and Mol (2008), Riehle (2009), Laine et al. (2009), Campbell-Kelly and Garcia-Swartz (2010), Harison and Koski (2010), Comino and Manenti (2011), Kilamo et al. (2011), Rolandsson et al. (2011), Andersen-Gott et al. (2012)

development services of the software vendor. Community, unlike formal vendor organizations, consists of an *unknown workforce* while software organization is treated as the client. Ågerfalk and Fitzgerald (2008) conclude by identifying service expectations from both vendor (community) and client (software organization) perspectives. In fact, interfacing is reported as an industry-wide phenomenon. Campbell-Kelly and Garcia-Swartz (2010) claim that commercial software vendors are interfacing with the OSS with a strategic intent. Hauge et al. (2010) identified six ways for an organization to interface with OSS. Starting from internal use of OSS components and technologies to use of OSS practices for business, organizations' interface can become increasingly oriented towards its business operations (Hauge et al., 2010, p. 40–41).

It may be deduced that the motivations and intended benefits of interfacing with OSS would eventually translate into actual, multi-modal *involvement*. This signifies the second aspect of the literature, which pertains to the techniques employed by organizations to reach their desired levels of involvement in OSS. The literature provides a repository of such techniques. For example, Dahlander and Wallin (2006) looked at the role of employees as one of the techniques. According to the authors, employees associated with formal organizations had higher levels of interaction with other community members.

The second technique is devising a participation architecture rooted in governance, access control, and ownership (West and O'Mahony, 2008). It is reported that organizations can develop participation architecture in both firm-sponsored and community-driven OSS projects. A firm may also undertake a phased-approach to actual participation, from adhering to open standards to opening the software product itself (West, 2003). In a related study, Dahlander and Magnusson (2008) explained the phased and tactical approach to participation, which begins with *accessing the community resources*, *aligning them to the organization's objectives*, and finally, *exploiting the resources*. Clearly, organizations can choose from a gamut of such techniques.

In a similar study, West and Gallagher (2006) outlined four modes of involvement. These are pooled research and development where firm can involve with OSS to co-develop software, spinouts where internally developed software is released under OSS licensing, donated complements where firm involves OSS communities to develop complementary aspects to the core innovation, and selling complements where the core product is released under open source while complementary services are provided at a price. Dahlander and Magnusson (2008) extended the notion of *absorptive capacity* and identified a set of tactics that can be employed by firms to get involved with OSS communities.

It may not be necessary for all organizations to adopt all mechanisms. In fact, their involvement depends on depth of their interface (i.e. intended benefits and motivations). Organizations that eventually develop OSS-driven business models end up involving OSS as a stakeholder in their functioning. At this stage, involvement becomes the way the business functions. We call this the *institutionalization* of OSS across the organization. Studies on the institutionalization of OSS in a software organization focus on OSS-driven business models.

Under this category of studies, the research issues revolve around developing, adopting, and succeeding under OSS-driven business models. As a corollary, it can be argued that all three actions are not obvious for organizations. Bonaccorsi et al. (2006) reported that is OSS approached as an entry strategy for firms in the software industry where organizations combine their profit motives with the OSS approach, leading to hybrid business models (business models having both proprietary and OSS licensing). Bonaccorsi et al. (2006) also asserted that such business models may not be suitable for all software product organizations. To strengthen their assertion, the authors highlighted organizational factors that influence the adoption of the hybrid business models.

Hemphill (2006) further explored an organization's motivations for adopting *both source business models*. In a more recent republication, Riehle (2012) provided a structural description of *single-vendor commercial open source* business models and explained the rationale for choosing such a business model over traditional single-source business models. Some studies devise classification schemes for business models built around software products. Krishnamurthy (2005) classified business models into four categories based on the roles of intermediaries connecting software producer to software users. Hemphill (2006) classified software business models into three classes, i.e., *closed source, open source,* and *both source* (analogous to hybrid business models). Some studies examined hybridization as an industry-wide phenomenon where the competition and increasing risk have compelled organizations to adopt hybrid business models (Campbell-Kelly and Garcia-Swartz (2010)).

The three phases can be presented as a framework of the phenomenon of interplay between organizations and OSS (Fig. 2). The first phase involves the interface capturing motivations and intended benefits. The second phase is the involvement where the stakeholders face-off. The last phase is institutionalization, where OSS and the organization cease to be multiple stakeholders with organizations adopting OSS-driven business models.

As revealed by the three dimensions, the researchers have placed considerable importance on assessing the dynamics of commercial organizations' interplay with the OSS environment. In this study, we focus on understanding the institutionalization aspects of literature. It is acknowledged that not all organizations can successfully employ hybrid business models (Bonaccorsi et al., 2006). Thus, the following question can be asked: *If an organization wants to achieve institutionalization, characterized by a hybrid business model, then what should it do*?

There are two important aspects to the research question: the *hybrid business model* and *organizational actions*. Business model arguably remains a highly misused term in strategic management literature with no agreed-upon definition (Hedman and Kalling, 2003; Morris et al., 2005). However, it is, in general, an agreement that a business model is a map of activities carried out by an organization to create and deliver value for its customers and generate revenue for itself (Chesbrough, 2003, 2006). Thus if one can devise and describe a configuration of activities (synonymous with organizational actions) so that a software product is produced, delivered, and sold, then a software product-based business model has been formulated.

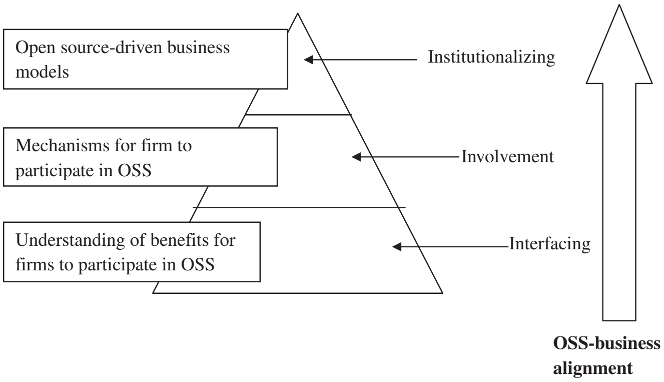
Open source-driven business models
Mechanisms for firm to participate in OSS
Understanding of benefits for firms to participate in OSS
Institutionalizing
Involvement
Interfacing
OSS-business alignment

Fig. 2. Firms' interplay with OSS: conceptual framework.

Thus, formulation of a business model boils down to the formulation of an organization's activities. These are conceptualized as organizational practices in the next section.

2.4. Defining practices and strategies

Organizational *practices* are becoming a trend in the study of organizational strategies. This trend, called *practice-turn* (Whittington, 2006), has deeper roots. Much like a business model, an organizational practice has been defined in many ways. Due to a strong organizational theme, we adopt the following definition: "an organization's routine use of knowledge for conducting a particular function that has evolved over time under the influence of the organization's history, people, interests, and actions." (Kostova, 1999; Kostova and Roth, 2002) This definition characterizes the strong contextual adherence to an organization that was missing from other definitions and frameworks of a business model.

2.5. Linking organizational practices and hybrid business models

As stated earlier, business model is often defined as set of interrelated activities that eventually deliver value to the customer and generate revenue for the organization (Hedman and Kalling, 2003; Rajala et al., 2003; Morris et al., 2005). However, such a conceptualization is likely to ignore the contextual aspects under which such activities take place. Context often becomes important when an organization changes its business model (*business model innovation*) (Chesbrough, 2010). Because organizational practices combine activities with context, it provides a suitable theoretical perspective for studying the formulation of the hybrid business model. The study thus formally defines a business model as: *a collection of interrelated organizational practices that link the value creation, resource organization, value delivery, and revenue generation with the organizational context characterized by interests, people, and history*. This definition can be modified as follows to highlight the idea of hybridization: *a collection of interrelated organizational practices that link value creation, resource organization, value delivery, and revenue generation but are developed by merging open source/proprietary licensing regimes.*

Having defined the business model in terms of organizational practice, the original research question can be rephrased as follows: *What organizational practices are employed by organizations to formulate hybrid business models*? Further, drawing from practice-turn, the question can be augmented: *are there any emergent strategies embedded across these practices*? The paper answers these questions through a case-study research design of a software product having an underlying hybrid business model. The formal research questions for the study are as follows:

3. Research methodology

A case study was adopted as the research methodology. In disciplines like medicine and law, case study research has been the most favored method of investigation while in organizational and social sciences, case study research is gaining acceptance (Lee, 1989; Eisenhardt, 1989; Yin, 2009). The increasing importance of case study research as a methodology is rooted in its potential to expose phenomenon and researcher to each other in a myriad of empirical methods that are otherwise not possible through a positivistic paradigm.

Case study research is suitable under certain conditions only. According to Yin (2009), case study research is appropriate to investigate a phenomenon within its real-life context when the investigator has little control over the events. The research questions at hand satisfy these criteria. The questions were framed to *examine the hybrid business model as a configuration of highly contextual and non-trivially transferrable organizational practices and emergent strategies.* Therefore, a case study method was chosen as the most appropriate approach for this research.

Unlike other popular methodologies, the epistemological roots of a case study are not apparent (for example, survey methodology is assumed in the methodological manifestation of a positivistic stance). Hence, it is important to provide a separate explanation of the study's epistemological stance. In conducting the case studies, the *modified* grounded theory approach was adopted. Conceived first by Glaser and Strauss (1967), grounded theory pertains to the identification of constructs and their relationships by examining the raw data collected from the field. Raw data can be collected using multiple techniques, such as interviews, participant and non-participant observations, and archival documents.

The philosophical roots of classical grounded theory relate to the *interpretive paradigm of research.* The interpretive paradigm negates the idea of absolute truth, which is central to the positivistic paradigm. An interpretive researcher would construct his/her view of reality by interpreting the data (Klein and Myers, 1999). Thus, if a researcher is able to induce any analytically generalizable statements about the phenomenon, then that statement is grounded in the data. This approach of inducing theory from the data reverses the deductive logic where the researcher proposes a relationship between constructs and then proceeds to test this relationship.

Much like the purists in the interpretive paradigm argue, grounded theory, in its traditional sense, demanded a theoretical clean slate. In fact, this was one of points of argument between the creators of grounded theory. Strauss and Corbin (1990) outlined a more formative, structured approach toward conducting research using the grounded theory approach, which was rebutted by Glaser (1992), who stated that grounded theory should involve the creativity of the researcher in inducing knowledge from raw data.

However, it has become increasingly acknowledged that with advancements in theoretical foundations and increased exposure, it is very difficult for a researcher to have a clean slate about any phenomenon (Eisenhardt, 1989; Pettigrew, 2000). Every researcher is likely to have certain preconceptions about the research problem(s) at hand. Thus a researcher goes into the field with his mental models in place and guides the data collection and analysis procedures; this is *modified grounded theory*. The challenge, however, is to avoid enforcing this model on the data but, instead, allow the data to confirm the model. If the data does not fit the model, then the model has to change.

It is clear from the earlier theoretical treatment of organizational practices that authors indeed had certain preconceptions about the phenomenon. It could be described as *hybrid business model, formulated through configurations of organizational practices that in turn are manifested through actions and contextualized through people, history, and interest*. This understanding guided the data collection and analyses procedures; therefore modified grounded theory was employed in this study.

Before any case study endeavor is undertaken, important considerations had to be handled. In the subsequent sections, treatment of these considerations in the context of the study is explained. The first issue is about the *number of cases*. As outlined by Yin (2009), a multiple case study approach is suitable for a theory development exercise where each case is a *separate experiment* in itself. Each case compares the theoretical understanding and the new empirical evidence. On the other hand, single case study is suitable when a phenomenon requires studying a unique, critical, or revealing case. Prominent examples of such cases in organizational and social research include Kling's (1962) study of the Cuban missile crisis and Markus and Pfeffer's (1983) study of the implementation of an accounting system.

In using the single case, the idea was to provide insights into hybrid business models as a configured system of organizational practices and emergent strategies, which had not been previously examined. The authors did not aspire to make grand claims or generalizations about a set of practices that would be part of *all forms of hybrid business models*; rather, we found it useful to conduct an in-depth analysis of a single case study to identify specific practices and strategies.

The second consideration in case-based research pertains to the case selection criteria. Explicit mention of case selection criteria is a key indicator of the rigor in case research (Yin, 2009). Selection criteria for this study included the *presence of a hybrid business model associated with a successful software product*. The idea was to ensure the fit of the case with the research questions at hand. With these case selection criteria, we chose Openbravo ERP, a firm-sponsored ERP product with a hybrid business model, as a case study.

Thirdly, case study protocol was used to guide data collection. We developed protocol with an objective of uncovering process data from Openbravo ERP. Thus, most of the questions in the protocol focused on the how aspect of the operation of Openbravo ERP. Table 2 provides a basic overview of the primary data-collection process. We used interviews as the primary data collection vehicle. Interviews are considered particularly suitable for interpretive case studies (Walsham, 1995). Due to geographical limitations, interviews were conducted and recorded through video conferencing. Later, these interviews were transcribed (for description of analysis the process, see Section 4). We also used data from published news articles and social media platforms.

The fourth consideration is of analysis and it relates to keeping one's theoretical hold on a large amount of data. We follow a three-step analysis strategy that ends in highlighting strategies from reported organizational practices (for a detailed explanation of the analysis process, see Section 4).

3.1. About openbravo

Established in 2001 under the name Tecnicia, Openbravo began functioning as an open source company in 2006. Before this adaptation, they were involved in custom application development and consulting for Spanish universities. Openbravo's key offerings include an open source web-based ERP called Openbravo ERP and a Point-of-Sale system called Openbravo POS. Both these products are positioned as platform products targeted at small and medium enterprises. In this study, we focus

Table 2
Timeline of primary data collection.

Respondents	Date of interview	Description
Chief executive officer	11th February, 2010	Initial contact with Openbravo for explaining case proposal and access requirements
Chief technology officer	23rd February, 2010	Responsible for overall product strategy
Product development manager	3rd March 2010	Responsible for managing internal processes for product development
Business development manager	5th March 2010	Responsible for overall product marketing and business development
Manager, Openbravo Forge	11th March 2010	Responsible for maintaining Openbravo Forge; a custom platform for developing software related to Openbravo ERP and Openbravo POS
Manager (Asia–Pacific), partners' network	16th March 2010	Responsible for maintaining network of ISV, retailers etc. in Asia–Pacific
Chief technical officer (email communication)	12th April, 2010	Sharing of the first draft of the case to seek further inputs on the key management practices identified and areas of strategic importance
Business development manager and product development manager (email communication)	June 21st 2010	Sharing of the second draft of the case to seek further inputs on the key management practices identified and areas of strategic importance

exclusively on Openbravo ERP, which uses a customized license (Openbravo Public License) that is based on the more popular Mozilla Public License (MPL).

Openbravo ERP has attained significant distribution. Data collected from news items stated that Openbravo ERP released the source code under an open source license in April 2006. In September 2006, Openbravo ERP was ranked number three on SourceForge.net (one of the largest hosting platforms for open source products), and in October 2006, Openbravo ERP was ranked the number one OSS product on SourceForge.net, which at that time hosted 130,000 products. In January 2007, the download count of Openbravo ERP reached 200,000, and it swelled to 500,000 in February 2008.

Openbravo ERP was also successful in securing considerable venture capital funding. In May 2008, Openbravo ERP received 12 million Euros in venture capital funding. Its first round of funding was worth 5 million Euros, which was received in January 2006. The product also succeeded in acquiring partners through its partnership program launched in March 2007. As of July 2009, Openbravo's partner network consisted of over 100 partners in 40 countries. Partners not only contributed to product distribution but also to product development as many partners also participated in creating vertical extensions and localization modules.

4. Analysis

We followed a three-step process for analysis. In the first step, we actually identify the practices by comparing excerpts from the interviews. An interview excerpt is treated as explanation of an organizational practice if it satisfies any of the following criteria: an explanation the procedure of doing a certain task (*actions*), *a* perceived benefits from a certain action (*interests*), a historical reason for making a decision about an action (*history*), or the role of different sections of the product community and employees in an action (*people*). These criteria are derived from the definition of organizational practice given in Section 2.4.

Heuristics that were used to identify an interview excerpt with any of these four categories are as follows: an excerpt explaining a historical event – "*History of Openbravo was that*...", "*We used to have it in early years like*..." – was attributed to history. Interview excerpts explaining the action – "*we do*...", "*the process we follow is*...", "*we collaborate with the community by*..." – were attributed to action. Excerpts explaining the rationale for an action – "*to balance the commercial efforts*...", "*to enforce the businessmodel*...", "*in order to have*..." – were attributed to interest, and finally, excerpts explaining the role of the product community and/or Openbravo employees – "*when I first joined thecompany*...", "*there are always people in the community that*..." – were classified under people. Table 3 provides a summary of the all codes identified in each of the four categories.

Once the interview excerpts were classified into these four categories, we scanned through these categories to identify possible links. In order to establish a link, we focused on excerpts categorized under action and tried to see if excerpts in other categories described the context for the action to take place. Once, the link was established, we treated each link as an organizational practice. This exercise led to the identification of eight organizational practices (Table 4). When no further links could be established (i.e., no quotes from interest, history, or people could provide a meaningful context for a quote from action), the identification of practices was deemed complete. This approach is similar to theoretical saturation, explained by Eisenhardt (1989).

In the second step of the analysis, we developed organizational practice stories for a detailed description of the practice by explaining the role of the organization's history, people, interests, and/or actions. Organizational practice stories as a term, was adapted from *decision stories* coined by Eisenhardt and Bourgeois (1988).

Table 3
Summary of the excerpts generated in each of the categories.

Seed category	Total Number of excerpts	Examples of excerpts
Interest (contextual)	22	*"they had an objective to serve local companies with an ERP"* *"they wanted to leverage Open source technology components"* *"our justification was that this is a professional service that costs us a lot of money to provide because we need to release lot of these maintenance packs therefore it's something that should be exclusively available to our professional edition users"* *"So having an active and involved community was very important for us"*
History (contextual)	4	*"History of Openbravo was that it was a project started by a group of local software developers in Pamplona"* *"that time in 2006 there were only three open source ERPs so the space wasn't as crowded as it is now"*
People (contextual)	28	*"We have been very lucky we had at that time a community manager who had a lot of experience in open source, was one of the leading open source people in Spain, had published books, participated in localization of Open Office. He helped manage the community and put in place a very community friendly culture within the company that allowed us to really maintain a very healthy community"* *"In enterprise information system, your end-users are business people like accountants, CFO, manufacturing... they do not have interest and do not have skills to contribute to the project."* *"Our community is a different from other communities in a sense that we are a bit removed from the end users and our community is more made of people that provide services around our product."*
Action (contextual)	30	*"They decided to recruit management, people with more experience that could move project forward"* *"We have an indirect distribution channel both from our commercial standpoint but also from community standpoint."* *"At the beginning of each development cycle we have a kind of an open discussion with our community about what should be the features that should be targeted in the next release. We do that leveraging the forum, issue tracking system, but we also use user-voice to do voting so we have an open voting."*

Table 4
List of organizational practices.

Organizational practices	Role in proprietary software business model
Functional differentiation between community and enterprise editions	Proprietary software is developed under a single licensing regime and therefore organization does not have to maintain two different editions. Therefore, this practice may not hold true for proprietary software business models and hence was taken up for further analysis
Linking different sections of the community for product development	The product development is largely in house and hence software development efforts are easier to coordinate as *only the insiders have access to the source code and can make modifications* (von Krogh & von Hippel, 2003)
Providing Freedom to Develop Proprietary Extensions	Most of the code-base developed proprietary software is by definition proprietary. Hence, the question of choosing to develop proprietary or open source extensions does not exist. Therefore, this practice may not be applicable for proprietary software business model and hence it was taken for further analysis
Free Access to Maintenance Packs for Community Edition	As stated earlier, there is usually only one licensing regime for proprietary software. There is no issue of deciding on support and maintenance of software's free edition. Therefore, this practice may not be applicable for proprietary software business model and hence it was taken for further analysis
Customized Platform for Open Product Development	Product development is largely in-house with little code-level contributions coming from outside. Hence, organization may not invest in developing such a customized platform for supporting product development. Hence, this practice was taken for further analysis
Adopting platform-based strategy for software product	This practice is general as outlined by Rajala et.al (2003) and may therefore be adopted by proprietary and hybrid business models alike. Under this practice, the product is released as a standardized product on top of which more specialized products can be developed and run (for example, ERP is a platform with module for Shipping industry is an extension running on it)
Developing partners-based distribution model	Developing partner-based distribution model is a common practice (Rajala et.al, 2003) across software products. Under this practice, the product is made available through a network of intermediaries and the software vendor remains insulated from the end consumer
Product development as per a development roadmap	As the product development manager for Openbravo ERP indicated, most organizations follow a structured product development roadmap: "*Maybe this is what any company does not only open source or commercial. Of course we have the roadmap of our development*" [Product development manager, Openbravo ERP; Classification: Action]. Roadmap represents an artifact that outlines the progress of the product and captures the feedback given by the users (feedback usually translates into incorporation of new features and fixing issues in existing functionality)

Out of the eight practices, five were found to be unique to a hybrid business model (these practices had both open source and proprietary components) while the other three were found to be applicable even in pure proprietary business models. A practice was considered proprietary if *it was independent of the underlying licensing system*. For example, a practice of developing a partner-based distribution model can be adopted by any software product firm because it is independent of the underlying licensing system. Because non-hybridized practices did not signify the hybrid nature of the business model (they could be adopted even by software product organizations with a non-hybrid business model), these practices were not elaborated upon.

Thus, five stories (corresponding to five hybridized practices) were generated (representing five organizational practices for a hybrid business model). These stories were then shared with the respondents to get their feedback. This step ensured the rigor in case-based research (Yin, 2009; Dubé and Pare, 2003). Based on the feedback, the organizational practice stories were refined.

Organizational practices were transferred to hybrid business models either from pure open source or pure proprietary approaches. The transfer of practices is a non-trivial issue, primarily because of the high contextual adherence to a particular organizational setting (Kostova and Roth, 2002). In the case of the hybrid business models, adopted practices were tweaked from their original form to suit the hybridized nature. In developing practice stories, an attempt was also made to identify the root practices in open source or proprietary approaches.

In the last step of the analysis, we further examined organizational practice stories to identify if these practices complimented each other. One feasible way, which we eventually adopted, was to compare the organization's interest (elicited in second step) in developing a certain organizational practice. The underlying assumption of this approach was that strategies eventually translate into practices (For example, Schuler and Jackson (1987) linked competitive strategy with the human resource practices in an organization.). If practices complemented or contrasted each other, we explored corresponding interests for such practices, and if we could argue on the rationale for the existence of these complementary or contrasting practices, then we merged them into one group. Such groups were then named based on the common interest served by the combined organizational practices. This exercise led to the identification of three underlying strategies that are at a higher level of abstraction than the aforementioned practices. Much like the theoretical saturation achieved in the identification of practices, the identification of strategies was deemed complete when *no interest-driven combination of practices could devise a new emergent strategy*. This completed the procedures for analysis (Fig. 3). Subsequent sections describe the practices and emergent strategies. All the excerpts related to each practice were not provided due to space constraints.

4.1. Practice I: Functional differentiation between community and enterprise editions

Openbravo ERP was available under two editions. The first edition was the community edition, and the second was the enterprise edition. The purpose of the former was to attain distribution of the software while that of the later was to generate revenue through sales of the enhanced functionality and support. In addition, it was easier to convert the users of the community edition into users of the proprietary edition. Following quote from Business development manager clarifies the rationale of having multiple editions.

> *For us, the community product [having a] commercial friendly MPL-based license is [like an] ambassador. We want a small company [to] take it and simplify their business and actually use it in production. We are happy with that. We want a large pool of small companies [to use] our community edition*[Business development manager, Openbravo; Classification: Interest].

However, the community edition, with its broad-based functionality, acted as a competitor to the proprietary edition. The third-party service providers developed their own business models around the community edition, stifling the enterprise edition.

> *The problem [we] actually run into is that [our] partners in most cases come from open source community. They are the services guys. They see our ERP as service and delivery platform [but] they are not geared up for selling enterprise software [...] So one of*

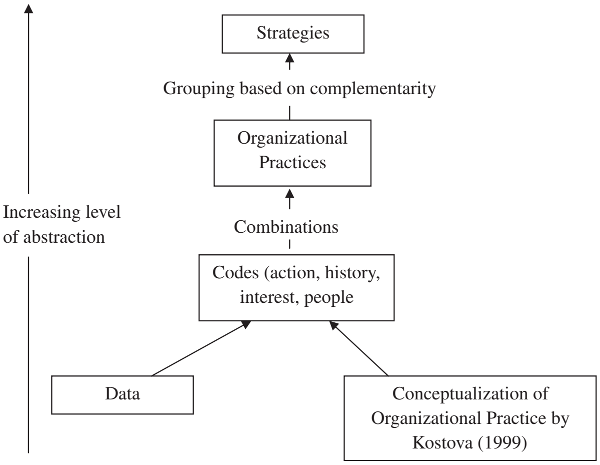
Strategies
Grouping based on complementarity
Organizational Practices
Combinations
Codes (action, history, interest, people
Data
Conceptualization of Organizational Practice by Kostova (1999)
Increasing level of abstraction

Fig. 3. Analysis and theory development process.

> *our challenges has been that our community edition is very broad and it does a lot combined with indirect channel with partners servicing it*[Business development manager, Openbravo; Classification: People].

Most of existing studies on the competitive dynamics of OSS compare OSS and proprietary software that were developed by different vendors as competitors with a classic case of competitive analysis in the operating system market (Casadesus-Masanell and Ghemawat, 2006). However, the community and professional editions of Openbravo ERP were developed by single vendor; yet, they faced competition from each other because of increasing the similarity of their functions. Thus, the challenge was to maintain a community edition as a diffusion vehicle and, at the same time, maintain a sufficient distinction between the community and enterprise editions.

For this, Openbravo decided to differentiate between the community and professional editions through functionality and support. As the business development manager stated, this approach included "narrowing the functional base of the community edition":

> *You need your professional edition [of the software product] to be sufficiently differentiated from your community edition. So if you look at our roadmap, one of the key points in this roadmap is narrower functional base*[Business Development Manager, Openbravo; Classification: Interest].

Thus, in order to thwart competition from the community edition, Openbravo had to create multiple support- and functionality-based differentiations between the community and enterprise editions.

> *In terms of distribution of maintenance patches; our source code branch is open for everybody but of course you need to know [about using] the source code repository and you need to be a bit techie to download [source code from repository], to pull changes from the repository, and install it to [your] system. For professional, we provide two-click installation of maintenance pack, which is distributed for professional edition in a pre-packaged manner. In our professional editions, we have some professional tools and functionalities. Our professional edition customers have a warranty that the bugs that they report will be fixed in a pre-defined period, and secondly, they have a warranty that bugs will be back-ported to their previous release of the Openbravo system. Therefore, we [provide] backward support [for] several major releases. For the community [edition], we treat [bugs] and resolve them but the community doesn't have warranty [of definite time period of fixing issues]*[Product Development Manager, Openbravo; Classification: Action].

In summary, Openbravo ERP developed differentiation between their community and enterprise editions at both the functionality- and support-level. However, both the editions depend on the same code-base.

4.2. Practice II: Linking different sections of the community for product development

Openbravo's product community is made up of three entities: end-users (enterprises where Openbravo ERP actually was implemented), third-party service providers who provided services based on Openbravo's free version, and partners for Openbravo ERP (service providers engaged in providing services for the enterprise edition of Openbravo ERP and had a formal, contractual agreement with Openbravo).

> *Our community is made of three constituents. On one hand, we have end users, which I would say, is probably the minority [segment] of our community. The second segment is people that implement the products [for] their customers. So they are system integrators and value added resellers that provide services around our product, but they choose to work in the pure community vault without any commercial relationship with us. Third constituent are our business partners who are people that provide services but they choose to do it on professional edition and have commercial relationship with us*[Chief Technology Officer, Openbravo; Classification: People].

Amongst the members of the product community, end users possess little understanding of the technicality of the Openbravo ERP, with many end-users remaining oblivious to the fact that Openbravo ERP is open source. The product development contributions mainly come from the third-party service providers for both the free and enterprise editions.

> End-users are business people like [those from] accountants, CFO [etc.]... They do not have interest and do not have skills to contribute to the product [technically]. In certain cases, they don't even know that they are using an open source product. People who participate in the community, post on forums, log defects, make contributions, are system integrators and value added resellers that implement the product at the end customers[Chief Technology Officer, Openbravo; Classification: People].

These partners are globally distributed and Openbravo ERP, with its inherent nature, needed to be localized to accommodate spatially specific features. Thus, contributions coming from the partners were largely in the form of localization products. However, some features were general enough to be applicable in functionality but had cosmetic differences like language. Therefore, Openbravo acted as a link between community members making complimentary contributions in order to facilitate the extension of the functionally of similar features to new geographical areas. This practice was reported in the form of an anecdote.

One of the Openbravo partners had built a commercial extension specific to the hotel management industry on top of Openbravo ERP. However, that particular partner did not have the resources to customize the same extension to the Spanish market. Openbravo linked this partner to a Spanish partner who had the capability to localize the extension to Spanish market. In this way, the Spanish partner who localized the software module earned revenue through selling the extension in Spain. The partner who developed the module received a royalty per implementation, and Openbravo received a commission for being the intermediary.

4.3. Practice III: Providing freedom to develop proprietary extensions

Openbravo's distribution model is heavily dependent on the partner network. In order to attract partners Openbravo wanted to provide them with a mechanism to develop their own business on top of the Openbravo ERP platform. For this purpose, they derived a special license called the Openbravo Public License (OBPL) from the Mozilla Public License (MPL).

> *With regards of [choice of licensing between] MPL vs. AGPL, one of our objectives is to enable our partners to make business leveraging our product*[Chief Technology Officer Openbravo; Classification: Interest].

This facilitated the development of commercial extensions on top of Openbravo ERP and flexibility for users as this license safeguarded both commercial and non-commercial interests. Thus, Openbravo did not follow the usual dual-licensing approach; rather, it developed a single license that incorporated both OSS and commercial interests and provided flexibility to its partners to develop proprietary applications on top of Openbravo ERP.

On the other hand, the license also kept the open source aspect of Openbravo's business model intact. As with OBPL, those sections of the community with no access to the enterprise edition could still access the core of Openbravo ERP and contribute their code as well as their extensions.

> *We have, therefore, chosen MPL as the basis for our license because it makes the ERP code available for extension, customization, and integration with proprietary software and gives intermediate distributors and integrators the freedom to choose their licensing model*[Openbravo Licensing Document; Classification: Action].

4.4. Practice IV: Free access to maintenance packs for community edition

As explained earlier, the open source version of Openbravo ERP was used to drive the distribution of the software. However, subsequent maintenance was available with the professional edition of the ERP. This policy was used to enforce Openbravo's business model, which depends significantly on revenues generated through the sale of support services and maintenance.

> *At that time, we had a policy where maintenance was made available only on the professional edition*[Chief Technology Officer, Openbravo; Classification: Interest].

However, it was perceived by the community that the community edition was not usable at a production level. Therefore, the product community of Openbravo ERP wanted access to the maintenance solutions.

> *Some users took a very harsh stand saying that it is not possible to use the enterprise system without having [an] access to maintenance and that, therefore, community edition was not really a production-grade system without maintenance* [Chief Technology Officer, Openbravo; Classification: People].

Openbravo had to accept a tradeoff because providing free access to maintenance solutions would reduce the relative value of the enterprise version. Yet, Openbravo ERP altered its support policy to provide maintenance services (known as maintenance packs) freely to all its users.

> *Regarding the maintenance patches, we have the following policy. Our development pack branch and maintenance pack branch is open in terms of source code. So, anyone can access our source code repository and can get the maintenance patches*[Product Development Manager, Openbravo; Classification: Action].

Openbravo had the option of either providing or denying free access to the maintenance solutions for the community edition. The approval would reduce the value difference between the community and proprietary editions, but denial would render the community edition not ready for production-level usage. Eventually, Openbravo chose to provide the free access to its maintenance solutions for users of its community edition as well.

4.5. Practice V: Customized platform to support open product development

Openbravo ERP was initially hosted on SourceForge.net, which is a software-hosting platform. SourceForge.net is a popular platform for hosting OSS products (Comino et al., 2007; Lee et al., 2009). However, with evolution of the software, Openbravo ERP faced certain issues in hosting the product on a general platform. It became increasingly difficult to map the product structure with the platform:

> *We wanted to be able to customize the features that are available to our users. You have dozens of localization packages, charts of accounts, little extensions and adapters. All these things circulate around the main product and such a structure is difficult to*

> *map on a public space, such as SourceForge where [if] you are searching for something, [then] you [will] get thousands of results, but you will also get lot of non-related results. So, it is much more difficult to navigate, to browse for something related to Openbravo ERP [on a public development platform]*[Product Development Platform manager, Openbravo; Classification: Interest].

However, the platform has different functionalities for different sections of the community. For example, Openbravo's partners can decide on the accessibility of information concerning their participation. This also includes restricted access to the product development being carried out by the partner. On the other hand, the product development being carried out by other sections of the community is publicly available.

> *When you are hosting your project that you don't want to be public, this is a feature that we are reserving to our partners. A typical example would be partners having development project to adapt the product to a specific customer. They want to implement [Openbravo] at a specific company. They have extension and adaption. You [as a partner] still want to [have a] place where you, your consultants, and your customer representatives are [able to] discuss about features. You can still use this platform. It has the same benefits, but you control who is able to access it, so public will not be able to see it. That's a special-use case of the platform; there few more of [private] projects and for the general public, we are open*[Product Development Platform manager, Openbravo; Classification: Action].

4.6. Underlying strategies

So far, the paper has identified the management practices involved in hybrid business models. In this section, we identify and elaborate on the underlying strategies for the aforementioned five management practices. As stated in Section 2.4, strategy, in the context of this study, is defined as a pattern in a stream of decisions (Mintzberg, 1978). If the practices could be linked, then an emergent strategy exists. In order to identify the underlying linkages across practices, the basic aspects of history, interest, and people were used. Practices could be grouped if they had similar historical reason, if they involved similar people, or if they catered to a similar interest. Naturally, the three approaches resulted in three different groupings.

As the objective was to identify *strategies* that were focused on the organization's broad interest of formulating a suitable hybrid business model, we used interest as the grouping criteria. We focused on interview excerpts classified under interests. The idea was to see if these quotes revealed interests that complemented each other. If complementary interests were found, then corresponding practices were merged under one strategy. If two practices showed no links across their *constituent interests,* then they were not grouped. With this approach, we found that the five practices covered in Sections 4.1–4.5 resulted in three strategies. These strategies are *selective revealing, segmented meshing,* and *strategic licensing*. Table 5 provides an overview of the three strategies with corresponding management practices and descriptions.

4.6.1. Selective revealing

Practice one (differentiating between community and professional editions) and practice four (providing free access to maintenance fixes for customers of the community edition) have contrasting interests. Practice one differentiates between the two editions of Openbravo ERP while practice four reduces the differentiation between editions. In order to explain the underlying strategy for these two contradictory practices, we expand on the work of Henkel (2006), who based on the study of firms in embedded Linux domains, proposed that software organizations can leverage associated external contributions by striking the right balance between sharing and protection (Henkel, 2006, pp. 967). Henkel (2006) calls this phenomenon selective revealing. Further, Henkel (2006) identified certain organizational characteristics, such as size, experience with OSS, and policies, as determinants of selective revealing. In addition, Henkel (2006) identified organizations' objectives to reveal a certain set of functionality. The role of an organization's objective is evident in practice; one illustration is Openbravo's restricted access to some extensions of the software. Openbravo's objective was to create differentiation between the community and professional editions. Thus, in practice one; Openbravo's objectives resulted in selective revealing.

However, in practice four, Openbravo was actually compelled to reveal the maintenance solutions for its community edition, and it continues to provide free access to the maintenance patches. As the users of the community edition perceived maintenance solutions to be of critical importance in running the community edition of Openbravo ERP for production purposes, Openbravo had to release these maintenance solutions to ensure the stability of its community edition. Thus, we propose that in addition to organization's motivation, the value of revealing particular aspect of the software is another determinant of the selective revealing exercise.

Table 5
Mapping between strategies and practices.

Strategy	Constituent Practices
Selective revealing	Free access to maintenance packs for community edition, Functional differentiation from community edition
Segmented meshing	Linking different sections of the community for product development, customized platform to support open product development
Strategic licensing	Providing freedom to develop proprietary license

Hence, we propose that selective revealing is an underlying strategy for practice one and practice four, and it can be defined as follows: the free provision of otherwise economically rentable aspects of the software. The selection of the software part/s for revealing is contained in the organizations' objectives and the value of revealing a particular aspect of the software.

4.6.2. Segmented meshing

Similar to the selective revealing practices; practice two and practice six are contradictory. Practice two (linking community members involved in complimentary development efforts) is aimed at linking different sections of the community, whereas practice six (providing customized product development platforms where some part of the platform may be accessible by only certain sections of the community) separates a certain section of the community from others. The answer to this contradiction lies in the way community is conceptualized by Openbravo.

> *Our community is made of three constituents. From one side, we have end users, which I would say, as I said before, is probably the minority of our community, and second ones are people that implement the products... the projects at their customers. So they are system integrators and value-added resellers that provide services around our product, but they choose to work in the pure community vault without any commercial relationship with us, and third constituents are our business partners that are people that provide services, but they choose to do it on professional edition and having commercial relationship with us*[Chief Technology Officer, Openbravo ERP; Classification: People].

Existing researchers in OSS have treated the community as a single social structure when it is perceived externally. For example, West and O'Mahony (2008), in their conceptualization of community participation architecture, did not attempt to rationalize the unified view of the OSS community. From practice two and practice six, as well as from the quote above, we propose that a software firm with a hybrid business model should look to either segment or merge with the open-source community, based on role they play in different aspects of the hybrid business model. One way of achieving segmentation or meshing can be through the facilitation of development efforts by the organization. We called this strategy segmented meshing.

In practice six, Openbravo separates some sections of the community (namely partners) from other sections (such as users of the community edition) as the separated sections have different roles to play in Openbravo's business model. Users of the community edition drive the distribution of the software. On the other hand, partners help generate revenue for Openbravo. Clearly, the roles played by the two sections of the community are very different from the perspective of the software distribution and revenue models. Therefore, for Openbravo, it is reasonable to segment the two sections.

However, in practice two, Openbravo links different community members based on the similarity in the roles these members play in software development. Hence, it is important for Openbravo to mesh different community members to foster software development (in the reported anecdote, it is faster localization). Thus, it is evident that segmented meshing is the underlying strategies for these two practices (practice two and practice six).

4.6.3. Strategic licensing

Openbravo ERP chose a product license that facilitated two aspects of its business model. First, it demonstrated adherence to the standard norms of an open source license as it was based on an already existing open source license (Mozilla Public License), and secondly, it allowed its partners to develop proprietary extensions. For Openbravo ERP, the partner network was considered to be the most critical aspect of the business model as it depended almost entirely on its partners to manage the implementation and maintenance of the software for the end-user.

> *So, whatever business we do, we do through partners. Whatever services we provide to end customers are also through partners [...] even it happened that we have lost some opportunities because they wanted to work directly with us*[Partner network manager, Openbravo ERP; Classification: action].

In other words, for Openbravo ERP, the choice of licensing was important not only for ensuring the sustainability of the open source aspects of its business model, but it was also important to foster its partner-driven distribution model. Thus, we propose that Openbravo's approach to licensing, which promoted its partner-driven distribution model (practice three) and kept the open source aspect of the business model intact, is linked with an underlying strategy to strengthen different aspects of business model through licensing.

We call this strategic licensing and define it as follows: the selection of a software product license to complement the unique and competitively advantageous aspect of the business model, as perceived by the organization, as well as to sustain the product community.

5. Discussion and future research

Organizational scholars are beginning to acknowledge the increasing use of hybrid business models for software products (Fitzgerald, 2006; Bonaccorsi et al., 2006; Ågerfalk and Fitzgerald, 2008; Hemphill, 2006; Krishnamurthy, 2005). This study explores the phenomenon of hybrid business models using an organizational practices perspective and tries to identify organizational practices and emergent strategies. The study identifies five management practices and three underlying strategies. To a certain extent, this study answers Fitzgerald's (2006) call for doing more *external research* in OSS.

5.1. Selective revealing

Henkel's (2006) idea of selective revealing is extended in the formulation of the first strategy. Henkel (2006) conceptualized selective revealing as a *conscious effort by organizations to toggle between restricting and publicizing their contributions to OSS products*. In this study, the actual manifestation of the strategy, in terms of practices, is explained. To the best of the authors' knowledge, this is one of the first empirically grounded studies of selective revealing as a strategic answer to the dilemma of merging proprietary and open source approaches.

Based on the combined practices under selective revealing, it can also be conceptualized as a framework for balancing the competing paradigms of OSS and proprietary software. The first aspect of the framework was called *functional generality*. If a particular software feature/function is usable by most of the intended user-base, then it can be released under an open source regime.

The second aspect is *functional differentiation*. The success of the OSS aspect of the hybrid business model depends on the success of its proprietary aspect and *vice versa*. Hence, none of the aspects are indispensible. As most hybrid business models would characterize these divergent approaches through multiple editions, there has to be enough differentiation between these editions to warrant the presence of two approaches. This inference is, in fact, a precursor to the first aspect of the framework, termed as functional generality.

The final aspect of the framework is *external knowledge*. There is no practice that explicitly refers to this dimension. Rather, it refers to the broad interest of the combined practices under selective revealing. Scholars have always stressed the importance of external knowledge (knowledge beyond the boundaries of the organization) in an organization for purposes like generating new product/service ideas, improving existing products/services, and marketing existing products/services (Cohen and Levinthal, 1990; Chesbrough, 2003). In case of a hybrid business model, the open source/community/free edition was used to leverage external knowledge.

5.2. Segmented meshing

The second strategy of segmented meshing focuses on the interface between the organization and the product community. Such interfaces are called boundary organizations, and they work toward managing the divergent interest of the parties involved (O'Mahony and Bechky, 2008). Segmented meshing can be treated as a strategic response used to manage boundary organizations. The idea behind segmented meshing comes from Fitzgerald's (2006) request of *changing the nature of OSS communities*. Existing literature on OSS looks at communities as a homogenous social structure (Shah, 2006; von Hippel and von Krogh, 2003). However, the adoption of hybrid business models for the software products that are being developed for enterprises involves a product community that is made up of multiple segments: developers, business users, partners, users of the community edition only, etc. To further complicate the matter, the boundaries of these segments are open. For example, a business partner for the community edition can become a business partner for the proprietary edition. Similarly, customers of the community edition may become customers of the proprietary edition. These open segments have a bearing on the nature of the participation, where each community is interested in a different aspect of the product. In such scenarios, sourcing and coordinating contributions from the community becomes important. Segmented meshing brings forth this divided view of community (hence the term *segmented*). *Meshing* refers to generating and coordinating contributions from different community segments. The mere identification of segments is not sufficient. Coordinating development across these segments is equally important. By coordinating the development process through a customized development platform, Openbravo ERP could not only provide a range of related products but could also bring all the contributors to a single platform for development. This naturally facilitated the coordination across the community segments (*meshing*).

5.3. Strategic licensing

Finally, the idea of strategic licensing builds on the literature of the choice of OSS licensing, where the existing OSS licensing literature is dominated by the restrictive and non-restrictive paradigms (Lerner and Tirole, 2002, 2005) and is governed by factors like the need to attract developers to the product (Sen et al., 2008) and the social influence on the licensors (Singh and Phelps, 2009). We proposed that in addition to attracting developers to the product, organizations might need to look at different aspects of their business models and then choose the license that fits well with the strongest aspect of their business model.

Licensing can also be seen as an incentive strategy for sourcing contributions from different segments of the OSS community. Developers' participatory motivations have been a topic of significant research attention in OSS (Lerner and Tirole, 2002; Hertel et al., 2003; Wu et al., 2007). However, motivation will be different for different segments in the community. Indeed, the community consists of different segments, and these segments are interested in the software product for different reasons. In the case of Openbravo ERP, licensing was devised to provide an incentive for its business partners to develop their own business models on top of Openbravo ERP (much like the platform strategy explained by West (2003)). Thus, organizations can choose a license for their software products that satisfies the needs of different segments of the community, which motivates the community members to participate in the development of the software.

5.4. Organizational practices for the hybrid business model: grounding in literature

Much like the research grounding for strategies, organizational practices can be viewed as logical extensions of findings reported in earlier studies. For example, Dahlander and Magnusson (2008) identified *ownership-driven licensing* as a tactic for aligning an organization and community objectives and avoiding any ownership-related conflict (p: 641). The practice of providing a license that allows for proprietary usage extends the scope of this tactic from *avoidance of conflict* to *satisfying profit motives* for some sections of the community (for example, Openbravo's partners developed their *own* proprietary extensions, allowing them to leverage Openbravo ERP as a platform for their businesses).

Similarly, the tactic of *feeding the non-strategic code to the community* is extended through the practice of providing free access to maintenance packs. Although maintenance packs were a strategy for generating revenue, Openbravo ERP provided maintenance packs as part of their community edition also. In other words, instead of *releasing non-strategic code*, practice four focuses on *product-critical code* that is essential for the functioning of the community edition of the software product. Similar comparisons can be drawn from West and O'Mahony (2008). West and O'Mahony (2008) propose allowing community members to create and own sub-projects as techniques for firms' participation is OSS. These techniques may be considered analogous to allowing community members to create proprietary extensions. Further the practice of choosing licensing from restrictive or non-restrictive paradigms is extended by including community segments as a determinant in licensing choice.

In summary, it can be argued that the practices identified in the case study relate to and extend the existing understanding of the strategy for firms participating in OSS.

5.5. Factors influencing adoption of hybrid business models

Software product organizations are adopting hybrid business model as a way of doing business (Campbell-Kelly and Garcia-Swartz, 2010). However, the adoption process can be tricky and as we found in case of Openbravo, it is influenced by number of factors.

Leadership remains a key influencer on emergence of operating mechanisms for OSS communities (Lerner and Tirole, 2002; O'Mahony and Ferraro, 2007). Some projects do not have a single leader but instead rely on community leadership. For example, Apache Foundation has a distributed form of leadership structure where a board takes policy decisions and each project under the foundation is empowered to make decisions on technical progress and development activities (Fielding, 1999). The structure also involves voting rights to members of Apache Group. In words of Roy Fielding, co-founder of Apache Foundation, "there was no Apache CEO, president, or manager to turn to for making decisions. Instead, we needed to determine group consensus, without using synchronous communication" (Fielding, 1999). On the other hand, Linus Torvalds; founder of Linux operating system, holds the final decision making authority. He relies on a core team of project managers who are responsible for advising him on various project-related issues. Termed as benevolent dictator leadership (Raymond, 2001), this approach differs from that of Apache foundation. Under this approach, leader keeps the complete authority yet fosters collaborative operating mechanisms. Clearly, open source does not mandate a single form of leadership.

In the case of Openbravo, the key challenge was balancing the open source and proprietary facets of the business model. Practices I and IV specifically addressed the challenge. The community leader with 'right pedigree' positively influenced attempts to tackle this challenge. A community manager was associated to open source projects.

> *What we found that was a bit challenging over the years was to maintain the community and to balance the commercial efforts with the community efforts and trying to figure out a fair business model that would allow us to generate revenues but at the same time would not have been perceived by our community as being exploitation of open source. That was a quite a bit of a challenge for us. We have been very lucky we had at that time a community manager Jordi Mas that had a lot of experience in open source, was one of the leading open source people in Spain, had published books, participated in localization of Open Office. He helped manage the community and put in place a very community friendly culture within the company that allowed us to really maintain a very healthy community*[Chief Technology Officer, Openbravo ERP]

Conversely, if community perceived the leader to be deviant from open source philosophy, then it also generated negative reactions; indicating that community members considered leader's past adherence to open source philosophy as an important attribute. It can therefore be concluded that previous association of community leader with open source had a positive influence on execution of Openbravo's hybrid business model.

> *My background is with [Proprietary Software Product Company X] and at a point we have another person from [Proprietary Software Product Company Y] so people took that as... that we didn't have the right pedigree in the open source... to involve in an Open source project... that company was hiring too many people with the proprietary background.*[Chief Technology Officer, Openbravo ERP]

The other issue referred to creation of knowledge sharing practices. Open source approach differs from proprietary one in its degree of collaborative efforts (Lerner and Tirole, 2002). The effort is driven by an environment where community members are encouraged to share knowledge amongst each other (Kuk, 2006). As the knowledge often relates to the software itself, its sharing has a positive influence on the development efforts. However, there is no clear answer as to why people participate in knowledge sharing as both intrinsic and extrinsic motivations are at play (Wasko and Faraj, 2005; Lakhani and Wolf, 2005). The question of motivations for participation is one of most researched avenues in OSS domain (von Krogh and Spaeth, 2007). It is often observed that all members in OSS community do not equally involve themselves in sharing knowledge in spite of using the software (Lakhani and Wolf, 2005).

In the case of hybrid business model, knowledge sharing becomes even more complex challenge, mainly because sharing is supposed to happen between actors sitting on either side organizational boundaries. The scenario has been examined through various theoretical lenses such as absorptive capacity (Dahlander and Magnusson, 2008) and open innovation (West and Gallagher, 2006). However, there is a general consensus that institutionalizing knowledge sharing mechanisms requires organizational intervention.

Openbravo ERP was no exception to this. Knowledge sharing was facilitated through both the technological and process-oriented interventions. Development infrastructure of Openbravo was publicly accessible where users could review the progress and share their feedback. Development process artifacts such as product roadmap were articulated through consensus building across community members supported by a formal process of polling for inclusion/exclusion of functionality in the ongoing release plan. These activities often included external as well as internal actors.

> *Before publishing this roadmap we have the open forums which we have the discussion of product strategy and the bottlenecks in the system and the things we envisioned important to product and our community. There is a poll and everybody in the community can participate in a poll and there is a rating after the poll. [...] Once we have the functionality we blog about it, we dedicate separate forums for this functionality and any person can try this functionality, provide us feedback, based on this feedback we can tune this functionality*[Product development manager, Openbravo ERP]

Also, there were public discussion platforms where users could discuss various issues related to product. These discussions were not capsuled but were fed back to the product development and release management. These activities were not peripheral to Openbravo's business model but were integrated into its operationalization.

> *Our public Wiki where project has a section on the Wiki where there are open specifications, open technical designs and everything is openly discussed and openly documented so that the people in the community can monitor the progress [...] It is the loop that feeds itself because once they test, they report issues, raise new ideas, new feature requests and that comes into the scheduling of the next release.*[Chief Technology Officer, Openbravo ERP]

Lastly, as the hybrid business model is a configuration of organizational practices that are transferred from OSS and proprietary approaches, its operationalization would also be subject to factors that influence such a transfer. Kostova and Roth (2002) highlighted two factors that affect the inter-organizational transfer of practices: *the institutional profile of the host country for the subsidiary* and *the relational congruence between the subsidiary and parent company*. Kostova & Roth's (2002) work discussed subsidiary and parent companies. However, their factors seem relevant in hybrid business models. Let's examine the possible role of the institutional profile of a country. Kostova and Roth (2002) define an institutional profile as *the issue-specific set of regulatory, cognitive, and normative institutions in a given country*. When extended to the OSS phenomenon, these two factors can be identified as attributes of the institutional profile: *the nature of the local software industry* and *government policies toward OSS*.

These factors affect the adoption of OSS in a given country (Noonan et al., 2008; Comino and Manenti, 2005). Because the presence of OSS is a critical aspect for any software product organization wanting to adopt a hybrid business model (such hybrid business models treat the community as a sourcing partners (Ågerfalk and Fitzgerald, 2008). Government policies and the nature of the local software industry affect the adoption of OSS in a given geographical area and, in turn, influence the transfer of practices from OSS to a software product organization in a given geographical setting.

Thus, a software product organization has to manage its relational congruence with OSS. The question, however, is *how such contexts can be assessed*. From the case of Openbravo ERP, at least two types of relational congruence assessments can occur: *people* and *tools*. People-oriented relational congruence is captured in earlier explanation of leadership styles and leaders' open source pedigree.

Tools-oriented congruence reflected the technological imperative. Openbravo ERP was developed using open source technologies such as MySQL, Java and Apache. This was a precursor to the adoption of organizational practices from OSS. As is clear from the following excerpts, the transition to a hybrid business model was a natural course of action based on the open source nature of the underlying technologies.

> *To begin with, they decided that they wanted to license it under an Open source license primarily because they wanted to leverage open source technology components. Because they were embedding technology components, they decided that the overall project was to be licensed under Open source license*[Chief Technology Officer, Openbravo ERP].

A resultant word of caution for a software product organization: operationalization of hybrid business model is influenced by numerous factors. In case of Openbravo, three factors were identified: *relational congruence* (use of open source technologies, leadership's open source pedigree), *institutional profile of the country* (government policies, nature of local software industry), and *creation of knowledge sharing practices*.

5.6. Sourcing contributions from community

Hybrid business models thrive on contributions from product communities (*open sourcing*) (Ågerfalk and Fitzgerald, 2008). However, it is important for an organization to understand community contributions. There have been several attempts to theorize the idea of contributions from stakeholders outside organizational boundaries. Some prominent contributions include: *external knowledge, absorptive capacity* (Cohen and Levinthal, 1990), *user innovation* (von Hippel, 2005), and *open innovation* (Chesbrough, 2006). However, these do not relate contributions to different dimensions of the business model. Openbravo's case study revealed that community contributions relate to the development and distribution dimensions of the *hybrid business model*. In the development phase, the community contributes through the *identification of issues* and *suggestion of new features*. This contribution is usually captured in the product roadmap, which is an important tool in product development. In the distribution phase, the community participates by *creating awareness* and the *redistribution* of the software. Indeed, community contribution cannot be viewed outside the context of the underlying business model. With these inferences, community contributions in hybrid business models are defined as *value additions achieved across the dimensions of the underlying business model through the participation of stakeholders outside the boundaries of an organization*.

5.7. Software services and hybrid business model

The proprietary software business thrives on services bundled with software products and licenses (Popp and Meyer, 2010; Light and Sawyer, 2007; Qu, 2008). In fact, service quality is considered one of the determinants of the adoptive success of an information system (DeLone and McLean, 1992, 2003). However, open source software does not come with any service guarantees. The very licensing of the software eradicates any responsibility of service provision and enables it to be distributed as is (Lerner and Tirole, 2005). Users of OSS, therefore, tend to rely on informal support garnered from other community members, and the availability and quality of such support is a determinant of OSS success (Lee et al., 2009).

The hybrid business model exhibits impressions of both approaches. Therefore, it is worthwhile examining if product hybridization (achieved usually through software licensing) also translates into service hybridization. Studies conclude that almost all variants of the hybrid business model depend on offering services where the software itself is distributed under open source (Krishnamurthy, 2005; Fitzgerald, 2006).

In case of Openbravo ERP, some evidence was found indicating the presence of both formal and community-driven delivery of services. The community edition of Openbravo ERP was devoid of any formal service contracts. One segment of Openbravo's partners was implementing the open source edition for their clients without any commercial relationship with Openbravo ERP. These partners drew service support from both community members and their own, in-house expertise. On the other hand, subscriptions of professional edition were bundled with services, such as training and certification. Delivery of these services was governed by formal contractual agreements. Partner organizations with subscriptions for the professional edition received support services directly from Openbravo.

> *We have people that implement the products…the projects at their customers. So they are system integrators and value-added resellers that provide services around our product, but they choose to work in the pure community vault without any commercial relationship with us […]our business partners that are people that provide services but they choose to do it on professional edition and having commercial relationship with us […]We have community service providers that have a platform that they can use and leverage for free to provide services and generate revenues for themselves*[Chief Technology Officer, Openbravo ERP].

Such duality of service delivery indicates that both formal service bundling and community service offerings were made available from Openbravo ERP. This assumption furthers the argument that services are important for hybrid business models. It can also be suggested that hybrid business models achieve product as well as service hybridization with each form of service delivery (formal versus community) adding value to different editions of the software product.

5.8. Strategic purity versus hybridization: a larger debate

The inherent tension of combining mutually exclusive approaches (open source and proprietary) is an instance of a larger debate of *strategic purity*, which is defined as an organization's adherence to a single strategic direction using generic strategies (Thornhill and White, 2007). The central tenet is that general organizational strategies are distinct from each other, and when an organization tries to adopt multiple and, often conflicting strategies, it leads to an organization stuck in the middle, where none of the strategies is likely to succeed. This situation is often perceived as undesirable for an organization, largely by the Porter's school of strategic thinking (Thornhill and White, 2007). However, an organization is forced to forego the opportunities presented by other, divergent strategic directions. For example, an organization looking to adopt cost leadership may not be able to pursue a more resource-intensive differentiation strategy (as it would conflict with a cost reduction strategy).

Software product organizations seek to overcome this limitation. In proprietary businesses, the idea is to maintain the *product differentiation* through vendor lock-in and strong appropriation regimes while the increasing costs of development require organizations to *reduce costs*. Hybridization is increasingly becoming the accepted answer to fight this dichotomy. This is apparent in the increasing acceptance of hybrid business models and strategies that compel organizations to adopt multiple, and often conflicting, directions (Campbell-Kelly and Garcia-Swartz, 2010). Open source and proprietary software approaches have operationally and philosophically divergent roots, and organizations find it challenging to merge the two. However, organizational practice, with its holistic conceptualization, provides a vehicle for software product organizations to undertake a hybridization effort. In addition, these practices provide organizational scholars with a conceptual perspective for investigating such instances.

5.9. Limitations and future research

This study is not without its limitations. First, the methodology of a single case study can pose challenges to the generalization of the study's findings. We caution the reader to scrutinize the findings within the context of the case. This is rather a methodological limitation. As the phenomenon is studied in its natural context, researcher may not have control over the settings or the actors involved (Yin, 2009). Naturally the influence of context on study's findings cannot be negated. Secondly, strong bias toward the organizational perspective concerning the product has led to the omission of the perspectives of other stakeholders in the Openbravo environment (for example, partners, clients, and competitors). Better triangulation can be achieved by incorporating evidence from these sources. Thirdly, the study does not attempt to develop any particular theory, as outlined by Gregor (2006), but has the more humble objective of making an observed anomaly clearer through the understanding of management practices and strategies.

We suggest several research avenues. First, it would be worth examining *if some of the reported practices are best practices in the software product industry*. This can be done through the use of multiple case studies selected through replication logic (Eisenhardt, 1989), where multiple OSS products with hybrid business models may be studied and the application of techniques, such as pattern matching, would reveal the presence of these practices across products. The outcome of such exercises could be a theory approximation (Weick, 1995). Thus, there are considerable opportunities for developing new theories in this domain. One may also undertake a more general study of other industries, such as pharmaceuticals and automobiles, where organizations are employing hybrid business models.

Secondly, it might be worthwhile to examine if any of the reported practices indeed affect the development and/or usage of OSS products with hybrid business models. The success of OSS products has been studied to some extent (Comino et al., 2007; Subramanian et al., 2009), but there is little exclusive focus on OSS products with hybrid business models. One possible approach includes examining the development and usage of a large sample-set of OSS products, and through a typical variance research design, it can be determined if the presence of these practices has any significant relationship with the success of the product.

Thirdly, it might be worthwhile to contrast the idea of organizational practices against other theoretical perspectives in strategic management, such as *capabilities*. Capabilities refer to vehicles that are used to manifest the strategy formulated. For example, Peppard and Ward (2004) define information system capability as "the ability to translate the business strategy into long term information architectures, technology infrastructure and resourcing plans that enable the implementation of the strategy" (p: 177). In examining open innovation archetypes, Gassmann and Enkel (2004) outlined three capabilities: *relational*, *absorptive*, and *multiplicative*. A particular capability was critical depending on the open innovation approach adopted by the organization. It may be useful to know *if any of these capabilities are necessary for the adoption of the identified organizational practices or vice versa*. Such an examination can highlight the interplay between different paradigms within strategic management literature. Some efforts have already been made in this direction. For example, Orlikowski (2002) demonstrates that 'distributed organizing' as a capability is grounded in everyday practices of 'knowing' across distributed teams.

A longitudinal study can examine the life-cycle of these practices. Surely, with a heavy contextual dependence, practices undergo change. It is worthwhile examining how these practices change and how these changes link back to the hybrid business model. Some characteristic issues are as follows: *are hybridized practices stable? If no, then are their changes reflected in the underlying hybrid business model*?

Studies could also examine different factors that influence adoption and operationalization of hybrid business models. Present study does reveal *perception about leaders' open source pedigree* as a dimension of relational congruence. However, if one is to view developing software product in collaborative environment as a knowledge creation then leadership has a wider role; influencing "local knowledge creation, resource provision and providing strategic direction" (von Krogh et al., 2012). Therefore it is worth investigating *leadership traits that might influence tactical as well as strategic issues in knowledge creation in a hybridized context*.

There are other research directions as well. Hybrid business model may manifest in different forms (Hemphill, 2006; Krishnamurthy, 2005). In each manifestation, the knowledge creation may be organized separately or the nature of organization's participation would change (Dahlander and Magnusson, 2005). In such scenarios, *would the influence of leadership traits differ across manifestations of hybrid business model*? Even more fundamentally, *when organizations adopt a particular form of hybrid business model, how do they decide to formulate a particular leadership structure*? A longitudinal study on leadership styles and structures under different manifestations of hybrid business models would be suitable. It is observed that in communities involved in producing collective good (such as OSS), understanding of leadership evolves (O'Mahony and Ferraro, 2007). The change however is captured in resultant governance mechanisms. One direction can therefore be to study *analyze if knowledge creation mechanisms are influenced by changing autocratic and democratic governance mechanisms*.

Future research can also delve deeper into the actual knowledge creation and sharing mechanisms. In case of Openbravo, such mechanisms were integral to its hybrid business model. However, a proprietary organization transiting to a hybrid business model may have cultural ‘stickiness’ in adopting new knowledge sharing mechanisms (Bonaccorsi et al., 2006). It may be useful to study *evolution of ‘organizational practices’ for knowledge creation and sharing; especially in context of proprietary software product organization transiting to hybrid business model*. Naturally, a process study of an organization’s transition into hybrid business model with a specific focus on institutionalizing knowledge sharing practices would be interesting.

Knowledge sharing also posits an intellectual property risk. It may be through sharing of ‘too much key’ knowledge or improper handling of the organization’s IP by its partners (Trkman and Desouza, 2012). This is particularly of importance in OSS where the participating community members may remain unknown even to the software product organization (Ågerfalk and Fitzgerald, 2008). To some extent, the selective revealing approach would guard an organization against the sharing of critical knowledge. Yet further studies are required to ascertain how knowledge sharing mechanisms mitigate risk of *negative knowledge spillovers*.

At a more fundamental level, knowledge itself may not shareable; especially in light of organizational practices. Knowledge embedded in such practices may be context sensitive (Marabelli and Newell, 2012). Consequently, all knowledge coming from external resources like users and business partners may not be relevant. Future studies may examine *how organizations filter knowledge in-flows*. Finally, an organization’s demographics, such as its age and structure, may also influence adoption of the hybrid business model (Bonaccorsi et al., 2006). Studies may therefore examine *whether these factors have the same influence across all forms of hybrid business models.*

References

Andersen-Gott, M., Ghinea, G., Bygstad, B., 2012. Why do commercial companies contribute to open source software? International Journal of Information Management 32 (2), 106–117.

Ågerfalk, P.J., Fitzgerald, B., 2008. Outsourcing to an unknown workforce. Exploring opensourcing as a global sourcing strategy. MIS Quarterly 32 (2), 385–409.

Bitzer, J., 2004. Commercial versus open source software: the role of product heterogeneity in competition. Economic Systems 28 (4), 369–381.

Bonaccorsi, A., Giannangeli, S., Rossi, C., 2006. Entry strategies under competing standards: hybrid business models in the open source software industry. Management Science 52 (7), 1085–1098.

Bonaccorsi, A., Rossi, C., 2003. Why open source software can succeed. Research Policy 32 (7), 1243–1258.

Campbell-Kelly, Martin., Garcia-Swartz, D.D., 2010. The move to the middle: convergence of the open-source and proprietary software industries. International Journal of the Economics of Business 17 (2), 223–252.

Casadesus-Masanell, R., Ghemawat, P., 2006. Dynamic mixed duopoly: a model motivated by linux vs. windows. Management Science 52 (7), 1072–1081.

Chesbrough, H., 2003. Open Innovation: New Imperative for Creating and Profiting from Technology. Harvard Business School Publishing, Boston.

Chesbrough, H., 2006. Open Business Models: How to Thrive in the new Innovation Landscape. Harvard Business School Press, Boston.

Chesbrough, H., 2010. Business model innovation. Long Range Planning 43 (2–3), 354–363.

Cohen, W.M., Levinthal, D.A., 1990. Absorptive capacity: a new perspective on learning and innovation. Administrative Science Quarterly 35, 128–152.

Comino, S., Manenti, F.M., 2005. Government policies supporting open source software for the mass market. Review of Industrial Organization 26 (2), 217–240.

Comino, S., Manenti, F.M., Parisi, M.L., 2007. From planning to mature: on the success of open source products. Research Policy 36 (10), 1575–1586.

Comino, S., Manenti, F.M., 2011. Dual licensing in open source software markets. Information Economics and Policy 23 (3–4), 234–242.

Dahlander, L., Magnusson, M.G., 2005. Relationships between open source software companies and communities: observations from Nordic firms. Research Policy 34 (4), 481–493.

Dahlander, L., Wallin, M.W., 2006. A man on the inside: unlocking communities as complementary assets. Research Policy 35 (8), 1243–1259.

Dahlander, L., Magnusson, M., 2008. How do firms make use of open source communities? Long Range Planning 41 (6), 629–649.

DeLone, W.H., McLean, E.R., 1992. Information systems success: the quest for the dependent variable. Information Systems Research 3 (1), 60–95.

DeLone, W.H., McLean, E.R., 2003. The DeLone and McLean model of information systems success: a 10-year update. Journal of Management Information Systems 19 (4), 9–30.

Dubé, L., Pare, G., 2003. Rigor in information systems positivist case research: current practices, trends, and recommendations. MIS Quarterly 27 (4), 597–635.

Eisenhardt, K.M., 1989. Building theories from case study research. Academy of Management Review 32 (4), 532–550.

Eisenhardt, K.M., Bourgeois III, L.J., 1988. Politics of strategic decision making in high-velocity environments: toward a midrange theory. Academy of Management Journal 31 (4), 737–770.

Feller, J., Finnegan, P., Hayes, J., 2006. Open source networks: an exploration of business model and agility issues. In: European Conference on Information Systems Proceedings. Paper 95. <http://aisel.aisnet.org/ecis2006/95>.

Fielding, R., 1999. Shared leadership in the apache project. Communications of the ACM 42 (4), 42–43.

Fitzgerald, B., 2006. The transformation of open source software. MIS Quarterly 30 (2), 587–598.

Gassmann, O., Enkel, E., 2004. Towards a theory of open innovation: three core process archetypes. In: Proceedings of the R & D Management Conference. Lisbon, Portugal.

Glaser, B.G., Strauss, A.L., 1967. The Discovery of Grounded Theory: Strategies for Qualitative Research. Aldine Transaction, CA.

Glaser, B.G., 1992. Basics of Grounded Theory Analysis: Emergence vs. Forcing. Sociology Press.

Grand, S., von Krogh, G., Leonard, D., Swap, W., 2004. Resource allocation beyond firm boundaries: a multi-level model for open source innovation. Long Range Planning 37, 591–610.

Gregor, S., 2006. The nature of theory in information systems. MIS Quarterly 30 (3), 611–642.

Grewal, R., Lilien, G.L., Mallapragad, G., 2006. Location, location, location: how network embeddedness affects product success in open source systems. Management Science 52 (7), 1043–1056.

Harison, E., Koski, H., 2010. Applying open innovation in business strategies: evidence from finnish software firms. Research Policy 39 (3), 351–359.

Haruvy, E., Sethi, S.P., Zhou, J., 2008. Open source development with a commercial complementary product or service. Production and Operations Management 17 (1), 29–43.

Hauge, U., Ayala, C., Conradi, R., 2010. Adoption of open source software in software-intensive organizations: a systematic literature review. Information and Software Technology 52 (11), 1133–1154.

Hedman, J., Kalling, T., 2003. The business model concept: theoretical underpinnings and empirical illustrations. European Journal of Information Systems 12, 49–59.

Hemphill, T.A., 2006. A taxonomy of closed and open source software industry business models. International Journal of Technology and Innovation Management 3 (1), 61–82.

Henkel, J., 2006. Selective revealing in open innovation processes: the case of embedded Linux. Research Policy 35 (7), 953–969.

Hertel, G., Niedner, S., Herrmann, S., 2003. Motivation of software developers in open source products: an internet-based survey of contributors to the Linux Kernel. Research Policy 32, 1159–1177.

Kemp, R., 2009. Current developments in open source software. Computer Law Security Review 25 (6), 569–582.

Kilamo, T., Hammouda, I., Mikkonen, T., Aaltonen, T., 2011. From proprietary to open source-growing an open source ecosystem. Journal of Systems and Software.

Kim, B., Chen, P., Mukhopadhyay, T., 2006. Pricing open source software. In: International Conference of Information Systems. Paper 24. <http://aisel.aisnet.org/icis2006/24>.

Klein, H.K., Myers, M.D., 1999. A set of principles for conducting and evaluating interpretive field studies in information systems. MIS Quarterly 23 (1), 67–94.

Kling, M., 1962. Cuba: a case study of a successful attempt to seize political power by the application of unconventional warfare. The ANNALS of the American Academy of Political and Social Science 341 (1), 42–52.

Kostova, T., 1999. Transnational transfer of strategic organizational practices: a contextual perspective. The Academy of Management Review 24 (2), 308–324.

Kostova, T., Roth, K., 2002. Adoption of an organizational practice by subsidiaries of multinational corporations: institutional and relational effects. The Academy of Management Journal 45 (1), 215–233.

Krishnamurthy, S., 2005. An analysis of open source business models. In: Feller, Joseph, Fitzgerald, Brian, Hissam, Scott, Lakhani, Karim (Eds.), Making Sense of the Bazaar: Perspectives on Open Source and Free Software. MIT Press.

Kuk, G., 2006. Strategic interaction and knowledge sharing in the kde developer mailing list. Management Science 52 (7), 1031–1042.

Laine, M.O.J., Ronkko, M., Valtakoski, A., 2009. Software firm business models with virtual communities. In: AMCIS 2009 Proceedings. Paper 661, CA. <http://aisel.aisnet.org/amcis2009/661>

Lakhani, K.R., Hippel, E.v., 2003. How open source software works: "Free" user-to-user assistance. Research Policy 32 (6), 923–943.

Lakhani, K.R., Wolf, R., 2005. Why hackers do what they do: understanding motivation and effort in free/open source software products. In: Feller, J., Fitzgerald, B., Hissam, S., Lakhani, K. (Eds.), Perspectives on Free and Open Source Software. MIT Press, Cambridge, Mass, pp. 3–22.

Lakka, S., Varoutas, D., Martakos, D., 2009. Impact of oss on software markets – an evaluation. In: MCIS 2009 Proceedings. Paper 56, Athens. <http://aisel.aisnet.org/mcis2009/56>

Lee, A., 1989. A scientific methodology for MIS case studies. MIS Quarterly 13 (1), 33–50.

Lee, S.Y.T., Kim, H.-W., Gupta, S., 2009. Measuring open source software success. Omega 37 (2), 426–438.

Lerner, J., Tirole, J., 2002. Some simple economics of open source. Journal of Industrial Economics 50 (2), 197–234.

Lerner, J., Tirole, J., 2005. The scope of open source licensing. Journal of Law, Economics, and Organization 21 (1), 20–56.

Light, B., Sawyer, S., 2007. Locating packaged software in information systems research. European Journal of Information Systems 16, 527–530.

Mahadevan, B., 2000. Business models for internet based e-commerce. An anatomy. California Management Review 42 (4), 55–69.

Marabelli, M., Newell, S., 2012. Knowledge risks in organizational networks: the practice perspective. Journal of Strategic Information Systems 21 (1), 18–30.

Markus, L.M., Pfeffer, J., 1983. Power and the design and implementation of accounting and control systems. Accounting, Organizations & Society 8 (2/3), 205–218.

Mintzberg, H., 1978. Patterns in strategy formation. Management Science 24 (9), 934–948.

Morgan, L., Feller, J., Finnegan, P., 2011. Exploring Inner source as a form of inter-organizational open innovation (2011). In: ECIS 2011 Proceedings. Paper 151, Helsinki. <http://aisel.aisnet.org/ecis2011/151>

Morris, M., Schindehutte, M., Allen, J., 2005. The entrepreneur's business model: toward a unified perspective. Journal of Business Research 58 (6), 726–735.

Noonan, D. S., Baker, P. M., & Moon, N. W. (2008). Open Source Software Potential Index: Developmental Considerations. Red Hat: <http://www.redhat.com/f/pdf/OSSI_Research.pdf> (retrieved 8.8.11).

O'Mahony, S., Ferraro, F., 2007. The emergence of governance in an open source community. Academy of Management Journal 50 (5), 1079–1106.

O'Mahony, S., Bechky, B.A., 2008. Boundary organizations: enabling collaboration among unexpected allies. Administrative Science Quarterly 53 (3), 422–459.

Orlikowski, W.J., 2002. Knowing in practice. Enacting a collective capability in distributed organization. Organization Science 13 (3), 249–273.

Peppard, J., Ward, J., 2004. Beyond strategic information systems: towards an IS capability. Journal of Strategic Information Systems 13 (2), 167–194.

Pettigrew, S.F., 2000. Ethnography and grounded theory: a happy marriage? Advances in Consumer Research 27, 256–260.

Popp, K.M., Meyer, R., 2010. Profit from Software Ecosystems: Business Models, Ecosystems and Partnerships in the Software Industry. BOD, Norderstedt, Germany (ISBN:3839169836).

Pykäläinen, T., 2008. Model for profiting from software innovations in the new era of computing. Technovation 27, 179–193.

Qu, Z. (2008) Software Design Strategies in Markets with Open Source Competitors. In: ICIS 2008 Proceedings. Paper 42, Paris. <http://aisel.aisnet.org/icis2008/42>.

Rajala, R., Rossi, M., Tuunainen, V., 2003. A framework for analyzing software business models. In: Proceedings of the ECIS 2003, conference on information systems – new paradigms in organizations, markets and society, June 18–22, 2003, Naples, Italy.

Rajala, R., Nissila, J., Westerlund, M., 2006. Determinants of OSS revenue model choices. In: European Conference of Information Systems Proceedings. Paper 160. <http://aisel.aisnet.org/ecis2006/160>.

Raymond, E., 2001. The Cathedral and the Bazaar. O'Reilly, Sebastopol, CA.

Riehle, D., 2009. The commercial open source business model. In: AMCIS 2009 Proceedings. Paper 104, CA. <http://aisel.aisnet.org/amcis2009/104>.

Riehle, D., 2012. The single-vendor commercial open course business model. Information Systems and E-Business Management 10 (1), 5–17.

Rolandsson, B., Bergquist, M., Ljungberg, J., 2011. Open source in the firm: opening up professional practices of software development. Research Policy 40 (4), 576–587.

Sen, R., Subramaniam, C., Nelson, M.L., 2008. Determinants of the choice of open source software license. Journal of Management Information Systems 25 (3), 207–239.

Shah, S.K., 2006. Motivation, governance, and the viability of hybrid forms in open source software development. Management Science 52 (7), 1000–1014.

Shaikh, M., Cornford, T., 2009. Innovating with open sourcing: governance concerns for managers. In: AMCIS 2009 Proceedings. Paper 308, CA. <http://aisel.aisnet.org/amcis2009/308>.

Sharma, S., Sugumaran, V., Rajagopalan, B., 2002. A framework for creating hybrid-open source software communities. Information Systems Journal 12, 7–25.

Singh, P.V., Phelps, C.C., 2009. Determinants of open source software license choice: a social influence perspective. Available at SSRN: <http://ssrn.com/abstract=1436153>.

Sohn, S.Y., Mol, M.S., 2008. A strategic analysis for successful open source software utilization based on a structural equation model. Journal of Systems and Software 81 (6), 1014–1024.

Stam, W., 2009. When does community participation enhance the performance of open source software companies? Research Policy 38, 1288–1299.

Stewart, K.J., Ammeter, A.P., Maruping, L.M., 2006. Impacts of license choice and organizational sponsorship on user interest and development activity in open source software products. Information Systems Research 17 (2), 126–144.

Strauss, A.C., Corbin, J.M., 1990. Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory. SAGE, CA.

Subramanian, C., Sen, R., Nelson, M.L., 2009. Determinants of open source software product success: a longitudinal study. Decision Support Systems 46 (2), 576–585.

Schuler, R.S., Jackson, S.E., 1987. Linking competitive strategies with human resource management practices. Academy of Management Executive 1 (3), 207–219.

Thornhill, S., White, R.E., 2007. Strategic purity: a multi-industry evaluation of pure vs. hybrid business strategies. Strategic Management Journal 28 (5), 553–561.

Trkman, P., Desouza, K.C., 2012. Knowledge risks in organizational networks: an exploratory framework. Journal of Strategic Information Systems 21 (1), 1–17.

von Krogh, G., Nonaka, I., Rechsteiner, L., 2012. Leadership in organizational knowledge creation: a review and framework. Journal of Management Studies 49, 240–277.

von Hippel, E., von Krogh, G., 2003. Open source software and the "Private-Collective" innovation model: issues for organization science. Organization Science 14 (3), 209–223.

von Krogh, G., Spaeth, S., 2007. The open source software phenomenon: characteristics that promote research. Journal of Strategic Information Systems 16 (3), 236–253.

von Hippel, E., 2005. Democratizing Innovation. The MIT Press, Cambridge, MA.

Vitari, C., Ravarini, A., 2009. A longitudinal analysis of trajectory changes in the software industry: the case of the content management application segment. European Journal of Information Systems 18 (3), 249–263.

Walsham, G., 1995. Interpretive case studies in IS research: nature and method. European Journal of Information Systems 4 (2), 74–81.

Wasko, M.M., Faraj, S., 2005. Why should I share? Examining social capital and knowledge contribution in electronic networks of practice. MIS Quarterly 29 (1), 35–57.

Weick, K.E., 1995. What theory is not, theorizing is. Administrative Science Quarterly 40 (3), 385–390.

West, J., 2003. How open is open enough? Melding proprietary and open source platform strategies. Research Policy 32 (7), 1259–1285.

Whittington, R., 2006. Completing the practice turn in strategy research. Organization Studies 27 (5), 613–634.

Wu, C., Gerlach, J.H., Young, C.E., 2007. An empirical analysis of open source software developers' motivations and continuance intentions. Information & Management 44, 253–262.

West, J., Gallagher, S., 2006. Challenges of open innovation: the paradox of firm investment in open-source software. R&D Management 36 (3), 319–331.

West, J., O'Mahony, S., 2008. The role of participation architecture in growing sponsored open source communities. Industry & Innovation 15 (2), 145–168.

Yin, R., 2009. Case Study Research: Design and Methods, fourth ed. SAGE Publication.