Association for Information Systems
AIS Electronic Library (AISeL)

8-5-2011

Organizational Participation in Open Communities: Conceptual Framing and Early Findings

Matt Germonprez
University of Wisconsin - Eau Claire, germonr@uwec.edu

Julie Kendall
Rutgers University, kendallj@crab.rutgers.edu

Ken Kendall
Rutgers University, kendallk@camden.rutgers.edu

Brian Warner
Linux Foundation, brian.warner@linuxfoundation.org

Lars Mathiassen
Georgia State University, lars.mathiassen@eci.gsu.edu

Follow this and additional works at: http://aisel.aisnet.org/amcis2011_submissions

Organizational Participation in Open Communities: Conceptual Framing and Early Findings

Matt Germonprez
University of Wisconsin - Eau Claire
germonr@uwec.edu

Julie Kendall
Rutgers University
kendallj@crab.rutgers.edu

Ken Kendall
Rutgers University
kendallk@camden.rutgers.edu

Brian Warner
Linux Foundation
brian.warner@linuxfoundation.org

Lars Mathiassen
Georgia State University
Lars.Mathiassen@eci.gsu.edu

ABSTRACT

As design and development evolves within open communities, new affordances present new possibilities and organizations must balance 'contributions to' and 'differentiation from' the open community for reasons of cost, resource management, and time to market. Organizational participation in open communities is timely in light of recent analyses by the Linux Foundation indicating that 75% of kernel contributions are by paid developers. In this proposal, we build on principles of public sharing and collaboration using the Linux open-source community as our basis for understanding open communities (Fitzgerald, 2006). The focus of this project is why organizations participate with open communities and how they participate with open communities. We apply action research as a methodological approach within which a qualitative field study will be conducted (Chiasson et al., 2009). Action research supports our dual goal of developing a solution to a practical problem which is of value to the people with whom we are working, while at the same time developing theoretical knowledge of value to an academic community involved in research and pedagogy (Mathiassen et al., 2009). We found organizational participation to be primarily derived from the leveraged support, contribution to, and differentiation from open communities.

INTRODUCTION

Open communities create complicated issues for organizations and researchers because they are more complex than simple technology-enabled groups; they are a mix of power and knowledge, liberty and enlightenment, progress and intervention (Kelty, 2009). Open communities adapt to dynamically changing situations, accommodate altered plans, and engage in non-typical, cooperative work in which there is an emergence, never a guarantee, of stability (Germonprez et al., 2007). we define an open innovation community as a collection of varied organizational members where organizations approach the community as a strategic motivation and seek to leverage the community for organizational benefit (West and Lakhani, 2008). The technology used in an open community is only one-half of the design process. The other, equally important half includes the reflective, active, and interactive practices that community members engage in. Within open communities, members create new structural couplings in alignment with their domain of action in coordinating efforts, eliminating redundancy, pursuing options, and sequencing activities (Germonprez et al., 2007). As participation evolves within open communities, new affordances present new possibilities and organizations must balance 'contributions to' and 'differentiation from' the open community for reasons of cost, resource management, and time to market.

In this paper, we build on principles of public sharing and collaboration using the Linux *open-source* community as our basis for understanding (Fitzgerald, 2006). The Linux Foundation estimated the value of Linux to be $10.8 billion in 2008 with the number of participants surpassing 3,500, illustrating that the Linux open community is both viable and important for study. While open-source is strictly a licensing distinction that does not necessarily define an open community, it is often used to describe permissively-licensed software developed by an open community (Fitzgerald, 2006; Ågerfalk et al., 2009). The focus of this study is not on Linux *per se*; rather it is on open community participation associated with the design and development of Linux.

We explore three primary features of participation with the Linux open community: leverage, contribution, and differentiation. Leverage constitutes the power of open community to benefit all participants. How does the community provide advantages for participants? Contributions constitute the degree to which community participants play a part in the open community. Do they actively engage in the design and development of Linux? Differentiation constitutes the degree to which participants follow the primary release of the artifact. Do they use Linux as publicly released or do they differentiate it for internal reasons? Practice and research are beginning to address these issues through frameworks, theories, methods, and contributions of open communities (von Hippel and von Krogh, 2003; Henkel, 2006; Ågerfalk et al., 2009). To extend research on open community participation, we used relevant literature and the Linux open community to create frameworks relevant to both our problem and research domains. To begin, we assess the interaction between the open community and the corporate organization and consider what characteristics foster a relationship. In doing this, we address why organizations participate with the Linux open community, leading us to our first of two research questions:

Why do organizations participate with the Linux open community?

Determining the *why* of participation leads to the second research question of *how* organizations participate. In investigating the *how,* we specifically investigate organizational issues related to participation with the Linux open community (Wenger, 1999). We expect these patterns to be variable as members balance commercial and community responsibilities and knowledge sharing at the interface between the participating organization and the Linux open community (Henkel, 2006). We investigate how organizational decisions both determine and are determined by participation with the Linux open community. This understanding can act as a roadmap for both organizations considering open communities as a viable systems development option and researchers seeking to expand organizational theory around open community participation. Similar to our investigation of why organizations participate, we investigate how organizations participate with the community, leading to our second research question:

How do organizations participate with the Linux open community?

Across both research questions, we address the growing research streams associated with open communities. As open communities represent an emerging and fast-growing consideration for organizations, it is incumbent on practitioners and researchers to better understand this domain and to learn how the findings apply to a generalized grouping of open communities.

OPEN COMMUNITIES

Open communities are reaching a business-critical tipping point as organizations strive to better understand them, eventually seeking to deepen their own involvement with them (von Hippel and von Krogh, 2003; Fitzgerald, 2006). Open communities have become media darlings, garnering considerable popular recognition *and* success. Linux continues to make strong gains as a viable business option (Kelty, 2009). The social networking and micro-blogging site twitter boasts a market cap that has surpassed $1billion in 2009 (*The New York Times*). Flickr, the online photo-sharing site claims over 4 billion images and Wikipedia reports over 16 million articles. These are all tremendous successes where openness and adaptability are valued over management and control (Kelty, 2009). In these cases coordination, contribution, and compliance in open communities become the processes of design for new and emergent systems (Germonprez et al., 2007). Organizations must strive to balance their knowledge of intellectual property, their styles of management, and their notions of control within open communities composed of non-developers, casual bloggers, and corporations. While this is a daunting task in this jungle of development approaches, it is a necessary one in order to participate in open communities and leverage their advantages.

Much open source research has focused exclusively on a single open community and not its interface with participating organizations. For example, Sowe et al. (2008) examine knowledge sharing internal to the Debian open community and Kuk (2006) explores interactions within the KDE open community. In these cases, the focus is on the open community itself and *not* the relationship between the open community and organizational participants. Our study is positioned at the interface of participating organizations and existing open communities. To address this, we provide an iterative process of literature investigation and applied considerations as the research team members represent both academic and organizational interests regarding organizational participation with the open communities.

Across the questions of why and how we consider all participants in an open community to be of equal importance and do not predetermine organizations to be better or worse participants. We aim to understand *why and how* they participate in open communities and issues associated with the critical requirements, motivations, and challenges of participants. In

doing so, we assume that the ecosystem of an open community supports a variety of participants. In the next section we introduce action research as an important approach for contributing to these goals, using it to frame our qualitative field study of Linux open community participation.

RESEARCH APPROACH

We apply action research as a methodological approach within which a qualitative study is conducted (Chiasson et al., 2009). Action research allows us to specifically address practice and research cycles, providing critical structure in defining our project. Action research supports our dual goal of developing a solution to a practical problem which is of value to the people with whom we are working, while at the same time developing theoretical knowledge of value to a research community involved in research and pedagogy (Mathiassen et al., 2009).

We propose a dominant approach of action research to frame our study within which other, more localized research methods are applied (Chiasson et al., 2009). Action research requires specification of an area-of-concern under investigation, a problem-solving context, research frameworks, problem-solving and research methods, and their respective contributions (Mathiassen et al., 2009). Table 1 highlights these action research elements and their application in our project.

Action Research Elements	Action Research Elements in Project	Description
Area-of-Concern	The Organizational Value of IS IS Management	Why organizations participate in leveraged models and how organizations manage the interface with the Linux open innovation community.
Real World Problem Setting	Linux Open community	Organizational participant types understood around contributions and differentiation. A practical examination of organizational participation in the Linux open innovation community.
Framing Based on Area-of-Concern	Open Community Participation	Structure to the applied issues associated with open innovation community participation (von Hippel and von Krogh, 2003; Neus and Scherf, 2005)
Framing Independent of Area-of-Concern	Communities of Practice Open Community Interaction	Participation in open innovation communities of practice (Brown and Duguid, 1991; Wenger, 1999) and open innovation community theory (Chesbrough, 2003)
Problem-Solving Method	Leveraged Models	A 'leveraged' system that has shared value for all members with lower costs for each participant than if they developed on their own (Neus and Scherf, 2005).
Research Method	Interviews	Action research as a dominant approach, including interviews (Chiasson et al., 2009).

Table 1: Action Research Elements (Mathiassen et al., 2009)

Action Research: Development and Discovery Phase

Action research was used to achieve two outcomes. First was a developmental round of data collection to establish grounding for the project. To achieve this outcome, the investigation was rooted in practice, not academe, to foster a strong problem-solving connection. Rooting in practice provides an opportunity to embed practical concepts from the Linux open innovation community into our researched areas of concern. A similar approach was used by Davison and Martinsons (2002) to investigate how the practical use of GSS could inform organizational culture. As such, industry participants were interviewed regarding the broad issues of why and how organizations participate in open innovation communities. The primary outcome associated with this phase of the action research was the *development of the interview questions*. In all, three organizations were involved in the development of the research questions, iterating over the course of six months. The interview questions have a strong practice orientation, and their high applicability to a variety of open innovation community participants provided traction for our second outcome.

The second outcome of the action research approach was to discover the characteristics associated with why and how organizations participate in open innovation communities. Interviews with members of participating organizations were conducted in the *execution of the interview questions*. Participating organizations were identified through personal

contacts, Linux Foundation membership, and online media. Each interview lasted approximately 1 hour depending on the depth of the answers. To date, 15 interviews have been performed and analyzed thematically. The interviewees were both developers and managers directly associated with Linux open innovation community participation. The 15 interviewees represented 9 different organizations, all rooted in the technology industry.

EARLY FINDINGS AND INSIGHTS

These findings constitute our progress through one action research cycle. They represent the early phases of a long-term research project to engage organizations and provide usable data to better understand open community participation. Additionally, the findings are applicable to both practice and academe and are presented in a co-mingled fashion to increase their continuity and understandability. Within our presentation we provide quotations from interviews to highlight our findings.

Why Organizations Participate in Open Communities

Open communities provide flexibility and adaptability as an option through this fundamental principle: we all give a little; we all get a lot. This has the benefit of enabling 'leveraged design' of a system that has shared value for all participants. The design and development is agile and distributed through a community where the members have shared responsibilities for a system (Ågerfalk et al., 2009). This could be operating systems (Linux kernel), image aggregation (Flickr images), or content management (Wikipedia articles). Each respective system is built through a model where design and development are leveraged through participants, value is provided for all, and prediction, planning, and control are the domain of an open community, not the Linux Foundation, the executive committee of Flickr, or the founder of Wikipedia. Open communities represent a paradigm shift in how systems are designed and developed (Neus and Scherf, 2005; Kendall and Kendall, 2008, and Kelty, 2009). Organizations have traditionally approached designing systems as a proprietary process, so why shift to an open community model? The answer lies in the notion that the costs of designing and developing in an open community can be reduced via the leveraged development model (Figure 1).
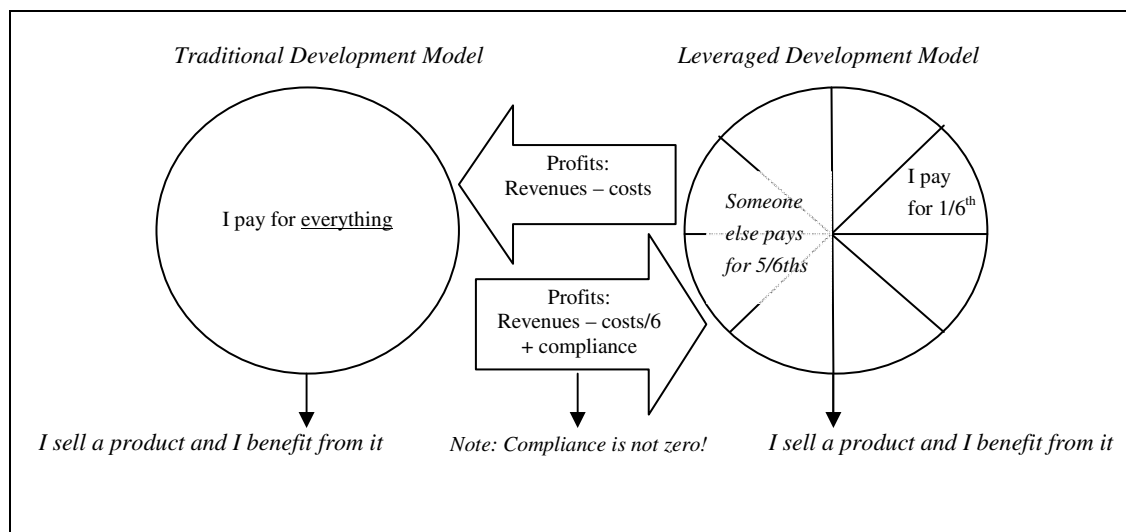

Figure 1: Leveraged Development Model

With the leveraged development model, systems can be developed through the 'leveraging' of the open community where participants contribute portions of a completed system (von Hippel and von Krogh, 2003). The complete system can be used as compliant with the rest of the open community or it can be differentiated for specific organizational settings. For example, "5/6ths" of the Linux open community can port the Linux kernel to a new chipset from which "1/6th" can benefit by selling support services for an industry-wide, compliant system. The "1/6th" can also benefit individually by differentiating the Linux kernel for specific, internal organizational systems. In each case, the complexities and costs of developing each respective issue is distributed throughout the open community, rather than a single organization, and in each case, *contributions and differentiation* play crucial roles in effectively participating in a leveraged development model (Fitzgerald, 2006).

Participating organizations also leverage the open community for both internal support and contracted support. Regarding internal support, organizations aim to have their contributions to the community accepted 'upstream' meaning

that the contributions are sent back to the community and are accepted and subsequently released in future versions of, in our case, the Linux kernel. This permits the original, participating organization to receive support from the open community through daily kernel testing. If contributions are accepted, support is implicitly provided by the open community.

Regarding contracted support, organizations are able to leverage the Linux open community when entering third-party contracts. Organizations can on one hand, contract development work to a third party (who is not an open community participant). Organizations can then actively participate with the Linux open community, aiming to have contracted work accepted 'upstream.' In doing this, they are able to return successfully contracted development work back to a client, while shifting support to the Linux open community. The consultant becomes 'free and clear' of the maintenance of the contribution, while at the same time maintaining their own strong citizenship within the open community.

Participation in the Linux open community is a balance of organizational management and individual developers. In our interviews, organizational management has been unilaterally aware and supportive of participation. From a management perspective, participation is generally driven by one of the aforementioned leveraged models to achieve specific strategic advantages. Individual developers often represent the day-to-day participation with the Linux community and generally have longstanding, personal ties within the community. This creates an interesting situation as individual developers participate with both their employing organization as well as the Linux community, becoming, in the parlance of communication networks, a liaison between the two (Tichy, Tushman, and Fombrun, 1979). The structure of how management and individual developers consider knowledge, power, and property appears quite loose and *ad hoc*, with no formal reporting. Individual developers also harbor differing ideas regarding their own participation and the advisability of learning appropriate skills (Kendall and Kendall, 2005). However, we do not suggest that this *ad hoc* structure is a negative as the structure appears well suited for successful, multi-level interactions necessary while participating in open communities (Hitt et al., 2007).
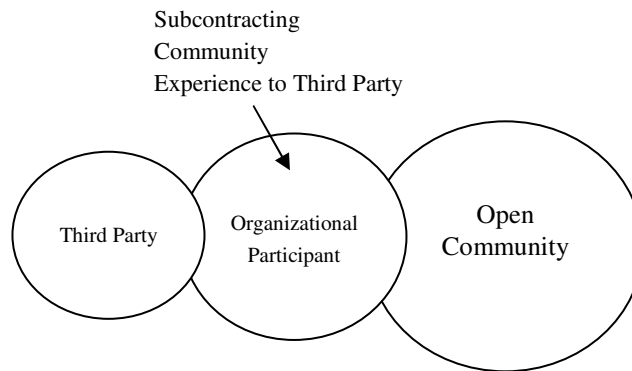
How Organizations Participate in Open communities

We found open community participation to occur in both direct and proxied forms. Direct participation has been evidenced in other open community literature (see Sowe et al. (2008) and Kuk (2006)). In direct participation, the focus is on the one-to-one relationship between the organization and the open community. The relationship is direct in the sense that there is a single community to which multiple organizations participate (Figure 2).
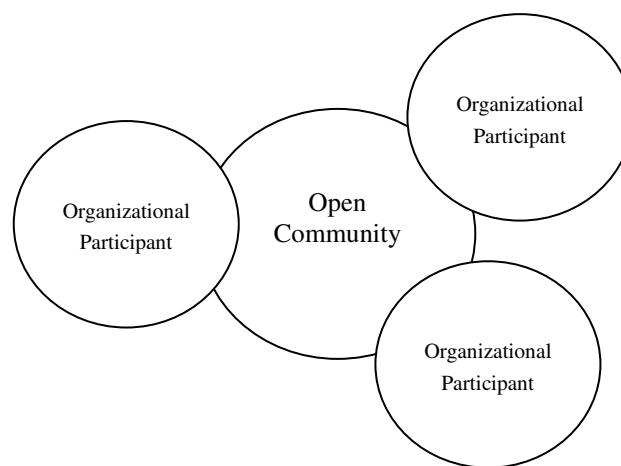
Figure 2: Direct Open Community Participation

We also found proxied participation as organizations provide contracted support. In proxied participation, there are multiple layers of participation, where open community participation is done by proxy. In this case, an organizational participant sub-contracts their reputation and experience within the open community to a third party (Figure 3).

Figure 3: Proxied Open Community Participation

In proxied participation, a third party organization works indirectly with the open community through an organization who participates directly. A third party organization may see value in contributing changes back to the community but is uninterested in learning the behaviors necessary for direct participation, contracting out those participatory activities.

In both direct and proxied designs, open communities require the commitment of participants dedicated to common goals. We see commitment as *contributing* in a variety of forms. Contributions are the degree to which participants supply committed changes to a product. Contributions are also the engagement with an open community to share, trade, test, and develop ideas (Wenger, 1999). In the context of our research, we identified contributions to the Linux open community as *high contributions* and *low contributions*. A high contributor is a participant actively engaging in the community by developing "1/6" in the leveraged models. A low contributor is a participant far less active with respect to contributions to the leveraged development model. Both types of contributors, high and low, are evident in ecology of open communities as the goals and applications of open community systems vary from participant to participant.

We also found organizational participation to be defined by the adherence to, or differentiation from the open community. *Differentiation* is the degree to which participants modify a stable, publicly-available product for specific organizational requirements. Differentiation requires participating with the open community, understanding changes, and differentiating a product away from the open community. Differentiation does not have a zero cost (Wenger, 1999); it requires internal development support from the differentiating organization but is expected to cost consistently less than non-leveraged development. Like the contributions, differentiation is viewed in two forms: *high differentiation* and *low differentiation*. Low differentiators are participants engaged in ways generally prescribed by the open community. As an example in the Linux open community, chip manufacturers could be low differentiating participants, as their processors should work with the largest, most stable release of the Linux kernel. High differentiators are participants engaged in specialized ways that are not necessarily in compliance with the majority of participants. As an example, manufacturers of embedded devices may differentiate a product in the development of tailored or customized devices specific to organizational strategies. High differentiating participants create new or 'forked' systems that are quite different from their open community. Table 2 illustrates a matrix of contributions and differentiation.

<table>
<tr><td colspan="2"></td><td colspan="2" align="center">Differentiation</td></tr>
<tr><td colspan="2"></td><td align="center">Low</td><td align="center">High</td></tr>
<tr><td rowspan="4">Contributions</td><td rowspan="2">High</td><td>- Industry standard, commodity (or close to it) system
- Large participant base outside of the company driving the innovation

Linux open community example: x86 chip vendors, Linux consultancy</td><td>- Highly specialized system developed by one company
- Feature exploitation requires very detailed and specialized knowledge

Linux open community example: Mainframes, UNIX-class systems, high end databases</td></tr>
<tr><td rowspan="2">Low</td><td>- Uses industry standard parts, with little specific differentiation
- Large and savvy user base
- Open system stack

Linux open community example: Commodity x86 server vendors, 'hackable' embedded devices like routers</td><td>- Highly specialized system/service meant to operate as a black box
- Interface is (intended to be) closed to the consumer / hacker

Linux open community example: Flat screen TV, DVR, cars, appliances</td></tr>
</table>

Table 2: Contributions and Differentiation

Participant Type: High Contributor/Low Differentiator

High contributors/low differentiators supply contributions that are compliant within the respective open community. They can be paid as employees contributing to the Linux kernel or volunteer participants contributing to their favorite charity's Wikipedia page. High contributors/low differentiators have the ability to help define and maintain a strategic roadmap for the open community. They focus on lowering overall community development costs, improving system time to market, and increasing the adoption of the system for a broad public. As a high contributor/low differentiator, effective communication, strong external relationship management, and internal organization structure for fostering contributions are expected.

Participant Type: High Contributor/High Differentiator

High contributors/high differentiators are contributors who choose to differentiate their application of the common or 'mainline' system. This is done when a mainline system is applied in a system-specific manner with knowledge concentrated and applied strategically within the organization. High contributors/high differentiators are active participants in their open communities to maintain an understanding of community processes and future integration in an existing organizational innovation stream. Like the high contributors/low differentiators, the high contributors/high differentiators are interested in lowering development costs and improving time to market. But what distinguishes them is that they are also interested in differentiating in an otherwise commodity market and maintaining ties to an existing innovation stream for skills availability for internal design and development. Challenges for high contributors/high differentiators come from earning and maintaining trust with the open community and communicating and aligning the internal and external motivations associated with the respective system of the open community (Henkel, 2006).

Participant Type: Low Contributor/Low Differentiator

Low contributors/low differentiators do not actively contribute to the open community, mainly participating by viewing the open community in a commodity-like role, considering the community responsible for the design and development of systems to run on top of or underneath a private solution. Perhaps in working with the 'mainline' system of the open community there is a potential for contributions through testing and use, but the overall participation is limited. Low contributors/low differentiators have a heavy reliance on industry standards and organizational product innovation is driven from elsewhere in the value chain (Henkel, 2006). The low contributor/low differentiator is a common role for organizations since the open community supports a broad range of solutions with little internal effort; 'a rising tide floats all boats' irrespective of their role within an open community. Rightfully, the low contributors/low differentiators have little influence on the open community design decisions and much less opportunity for specialization within the community.

Participant Type: Low Contributor/High Differentiator

Like low contributors/low differentiators, low contributors/high differentiators do not contribute back to the open community in a consistent way. They differentiate the mainline system of the open community, creating a black box around the new, differentiated, and private system. Instances could include the need to build systems or services with very specific needs, but this comes at the expense of having to singularly maintain the differentiated or forked system and sacrificing much of the leveraged development model (Neus and Scherf, 2005). At a minimum, low contributors/high differentiators must adhere to the open community licensing requirements. The low contributor/high differentiator is an excellent model for embedded applications and can result in major competitive advantages, using the open community as a launch pad for the differentiated system. As a high differentiator, maintaining and synchronizing parallel lines of similar systems can become onerous and expensive.

The aforementioned participant types of contribution and differentiation are community-based perspectives on how organizations participate with open communities. Within an organization, questions of "how" remain regarding the more pragmatic, daily relationships with an open community in the management of property, knowledge, and power. Knowledge appears to be managed through the multi-level interactions of the individuals who participate with the community on a daily basis, the organization who defines strategy for participating, and the community itself as a pillar of value (Hitt et al., 2007). The pragmatic, daily relationships regarding property, knowledge, and power are deeply intertwined. We have not unraveled the complexities but see, at a broad level, a relationship structure. At the community level there is a stable and rational meritocracy, where technical and social competence guides the direction of the community. At the individual level there is personal commitment to the community formed independently from

organizational commitment. Technical competence aside, the learning focuses on *how* to socially participate in the meritocracy of the Linux open community. At the organizational level there is a mutable management structure defining participation. Mutable management appears acceptable because of the power distribution shaping the multi-level interactions of the Linux community. To overly structure the management of daily participation could hinder the adaptability of the organization to react to changes that emerge from within the open community and the changes that are engaged by individuals. Management appears focused on realizing the strength of individuals and the community for the implementation of the leveraged models. The internal *how* questions of participation with the Linux open community require additional time and effort to comprehend their nuances and to then communicate at length.

CONCLUSIONS

As corporate participation increases in open communities, we expect to see increasing layers of understanding to emerge at the interface of organizations and open communities. Technical solutions have already emerged with a focus on managing applications and projects, not people. For example, technical solutions provide alerts to application owners who use open source code, where the mainline open source code shows signs of failure. In light of technical solutions, an opportunity remains for understanding the behavioral aspects of participation so the time it takes to enact community engagement and realize code acceptance can be improved, in essence, gaining an increased access to the leveraged models of development and support.

We have found social models of apprenticeship within organizations. These models are encouraging and can act as broader templates for establishing guideposts for individual learning inside participating organizations. We would expect behavioral approaches to vary based on the defining participant type (contribution and differentiation). As such, work remains to identify commonalities between and uniqueness within each participant type. Opportunity also remains as the apprenticeship model expands, not only within an organization, but also within university curriculum. By fostering a relationship with organizations participating in open communities, universities can 'relieve some of the burden' of teaching incoming employees about open community participation. This can create stronger practice-based relationships for universities as well as allow organizations to focus more on emerging issues in open community participation.

Open community participation is not a solution to all design and development projects. However, open communities and leveraged models represent viable approaches to real organizational issues (Fitzgerald, 2006). As the number of participating organizations continues to increase in a field that has only gained considerable strength in the last 15 years, many opportunities remain for both practice and academe. The evolution of these opportunities is unfolding before us and this domain is far from understood. This paper provides a first glimpse of *why* and *how* organizations participate with open communities. However, much of the story still remains to be told.

REFERENCES

Ågerfalk, P., Fitzgerald, B., and Slaughter, S. (2009) Introduction to the Special Issue, Flexible and Distributed Information Systems Development: State of the Art and Research Challenges, *Information Systems Research,* 20, 3, 317-328.

Brown, J. and Duguid, P. (1991) Organizational Learning and Communities-of-Practice: Toward a Unified View of Working, Learning, and Innovation, *Organization Science,* 2, 1, pp. 40-57.

Chiasson, M., Germonprez, M., Mathiassen, L. (2009) The Conventional Role of Research Methods in Information Systems Action Research, *Information Systems Journal (ISJ)*, 19, 31-54.

Fitzgerald, B. (2006) The Transformation of Open Source Software, *MIS Quarterly,* 30, 3, 587-598.

Germonprez, M., Hovorka, D., Collopy, F. (2007) A Theory of Tailorable Technology Design, *Journal of the Association for Information Systems (JAIS)*, 8, 6, 315-367.

Henkel, J. (2006) Selective Revealing in Open Innovation Processes: The Case of Embedded Linux*, Research Policy* 35, 7, 953-969.

Hitt, M., Beamish, P., Jackson, S., and Mathieu, J. (2007) Building Theoretical and Empirical Bridges Across Levels: Multilevel Research in Management, *Academy of Management Journal,* 50, 6, 1385-1399.

Kelty, C. (2009) *Two Bits: The Cultural Significance of Free Software.* Duke University Press.

Kendall, J.E. and Kendall, K.E. (2005) Agile Methodologies and the Lone Systems Analyst: When Individual Creativity and Organizational Goals Collide in the Global IT Environment, *Journal of Individual Employment Rights*, 11, 4, 333-347.

Kendall, J. E. and Kendall, K.E. (2008) Designing for Positive Outcomes: Using Positive Memes, Distributive Empowerment, Open Source Development, and Positive Metaphors in *Designing Information and Organizations with a Positive Lens: Advances in Appreciative Inquiry*, Volume 2, M. Avital, R.J. Boland, D. L. Cooperrider (Eds.), Amsterdam: Elsevier, 147-165.

Kuk, G. (2006) Strategic Interaction and Knowledge Sharing in the KDE Developer Mailing List, *Organization Science*, 52, 7, 1031-1042.

Mathiassen, L., Chiasson, M., Germonprez, M. (2009) Compositional Styles in Action Research: A Critical Analysis of Leading Information Systems Journals, Sprouts: Working Papers on Information Systems, 9, 35.

Neus, A. and Scherf, P. (2005) Opening Minds: Cultural Change with the Introduction of Open-Source Collaboration Methods, *IBM Systems Journal,* 44, 2, 215-225.

Sowe, S., Stamelos, I., and Angelis, L. (2008) Understanding Knowledge Sharing Activities in Free/Open Source Software Projects: An Empirical Study, *The Journal of Systems and Software,* 81, 3, 431-446.

Tichy, N., Tushman, M.L., and Fombrun, C. (1979) Social Network Analysis in Organizations, *Academy of Management Review*, 4, 4, 507-519.

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

Wenger, E. (1999). *Communities of Practice: Learning, Meaning, and Identity.* Cambridge University Press, New York.