Boundary Organizations: Enabling Collaboration among Unexpected Allies

Siobhán O'Mahony
University of California, Davis
Beth A. Bechky
University of California, Davis

Our research examines how parties challenging established social systems collaborate with defenders of those systems to achieve mutual goals. With field interviews and observations from four community projects in the open-source movement, we examine how these projects collaborated with firms defending proprietary approaches to software development. Drawing on social movement and organizational theory, we explain how challenging parties not only mobilize to achieve their goals but how they are able to transform contestation into collaboration. Open-source projects and firms held divergent interests but discovered areas of convergent interest and were able to adapt their organizing practices to collaborate through the creation of a boundary organization. By showing how boundary organizations help challengers and defenders manage four critical domains of organizing practices—governance, membership, ownership, and control over production—we provide analytic levers for determining when boundary organizations work. At the same time, we reveal the subsequent triadic role structure that unfolded among communities, the boundary organizations they designed, and firms.●

Social movements typically mobilize to support the claims of underprivileged or underrepresented parties. Pursuing these ends often entails challenge, contestation, or protest of established social systems (Gamson, 1975). Whether fighting for the environment (Hoffman, 1999; Lounsbury, Geraci, and Waismel-Manor, 2002), civil rights (McAdam, 1982; Andrews, 2001), or the benefit of farmers, the poor, or women (Piven and Cloward, 1977; Clemens, 1993; Ganz, 2000; Soule and King, 2006), members of social movements often confront vested interests from a position of limited resources and power (Gamson, 1975). Thus a social movement's ability to achieve its goals depends on its ability to frame its cause (Snow et al., 1986; Snow and Benford, 1992), mobilize supporters (McAdam and Paulsen, 1993), create tactical innovations (McAdam, 1983), and leverage political opportunities (McCarthy and Zald, 1977).

When social movements organize for change, some parties stand to gain, while others fear losing ground. Both theorists and observers assume that social movements and the targets of their collective action maintain diametrically opposing goals, with little hope for mutual gain. Sharp dichotomies are drawn between the interests of challenging and defending parties, and these interests are presumed to be fixed (Giugni, 1998). But, as Rucht (2004) argued, the reality is far more nuanced. Only some social movement conflicts involve direct opposition, and allies can be critical to a movement's survival (McCarthy and Zald, 1977). Unfortunately, the role of brokers and mediators in helping social movements to achieve their goals is understudied (Rucht, 2004), and little research has examined why or how those challenging a social system would collaborate with supporters of the systems they hope to change.

Recently, scholars taking a closer look at the intersection of social movements and organizations have suggested that social movements can create new organizational boundaries through truces among competing factions in a field (Rao,

0001-8392/08/5303-0422/$3.00.

●

This research was supported by the Stanford University Center for Work, Technology, and Organization and the Social Science Research Council Program on the Corporation as a Social Institution. We are indebted to Steve Barley for his support and guidance. Woody Powell, Mark Granovetter, Robert Sutton, Neil Fligstein, Kathleen Eisenhardt, Doug Guthrie, Doug MacQuarrie, Fabrizio Ferraro, Brent Williams, and Mark Zbaracki also helped shape the theoretical development of this research. We thank participants from the following seminars: Harvard Business School, Columbia School of Management, Boston College School of Management, MIT Sloan School of Management, INSEAD, Georgetown, and the University of California at Irvine. The participants in the Technology and Organizing Retreat of the Communications Department at the University of Washington offered useful suggestions for revision. We also thank Calvin Morrill and our reviewers for their very helpful ideas. Jan Chong, Lilly Irani, Clare Flaherty, and Alyssa Razook provided invaluable editorial and research assistance. We appreciate the time and consideration extended by informants from the free software and open-source communities.

1998; Rao, Morrill, and Zald, 2000). For example, a change in the framing of the recycling movement helped it grow from a marginal practice promoted by activists to a major for-profit industry (Lounsbury, Ventresca, and Hirsch, 2003). The movement, once hostile to commercial interests, shifted its conceptions of recycling from a means to restructure capitalistic production to a frame that enabled a profitable recycling industry (Lounsbury, 1997, 2005; Lounsbury, Ventresca, and Hirsch, 2003). Contrary to traditional predictions that capitalist interests could only come at the expense of social movements' interests, this frame change furthered the movement's goals, suggesting that opportunities for mutual gains among challenging and defending parties may be more robust than previously appreciated.

Understanding the complex relationships among social movements and other types of collaborators requires a more expansive conception than we currently have of the outcomes of social movements. Social movement scholars traditionally have viewed a movement's outcomes narrowly as the ability of a movement to achieve political or policy goals (e.g., Gamson, 1975; Amenta and Caren, 2004). Focusing on state-related (and more measurable) outcomes has kept scholars from developing a comprehensive understanding of how social movements can effect change in socio-economic and cultural contexts (Earl, 2000, 2004; Amenta and Caren, 2004). Change in such contexts requires considering outcomes external to the movement and attending to the mechanisms linking social movement activities and outcomes (Earl 2000, 2004). Movements often shift between modes of contestation against and mobilization within existing social structures (Schneiberg and Lounsbury, 2008). For example, as Clemens (1997) showed, social movements can "borrow" familiar forms for their own purposes to enhance their legitimacy in the eyes of external parties. To discover how challengers and defenders of established systems adapt through interaction, however, requires examining what happens not only during contestation and mobilization but also when contestation is transformed and settlements occur. This research takes a step in this direction by showing how challengers and defenders of the social system of software production discovered convergent interests and devised a means to collaborate that would not threaten their divergent interests.

MANAGING BOUNDARIES IN COLLABORATION

Although social movements have traditionally aligned themselves with socio-economic or class interests, more recently, they have mobilized to pursue interests such as learning and enhancing their ties with others (DiMaggio and Anheier, 1990), challenging dominant bodies of knowledge (Frickel and Gross, 2005), introducing new technologies (Hargrave and Van de Ven, 2006), or expanding modes of production (O'Mahony and Chen, 2007). Those involved in such movements are less likely to be interested in policy outcomes and more likely to contest cultural values and beliefs (Earl, 2004). This can lead to the construction of alternative systems of production (Rao, 1998; Carroll and Swaminathan, 2000; Schneiberg, 2002). For example the "slow food" movement (Pollan, 2007; Waters, 2007) and the craft movement (Walker, 2007)

engage in creative, small batch production as a form of protest against mainstream industrial production.

These social movements organize not to protest established systems but to further the collective production of scientific, artistic, technical, or general knowledge (e.g., Moore, 1996; Frickel and Gross, 2005; Hargrave and Van de Ven, 2006). Such movements are often less centralized than traditional movements, comprising loosely connected communities that independently organize or produce goods and services to further collective goals. To account for new forms of collective action that operate outside of political realms, Snow, Soule, and Kriesi (2004: 9) proposed a more expansive definition of social movement activity, suggesting that movements can be "considered as challengers to or defenders of existing *institutional authority*—whether it is located in the political, corporate, religious, or educational realm—or patterns of *cultural authority,* such as systems of beliefs or practices reflective of those beliefs."

Yet because most social movement theory examines the strategies of social protest, less is known about how these new social movements affect the behavior of others and create change outside the movement. Recent research shows that movements engaged in production as a mode of opposition have made significant creative and economic contributions to society (Lakhani and von Hippel, 2003; Dahlander and Magnusson, 2005; Shah, 2005; von Hippel, 2005). Large-scale scientific and technical enterprise often involves networks of collaborators from different types of organizations (Powell, Koput, and Smith-Doerr, 1996; Powell et al., 2004), such as firms, universities, communities, and movements. Although participants may share a superordinate goal that unites their collective effort, each party also maintains disparate interests.

Balancing Interests in Collaboration

Organizational theorists have found that collaborations can be difficult when the interests, goals, and practices of participants differ. For example, universities often struggle to define terms for collaboration with firms in patenting, licensing agreements, and joint ventures. Universities must balance their interest in fostering commercial partnerships with their interest in providing open access to new knowledge (Murray and O'Mahony, 2007). A similar balance must be found when firms participate in alliances, associations, and communities. Participants benefit by identifying mutual interests and opportunities for collaboration (Sabel, 1984; Rosenkopf, Metiu, and George, 2001), sharing risk (Tushman and Rosenkopf, 1992; Rosenkopf and Tushman, 1994, 1998), and exchanging technical information (Allen, 1983; von Hippel, 1988; Saxenian, 1994) but must weigh these benefits against their interest in maintaining a distinct competitive advantage.

Even when mutual gains from collaboration can be identified, managing the boundaries of collaborations is essential if parties are to preserve their disparate interests. In a study of 89 of the most research-intensive U.S. universities between 1991 and 1998, Owen-Smith (2003) found that collaborations between universities and commercial firms could benefit

both parties. The more connections university technology licensing offices maintained with industry collaborators, the greater the impact of their patent portfolios. Licensing offices that were too tightly connected to industry had less innovative patent portfolios (Owen-Smith and Powell, 2003). When boundaries between academia and science were not adequately maintained, it affected the nature of the knowledge produced (see also Mowery and Ziedonis, 2002; Murray, 2007).

As Rucht (2004: 203) suggested, social movements seeking alliances also hope to retain "some of their autonomy and distinctiveness, and therefore refrain from merging into a single entity." Thus if challenging and defending parties are to ally, they must find a way to bridge their differences without threatening the core values that make them distinct, but little research has explored the two facets of this relationship—how the interests of different parties become aligned and how their disparate interests are preserved. If we are to explain how mutual gains are created, particularly among challengers and defenders of established social systems, then examining the process by which such collaborations occur is essential.

Managing Boundaries in Collaboration

The science-studies tradition provides additional theoretical grounding for what happens during collaboration, as it focuses on how actors use boundary management strategies, behaviors, and objects to collaborate across diverging worlds. In the worlds of science and medicine, divergent interests often influence the outcomes of collaboration (Latour, 1987; Fujimura, 1988; Star and Griesemer, 1989). For instance, Timmermans and Leiter (2000) described how social movements and professional organizations created a new distribution system for the controversial drug Thalidomide despite competing concerns over its effects. Although this drug is well known to cause birth defects in utero, it can also be used effectively to treat serious diseases, including leprosy and AIDS. While the FDA deliberated over approval, activist AIDS patients pressured doctors for effective treatments, while activist Thalidomide victims pushed to minimize the likelihood of future birth defects. To distribute the drug safely and effectively, all of the actors involved—patients, physicians, regulators, victims, manufacturers, and pharmacists—negotiated from their interests.

Collaboration not only requires negotiating interests but can also involve changes of roles and relationships (Moore, 1996; Nelsen and Barley, 1997; Gieryn, 1999). Because of the social movements' actions in the Thalidomide case, the resulting distribution system "reconfigured the responsibilities and roles of the actors within the system"; for example, pharmacists were given the autonomy to refuse to fill prescriptions that did not satisfy agreed upon procedures (Timmermans and Leiter, 2000: 43). Similarly, in Murray's (2007) research on the first genetically modified Oncomouse, after academic scientists mobilized to contest access to the mouse, industry and universities designed new mechanisms for the mouse's distribution. What helped align disparate academic and indus-

try interests were new rules that distinguished between commercial and academic usage of the mouse.

In scientific and technical collaborations, participants create standards, methods, and objects to bridge the boundaries between different social worlds. Boundary objects have a common structure yet remain flexible in interpretation, which enables their use across worlds with different interests (Star and Griesemer, 1989). Research demonstrates how challenging and defending parties use boundary objects to shape the distribution and application of knowledge across distinct occupational communities (Carlile, 2002, 2004; Bechky, 2003a, 2003b). In these contexts, workers use boundary objects to transform domain-specific knowledge so that it can be used toward a shared goal (Bechky, 2003b).

Sociologists of science have proposed the related construct of a boundary organization to describe the intermediary organizations that align the divergent interests of science and politics (Guston, 1999, 2000, 2001; Miller, 2001). Boundary organizations facilitate collaboration between scientists and non-scientists by remaining accountable to both (Guston, 2001). They "perform tasks that are useful to both sides and involve people from both communities in their work but play a distinctive role that would be difficult or impossible for organizations in either community to play" (Guston, 2001: 403). Boundary organizations can enable challengers and defenders to substantively collaborate by building a bridge between divergent worlds that allows collaborators to preserve their competing interests. Boundary organizations make collaboration possible by enrolling actors on the basis of their convergent interests. As Latour (1987: 109) pointed out, the easiest way to forge collaboration is to "tailor the object in such a way that it caters [to] people's explicit interests," as this creates a tension that enables actors to choose elements that meet their goals.

Like boundary objects, boundary organizations can accommodate the varying interests of parties by providing a mechanism that reinforces convergent interests while allowing divergent ones to persist. Unlike boundary objects, however, the concept of boundary organizations allows us to focus on the organizational mechanisms and processes that enable collaboration. Rather than objects that are highly transportable (Fujimura, 1988) and "weakly structured" when used in different locations (Star and Griesemer, 1989: 393), boundary organizations are more durable structures that encourage parties to isolate and organize around their convergent interests. Though they are stable, boundary organizations share the interpretive flexibility of boundary objects, enabling parties' divergent interests to coexist, as they seek collaboration while pursuing mutual goals.

METHODS

We conducted an inductive, ethnographic study of the practices of communities and firms in the open-source software movement. The open-source movement fits Snow, Soule, and Kriesi's (2004) more inclusive definition of a social movement: organized collectivities challenging institutional authority in the software industry. The open-source social move-

ment targets a different actor than traditional movements: corporate actors and the established system of proprietary software production. Members believe that people should have access to alternatives to closed-source, proprietary software and thus they participate in community-managed projects to produce open-source software. These community-based projects are globally distributed and directly challenge the established industry system of closed-source, proprietary software. Rather than contest the perceived injustices associated with proprietary software as traditional movements might, however, their mode of challenge is to produce an alternative form of software. And the proprietary software industry did view the open-source software movement as a challenger. The dominant firm in the industry, Microsoft, performed a competitive analysis on the open-source software movement (Villoppillil, 1998). This document, released in October 1998, stated that "to understand how to compete against OSS *we must target a process rather than a company.*" Microsoft also named the open-source movement as a competitor to provide evidence of competition in the operating system market in its defense against monopoly charges.

The open-source community projects we studied were instantiations of participation in the movement, arenas in which the community challenged the established proprietary software production system. By producing software, open-source software community projects fulfilled the broad political goals of the movement. Because we were interested in changes in methods of production, we focused on the project level of analysis. Our informants considered themselves to be part of a social movement but used the emic term "community" to refer to their fellow participants in the movement and "project" to refer to their specific software project. We thus use their terms "community" and "project" in this article, rather than "movement," which reflects a more macro level of analysis. By focusing on the project level, we were able to engage in a micro-level study of interactions that would be more difficult to trace at the movement level.

A community-managed, open-source project is publicly initiated to produce open-source software and is managed by a distributed group of individuals who do not share a common employer (O'Mahony, 2007). Linus Torvalds' initiation of the Linux operating system may be the most familiar example. An individual or group of individuals create a mailing list and a publicly available space for the project on a ".org" domain name or a privately hosted site. Although access to the code base may be controlled more restrictively (von Krogh, Spaeth, and Lakhani, 2003; O'Mahony and Ferraro, 2007), access to the source code and development communication is available to the public. Sub-project maintainers feed their work into larger modules managed by package managers, subproject heads, or release coordinators, depending on the project. Technical discussions about what ideas should move the project forward occur on mailing lists. Though contributors may be sponsored by firms, technical decisions remain independent of employment relations: this is core to a community-managed open-source governance model (O'Mahony, 2007). Table 1 compares the open-source community-managed

Table 1

Comparison of Established Proprietary and Community-managed Open-source Software Development Models

	Established proprietary software development model	Community-managed open-source software development model
Initiator / owner	Proprietary firm initiates project and owns the intellectual property associated with it.	Individuals initiate projects and may retain copyright or assign it to a non-profit foundation.
Team / membership composition	Organized by a firm-appointed project manager; may include organizational members and contractors who are provided access to proprietary data.	Anyone can participate from anywhere in the world. Core team emerges based on merit and effort. Decision making at higher levels by invitation from the core team.
Technical context	Development site is proprietary, hosted within organizational firewalls.	Development site is public and hosted by an individual or nonprofit.
Resources	Firms dedicate labor and technical resources. Project may halt or change based on changes in strategy or the market.	Labor is mostly self-selected volunteers. Technical resources may be donated. Project will continue as long as there is interest.
Technical specifications	Technical requirements are based on customer analysis with input from marketing at early stages in the project.	Ideas emerge from a diverse pool of distributed contributors and are accepted or rejected by the core team based on a contribution's ability to help the project.
Release patterns	Customer's requests for changes may be incorporated into new versions. There may be a long lag time between versions with little on-going development visible.	New versions are released early and often. Development between versions is transparent. Stable and non-stable versions are available; work in progress is visible on the project site.
Customers / users	Governments, education, large firms, small and medium firms, and consumers.	Other open-source developers, firms in the computer industry and technical users.
User / developer relations	May provide preliminary releases to other external developers working on complementary applications, but these are managed according to corporate timetables. Developers have little access to work in progress.	Expanded base of users is critical for beta testing and bug-fixing work in progress. Few fixed deadlines limit participation. Users may progress to developers if qualified.

approach to software development with the established commercial proprietary approach.

To understand how social movement parties challenging established systems collaborate with defenders of those systems, we selected open-source software communities that varied in their receptivity to collaboration with commercial interests. A range in the degree to which projects desired association with firms maximized the variance under which common constructs would hold. Based on preliminary interview data, four projects, described in table 2, were selected along a continuum from cordial to distant community/corporate relations. A project's stance toward commercial activity was based on (1) the type of software license it used, (2) the degree and number of firms that the project was working with, and (3) attitudes toward commercial firms drawn from a sample of members on the projects. Community-managed projects can signal their receptivity to commercial collaboration by the type of software license they choose. Three out of four projects in this study used the GNU General Public License (GPL), the most common open-source software license. This license requires derived works to provide source code, effectively restricting a firm's ability to make proprietary, closed-source extensions of the code. The Webserver Project did not use this license but a version of an academic open-source license that permits any form of derived works

Table 2

Theoretical Rationale for Selection of Community-managed Open-source Projects

Project description	Theoretical rationale
Webserver Project: a project to develop and maintain a commercial-grade Web server	One of the first to forge formal commercial relations with a Fortune 500 firm, this project used a commercially friendly open-source license.
GUI Desktop Project: a project to develop a user-friendly graphical user interface (GUI) for the Linux desktop	Because further commercial adoption of Linux depends on GUI improvements, innovation in this area is critical. It also caught the attention of several large firms that wanted to improve user interfaces.
Compatibility Project: a project to develop a standard framework and tools for all Linux distributions	An open-source project dedicated to improving interoperability among Linux distributions, this project had crosscutting effects on both community and corporate open-source efforts.
Linux Distribution Project: a non-commercial Linux distribution	The largest and only non-commercial Linux distribution, this project had a strong identification with the founding principles of the open-source movement and less interaction with firms.

(Rosen, 2004). Licensing terms dictated the minimal legal obligations of any party making commercial extensions of the community's work.

The Webserver Project was one of the first to establish formal relations with a Fortune 500 firm; the GUI Desktop Project was building a commercially attractive user interface; the Compatibility Project was developing tools to support a common Linux distribution that could enhance adoption in commercial markets; and the Linux Distribution Project opted not to become involved in commercial distributions but allowed commercial third parties to distribute its code. These four projects were considered to be mature, successful projects: they were capable of producing commercial-grade code that was resold by third parties in commercial markets.

Data on the four community-managed projects were collected from three primary sources: (1) observation at user-group meetings, technical presentations, and conferences; (2) interviews with informants; and (3) project data archived on the Internet that detailed the project's interactions and structural developments. The first author observed informants at over 24 different events (project meetings, user-group meetings, and conferences) and conducted 70 semi-structured ethnographic interviews with core contributors in the open-source community between April 2000 and the end of 2001. Between 2003 and 2005, the first author conducted another dozen interviews and attended another six events to confirm findings from the first round of data collection. Table 3 provides descriptive data on informants.

Informants were assured of anonymity, and all interviews were recorded. We refer to the informants by pseudonyms throughout the paper, and the firms involved have also been disguised when anonymity might be threatened. The structure of the interviews varied according to the informant's role. Gaining an understanding of the informant's introduction to the project and subsequent role was a starting point for all interviews. Data on individual employment relationships and project experiences were followed by discussions of "how things were done on projects" and how project or firm practices had been affected by the rise of commercial interest in

Table 3

Description of Informants (N = 80)

	Proportion
Male	0.96
Female	0.04
Independent volunteers (contributors on their free time)	0.37
Corporate sponsored (paid by firms)	0.63
Identified through face-to face-contact	0.45
Identified through Internet or e-mail introductions	0.55
Interviewed face to face	0.77
Interviewed by phone	0.23
Percent of sample participating in trust network*	0.41
Trust network composition:	
Observer	0.03
Apprentice	0.07
Journeyer	0.34
Master	0.55

* Some members of the free software and open-source communities participate in a trust network called Advogato that allows community members to evaluate each other. Individuals can rank each other's technical ability at observer, apprentice, journeyer, and master levels. The trust network was not used in the formation of this sample as it was discovered during the course of the study. An ex post search on informants that participated in this study indicates that 41 percent of them participated in the network. These data are included to show how those informants who participated in this study were regarded by their peers.

open-source software. Fifty-five percent of informants were identified through face-to-face events, while others were identified online or through snowball sampling. About two-thirds of the respondents could be identified as having a corporate sponsor. Sponsored contributors work for firms but are authorized to work on community projects as part of their employment, while volunteers work on their own time.

Informants from three Fortune 500 firms that were dominant players in their industries (referred to as Alpha, Beta, and Gamma) were part of the sample. These firms were primarily developing products and services complementary to the software produced by community-managed projects. Obtaining the perspective of these different actors (volunteers and sponsored contributors; representatives from small open-source firms and three different Fortune 500 firms) was critical to identifying common themes that held up under variance in motivations, interests, and roles. Project data such as mission statements, charters, bylaws, meeting minutes, and mailing list archives were collected from online archives when available. Table 4 provides an overview of this information by project.

In the first phase of data analysis, we developed cases describing the history and practices of each project to identify how community-managed open-source software projects were affected by the adoption of their code in commercial settings. Areas of interaction and adaptation between firms and communities were identified. It was apparent that informants were both delighted and challenged by the software industry's emergent, yet selective, receptivity toward open-source software. Informants considered the norms of community-managed software development and the demands of commercial enterprise to be inherently and consistently in

Table 4

Description of Open-source Project Foundation Characteristics

	Webserver Project	GUI Desktop Project	Compatibility Project	Linux Distribution Project
Founding mission/goal	To create a commercial-grade freely available webserver	To build a free and easy-to-use desktop environment	To develop & promote compatibility among Linux distributions	To develop a free non-commercial operating system
Date project founded	Feb 1995	Aug 1997	June 1995	Aug 1993
Date of first release	April 1995	June 1998	May 1998	Jan 1994
Primary license used	BSD type	GPL	GPL	GPL
Foundation formed	Yes	Yes	Yes	Yes
Type of corporation	Public benefit	Public benefit	Mutual benefit	Public benefit
Date incorporated	June 1999	Aug 2000	May 2000	June 1997
Nonprofit status	501c(3)	501c(3)	501c(6)	501c(3)
Date awarded	April 2005	Pending	2001	June 1999
Membership association	Yes	Yes	Yes	Yes
Companies as members	No	No	Yes	No
Board officers	Elected	Elected	Elected	Appointed
Number of directors	9	11	9	8
Slated board	No	Yes	Yes	No
Fortune 500 participation	Yes	Yes	Yes	Yes
Corporate roles	Firms can be represented on boards	Advisory board	Advisory committee	Partner program

conflict but also freely identified areas in which interests converged.

Thus, in the second phase, we identified where communities' and firms' interests converged and where they maintained divergent interests and practices. We examined interests to see how these were leveraged, confronted, and worked through in the collaboration process. As we coded the practices, we noted when both types of actors were willing to adapt and when they resisted changing their practices and how this affected the creation of a new boundary organization. During this phase, we categorized the organizing practices that were adapted to foster collaboration into four distinct theoretical domains: governance, membership, ownership, and control of production. We noted the role that boundary organizations played in attenuating differences in the parties' interests and practices. In comparing the data across the projects, we focused on developing a theoretical framework that could explain how challenging and defending parties collaborate.

CREATING BOUNDARY ORGANIZATIONS TO FOSTER COLLABORATION

The boundary organizations created by all four open-source community projects provided an enduring organizational structure that solidified the convergent interests of the two types of parties and attenuated their most critical differences. At the same time, they allowed both parties to preserve critical aspects of their native worlds. Boundary organizations thus became part of a triadic role structure among firms, projects, and nonprofit foundations for producing open-source software. After exploring when the interests of projects and firms were and were not convergent, we show how both

types of actors adapted their organizing practices across four domains: governance, membership, ownership, and control over production. By adapting their organizing practices along these four domains, both parties preserved their divergent interests and yet still found a way to collaborate.

Convergent Interests

As Rucht (2004: 207) suggested, "alliances can be attractive because actors other than movements tend to have potential strengths that movements desperately lack." At the same time, non-movement actors are interested in the new ideas and the base of support brought by collaborations with social movements (Rucht, 2004). In open-source software production, while "there's an inherent and necessary conflict between an economic entity and the community," as an informant on the GUI Desktop Project noted, both parties saw reasons to collaborate. Members of the open-source movement were motivated to expand the scope of open-source software and felt they could benefit from the resources firms could provide. In turn, as open-source projects took hold among users, firms were motivated to tap an emerging market, giving open-source projects some leverage to engage their participation.

Both parties developed an awareness of convergent interests that motivated them to adapt their organizing practices to collaborate. Table 5 maps the interests that converged and diverged among the four community-managed projects and firms. First, almost all community project members were interested in enhancing the quality and the diffusion of their code and discovered that commercial partners could help with this objective. Many felt that part of the open-source movement's mission was to enhance the provision of software alternatives in a market dominated by "the beast in Redmond" (otherwise known as Microsoft). Their aim was to

Table 5

Convergent and Divergent Interests of Firms and Open-source Software Projects

Community-managed open-source software projects	Firms
Convergent interests	
Enhance technical capability, performance, and portability of software for use in the enterprise	Acquire access to technical expertise and improve recruitment of skilled programmers
Improve individual skill through exposure to new commercial performance challenges	Collaborate with skilled experts to solve difficult technical problems; learn how source code can be customized to solve customer problems
Achieve commercial legitimacy and recognition; establish traditional marketing channels	Alleviate power of industry monopoly and enhance their own market share
Enhance project's market share and diffusion	Increase margins through reduced licensing fees
Divergent interests	
Maintain communal form: informal collegial project practices and working norms	Influence project direction to align with firm strategy and timetable
Maintain individual technical autonomy	Acquire more predictability in the software development process to foster firm planning
Preserve transparency and open access to code development in order to foster full participation in community decision making	Pursue partnership and collaboration opportunities with discretion
Sustain project's vendor independence	Establish formal governance mechanisms to shape a project's future

provide people with less expensive and more customizable choices in the marketplace. To fulfill this goal, contributors to community projects needed to increase the market's awareness of their code. Although open-source projects did not engage in commercial sale of their work, project members were greatly interested in diffusing their code to others. Most informants were excited about the growing public enthusiasm for open-source software and derived satisfaction from furthering its reach. One informant, concerned that industry analysts were not accurately tracking his project's "market share" because it lacked traceable "commercial sales," visited industry analysts to provide alternative indicators they could use to learn of his project's diffusion.

Partnering with companies could further each project's interests in diffusion and growth. For example, many informants were aware that their user interfaces were designed for "techies" and were not easily accessible by others. This was particularly an issue for the GUI Desktop Project, whose mission was to create an application that would make Linux more accessible to more non-technical users. Commercial support could improve a project's user interface and expand the user base from a technical audience to one that served commercial enterprises. With greater resources, projects could not only improve the technical quality and reliability of their code but also promote their code through more traditional marketing channels. For example, at a GUI Desktop board meeting, members asked, "How do we get Alpha and other companies to work on technical aspects and promotion?" Although open-source projects did not engage in commerce, many members thought that commercial third parties could play a role in enhancing the code's quality and distribution.

Firms with compatible business models also had an interest in improving the quality and diffusion of open-source code and in taking advantage of the market "buzz" created by open-source software. Furthermore, because many open-source projects provided infrastructure software, these projects did not necessarily pose a threat to the profit margins of firms selling complementary products and services. As Bill, a senior executive at Fortune 500 company Alpha explained, ". . . the executives were kind of heartened by the idea that where open-source seemed to have the most impact was not in the places where we were at risk of losing a lot of money." Bill pointed out how his firm's interests aligned with those of open-source software:

> The closer you are to the operating system the less money we make and the more willing people are to use it [open-source software]. The closer you are to applications, where we make enormous amounts of money, the less people are willing to use it [open-source software]. So we could tell them [the executive council] honestly that while we believed open source is going to be a big transition, and in fact a paradigm shift—it is not going to threaten the places where we make a lot of money in the near term.

These complementarities made all four projects attractive to some commercial collaborators. Firms selling hardware and open-source complements benefited from substituting open-

source code for proprietary code in their products because this increased profit margins.

Other convergent interests were identified through early collaboration probes with firms. As the leader of the Linux Kernel Project (the core of the Linux Distribution Project) explained, an early experiment with a commercial collaborator brought new technical challenges to which volunteer programmers might otherwise not have been exposed:

> Commercial interests brought in a lot of problems that did not used to be there, like new interesting technical problems, like what do you do with terabyte disks and large-scale clustering? Things that many technical people are kind of interested in but they never get to actually play with. . . . For example, there's a lot of people who are interested in doing performance work on extreme loads, and the only place where that actually happens is the commercial setting.

Commercial support could create exciting new technical challenges for programmers to solve, thereby furthering individuals' interests in skill development. Firms could get their most complicated technical problems solved by some of the brightest programmers around, and those new challenges in turn helped motivate and attract volunteers. Without the introduction of technical challenges unique to commercial settings, volunteer contributors would have less exposure to solving them, and it would take longer for their software to become mature enough for use in commercial enterprises.

Both parties shared interests in solving difficult technical problems and improving the quality and expanding the market share of open-source code, but these interests were contingent. To the degree that commercial collaboration could achieve these objectives without compromising communal norms or the project's independence, project members were willing to pursue convergent interests with firms to create mutual gains. To the degree that an open-source project did not threaten a firm's revenue base but could enhance its margins and enable it to remain accountable to customers and stakeholders, firms were interested in collaborating with community projects. Yet there were many areas in which their interests were not aligned.

Divergent Interests

Informants frequently identified areas in which interests of open-source communities diverged from that of firms. Community-managed open-source projects were interested in maintaining a communal form as opposed to having a more formal organization. Informants also held fast to the ideal that community-managed open-source projects were not organized for profit but to produce the best and most resilient code possible through public and open means without regard for market implications. Pierre, a contributor to the Linux Distribution Project, often reminded outsiders who made requests to the project, "You realize that this is a project and not a company." Contributors saw themselves as part of a community, and while this often meant different things to different people, a few core principles were shared.

First was the principle of individual autonomy. Individuals wanted to participate and contribute toward the development

of software on their own terms. Contributors were attracted to these communities precisely because they relished the ability to solve technical problems and improve their skills without authoritative control (Hars and Ou, 2001; Dalle and Jullien, 2003; Hertel, Niedner, and Hermann, 2003; Lakhani and Wolf, 2005). Community members thus wanted to maintain their informal collegial working norms. For example, many contributors explicitly rejected the idea of formal deadlines or requirements: when they worked on a community project, they wanted minimal organizational constraints. As Jim, a contributor to the Webserver Project, explained at a meeting, "We don't want to run an organization. We don't consider that to be fun." The ability to have fun, enhance their skills, and produce great code they could be proud of was what attracted volunteers to contribute to an open-source project.

Second, an open-source community project required transparency and access. Members were proud of their meritocratic decision-making process: everyone defended his or her ideas in public forums, and the ideas that garnered the most support survived. But for all community members to weigh in on collective decisions, members needed equal access to project interactions. The public nature of the open-source development process included not only source code that anyone could modify but also public e-mail discussion lists, in which dialogue about technical problems could become heated. These mechanisms were supplemented by public archives that tracked all software changes and showed any mistakes made.

Third, community members valued their independence. Many were concerned that corporate influences might try to dominate the direction of their projects, which would threaten a project's ability to be independent or "vendor neutral." As Alex, a contributor on the Compatibility Project, explained, ". . . the dangers there are well known . . . that technical excellence will be compromised because of someone's commercial needs." Independence was valued for three reasons: to produce code that was "technically excellent," to ensure that the code did not bend toward a specific company's interests, and to ensure that the code remained under the community's control.

The open development approach that gave individuals technical autonomy, access, and independence challenged the hierarchical, controlled, and stable development approach favored by firms. Corporate interviewees soon discovered that traditional levers of hierarchical influence and positional authority were offensive in the open-source environment. Community members did not respond well to high-ranking corporate representatives who approached them hoping to shape the direction of the project. As Tim, a contributor to the Webserver Project, explained, "The only way to influence an open source project is to contribute code. That is exactly how it should be. . . . I can't stand up and say the [Webserver Project] has to do this."

The inability to make changes to an open-source project through command and control mechanisms stymied firms

interested in collaboration. Firms needed to maintain predictable development cycles to meet commitments to customers and fiduciary obligations. As Jason, an entrepreneur working with the GUI Desktop Project, explained, "If you're a company, you need to raise money or meet quarterly objectives, you actually need to have a release schedule and a road map. . . . We will die if we don't ship on time, so we need a release. The problem is that . . . the thing [code] never stops moving from under us." Project contributors did not release code until "it was ready," and the state of readiness was heavily debated within the community, leading to unpredictable changes. This created considerable costs and vulnerabilities for any company collaborating with the project. Jason continued, "[This is] very expensive for us. You come to work on a Monday and they changed all the method names, you have to change all of your software because they just totally changed the world from underneath you." Firms had an interest in using open-source software, but they did not want to build products that relied on this code without gaining some influence over its development.

Another aspect of open-source development that was incompatible with the commercial approach was the open and public nature of the development process. Firms feared that their every move would be in full view of their constituents: the press, competitors, customers, and investors. One product manager at Fortune 500 company Beta noted, "You can browse the archives of any CVS [concurrent versioning system] lists.[1] You can see every change that our developers mark. This is naked programming. We're developing, sort of, completely in the open." As a product manager at Fortune 500 company Alpha explained, his firm was attracted to the Webserver Project's market share, but "we had to figure out how we were going to interact with the community."

The open nature of the community affected the way in which projects and firms could explore new relationships. Ben, a manager in a startup supporting the GUI Desktop Project, acknowledged that his company's closed-door conversations with project leaders were "not a respectful, democratic way of doing things" but explained that "there was no place we could go to and have a confidential conversation." When it came to initiating a partnership between a community project and a firm, Ben was one of many of the firm representatives who found open-source community practices clashing with the traditional corporate approach to forging alliances. An Alpha executive, puzzled by the perspective of an alliance with the open-source community, asked, "How do we make a deal with a Webpage?" A community member referencing this conversation explained that as "an amorphous group of individuals who did not have any legal status or any way to make commitments," they challenged traditional approaches to "making deals."

Community-managed open-source projects wanted to preserve open access, individual autonomy, and their independence. This relatively unfettered approach to software development was at odds with firms' need for stability and predictability to meet customers' needs. Although many firms were interested in using open-source software in their

1
A concurrent versioning system manages version control by allowing different people to check code in and out of the same code base.

products and services, they were concerned about creating technical product dependencies that could be affected by the whims of a volunteer-based community. Motivated by a subset of convergent interests, open-source communities and firms leveraged those interests without compromising on areas in which their interests diverged.

Domains of Adaptation in Organizing Practices

All four projects struggled to make collaboration with firms work. To do so, project members and firms adapted some of their organizing practices but resisted changing others. In the process, all projects created nonprofit foundations to serve as boundary organizations. Creating these entities forced communities and firms to confront their interests and adapt their organizing practices with respect to four domains: governance, membership, ownership, and control over production. Changes in organizing practices in these domains also shaped a new triadic role structure that included communities, their nonprofit foundations, and firms.

Governance. The need to manage relations with outsiders more formally led to more defined governance mechanisms on all projects. A governance system outlines the rights available to members of an organization, their modes of representation, and the structures and processes that conduct power and resources (Davis, 2005). On an open-source project, a governance system also shapes how projectwide decisions are made and how representation is achieved (Markus, 2007). Prior to creating a governance structure, community projects relied on informal, de facto leadership and ad hoc problem solving. As Rodney, one of the founders of the Webserver Project, recalled, "With [Alpha] getting involved and wanting to figure out what the [project's] structure was, we realized that we needed to kind of solidify our processes a bit and put some formalism to it."

Members were keenly aware of the vulnerabilities of "governance by personality" and wanted to establish a permanent institution so that projects could live beyond the efforts of individuals. As Adam, a project leader on the Webserver Project, explained, "What is the on-going governance going to be for this group, and how do we make it so it lasts longer than just a few strong personalities? That is always the Linus question: 'What happens if Linus gets hit by a bus?'" Continuing with the Linus example, Adam acknowledged that Linus's lieutenant could take over, but that this could result in a "kingdom." "We wanted something much more like a republic where people didn't have to be there forever, where it wasn't seen as a regal kind of thing, a king passing along to a prince." Though project members were not eager to impose a "command and control structure onto the community," this desire for "republics" led the projects to adapt a governance structure that established project representation and preserved pluralistic control.

Establishing project representation. All projects created a governance structure that identified a point of contact to represent the project. This provided firms and outsiders with a formal communication channel. Ben, a manager in a startup contributing to the GUI Desktop Project, acknowledged that

the inability to interact privately could be a barrier for firm-community collaboration: "We can't talk to these guys without being in the public eye, therefore we cannot have exploratory conversations. Therefore, we can't do business with them, right?" Corporations needed an arena within which they could explore the parameters of both commercial and communal relationships.

Because firms did not want to share their product development plans publicly, some initially circumvented the community process, approaching open-source project leaders in private rather than on public mailing lists. Private "off-channel" conversations violated the open-source movement's approach, for "off-channel" discussions were not equally accessible to all members and thus weakened participation in the community's decisions. Gamma threatened a project's collective authority when it approached *other firms* to learn how to work with the community:

> Say, [Gamma] wanted to get involved in [GUI Desktop Project], right? Say the Foundation had existed, they could have talked to the Foundation. What they did instead was they talked to particular companies like, you know, for months and tried to plan everything in private, you know without [GUI Desktop Project] itself being involved. (David, sponsored contributor, GUI Desktop Project)

Even though the GUI Desktop Project shared some clear convergent interests with Gamma, the company's approach to collaboration violated the project's ethos and undermined members' desire to collaborate with them. Like David, other informants were distressed by approaches that circumvented community processes and indicated that a boundary organization could help bridge community and commercial worlds.

With open-source software deployed in more contexts than the developers who created it ever imagined, the need to form legal agreements became more pressing. A boundary organization could provide projects with representation for legal and commercial transactions and mitigate some of their differences with firms. Sam, a founder on the Webserver Project, explained how the project's newly created foundation helped the project meet new commercial demands:

> As the project's popularity has grown, we get a lot more requests from companies that we want to use your code in this situation, we want to name our product X, we want the [project]'s involvement in this. It is for those sorts of things that a foundation really helps us to, for example, sign a contract with a company.

Establishing project representation was mutually beneficial. It helped project members solve some of the problems that ensued when firms did not approach the community directly. Firms were relieved to know who was authorized to represent the project to discuss their use of the community's code. The foundations' charters reflected both parties' recognition of this mutual benefit. For example, the GUI Desktop charter stated that the foundation would act as a "liaison with the press and corporations who want to be involved with the project" but also "voice the decisions and positions of the [GUI Desktop] project." Establishing project representation leveraged convergent interests: projects' interest in

open participation and access and firms' interest in a stable and formal organizational structure.

Preserving pluralistic control. All projects allowed qualified contributors to join their project's nonprofit foundation, giving them a say in governance and providing pluralistic control. Pluralistic control helped communities maintain their independence because it allowed many approaches, methods, or points of view when pursuing a course of action. Members decided to elect project leaders and board members directly through democratic elections. These leaders assumed responsibility for managing the project's resources and projectwide decisions and provided an enduring governance structure, but control by the plurality was not assured with the introduction of these practices. Because firms could sponsor volunteer contributors as members of the foundation, project members were concerned that a firm (via its representatives) could still acquire majority control. Two projects therefore adopted explicit terms in their charters to ensure that contributors sponsored by the same firm could not hold a majority of board seats. The Compatibility Project allocated board seats for three different classes of members: three corporate representatives, two nonprofit representatives, and four individual representatives. With this arrangement, a firm would have to gain the support of either a nonprofit organization or an individual to establish a majority. Sandy, the primary architect of the charter, was concerned that firms could still acquire controlling influence. Thus he stipulated that if three or more members from the same organization were elected to the board across all classes, one would have to step down. In Sandy's eyes, this practice preserved the project's independence: "... when I'm no longer involved in this, and the people who are involved in it today are not involved in it ... this ensures there is not only a culture but a structure for maintaining sanity and goodness."

While project members had a strong interest in maintaining plurality and independence, firms had an interest in acquiring representation in governance, which put them at odds with each other. For instance, firms involved in the GUI Desktop Project desired designated board positions; GUI Desktop Project members insisted on designing a board comprising directly elected project members:

> This is about openness and democracy and no corporate influence poisoning the whole thing, and the other half of your time, you spend with these corporate folks, right? Some of them are pretty heavy handed, some of these folks are saying things like, "if we don't have a board member, *we will not join this movement*. We must be on the board of directors. The board of directors should be composed of the company representatives." Right? This is the sort of conversation, then you go, well, you know I'm not sure that our hacker community is ready for that. (Patrick, sponsored contributor, GUI Desktop Project)

Divergent interests led the parties to negotiate over representation in governance. But firms' attempts to influence the design of the GUI Desktop's board were not effective, as an informant indicated: "[The project founder] didn't allow them [Gamma] to do some of the things they would have loved to do had they written the charter." Initially, the GUI Desktop

governance allowed firms to sponsor up to 50 percent of board members. After heated debate, this was changed to 40 percent to meet the community's interest in pluralistic control.

Tensions between maintaining pluralism and having free elections also led to debates on both the Webserver and Linux Distribution Projects. In a Webserver Project meeting, for instance, an argument broke out among members as to whether an individual's corporate affiliation should affect his ability to represent the project on the board of directors. One member pointed out, "Up to this stage, we have been coming in as independent coders, but what if seven members, a majority of the board, are from the same company?" A second responded, "We trust each other, what would we do? Force people to step down?" A third member, getting angry, asked, "Who controls the board? The members or the outside? I mean this could come about in the next few minutes!" The conflict that surfaced exposes the tension between individuals' desire to trust and recognize each other as individuals and their need to recognize that some individuals maintained corporate affiliations that could upset pluralistic control. Ultimately, both projects kept the selection of their board of directors a democratic process without regard for an individual's affiliation.

Though maintaining pluralistic control satisfied the interests of the community, it constrained firms from directly realizing their interests in acquiring control over a project's governance. As one Linux Distribution Project member noted, this frustrated firms, whose representatives felt they had "so much trouble having any kind of influence on the development and direction of Linux." Consequently, some firms still wanted a forum to provide input to a project's technical direction. Conflict over the degree of input firms could provide resulted in the creation of mechanisms that provided firms with a means to voice their opinions without providing direct technical decision-making rights. The Linux Distribution Project created a partners program to acknowledge firms that supported them and allow partners to express their opinions. The GUI Desktop Project created an advisory board for firms to voice their opinions but emphasized that firms had "no power to make decisions."

Role of the boundary organization. With respect to governance, boundary organizations helped projects and firms realize their convergent interests by creating new structures. Table 6 shows how organizing practices were adapted. With an established means of project representation, firms were relieved of the ambiguity that made it difficult for them to follow standard corporate procedures in pursuing alliances. This change also helped community members manage corporate relations in a more coordinated fashion, simultaneously reinforcing open and participatory processes. By adapting directly democratic governance procedures that accounted for firms' affiliation, communities preserved their interest in independence and pluralistic control. Although these practices did not necessarily serve firms' interest in acquiring representation in governance, they did offer firms some opportunity to exercise influence without direct control. Adapting their organiz-

Table 6

Role of a Boundary Organization in Enabling Collaboration

Organizing Practices Adapted	Interests Satisfied: Community-managed open-source software projects	Interests Satisfied: Firms
Governance		
Establishing project representation	Provides open access and participatory processes	Reduces ambiguity and provides some degree of discretion
Pluralistic control	Ensures independent and collective control without undue firm influence	Provides some voice on project direction without direct control
Membership		
Defining rights of members	Preserves individual basis of membership and independence of the community	Firms cannot gain formal rights, only sponsor contributors
Sponsoring contributors	Provides additional resources to help project improve	Offers firms a means of direct access to development process
Ownership		
Obtaining work assignment rights	Reinforces individual autonomy and independence	Ensures clear provenance of code
Developing contribution agreements	Ensures clear provenance of code	Ensures clear provenance of code
Managing code donation	Enhances technical quality and reach of the project	Improves efficiency: no separate code base to manage
Control of production		
Community control of code contribution	Allows community to preserve autonomy and independence	Sponsored contributors provide firms with visibility and access to code development
Managing technical direction	Allows community to preserve autonomy and independence	Sponsored contributors provide firms with informal influence on code development

ing practices enabled the two parties to collaborate by delineating the conditions for representation in a project's governance without threatening either party's core interests.

Membership. Firms and community projects adapted two practices around membership: defining the rights of members and of sponsoring contributors. Community members realized that collaborating with firms could advance their project's technical progress. Firms wanted to leverage open-source code in commercial markets, but community members worried that if they allowed firms to become full project members, their interest in maintaining open communities would be threatened. Thus three out of four projects held that firms could not technically collaborate with the project as "members." But firms could not integrate open-source code in commercial products without being able to affect code development. To resolve this conflict, firms gained access to the community development process by sponsoring (hiring) volunteer contributors in areas aligned with their interests. The new membership practices satisfied parties' convergent interests: firms gained a concrete way to participate in software development, and projects reinforced membership on an individual basis.

Defining the rights of members. Project members' interests in preserving their autonomy were reflected in their conceptualization of members' rights. All four projects restricted their foundations' authority over technical decisions and conferred this authority on those who worked closest to the projects: individuals typically called committers, maintainers, or developers. Members on all projects were guaranteed the right to contribute on their own terms and to submit code modifications freely. For example, the Linux Distribution Project's constitution granted individuals the right to "make any technical or non-technical decision with regard to their own work" and to run and vote in elections of leaders.

The creation of a class of members helped not only to further projects' interest in preserving individual autonomy but also ensured that only those contributors who shared the communities' values acquired a voice in governance. As one informant from the Compatibility Project explained, "We're participating just through the fact that I'm contributing and not because I joined as a member. So all those people out there on the project mailing list are resource providers. And once the foundation gets going, those people will become more formal members." Each project varied as to how "resource providers" become members (e.g., von Krogh, Spaeth, and Lakhani, 2003), but only project members gained the right to a voice in a project's governance.

Sponsoring contributors. By hiring project contributors, firms gained credibility in the community, but this credibility would be short-lived if other project members did not perceive the sponsored contributor to be independent. Community members stipulated that new sponsored contributors would have to earn their way onto the project on the same terms as volunteers. Furthermore, as Evan from the Webserver Project explained, the stature and level of access that a person earned on a project remained with the individual, not the firm:

> [Webserver Project] was not an industry consortium, right? It was a collection of individuals, so when an individual [Alpha] engineer got core commit access, if that individual left [Alpha] and went somewhere else to work on [the project], they would still have the same status within [the project]. And [the firm] would have to get someone else.

A firm's ability to dedicate developers to a project was generally welcomed, but some project members were concerned about corporations "taking over." When Alpha wanted to have a team of developers join the Webserver Project all at once, project members initially rebuffed the idea: "The word back [from project members] was, 'you start giving us code that is meaningful and significant. We don't want a lot of your people contributing code, because that will overwhelm the group.'" Realizing that independence was a strongly held community interest, firms such as this one adapted their approach to sponsor a smaller number of contributors to a project and grant them greater independence from the firm. When the first author asked a sponsored contributor on the Compatibility Project if he represented the company he worked for, he explained, "No, and in fact, when I was

employed, it was kind of stated to me that I'm expected to be kind of independent from them."

Although sponsored contributors reinforced their independence from their employers, to do so effectively required their employers to adapt as well. To reduce the threat firms posed to a project's independence, Fortune 500 firms deemphasized their size and power, initially limiting contact with community projects to key individuals who might win the community's respect.

> We sent out a company-wide memo before we announced [our open source initiative] saying that any [Webserver Project] staff had to go through [one person], initially it was one person that was allowed to contribute. Then while we were negotiating [with the Webserver Project] we got a good feeling that we were going to get an OK, and we got an OK to have one guy start contributing. (Sponsored contributor from Alpha to Webserver Project)

By adapting their approach, firms demonstrated their willingness to work with community projects on the community's terms in hopes of gaining the community's trust.

To fulfill their interests in achieving a presence on open-source projects, firms tried to use sponsored contributors to claim membership, often expecting them to represent the firm, despite granting them "independence," but members resisted. For example, one firm initially wanted to claim credit for any contributions made by its contributors. A sponsored contributor recalled how this conflict played out in the seemingly innocuous e-mail signature file: "[Beta] had people working at [the project] attaching an e-mail disclaimer to their work 'donated by [Beta].' I told them I was not going to attach that. Now they don't do that anymore." Because e-mail was the dominant communication medium on an open-source project, such an attachment could be a salient reminder of an individual's firm affiliation and threaten project members' interest in autonomy and independence.

Role of the boundary organization. By creating a boundary organization that carefully delineated membership rights for individuals, projects and firms found a way to attenuate their divergent interests and leverage their convergent interests. By insisting that membership remain on an individual basis, projects preserved their independence and autonomy. Sponsoring contributors provided firms with access to the community's software development process. Contributing independent technical talent offered firms a way to collaborate with communities and pursue their interest in building commercial markets, but it also occasioned changes in another organizing practice: contributors' rights to the intellectual property they created.

Ownership. Creating new mechanisms for collaboration among community projects and firms occasioned the adaptation of three types of ownership practices: obtaining work assignment rights, developing contribution agreements, and managing code donations.

Obtaining work assignment rights. Firms that sponsored open-source contributors soon discovered that they needed to reconsider how to manage any intellectual property their

employees generated. Typically, programmers signed a "Confidentiality and Invention Assignment Agreement" that transferred ownership of all works created on company premises using company resources to the company. Firms had an interest in seeing that their employees' efforts were devoted to creating intellectual property for the firm, but such an agreement undermined an open-source project's interests in preserving open access and maintaining independent, collective ownership. Thus, to contribute to an open-source project, sponsored open-source contributors found themselves renegotiating their work assignment rights agreements with their employers. These negotiations were often initiated by the sponsored contributors who moved between community and corporate worlds. As one leader in the Linux Distribution Project explained, "There was just a standard paper that everyone was supposed to sign. And I looked at it and said, 'No I am not going to sign.' And we changed like five words. And basically it was adding an 'except for Linux.'" Modifying work assignment agreements was a departure for an industry that traditionally considered intellectual property its principal asset and a source of competitive advantage. Firms had to release their employees from any claim of ownership over the intellectual property they generated in order for individuals to contribute to an open-source project. By allowing sponsored contributors to retain the rights to their work, firms also helped further communities' interests in preserving open access to their code. This was a necessary precondition for the second ownership practice, developing contribution agreements.

Developing contribution agreements. Open-source projects were not interested in receiving code contributions that could be subject to a firm's claim of ownership. Competing ownership claims could threaten the provenance of the code. This was a concern for both parties: project members worried about lawsuits from inadvertent copyright violations, while firms were equally worried about their potential liability to customers. Thus projects drafted "Contributor Agreements" that assigned an individual's copyright to the project's nonprofit foundation, most often nonexclusively. In some cases, these agreements became a condition for receiving membership, assuring both firms and community members of the integrity of the intellectual property a project received. Although the parties had different motivations for developing contributor agreements, such agreements served the interests of both parties.

Project leaders were unsure if these agreements adequately assured the provenance of an individual's contribution. At one Webserver Project meeting, members debated asking sponsored contributors to submit an ownership disclaimer from their employers. Some members protested that this type of oversight violated the community's culture and values. A contributor agreement was a legal relationship between an individual and the project's foundation. To invite firms into this relationship by requiring an ownership disclaimer from the sponsoring firm would weaken membership terms rooted in individuality. The idea was rejected, leaving the project to

trust that individuals owned the rights to their work and were free to assign them to the project.

Managing code donations. Firms could also advance a project by making direct donations of code that had been previously developed internally, but firms were reluctant to donate code to a project without also transferring responsibility for it, and this was impossible to accomplish without a legal owner. The establishment of a foundation gave firms a legal entity to which ownership could be transferred. As Henry, a member on the Webserver Project, explained, "When [Gamma] and [Alpha] donated code to us, they signed contracts that said we sign over copyright . . . [and] we can consider that our code. And thus the [project] Foundation is liable for it."

Firms had several different motivations for donating code to open-source projects. In one instance, a firm wanted to accelerate development on a piece of software that would be incorporated into its commercial product. In another case, a firm wanted to donate a piece of code that was no longer of commercial value but could potentially help the project. The majority of code donations furthered firms' interests in maintaining an integrated code base so that firms did not have to expend additional effort to maintain a parallel but separate code base. Thus the ability to make "official" code donations advanced firms' interests in an efficient and stable software production process:

> [The firm] didn't want to maintain a different set of patches from the normal [project] code base. . . . Remember the ultimate goal is to sell more software. That is why [the firm] did this. They didn't do it out of good faith . . . and the closer they could keep their version of [the code to the project], they could sell it that much easier. (Marshall, sponsored contributor, GUI Desktop Project)

As Marshall indicated, project members were aware of the pragmatic motivations that drove code donations but still welcomed such donations, as they satisfied their interest in improving the code's reliability and performance.

Role of the boundary organization. New ownership practices had two implications for relations among community-managed projects, their boundary organizations, and firms. First, changes in employees' work assignment rights reinforced sponsored contributors' autonomy and helped solidify their role as legitimate, independent contributors to the project. This simultaneously furthered firms' interest in gaining voice on a project and communities' interest in preserving independence. Second, contributor agreements allowed open-source projects to collectively hold the intellectual property that both individuals and firms created. Taken together, the adaptation of these ownership practices leveraged communities' and firms' shared goals of improving the code and addressed their convergent concerns about the provenance of the code.

Control of production. The only way open-source project members were willing to support the idea of creating a nonprofit foundation was by ensuring that its role in controlling software development was limited. By restricting the foundation's role to primarily legal and administrative domains, as opposed to technical domains, project members furthered

their interests in autonomy and independence. At the same time, firms wanted stable and predictable code development and therefore were interested in using boundary organizations to influence the project's technical direction. As one leader on the GUI Desktop Project explained:

> The corporations, they want this industry consortium that they can run and control and they can do their corporate politics in. . . . that's what you want if you're a company. You want a standard-setting body and you want to send all your folks to it and you want to try to control the thing as much as you can. The hackers, they want happy hacker habitat . . . if you look at the first draft of the charter that [X] wrote, it was like we will have a virtual worldwide consortium of hackers.

Though boundary organizations helped attenuate some of these differences with respect to governance, membership, and ownership, of the four domains, establishing control over production triggered the most contestation. This conflict manifested in two primary arenas: community control of code contributions and managing the technical direction of the projects.

Community control of code contributions. Project members insisted that their foundations should not have a role in guiding code-level decisions. Informants believed that the right to accept or reject code contributions into the code repository belonged to individual members alone. The key criteria for accepting code were the code's technical merit, its ability to help the project progress, and its impact on other project subunits. Although the perceived experience and expertise of an individual might affect how a code contribution was received, all decisions were made at the will of the community. Thus technical autonomy, as Marty, a sponsored contributor to the Webserver Project pointed out, was negotiated through interactions with one's peers: "Most of the group will acknowledge that I have done a good portion of that work. That has been the last two years of my life full time plus some [work] that I have done. But I still have to defend what I want to do. If I want to do it and everybody goes, 'No you are just wrong,' then it doesn't go into the code."

Community members resented foundations' efforts to control code contributions. Roger, a contributor to the GUI Desktop Foundation, expressed his frustration when he thought the foundation overstepped its limits: "Is this [foundation] supposed to run [GUI Desktop Project]? People say, no, and people still say no. [GUI Desktop Project] is run by the maintainers of the packages. . . . If you're the maintainer of a package, you own a package." Code contributions that were customized for a firm's product endangered communities' interest in independence. The more firm-specific the contribution was perceived to be, the more likely that project members would regard it with criticism or suspicion. As Tony, a sponsored contributor on the Linux Distribution Project, noted, firm-specific contributions did not often meet with wholehearted acceptance:

> If it's a good change and somebody did the work, [the Linux Distribution Project] would be happy to take it. It would only be a case [for not accepting code] when the company says "[use] this specific

library," and we'd say "that's nasty" or when the whole system is specific to your company and is not going to help others and it will affect other systems.

Sponsored contributors sensitive to communities' interest in independence found a way around this problem by contributing code that was not specific to their firm. For instance, sponsored contributors from one firm deliberately expanded their work to areas not related to their firm's product to avoid the perception of focusing only on firm-specific technical goals. This "good citizenship" in the community furthered perceptions of independence while simultaneously helping sponsored contributors accrue influence. As Anand commented, "They were so skeptical of us here; it could take a little while. We had to give them a little: 'Okay, these guys aren't so bad, and they are doing cool things. They are going to work on things that we care about.' So we were really trying to also fix some bugs that were normal bugs, not just [Beta] bugs." Several sponsored contributors in this firm used this tactic early in their relationship with the project to gain trust, support the firm's interest in influencing the project's development, and meet the community's interest in preserving the independence of contributing members.

Managing technical direction. Control of production also included the strategic direction of a project, including changes in a project's architecture, its "release" content, and timing. A release is a full version of software that is made available to the public at a point when developers feel that it can be safely used by others. Release coordination was typically managed by domain experts, individuals not in foundation leadership roles. This further reinforced individuals' autonomy and limited the foundation's control over the project's technical direction. For example, the Linux Distribution Project limited the power of its foundation and framed its role as serving the needs of the project:

> [The foundation] itself is just sort of in charge of ideals and assets rather than a project directly. That is the intent, that a project can just say "Look, we need this kind of an organization behind us, but we don't want them telling us no, you are not going to release right now, you are going to release in two months," because it makes more sense, you know. You have to let the projects develop how they want to develop. (Jason, volunteer contributor, Linux Distribution Project)

The GUI Desktop Project was the only project to explicitly grant release authority to its foundation, as its charter stated: "The foundation bears the responsibility of coordinating each subsequent release of [GUI Desktop Project]. For each release, this will include setting a schedule (whether or not it is overlooked), choosing the set of modules which are a part of the release, and preparing the appropriate marketing materials." Because of the foundation's role in coordinating releases, members on this project had a more difficult time guarding their individual autonomy. Some members contested what they saw as too active a role for the foundation. They viewed the foundation's role in release coordination as disrespectful of the limits of its authority and a violation of community interest in independent development. As Jackson, a volunteer contributor to the GUI Desktop Project, explained,

"I mean, fair enough, you want to have an organization that can say, 'This is [GUI Desktop Project]' or 'That's not [GUI Desktop Project].' But they try to get into strategy, and they try to produce a vision and it's not really their place to do . . . they're interfering in technical matters."

Although a foundation's formal control was limited, there was room in the structure of the boundary organization for firms to informally influence technical direction through the work of sponsored contributors. To gain this influence, however, firms could not rely on hierarchical authority but had to rely on the code contributions of sponsored contributors. George, a contributor from Beta, explained that "big companies need to learn this when you start working with an open-source group. You are not dictating how everything works. You are making suggestions, and you are contributing the manpower to make it work the way you want it to."

Sponsored informants admitted to shifting their priorities by working on aspects of the project that were most important to the firm. Although they recognized that their employer's wishes could affect the project's progress, they did not perceive this to threaten a project's independence. Francis a sponsored contributor and board member of the GUI Desktop Project said, "They [Beta] don't have any real influence on what [the program] will do as far as what features it will have or when it will be released. But what they can do is that [they] can ask, on company time, of course, that we work on one thing or another thing because that's the thing that they would like to see move forward." He went on to explain that the public nature of the project helped countervail the ability of any one firm to influence the project, because the project goals were made by the community and discussed in public: "And so that's not controlled by any company, but if there's ten features that we want and some company actually implements one of them and no one implements the other nine, that one will be the one that got implemented, right?" Thus what firms could influence was the level of effort and the ordering of priorities on a project. Shifting priorities could hasten a project's progress in a direction desired by a firm, but because all community members wanted to see their project advance to commercial grade, this type of influence leveraged convergent interests and was not perceived to constrain communities' interest in independence.

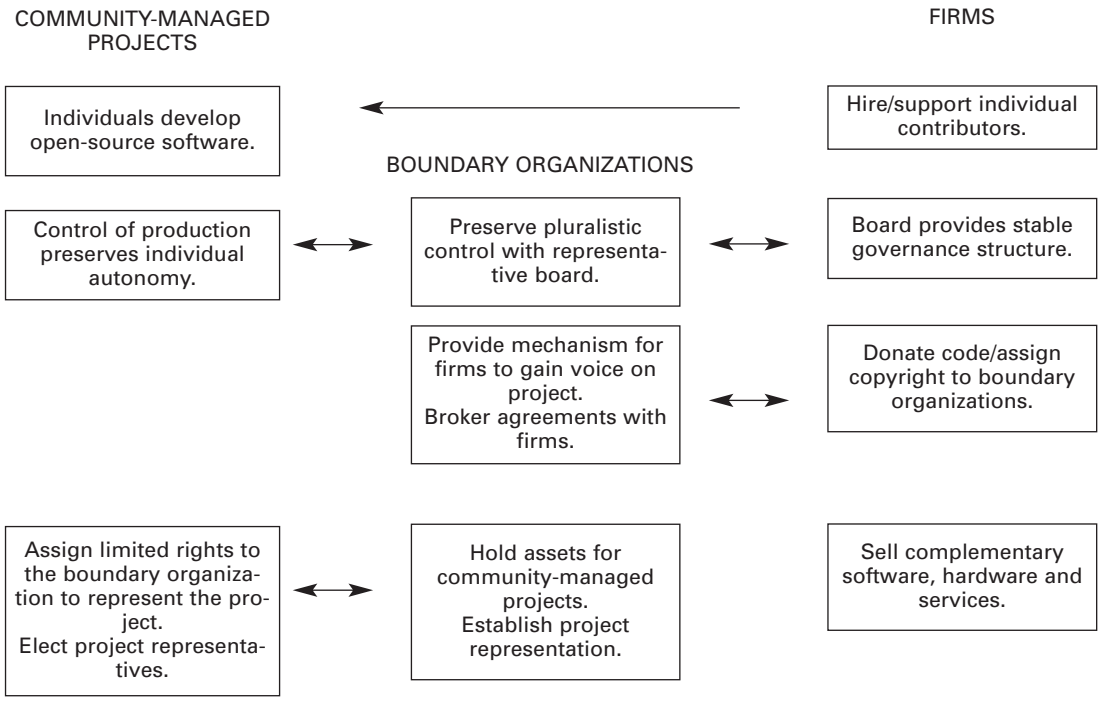
Role of the boundary organization. Boundary organizations provided some structure for controlling production but were minimally involved in the hands-on aspects of software development. Project leaders recognized that fostering the technical autonomy of volunteers was critical to sustaining members' commitment to the project. Unfettered influence by firms on a project's technical direction could diminish community members' motivation for contributing to the project. By limiting a boundary organization's ability to control production, communities furthered their interest in individual autonomy and independent software development. Boundary organizations also fostered firms' interests in maintaining a predictable software development process. Some firms would have liked boundary organizations to take a more active role in coordinating releases, and although a few firms

attempted to satisfy this interest, they did not have the power to make it happen. Instead, they pursued their interests by informally influencing projects' technical direction through sponsored contributors.

A triadic role structure. Figure 1 summarizes how adaptation across the four domains resulted in a new triadic role structure among community-managed open-source projects, boundary organizations, and firms. Boundary organizations enabled lasting collaboration between unexpected allies: open-source community projects and commercial firms. As Jack, a volunteer contributor to the Webserver Project, put it, "I think to a large extent, at least as far as I can tell, we have created an organization that can live with the community and the community can live with it and work together towards maintaining our software over a long period." Boundary organizations did not resolve all conflict, and divergent interests remained. Not every contributor shared the same level of enthusiasm for commercial support. Yet informants spoke of this transformation in ways that suggested they had settled into a productive working relationship. As Doug, a sponsored contributor on the Linux Distribution Project, said, "I think that it has happened gradually enough that people have had time to get used to the idea. There certainly used to be a lot of friction, there are still people that think commercialism is bad. At least all the developers I talk to are fairly happy with commercial efforts now."

Boundary organizations enabled collaboration between open-source communities and firms by building on convergent interests while providing space that preserved the divergent interests of the parties. When participants in intersecting worlds collaborate, ". . . resolution does not mean consensus. Rather, representations . . . contain at every stage the

Figure 1. The triadic role structure in open-source software development.



traces of multiple viewpoints, translations, and incomplete battles" (Star and Griesemer, 1989: 413). In our study, neither party compromised or changed their interests; adapting each organizing practice partially fulfilled particular interests. Both parties adapted a selective set of organizing practices around boundary organizations that enabled them to continue to pursue divergent interests.

DISCUSSION

Our research examined how challengers of an established system, open-source community projects, and defenders of that same system, firms in the proprietary software development tradition, collaborated despite having divergent goals. Observers and social movement theorists rarely focus on the potential for mutual gains among parties with divergent goals. As our research shows, however, it is possible for challenging and defending parties to pursue divergent goals but also isolate a subset of convergent interests and create new terms for collaboration. One informant from a Fortune 500 firm noted as much when he called his firm "an unexpected ally" in the open-source movement. Yet even when convergent interests are easily identified, making such a collaboration work remains a challenge—it requires both parties to adapt their organizing practices.

The parties had very different ideas about what a boundary organization should look like: either a "happy hacker habitat" or an "industry consortium for corporate politics." Neither option would mediate their differences: the former implied no adjustment on the part of communities; the latter implied no adjustment on the part of firms. Instead, both open-source projects and firms adapted their organizing practices to realize convergent interests. By tracing the changes that both parties made, we showed how differences that could have prevented collaboration were settled. We identified specific practices with respect to governance, ownership, membership, and control over production that ameliorated differences and helped preserve each party's distinct interests. Our findings revealed that open-source community projects, their foundations, and firms developed a triadic role structure. What is counterintuitive is that boundary organizations enabled collaboration not by blurring boundaries but by reinforcing convergent interests and articulating how interests diverged. The job of a boundary organization was not to collapse or merge divergent worlds but to preserve each world's integrity while building a bridge between them. Only by preserving the boundaries that separated the two parties could boundary organizations sustain their ability to represent either party.

Reconceptualizing Social Movement Processes and Outcomes

Our research elaborates on the recent work of scholars at the juncture of organizational and social movement theory (Clemens, 2005; McAdam and Scott, 2005; Davis and McAdam, 2000). Scholars have examined how social movements arise inside firms (Scully and Segal, 2002) and effect organizational and institutional change (Thompson and Davis,

1994; Zald, Morrill, and Rao, 2005). Yet a narrow conception of social movement outcomes (Giugni, 1998; Andrews, 2001) and a focus on macro-social dynamics have prevented this dialogue from realizing its promise. What is not well understood is how actors with seemingly opposing goals discover mechanisms to integrate their conflict while sustaining divergent interests. To understand the outcomes that are possible when challengers and defenders of established systems collaborate, we must go beyond narrowly defined policy goals to conceptualize how mutual adaptation occurs and how subsequent social relations are affected.

Our findings help reconceptualize both social movement processes and outcomes. While social movement theory offers a solid basis for understanding the dynamics of contention, there is a lack of understanding of the process by which social movements move from contention to collaboration. In their comparative social movement research, McAdam, Tarrow, and Tilly (2001: 143) identified several mechanisms that enable mobilization, including brokerage, which "creates new boundaries and connections among political actors." Although brokerage explains how actors connect with one another, it falls short of explaining how they might ally: brokerage alone would not lead parties to adapt their organizing practices, integrate their conflict, and transform their role structure.

Our research expands the repertoire of social movement mechanisms by specifying the mechanisms enabling the transformation from contention to collaboration. In the process of creating boundary organizations, parties confronted areas in which their interests converged and diverged. Hargrave and Van de Ven (2006) argued that when opposing actors confront each other, they must create new working rules or organizational arrangements to work together. We showed more specifically how constructing a boundary organization triggered community members and firms to confront and articulate their interests. Participants were aware that the organizing decisions they made when forming boundary organizations had lasting consequences. Thus the durable nature of boundary organizations served as a catalyst for delineating interests, reinforcing the boundaries of each world, and remapping the organizing practices of both parties.

We also articulated an outcome—collaboration among unexpected allies—that lies beyond the narrow definition of outcomes in the literature (Giugni, 1998; Earl, 2000, 2004; Amenta and Caren, 2004). Recently, theorists have argued that we must move beyond exploring outcomes internal to the movement to examine broader, external cultural and institutional outcomes (Earl, 2000, 2004). Yet these external outcomes, while as important as traditional policy goals, can be more difficult to measure. Earl (2000: 8) pointed to a danger in sticking to traditional social movement measures: "valuable information can be lost by focusing on changes in degree and ignoring changes in form."

Our own study demonstrates the rewards of taking on the difficult task of examining a change in form. We showed how

the creation of a boundary organization triggered the adaptation of organizing practices. These outcomes were a precondition for a change in social relations among challenging and defending parties. When we revisit studies of social movement organizations, we find evidence of this collaborative form—boundary organizations—at work in the civil rights movement (McAdam, 1982), the pro-choice movement (Staggenborg, 1988), and in public-private partnerships furthering social welfare goals (Austin, 2000). Our research indicates that boundary organizations may be a vital mechanism for preserving a social movement's mission, intent, and form while fostering the absorption of its message into other elements of society. By showing how this outcome unfolded, we developed a theoretical explanation of how organizations mediate the impact of social movements and expand the range of social movement outcomes that lie between the acquisition of policy changes and cooptation.

By confronting divergent interests in the creation of a boundary organization, participants became acutely aware of the balancing and bridging functions of the boundary organization. We uncovered these mechanisms only by examining the micro-social processes of collaboration. Social movement scholars have lamented the fact that they have too often turned off their cameras when protesters leave the street.[2] As a result, we don't know what happens when challenging and defending parties meet at the table inside or the outcomes that can emerge. Exploring the process by which boundary organizations were created therefore helps social movement and organizational scholars develop a framework for understanding not only how contestation is mobilized and diffused but how it transforms social relations.

Boundary Organizations: Specifying Mechanisms for Collaboration

Organizational scholars have shown how boundary work (Gieryn, 1983; Nelsen and Barley, 1997), boundary-spanning roles (Tushman, 1977; Podolny and Stuart, 1995), and boundary objects (Carlile, 2002; Bechky, 2003a, 2003b) help actors collaborate across different worlds, yet the role of organizations in performing boundary work has been underappreciated (Moore, 1996). Science-studies scholars have devoted more attention to the study of boundary organizations (Guston, 2000), but this research has not articulated the mechanisms that make boundary organizations work. As a result, this construct has been trapped in its context of study and its portability limited. Our analysis suggests three critical characteristics of boundary organizations that transform social relations: they trigger adaptation around key organizing domains, they delineate boundaries between convergent and divergent interests, and they provide a durable structure to reinforce mutual adaptation.

By pinpointing the four domains of organizational adaptation—governance, membership, ownership, and control of production—we provided a grounded theoretical framework for investigating the role of boundary organizations. The domains of organizing practice that were adapted in this study can help scholars articulate the role structure of other

2
According to Dick Scott (Social Movements and Organization Theory Conference at the University of Michigan, May 10–11, 2002), one thing that Doug McAdam said he would have liked to have done in his landmark studies of the civil rights movement was "not turn off my camera when people left the street."

types of partnerships, particularly those that are production-oriented. We found that control over production was the most contested domain during the course of the study and remains so for some projects. Future research should examine whether these domains are equally critical for other types of partnerships and under what conditions domains are more or less contested.

As communities and firms created boundary organizations, they differentiated the interests that could be pursued collaboratively from those that diverged. By identifying and confronting where their interests diverged, both parties more precisely clarified the boundaries that separated them. Clearly defined boundaries, coupled with both parties' adaptation of their organizing practices, not only defined the role of the boundary organization but delineated a new triadic role structure. Boundary organizations transformed social relations because they brought unexpected allies together. Decisions about the role of the foundation in the triadic role structure were thought to have lasting consequences. Parties were concerned about establishing precedent and the implications of their decisions on their practices. Thus we found that collaboration in these settings was not contingent on conjoining, cooptation, or collapse of boundaries. Instead, collaboration between open-source community projects and firms was accomplished by delineating boundaries across organizing domains to form a relatively durable boundary organization.

These three characteristics—adaptation around key organizing domains, delineation of interests, and durability of structure—help distinguish boundary organizations from other conceptual frameworks for collaboration. Boundary organizations share the interpretive flexibility of boundary objects or inscriptions (Latour and Woolgar, 1979; Star and Griesemer, 1989; Bechky, 2003a): they are flexible in use, bridging divergent worlds while preserving elements that are distinct to each. But they differ from boundary objects in their durability. Boundary objects are mobile, material representations that move from party to party in a process of enrollment or problem solving (Fujimura, 1988; Henderson, 1999; Bechky, 2003a). Because they are more durable, boundary organizations enforce a confrontation of interests that is rarely seen with boundary objects, which can be ignored, lost, or made irrelevant (Henderson, 1999; Bechky, 2003b).

The durability of boundary organizations and their instigation of change around key organizing domains also creates collaborative conditions that are different than those suggested by other scholars of science and technology. For instance, in some social worlds, such as physics (Galison, 1997) and advertising (Kellogg, Orlikowski, and Yates, 2006), collaboration occurs within trading zones. Trading zones are emergent, provisional spaces in which disparate communities meet and temporarily coordinate their activities (Galison, 1997; Kellogg, Orlikowski, and Yates, 2006). But the practices involved in a trading zone are informally structured: issues of governance and membership are not articulated, let alone formalized. In contrast, boundary organizations require participants to make lasting decisions about key organizing domains, such as gov-

ernance, which forces them to confront and delineate interests.

The Role of Boundary Organizations in Organizational Theory

The notion of boundary management (Tushman, 1977; Podolny and Stuart, 1995) has always been important to organization theory. Recently, Santos and Eisenhardt (2005) urged scholars to examine organizations' boundary decisions, expanding our attention beyond efficiency concerns to problem-driven decisions in contemporary practice and new settings. This research takes a step in this direction and contributes to organizational theory by explicating the mechanisms that enable unlikely allies to collaborate, which may be especially critical for emerging or contested fields.

Even though they have not been labeled as such nor appreciated for the role they play, boundary organizations are more prevalent than scholars may realize. For example, community-based nonprofit organizations broker federal dollars in the provision of social services (Marwell, 2004). Nonprofit organizations enable environmental activists and businesses to collaborate on projects to enhance the natural environment (Hoffman and Ventresca, 2002). Biological resource centers and technology transfer offices bridge academic and commercial science by aligning interests along dimensions similar to those in our findings (Stern, 2004; Owen-Smith, 2005; Murray and O'Mahony, 2007). In all these contexts, boundary organizations help actors with divergent goals further a subset of convergent interests. Our explanation of how such boundary organizations are formed deepens our understanding of the collaborative work involved in these contexts.

The organizing domains we identified also are relevant to both organization and social movement theory. As Star and Griesemer (1989: 398) described, for the Berkeley Museum of Vertebrate Zoology to succeed required not only "training on the part of the scientific staff; it also required changes in basic collecting and curating procedures"; thus the parties involved in the museum had to adapt their organizing practices. Similar organizing domains are also important for collaboration in traditional social movement arenas. For example, Staggenborg (1988) found that pro-choice social movement organizations were better able to coordinate their efforts when they had staff who could act as "organizational representatives." Andrews' (2001: 89) study of the civil rights movement found that "movements were most influential when they built local organizations that allowed for an oscillation between mass-based tactics and routine negotiation with agency officials."

A lack of comparative research has left a theoretical gap in our understanding of how convergent interests can be leveraged despite the presence of divergent interests. By bringing the concept of boundary organizations to the nexus of social movement and organizational theory, we begin to fill that gap. Our research, which compares across four cases, shows more precisely what it is that boundary organizations do—they provide an enduring organizational bridge across different worlds by requiring adjustment of practices in the

domains of governance, membership, ownership, and control of production, but without requiring the parties to concede on their divergent interests. This contribution is particularly important given the rise of social movements that use modes of opposition that do not rely on contestation (e.g., Calhoun, 1995; Earl, 2004).

Future Research

This research was initiated to investigate how open-source projects responded to commercial attempts to collaborate. In the course of our research, we discovered that adapting organizing practices and creating boundary organizations was important to establishing a working relationship with companies on all four projects. These boundary organizations were successful only in enabling collaboration to happen. We did not examine differences in the operating success of the boundary organizations under study, primarily because they had just been created. Yet it is likely that the operating success of boundary organizations may vary over time, and the conditions determining such success would be a fruitful area for future study.

Boundary organizations do not resolve all conflict. In our study, they enabled collaboration between social movements and organizations, but different ideas about the degree of control that boundary organizations should assume persisted. In addition, many aspects of the commercial licensing of open-source software remain contested and controversial terrain. Future research can help us understand what types of divergent interests cannot be adjudicated with the help of a boundary organization. It may be that a legal, neutral ground is necessary to delineate and take into account divergent interests and that nonprofit organizations are well suited to the boundary organization role (e.g., Powell and Clemens, 1998). But not all forms of collaboration among challenging and defending parties may require the creation of an enduring organization, let alone a nonprofit organization. Future research should investigate whether other types of organizational forms can become boundary organizations that serve collaborative goals.

If social movements are to be a more recognized part of organizational life, then it is imperative that we develop a framework for understanding not only how contestation is mobilized but how it is transformed. Otherwise, we will have an incomplete understanding of how cycles of contestation ebb and flow, for contestation is not likely to remain constant. By explaining how productive working relationships can be created despite divergent interests, we expand our understanding of the outcomes of social movement activity—from policy change and resource distribution to the redrawing of organizational boundaries and the reshaping of social and production relations. These types of outcomes may produce societal changes of equal if not greater import.

REFERENCES

Allen, R. C.
1983 "Collective invention." Journal of Economic Behavior and Organization, 4: 1–24.

Amenta, E., and N. Caren
2004 "The legislative, organizational, and beneficiary consequences of state-oriented challengers." In D. A. Snow, S. A. Soule, and H. Kriesi (eds.), The Blackwell Companion to Social Movements: 461–488. Malden, MA: Wiley-Blackwell.

Andrews, K. T.
2001 "Social movements and policy implementation: The Mississippi civil rights movement and the War on Poverty, 1965 to 1971." American Sociological Review, 66: 71–95.

Austin, J.
2000 The Collaboration Challenge: How Nonprofits and Businesses Succeed through Strategic Alliances. San Francisco: Jossey-Bass.

Bechky, B. A.
2003a "Object lessons: Workplace artifacts as representations of occupational jurisdiction." American Journal of Sociology, 109: 720–752.
2003b "Sharing meaning across occupational communities: The transformation of knowledge on a production floor." Organization Science, 14: 312–330.

Calhoun, C.
1995 "'New social movements' of the early nineteenth century." In M. Traugott (ed.), Repertoires and Cycles of Collective Action: 173–216. Durham, NC: Duke University Press.

Carlile, P.
2002 "A pragmatic view of knowledge and boundaries: Boundary objects in new product development." Organization Science, 13: 442–455.
2004 "Transferring, translating, and transforming: An integrative framework for managing knowledge across boundaries." Organization Science, 15: 555–568.

Carroll, G. R., and A. Swaminathan
2000 "Why the microbrewery movement? Organizational dynamics of resource partitioning in the US brewing industry." American Journal of Sociology, 106: 715–762.

Clemens, E. S.
1993 "Organizational repertoires and institutional change: Women's groups and the transformation of U.S. politics, 1890–1920." American Journal of Sociology, 98: 755–798.
1997 The People's Lobby: Organizational Innovation and the Rise of Interest Group Politics in the United States, 1890–1925. Chicago: University of Chicago Press.
2005 "Two kinds of stuff: The current encounter of social movements and organizations." In G. F. Davis, D. McAdam, W. R. Scott, and M. N. Zald (eds.), Social Movements and Organization Theory: 351–366. New York: Cambridge University Press.

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

Dalle, J. M., and N. Julien
2003 "'Libre' software: Turning fads into institutions?" Research Policy, 32: 1–11.

Davis, G. F.
2005 "Firms and environments." In N. J. Smelser and R. Swedberg (eds.), Handbook of Economic Sociology, 2d ed.: 478–502. Princeton, NJ: Princeton University Press.

Davis, G. F., and D. McAdam
2000 "Corporations, classes, and social movements after managerialism." In B. M. Staw and R. I. Sutton (eds.), Research in Organizational Behavior, 22: 193–236. New York: Elsevier/JAI.

DiMaggio, P. J., and H. K. Anheier
1990 "The sociology of nonprofit organizations and sectors." Annual Review of Sociology, 16: 137–159.

Earl, J.
2000 "Methods, movements, and outcomes: Methodological difficulties in the study of extra-movement outcomes." In P. G. Coy (ed.), Research in Social Movements, Conflicts, and Change, 22: 3–25. Stamford, CT: JAI Press.
2004 "The cultural consequences of social movements." In D. A. Snow, S. A. Soule, and H. Kriesi (eds.), The Blackwell Companion to Social Movements: 508–530. Malden, MA: Wiley-Blackwell.

Frickel, S., and N. Gross
2005 "A general theory of scientific/intellectual movements." American Sociological Review, 70: 204–232.

Fujimura, J. H.
1988 "The molecular biological bandwagon in cancer research: Where social worlds meet." Social Problems, Special Issue: The Sociology of Science and Technology, 35 (3): 261–283.

Galison, P.
1997 Image and Logic: A Material Culture of Microphysics. Chicago: University of Chicago Press.

Gamson, W. A.
1975 The Strategy of Social Protest. Homewood, IL: Dorsey.

Ganz, M.
2000 "Resources and resourcefulness: Strategic capacity in the unionization of California agriculture, 1959–1966." American Journal of Sociology, 105: 1003–1062.

Gieryn, T. F.
1983 "Boundary-work and the demarcation of science from non-science: Strains and interests in professional interests of scientists." American Sociological Review, 48: 781–795.
1999 Cultural Boundaries of Science: Credibility on the Line. Chicago: University of Chicago Press.

Giugni, M. G.
1998 "Was it worth the effort? The outcomes and consequences of social movements." American Sociological Review, 98: 371–393.

Guston, D. H.
1999 "Stabilizing the boundary between US politics and science: The role of the Office of Technology Transfer as a boundary organization." Social Studies of Science, 29: 87–111.

2000 Between Politics and Science. Assuring the Integrity and Productivity of Research. Cambridge: Cambridge University Press.
2001 "Boundary organizations in environmental policy and science: An introduction." Science, Technology and Human Values, 26: 399–408.

Hargrave, T. J., and A. H. Van de Ven
2006 "A collective action model of institutional innovation." Academy of Management Review, 31: 864–888.

Hars A., and S. Ou
2001 "Working for free?—Motivations of participating in open source projects." Proceedings of the 34th Hawaii International Conference on System Sciences.

Henderson, K.
1999 On Line and on Paper: Visual Representations, Visual Culture, and Computer Graphics in Design Engineering. Cambridge, MA: MIT Press.

Hertel, G., S. Niedner, and S. Hermann
2003 "Motivation of software developers in the open source projects: An Internet-based survey of contributors to the Linux kernel." Research Policy, 32: 1159–1177.

Hoffman, A. J.
1999 "Institutional evolution and change: Environmentalism and the U.S. chemical industry." Academy of Management Journal, 42: 351–371.

Hoffman, A. J., and M. Ventresca
2002 Organizations, Policy, and the Natural Environment: Institutional and Strategic Perspectives. Palo Alto, CA: Stanford University Press.

Kellogg, K. C., W. J. Orlikowski, and J. Yates
2006 "Life in the trading zone: Structuring coordination across boundaries in post-bureaucratic organizations" Organization Science, 17: 22–44.

Lakhani, K. R., and E. von Hippel
2003 "How open source software works: Free user to user assistance." Research Policy, 32: 923–943.

Lakhani, K. R., and R. Wolf
2005 "Why hackers do what they do: Understanding motivation and effort in free/open source software projects." In J. Feller, B. Fitzgerald, S. Hissam, and K. Lakhani (eds.), Perspectives in Free and Open Source Software: 3–22. Cambridge, MA: MIT Press.

Latour, B.
1987 Science in Action: How to Follow Scientists and Engineers through Society. Cambridge, MA: Harvard University Press.

Latour, B., and S. Woolgar
1979 Laboratory Life: The Construction of Scientific Facts. Beverly Hills, CA: Sage.

Lounsbury, M.
1997 "Exploring the institutional toolkit: The rise of recycling in the U.S. solid waste field." American Behavioral Scientist, 40: 465–477.
2005 "Institutional variation in the evolution of social movements: The spread of recycling advocacy groups." In G. F. Davis, D. McAdam, W. R. Scott, and M. N. Zald (eds.), Social Movements and Organization Theory: 73–95. Oxford: Oxford University Press.

Lounsbury, M., H. Geraci, and R. Waismel-Manor
2002 "Policy discourse, logics, and practice standards: Centralizing the solid-waste management field." In A. J. Hoffman and M. J. Ventresca (eds.), Organizations, Policy, and the Natural Environment. Institutional and Strategic Perspectives: 327–345. Stanford, CA: Stanford University Press.

Lounsbury, M., M. Ventresca, and P. M. Hirsch
2003 "Social movements, field frames and industry emergence: A cultural-political perspective on US recycling." Socio-Economic Review, 1: 71–104.

Markus, M. L.
2007 "The governance of free/open source software projects: Monolithic, multidimensional, or configurational?" Journal of Management and Governance, 11 (2): 151–163.

Marwell, N. P.
2004 "Privatizing the welfare state: Nonprofit community-based organizations as political actors." American Sociological Review, 69: 265–291.

McAdam, D.
1982 Political Process and the Development of Black Insurgency 1930–1970. Chicago: University of Chicago Press.
1983 "Tactical innovation and the pace of insurgency." American Sociological Review, 48: 735–754.

McAdam, D., and R. Paulsen
1993 "Specifying the relationship between social ties and activism." American Journal of Sociology, 99: 640–667.

McAdam, D., and W. R. Scott
2005 "Organizations and movements." In D. McAdam and W. R. Scott (eds.), Social Movements and Organization Theory: 4–40. New York: Cambridge University Press.

McAdam, D., S. Tarrow, and C. Tilly
2001 Dynamics of Contention. New York: Cambridge University Press.

McCarthy, J. D., and M. N. Zald
1977 "Resource mobilization and social movements: A partial theory." American Journal of Sociology, 82: 1212–1241.

Miller, C.
2001 "Hybrid management: Boundary organizations, science policy, and environmental governance in the climate regime." Science, Technology and Human Values, 26: 478–500.

Moore, K.
1996 "Organizing integrity: American science and the creation of public interest organizations, 1955–1975." American Journal of Sociology, 101: 1592–1627.

Mowery, D., and A. Ziedonis
2002 "Academic patent quality and quantity before and after the Bayh-Dole Act in the United States." Research Policy, 31: 399–418.

Murray, F.
2007 "The Oncomouse that roared: Resistance and accommodation of patenting in academic science." Unpublished manuscript, MIT Sloan School of Management.

Murray, F., and S. O'Mahony
2007 "Exploring the foundations of cumulative innovation: Implications for organization science." Organization Science, 18: 1006–1021.

Nelsen, B. J., and S. R. Barley
1997 "For love or money? Commodification and the construction of an occupational mandate." Administrative Science Quarterly, 42: 619–653.

O'Mahony, S.
2007 "What does it mean to be community managed?" Journal of Management and Governance, 2: 139–150.

O'Mahony, S., and K. Chen
2007 "Selective synthesis of competing logics." Unpublished manuscript, Graduate School of Management, University of California, Davis.

O'Mahony, S., and F. Ferraro
2007 "Managing the boundary of an open project." Academy of Management Journal, 50: 1079–1106.

Owen-Smith, J.
2003 "From separate systems to a hybrid order: Accumulative advantage across public and private science at research one universities." Research Policy, 32: 1081–1104.
2005 "Dockets, deals, and sagas: Commensuration and the rationalization of experience in university licensing." Social Studies of Science, 35: 69–97.

Owen-Smith, J., and W. W. Powell
2003 "The expanding role of university patenting in the life sciences: Assessing the importance of experience and connectivity." Research Policy, 32: 1695–1711.

Piven, F. F., and R. A. Cloward
1977 Poor People's Movements: Why They Succeed, How They Fail. New York: Pantheon Books.

Podolny, J., and T. E. Stuart
1995 "Role-based ecology of technological change." American Journal of Sociology, 100: 1224–1260.

Pollan, M.
2007 The Omnivore's Dilemma: A Natural History of Four Meals. New York: Penguin Books.

Powell, W. W., and E. S. Clemens (eds.)
1998 Private Action and the Public Good. New Haven, CT: Yale University Press.

Powell, W. W., K. W. Koput, and L. Smith-Doerr
1996 "Interorganizational collaboration and the locus of innovation: Networks of learning in biotechnology." Administrative Science Quarterly, 41: 116–145.

Powell, W. W., D. R. White, K. W. Koput, and J. Owen-Smith
2004 "Network dynamics and field evolution: The growth of inter-organizational collaboration in the life sciences." American Journal of Sociology, 110: 1132–1205.

Rao, H.
1998 "Caveat emptor: The construction of nonprofit consumer watchdog organizations." American Journal of Sociology, 103: 912–961.

Rao, H., C. Morrill, and M. N. Zald
2000 "Power plays: How social movements and collective action create new organizational forms." In B. M. Staw and R. I. Sutton (eds.), Research in Organizational Behavior, 22: 237–281. New York: Elsevier/JAI.

Rosen, K.
2004 Open Source Licensing: Software Freedom and Intellectual Property Law. New York: Prentice Hall.

Rosenkopf, L., A. Metiu, and V. P. George
2001 "From the bottom up? Technical committee activity and alliance formation." Administrative Science Quarterly, 46: 748–772.

Rosenkopf, L., and M. L. Tushman
1994 "The coevolution of technology and organization." In J. A. C. Baum and J. V. Singh (eds.), Evolutionary Dynamics of Organizations: 403–424. New York: Oxford University Press.
1998 "The co-evolution of community networks and technology: Lessons from the flight simulation industry." Industrial and Corporate Change, 7: 311–346.

Rucht, D.
2004 "Movement allies, adversaries, and third parties." In D. A. Snow, S. A. Soule, and H. Kriesi (eds.), The Blackwell Companion to Social Movements: 197–216. Malden, MA: Wiley-Blackwell.

Sabel, C. F.
1984 "Industrial reorganization and social democracy in Austria." Industrial Relations: A Journal of Economy and Society, 23: 344–362.

Santos, F., and K. M. Eisenhardt
2005 "Organizational boundaries and theories of organization." Organization Science, 16: 491–508.

Saxenian, A.
1994 Regional Advantage: Culture and Competition in Silicon Valley and Route 128. Cambridge, MA: Harvard University Press.

Schneiberg, M.
2002 "Organizational heterogeneity and the production of new forms: Politics, social movements and mutual companies in American fire insurance, 1990–1930." In M. Lounsbury and M. J. Ventresca (eds.), Research in the Sociology of Organizations, 19: 39–89. New York: Elsevier/JAI.

Schneiberg, M., and M. Lounsbury
2008 "Social movements and institutional analysis." In R. Greenwood, C. Oliver, J. A. Kahl, and R. Suddaby (eds.), Handbook of Institutional Theory: 650–672. London: Sage.

Scully, M., and A. Segal
2002 "Passion with an umbrella: Grassroots activists in the workplace." In M. Lounsbury and M. J. Ventresca (eds.), Research in the Sociology of Organizations, 19: 127–170. New York: Elsevier/JAI.

Shah, S.
2005 "Open beyond software." In C. DiBona, D. Cooper, and M. Stone (eds.), Open Sources, 2: 339–360. Sebastapol, CA: O'Reilly.

Snow, D. A., and R. D. Benford
1992 "Master frames and cycles of protest." In A. D. Morris and C. M. Mueller (eds.), Frontiers of Social Movement Theory: 133–155. New Haven, CT: Yale University Press.

Snow, D. A., E. B. Rochford, S. K. Worden, and R. D. Benford
1986 "Frame alignment processes, micromobilization, and movement." American Sociological Review, 51: 464–481.

Snow, D. A., S. A. Soule, and H. Kriesi
2004 "Mapping the terrain." In D. A. Snow, S. A. Soule, and H. Kriesi (eds.), The Blackwell Companion to Social Movements: 3–16. Malden, MA: Wiley-Blackwell.

Soule, S. A., and B. G. King
2006 "The stages of the policy process and the Equal Rights Amendment, 1972–1982." American Journal of Sociology, 111: 1871–1909.

Staggenborg, S.
1988 "The consequences of professionalization and formalization in the pro-choice movement." American Sociological Review, 53: 585–606.

Star, S. L., and J. R. Griesemer
1989 "Institutional ecology, 'translations' and boundary objects: Amateurs and professionals in Berkeley's Museum of Vertebrate Zoology." Social Studies of Science, 19: 387–420.

Stern, S.
2004 Biological Resource Centers: Knowledge Hubs for the Life Sciences. Washington, DC: Brookings Institution.

Thompson, T. A., and G. F. Davis
1994 "A social movement perspective on corporate control." Administrative Science Quarterly, 39: 141–173.

Timmermans, S., and V. Leiter
2000 "The redemption of Thalidomide: Standardizing the risk of birth defects." Social Studies of Science, 30: 41–71.

Tushman, M. L.
1977 "Special boundary roles in the innovation process." Administrative Science Quarterly, 22: 587–605.

Tushman, M. L., and L. Rosenkopf
1992 "Organizational determinants of technological change." In B. M. Staw and L. L. Cummings (eds.), Research in Organizational Behavior, 14: 311–347. Greenwich, CT: JAI Press.

Valloppillil, V.
1998 Halloween Document I (Version 1.17), Open Source Software: A (New?) Development Methodology. http://www.catb.org/~esr/halloween/halloween1.html. Accessed October 30, 2006.

von Hippel, E.
1988 The Sources of Innovation. New York: Oxford University Press.
2005 Democratizing Innovation. Cambridge, MA: MIT Press.

von Krogh, G., S. Spaeth, and K. R. Lakhani
2003 "Community, joining, and specialization in open source software innovation: A case study." Research Policy, 32: 1217–1241.

Walker, R.
2007 "Handmade 2.0." New York Times Magazine, December 16, Section 6: 76–81.

Waters, A.
2007 The Art of Simple Food: Notes, Lessons and Recipes from a Delicious Revolution. New York: Clarkson Potter.

Zald, M. N., C. Morrill, and H. Rao
2005 "The impact of social movements on organizations: Environments and responses." In G. F. Davis, D. McAdam, W. R. Scott, and M. N. Zald (eds.), Social Movements and Organization Theory: 253–279. New York: Cambridge University Press.