Understanding Collaboration in Fluid Organizations, a Proximity Approach

Dawn Marie Foster

A Thesis submitted in partial fulfilment of the
requirements of the University of Greenwich
for the Degree of Doctor of Philosophy (PhD)

April 2018

DECLARATION

I certify that the work contained in this thesis, or any part of it, has not been accepted in substance for any previous degree awarded to me, and is not concurrently being submitted for any degree other than that of Doctor of Philosophy (PhD) being studied at the University of Greenwich. I also declare that this work is the result of my own investigations, except where otherwise identified by references and that the contents are not the outcome of any form of research misconduct.

Student:

Dawn M. Foster Date: 9 April 2018

First Supervisor:

Guido Conaldi Date: 9 April 2018

ACKNOWLEDGEMENTS

This body of work would not have been possible without the help and encouragement of so many people. Thank you to my parents, the rest of my family, and the friends who provided encouragement for this crazy idea to quit a great job, put my career on hold, and move to London to pursue a PhD. A special thank you to my partner, Paul, who provided suggestions and encouragement while putting up with me through all of the highs and lows on this journey.

Thank you to my supervisors, Guido Conaldi, Riccardo De Vita, and Bruce Cronin who provided valuable insights, help and feedback along every step of the way. This work would not have been possible without their ongoing support and mentorship. Thank you to Francesca Pallotti and Gabriella Cagliesi for providing comments on an earlier version of this research, and thank you to the entire Centre for Business Network Analysis group at the University of Greenwich for ongoing feedback.

Thank you to the many people who provided invaluable data and technical assistance, including my 16 anonymous interview participants, Greg Kroah-Hartman at The Linux Foundation for his affiliation dataset along with other guidance, and Matthew Wilcox for pointing out a missing variable that had a big impact on the results. A special thank you to Jesus Gonzalez-Barahona, Daniel Izquierdo Cortázar, Santiago Dueñas, and Jose Manrique Lopez de la Fuente along with the rest of the group at Bitergia and the Universidad Rey Juan Carlos in Madrid for spending many hours helping me store and process large amounts of mailing list and source code data.

ABSTRACT

As fluid organizations become increasingly important and more commonly used, continued evolution in approaches to understanding collaboration within organizations is required. The aim of this study was to understand collaboration in a fluid organization through the exploration of proximity and the role of networks. This study used proximity theory to determine the role of cognitive, organizational, social, institutional and geographical proximity on the likelihood of collaboration within the Linux kernel, a fluid organization. This research contributes to the literature on fluid organization in three ways. First, five criteria are proposed to determine whether an organization is fluid, and those criteria are used to demonstrate that the Linux kernel is a fluid organization. Second, the research demonstrates that proximity theory can be used as a theoretical lens to better understand intraorganizational collaboration in fluid organizations. Third, the impact of third party organizations is shown to influence collaboration in fluid organizations. In addition to these contributions to theory, several implications for practice are also explored. The results of this work showed that cognitive and social proximities increased the likelihood of collaboration, and that individuals were also more likely to collaborate with others who work for the same employer. The findings for geographical proximity were mixed, but indicated that it provides a small increase in the likelihood of collaboration. There was no consistent evidence that institutional proximity influences the likelihood of collaboration. This research also demonstrated the use of several alternative ways to operationalize proximity and found several interactions between dimensions of proximity. Finally, it was found that network effects also influenced the likelihood of collaboration in this fluid organization.

CONTENTS

TABLES

FIGURES

CHAPTER 1. INTRODUCTION

Fluid organizations are expected to become increasingly important as new communications technologies, an increasingly networked society, and the proliferations of alternative working arrangements (e.g. contracting and outsourcing) result in organizations becoming increasingly fluid (Dobusch and Schoeneborn 2015). Gulati et al. (2012) argued that as collaboration within fluid organizations increases, continued evolution in how we think about organizations in needed.

March and Simon (1993) defined organizations as systems for coordinating activities between individuals to facilitate cooperation among people with diverse backgrounds and interests with a focus on supporting decision-making processes. Thus, the notion of organization can be expanded to include fluid organizations where the processes for coordination and decision-making are not rooted in hierarchical, top-down organizational structures nor in market coordination. In fluid organizations, the organizational boundaries and structures within a business are flexible, thus allowing for fluid movement within the organization (Ashkenas et al. 2002); however, fluid organizations may also emerge when groups of people collaborate and make decisions within a community that is recognized by its collective identity (Dobusch and Schoeneborn 2015).

Especially for the development of technology, which depends on information sharing across boundaries, a flexible organizational structure can facilitate collaboration (Allen 1977; Tushman 1977; Allen and Henn 2007). Some fluid organizations are based on global virtual work where work is performed online across many time zones with collaboration between people from different backgrounds (Nurmi and Hinds 2016), and these organizations may also include individuals from different firms and different types of institutions (O'Mahony and Bechky 2008). Fluid organizations can draw participants from various geographic locations, firms, and other types of institutions in a setting where similar knowledge (cognitive) and social or networked relationships facilitates collaboration. Boschma's (2005) five dimensions of proximity, cognitive, organizational, social, institutional and geographical, can be used to better understand collaboration between diverse individuals. It follows that individuals may be at the same or completely different levels across each of Boschma's (2005) five dimensions, thus understanding how proximity works within the context of collaboration within fluid organizations is relevant. This idea is aligned with recent proximity theory literature where proximity has been used in many studies to better understand collaboration (e.g. Cantner and Graf 2006; Knoben and Oerlemans 2006; Balland 2012; Crescenzi et al. 2016).

Each of Boschma's (2005) five dimensions of proximity help further understanding and provide different insights into collaboration within a fluid organization. Collaboration within traditional organizational forms can be mandated by those at the top of the hierarchy; however, in fluid organizations with flexible boundaries and evolving structures, participants must rely more on finding common ground in shared knowledge and relationships to facilitate effective collaboration. By considering all five dimensions of proximity, a variety of influences on collaboration within a fluid organization can be investigated. Organizational proximity accounts for the impact of collaboration based on the employer. Institutional proximity looks at the impact on collaboration from the type of institution (firm, nonprofit, academia) where individuals are employed. Cognitive proximity is based on technical knowledge, which may stem from knowledge gained through current employers or shared within an organization. Social proximity looks at relationships or interactions between people within an organization. Geographical proximity considers physical location or time zones of organization members.

This research focuses on proximity theory as a theoretical framework to understand collaboration within a fluid organization using an open source software project, the Linux kernel, as the empirical setting with the individual participant as the unit of analysis. Open source software is software that is developed in open, online communities where the source code is visible for anyone to use, share and contribute to the software. A kernel is a piece of software that provides the interface between the operating system and the computer hardware as described in Figure 1. Users of a computer do not interact with the kernel directly; they interact with the operating system, which uses the kernel to interact with the hardware in the computer. The Linux kernel is open source software that allows the various Linux-based operating systems (e.g. Red Hat Linux, Ubuntu, Debian) to interact with the computer hardware (e.g. CPU, memory, disk drives, printers, graphics cards) used by that operating system. The Linux kernel is loosely organized as a collection of subsystems, each focused on different functionality within the kernel.

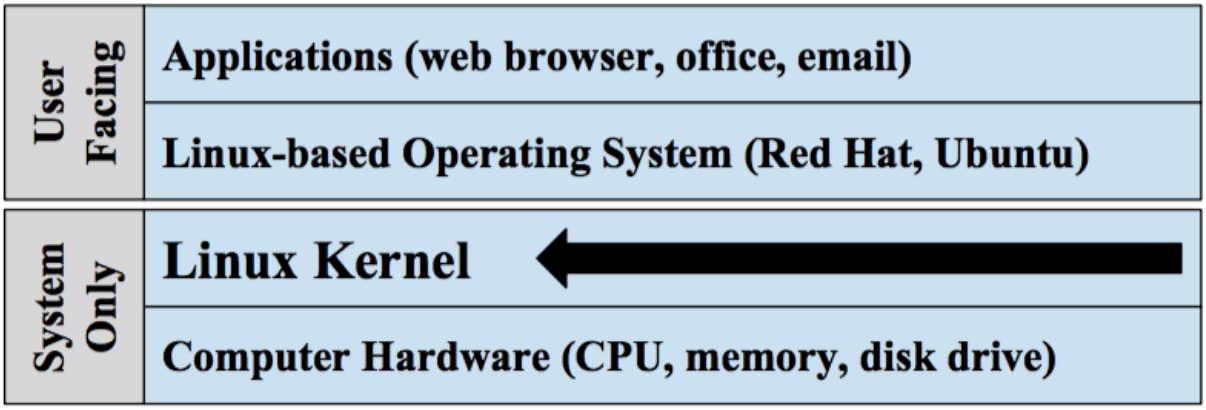

Figure 1: The Linux kernel explained

The justification for defining the Linux kernel as a fluid organization is developed later in this study and can be found in Phase 1, Section 4.5.1. From a terminology standpoint, this thesis describes the Linux kernel as a "fluid organization", and the term "third party organization" is used when referring to the firms, nonprofit organizations, academic institutions, or other organizations that employ people to participate in the Linux kernel. This is to avoid confusion when it might be ambiguous about whether the term "organization" refers to the Linux kernel itself or the third party organizations who employ the people who contribute to the Linux kernel.

In this setting, almost all of the participants contribute to this fluid organization as part of their employment relationship (Corbet and Kroah-Hartman 2017); however, in the existing literature on open source software, most of the research has looked at individual motivations for participating without fully considering the ways that this employment relationship influences participation (Iivari 2011; Crowston et al. 2012). Several of these studies were focused on why so many people contribute without receiving compensation (Hars and Ou 2002; Hertel et al. 2003; von Krogh et al. 2012); however, they found at least some element of participation from professional software developers (Hertel et al. 2003; Lakhani and Wolf 2005; Henkel 2006). Despite increased participation from software developers who are employed to contribute to open source software projects, very little research has been conducted to investigate collaboration as it relates to organizational and institutional proximity based on the third party organizations who employ these contributors. Because *why* individuals participate in open source has been well covered in the literature on motivation, this study explores *how* people participate by focusing on the influence of proximity on collaboration between participants who are employed to contribute to a fluid organization.

While several contributions investigate networks within fluid organizations, most, if not all of them, are looking solely at the ties between individuals without considering the third party organizations that employ many of these individuals. As employers become increasingly involved in fluid organizations, like open source software projects, gaining a better understanding of how employment affiliations impact interactions between individuals within fluid organizations is needed. A concrete example of how employment affiliations can impact participation in a fluid organization comes from 2010 when Google's Android code was removed from the Linux kernel because of issues with secrecy and the design of the code, but this did not just impact Google's employees within the Linux kernel, it also had real implications for employees at companies like HTC, Motorola, and Samsung who were also working with Google employees on Android (Korpi 2010). Because proximity theory can be used to understand collaboration across cognitive, organizational, social, institutional and geographical dimensions, it provides a diverse approach to investigating this phenomenon.

1.1. Contribution

Contributing to the literature on fluid organizations is becoming increasingly important as the number of fluid organizations continues to increase. With new communications technologies, an increasingly networked society, and the proliferation of alternative working arrangements (e.g. contracting and outsourcing), organizations are becoming increasingly fluid (Dobusch and Schoeneborn 2015). Gulati et al. (2012) argued that as collaboration within fluid organizations increases, continued evolution in how we think about organizations is required. This research contributes to the literature on fluid organizations in several ways.

First, this study proposes five criteria to determine whether an organization is a fluid organization and demonstrates that the Linux kernel is a fluid organization. The literature on fluid organizations is diverse with a wide variety of names and definitions used to describe fluid organizations. This research reviews some of the more common names and definitions to derive five criteria outlined in Section 2.4 of the Literature Review. This criteria is then applied to the Linux kernel to establish that the Linux kernel meets the criteria to be defined as a fluid organization.

Second, this research demonstrates that proximity theory can be used as a theoretical lens to better understand intraorganizational collaboration in fluid organizations. In traditional organizations, collaboration can be enforced by the hierarchy; however, the flexible boundaries and evolving structures of fluid organizations require that participants rely on common ground to facilitate collaboration, and this research shows that proximity theory is

one way of investigating this common ground for Boschma's (2005) five dimensions of proximity and their interrelationships. This research also demonstrates how alternative proximity measurements for cognitive, social and geographical proximities along with their interactions can be used to investigate collaboration within a fluid organization.

Third, this research adds to the body of knowledge on fluid organizations by highlighting the impact that third party organizations have on collaboration. Most of the existing literature on open source software has focused on individual motivations for participating in these fluid organizations (Crowston et al. 2012) with little concern for the increasing influence of third party organizations who employ these contributors. This research shows that for collaboration within fluid organizations, employer affiliation has a significant impact on the likelihood of collaboration.

In addition to contributing to the literature on fluid organizations, this research makes methodological contributions in the context of the analysis of interrelationships between proximity dimensions. The study demonstrates the importance of looking beyond coefficient signs and using visualization when interpreting interactions between variables. When considering interactions for two continuous variables, this study shows that it is important to investigate the variables in multiple ways by reversing the moderating variable. This provides valuable insights into the interaction at all levels of each variable to avoid possible misinterpretation at some levels.

1.2. Thesis Outline

The goal of this research is to answer these two questions, "How do participants who are employed by third party organizations collaborate within a fluid organization?" and "What is the role of proximity in these collaborations?" This thesis contains three phases that explore different aspects of this topic to answer these questions. Phase 1 uses qualitative research interviews and Boschma's (2005) five dimensions of proximity to better understand how participants collaborate within this fluid organization. In addition to contributing to the literature on fluid organizations, these results were also used as input into the remaining phases of the study. Phase 2 focuses on understanding the likelihood of a collaboration event using the dimensions of proximity, network effects, and several empirical setting control variables in a longitudinal relational event model. While the results from Phase 1 highlighted several potential interrelationships between the dimensions of proximity, Phase 2 focused on exploring each dimension of proximity separately without including interrelationships. Phase 3 builds on the earlier phases by adding proximity variable interactions to the relational event

model to further explore the likelihood of a collaboration event through understanding the interrelationships between proximity dimensions.

The thesis begins with a literature review in Chapter 2 that explores the topics of fluid organizations, proximity theory, and collaboration. Chapter 3 provides insight into the research design with details about the goals, approach, and dataset used for this research. Phase 1 explores the research setting in more detail, including a justification for the Linux kernel as a fluid organization, through a series of qualitative interviews and additional data in Chapter 4. Chapters 5 and 6 explore Phases 2 and 3 using quantitative methods each with introduction, methods, results, and discussion sections. The thesis wraps up with Chapter 7 highlighting the contributions and other conclusions from this research, and Chapter 8 outlining future work.

CHAPTER 2. LITERATURE REVIEW

2.1. Fluid Organizations

Many organizations are undergoing revolutionary changes with moves from rigid, hierarchical structures to more flexible, fluid ones (Ashkenas et al. 2002). Fluid organizations are ones where connections between individuals within the organization can become stronger or weaker and groups of people may form or disband to more effectively move ideas, information, and other resources as needed for the current situation (Glance and Huberman 1994; Ashkenas et al. 2002). Fluid organizations can be found in a variety of settings, including open source software (O'Mahony and Bechky 2008; Ferraro and O'Mahony 2012); a community and art festival (Chen and O'Mahony 2009); a social collective (Dobusch and Schoeneborn 2015); a research and development organization (Ahuja and Carley 1999), and other corporate environments (Powell 1990; Ashkenas et al. 2002). The advantage over traditional hierarchical structures is that fluid organizations have improved collaboration between individuals who can restructure the organization from within to facilitate cooperation (Glance and Huberman 1994). This does not necessarily mean that fluid organizations are self-organizing; in some cases, these organizations are shaped and designed in flexible ways by people who have either a formal or informal role within the organization (Gulati et al. 2012). This organizational fluidity can also create challenges as decision making processes cope with goal ambiguity and people who have other concerns that lead to varying levels of commitment for participation in decisions (Cohen et al. 1972).

The research on fluid organizations provides new ways of thinking about organizational theory as organizations evolve along with changes in technology and new ways of working. In particular, the proliferation of global virtual work with collaboration and communication occurring online across time zones with people located around the world is changing how people work within organizations (Nurmi and Hinds 2016). This virtual collaboration changes the way people think about organizing, since technology can replace hierarchy for coordination of activities while also giving individuals visibility into more aspects of the work processes (Zammuto et al. 2007).

New ways of working within fluid organizations were explored by Puranam et al. (2014) using an open source software project as the setting to show how existing organizational theories could be applied to fluid organizations. Dobusch and Schoeneborn (2015) used a fluid social collective to show how a loosely-formed group of people can be

considered an organization based on three criteria: decision-making within the group, those decisions being attributed to a collective entity, and a collective identity recognized by internal and external actors. Chen and O'Mahony (2009) looked at an environment where traditional organizational structures were not suitable for the needs of open source software projects, and more fluid organizational structures had emerged, instead. Similarly, O'Mahony and Bechky (2008) showed how fluid organizations emerging around open source software projects were used to facilitate governance, membership, ownership, and production. With the increasing numbers of fluid organizations coming into existence around open source software projects, understanding fluid organizations continues to be a worthwhile research topic.

2.2. Proximity Theory

Collaboration within strongly hierarchical organizations can be mandated from the top; however, in fluid organizations, each participant must navigate across boundaries in evolving structures, which requires finding common ground in shared knowledge and relationships to facilitate effective collaboration. Proximity theory is one framework that can be used to investigate this common ground across several dimensions that are relevant to collaboration within fluid organizations. In many fluid organizations, collaboration occurs between individuals from a wide variety of corporate, nonprofit, or academic affiliations; different backgrounds and knowledge; and disparate geographical locations in a flexible setting where the social relationship takes precedence over traditional organizational hierarchy. Proximity theory can be used as a framework to investigate how third party organization / institutional affiliation, cognition, geography, and social relationships influence collaboration (e.g. Knoben and Oerlemans 2006; Balland 2012; Crescenzi et al. 2016).

In general, proximity can be thought of as "being close to something measured on a certain dimension" (Knoben and Oerlemans 2006 pp.71–72). There are multiple approaches for using proximity theory and various, often overlapping, ways to define and measure the many dimensions of proximity (Knoben and Oerlemans 2006). Knoben and Oerlemans (2006) started with 11 dimensions, but reduced them to three primary dimensions that are most relevant for interorganizational collaboration with other dimensions included within the definitions of the primary three, which are organizational, technological, and geographical. A focus on interorganizational collaboration led Knoben and Oerlemans (2006) to conclude that social, cognitive, and institutional proximities can all be included within organizational proximity based on the assumption that shared culture and social relationships facilitate interactions between organizations. However, when looking at intraorganizational

collaboration at the individual level within a fluid organization, it is logical that social proximity and cognitive proximity should be considered separately from organizational proximity, since the relationships between people (social) and their subject matter knowledge (cognitive) do not necessarily stem from the third party organization where they are employed. This rationale is outlined in more detail in the next paragraphs.

For a study of collaboration within a fluid organization at the individual level of analysis, a more appropriate approach is to use Boschma's (2005) five dimensions of proximity: cognitive, organizational, social, institutional and geographical. These five dimensions are designed to reduce overlap and isolate the effect of each dimension relative to the others to determine their impact on learning and innovation (Boschma 2005). Boschma (2005) also looked at the effects of too much proximity versus too little proximity, arguing that a moderate amount of distance is necessary for learning and innovation, and this optimal level of proximity is referred to as the "proximity paradox" (Boschma and Frenken 2010). The various dimensions may also interact with each other where too much proximity in one dimension may be compensated by a greater distance on another dimension (Broekel and Boschma 2012). In this case and throughout this work, distance is defined as the inverse of proximity for any given dimension and does not necessarily refer to physical distance.

The role of proximity is especially important to the understanding of fluid organizations. In hierarchical organizations, management within the formal structure can require that people collaborate with other team members; however, in fluid organizations, collaboration occurs more organically. With a fluid organizational structure that adapts based on need, collaboration between individuals occurs as the need arises, rather than being assigned in advance as a result of hierarchy. This requires that participants have enough common ground to be able to communicate with each other to fulfill the goals of the organization. Each dimension of proximity can contribute to this common ground to facilitate collaboration between individuals. This highlights an important aspect of how fluid organizations are distinct and different from more rigid hierarchical types of organizations, which requires further investigation. Proximity theory provides a solid framework for this investigation of collaboration within fluid organizations. The next few sections discuss several elements of proximity theory in more detail, including each of Boschma's (2005) five dimensions of proximity, interrelationships between dimensions, proximity in collaboration, and proximity as it relates to fluid organizations.

2.2.1. Dimensions of Proximity

Cognitive Proximity

Cognitive proximity is required for two actors to effectively share knowledge, since a certain amount of similarity is required to process and understand new information and to facilitate communication, but on the other hand, some cognitive distance is useful for generating new ideas and innovation (Boschma 2005). Cognitive proximity has been operationalized in a variety of ways within the empirical literature. One common approach is to use technological codes: patent technology classification codes (e.g. IPC) (Nooteboom et al. 2007; Crescenzi et al. 2016) and industry technology classification codes (e.g. NACE) (Broekel and Boschma 2012). Other approaches for measuring cognitive proximity of individuals include using data based on educational backgrounds (Hansen 2015), documented skills (Criscuolo et al. 2010), and journal publications (Hardeman et al. 2015).

Nooteboom hypothesized that cognitive distance can be modeled as an inverted U-shaped function with communicability (1999) or comprehensibility (2000) on one side and novelty value on the other with the optimal cognitive distance at the peak of the U-shaped curve. Using technological distance to investigate cognitive distance in an interorganizational setting, Nooteboom et al. (2007) and Gilsing et al. (2008) confirmed the inverted U-shaped effect earlier hypothesized in Nooteboom (1999). In later works, Nooteboom (2009) has replaced communicability / comprehensibility with "ability to collaborate" as the balance to novelty value in the inverted U-shaped function of cognitive distance; in other words, increasing cognitive distance hinders the ability to collaborate, but facilitates the creation of innovative ideas both for organizations and the individuals embedded within those organizations.

Within fluid organizations where people may be distributed, technologies that facilitate virtual collaboration along with cognitive proximity in the form of shared knowledge allow effective communication for coordination of activities, thus enabling collaboration (Puranam et al. 2014). As mentioned previously, finding common ground for collaboration is important for fluid organizations, so collaboration requires that participants have enough cognitive proximity to be able to effectively communicate to accomplish their goals.

Organizational Proximity

Organizational proximity can be defined by the relationship in an organizational structure among people within and between organizations, but again there are extremes where

a lack of flexibility can result from too much organizational proximity and loss of control may result from too much organizational distance (Boschma 2005). The literature investigating organizational proximity is varied with some studies using the organization as the unit of analysis for interorganizational research with organizational proximity defined as organizations belonging to the same corporate group via parent / subsidiary relationships (e.g. Balland 2012). Other research uses individuals as the unit of analysis with organizational proximity operationalized using employment relationships of individuals to organizations. For example, organizational affiliation of individual inventors has been used to determine organizational proximity in several recent studies of collaboration on patents (e.g. Cassi and Plunket 2015; Crescenzi et al. 2016).

The proximity research using employment affiliations for organizational proximity is more applicable to this study of fluid organizations where participants may come from a wide variety of different organizations. Within some fluid organizations, participation from individuals who contribute on behalf of their employer is well documented (Mockus et al. 2002; Roberts et al. 2006; Jensen and Scacchi 2007; O'Mahony and Bechky 2008), thus organizational affiliation should be considered when studying collaboration within fluid organizations.

Social Proximity

The concept of social proximity is derived from the embeddedness literature (Granovetter 1985) and looks at relations between actors with trust coming from sources including friendship and experience (Boschma 2005). Social proximity is typically operationalized as the inverse of the distance between two people in the network (Singh 2005; Cassi and Plunket 2014; Ter Wal 2014; Cassi and Plunket 2015; Crescenzi et al. 2016). In other words, the inverse of the path length or number of people required to reach one person starting from another person based on network ties. Again, Boschma (2005) discussed issues of the extremes on both sides with too much social proximity as a detriment to learning and too much social distance associated with reduced trust and commitment. This is similar to Uzzi's (1997) findings about how embeddedness has several organizational benefits (e.g. efficiency, adaptation), but embeddedness when taken too far can reduce the flow of new information and increase the chance that problems faced by key connections will have significant impacts.

In fluid organizations, the evolving and changing networks of individuals facilitate collaboration as people rely on existing connections or make new ones as they change the

organizational structures to meet new needs (Glance and Huberman 1994). While personal connections and networks influence collaboration within almost every organization, they are even more important within fluid organizations, since fluid organizations often rely on these informal connections as the basis for how people work together (Glance and Huberman 1994). The importance of personal connections for collaboration within fluid organizations makes social proximity particularly relevant to consider.

Institutional Proximity

Institutional proximity is defined by actors sharing both formal (rules) and informal (norms and values) ideas, but too much institutional proximity can lock people into old ideas and too much institutional distance can deter collective action as a result of not sharing common values (Boschma 2005). At an interorganizational level, it can be used to investigate whether organizations tend to collaborate with other organizations who share the same institution type, and at an intraorganizational or individual level to understand whether individuals tend to collaborate more with others in a similar institutional setting. Institutional proximity is often operationalized as a dummy variable determining whether both actors belong to the same institutional setting, e.g. corporation, nonprofit, academic institution, government, or individual (e.g. Ponds et al. 2007; Balland 2012; Hardeman et al. 2015; Crescenzi et al. 2016). However, there are a number of very different ways that some authors have defined institutional proximity, including similarity between organizational culture and norms of two organizations (Hansen 2015) and whether two organizations are based in the same country (Balland et al. 2013).

Some participants in fluid organizations are employed by corporations, nonprofits, and universities, while others contribute as volunteers. In fluid organizations where individuals from a variety of institutional settings are collaborating together, understanding the role of institutional affiliation relative to the impact it has on collaboration should not be overlooked.

Geographical proximity

Geographical proximity typically refers to the physical, spatial distance between two entities (Boschma 2005). In this case, the proximity paradox should be thought of not as trying to find some optimal geographical distance, but rather as a balance of local and non-local contacts (Boschma and Frenken 2010). While too much proximity can reduce the potential for innovation, the idea of too much geographical distance is more complex, since other forms of proximity may act as substitutes for, or complements to, geographical proximity (Boschma 2005). It is typically operationalized using a measure of physical distances in kilometers /

miles or average travel times to determine proximity between organizations or individuals within a network (e.g. Ponds et al. 2007; Broekel and Boschma 2012; Ter Wal 2013; Cassi and Plunket 2014); however, in some intraorganizational or team studies geographical proximity may be based on floor plans, other office configurations, or time zones (O'Leary and Cummings 2007; Criscuolo et al. 2010).

For some types of fluid organizations, collocation is often impractical, and online communication technology is often used as a substitute for collocation (Gulati et al. 2012). In the case of networks and online communities, which are often not geographically localized, there may not be an inherently spatial dimension (Boschma 2005; Torre 2008). When interactions between people occur entirely online, the relationship to physical location tends to disappear (Torre 2008), but using time zones to indicate overlapping online work hours (O'Leary and Cummings 2007) is a feasible approach for settings where spatial distance could be irrelevant.

2.2.2. Proximity Interrelationships

While Boschma's (2005) five dimensions are designed to reduce overlap and isolate the effect of each dimension relative to the others, interrelationships between dimensions have been demonstrated in the literature. Dimensions of proximity act as complements when they have an additive effect, thus working together to influence the variable of interest. Organizational and cognitive proximity have been suggested as complementary as a result of a common knowledge often being shared within an organization (Nooteboom 2000; Boschma 2005; Cassi and Plunket 2014). In a study of collaboration between inventors, Cassi and Plunket (2014) found that cognitive and social proximity were complements.

In other cases, dimensions of proximity can be substitutes for one another where a lack of proximity in one dimension can be compensated for by the existence of proximity in another dimension (Balland et al. 2015). Crescenzi et al. (2016) and Hansen (2015) found that cognitive proximity acts as a substitute for geographical proximity in collaboration. Agrawal et al. (2006) and Cassi and Plunket (2015) both found that social proximity can be a substitute for geographical proximity. Sorenson et al. (2006) and Cassi and Plunket (2015) found that organizational proximity can act as a substitute for social proximity. In sum, the interrelationships between dimensions of proximity, both complementary and substitution effects, can help to understand the influences of proximity on the likelihood of collaboration within a fluid organization. This topic is explored in greater detail in Phase 3.

2.2.3. Proximity in Collaboration

While much of the proximity literature focuses on organizational innovation and learning, including Nooteboom's (2000) and Boschma's (2005) seminal theoretical works, proximity has also been used, especially in some of the more recent empirical literature, to understand collaboration. Collaboration is a complex subject that has been investigated within the proximity literature in a few different ways. Cassi and Plunket (2014) examined both the likelihood of forming collaboration ties and the inventive performance of those collaborations while their 2015 paper investigated determinants of tie formation for research collaborations between individual inventors. Crescenzi et al. (2016) looked at which dimensions of proximity influence collaboration between inventors that lead to knowledge creation in the form of patents, and Sorenson et al. (2006) focused on knowledge transfer within an inventor collaboration network. Hansen (2015) used interview data to investigate the importance of interrelationships between proximity dimensions on collaborative innovation processes, but between firms, rather than individuals.

This thesis is focused on which of the various dimensions of proximity influence the likelihood of collaboration between individuals, which is different from collaboration output performance (Cassi and Plunket 2014), collaborative knowledge creation (Crescenzi et al. 2016), knowledge transfer between collaborators (Sorenson et al. 2006), and collaborative innovation between firms (Hansen 2015). While some of these aspects of collaboration are not the same as this work's focus on the likelihood of collaboration, these empirical studies are similar enough to provide valuable insights while recognizing that there are likely to be subtle differences that might impact this study.

2.2.4. Proximity in Fluid Organizations

As mentioned earlier, proximity has an important role in understanding collaboration within fluid organizations where collaboration is organic, rather than being encouraged or enforced by management within a hierarchical organization. Individuals must find common ground with others as the need for collaboration arises within the fluid organization, and Boschma's (2005) dimensions of proximity provide a framework for understanding some of the similarities or differences that might influence the likelihood of collaboration between individuals in absence of a rigid hierarchy. This is an important distinction between fluid organizations and traditional, rigidly hierarchical organizations that is worthy of additional investigation.

Much of the existing collaboration research using proximity theory is focused on an interorganizational view with the firm as the primary unit of analysis (e.g. Knoben and Oerlemans 2006; Balland 2012; Balland et al. 2013), and there are very few proximity studies considering an intraorganizational perspective to look at individuals within an organization (e.g. Boschma 2005; Singh 2005). However, there are more studies, particularly ones focused on understanding collaboration between co-inventors on patents, using individuals as the unit of analysis from a network perspective (e.g. Cassi and Plunket 2014; Ter Wal 2014; Cassi and Plunket 2015; Crescenzi et al. 2016). The approach to studying proximity and operationalization of variables is similar, but the unit of analysis will often be different. For example, Balland (2012) defined social proximity as the inverse of the distance between two organizations within a collaboration network in an interorganizational study while Ter Wal (2014) used the same measure but between individual co-inventors on patents. For cognitive proximity, Nooteboom et al. (2007) used technology classifications on patents to determine cognitive proximity between organizations in an interorganizational study, and Crescenzi et al. (2016) used the same technology classification data but for individuals.

This research uses an intraorganizational approach to look at the likelihood of collaboration within a fluid organization, but it draws from the existing interorganizational, intraorganizational, and especially the network proximity literature because of the nature of fluid organizations and this empirical setting. Fluid organizations and this empirical setting make use of networks to facilitate collaboration. Within a fluid organization, collaboration and cooperation are not enforced by a hierarchy, but are dependent on the work being facilitated through social, cognitive, and geographical dimensions of proximity that provide shared experiences, knowledge, and timing. Because people participate in this fluid organization from many different third party organizations across corporate, nonprofit, and academic institutions, organizational and institutional proximity become important considerations for this fluid organization as part of an intraorganizational study. The interrelationships between these dimensions aids in understanding collaboration within the context of a fluid organization.

This research contributes to the fluid organization literature by showing how proximity theory can be used to better understand intraorganizational collaboration in a fluid organization along Boschma's (2005) five dimensions of proximity, several of which are used to better understand the impact that employers have on collaboration. With very little research devoted to investigating how employer affiliation of individuals impacts collaboration in open source projects, using proximity theory to understand collaboration within fluid organizations fills a gap in the existing literature.

2.3. Collaboration as a Network Phenomenon

Networks are important for understanding the economic actions and behavior that provide the basis for organizational collaboration. Powell (1990) made it clear that markets and hierarchies are not sufficient for understanding the behavior associated with organizations, but that network forms of organization provide reciprocal communication and exchanges that facilitate long-term cooperation, provide incentives for learning and sharing information, and allow for flexible use of resources in uncertain environments. White (1981 p.518) described markets as "self-reproducing social structures among specific cliques of firms and other actors who evolve roles from observations of each other's behavior." Granovetter's (1985) notion of "embeddedness" highlighted how economic actions are situated within a web of social relations. These seminal works highlighted the importance of networks, but from slightly different perspectives. White (1981) approached networks as evolving out of markets of producers and buyers using a systemic approach, while Powell (1990) viewed networks as a third mode of organization in addition to markets and hierarchies. With his view that networks play a central role with most individual behavior embedded within network relationships, Granovetter's (1985) micro level approach can be thought of as being on the opposite end of the spectrum from White's systemic macro level approach. Regardless of the differences in perspectives, these works demonstrated that networks should be taken into account when seeking to understand behavior, including collaboration, within organizations.

2.3.1. Collaboration and Networks

There is a rich history of studying networks both within and between organizations that has evolved from studying informal structure to a more formal quantitative analysis for how individuals and organizations are interconnected (Baum and Rowley 2002). At the most basic level, collaboration is simply the act of working with another to produce some output, and this act is often performed in the context of organizations. As people collaborate together within organizations, the process for doing so is not necessarily hierarchical, even in hierarchical organizations, but will also involve relationships as influence and control moves in other directions not dictated by the hierarchy (March and Simon 1993). Rank et al. (2009) found that the hierarchical structure has limited influence for intraorganizational cooperation; however, network ties (friendship in this study) when compared to the formal structure and procedures to some extent acted as a stronger governance mechanism and shared interests had a larger influence on the formation of collaborative relationships. Within organizations, an

increasing amount of work is accomplished by collaboration that stems from informal networks, instead of hierarchical structures, and social network analysis provides insights into these informal interactions (Cross et al. 2002). In fluid organizations with their flexible and evolving hierarchy, the role of networks is especially important for understanding collaboration.

When discussing networks, there are some common concepts and terminology that are important to understand. The social entity for the unit of analysis (e.g. individual, corporation, department) is referred to as the "actor," and these actors are linked together by "ties" based on a wide variety of potential linkages (e.g. friendship, respect, business transactions, behavioral interactions) (Wasserman and Faust 1994). These ties may be undirected when there is no order to the relationship (e.g. neighborhood or classroom affiliation) or directed when a tie has a specific orientation from one actor to another (e.g. advice or trust) (Wasserman and Faust 1994). The person initiating the tie is referred to as the "ego" and the receiver of the tie as the "alter" for directed networks. The ties linking actors together can be dyadic to investigate various properties associated with pairs of actors or triadic to understand how the dyad is embedded in the sociological processes introduced by the influence of a third person on the interaction (Granovetter 1985; Wasserman and Faust 1994; Kilduff and Tsai 2003).

2.3.2. Network Effects

The ties linking actors together create local network structures, and how actors are embedded within these local structures is an important element of understanding collaboration. Relational embeddedness is based on the cohesive ties formed through cooperation between two actors at the dyadic level, and structural embeddedness considers the structure of relationships surrounding two actors at the triadic level (Gulati and Gargiulo 1999). Both of these forms of embeddedness are considered in more detail in this section.

Relational embeddedness. Because collaboration is based on two actors working together to produce some output, the most basic level of network analysis is the dyad, which can be considered in a variety of ways. One way of investigating dyadic relationships between actors in directed networks is to understand the extent to which the relationship between the ego and the alter is reciprocal (also referred to as mutual or symmetric) where the ties go in both directions between the ego and the alter (Wasserman and Faust 1994; Kilduff and Tsai 2003). For example, if one person claims to be friends with a second person, reciprocity is only demonstrated when the second person also considers the first to be a friend, thus showing

that the relationship is mutual. Quintane et al. (2013) found that collaboration within project teams tends to be reciprocal with short-term reciprocity demonstrated by replies to emails, but they also found that this persisted over longer periods of time showing sustained reciprocal relationships between team members. Within fluid organizations where collaboration is heavily influenced by the network, understanding reciprocity is important for understanding how past collaboration exchanges between two actors influence the likelihood of future collaboration between them. One way of understanding reciprocity is based on recency by looking at communication between two collaborating individuals to see if communication from one person follows a recent communication from the other individual (Butts 2008). Participation shifts based on conversational norms related to expectations of turn-taking in network relationships over time are also a type of reciprocity (Butts 2008). Network relationships between two actors can also be understood in the context of the persistence of the relationship over time indicating that past contacts become future contacts with a tendency toward inertia in these relationships represented by repeated interactions between actors (Butts 2008). Understanding how repeated interactions between actors influences the likelihood of future collaboration is especially important for fluid organizations where participants must rely on existing relationships and shared knowledge gained in past interactions for collaboration in absence of management enforced hierarchical collaboration.

Structural embeddedness. While dyadic relationships are important, it is also important to consider how relationships are embedded within the broader social structure (Granovetter 1985). Interactions between two people in an organization are often influenced by other third parties who may know one or both of the other participants, and those third parties can have a variety of influences (e.g. introductions, mediation of disagreements, new ideas) (Kilduff and Tsai 2003). Simmel (1950) provided a theoretical justification for how triadic relationships have a fundamentally different dynamic than dyadic relationships and claimed that the addition of more actors into a group beyond the triad does not appreciably change the dynamic. When considering triads, each of the three people may or may not be connected to the other two people making up the triad. When all three actors are connected via ties to each other, the triad is considered closed as shown in Figure 2. For example, this process of "closure" can be interpreted as a "friend of a friend effect" and is what leads to stable triadic states.

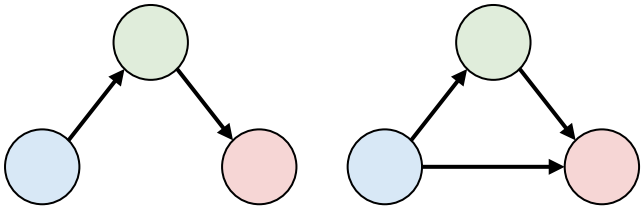

Figure 2: Closure example diagram

In directed network analysis there are different configurations that can be used to represent these triadic relationships (Robins et al. 2009). The four closure configurations described by Robins et al. (2009) are transitive closure (also called path closure), cyclic closure, shared partnership inbound (also called popularity closure), and shared partnership outbound (also called activity closure). These four closure configurations, which will be described in more detail in Section 5.2.3, together provide a greater understanding of the network phenomenon that might not be obvious when considering only a single triadic effect. Understanding the variety of influences that a third party might have on a collaboration event between two people is important to understand within the context of the flexible and changing structures associated with fluid organizations. For example, a recent longitudinal study of Wikipedia, a fluid organization, found that triadic closure effects influenced hierarchy formation between collaborating editors (Lerner and Lomi 2017).

Triadic relationships form the basis for more sophisticated analysis of the network structure. One of the most influential contributions to the literature on social networks for organizational scholars is Burt's (1992) structural holes theory (Krackhardt 1999). Burt (1992 p.18) defined structural holes as "a relationship of nonredundancy between two contacts" where these two actors are disconnected, thus leaving a gap in the network that can be filled by a broker who gains power within the network by acting as a bridge between the disconnected actors. Robins et al. (2009) extended Burt's (1992) concept of structural holes, which are often conceived at the global network level, and applied the concept more specifically to triads. Krackhardt (1999) questioned whether the role of a broker bridging structural holes is always an advantage, especially in the presence of Simmelian ties which are strong reciprocal ties between two actors who also have strong reciprocal ties with the same third party. These strong triadic relationships can result in pressure from the other members of the triad to conform to certain norms, thus restricting their independence, autonomy and power, which can result in actors becoming more constrained when they have Simmelian ties as part of multiple triads (Simmel 1950; Krackhardt 1999). Krackhardt (1999) suggested that both theories are valid, but structural holes theory might have more benefits for brokers when

the interactions are private and different groups are unaware that behaviors of the broker vary across groups, while Simmelian ties might create more restrictions when the interactions are public and open to the scrutiny of the rest of the group who can apply pressure to confirm with group norms. This highlights the importance of understanding these triadic network relationships and their influence on collaboration.

2.3.3. Network Models

In the previous sections, there are several references to longitudinal studies of networks. As with almost anything involving humans, collaboration is a dynamic activity that evolves over time. The people that someone collaborated with a year or even a month ago may not be the same people today, or the nature of the collaboration with those people may have changed. Within fluid organizations where structures are flexible and people can collaborate with others based on a particular need, collaboration may become even more dynamic. Since collaboration between actors evolves over time and is not static, especially within fluid organizations, the analysis of network influences on collaboration should be considered over time (Cross et al. 2002).

There are different ways of understanding collaboration as a network phenomenon over time. Including time as an element of the network analysis to understand collaboration can be accomplished using two different approaches: panels (discrete time) or sequences. For longitudinal network analysis, data may be based on panel data measured at regular intervals or a sequence based on the exact time or order in which something occurred. Panel data is particularly well suited for data that is relatively stable over a period of time, and some information is only available as panel data (e.g. data gathered via periodic interviews and industry data gathered on a yearly basis). For example, panel data is commonly used in understanding networks of advice relationships, interorganizational relationships, and friendship. Agneessens and Wittek (2012) used panel data collected in six month intervals within a Dutch housing company and found that advice relationships tend to be reciprocal at the dyad level, but at the triad level, higher status actors rarely look to lower status actors for advice. Checkly and Steglich (2007) found that ties between venture capital firms originate at the interorganizational level and were not the result of interpersonal ties between partners who have changed firms based on panel data gathered yearly over a period of 10 years. Sequence data is more suited to understanding a rapidly evolving phenomenon, like collaboration or communication dynamics. In the Lerner and Lomi (2017) study mentioned earlier about hierarchy formation between collaborating Wikipedia editors, they used sequence data for

article revisions. Sequence data from open source project bug tracking systems has also been used to understand organizational problem-solving behaviors (Quintane et al. 2014). With collaboration as the subject of this research, the focus here is on sequence data.

There are three longitudinal network models that can be used to understand how networks evolve over time: exponential random graph models, stochastic actor oriented models, and relational event models (Quintane et al. 2014). Exponential random graph models are used to understand network configuration and structure, which emerges from combinations of sub-structures within the network (Robins et al. 2009). While exponential random graph models were not designed as longitudinal models, they have been extended by adding temporal capabilities to model network evolution (Hanneke et al. 2010). Stochastic actor-oriented models look at network evolution over time from the perspective of actions from individual actors focused on optimizing their own individual utility in light of network constraints and / or external influences (Snijders 1996). However, both of these models have the same limitation relative to their use in longitudinal studies; they can only be used with panel data. Exponential random graph models and stochastic actor oriented models can only be used with sequence data if the data are aggregated into discrete time frames and restructured to create panel data, but relational event models can model sequence data without losing information through aggregation (Quintane et al. 2014). Relational event models use a sequence of actions generated by a sender directed toward a receiver to understand events by putting them in context of the network and the actions preceding them (Butts 2008). Because this research is focused on understanding collaboration events between individual participants in a fluid organization, the focus is on a relational event approach to network analysis using directed ties based on a collaboration event linking two individual participants as actors with the person initiating the collaboration event as the ego and the receiver as the alter. The relational event model and its application to this study will be described in more detail during Phase 2 in Section 5.2.2.

2.4. Types of Fluid Organizations

It is important to recognize that not every fluid group of people is an organization, and not every organization is fluid. For social groups, Ahrne (1994) stated that a social unit must meet all four of the following criteria before being considered an organization: affiliation, collective resources, substitutability of individuals, and recorded control. First, affiliation includes recognition of members and exclusion of non-members, but also demonstrates one belongs to a collective identity as part of this affiliation (Ahrne 1994; Dobusch and

Schoeneborn 2015). Second, the collective resources of an organization can be used by members, but there is often also an expectation that members will contribute to their production and maintenance in addition to making decisions about how to use these resources to achieve the goals of the organization (Ahrne 1994). This decision-making aspect is commonly seen as one of the most fundamental defining features of organizations (March and Simon 1993; Ahrne and Brunsson 2010; Dobusch and Schoeneborn 2015). Third, substitutability means that the organization is not defined by any specific individual, but that individuals leave and new ones join while continuing to fulfill the goals of the organization (Ahrne 1994). Finally, recorded control refers to the act of keeping track of contributions and making decisions to provide corrections or sanctions as well as to reward members for good performance, but this does not necessarily imply that the decisions about control come from a hierarchical structure (Ahrne 1994; Ahrne and Brunsson 2010). Dobusch and Schoeneborn (2015) demonstrated that a fluid social collective meeting these three criteria can be considered an organization: first, decision-making within the group; second, those decisions being attributed to a collective entity; and third, a collective identity recognized by internal and external actors. These three criteria can be described as falling within Ahrne's (1994) criteria. These criteria also demonstrate that there are other types of social groups that should not be considered organizations, since they don't meet all four criteria: e.g. families (individuals cannot be substituted) and tourist groups (do not meet recorded control criteria) (Ahrne 1994). These criteria are summarized in Table 1.

Table 1: Criteria for organizations

Ahrne	**Dobusch and Schoeneborn**	**Overview**
Affiliation	Collective Identity	Members identify with their organizational affiliation and non-members recognize this identity.
Collective Resources	Decision-making	Members make decisions about how to use the collective resources of the organization.
Substitutability	Collective Identity	Any role can be filled by another person when an individual leaves or changes roles while maintaining the identity of the organization.
Recorded Control	Decision-making	Keeping a record of contributions and making decisions about whether to reward or sanction individuals for their performance within the organization.

Fluid organizations must go beyond meeting the criteria to be considered an organization and also demonstrate their fluidity. The literature on fluid organizations contains overlapping concepts with varied terminology referring to organizations with flexible, changing organizational structures. A variety of types of organizations can be considered fluid organizations including: boundaryless organizations, network organizations, virtual organizations, chaotic organizations, ad hoc organizations, and meta-organizations (Ashkenas et al. 2002; Gulati et al. 2012).

Boundaryless Organizations. Along with flexible structures, fluid organizations also tend to have flexible boundaries both internally between the individuals and the organization along with the external boundary between the organization and the market (Chen and O'Mahony 2009). These flexible boundaries have led some to refer to fluid organizations as boundaryless organizations, but this term can be misleading, since these organizations only have more fluid boundaries, rather than being entirely without boundaries. For example, in one characterization of a boundaryless organization, the authors do not advocate for removal of all boundaries, which they claim would generate disorganization, but they do suggest that boundaries can be more permeable and fluid than they have historically been (Ashkenas et al. 2002). One challenge is to find ways to manage these boundaries and allow for an organization to be fluid while still providing enough structure to ensure the quality of the output produced (Ferraro and O'Mahony 2012). Much of the literature on open source software communities as fluid organizations comes from the literature on boundaries.

Network Organizations. Podolny and Page (1998 p.59) defined network organization as "any collection of actors (N >= 2) that pursue repeated, enduring exchange relations with one another and, at the same time, lack a legitimate organizational authority to arbitrate and resolve disputes that may arise during the exchange." Their study found that network organizations facilitate learning, create status / legitimacy, manage resource dependencies, and provide other benefits. Powell (1990) pointed out that network forms of organizations are more flexible and can facilitate efficient collaboration, cooperation, and knowledge sharing, but this comes with more complexity and the potential for conflict that cannot be simply solved via the hierarchy. Snow et al. (1992) discussed how within network organizations, multi-level hierarchies are being replaced by business units coordinated by market mechanisms, rather than layers of managers to provide both the efficiency and flexibility to adapt and compete in an increasingly challenging global market.

Meta-organizations. Meta-organizations are sometimes defined narrowly as organizations with other organizations as members (Ahrne and Brunsson 2005); however,

Gulati et al. (2012 p.573) expands this definition to include individuals: "meta-organizations comprise networks of firms or individuals not bound by authority based on employment relationships, but characterized by a system-level goal." While there is no formal authority, members within meta-organizations like the Linux kernel and Wikipedia collaborate and work toward a common goal, often using online tools to communicate across physical distances (Gulati et al. 2012).

Virtual Organizations. DeSanctis and Monge (1999 p.693) described a virtual organization as "a collection of geographically distributed, functionally and/or culturally diverse entities that are linked by electronic forms of communication and rely on lateral, dynamic relationships for coordination. Despite its diffuse nature, a common identity holds the organization together in the minds of members, customers, or other constituents." Ahuja and Carley (1999) mentioned that virtual organizations are often described in the literature as decentralized and nonhierarchical; however, in their study, they found centralization and hierarchy, but as something that had emerged and evolved out of the network, rather than being predefined in the manner typically found in traditional organizations.

This research uses the term fluid organizations to refer to this combined concept of boundaryless organizations, network organizations, meta-organizations, virtual organizations, and other names for these newly evolving organizations defined by their flexible, evolving structures. Drawing from these various concepts and definitions, a new framework containing five criteria required for an organization to be considered a fluid organization is proposed and summarized in Table 2 in contrast to traditional, non-fluid organizations. First, the fluid organization must meet the requirements drawn from Ahrne (1994) and Dobusch and Schoeneborn (2015) that is outlined in Table 1 to determine whether the group can be considered an organization. The four criteria for being considered an organization are affiliation, collective resources, substitutability, and recorded control with collective identity and decision-making as key components of the four criteria. Second, hierarchy within fluid organizations may be absent, or if there is a hierarchy, it emerges organically from the network and evolves as needed in a flexible manner. The literature on virtual organizations, in particular Ahuja and Carley (1999), discussed that these organizations are often described as non-hierarchical, but provide evidence for an evolving and emergent hierarchy. Third, boundaries are either flexible or non-existent, which makes it easy for members of the fluid organization to collaborate across sub-groups or teams. This criteria comes out of the literature on boundaryless organizations allowing collaboration to flow across more permeable boundaries (Ashkenas et al. 2002; Chen and O'Mahony 2009). Fourth, collaboration is

organic and facilitated by members finding common ground that reduces coordination costs rather than being enforced by the hierarchy. The origins of this criteria are a bit more nuanced and emerged out of the idea that members are finding some type of common ground to facilitate collaboration through working toward a common goal without formal authority as in meta-organizations and the dynamic relationships coming from virtual organizations and network organizations (Podolny and Page 1998; DeSanctis and Monge 1999; Gulati et al. 2012). Finally, the network has a pronounced role in facilitating collaboration between individuals, and the network also influences changes within the hierarchy to adapt to the needs of the fluid organization. This final criteria comes out of the literature on network organizations (Powell 1990; Podolny and Page 1998). This research proposes that all five criteria must be present for an organization to be considered a fluid organization.

Table 2: Criteria for fluid organizations

Characteristic	Fluid Organization	Traditional Organization
1. Organization	Meets the criteria set out in Table 1 to be considered an organization.	
2. Hierarchy	If present, evolves as needed in a flexible manner and most likely emerges organically from the network. May lack a defined hierarchy.	Defined in a top-down manner by management as a rigid, formal structure.
3. Boundaries	Members collaborate across flexible or non-existent boundaries between sub-groups or teams.	Organizational units have clear and rigid boundaries defined within the hierarchy.
4. Collaboration	Individuals collaborate organically facilitated by finding common ground.	Driven within the hierarchy and enforced by management.
5. Network	Pronounced role in collaboration between individuals that influences changes within the hierarchy.	Informal role in collaboration between individuals outside of the hierarchy.

2.5. Open Source Software Communities as Fluid Organizations

In the early days of open source software, the projects were community managed by definition; however, the term open source software now covers a much broader range of projects to also include projects released under open source software licenses, but managed in part or entirely by corporations and other third party organizations (O'Mahony 2007). When an open source project is not community managed, but is driven from within a third party

organization, the project hierarchy would come from the hierarchy of the third party organization, which may or may not be a fluid organization. For this reason, throughout the rest of this thesis, the term "open source software" refers only to community managed open source projects, which have been frequently studied as one type of fluid organization.

With participation data available and publicly accessible, open source software projects provide a rich environment for conducting research on fluid organizations. For example, a Chen and O'Mahony (2009) study sought to better understand the fluid organizations formed by open source software projects to enable software production in an environment where more conventional organizational structures did not meet the needs of the project. Puranam et al. (2014) used an open source software project as one setting to assess the extent that novel, fluid forms of organizing are truly new as a first step to providing insight into whether they can be interpreted in the context of existing organizational theories, rather than requiring completely new organizational theories. Alexy et al. (2013) discussed how open source software development is an organizational innovation that spans organizational boundaries. O'Mahony and Bechky (2008) used several open source software communities as the setting to investigate how fluid organizations facilitate collaboration by reinforcing aligned interests and building bridges between three groups with divergent interests by looking at the roles of the open source community (individuals), the foundations formed to support the projects (nonprofits), and third party organizations that sponsor individual participants. This forms a triadic role structure between a) the community where individuals collaborate directly, b) a nonprofit foundation that acts as a legal entity for the community, and c) third party organizations that employ contributors (O'Mahony and Bechky 2008).

In community managed open source software projects, decisions are made and contributions are accepted from individuals, rather than third party organizations, leaving no direct way for employers to participate in these fluid organizations (O'Mahony 2007). However, one of the ways that a third party organization can contribute to these open source software projects is by having employees contribute, and projects are seeing increased participation from software developers employed to contribute (Roberts et al. 2006; Jensen and Scacchi 2007). Individuals, many of whom are employed to participate as part of their job, devote significant time and resources to contribute to open source software projects that are ultimately released as a public good (von Hippel and von Krogh 2003; Grand et al. 2004). While some researchers have emphasized the antagonism between corporate and community interests (Hars and Ou 2002), others have found that there are benefits to having third party organizations involved in open source communities for innovation, knowledge creation,

productivity, and more (Mockus et al. 2002; Grand et al. 2004; Henkel 2006). It has become quite common for third party organizations to employ many of the people working on open source software projects, including extending offers of employment to existing community members (Mockus et al. 2002; Jensen and Scacchi 2007). A vibrant industry has been built up around employers who hire software developers to participate in open source software projects, and this commercial involvement benefits the projects in a variety of ways, including implications for recruiting and retaining of software developers (Roberts et al. 2006).

Despite the increased participation from individuals contributing on behalf of their employer, the existing research has focused more on understanding what motivates individual people to participate in open source software projects without considering the complexities introduced by the involvement of third party organizations (Iivari 2011). The motivation for individuals to contribute to open source software projects has been studied extensively with a focus on looking at why so many people contribute without receiving compensation (Hars and Ou 2002; Hertel et al. 2003; von Krogh et al. 2012). However, while many of these studies began with the premise that contributions were voluntary, they found at least some element of participation from professional software developers. An early survey conducted in 2000 on motivation for contributions to the Linux kernel was designed to understand why software developers participate "for free"; the study found that 20% of the software developers were paid to contribute as part of their regular job and another 23% were sometimes paid for their Linux work (Hertel et al. 2003). In a survey of open source software developers, Lakhani and Wolf (2005) found that 40% were being paid to contribute to open source software projects and that some of the leading motivations for this group to participate included work-related needs and enhancing professional status. One study found that many "hobbyist" software developers are actually skilled software developers who are employed at third party organizations, often in management positions, but who contribute to open source software projects on their own time as a creative outlet or for an additional challenge (Shah 2006).

In one study of embedded Linux developers, the majority of the people participating in embedded Linux projects were employed software developers who had an average of over 14 years of programming experience (Henkel 2006). Software developers who are employed to contribute to open source software projects are often motivated to make more contributions to the project (Lakhani and Wolf 2005; Roberts et al. 2006), have a stronger desire to get their code incorporated into the project, and some eventually move into leadership roles within the project (Shah 2006). The motivations for software developers employed to contribute to open source software projects varies quite a bit from software developers who contribute on a

volunteer basis. Software developers employed by third party organizations tend to be motivated more by external rewards, including a personal need for the software, self-marketing and sales of related products, while hobbyist developers were motivated by internal rewards, like altruism and community identification (Hars and Ou 2002). von Krogh et al. (2012) provided a comprehensive summary of the research into the motivations of open source software contributors and suggest that individual motivations should be understood in a broader context of the social practices involved. As more third party organizations dedicate employees' time to open source software project work, understanding fluid organizations and how collaboration occurs within these organizations is becoming increasingly important.

As introduced in a previous section, collaboration is a network phenomenon and thus there is a growing body of research on networks with fluid organizations, including open source software projects, as the setting. One of the first published studies that analyzed open source software projects using network analysis looked at a large number of projects and individual software developers modeled as a collaborative social network with software developers as nodes in the network and joint project membership represented as a link to show participation in one or more projects (Madey et al. 2002). Another research article looked at case studies of three separate open source software projects, providing details about how to use source code repository data to build a "modules" network that linked a module when one or more people had contributed to the same module and a "committers" network of people who had worked together on the same module (López-Fernández et al. 2006). While source code is one of the more popular data sources for open source software projects, other network analysis studies use data from bug trackers. For example, one study used bug tracker data to look at communication between software developers and found surprisingly diverse centralization with some project teams highly centralized and others decentralized (Crowston and Howison 2006), and another studied organizational problem solving using data from bug trackers to look at collaboration and contributions from core and peripheral project team members (Conaldi and Lomi 2013). Another looked at network density, centralization and boundary spanning activities of team members to study the collaboration structures of over 100 open source software projects to determine the impact on productivity and quality of the projects (Colazo 2010). However, most of these studies focus on networks of individuals without considering the affiliations to the third party organizations that employ these software developers.

2.6. Summary

The study of collaboration between individuals is complex with many different things coming together to influence whether two people will collaborate. Within traditional, rigidly hierarchical organizations, there is an expectation that people within the same group will collaborate, and managers within this structure can ensure that people are collaborating. However, fluid organizations cannot rely on the hierarchy to enforce expectations of collaboration, so collaboration must occur organically between individuals. This organic collaboration influences the evolving and flexible structures found within fluid organizations. Individuals must work within these evolving structures and find enough common ground to facilitate collaboration through shared experiences or knowledge that allow them to work together in a effective manner over time. As individuals find common ground and begin collaborating with others, they become embedded within a network of other individuals who are also collaborating within the fluid organization. Individuals can rely on these network relationships with others to facilitate further collaboration in the future. Understanding the dimensions along which individuals have common ground is a gap in the existing research on collaboration within fluid organizations. Boschma's (2005) five dimensions of proximity provide a way to fill this gap and understand the various ways that individuals have enough common ground to enable collaboration. This leads to the research question, "What is the role of proximity in these collaborations?"

Open source software has often been used as the setting to understand fluid organizations, but most of the research fails to consider the influences that employer affiliation might have on contributions and collaboration within these projects, and because employers can influence how employees spend their time, this affiliation should be factored into the analysis. The Linux kernel is an open source project and fluid organization where most of the participants are employed by third party organizations, which makes it a good setting for addressing this gap in the literature on fluid organizations with the following research question, "How do participants who are employed by third party organizations collaborate within a fluid organization?" The following section on Research Design demonstrates how these questions are answered in the three phases of this research project.

CHAPTER 3. RESEARCH DESIGN

3.1. Goals and Approach

An instrumental case study approach is used for this research (Stake 1995) with the Linux kernel as the subject of the case study. A case study provides an environment for empirical inquiry with an in-depth investigation of a current phenomenon using multiple data sources and approaches within a bounded system (Creswell 2009; Yin 2009). A mixed methods approach is used with a combination of qualitative interviews and quantitative data (Creswell 2009). Pragmatism is the research paradigm being used for this research project. Using a pragmatic approach puts the focus on discovering answers to the research questions using a variety of methods best suited to achieving the project objectives (Mackenzie and Knipe 2006; Creswell 2009). This puts the results of the research at the forefront with decisions about method selection being based on the strengths of each method relative to the needs of the research project (Rossman and Wilson 1985; Cherryholmes 1992; Maxcy 2003). Since answers to the research questions will be useful to both practitioners and researchers, the pragmatic focus on both practice and theory results in this paradigm working particularly well for this type of research (De Waal 2005; Bryman 2009).

The goal of this research is to answer these questions, "How do participants who are employed by third party organizations collaborate within a fluid organization?" and "What is the role of proximity in these collaborations?" To answer these questions, this thesis explores the literature, the empirical setting, and other aspects of this research topic as three phases of research. Each phase is explored within a separate chapter containing further details about the research design specific to that chapter. Phase 1 is based on qualitative research interviews to explore the empirical setting in more detail and investigate collaboration using Boschma's (2005) five dimensions of proximity. The results and contributions from this phase are described in Chapter 4 and were used as input into the other two phases of the study. Phase 2 uses a longitudinal relational event model to focus on understanding the likelihood of a collaboration event within one subsystem mailing list as influenced by the five dimensions of proximity, network effects, and several empirical setting based controls. Phase 3 builds on the earlier phases by adding interactions between proximity variables to the longitudinal relational event model from Phase 2 to further explore the likelihood of a collaboration event through understanding the interrelationships between proximity dimensions.

Phase 1 begins with an exploration of the Linux kernel as the empirical setting with details about why it was selected for this case study and a justification for considering it to be a fluid organization. In addition to the primary research questions, two additional sub-questions are explored in Phase 1: "How do participants collaborate with people who work for competing third party organizations vs. other participants" and "What is the role of the employer in participation within a fluid organization?" The theoretical framework, proximity theory, was investigated using a series of qualitative interviews and developed using a fluid organization as the setting. As described in the Literature Review, little research has been published that focuses on how open source software developers who are employed by third party organizations collaborate within a fluid organization, so a series of qualitative interviews with employed software developers who are working within the Linux kernel community was conducted as the focus of Phase 1 of this research. The interviews focused on understanding how software developers, who are employed by a wide variety of third party organizations, work together to collaborate within a fluid organization, the Linux kernel, and on understanding how the various dimensions of proximity impact collaboration between these software developers. Much research has been conducted using online data and surveys, but few, if any, studies have used qualitative interviews to collect data about Linux kernel developers. The results from the qualitative interviews contribute to the literature on fluid organizations by providing insight into how the various dimensions of proximity theory influence collaboration between Linux kernel developers, and additionally, the interviews were also used to better understand the empirical setting to drive research decisions for the remaining phases of the research.

Phase 2 of this research was designed to determine the extent to which the dimensions of proximity along with network effects and several empirical setting specific controls contribute to collaboration within this fluid organization. "What dimensions of proximity contribute to collaboration by participants who are employed to collaborate within a fluid organization?" is the sub-question explored in Phase 2. Because the interviews from the Phase 1 indicate that work on the Linux kernel happens within the various subsystems, a specific subsystem was selected for analysis using a longitudinal relational event model to investigate the effect of each variable on the likelihood of a collaboration event within this subsystem.

Phase 3 builds on the analysis from Phases 1 and 2 to look specifically at how the dimensions of proximity are interrelated, operating as complements or substitutes. The research sub-question explored in this phase is, "What is the role of interrelationships between proximity dimensions on collaboration within a fluid organization where the majority of

participants are employed to contribute?" Several potential interrelationships were identified out of the Phase 1 interviews, but were not explored in Phase 2, which focuses on understanding how each variable influences the likelihood of collaboration without including variable interactions. The longitudinal relational event model from Phase 2 is used again for Phase 3, but with variable interactions between proximity variables added to the model to determine the effect of these interrelationships on the likelihood of a collaboration event. The questions and methods for each phase are summarized in Table 3.

3.2. Datasets

While the datasets are described in more detail in each of the phases of this research, it is important to understand that much of the same data are used throughout all three phases and significant effort went into building these datasets. Section 4.2 in Phase 1 provides additional context about the importance of these data within the Linux kernel setting, but an overview is provided here to highlight how the data are shared between the three phases of research. How the data fits into the overall research design is summarized in Table 3.

Contact information for the interview participants was readily available because the Linux kernel is developed online as an open source project, so the email addresses are publicly visible for the participants in this project both on the mailing lists and within the source code repository. The Linux kernel source code is publicly available and was downloaded and stored into a database using the CVSAnalY tool similar to the approach used by López-Fernández et al. (2006), and 20 of the top mailing lists were imported into a database using the MailingListStats tool (Robles et al. 2009).

The list of employer affiliations used in the yearly Linux kernel development reports, see Corbet and Kroah-Hartman (2017) for a recent example, was obtained from The Linux Foundation as a starting point to determine where the participants were employed. However, significant additional, manual data cleaning and validation was conducted to fill in the gaps and gather more precise dates for when software developers who have changed jobs were affiliated with a particular third party organization. To map these employer affiliations into the databases created by CVSAnalY and MailingListStats, an additional tool called Sortinghat was used. This was a time consuming and difficult process that is described in more detail in Phase 2, Section 5.2.3 describing the operationalization of the organizational proximity variable.

Table 3: Research design summary

Primary Research Questions "How do participants who are employed by third party organizations collaborate within a fluid organization?" and "What is the role of proximity in these collaborations?"			
Phase	**Research Sub-questions**	**Methods**	**Main Data Source**
1	"How do participants collaborate with people who work for competing third party organizations vs. other participants" and "What is the role of the employer in participation within a fluid organization?"	Qualitative semi-structured interviews.	16 interview transcripts.
2	"What dimensions of proximity contribute to collaboration by participants who are employed to collaborate within a fluid organization?"	Relational events in a conditional logistic regression model.	PCI mailing list posts.
3	"What is the role of interrelationships between proximity dimensions on collaboration within a fluid organization where the majority of participants are employed to contribute?"	Relational events in a conditional logistic regression model with interactions between proximity variables.	PCI mailing list posts.
Data Shared Across All Phases Employer affiliation data based on Linux Foundation data, mailing list database, and source code repository. Linux-stable source code repository, including MAINTAINERS file for leadership positions.			

For Phase 1, the employer affiliation data along with the source code and mailing list databases were used to determine key attributes for potential participants, including number of commits, maintainership (leadership) positions, and a general sense of project activity. While the participants for the qualitative study were selected strategically as described later in Phase 1 Section 4.3, the researcher's contacts in the industry were useful in getting some of these busy people to respond to requests for interviews. The interviews were stored, transcribed, coded, and analyzed using MAXQDA11 software as described in more detail in Phase 1 Section 4.3.

For Phases 2 and 3 of the research, a quantitative dataset of collaboration events was built using the mailing list database. The source code repository and employer affiliation data were used for calculation of certain variables as described later in Section 5.2.3. While the

databases provided a base of information, to extract the data from the databases and calculate the relevant variables, this researcher wrote thousands of lines of code, mostly in custom Python scripts along with some shell scripts. The dataset was then imported into R where additional R scripts were written by this researcher to transform variables and conduct the statistical analysis, which will be described in more detail in Phases 2 and 3 of the research.

CHAPTER 4. PHASE 1: DEFINING COLLABORATION AND PROXIMITY DIMENSIONS IN A FLUID ORGANIZATION

4.1. Introduction

Fluid organizations with their flexible hierarchical structures are becoming more common (Ashkenas et al. 2002), but this shift from strictly defined and rigid hierarchical structures of more traditional organizations changes how people collaborate. When management and hierarchy can no longer dictate collaboration and coordination of activities, personal connections and other mechanisms need to act as replacements. One advantage of fluid organizations is that individuals can be flexible in how they connect with others to allow for more useful organizational structures that facilitate cooperation and collaboration through flexible movement of ideas, information and resources (Glance and Huberman 1994; Ashkenas et al. 2002). Because fluid organizations are so flexible and diverse with many different types of fluid organizations, the mechanisms of collaboration vary depending on the organization. It follows that understanding more about the mechanisms of collaboration is an important first step when researching collaboration within a fluid organization.

This first phase of the research focuses on understanding the mechanisms of collaboration, including Boschma's (2005) five dimensions of proximity, using qualitative interviews with software developers who are employed by third party organizations to contribute to the Linux kernel, a fluid organization. Much of the research on the Linux kernel comes from online data and surveys with very few studies using qualitative interviews to better understand this setting, since many of these people are busy, and it can be difficult to convince them to give up the time required to do interviews. While difficult, the insights obtained were worth the time required to convince people to be interviewed. In addition to learning more about the influence of the various dimensions of proximity on collaboration between Linux kernel developers, the interviews also helped to understand more about the mechanisms of collaboration within this empirical setting that drove research decisions for the remaining phases of the research.

Specifically, this study uses proximity theory to answer the following primary research questions, "How do participants who are employed third party organizations collaborate within a fluid organization?" and "What is the role of proximity in these collaborations?" by

focusing on these two sub-questions: “How do participants collaborate with people who work for competing third party organizations vs. other participants” and “What is the role of the employer in participation within a fluid organization?” The results indicated that collaboration occurs primarily on the Linux kernel mailing lists between individuals, and while the work tends to support their employer’s products, they are given little direction for their day to day work. It also found consistent evidence that cognitive, organizational, and social proximities impact how people collaborate within this fluid organization, but less support for institutional and geographical proximities influencing collaboration. Additionally, there is also some indication that the various dimensions of proximity are interrelated and working together to influence collaboration.

The next section explores the empirical setting followed by a section describing the methodology, the research design for the qualitative interviews, and a pilot study that influenced the design of this study. The results section uses proximity theory to describe the findings from the qualitative interviews, and the final section provides a justification for the Linux kernel as a fluid organization along with discussion of the relevance of the findings and some concluding remarks.

4.2. Exploring the Empirical Setting: The Linux Kernel as a Fluid Organization

The Linux kernel as a fluid organization meets the broad definition of organizations as structures for facilitating cooperation and decision-making processes (March and Simon 1993). More specifically, the Linux kernel meets the five criteria for a fluid organization proposed and outlined in Section 2.4: organization, affiliation, boundaries, collaboration, and network. Support for defining the Linux kernel as a fluid organization comes across in the interviews, which can be found later in this chapter, and is explored in Section 4.5.1 with details about how the Linux kernel meets each of these criteria.

In a recent keynote presentation during LinuxCon North America 2016, Linus Torvalds talked about how the Linux kernel is a fluid organization: “we have this fairly fluid org chart ... it literally has been changing and it's not even a hierarchy. I would say it's more of a network of people that sometimes goes across boundaries just because you can. Email doesn't really care who you send it to” (Linux Foundation 2016b). This provides some support for classifying the Linux kernel as a fluid organization. Gulati et al. (2012) mentions that central actors, like Linus Torvalds, influence and shape these fluid organizations even without

formal authority, i.e. an employment relationship, over the people participating in the organization.

The Linux kernel is loosely organized as a collection of subsystems. These subsystems are focused on different functionality corresponding to specific sections within the Linux kernel source code, but the reality can be complex with narrowly defined subsystems being contained within broader subsystems. Detailed information about the subsystems can be found in the MAINTAINERS file, and for the purposes of this research, any functionality with a separate mailing list will be considered a subsystem (Kernel development community 2017b). The larger subsystems cover multiple sections of the Linux kernel source code with different maintainers (leaders) responsible for specific areas as defined in the MAINTAINERS file (Linux Kernel Organization 2017). In some cases, an individual maintainer works across boundaries and is responsible for different areas of the source code. In looking at this file and its history of changes, it becomes clear that the Linux kernel is not organized as a traditional management hierarchy, but rather as something more organic and fluid with people moving in and out of leadership positions to suit both the needs of the project and the individuals.

4.2.1. Linux Kernel Case Selection

The Linux kernel was selected for studying the phenomenon of people who are employed to contribute to a fluid organization for several reasons. First, only about 8% of contributions to the Linux kernel are made by unaffiliated software developers who participate on a volunteer basis (Corbet and Kroah-Hartman 2017), so there are a large number of software developers employed by third party organizations. Second, the Linux kernel community is a neutral project where many third party organizations participate, but none of them have control over the project (O'Mahony and Bechky 2008). Linus Torvalds, employed by The Linux Foundation, is the ultimate decision-making authority, but he delegates much of this to other maintainers who work for many different employers (O'Mahony and Bechky 2008; Schneider et al. 2016; Linux Kernel Organization 2017). While The Linux Foundation has no decision-making authority for the Linux kernel (West and O'Mahony 2008), they do employ a few software developers and support the Linux kernel community in other ways. Third, it is a large, established project started in 1991 that contains almost 25 million lines of code contributed by 15,600 software developers from more than 1,400 third party organizations (Corbet and Kroah-Hartman 2017). It is well-established in the literature as a case used to study innovation, knowledge creation, organization, motivation and other

elements of collaboration (e.g. Hertel et al. 2003; Lee and Cole 2003; O'Mahony 2003; Henkel 2006; West and O'Mahony 2008; Puranam et al. 2014).

As mentioned previously in the literature review (Section 2.5), open source software projects often have a triadic role structure with a) the community of individuals, b) a nonprofit foundation, and c) third party organizations that employ contributors (O'Mahony and Bechky 2008). In many cases, the nonprofit foundations formed by members of an open source software project provide entities for ownership of the project, management of assets, and governance (O'Mahony and Bechky 2008). However, the Linux kernel is an exception in this case. For the Linux kernel, The Linux Foundation plays a much more limited role with no authority or ownership over the project (West and O'Mahony 2008). For example, while The Linux Foundation handles administration tasks for the Linux trademark, the trademark is owned by Linus Torvalds, not the foundation (Linux Foundation 2016a). The Linux kernel is controlled by Linus Torvalds along with a group of people referred to as "maintainers" who are responsible for decisions related to reviewing and accepting contributions from other participants into the Linux kernel and can be found documented in the MAINTAINERS file (Schneider et al. 2016; Linux Kernel Organization 2017). The file documenting the list of maintainers changes multiple times per month and provides a fluid structure for decision-making authority within the Linux kernel. While anyone can propose changes or write new code for the Linux kernel and send this code to the mailing list as a "patch", the code can only be incorporated into the Linux kernel by one of these maintainers where it becomes a "commit." As a result of the ownership, governance, and decision-making authority resting with Linus Torvalds and the other maintainers, The Linux Foundation has a supporting role and should not be considered the organization responsible for development of the Linux kernel. In this supporting role, The Linux Foundation does employ several key developers, including Linus Torvalds, while also providing quite a bit of administrative support for infrastructure, conferences, and other non-development activities. The funding for these activities comes indirectly from a variety of third party organizations who are members of The Linux Foundation. However, the official role of organization falls to the fluid community of individuals, Linus Torvalds and the maintainers, who have the final decision-making authority for the project. While the third party organizations who employ these individuals have only an informal role, their participation has been increasing in recent years with contributions from unaffiliated Linux kernel developers at only 8.2% in 2017 (Corbet and Kroah-Hartman 2017) compared to 18.9% in 2010 (Corbet et al. 2010). This triadic role structure, as it applies to the Linux kernel, is summarized in Figure 3.

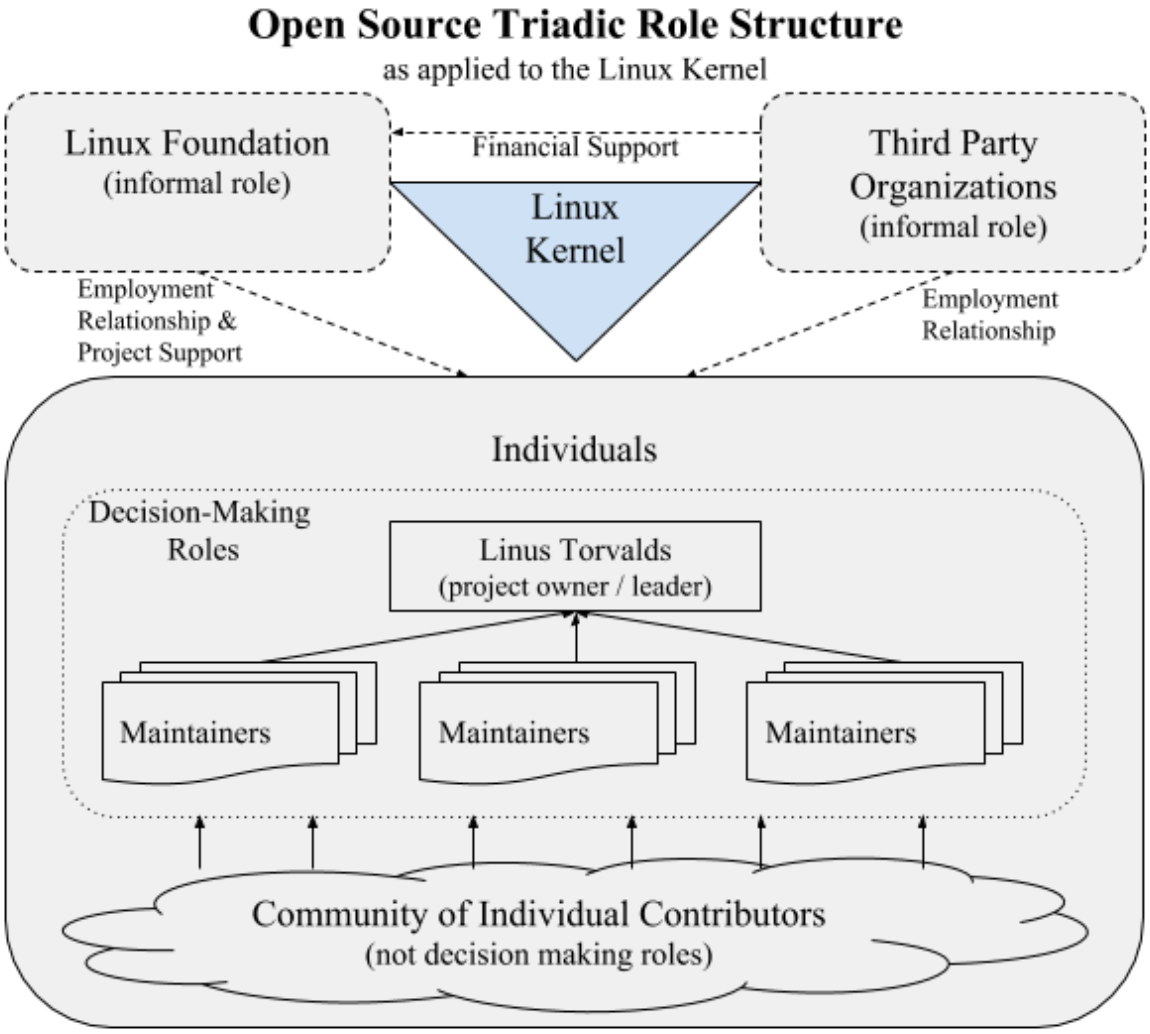

Figure 3: Open source triadic role structure

While it is a single case, the Linux kernel is similar to many other open source projects, especially those projects with large numbers of participants who are employed to contribute. Despite the role of The Linux Foundation acting in an unofficial, rather than a legal capacity, for the project, the foundation still performs many of the expected functions, so there is still a possibility to generalize to similar open source projects and other fluid organizations.

4.2.2. Empirical Setting Summary

The Linux kernel is a fluid organization that provides the setting for this case study of collaboration with proximity as the theoretical framework. As a fluid organization, the Linux kernel facilitates cooperation and has decision-making processes via the maintainers who provide leadership by making decisions about which source code to accept into the Linux

kernel. This leadership occurs in a fluid manner without a formally defined hierarchy, and these leadership roles remain with an individual even if they change employers. Because leaders and other participants contribute to the Linux kernel as individuals, third party organizations have no direct method of participating, but they are increasingly becoming involved by employing people to contribute. As more third party organizations dedicate employees' time to open source software project work, studying participation within these fluid organizations is becoming increasingly important. More details about the overall approach and dataset for this empirical setting can be found in the following section on research methodology.

4.3. Research Methodology

4.3.1. Research Design

A series of 16 qualitative interviews were conducted using a semi-structured, interview guide approach (Patton 2002; Kvale and Brinkmann 2009). As long as the interviewer adheres to the topics specified in the interview guide, the actual questions asked vary from interview to interview as the researcher probes for more detail or adjusts the questions based on previous responses (Patton 2002; Kvale and Brinkmann 2009). Active listening and being attentive to nonverbal cues are critical for the continual decisions that the researcher makes to decide which aspects of the response to pursue in more detail for follow-up questions (Kvale and Brinkmann 2009). Using a pragmatic approach, as described earlier, for a semi-structured interview allows the researcher to focus on answering the research questions, along with how the research results can be used in practice (Kvale and Brinkmann 2009).

The primary research questions, "How do participants who are employed by third party organizations collaborate within a fluid organization?" and "What is the role of proximity in these collaborations?" along with the two Phase 1 subquestions, "How do participants collaborate with people who work for competing third party organization vs. other participants" and "What is the role of the employer in participation within a fluid organization?" were used in this phase. The research questions were translated into the following three objectives for the interview guide. First, gain a better understanding of how software developers who are employed by third party organizations participate and the role that the employer plays in their work, especially with respect to collaboration and competition. Second, learn more about the interactions of software developers who are employed to contribute with a focus on the dynamic between collaboration and competition. Third, gain a

better understanding of the role that the major dimensions of proximity have on collaboration. See Appendix A for a complete copy of the interview guide and more details about the types of questions used to explore each of these three topics.

The sampling strategy was designed to capture a wide variety of information from people highly knowledgeable about Linux kernel development. The sampling strategy used is a mixed purposeful sampling strategy that combines two sampling approaches: maximum variation sampling and intensity sampling (Patton 2002). The study used intensity sampling to find and interview people who embody the phenomenon of interest, which in this case are software developers who are employed to contribute to the Linux kernel, as highly knowledgeable, rich examples of this phenomenon (Patton 2002). The participants were selected from different third party organizations using a maximum variation sampling strategy (Kuzel 1999; Patton 2002) on the employers to include representatives from a variety of sectors with third party organizations of all sizes from very small nonprofits to large, multinational corporations. While the some of the participants were either known to the researcher or were introduced by another professional connection, they were selected purposefully and strategically because they were excellent and rich examples of employed software developers (intensity sampling) coming from different third party organizations selected for variety of characteristics (maximum variation sampling). To determine if someone was an excellent and rich example of an employed software developer in this setting, it was necessary to ensure that the participants had enough relevant experience to provide accurate responses. Thus, the subjects selected for the study had to meet the following criteria: first, Linux kernel developers who are currently or have within the past 12 months been employed to develop code used in the Linux kernel, and second, Linux kernel developers who have commits (changes) that have been accepted into the linux-stable kernel code repository.

To further increase the variation of the sample, interviews from people on the periphery of the phenomenon (Miles et al. 2013) were included as follows: two individuals who actively contributed in the past, but now rarely contribute; someone who has transitioned from being employed at a commercial firm to a work at a nonprofit; and someone in an academic institution. Because of the researcher's familiarity with the phenomenon under study, the maximum variation sampling is also designed to challenge assumptions and preconceived notions to ensure that a broad range of perspectives are included (Kuzel 1999). The number of interviews was not decided in advance, but were conducted up to the point of saturation when responses were no longer yielding much new information (Lincoln and Guba 1985; Kuzel 1999). As early as the pilot interviews, it was clear that despite having

participants from very different third party organizations, the interviews were providing similar information. The decision to use the point of saturation was also made in part because it is quite difficult to get Linux kernel developers to agree to being interviewed, so the process of securing interviews was very time consuming. While this researcher has access to many of these software developers, or other people who know them, as a result of her prior work in the technology industry and her contacts within The Linux Foundation and other participating companies, it was still difficult to secure interviews from people meeting the selection criteria. While this yielded fewer interviews than what might typically be found in other published studies using qualitative interviews as the primary data source, the later interviews were providing few, if any, new insights and were mostly confirming statements made in previous interviews. At 16 interviews, the decision was made to stop spending large amounts of time to secure more interviews when the interviews were not providing additional insights. The participant demographics are described in Table 4.

University Research Ethics Committee approval for the interview process was obtained on 1 May 2015. The interviews were between 30 and 80 minutes in duration and were mostly conducted via online video chat with one in-person interview and two conducted via email. Each interview was audio recorded and either transcribed by the researcher or transcribed by a professional with a word for word verification by the researcher. The transcriptions are verbatim with the following exceptions.

- Excluded filler sounds (mmm, uh huh) and short filler words (e.g. yes, so, you know) where they did not add to the sentence.
- Excluded repeated words / stutters where a word or short phrase is repeated one or more times.
- Included within ( ) for anything significant, non-textual that was important for context (e.g. laughter, explanation of terms)
- Unclear or garbled text that could not be transcribed were designated by [???].

The transcriptions were coded and analyzed using MAXQDA11 software. The coding strategy was primarily data-driven, meaning that the codes were selected by the researcher during the coding process as the transcripts were being read, rather than being pre-selected (Gibbs 2007); however, many of the top level categories for the codes were created to align with the sections and themes found in the Interview Guide. The codes are included in Appendix B for more detail.

Table 4: Participant demographics

Total Interviews	**16**				
Gender	Male 13	Female 3			
Maintainer	Yes 13	No 3			
Employer Type	Corporate 14	Nonprofit 1	Academic 1		
Employer HQ Region	N. America 9	S. America 0	Europe 5	Asia 2	
Participant Region	N. America 9	S. America 1	Europe 5	Asia 1	
Size (num employees)	< 100 2	100 - 999 2	1000 - 9,999 6	10,000 - 99,999 4	100,000+ 2

4.3.2. Pilot Study

The pilot study provided several critical inputs into the research methodology for this work across all three phases of the study. While these inputs are described in greater detail in later sections, several key inputs are highlighted in this section to provide context for how the pilot study influenced the design and methodology of the research. First, and most importantly, the original idea for this research was to focus on productivity to better understand how the productivity of Linux kernel developers is influenced by employment affiliations (or lack of) and network relationships over time as a longitudinal study. However, it became clear that measuring productivity within this setting is problematic, so productivity was deemed to be unsuitable as a result of the responses to interview questions for that topic as highlighted later in the pilot study results. Proximity theory was selected as a replacement based on the results of the pilot study where several dimensions of proximity were mentioned organically many times throughout the pilot interviews despite not directly asking questions about proximity. Second, the interview plans for the qualitative study in Phase 1 of this research were amended to be a shorter duration; conducted online, rather than in-person; had fewer participants; contained updated questions to include proximity theory; and improved wording for some questions. Third, the pilot study results provided greater insight into how Linux kernel developers collaborate, which led to a focus on using mailing lists as the primary

data source for the collaboration event datasets used in Phases 2 and 3 of this research. Overall, the pilot study was an important element of the research process.

During the pilot, the majority of the questions were contained in the first part of the interview focused on testing the interview process, and the remainder of the interview contained a few questions focused on getting feedback on the pilot study and the research project, including questions to learn more about how this group defines collaboration. One interview was conducted in-person at their firm's office in London to test an in-person interview process; however, the rest of the interviews were conducted via online video chat using Google Hangouts or similar services. There were three primary interview topics that were being piloted for use in the main study and two additional topics for the pilot study. Each of these five topics is outlined at a high level below, but the results for each topic are addressed in more detail in the following sections. Interview Topics:

- Employed Developer Participation: Understand participation from affiliated software developers who are employed to participate and the role that their employer plays in their work with respect to collaboration, competition and productivity.
- Developer Productivity: Understand the productivity of Linux kernel developers as it relates to software developers who are employed to participate vs. unaffiliated volunteers.
- Competition and Collaboration: Learn more about the interactions of employed software developers with a focus on the dynamic between collaboration and competition.

Pilot Study Topics:

- Collaboration Definitions: Validate definitions of collaboration to understand how they resonate with Linux kernel developers.
- Pilot Study Feedback: Get feedback on the interview process to better understand what worked well, what did not, and what could be improved before the main body of the interviews were conducted.

Collaboration Definitions

While this was the final section of each interview, it provided useful insights into how the pilot participants view collaboration to introduce the discussion of the results. Each item in the following collaboration framework was discussed with the participants, and they were

asked to describe to what extent each of the items was as an indicator of collaboration within the Linux kernel community:

- Code review / test as designated by the addition of Acked-by, Tested-by, or Reviewed-by lines.
- Provide feedback on patches.
- Provide feedback or comments on a bug.
- Mailing list discussions of a general nature.
- Working on the same file or subsystem.
- Real-time discussions and other collaboration in-person at events.

Code review / test as designated by the addition of Acked-by, Tested-by, or Reviewed-by lines. These are tags added to a patch to signify that a person has acknowledged, tested or reviewed the patch. Most pilot participants said that this is not a good indicator of collaboration, so this will not be used in future phases of the research. One participant said that in most of his work, they do not even bother to use these designations. Another said that it was a weak measure of collaboration because anyone can acknowledge a patch. Three of the four participants talked about how these are good contributions, but they can be done in isolation without ever collaborating with the person who wrote the code.

Provide feedback on a patch. While you can review a person's patch as described previously, the collaboration element is found in actually providing feedback on the work of others, and most code contributions to the Linux kernel come in the form of patches. This was one of the strongest measures of collaboration discussed by the participants in the pilot study. Quotes from three separate participants summarize this well: "the discussion on the patch is the measure of collaboration;" "feedback on patches is very important;" and "definitely a way of collaborating." Most of this feedback happens on the mailing lists, which can be measured quantitatively in future phases. However, people also mentioned that some patch feedback happens in other ways, such as IRC and in informal discussions with people.

Providing feedback or comments on a bug. Many open source software projects make consistent use of a bug tracking system where users file bugs and software developers fix them; however, this is not true within the Linux kernel community. While they have a bug tracker, very few people use it, so bugs tend to be posted to a relevant mailing list, instead. One participant said, "The best kind of feedback you can give to a bug is the patch that fixes it," and others had similar reactions. If a Linux kernel developer provides a patch as part of a bug discussion, the collaboration around that patch would happen as described in the

preceding paragraph. In summary, the pilot participants indicated that this was a weak measure of collaboration.

Mailing list discussions of a general nature. This was consistently mentioned by the pilot participants as one of the strongest measures of collaboration for the Linux kernel community. Because most of the measurable discussions about patches and bugs also happen on the mailing lists, those could be considered a subset of overall mailing list collaboration. This is best summarized with comments from two of the pilot participants: "The 24/7 collaboration that happens is on the mailing list discussions. That's the big measure" and "Email is a hard medium to have an argument in. We are probably better at it than anybody else in the world because we do it all the time."

Working on the same file or subsystem. Results on this measure were a bit mixed, and it seems to depend on the type of file or subsystem along with other factors. One person said that Linux kernel developers who tend to work in one area over time come to trust each other and build collaborative relationships and become a "collaborative group", while other people send one-off patches or bug fixes on those same files or subsystems without collaborating. Another said, "most of the time if you are working within the same file, you need to be collaborating," but also mentioned that work on certain drivers is almost always done without collaborating with other people, since no one else cares about the work on that driver. Someone else said that it was an indicator of collaboration, but "depending on the file, I think you definitely could work in the same files on different pieces … without necessarily dealing directly with the person."

Real-time discussions and other collaboration in-person at events. Along with mailing list collaboration, pilot participants mentioned that this was one of, if not the strongest, methods of collaboration. All of the participants talked about how collaboration happens at events, and most mentioned the Linux Kernel Summit and the mini-summits that happen around specific topics or subsystems at key events where collaboration happens. These events are organized by The Linux Foundation and have typically coincided with other Linux Foundation events. One interesting point made by several people is that the collaboration at events serves different purposes than the daily collaboration that happens over the mailing lists and other online channels. In particular, the summits were mentioned as especially good for brainstorming, planning and solving tough problems or issues. One person mentioned, "I think those are where the breakthroughs actually happen." These in-person events were also described as a good way to build lasting relationships with people and make it easier to work with those people online in the future. A participant described this relationship building in the

following way, “if you've sat and shared a beer with somebody, that makes you better at arguing with them in email, anyway. You don't actually have to have the beer in your hand the next time.”

Employed Developer Participation

This topic was designed to gain a better understanding of how affiliated software developers participate and the role that their employer plays in their work. This topic covered four primary areas:

- Employment situation: how each person started their career as a Linux kernel developer and their current role.
- Reasons for employers to pay Linux kernel developers.
- Employer involvement in day-to-day work.
- Differences between affiliated and unaffiliated software developers.

Employment situation. Three of the four pilot participants started their work on the Linux kernel in a professional capacity as part of their employment as a software developer moving into the role of Linux kernel developer from a variety of positions including Unix kernel development and project manager for Linux. The fourth participant started doing Linux kernel development as a hobby while at university before turning it into paying work during an internship and later as a career in Linux kernel development. In their current positions, the pilot participants all do a variety of different types of software development with the most common being driver development / porting to new hardware, maintaining code (this is a leadership role), and submitting patches (code). Three of the four also talked about providing advice and training for other people at their employer as part of their Linux kernel work. The participants varied drastically in the amount of time spent per week on Linux kernel activities: one to two hours, 10 to 20 hours, 20 hours, and 40 to 50 hours.

Reasons for employers to pay Linux kernel developers. The most common reason for employers to pay Linux kernel developers is to gain influence (prestige, legitimization, and credibility) within the Linux kernel to help set direction in areas of interest to their employer. They also all mentioned that their work on the Linux kernel helps to enable other products for their employer. Other common reasons included feeding information back into the third party organization, visibility / marketing for their employer, and giving back to the community.

Employer involvement in day-to-day work. These four people receive very little direction for their day-to-day work with a high degree of trust from their employers to do useful work without much direction; however, they are all occasionally asked to do some

specific piece of work or to take an interest in a particular area that is important for their employer, but this seems to be the exception, rather than a common occurrence.

Differences between affiliated and unaffiliated software developers. Three of the four participants talked about how the two groups really are not that different, and it can even be hard to distinguish between the groups in some cases. With very few Linux kernel contributions coming from unaffiliated software developers (Corbet and Kroah-Hartman 2017), one participant said, “I think we've all gotten to the point where we sort of assume that everybody's being paid that we talk to on a regular basis.” All of the pilot participants talked about some perceived differences, but there was not much general consensus on those differences.

Developer Productivity

The productivity questions were the ones that all of the participants struggled to answer. While they could give examples of what makes someone productive, none of them provided relevant measurements. Three of the four talked specifically about how productivity cannot be measured in a way that makes sense for the Linux kernel. Some examples they gave, “Sometimes you get one line and sometimes it makes all the difference, whereas sometimes you just spend all of your time rewriting something and you have 87 patches and 7000 lines, and you didn't really do very much.” and “Do you measure it by number of patches? Or do you measure it by number of features? Do you measure it by the value-add of just making sure the releases are working? Because the way Linux development process works is we're all developers, we're all also testers, we test other developers' code, so the way the whole process works is we're all in it together.” One person also mentioned that productivity is especially hard to measure for a mature product, like the Linux kernel.

Competition and Collaboration

The competition section of the interview was designed to better understand how Linux kernel developers interact with other software developers who work for competing third party organizations. This section had two related topics:

- Competition interactions and collaboration.
- Employer guidelines for competitor interactions.

Competition interactions and collaboration. All four of the pilot participants collaborate with their competitors on a regular basis, and they all talked about how within the Linux kernel, they interact with each other on a personal level as individuals. Some examples: “we view our little group of developers as sort of the important team and then who's paid by

who becomes a secondary concern;" "when we're dealing with other Linux kernel developers, we are dealing with them as kernel developers, rather than as competitors, per se;" and "we really leave our companies behind when we are working in open source, otherwise you cannot maintain credibility in open source." One participant even mentioned that one of his employer's competitors "hasn't had as many kernel contributions as we would like." While, all of the participants mentioned needing to be careful about what information is shared to protect confidential employer information, they also talked about sharing information in informal ways or sharing only high-level information and avoiding specific details. Here are a few examples of how this works in practice: "you do get a certain amount of wink and nod and you get terms like major CPU vendor is doing this or that … you certainly see that at conferences" and "some stuff is under NDA, … but everybody knows. As long as you don't say it out loud, you're fine basically ... There is whole spectrum of how you release information that technically you probably shouldn't have done, but in practice, you need to, to get your job done, and yeah, so to a certain extent, people look the other way."

Employer guidelines for competitor interactions. None of the participants mentioned having specific guidelines or processes related to their work with competitors on the Linux kernel, but there were several mentions of more broad guidelines around things like insider trading information, confidential document classifications, or contributing to open source software projects. When asked about balancing what they know about their employer's confidential data with their work on the Linux kernel, three of the pilot participants mentioned that they were not generally privy to some of the more confidential information. One participant who works at a very large company talked about how difficult it was to get access to this type of information. Two of the others just did not generally need it for the type of work they do on the Linux kernel, so getting access to confidential data rarely came up.

Pilot Study Feedback from Participants

There was some good feedback from participants about the pilot. One participant suggested that the researcher spend more time introducing herself at the beginning of the interview to talk about her background in working with open source software. A participant cautioned against doing interviews at conferences where the interviewees would only include people who work for employers that send them to conferences, which would exclude the types of employers that have a reputation for not participating outside of their specific drivers and that tend to take a narrow view and only contribute exactly what they need without taking a more holistic view of the project as a whole. This person suggested reaching out to a couple of

third party organizations who are not typically represented at conferences. Several people also suggested attending some of the mini-summits and looking at collaboration for a few specific subsystems.

Pilot Study Summary

While there are many reasons for employers to pay software developers to contribute to the Linux kernel, some of the most common reasons mentioned by pilot study participants included: gaining influence (prestige, legitimization, and credibility), enabling other products for their employer, feeding information back into the third party organization, demonstrating visibility / marketing for their employer, and giving back to the community.

The pilot study results showed that Linux kernel developers think of each other as individuals with corporate affiliations as a secondary and lesser concern, which begins to explain the extensive collaboration that happens within the Linux kernel between employees who work for competing third party organizations. They also tend to share more information than might be strictly permitted by their employers, and people tend to know quite a bit about what their competitors are doing. This sharing of information may also contribute to the culture of collaboration, which occurs primarily over mailing lists and at in-person events held mostly at Linux Foundation conferences.

Based on what was learned during the pilot, several changes were made to the plans for future phases of the research. First, it was quite difficult to get people to agree to do a 90-minute interview. The original goal of the pilot was to conduct five interviews, but getting a fifth participant proved to be a challenge that was not overcome in the time available for the pilot study. With a bit more focus, the interviews were shortened to about 45 minutes to increase participation.

Second, the original intent was to conduct the rest of the interviews in-person at conferences; however, there were several reasons to conduct the interviews online as video calls using Google Hangout or similar technologies. The point made earlier by one of the participants in the pilot study about how conducting interviews at conferences could bias the results by effectively sampling only within the subset of third party organizations who send their employees to these conferences was an astute point. Also, after thinking more about the conference environment, which can be hectic and quite distracting, conducting interviews online at a time and place that is most convenient for the participant should result in better conversations that are not rushed and higher quality recordings with less background noise for the transcripts.

Third, the initial plan was to conduct 20 to 30 interviews, but it was determined that may be more than what is really needed. The decision was made to conduct interviews up to the point of saturation when responses were are no longer yielding much new information. With the similarities in responses from four pilot participants, despite sampling for maximum variation, saturation was anticipated before reaching the original proposal of 20 to 30 interviews.

Finally, the results from the productivity section of the interviews indicated that it might be impossible to measure productivity for this audience. Traditional measures of software developer productivity usually include output, like lines of code, divided by effort, often person-months (Fenton & Bieman 2015). This would be an achievable measure if some assumptions could be made about the number of hours contributors spend per week on Linux kernel activities; however, with the pilot results of one to two hours per week for one person and 40 to 50 hours per week for another, any assumption made would likely be flawed and open to valid criticism from reviewers. The pilot participants also raised quite a few concerns about measuring productivity, which can be found in more detail in the previous Developer Productivity section. Because of the potential issues with focusing on productivity, several other theoretical frameworks were considered in light of the pilot study results with proximity theory being selected as the most promising fit for the research questions. A careful review of the pilot interviews uncovered that all five of Boschma's (2005) proximity dimensions appeared organically during other interview questions, which helped validate the decision to use proximity theory and allowed content from the four pilot interviews to be included in the final results for Phase 1 of the research. These results are described in more detail in the following section.

4.4. Results

4.4.1. Collaboration of Employed Developers in a Fluid Organization

In order to interpret the results about how proximity influences collaboration, it is important to understand how employed software developers collaborate within a fluid organization, especially one where collaboration occurs online with people located around the world since this changes how people work (Nurmi and Hinds 2016).

Understanding the employment situation of individual participants began to explain the role of the third party organization in collaboration both from the standpoint of organizational proximity as well as cognitive proximity. The participants all described performing a variety

of different types of software development with the most common being driver development / porting to new hardware, submitting patches, fixing bugs, maintaining code (this is a leadership role), and providing advice and training for other internal employees as part of their Linux kernel work. They varied drastically in the amount of time spent per week on Linux kernel activities. In some weeks, an individual may not spend any time at all when devoted to other internal tasks for their employer, and in other weeks, they may spend 40 hours. Seven of the participants consistently spend more than 25 hours per week working on Linux kernel development.

The most common reason for third party organizations to employ Linux kernel developers, mentioned by almost all of the participants, was that their work on the Linux kernel is used to add functionality or improve performance for their employer's software or hardware products or services, and that their employers were also interested in gaining influence (prestige, legitimization, and credibility) within the Linux kernel to help set direction in areas of interest to their employer. Other common reasons included feeding information back to other employees and visibility / marketing for their employer. Less common reasons mentioned by at least three participants were giving back to the community, easier to test / maintain code, and providing advice for others. This helps explain why so many third party organizations employ people to participate and collaborate within this fluid organization.

The participants receive little direction for their day-to-day work with a high degree of trust from their employers; however, they are occasionally asked to do some specific piece of work or to take an interest in a particular area that is important for their organization. One person mentioned that management expected them to "come up with stuff to do on my own, and trusting me to set the direction pretty much myself." Another person who is in a leadership position over other Linux kernel developers, while also occasionally contributing code directly said that "They have a lot of self-initiative; they find problems themselves; they go and solve them … The engineers have a lot of freedom to do things the way they want." In the cases where someone is asked to do something specific, it was described by one participant as being asked to provide "support in the kernel for feature X, Y, Z" with management not caring how it was implemented. This shows that while third party organizations do influence the areas where employees contribute, individuals have quite a bit of freedom to work with different people in a variety of areas.

On a regular basis, most of the participants collaborate with people from various types of institutions: companies (including their employer's competitors), academia, nonprofits, and

unaffiliated hobbyists. They talked about how within the Linux kernel, they interact with each other on a personal level as individuals. One participant said that “I've never really felt that working with people who work for competitors as being a problem ... I think there's an effective social contract of you are willing to help people who work for competitors on the assumption that you'll get about the same amount of benefit from them.” Another participant mentioned that they had recently invited their primary competitor to a Linux kernel meeting hosted by their firm because the competitor was working on similar challenges that both would benefit from resolving. Several of the participants talked about how the Linux kernel has a practice of informal sharing of somewhat confidential information in ways that focus only on high-level information while avoiding details. However, one of the other participants strongly disagreed with this approach, “I wouldn't leak confidential information because I'm not allowed to do that. I mean every worker has a contract with an employer which says that you can't leak confidential information, so even if I trust my mother, I wouldn't exactly leak confidential information to her either.”

This brief discussion of collaboration of employed developers begins to articulate the nuanced role of the third party organization in how employed software developers participate in the Linux kernel, and it highlights some considerations for the role of proximity on collaboration. The contributions to the literature on fluid organizations from these findings can be found in the Discussion section for this chapter, section 4.5. The next section contains detailed findings about the role of each dimension of proximity on collaboration within a fluid organization.

4.4.2. Proximity

The proximity questions in the interview were designed to better understand how participants think about collaboration with other contributors in a fluid organization and how they judge whether other contributors are similar / dissimilar in some respect. Because proximity theory provides a framework for investigating the common ground that participants need for collaboration within fluid organizations, these five dimensions of proximity were used as a framework for the interview questions:

- Cognitive: shared knowledge and experience
- Organizational: Linux kernel as an organization and the employer relationship
- Social: friendships, professional relationships, and the role of trust
- Institutional: software developers employed by third party organizations compared to unaffiliated, volunteer developers.

- Geographical: physical location, time zones, and temporary geographical proximity.

Cognitive proximity

In some ways, Linux kernel developers tend to have diverse backgrounds, experiences, and knowledge, but in other ways, they tend to be quite similar. While most of the participants came from traditional computer science, engineering, and mathematics backgrounds, at least one of the participants had a degree in an unrelated field. One participant talked about the differences and similarities in people's backgrounds:

> "I know great kernel developers that didn't go to university for one reason or the other. I know other great developers that have been you know entrenched in academia for 5 years even after university before they then went into the industry or maybe they're still tied to the academic side of things. So, I think it depends a lot on the individual. I don't think there's any core commonality to that. I think we're all really different in some ways (laughing). And then in other ways, we're all really much alike, right. We're mostly white guys, so there's definitely a set there where we're too much alike, I think."

Some familiarity with the C programming language is required. According to one participant, "A level of experience with the C programming language, a level of awareness of core parts of the kernel, but my experience has certainly been that many people have very little expertise in the kernel in general, and they are much more focused on particular smaller areas."

Because the Linux kernel is made of many different components called subsystems (e.g. audio, networking, memory management), various subsystems require different skills and knowledge. When asked about cognitive proximity in the context of similar backgrounds and knowledge, most people talked about subsystem knowledge and how people working within a particular subsystem tend to have similar knowledge about the topic covered within the subsystem. For example, one person said, "it seems that in that subsystem they all have similar knowledge," and another mentioned, "people who end up working on a specific part of the kernel, specific subsystem, means they're very familiar with that subsystem ... I think that for 70-80% of kernel hackers their background is pretty much the same. The only difference was which subsystem they chose to specialize in."

Many people contribute to several different subsystems, which could provide some diversity of opinion and increases in innovation aligned with Nooteboom's (1999) inverted U-shaped curve as suggested by one participant,

> "I think the similar knowledge we are searching here is about the subsystem, ... but I don't think they should have the same background, the same knowledge, because they will bring different knowledge from different subsystems to improve the current subsystem. So I think that both directions, actually. If people have different

knowledge, the subsystem will improve even more, we'll have ideas all over the place."

The basic knowledge required for all kernel developers is mostly limited to very generic concepts (C programming language and low-level software development), so cognitive proximity seems to be best investigated by looking at collaboration within the various component technologies (subsystems) in this fluid organization. Using similarities in technologies utilized by actors as a measure of cognitive proximity is consistent with the recent literature (Huber 2012; Crescenzi et al. 2016). The results suggest that people collaborating within the same subsystem will have more cognitive proximity, but it may also be important to have people who work across multiple subsystems to bring in new, innovative ideas as supported in the literature by Nooteboom (1999).

Organizational proximity

Most of the participants talked about working on the kernel with other people working for their employer, and some third party organizations have large teams of people devoted to contributing to the Linux kernel. Participants mentioned some similarities, but also quite a few differences between how they interact with other employees vs. Linux kernel contributors outside of the third party organization where they are employed. Several people mentioned that in addition to using the Linux kernel mailing lists and IRC channels, they also use internal channels or in person communication within their employer. Aside from the communication channels, there was not much consensus about how these interactions were different, and participants mentioned a wide variety of differences. For example, a participant from Asia mentioned that it was similar because they communicated mostly via email, but different because they communicated in the local language with other employees. Another participant talked about how other employees feel obligated to accomplish their employer's goals, "The difference is that if I have a timeline, I can tell [Third Party Organization Name], 'I have this deadline, I need you to help me', and I cannot do the same thing in an open source community, not just the Linux kernel." One participant mentioned that they get job satisfaction from being able to make an impact by working with other teams and employees to have "the freedom and flexibility to do upstream work, but at the same time being able to do something that matters internally."

The people interviewed identify with the Linux kernel as an organization, and most of them considered their affiliation with the Linux kernel to be more important than their affiliation with their employer, so they consider themselves a Linux kernel developer first, an employee second. This quote is indicative of the sentiment from most of the others, "At the

core I'm a Linux kernel guy. … At some point, I'm probably going to have the inkling to try something else, and then … I'll be a Linux kernel guy at the next place." There were only a couple of exceptions, including two people who felt that the kernel and their employer affiliations were equally important, one who in the past identified more with the kernel but now identifies more with their employer, and one that felt that their employer affiliation was more important.

This is consistent with previous research from Lakhani and Wolf (2005) who found that 83% of the open source developers they surveyed somewhat or strongly agreed that a primary part of their identity was their affiliation with the hacker community, and from Alexy et al. (2013) that an individual's identification with the open source software community has a positive impact on their support for their employer's engagement in open source software development. This finding provides support for considering the Linux kernel to be a fluid organization. This research also indicates that organizational proximity should influence collaboration within a fluid organization, like the Linux kernel, when organizational proximity is defined as being high when people work for the same third party organization. This is consistent with several recent studies of collaboration that used the organizational affiliation of individual inventors within networks to determine the impact of organizational proximity on collaboration (Cassi and Plunket 2015; Crescenzi et al. 2016).

Social proximity

In all of the interviews, participants were able to name several people that they worked more closely with than others. In some cases, these were strictly professional relationships, but in others, they later developed into friendships. Here is one example:

> "There are many community developers who I feel very comfortable with at a social level, who I will make an effort to see if they're in town, who I look forward to getting to spend time with if we're … at conferences. … In some cases, I'd say they're genuine friendships. These are the people who I know pretty well at a social level. In other cases, it's a level of social familiarity that maybe goes a little bit beyond just having a professional relationship, but is not quite at that level."

Almost all of the participants talked about how existing relationships, both professional and friendship, made it easier to collaborate with other Linux kernel developers. An example from one participant about professional relationships mentioned that "it's helpful because it still remains professional and if you need help from them, or if you have questions or whatever, you can still ask them." An example from another participant highlighted another benefit, "having a functional social relationship ... makes it much easier to feel that asking

them for a favor is justifiable and with a strong expectation that you'll be able to return that favor at some point in the future."

Trust also plays a role in these relationships, especially the professional relationships, and in the day to day work of the Linux kernel. For example, "Linus places a lot of trust in you ... and I, in return, when people send me stuff, we'll rely a lot on trust. … it factors in a lot into how you approach the patches or code that is sent to you." Here is a similar example from another participant, "I think it's something that evolves with time after you see someone's work for a few months, you kind of know what he's good at and what he's not good at, and then you can easily validate him professionally, you know which aspects you can fully trust him and which you might question."

This research suggests that professional and friendship relationships play an important role in this setting suggesting that collaboration is facilitated by higher levels of social proximity. This is consistent with the many proximity studies showing the importance of social proximity on collaboration by using co-inventors on patents to measure social proximity within networks of individuals (Cantner and Graf 2006; Ter Wal 2014; Cassi and Plunket 2015; Crescenzi et al. 2016). To further reinforce the idea that social proximity is important for collaboration in this fluid organization, these interviews indicate that trust comes from past experience interacting with certain Linux kernel developers, similar to Boschma's (2005) descriptions of trust in social proximity.

Institutional proximity

In most cases, the participants are not concerned about whether the person works for a corporation, a nonprofit, an academic institution, or whether they are an individual without an institutional affiliation. For example,

> "But whether they are fresh out of school in Hungary or whether they've been working for Google for 20 years or somewhere else. Personally, I don't really care. I think it's a lot more about how it is to work with that specific person than the origins of where they're from or who they're working for. I don't really care about that a lot."

For contributors who use employer email addresses, their affiliation is quite obvious, unlike contributors using personal email addresses. As a result of not caring much about the affiliation, in some cases, the participants interviewed don't always know the affiliation of the people they are collaborating with. One participant mentioned, "If I don't know them personally or if they don't use their work email, I don't necessarily know." However, several participants mentioned that they do generally know the affiliations of the people that they collaborate with on a regular basis. For example, "For most people, you can just see it on their

email address who they work for, but if you start to have more contact with somebody than just a few patches, then I mean it's kind of in the general interest to know why they are there essentially." From another participant, "If they're someone I've worked with previously, I generally have a reasonably good idea, and certainly if they are someone that I've met or spent time with at a conference, then there's a much higher probability that I'm aware of who they work for if only because paying attention to the employment situation seems like the socially polite thing to do."

The one case where institutional affiliation mattered for some of the participants is for those people without an institutional affiliation who are participating as volunteers, instead of as a part of their employment. Five of the participants mentioned giving volunteer software developers a bit more leeway and help than they would for people who are being employed to do similar work. For example, "There's certainly an element of spending time working with someone who's just doing this in their spare time or doing this at university means that they are potentially someone who's worth spending time trying to recruit if we're in a situation where we're looking for further kernel developers." Another participant says that "I would be a bit more forgiving on not necessarily dotting every i and crossing every t. ... I would give them maybe a little bit more kid glove treatment if I knew they were not being paid to do it."

The interviews indicate that institutional proximity should have little to no effect on how people collaborate within this fluid organization with the possible exception of volunteers / unaffiliated Linux kernel developers who might be given a bit more leeway when collaborating with others.

Geographical proximity

For the participants interviewed, physical location is not important, since collaboration happens on mailing lists where people respond asynchronously. For example, "It doesn't matter where people work, I think that's the primary point." and "Mostly I use email because it's the most persistent and geographically distributed way of handling things, and also it has a natural archiving." One went as far as saying that "The Linux community doesn't care where you're located, ever. You can be on the moon as long as you have a good internet connection."

However, some people are aware of time zones for key collaborators, but most also claimed that time zones do not really matter that much. For example, one participant said, "I know really well which person is in which time zone," and he uses this information to know when to expect replies, but he also said that he doesn't really work more closely with people who are online at similar times. Another participant mentioned that "Similar time zones can be

more helpful because I can get a reply immediately. But it is not super important in my opinion." Several people mentioned that by using mailing lists, which are email-based and by nature mostly asynchronous, it makes it easy to collaborate with people across many time zones. For example, one person said that "email is this kind of store and forward technology where I don't really think about time. I just shoot the message, and hopefully something will come back at some point."

Temporary geographical proximity, defined as short-term travel to a common location, often conferences and meetings, as opposed to permanent co-location (Torre 2008) appears to have a role in collaboration within the Linux kernel. All of the participants mentioned in-person collaboration at conferences and other meetings. For example, "I think it was easier to build the kind of trust relationships I was talking about earlier with in-person interaction and spending time with people at conferences ... having a better understanding of a person as a real thing, rather than an email address makes a surprising difference in the kinds of mental model of interaction with them." Another participant mentioned, "and then we have conferences and things where you really can sit down with a beer and hash things out, and come to a consensus … I think the Linux Kernel Summit, we do every year is massively useful for that kind of thing."

The results for geographical proximity are a bit mixed. The results indicate that physical location is mostly irrelevant in this fluid organization, which aligns Torre's (2008) idea that physical location tends to disappear when interactions between people occur entirely online. However, when collaborating with others online, the results showed that time zones might be relevant in this setting, which is consistent with O'Leary and Cummings (2007) who used time zones to indicate overlapping online work hours as one of several elements of geographical proximity. While participants claimed that time zones didn't matter, the research showed that they sometimes keep them in mind and that they may get replies more quickly from people who are in similar time zones, so the results are unclear to what extent they may or may not play a role in collaboration. These findings do show a relationship between collaboration and temporary geographical proximity with attendance at conferences seen by the participants as an important part of collaboration within this fluid organization.

Interrelationships Between Proximity Dimensions

Some dimensions of proximity have been demonstrated in the literature as complements or substitutes for other dimensions as described in more detail in Section 2.2.2 of the Literature Review. While the interview questions did not ask explicitly about

relationships between dimensions of proximity, a few relationships emerged organically, but consistently, in participant responses.

The findings indicate that social proximity and geographical proximity are interrelated. Attendance at key conferences is one of the primary ways that Linux kernel developers build social relationships with each other. This temporary geographical proximity is used to build both professional and friendship relationships, which helps facilitate collaboration over larger geographic distances after returning from the conference. For collaboration over the long-term, social proximity is most likely acting as a substitute for spatial geographical proximity. This concept is nicely summarized by one of the participants who said that "another great thing is meeting people face-to-face at conferences ... I like forging new acquaintances and friendships along the way, but the other thing is that it really smooths over the working relationship on the mailing list."

Geographical proximity based on physical location is also related to social proximity in cases where people live near other Linux kernel developers and to organizational proximity when people work in the same office. In both of these situations, interviewees mentioned getting together in person with people for professional discussions or in more informal settings, like meeting for coffee. This complex interrelationship between social, geographical, and organizational proximity allows people to collaborate and work more closely with each other, which is demonstrated by this quote from one of the interviews: "I have one person in my company who lives in the same city ... I work more closely with him because he's my mentor in the company, and sometimes we go out to drink a beer and talk about the company." Geographical, social, and organizational proximities seem to be complements in this case.

Cognitive and organizational proximities are also interrelated in a complementary fashion in this fluid organization. Many participants talked about how their work in specific subsystems (cognitive proximity) related to the work they do for their employer (organizational proximity). In other words, the knowledge required to contribute to a subsystem often complements the work required for their employer. In some cases, subsystems are directly based on a particular third party organization's technology and many of the people working in that area are employees for that third party organization. For example, several subsystems are based on IBM's S/390 processor technologies and are currently maintained by over a dozen IBM employees (Linux Kernel Organization 2017). This shows that sometimes people who are working for the same employer would also be working in the same areas of the code, and people with similar knowledge may become grouped together in one organization. This complementary relationship is not unexpected when you

look at recruiting practices of these third party organizations, which often hire people because of specific knowledge of a subsystem. For example, when asked about how their employer hires Linux kernel developers, one interviewee said, “we also are interested in certain areas of the kernel so if we have somebody that is already participating, … active contributor in that area.”

4.5. Discussion

This chapter looked at participants who are employed to collaborate within the Linux kernel using interviews with a varied and diverse sample of people. The participants identify with the Linux kernel as an organization, and most of them valued this affiliation with the Linux kernel over their affiliation with their employer. These participants receive very little direction from their employer for the day to day work on the Linux kernel, but they are occasionally asked to do specific work. Since participants’ work almost always supports their employers’ products, the third party organizations employing these participants do have some influence on the type of work being performed within the Linux kernel. In general, Linux kernel contributors claim to interact with each other as individuals, rather than focusing on employer affiliations, and participants collaborate with their competitors on a regular basis. This collaboration occurs primarily on the Linux kernel mailing lists.

4.5.1. Linux kernel as a Fluid Organization

As indicated in the interviews and the empirical setting section (4.2), the Linux kernel is a fluid organization meeting the five criteria proposed and outlined in Section 2.4: organization, hierarchy, boundaries, collaboration, and network.

Organization. The Linux kernel meets the four criteria for an organization from Ahrne (1994) and Dobusch and Schoeneborn (2015) described in Table 1: affiliation, collective resources, substitutability, and recorded control. The interviews showed that participants identify with the Linux kernel as an organization, and most of them valued this affiliation with the Linux kernel over their affiliation with their employer. Participants make decisions on the mailing lists about contributions to the source code repository as a collective resource. Substitutability is demonstrated by looking at the history of the MAINTAINERS file to see that leaders are removed, added, and replaced frequently. Recorded control can be seen in the archives of the mailing lists and source code repository where decisions about which individuals’ source code is accepted (reward) or rejected (sanction) are recorded.

Hierarchy. In fluid organizations, if present, the hierarchy evolves as needed in a flexible manner and most likely emerges organically from the network. As mentioned earlier, Linus Torvalds has talked about how the organization is fluid and not really a hierarchy (Linux Foundation 2016b). While it is not a hierarchy in the traditional sense, the MAINTAINERS file shows that there are some hierarchical elements with defined leadership for certain areas of the source code. This leadership is fluid and changes are frequently made to this file as people move in or out of maintainer positions with the network likely influencing who moves into leadership roles. The interviews indicated participants treat each other as individuals and stay focused on the work, which demonstrates that relationships between people are flexible based on need, rather than being enforced by a top down hierarchical structure.

Boundaries. Members collaborate across flexible or non-existent boundaries between sub-groups or teams in fluid organizations. Again, Linus Torvalds mentioned that people work across boundaries within the Linux kernel (Linux Foundation 2016b). This can also be seen in the MAINTAINERS file where the same maintainer is in some case responsible for several very different areas of the code, thus requiring the maintainer to work across boundaries and on multiple email lists. The interviews also indicated that people often work across boundaries on multiple subsystems in addition to working closely with others across organizational and institutional boundaries as well.

Collaboration. Within fluid organizations, individuals collaborate organically facilitated by finding common ground. Throughout the interviews, people talked about how existing relationships, both professional and friendship, made it easier to collaborate with other Linux kernel developers, showing that people find common ground through social proximity. Participants also talked about how people with cognitive proximity through similar, shared knowledge collaborate in various subsystems, and how they collaborate within the Linux kernel with people working at the same employer for organizational proximity.

Network. Networks in fluid organizations have a pronounced role in collaboration between individuals that influences changes within the hierarchy. Linus Torvalds described the Linux kernel as more of a network than a hierarchy (Linux Foundation 2016b). In the interviews, participants talked about how they collaborate more closely with some people and that trust plays a role in their relationships with other contributors, which indicates that network structures influence collaboration. These relationships can lead to evolution within the hierarchy as trusted people move into maintainer positions.

4.5.2. Proximity

The results of this study found consistent evidence that cognitive, organizational, and social proximities impact how people collaborate within this fluid organization. Cognitive proximity can be determined by investigating collaboration within the various subcomponent areas within this fluid organization. The results indicate that cognitive proximity is high for people collaborating within the same subcomponent areas, but it may also be important to have people who work across multiple areas to bring innovative new ideas aligned with Nooteboom's (1999) findings about the U-shaped curve of cognitive proximity. The findings showed that in most cases, participants worked in different ways with other employees at their third party organization within this fluid organization, so when organizational proximity is measured by people working for the same employer, it should have some influence on collaboration within a fluid organization, like the Linux kernel. This research suggests that collaboration is facilitated by higher levels of social proximity as indicated by how professional and friendship relationships play an important role in this fluid organization and how trust comes from past experience interacting with certain Linux kernel developers, similar to Boschma's (2005) descriptions of trust in social proximity.

The results provided less support for institutional and geographical proximities influencing collaboration within this fluid organization. With the possible exception of volunteers / unaffiliated Linux kernel developers who might be given a bit more leeway when collaborating with others, these findings indicate that institutional proximity should have little to no effect on collaboration within this fluid organization. The results for geographical proximity indicate that participants consider physical location to be irrelevant because of the online, virtual nature of this fluid organization, which aligns with Torre (2008). However, the results were unclear about whether time zones are relevant for collaboration in this setting, since participants claimed that they did not matter, but also said that they may get replies more quickly from people who are in similar time zones. These findings do show that temporary geographical proximity is relevant, since attendance at conferences is seen by the participants as an important part of collaboration within this fluid organization.

That dimensions of proximity are often interrelated and can operate as complements or substitutes has been widely described in the existing body of literature on proximity (e.g. Boschma 2005; Balland et al. 2015; Crescenzi et al. 2016; Heringa et al. 2016), thus highlighting the importance of looking at how proximity dimensions are related, rather than looking at each one only in isolation. The findings indicate that there are several relationships between dimensions of proximity in this fluid organization. Cognitive and organizational

proximities are interrelated as work in specific subsystems (cognitive proximity) is complementary to the work required for an employee's specific role within a third party organization (organizational proximity), which is consistent with Boschma (2005) who described a complementary relationship between cognitive and organizational proximities. Social proximity and geographical proximity are interrelated because attendance at conferences is a primary way that Linux kernel developers build both professional relationships and friendships that facilitate lasting collaboration over geographical distances long after both attendees have returned home from the conference, thus social proximity is likely acting as a substitute for geographical proximity. Geographical proximity based on physical location is also related to social proximity in cases where people live near other Linux kernel developers and to organizational proximity when people work in the same office, thus allowing for professional discussions and collaboration in more informal settings. In this case, geographical, organizational, and social proximities may have complementary relationships. The relationships between geographical, social, and organizational proximity are consistent with findings from Breschi & Lissoni (2009) and Boschma's (2005) findings that geographical proximity may act as a substitute or complement to the other dimensions of proximity.

4.5.3. Summary

In sum, this research shows that proximity theory can be used effectively as a theoretical lens when considering collaboration in fluid organizations and that the Linux kernel can be considered a fluid organization. Almost all of the people interviewed considered their affiliation with the Linux kernel as an organization to be more important than their affiliation with their employer. This relationship of organizational affiliations with third party organizations employing participants to collaborate within a fluid organization shows the importance of considering organizational and institutional proximities. This research also provides support for the idea that social, cognitive, and institutional proximities should be considered separately when looking at intraorganizational proximity within a fluid organization, unlike Knoben and Oerlemans (2006) who found that these could be included within organizational proximity for interorganizational collaboration. With intraorganizational collaboration in fluid organization, the individual relationships between people (social) and their knowledge (cognitive) are not necessarily directly related to their employer and should be considered separately from organizational proximity.

These Phase 1 results contribute to the literature on fluid organizations in several important ways. First, this research reviews the diverse literature surrounding the concept of fluid organizations and uses it to propose five criteria to determine whether an organization is a fluid organization, and then uses this criteria to demonstrate that the Linux kernel is a fluid organization. The five criteria for fluid organizations are organization, flexible hierarchy, flexible boundaries, organic collaboration, and pronounced role of networks. The Linux kernel can be described as an organization with organizational affiliation, collective resources, substitutability of resources, and recorded control. The hierarchy is flexible and evolves as needed with people collaborating organically across boundaries while relying on their network for collaboration. Thus, the Linux kernel is shown to be a fluid organization.

Second, the results from Phase 1 demonstrate that dimensions of proximity can be used to better understand the common ground between participants that is required for intraorganizational collaboration within fluid organizations. Collaboration within traditional organizations can be enforced by a rigid hierarchy. In contrast, fluid organizations have flexible boundaries and evolving structures where participants must rely on common ground to facilitate effective collaboration. This research shows that proximity theory can be used to explore this common ground using Boschma's (2005) five dimensions of proximity and their interrelationships. This contributes to the literature on fluid organizations by demonstrating that proximity theory can be used to understand the common ground between participants that is needed for intraorganizational collaboration within fluid organizations.

Third, this research adds to the body of knowledge on fluid organizations by demonstrating that third party organizations have an impact on collaboration. While this concept falls within organizational proximity, it was identified as a gap in the Literature Review section and is significant enough to be highlighted here as a specific contribution. One role of the employer is to direct the work of their employees, which can influence how employees spend their time. To account for this influence, affiliation with third party organizations should be factored into the analysis of collaboration in fluid organizations. During the interviews, participants talked about how their employers requested work on specific areas of the Linux kernel, thus influencing the areas of their collaboration with others. This shows that employer affiliation influences collaboration and should be included in the analysis of collaboration within fluid organizations. All three of these contributions are also discussed in Section 7.1 in the Conclusions chapter along with contributions from the other phases of this research.

Fourth, this research can impact other types of fluid organizations. The interviews indicated that cognitive, social, and organizational proximities are important for collaboration, while institutional and geographical proximites have little to no influence on collaboration. Other fluid organizations, and possibly some types of traditional organizations, are likely to have similar findings. This can be illustrated in the context of a few examples: research organizations, collaborative editing projects, and social collectives. In fluid organizations doing collaborative research (Ahuja and Carley 1999), which may include research groups with participants from industry, academia, or both, organizational proximity is likely to influence collaboration along with social proximity based on previous interactions and cognitive proximity from areas of research expertise. For research groups that communicate primarily online using email or other asynchronous methods, geographical proximity would be expected to have little influence on collaboration; however, geographical proximity might influence collaboration in research groups where most of the participants are in a single location. The impact of institutional proximity on collaboration might depend on the extent to which participants from both industry and academia work together within the organization. Collaborative editing projects, like Wikipedia (Lerner and Lomi 2017), where participants come together to create online resources, have many similarities with this setting. Due to the distributed nature of the work with editors coming from locations all over the world, it would be expected that geographical proximity has little influence on collaboration while social proximity via discussions about edits (talk pages in the case of Wikipedia) and cognitive proximity based on areas of expertise would be expected to influence collaboration. Collaborative editing projects, like Wikipedia, that maintain a culture of avoiding conflicts of interest would likely see little influence on collaboration from organizational proximity and institutional proximity. Similarly, in social collectives, like Anonymous (Dobusch and Schoeneborn 2015), cognitive and social proximities would be expected to influence collaboration as people who have worked together in the past and people with similar knowledge are likely to collaborate. Geographical proximity would be expected to have little influence for social collectives with mostly online participation. Like with the research organization example, the impact on collaboration from organizational and institutional proximities may depend on the context and whether those affiliations are known and / or relevant for the particular collective.

While limitations will be discussed in more detail in Chapter 7 (Conclusions), one limitation of this study is that it is based solely on interview data, and the results were not confirmed using other types of data. Because open source projects work in the open with

publicly archived conversations in mailing lists and source code repositories, these data sources are used to validate some of this research during the remaining phases of the study in Chapters 5 and 6.

CHAPTER 5. PHASE 2: ANALYZING THE IMPACT OF PROXIMITY AND NETWORK STRUCTURE ON COLLABORATION

5.1. Introduction

In traditional organizations with rigid hierarchies, collaboration can be enforced by leadership within the hierarchy. In contrast, fluid organizations must rely on finding common ground to facilitate collaboration within the evolving structures and across flexible boundaries. Proximity theory provides a framework for understanding the common ground between individuals using Boschma's (2005) five dimensions of proximity. The flexible structures within fluid organizations evolve as needed to meet the ongoing requirements of the organization with changes driven by people with formal or informal roles within the fluid organization to facilitate collaboration (Glance and Huberman 1994; Gulati et al. 2012). With networks playing a role in collaboration between individuals and influencing changes within the hierarchy, it is important to look at collaboration within fluid organizations longitudinally as a network phenomenon. Using proximity theory to find common ground and understanding the network influences provide a structure for understanding collaboration within fluid organizations in a longitudinal manner.

Like with many open source software projects, collaboration within the Linux kernel occurs using mailing lists that are publicly available for anyone to join. As early as the pilot study interviews from Phase 1 of this research, it was clear that mailing lists are the primary collaboration tool for the Linux kernel, and mailing lists as the tool for collaboration continued to be mentioned by participants throughout the study. The kernel documentation states that if a participant wants to contribute source code into the Linux kernel, the code must be submitted in the form of a patch to the relevant mailing list where other Linux kernel developers can review and comment on it (Kernel development community 2017a). Patches that pass this level of scrutiny and move to the next stage will then be accepted by a maintainer for inclusion into areas of the source code used for testing by a larger number of people (maintainer and -next trees) before eventually being accepted (or declined) for inclusion into a released version of the kernel source code (Kernel development community 2017a). The mailing lists provide a way for all of these developers to collaborate on a wide

variety of source code contributions regardless of physical location, employer, specific areas of technical expertise, or other factors.

The results from the Phase 1 qualitative interviews in the previous chapter indicated that several dimensions of proximity influence collaboration between Linux kernel developers who are employed to participate. This second phase of the research builds on those results to explore each dimension of proximity and gain additional insights into their influence on collaboration. Because collaboration within fluid organizations is a network phenomenon, the network influences are explored along with the dimensions of proximity to provide a more complete understanding of collaboration over time within this setting. As an open source project, the Linux kernel mailing list archives and source code contain rich data for a quantitative study that uses proximity theory and network measures to understand collaboration within a fluid organization, thus addressing the limitation of using only interview data from the previous qualitative work in Phase 1. Phase 2 of the research in this chapter answers the question, "What dimensions of proximity contribute to collaboration by participants who are employed to collaborate within a fluid organization?"

In answer to this research question, Phase 2 demonstrates that four out of the five dimensions of proximity influence the likelihood of collaboration within fluid organizations with only geographical proximity providing no evidence of an impact on the likelihood of collaboration. Additionally, it is also clear that several setting-specific variables and network variables also influence the likelihood of collaboration between two participants. The remainder of this chapter is organized as follows. The next section contains a description of the methods used in this phase, including information about the data, variables, relational event model, and model estimation using a conditional logit model. The following section contains details about the models and results for how proximity, network, and setting-specific variables influence the likelihood of collaboration. The final section contains a discussion of the findings.

5.2. Methods

5.2.1. Data

Linux kernel development happens in cycles with regular releases and periods of time right after the release where new code is included into the upcoming release, called a merge window, to give other people time to test the new code before the next release. To align with these release cycles, the dataset contains mailing list messages over a period of almost two

years using the 3.12 release of the Linux kernel on 2013-11-03 as the start date and the 4.3 release on 2015-11-01 as the end date. During this time period there were 12 Linux kernel releases with 63 days as the median time between kernel releases, and all releases occurred on a Sunday.

As discussed in the previous chapter, the Linux kernel is composed of numerous subsystems allowing groups of people to collaborate on specific sections of the source code. This subsystem collaboration occurs over more than 240 separate mailing lists. Each of these mailing lists are included in the MAINTAINERS file following an "L:" designation, which indicates that it is the mailing list relevant for a specific section of the code, and patches or other discussions involving this section of code should be sent to the relevant mailing list (Linux Kernel Organization 2017). Because collaboration on specific subsystems happens on specific mailing lists, the decision was made to select one mailing list for study.

To select one list, the process started with the top 25 subsystem mailing lists with top defined as having been mentioned in nine or more sections in the MAINTAINERS file as of March 3, 2016 when the mailing lists were initially downloaded and processed. Eight lists were excluded because the archives were incomplete or not available. An additional four lists were excluded because they targeted subsystems based on the technology of a single third party organization and were not good candidates for looking at collaboration between individuals who work for a variety of third party organizations. The ARM list was excluded, since it overlaps heavily with other lists due to the common practice of copying the ARM list when emailing other lists. The main kernel mailing list, LKML, was also excluded, since it is not a subsystem list, and even the official Linux kernel documentation points out than many developers do not read the LKML due to the high volume of email on that list (Kernel development community 2017b), and it even goes as far as recommending that people bypass the inbox for the LKML by sending these emails directly to a folder to avoid seeing all of the messages while using email filters to only see posts on topics of interest (Kernel development community 2017a).

Out of the 12 remaining subsystem mailing lists, the linux-pci@vger.kernel.org mailing list, which is where Peripheral Component Interconnect (PCI) drivers for the Linux kernel are developed, was selected for two primary reasons. First, the PCI mailing list is widely used. It is one of the top 20 mailing lists as measured by the number of times it is listed in the MAINTAINERS file (24 times), and it has over 400 subscribers (vger.kernel.org 2016). Second, the PCI mailing list is a typical example of these top lists as defined by being closest to the median for both the overall number of replies and the time it takes for people to reply to

a message. Almost all mailing list archives suffer from some incompleteness often due to encoding errors or other data errors that prevent messages from being stored or retrieved from the archive. For the 12 mailing lists used in the selection, the incompleteness scores ranged from 3.2% to 8.9% where the details associated with the original message being replied to was not available in the dataset. For the PCI mailing list, the incompleteness score was 8.7%.

The dataset focuses on collaboration between individuals with collaboration operationalized as replies to mailing list posts. This measure was selected based on the results of the qualitative interviews and kernel documentation (Kernel development community 2017a), which both indicate that collaboration on contributions in the form of patches occurs as mailing list discussions before the source code is accepted into the Linux kernel. The event history dataset was constructed using the 10,513 replies to messages on the PCI mailing list over the period from 2013-11-03 to 2015-11-01. Each of the replies in the event history dataset creates a network tie indicating a collaboration event between the person replying to the message, the ego, and the person being replied to, the alter. The data contain 654 total actors made up of 567 egos who replied to messages that were sent to the list by 574 alters with quite a bit of overlap, since most actors are both egos and alters for different collaboration events. The dataset is summarized in Table 5.

Table 5: Phases 2 and 3 dataset summary

Mailing List	Peripheral Component Interconnect (PCI) drivers at linux-pci@vger.kernel.org
Timeframe	12 releases from 2013-11-03 (3.12 release) to 2015-11-01 (4.3 release)
Events	10,513 replies: network tie / collaboration event
Actors	654 total actors: 567 egos and 574 alters

5.2.2. Relational Event Models

Butts (2008) introduced a flexible relational event framework that can be used for modeling events or actions in social settings using likelihood-based inference for effects with complex interdependence that influences behavior. Relational event models are based on relational events, or actions generated by a sender directed toward a receiver and are represented by sender, receiver, action type, and time (Butts 2008). These models assume that past events create the context for a current action, and when this new action occurs, the process begins again with that new action added to the history of previous actions to be considered for the next action (Butts 2008). Mailing list replies with a sender, receiver (person

being replied to), and time stamp for each message provide the data required for relational event models to explain the likelihood of a collaboration event between two people given the influence of various effects. As introduced in the Literature Review, there are other network models (exponential random graph models and stochastic actor oriented models) that are suitable for longitudinal analysis of network data, but the relational event model was a better choice for analyzing a full sequence of collaboration events.

Predicting events in an ordinal sequence is product of multinomial likelihoods (Butts 2008). The ordinal model can be estimated using conditional logistic regression, and one option is to use a Cox regression estimated using maximum likelihood estimates (Quintane et al. 2013). While the dataset contains exact time stamps, the ordinal version of Butts' (2008) model was selected for computational reasons. The probability of a collaboration event between two individuals, *i* and *j* can be estimated using a conditional logit model as described by Greene (2012) and used in a similar study by Cassi and Plunket (2015)

$$P_{ij} = \frac{exp(x'_{ij}\beta)}{\sum_{j=i}^{J} exp(x'_{ij}\beta)}$$

where x represents a vector of covariates and β represents a vector of the parameters to be estimated.

The model was implemented in R using clogit within the survival package, which makes use of coxph (Cox proportional hazards regression model), since the dataset was too large to use the "relevent" package described in Butts (2008). It is important to note that the coxph function used by clogit in R scales and centers the variables, which leads to more numerical stability without changing the results of the regression analysis, so the raw variables described in the next section are used as inputs into clogit, but the outcome has scaled and centered variables.

5.2.3. Model Estimation and Variables

Dependent Variable and Estimation

The dependent variable is the collaboration event operationalized as a reply to a message on the mailing list to determine what factors influence the likelihood that an ego will reply to a message previously posted by an alter on the mailing list. As shown in Figure 4, the PCI mailing list gets anywhere from only a few posts to over 140 posts per day, so there is quite a bit of variability from day to day. Mailing list replies are also not equally likely over

the entire dataset i.e. it is highly unlikely that a two year-old message will ever receive a reply while recent messages are much more likely to receive replies. To control for this temporal variation, realized events should be compared only to recent messages that are likely to receive a reply with recent messages defined as seven days for two reasons. First, each weekday has more than four times the number of messages posted on the PCI mailing list as compared to a weekend day (see Figure 5), so a time period that is a multiple of seven is required to take this variance into account. Second, most replies on the PCI mailing list occur within a short time from the message being replied to (median is 7.2 hours and third quartile is 1.5 days), and 89.3% of replies to original messages on the PCI mailing list are sent within seven days of the original message making seven days a reasonable choice given the peculiarities of this empirical setting.

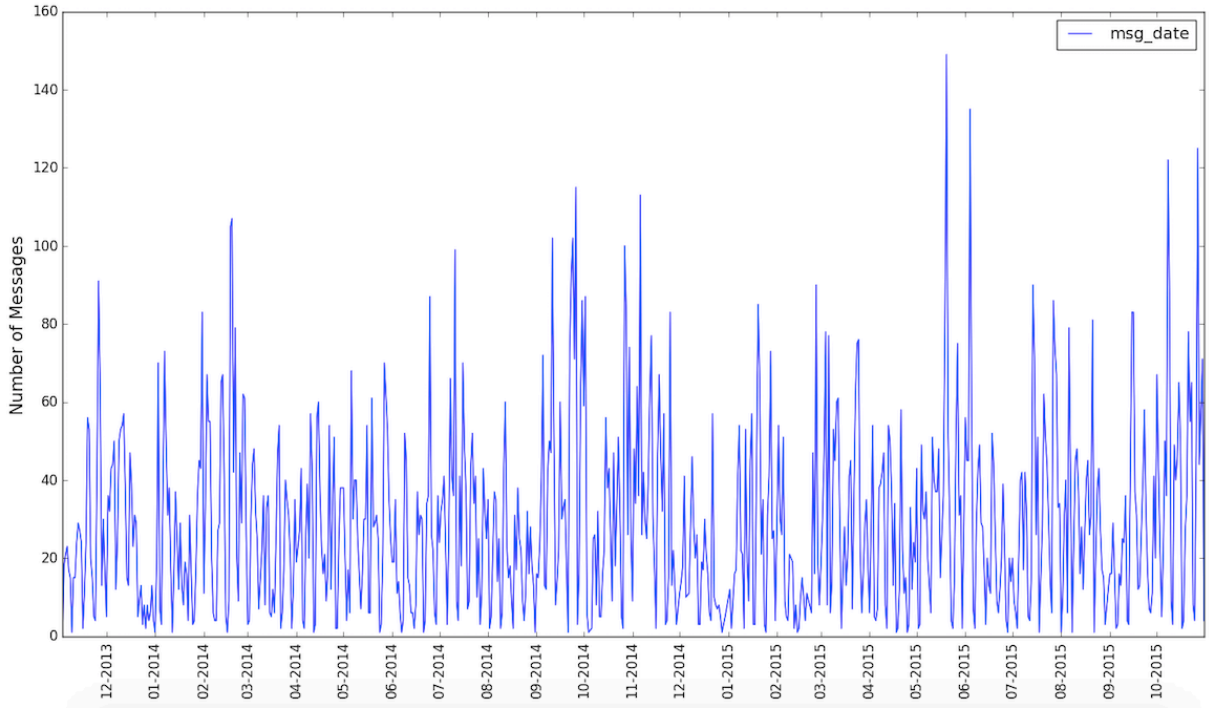

Figure 4: Messages per day

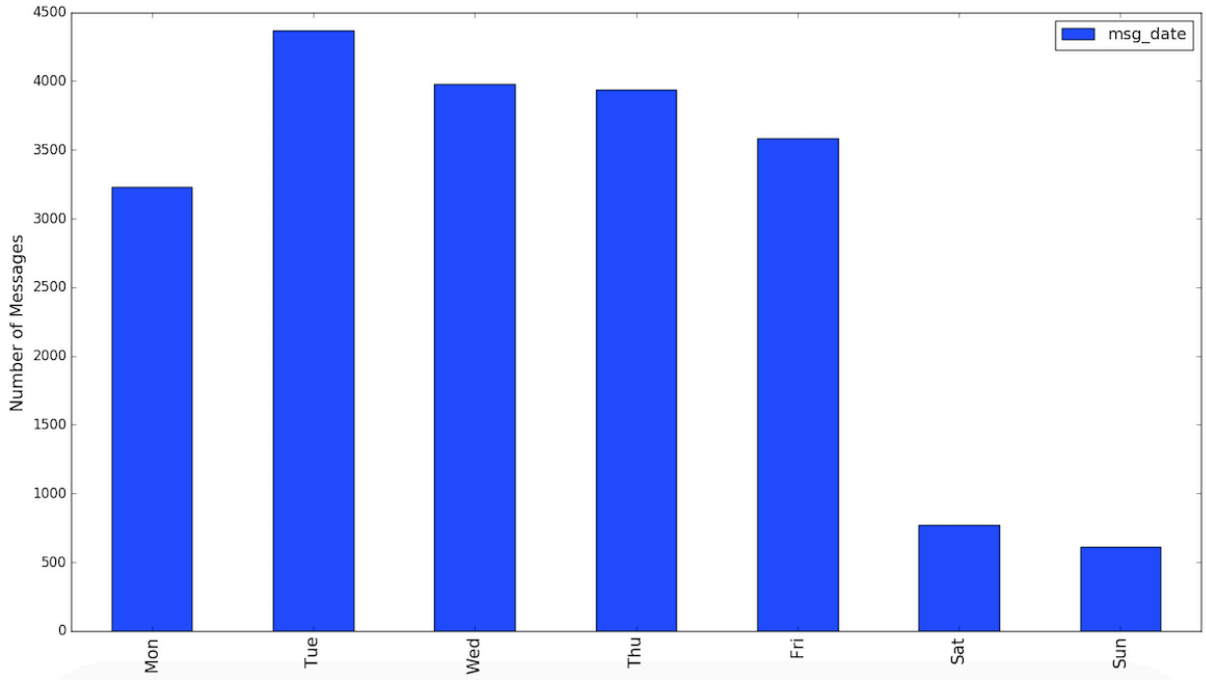

Figure 5: Messages by day of week

With 10,512 realized events representing all of the replies sent to the target mailing list over the approximately two year time period and 20,479 total messages that may or may not have been replied to, it would be computationally prohibitive to calculate all the variables for every possible reply. Even with using only seven days of original messages as the unrealized comparison events, this would result in a dataset of over 1.8 million events, each with a full set of variables, several of which are computationally intensive to calculate. Therefore, a case-control approach (Sorenson and Stuart 2001; Sorenson et al. 2006; Cassi and Plunket 2015) was used with a sampling strategy where each actual, realized event (a message replied to by an ego) is compared to a sample of unrealized events made up of randomly selected messages that an ego could have, but did not, select for a reply. These unrealized events are sampled at random from a pool of recent messages that could have been replied to as alternatives to the realized event. To control for the temporal variation described previously, the sample of recent messages as controls are drawn at random from messages posted in the previous seven days.

A sample size of five unrealized controls was selected after reviewing several studies using similar models. Cassi and Plunket (2015) used proximity theory to study collaboration between co-inventors on patents with undirected ties by sampling five controls per co-inventor for a total of ten controls per event. In another proximity study, Sorenson et al. (2006) investigated knowledge flow via patent citations using a random sample of four patents that were not cited as controls. Other studies have used only one event as a control. For example, Sorenson and Stuart (2001) studied venture capital networks by sampling one unrealized

venture capital investment as a control, and Agrawal et al. (2006) used a single patent as a control for each realized patent that could have cited it, but did not.

With a matched case-control approach, the proportion of realized events to controls is higher than the proportion of possible events in the population, which can result in underestimated coefficients, so smaller sample sizes may have an advantage over larger samples (Sorenson et al. 2006). To adjust for potential correlation within each group of realized events plus controls, the cluster robust option is used in the model to obtain robust standard errors (Cassi and Plunket 2015) while keeping in mind that robust standard errors might not fully correct for heteroskedasticity in error terms for non-linear models. In some instances, rare event models might be appropriate to address this issue when the proportion of realized events to possible unrealized events is quite small (less than 0.005%) (Cassi and Plunket 2015); however, with a median of 25 posts per day over seven days, the five unrealized control events will be sampled from a pool of approximately 175 messages, so the events are not particularly rare; therefore, a rare event model was not used.

Independent Variables

Each independent variable is calculated for each randomly selected, unrealized event in addition to the realized event to allow the model to compare the events that could have occurred with the event that actually occurred to determine which variables influence the likelihood of a collaboration event. Because the ego is the same for the realized event and the randomly selected unrealized events, the ego remains constant and ego effects cannot be directly measured using this approach, so the independent variables are focused on alter effects and dyadic covariates (Cassi and Plunket 2015).

Some of these independent variables (e.g. network measures, social proximity, and cognitive proximity) are calculated using past history over a moving window of time. Because Linux kernel development happens in cycles with regular releases, the median kernel release cycle timing of 63 days was selected as the moving window length to capture as much of the cycle variation as possible. This also allows the moving window to be a multiple of seven to ensure that each moving window includes full weeks of data to take into account the weekday / weekend variance described earlier.

Control Variables. Control variables are used to take into account three factors specific to this empirical setting that may influence collaboration. First, maintainer variables were used to take leadership positions into account for people who were maintainers at the time of the event. These maintainers are the people responsible for reviewing contributions and

determining which code is eventually accepted (committed) into the Linux kernel (Lee and Cole 2003; Schneider et al. 2016). For maintainers, the process of reviewing contributions is often collaborative. Maintainers reply to mailing list messages with feedback or questions and others reply to provide answers or additional information, both of which would generate additional collaboration events. *Alter maintainer* is a dummy variable set to 1 if the alter for the event is a maintainer and 0 if they are not a maintainer. While ego effects cannot be included directly in the conditional logit model, the ego effect for maintainer can be inferred by comparing the *Alter maintainer* effect with a second variable that measures whether either the ego or the alter is a maintainer, since any change in the likelihood of collaboration when compared to *Alter maintainer* would indicate an effect that could be attributed to ego maintainers. *Either maintainer* is a dummy variable is set to 1 if the ego and/or the alter are in a maintainer role and set to 0 if neither is a maintainer.

Second, commit variables are used to determine the influence on collaboration for people who have submitted code that has been included into the Linux kernel during the moving window. Code commits demonstrate that a person is involved in the project beyond mailing list conversations and the number of commits acts as a measure of activity or technical contribution to a project (von Krogh et al. 2003; Dahlander and O'Mahony 2010). Within the Linux kernel, committing code is also a collaborative process. Since committers are more deeply involved in the project, they would be expected to be more active on the mailing list and thus generate more collaboration events. When a committer contributes new code, they post it to the mailing list in the form of a patch where they would then be expected to respond to feedback or answer questions, which would generate additional collaboration events. It is also possible that some committers would review and provide feedback on code submitted by others, especially in areas related to previous contributions or changes to code they have authored or previously modified, which would again generate additional collaboration events. *Alter committer* is a dummy variable set to 1 if the alter for the event has committed code and 0 if they have not. Like with the maintainer variables, a second variable measuring whether either the ego and / or the alter have committed code can help in understanding the ego effect. *Either committer* is a dummy variable set to 1 if the ego and/or the alter have committed code and 0 if neither has committed code.

Third, whether the ego was explicitly included in the "to" or "cc" field of the email being replied to in addition to the email being sent to the mailing list has been included as a variable, since this is a recommended practice within this setting (Kernel development community 2017a). This is often done to get the attention of the maintainer when submitting

Linux kernel patches (Kernel development community 2017b). It is also used when replying to preserve the email address of the person being replied to, along with any other individual email addresses in the "cc" field (Kernel development community 2017a), which can be included to get the attention of people who are likely to be interested in a particular patch or discussion. Because the Linux kernel mailing lists can generate hundreds of email messages per day, many Linux kernel developers use sophisticated email filters that send the messages to folders unless they are explicitly mentioned in the "to" or "cc" field. Including someone in the "to" or "cc" field is intended to increase the likelihood of a reply, which would generate a collaboration event. *Ego to cc* is set to 1 if the ego was explicitly included in the "to" or "cc" field of the original email that was replied to and otherwise is set to 0.

Proximity variables. *Geographical proximity* is operationalized using time zone similarity (O'Leary and Cummings 2007), because some fluid organizations, including the Linux kernel, where developers collaborate in an online community without physical collocation, there is no spatial dimension to measure (Boschma 2005; Torre 2008; Gulati et al. 2012). By using the time zone tags included in the mailing list archives, the difference between the time zone offsets in seconds was calculated for the original message sent by the alter and the ego's reply, which provided a measure of geographical distance. Geographical distance is normalized to a value between 0 and 1, and 1 minus the normalized geographical distance is used as the measure for the *Geographical proximity* variable.

Organizational proximity measures whether both the ego and the alter work for the same employer. Employer affiliation is based on the actor's affiliation at the time of the collaboration event (mailing list reply) and was determined using a number of different factors. First, the dataset containing affiliations for code contributors, which is used in the yearly Linux Kernel Development Report (Corbet and Kroah-Hartman 2017), was obtained from The Linux Foundation; however, this was incomplete, most notably for people participating on the mailing list who have not contributed source code. Second, employer email domains were used to determine affiliation, but in some cases where people changed jobs, there were gaps or overlaps that did not provide reliable dates for the job change. Third, an attempt was made to find this information using other online resources, including blog posts, contributions to other open source projects, or mailing list posts mentioning a job change. This was also the method used for determining affiliation for people using personal email addresses who were not in The Linux Foundation dataset. Finally, where no better information was available, the midpoint between dates of posts from employer email addresses was taken as the date of the job change. Due to limitations in the Sortinghat

software and to allow for the calculation of a single *Organizational proximity* and *Institutional proximity variable* per dyad, it is assumed that a person only has one employer affiliation at a time. *Organizational proximity* is calculated as a dummy with a value of 1 indicating that both work for the same employer or 0 for different employers in a method similar to several recent proximity studies (Cassi and Plunket 2015; Crescenzi et al. 2016).

Institutional proximity uses the employer affiliation data from the *Organizational proximity* variable with a mapping that matches employers to type of institution. The dataset from The Linux Foundation also contained some of the data used to map employers and individuals to institutions, especially in the case of academia and hobbyist affiliations. *Institutional proximity* is operationalized as a dummy variable using similarity across four types of institutions: corporation, non-profit, academic, and hobbyist (unaffiliated). If both actors are employed by the same type of institution, *Institutional proximity* is set to 1, otherwise, it is 0. If an actor's affiliation cannot be determined, it is assumed that the person is unaffiliated and included in the hobbyist category.

Social proximity is typically operationalized using the shortest path or number of people required to reach the alter from the ego based on network ties. However, because the dependent variable was operationalized using replies (network ties) from an ego to an alter, it was appropriate to use a different measure for *Social proximity* for this study. Participation on mailing lists occurs within threads. In its simplest form, a thread can be made up of a single post to a mailing list and a reply to that post, but in many cases, these threads can branch out and include many replies with some of them being replies of other replies stemming from that original source message. Forming connections and relationships with each other occurs as individuals participate in the same threads over time. In this study, *Social proximity* is a measure of the number of times an ego and alter dyad participated in same thread within the mailing list.

Cognitive proximity is operationalized by considering the similarity between sections of the Linux kernel code where two individuals have contributed. Because this study focuses on a specific subsystem, the PCI subsystem, the *Cognitive proximity* measurement is based on sections of the source code as defined by the sections of the MAINTAINERS file that use the PCI subsystem mailing list to provide more granular data. Each section of the MAINTAINERS file contains a specification for the files and / or directories within the source code that are covered by that section. *Cognitive proximity* is operationalized by determining similarity in contributions to these sections of the source code using a cosine similarity formula, which has been previously used in the proximity literature to operationalize

Cognitive proximity, but with journal contributions, instead of source code contributions as the source (Hardeman et al. 2015). The total number of sections of the code that are shared by the ego (A) and the alter (B) is divided by the product of the square root of sums squared for the ego and the alter.

$$\text{Cosine Similarity} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

This results in a number between 0 and 1 with 0 indicating that the ego and alter have not contributed to any of the same sections of the source code, 1 indicating that they have contributed to exactly the same sections of the source code, and 0.5 if each person has contributed to more than one section of the source code with half of them shared and the other half not shared. The variable is also set to 0 if either person has not committed code within the moving window.

Empirical research within the proximity literature has shown that cognitive proximity and social proximity may take the form of an inverted u-shaped curve indicating an increase in the variable of interest only up to a certain point where further increases in cognitive or social proximity start to have diminishing returns (Nooteboom 1999; Sorenson et al. 2006; Nooteboom et al. 2007; Gilsing et al. 2008). This has been tested by including a quadratic effect using the squared version of the variable in addition to the original variable in the regression model. When the coefficient for the original variable is positive and the squared effect for that variable is negative, this usually indicates that the variable has an inverted u-shaped curve (Nooteboom et al. 2007; Gilsing et al. 2008).

Network variables. Collaboration is a network phenomenon, and in fluid organizations with their flexible and evolving hierarchies, the role of networks is especially important for understanding collaboration. Because networks evolve over time, all of the network variables are calculated over the 63 day moving window, which corresponds to the median length of a Linux kernel release cycle. Both dyadic and triadic effects are investigated to understand not just how past interactions between two people influence future collaboration, but also to understand the influence that third parties have on collaboration between two people.

Three dyadic effect variables were used to investigate how past behavior between the ego and the alter influences the likelihood of a collaboration event. *Repeated events* is operationalized as the number of times the ego replied to messages from the alter within the moving window and is a measure of persistence. *Participation shift* is operationalized as a

dummy variable with a value of 1 if the last person the alter replied to on the mailing list was the ego within the moving window. The *Recency effect* is measured as $\frac{1}{n}$ with n defined as the number of people the alter emailed on the mailing list before the ego within the moving window (Butts 2008). Both *Participation shift* and *Recency effect* are measures of reciprocity with *Participation shift* being a specific case of the *Recency effect* where the *Recency Effect* is equal to 1. These dyadic effects are illustrated in Figure 6.

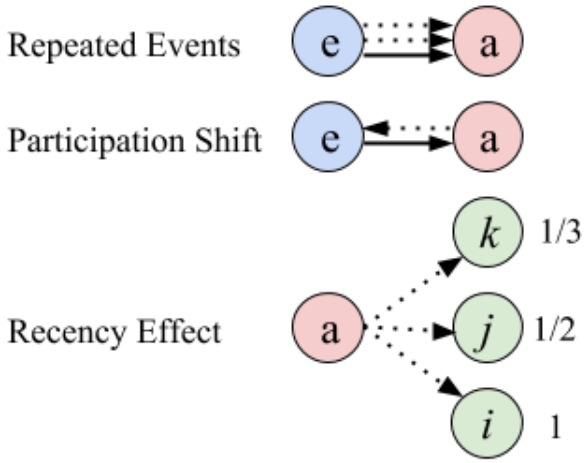

Figure 6: Dyadic network variables illustrated

To better understand the relationship of third parties relative to the likelihood of a collaboration event between an ego and alter, four triadic effects were considered and measured within the 63 day moving window. *Transitive closure* is measured by counting the number of third parties that an ego has replied to where those third parties have also replied to the alter. *Cyclic closure* measures the effect in the other direction by looking at the number of third parties an alter has replied to where that third party has also replied to the ego. *Shared partnership inbound* also referred to as popularity closure is a structural homophily effect rooted in shared popularity where the ego and alter are both popular connections from the same set of people (Robins et al. 2009). *Shared partnership inbound* or popularity closure is operationalized as the count of third parties who have recently replied to both the ego and the alter. *Shared partnership outbound* is also a structural homophily effect representing a similarity in choice of connections or shared network activity and is also called activity closure (Robins et al. 2009). *Shared partnership outbound* or activity closure is measured by counting the number of times the ego and the alter have replied to messages by the same third party. These triadic closure effects are illustrated in Figure 7.

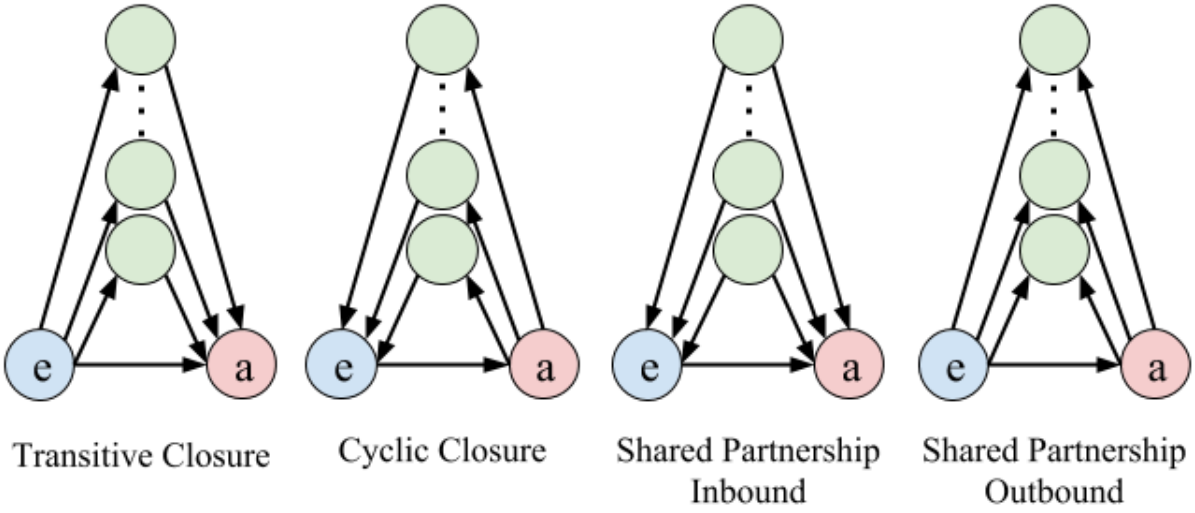

Figure 7: Triadic closure variables illustrated

Because the calculation of network variables over time is complex, Figure 8 provides a brief, simplified example of how the network variables are calculated from the events that occur over a 63 day moving window to aid in understanding the application of the relational events model. Starting at the bottom with time t = 0 is the target event (realized collaboration event) where the current ego (e) emailed the current alter (a). In this example, all of the network variables are calculated relative to this target event from all of the collaboration events (mailing list replies) over the past 63 days corresponding to the median Linux kernel release cycle for the moving window. The same network variables are also calculated over the moving window for each of the five sampled unrealized events corresponding to the target event to be used as a comparison in the relational event model. Events involving other third parties (*i, j, k*) are used in the dyadic recency calculation and to calculate the triadic network effects as shown in Figure 8. For quick reference, Appendix C contains a summary of the operationalization of all variables in Table 10 and variable descriptive statistics and correlations in Table 11.

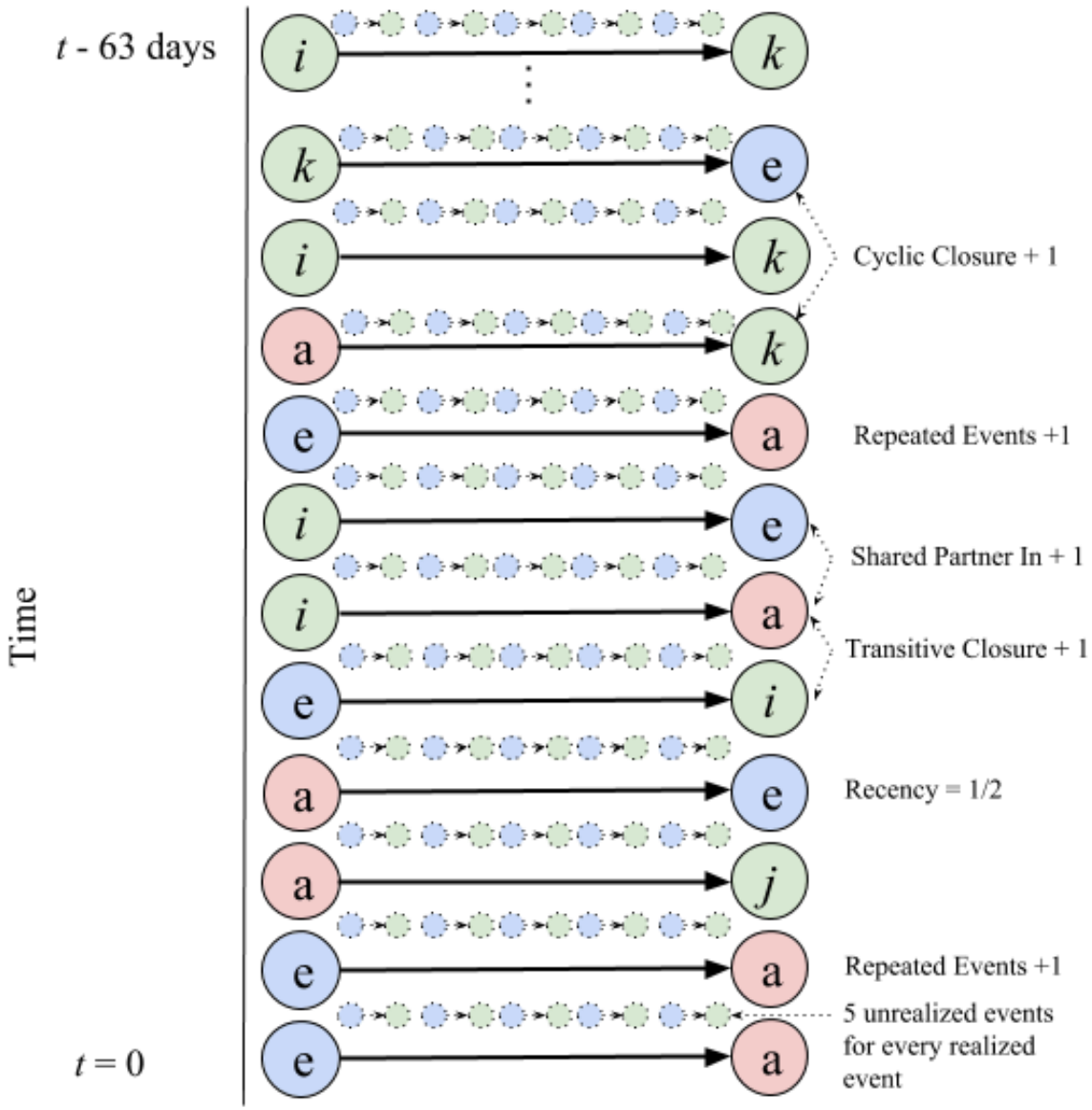

Figure 8: Network variable calculations example

5.3. Results

Table 6: Relational event models

Variables	Model 1		Model 2		Model 3	
Alter Maintainer	-0.059	(0.014) ***	-0.097	(0.018) ***	-0.059	(0.015) ***
Either Maintainer	0.335	(0.074) ***	0.519	(0.107) ***	0.218	(0.075) **
Alter Committer	-0.196	(0.043) ***	-0.520	(0.109) ***	-0.217	(0.053) ***
Either Committer	0.296	(0.111) **	0.752	(0.176) ***	0.637	(0.145) ***
Ego To CC	3.391	(0.798) ***	2.935	(0.738) ***	2.618	(0.688) ***
Geographic Proximity			0.433	(0.107) ***	0.137	(0.074) .
Organizational Proximity			0.900	(0.180) ***	0.627	(0.133) ***
Institutional Proximity			-0.317	(0.066) ***	-0.131	(0.049) **
Social Proximity			0.658	(0.148) ***	1.052	(0.243) ***
Social Proximity Squared			-0.043	(0.011) ***	-0.050	(0.013) ***
Cognitive Proximity			1.534	(0.363) ***	1.856	(0.403) ***
Cognitive Proximity Squared			-4.517	(1.003) ***	-4.287	(0.888) ***
Repeated Effect					0.002	(0.001)
Participation Shift					-0.163	(0.151)
Recency Effect					0.882	(0.227) ***
Transitive Closure					0.036	(0.010) ***
Cyclic Closure					0.086	(0.019) ***
Shared Partnership Inbound					-0.077	(0.018) ***
Shared Partnership Outbound					-0.127	(0.029) ***
BIC	23357		20364		18033	
Observations (realized events + controls)	63072		63072		63072	
Realized Events	10512		10512		10512	
Significance p < 0.001 '***' p < 0.01 '**' p < 0.05 '*' p < 0.1 '.'						
Robust Standard Errors are shown in parentheses						

The results from three nested models with their associated coefficients, robust standard errors, and Bayesian Information Criterion (BIC) scores are contained in Table 6. Model 1 contains only the control variables for the empirical setting; Model 2 adds the five proximity variables; and Model 3 is the final model with all three sets of variables (control, proximity, and network). BIC scores have been previously used by Butts (2008) to determine model fit for relational event models with lower BIC scores indicating better model fit. For these models, the lower BIC score for Model 3 confirms a substantial improvement in fit from Models 1 and 2, which both had higher BIC scores.

5.3.1. Model 1: Controls

The *Alter maintainer* effect was negative and significant indicating that an ego is 6% (exp(-0.059) - 1) less likely to reply to a message if it was sent by a maintainer - other things being equal, so collaboration events are less likely to occur if the alter is a maintainer. While ego effects cannot be directly measured for the reason previously described, the *Either maintainer* variable allows for an indirect measurement of the ego maintainer effect in this situation. The negative and significant *Alter maintainer* effect combined with a positive and significant *Either maintainer* effect indicates that a collaboration event is more likely if the ego is a maintainer. In sum, maintainers are more likely to reply to mailing list messages, but are less likely to be replied to.

The results are similar for collaboration events between people who have contributed source code that has been accepted into the Linux kernel during the 63 day moving window. The *Alter committer* effect was negative and significant indicating that a collaboration event or a reply to a message is 18% (exp(-0.196) - 1) less likely if the person being replied to has had source code committed. Like with the *Either maintainer* variable described above, combining the negative and significant *Alter committer* effect with the positive and significant effect of the *Either committer* variable indicates that an ego who has committed code is more likely to reply to a message and create a collaboration event. In sum, more experienced people (committers, maintainers) are more likely to create collaboration events.

The strongest effect in the model is the *Ego to cc* variable, which was positive and significant. The results indicated that an ego is almost 30 times (exp(3.391) - 1) more likely to reply to a mailing list message if the ego's email address is included in either the "to" or "cc" fields of the email. It is a documented practice when submitting Linux kernel patches to copy the maintainers' email addresses (Kernel development community 2017b), and when replying to a mailing list post, the email address of the person being replied to, along with any individual email addresses in the "cc" field should be preserved (Kernel development community 2017a), so individual email addresses are frequently in the "to" or "cc" fields in addition to the email address of the relevant mailing list.

5.3.2. Model 2: Controls and Proximity

Institutional proximity was negative and significant indicating that a collaboration event is 27% (exp(-0.317) - 1) less likely if the ego and the alter are from the same type of institution (company, non-profit, academic, or unaffiliated). In contrast, *Organizational proximity* was positive and significant, which indicates that a collaboration event almost 1.5

times (exp(0.9) - 1) more likely if both people are employed by the same third party organization. This is unexpected given that employees would also have numerous internal methods that they could have used for collaboration, but it could indicate that they are providing support for each other.

The effect of *Cognitive proximity* was positive and significant, indicating that collaboration events are more likely between two people who have contributed to the same sections of the Linux kernel source code during the 63 day moving window. In combination with the negative and significant squared effect, the results indicated that *Cognitive proximity* has an inverted u-shaped curve leading to the conclusion that collaboration only increases to a point and then the marginal effect of an increase in *Cognitive proximity* has diminishing returns for collaboration events between people who contribute to many of the same sections of code.

The effect for *Social proximity* was positive and significant, which indicates that a collaboration event is more likely between two people who have participated in the same threads on the mailing list during the moving window. Like with *Cognitive proximity*, the squared effect for *Social proximity* was negative and significant, again indicating that the likelihood of collaboration increases initially but has diminishing returns for people who have participated in many of the same threads.

In Model 2, *Geographical proximity* was positive and significant indicating that a collaboration event is 54% (exp(.433) - 1) more likely if both people are in similar time zones. However as shown in the next section, this variable loses significance in Model 3.

The correlation between *Cognitive proximity* and *Social proximity* was quite high at .66, and cannot simply be explained by the variables measuring proximity between contributors along the same axis, since these two variable calculations come from different sources. *Cognitive proximity* is based on code contributions, which originates in the source code repository database, while *Social proximity* is based on threads from the mailing list. *Organizational proximity* is also highly correlated with both *Social proximity* (.55) and *Cognitive proximity* (.61), and with *Organizational proximity* coming from employer affiliation data, all three of these variables are from different sources. More details about the variable correlations can be found in Table 11 of Appendix C.

5.3.3. Model 3: Final model

Model 3 is the full model with control, proximity, and network variables together in this final model. As mentioned earlier, this model demonstrated an improved fit over Models 1 and 2 as indicated by the lower BIC score.

Triadic effects. The effects for *Transitive closure* and *Cyclic closure* were both positive and significant. For *Transitive closure*, the results indicated that each collaboration event that leads to transitive closure increases the likelihood of collaboration by 4% (exp(0.036) - 1). The *Cyclic closure* effect was positive and significant indicating that each collaboration event that leads to cyclic closure increases the likelihood of collaboration by 9% (exp(0.086) - 1). Both of the *Shared partnership inbound* and *Shared partnership outbound* effects were negative and significant indicating that each collaboration event that leads to inbound / outbound shared partners decreases the likelihood of collaboration by 8% (exp(-0.077) - 1) for inbound and 12% (exp(-0.127) - 1) for outbound. In sum, all four of the triadic effects have a significant impact on the likelihood of collaboration.

Dyadic effects. The *Recency* effect was positive and significant indicating that a collaboration event is more likely if the alter has recently emailed the ego; therefore, reciprocity has a role in collaboration within this network. The *Participation shift* and *Repeated events* effects were both insignificant, so the model provides no evidence that these effects influence collaboration.

The biggest change between Models 2 and 3 is for *Geographical proximity.* In Model 3, the *Geographical proximity* effect was not significant, thus the model provides no evidence that being in similar time zones influences the likelihood of collaboration. This is a change from Model 2, where geographical proximity was positive and significant. Since it became insignificant when adding the network variables, it appears that network variables, rather than time zones, were influencing the likelihood of collaboration. After an experiment conducted by removing the network variables from the model and adding them back in one by one, it was determined that *Geographical proximity* becomes insignificant after adding *Transitive closure* and / or *Shared partnership inbound*.

For the dyadic effects, *Participation shift* and *Recency* effects were highly correlated (.90), which is expected given that they are both measures of reciprocity with *Participation shift* being a specific case of the *Recency* effect such that the events captured by *Participation shift* are a subset of those captured by the *Recency* variable. This implies that recent reciprocity between the alter and the ego, but not the *most* recent event, influences the likelihood of collaboration.

It is worth noting that these four closure variables are highly correlated with each other with correlations ranging from .92 to .96 (see Table 11), but this correlation is to be expected between closure variables and some models can distinguish between them (Robins et al. 2009). An experiment with the model indicated that removal of any of these variables resulted in higher (less optimal) BIC scores. Since all four were significant, and the model was not failing due to multicollinearity, the decision was made to leave all four closure variables in the final model to allow for a broader understanding of the closure mechanisms and how they influence the likelihood of collaboration.

All of the triadic effects are also highly correlated with both *Social proximity* and *Cognitive proximity* with correlations ranging from .52 to .79. It is expected that the correlation would be high for *Social proximity* because thread participation would include triadic effects, but with the addition of the *Cognitive proximity* correlations, this lends further support to the idea that collaboration, including collaboration leading to triadic closure, on mailing list threads could be related to collaboration on the same areas of the source code as discussed in the proximity results from Model 2.

5.4. Discussion

These results have meaningful implications for how researchers and practitioners understand collaboration within fluid organizations. Using a relational event model on mailing list replies as collaboration events, the findings demonstrate that within fluid organizations, both proximity and networks influence the likelihood of collaboration.

This research demonstrates how alternative proximity measurements can be used to investigate collaboration within a fluid organization. Because collaboration in this setting occurs in online communities, physical location is somewhat irrelevant (Boschma 2005; Torre 2008), but time zones can provide some insight into geographical proximity as it relates to the times people are available to collaborate. While O'Leary and Cummings (2007) used time zones as a portion of their geographical proximity measurement, it is more common for people to use physical distance as a measure. Hardeman et al. (2015) used cosine similarity to operationalize cognitive proximity looking at contributions to similar journals; however, this thesis may be the first time cosine similarity on source code contributions has been used as a measure of cognitive proximity. Likewise, this may be the first time that participation in mailing list *threads* (as opposed to individual emails) has been used to operationalize social proximity, instead of more typical measures like using network distance (i.e. shortest path, geodesic distance) to determine social proximity. As described in more detail in the earlier

section (5.2.3) on model estimation and variables, social, cognitive, and geographical proximity were operationalized in atypical ways that were still consistent with the conceptual definitions in previous literature, but that mapped more directly to this empirical setting.

The likelihood of collaboration decreases when people work for the same type of institution (corporation, non-profit, academia, or unaffiliated hobbyist). This result is consistent with Cassi and Plunket (2015) who found that for tie formation in patent collaboration networks, institutional proximity had a negative effect that could be a result of the risk associated with working with competitors. With participants employed by many competing firms, the negative influence on the likelihood of collaboration in the Linux kernel could also stem from competitive pressures; however, there are also a number of other possible explanations, several of which could be contributing to this result. This could be partly explained by the Phase 1 interview result indicating that people employed by third party organizations were more willing to provide help to unaffiliated hobbyists than to developers who are employed to work on the Linux kernel. This could generate more messages (collaboration events) in the process of helping someone, which could contribute to the negative effect for institutional proximity. Additionally, this could be demonstrating a strong collaboration between people working for corporations and non-profit organizations with the strong participation from people employed by non-profit organizations like Linaro and The Linux Foundation, both of which are major contributors to the Linux kernel. Linaro, for example, is the third largest employer for code contributions to the Linux kernel behind only two large corporations, Intel and Red Hat (Corbet and Kroah-Hartman 2017).

People working for the same employer are more likely to collaborate demonstrating that organizational proximity influences collaboration in this fluid organization. This result is consistent with studies on patent collaboration, which show that inventors overwhelmingly tend to file patents with other employees as co-collaborators (e.g. Cassi and Plunket 2015; Crescenzi et al. 2016). In some cases for the Linux kernel, if the technology is closely tied to a third party organization's technology, other employees might be the ones with the most expertise to provide feedback and collaborate around a particular technology. Even when the technology is not specific to a third party organization's technology, this positive result could indicate that employees are participating on the mailing list to provide support for each other or to help answer questions and respond to feedback. Given the number of other, internal communication channels available to people employed at the same third party organization, it is somewhat unexpected that this collaboration between employees occurs on public mailing lists. While the ideal and expectation within open source communities is to collaborate on the

public mailing lists, the Phase 1 interviews demonstrated that this is not always the case, since several people mentioned that they collaborate using internal channels with other employees in addition to using the Linux kernel mailing lists. While there may still be some collaboration occurring internally between employees, they appear to be adopting the practice of collaborating on the mailing lists, which could possibly be because they or their employer want this work to be visible and accessible to the rest of the participants in this fluid organization.

Collaboration events are more likely between people who have recently contributed to the same areas of the Linux kernel source code, thus showing that cognitive proximity influences collaboration in this fluid organization. Further, the results indicate that cognitive proximity has an inverted u-shaped curve leading to the conclusion that cognitive proximity has diminishing returns for collaboration events between people who have high cognitive proximity. This inverted u-shaped curve is consistent with the literature on cognitive proximity (Nooteboom 1999; Nooteboom et al. 2007; Gilsing et al. 2008). This could indicate that people working in the same areas of the Linux kernel source code tend to provide feedback and otherwise engage on the mailing lists to create collaboration events with others who are submitting patches or discussing the same areas of the source code. The diminishing returns may be showing that people with very similar knowledge and expertise tend to have similar approaches to the Linux kernel source code, thus requiring fewer collaboration events to effectively collaborate with each other.

Social proximity also increases the likelihood of a collaboration event in this fluid organization. Like with cognitive proximity, social proximity has an inverted u-shaped curve indicating diminishing returns on the likelihood of collaboration for people who have participated in many of the same threads. This inverted u-shaped curve for social proximity is consistent with findings from Sorenson et al. (2006). For people who have participated in many of the same threads, thus having a high level of social proximity, the diminishing returns could indicate that this shared history makes it easier for two people to collaborate. In this case, additional participation in shared threads (higher social proximity) does not further increase the likelihood of collaboration once a certain threshold is reached.

The findings and variable correlations indicated that cognitive proximity and social proximity might have some relationship to each other. A likely explanation for this relationship is that people have an increased likelihood of collaboration when they are focused on similar things, thus they tend to participate in the same mailing list threads (social proximity) and also contribute to the same areas of the source code repository (cognitive

proximity). These two dimensions of proximity working together in an additive manner to increase the likelihood of collaboration could indicate that social and cognitive proximity are complementary. Organizational proximity also appears to have some relationship with both social proximity and cognitive proximity. This is to be expected, since people working for the same employer are likely to be working in similar areas of the source code (cognitive proximity), which could also cause them to participate in many of the same threads (social proximity) if those threads contain topics that are important to the work at their employer. Interrelationships between proximity dimensions have been explored in the literature (see Literature Review Section 2.2.2) and will be studied in greater depth in Phase 3.

Network effects also influence the likelihood of collaboration in this fluid organization, which is expected given that collaboration is a network phenomenon. Only one of the dyadic effects, recency, influenced the likelihood of collaboration. With mailing list collaboration, it is common for mailing list threads to include several back and forth messages between two people in a short timespan as they collaborate. This is often in the form of questions and answers or feedback and response, but it can also include a variety of other discussions.

The results of the triadic network effects demonstrate that relationships with other third parties have significant effects on the likelihood of collaboration between two people in a variety of different ways. Shared partnership inbound and shared partnership outbound are both structural homophily effects where the ego and the alter share some form of structural equivalence (Robins et al. 2009). In this study, both were associated with negative influences on the likelihood of collaboration, which indicates that participants sharing structural similarities are less likely to collaborate with each other. The nature of how collaboration occurs within the Linux kernel might explain these negative effects. For example, if two people have both contributed patches to the Linux kernel in similar, but separate areas, they might get several emails with feedback from the same set of people (shared partnership inbound) and reply to that feedback (shared partnership outbound), but these two contributors would be less likely to reply to each other (less likely to generate a collaboration event).

Taken together, the positive effects of both transitive closure and cyclic closure indicate a tendency at the triadic level for structural holes to close indicating that collaboration occurs between closely interconnected people, instead of people acting as brokers who pass information between unconnected parties (Robins et al. 2009). Note that the interpretation here from Robins et al. (2009) refers to structural holes at the triadic level, which builds on, but is slightly different, from Burt's (1992) definition of structural holes at the global network

level. This is to be expected in the context of mailing list communication where anyone can reply to any other person, thus generating a collaboration event, as opposed to other types of networks where a person would need to have a previous relationship with a person to know who to contact. From a communication patterns standpoint, this result could simply indicate that messages are exchanged in a variety of ways, which represent different communication patterns leading to different types of closure. Both transitive and cyclic closure may arise out of different situations and types of communication patterns. The positive effects of both transitive and cyclic closure could also have a more structural interpretation. The positive transitive closure indicates that there could be some hierarchical influences on the creation of collaboration events, while a positive cyclic closure would typically indicate that collaboration events come from a flatter, non-hierarchical structure for collaboration. While both of these results together might be unexpected in a more traditional organization, fluid organizations can have a hierarchical structure, but it tends to emerge organically from the network and adjust in a fluid manner as needed with members collaborating across hierarchical and group boundaries. In the Linux kernel, maintainers can be considered a loose hierarchy, but these results demonstrate that when creating collaboration events, participants work both within this hierarchy and outside of it as needed in a fluid manner as they collaborate with other members. Thus, showing that the creation of collaboration events is influenced by third parties in triadic structures, which may or may not align with the fluid hierarchy of the Linux kernel.

CHAPTER 6. PHASE 3: ANALYZING THE IMPACT OF PROXIMITY DIMENSION INTERRELATIONSHIPS ON COLLABORATION

6.1. Introduction

Boschma's (2005) five dimensions of proximity were designed to reduce overlap and isolate the effect of each dimension relative to the others, but they do not exist in isolation. It is widely documented in the body of literature on proximity and covered in more depth in the Literature Review Section 2.2.2 that these dimensions of proximity are often interrelated, operating as complements or substitutes (Boschma 2005; Balland et al. 2015; Crescenzi et al. 2016; Heringa et al. 2016), which highlights the importance of considering the relationships *between* proximity dimensions in addition to looking at each one separately. For example, organizational and cognitive proximity may be complements (Boschma 2005; Cassi and Plunket 2014), cognitive and social proximity may also be complementary (Cassi and Plunket 2014), and other dimensions of proximity may act as substitutes for geographical proximity (Boschma 2005).

When two proximity variables work together with an additive effect on the variable of interest, then the relationship is complementary. This complementary relationship can be demonstrated when higher levels of both dimensions of proximity result in increases in the probability of a successful outcome (Cassi and Plunket 2014). Hansen (2015) refers to this as an *overlap* effect where one dimension of proximity helps facilitate the other as the two dimensions work together to influence the outcome.

When one dimension of proximity compensates for another the relationship is substitution. In a substitution relationship, proximity is only needed in one of the two dimensions for a successful outcome, and proximity in the other dimension adds little or no increase in the probability of success (Boschma and Frenken 2010). The proximity literature typically indicates that dimensions of proximity acting as substitutes will have a negative interaction term, which is taken to mean that one dimension matters less in the presence of the other dimension (Cassi and Plunket 2014).

As discussed in the Literature Review, the proximity literature explores various elements of collaboration, including collaboration output performance (Cassi and Plunket 2014), collaborative knowledge creation (Crescenzi et al. 2016), knowledge transfer between

collaborators (Sorenson et al. 2006), and collaborative innovation between firms (Hansen 2015). While these differ from the likelihood of collaboration that is the focus of this research, the literature showing the influence of interactions between proximity dimensions on various elements of collaboration is similar and was selected as the basis for the hypothesis development in this chapter. These hypotheses were selected over other combinations of proximity interactions because they had been found to be significant in multiple studies related to collaboration.

This leads to the research question for the final phase of this thesis, "What is the role of interrelationships between proximity dimensions on collaboration within a fluid organization where the majority of participants are employed to contribute?" The remainder of this chapter is organized as follows. The first section outlines five hypotheses with a theoretical justification for each one. This is followed by a methods section describing how the research was conducted. The next section contains detailed results with support or rejection for each hypothesis. The final discussion section reflects on the impact of these results.

6.2. Theory and Hypotheses

Increases in proximity result in a lowered cost of future collaboration due to a decrease in coordination and communication costs (Boschma 2005; Balland et al. 2015). The relationship between proximity and coordination costs for collaboration are explored in the remainder of this chapter through a series of five hypotheses. Along with the theoretical justification for exploring the relationships between dimensions of proximity described in the preceding paragraphs, the qualitative results from Phase 1 of this research found several potential interrelationships between proximity dimensions that can now be explored.

6.2.1. Social and Cognitive Proximities as Complements

Several studies point out that in order to effectively collaborate, communicate, and learn from each other, individuals require cognitive proximity and the shared knowledge / shared technical language that comes with it (Boschma 2005; Huber 2012; Balland et al. 2015). The literature on the relationship between cognitive and social proximities is mixed and somewhat thin with few studies looking at this interaction. Cassi and Plunket (2014) found that social and cognitive proximities function as complements, thus resulting in higher quality collaborations possibly due to exploitation of a common technological specialization between two individuals that is facilitated by coordination via the improved trust and control from their

social connection. In an intraorganizational study of knowledge sharing within a professional services firm, Criscuolo et al. (2010) claim that social and cognitive proximity are substitutes, but in looking at their results, it would seem that they are only substitutes at certain levels of those proximities.

Because the literature on this interaction is mixed and thin, more weight is being given to results from earlier phases of this study in the development of this hypothesis. In the previous chapter describing the results from Phase 2 of the research, social proximity and cognitive proximity were suggested to be complementary in part due to the high correlation between the two variables despite cognitive proximity coming from the source code database and social proximity resulting from common participation in mailing list threads. It is likely that people who participate in the same mailing list threads also contribute to the same areas of the source code repository. Code contributions, in the form of patches, are discussed in mailing list threads where people respond to contributions with feedback, suggestions, concerns, and more. The people who have already worked on the areas of the source code involved in a contribution would be familiar with the existing code and likely to be the ones to respond to threads with feedback on new contributions in the same areas because of their knowledge of that portion of the source code. These two activities, participation in threads for social proximity and contribution to source code as cognitive proximity, are aligned in the process of collaborating on the Linux kernel with further code contributions and participation in additional threads likely building on each other in an additive fashion. Between two people, participating in the same threads increases social proximity and being involved in similar technologies increases cognitive proximity, which working together creates familiarity that lowers coordination costs between individuals. Cognitive and social proximity are likely to complement each other and reduce coordination costs leading to this first hypothesis.

Hypothesis 1: Two individuals who have participated in the same mailing list threads (social proximity) and have contributed to the same areas of the source code (cognitive proximity) are more likely to collaborate in the future.

6.2.2. Organizational and Cognitive Proximities as Complements

The idea that organizational and cognitive proximities act as complements has both theoretical and empirical support within the literature. Nooteboom (2000) and Boschma (2005) both suggested that organizational and cognitive proximity were complements with a similar reasoning based on the idea that people with cognitive proximity may be grouped together from an organizational perspective. The idea that organizational and cognitive

proximity are complementary for collaboration is also supported by Cassi and Plunket (2014) who found improved collaboration results from individuals employed by the same third party organization (organizational proximity) with common technology specializations (cognitive proximity). They described the relationship between organizational and cognitive proximity as multiplicative and complementary for individuals collaborating on patents (Cassi and Plunket 2014).

As an example, results from the Phase 1 interviews indicated that cognitive and organizational proximities might be complements. Participants mentioned that their work in specific subsystems (cognitive proximity) was directly related to the work of their employer (organizational proximity), so the skills and knowledge required to contribute to a particular subsystem often complements the type of work performed at their employer. In an extreme case, some subsystems are based on a single third party organization's technology where many of the participants are employed by that third party organization. For example, several subsystems are based on IBM's S/390 processor technologies and are currently maintained by over a dozen IBM employees (Linux Kernel Organization 2017). This demonstrates that people employed by the same third party organization may also be contributing to the same subsystems.

Working on similar technologies increases cognitive proximity through exposure to the same areas of the Linux kernel source code, thus creating familiarity that lowers coordination costs between individuals. Coordination costs are also lowered through organizational proximity by the fact that they both know their employer's needs and plans about the technologies being included in the kernel. Cognitive and organizational proximity are expected to work together in a complementary fashion to reduce coordination costs, which leads to this second hypothesis.

Hypothesis 2: Two individuals working for the same employer (organizational proximity) and having contributed to the same areas of the source code (cognitive proximity) become more likely to collaborate in the future.

6.2.3. Social and Geographical Proximity as Substitutes

The relationship between social and geographical proximities is widely discussed in the literature. Boschma (2005) concluded that other dimensions of proximity, including social proximity, act as substitutes for geographical proximity to improve coordination and points out that geographical proximity on its own is not a sufficient mechanism for coordination. Cassi and Plunket (2015) found that social proximity substitutes for geographical proximity in

collaboration and that geographical proximity matters less when individuals are very close within the network as defined by both having collaborated with the same partner in the past. Agrawal et al. (2006) looked at whether social proximity was retained after individuals become separated geographically to conclude that social proximity acts as a substitute for geographical proximity.

Participants in the Phase 1 interviews talked about how attendance at conferences helps them build social relationships that facilitate collaboration on the mailing lists across great physical distances, which led to the Phase 1 result that social proximity may be a substitute for geographical proximity. This is a different way of conceptualizing social proximity; however, it is interesting to consider whether social proximity, operationalized as participation in the same mailing list threads, could also provide relationships that endure over greater geographical distances. The rationale is that social proximity, and the relationships that result from the familiarity with a person over time, can lower the cost of coordination between two people even when those individuals are separated by great distances. This brings us to the next hypothesis indicating that social and geographical proximity are substitutes.

Hypothesis 3: Two individuals who have participated in the same mailing list threads (social proximity) will have an increased likelihood of collaboration even when they are not in similar time zones (geographical proximity).

6.2.4. Cognitive and Geographical Proximities as Substitutes

As mentioned previously, there is wide support in the literature indicating that individuals require the shared technical language and knowledge associated with cognitive proximity to facilitate collaboration, communication, and learning (Boschma 2005; Huber 2012; Balland et al. 2015). Hansen (2015) found that for collaborative innovation projects, cognitive proximity is a substitute for geographical proximity, suggesting that having expertise in common allowed for collaboration over large geographic distances. Crescenzi et al. (2016) suggested that cognitive and geographical proximity are substitutes in some situations, but not others, for collaborative knowledge creation.

The results from the Phase 1 interviews, indicated that cognitive proximity comes from knowledge of a particular area within the kernel where people working on a particular subsystem are likely to have similar knowledge. The interviews also indicated that geographical proximity, whether measured by distance or time zone, was not relevant because collaboration occurs asynchronously on mailing lists. The idea is that cognitive proximity resulting from working within the same areas of the code can reduce the cost of coordination

between individuals even when they are working across many time zones. This leads to the next hypothesis indicating that cognitive and geographical proximities act as substitutes.

Hypothesis 4: Two individuals who contribute to the same areas of the code (cognitive proximity) will have an increased likelihood of collaboration even when they are not in similar time zones (geographical proximity).

6.2.5. Organizational and Social Proximity as Substitutes

Cassi and Plunket (2015) found that organizational proximity can act as a substitute for social proximity when two individuals who are collaborating have higher levels of social proximity, but the effect become insignificant with slightly lower levels of social proximity depending on where they set the threshold for their operationalization of social proximity. Similarly, Sorenson et al. (2006) suggested that organizational proximity acts as a proxy for social proximity within a collaboration network.

The qualitative interviews indicated that people collaborate within the Linux kernel with other people from their organization. While this collaboration often occurred on the mailing lists, there were also mentions of additional interactions with other employees using internal channels of communication or sometimes face to face discussions. This suggests that at least in some instances, organizational proximity may substitute for social proximity when conversations and collaboration occur between employees outside of participation in the same mailing list threads (social proximity). In other words, the coordination costs between employees of the same third party organization are lower even when those same individuals participate fewer of the same mailing list threads. This leads to the final hypothesis indicating that social and organizational proximities are substitutes.

Hypothesis 5: Two individuals who work for the same employer (organizational proximity) will have an increased likelihood of collaboration even when they participate in fewer of the same mailing list threads (social proximity).

These five hypotheses are tested by building on the relational event model from the previous chapter and adding interactions between the proximity variables. The next section contains details about the methods used in this chapter. This is followed by a results section describing the findings and a discussion section reflecting on the impact of the results from this chapter.

6.3. Methods

The methods for this final phase of the research build on the Phase 2 study described in the previous chapter. The dataset, variable operationalization, and modeling approach are identical in both studies. This chapter takes Model 3 from the Phase 2 study and adds proximity variable interactions based on the hypotheses just described in the previous section to create Model 4. Because the methods are mostly the same, this section provides a recap and summary, but more detailed information can be found in the previous chapter in Section 5.2 for the dataset, variable operationalization, and model. The variable interactions are new and will be described in more detail later in this section.

6.3.1. Data

The Phase 3 dataset used in this chapter is identical to the one used for Phase 2 in the previous chapter and is summarized in Table 7. To review, the dataset contains 10,513 replies to messages on the PCI mailing list over a time period starting on 2013-11-03 and concluding on 2015-11-01. This nearly two year period spans across 12 Linux kernel releases (3.12 to 4.3). Each reply in this event history dataset creates a collaboration event as a network tie between the person replying to the message, the ego, and the person being replied to, the alter. The data contain 654 total actors with 567 egos who replied to messages that were sent to the list by 574 alters with overlap due to some actors being both egos and alters for different collaboration events.

Table 7: Phase 2 and 3 dataset summary

Mailing List	Peripheral Component Interconnect (PCI) drivers at linux-pci@vger.kernel.org
Timeframe	12 releases from 2013-11-03 (3.12 release) to 2015-11-01 (4.3 release)
Events	10,513 replies: network tie / collaboration event
Actors	654 total actors: 567 egos and 574 alters

6.3.2. Model Estimation and Variables

As mentioned, the variable operationalization for this study is identical to Phase 2, but a brief recap is provided for completeness here.

Dependent Variable and Estimation

The dependent variable is the collaboration event operationalized as a reply to a message on the mailing list to determine which factors influence the likelihood that an ego will reply to a message previously posted by an alter on the mailing list. Because mailing list replies are not equally likely over the entire dataset, a matched case-control approach is used to compare each realized event (a message replied to by an ego) to a sample of five unrealized events made up of randomly selected messages from the previous seven days that an ego could have, but did not, select for a reply.

Independent Variables

The independent variables are calculated for both the realized event and the unrealized, sampled events to allow the model to determine which variables influence the likelihood of a collaboration event. A moving window of 63 days (median kernel release cycle length) is used for the variables that are calculated over time. The control, proximity, and network variables are described in more detail in Phase 2, Section 5.2.3, but are also summarized here in Table 8.

Table 8: Variable operationalization summary

Dependent Variable	Collaboration event operationalized as a reply to a message on the mailing list
Control Variables:	
Alter maintainer	1 if the alter is a maintainer, otherwise 0
Either maintainer	1 if the ego and/or the alter are maintainers, otherwise 0
Alter committer	1 if the alter has committed code within the moving window, otherwise 0
Either committer	1 if the ego and/or the alter have committed code within the moving window, otherwise 0
Ego to cc	1 if the ego was explicitly included in the “to” or “cc” field of the email that was replied to, otherwise 0
Proximity Variables	
Geographical	1 minus the normalized geographical distance calculated as the time zone offsets in seconds for a measure of Geographical proximity that ranges from 0 (maximum time zone distance) and 1 (same time zone)
Organizational	1 if both work for the same employer, otherwise 0
Institutional	1 if both work for the same type of third party organization, otherwise 0
Social	Number of times ego and alter participated in same thread within the moving window
Cognitive	Cosine similarity on contributions to areas of the source code with 0 indicating no overlap and 1 if both have contributed to exactly the same areas in the moving window
Network Variables:	
Repeated events	Number of times the ego replied to messages from the alter within the moving window
Participation shift	1 if the ego was the last person the alter replied to on the mailing list within the moving window
Recency effect	1/n with n defined as the number of people the alter emailed on the mailing list before the ego within the moving window
Transitive closure	Number of third parties that an ego has replied to where those third parties have also replied to the alter within the moving window
Cyclic closure	Number of third parties an alter has replied to where that third party has also replied to the ego within the moving window
Shared partnership inbound	Number of third parties who have replied to both the ego and the alter within the moving window
Shared partnership outbound	Number of times the ego and the alter have replied to messages by the same third party

Variable Interactions

There are several ways that variables can be related: *confounder* variables change the relationship between another independent variable and the dependent variable; *mediator* variables cause the dependent variable after being caused by another independent variable; and finally, a *moderator* variable changes the strength or direction of the relationship between another independent variable on the dependent variable typically using an interaction effect (MacKinnon et al. 2012; Dawson 2014). Investigating moderation effects using variable interactions in regression models has been commonly used in previous studies to determine the interrelationships between proximity dimensions (e.g. Singh 2005; Ter Wal 2014; Cassi and Plunket 2015; Crescenzi et al. 2016; Heringa et al. 2016). For example, cognitive proximity can be used as a moderator variable to determine the relationship with social proximity to influence the likelihood of collaboration. This is typically accomplished by using the product of two independent variables in a regression analysis to understand the effect on the dependent variable (Jaccard and Turrisi 2003), which in this example would include the product of cognitive proximity and social proximity to understand the effect on the likelihood of collaboration. The following variable interactions will be used to test each of the five hypotheses:

- Hypothesis 1: *Social Proximity * Cognitive Proximity*
- Hypothesis 2: *Organizational Proximity * Cognitive Proximity*
- Hypothesis 3: *Social Proximity * Geographical Proximity*
- Hypothesis 4: *Cognitive Proximity * Geographical Proximity*
- Hypothesis 5: *Organizational Proximity * Social Proximity*

While regression analysis results provide an indication of the direction of the effect with a positive or negative coefficient and the significance of the effect, this tells us very little about the nature of the effect across different levels of each variable; however, plotting the effect provides a way to visually interpret the effect at various levels of each interacting variable (Dawson 2014). Within the proximity literature, significant interaction effects with negative coefficients are typically interpreted as a substitution effect and positive coefficients are interpreted as a complementary effect between the interacted variables, but various plotting approaches are also used to illustrate the relationships between interacted variables (e.g. Cassi and Plunket 2014; Ter Wal 2014; Cassi and Plunket 2015; Heringa et al. 2016).

There are numerous tools available to assist in plotting the interactions; unfortunately, none of these tools can be used with the output from conditional logistic regression models (Ter Wal 2014), so a more manual approach was used to plot the effects. Dawson (2014)

provides an example of using a logistic regression equation and calculating the expected value of a dependent variable by substituting coefficients and variable values to plot an interaction effect. Dawson's (2014) equation is:

$$Y = \frac{e^{b0 + b1X + b2Z + b3XZ + \varepsilon}}{1 + e^{b0 + b1X + b2Z + b3XZ + \varepsilon}}$$

where Y is the probability of a successful outcome, b0 is the intercept, b1 is the coefficient for variable X, b2 is the coefficient for the variable Z, b3 is the coefficient of the interaction effect between variables X and Z, and ε is the error term. This approach can be slightly modified to accommodate a conditional logistic regression model as follows:

$$Y = \frac{e^{b1X + b2Z + b3XZ + \varepsilon}}{1 + e^{b1X + b2Z + b3XZ + \varepsilon}}$$

In conditional logistic regression and other stratified fixed effects models, the intercept drops out of the model (Kleinbaum 1994 p.114; Dougherty 2011 p.518), so *b0* is removed from the equation. This equation was implemented within R by inputting variable values and coefficients to get the expected values of the dependent variable (likelihood of a collaboration event) as probabilities to produce the plots that are used for interpretation of the interaction effects. Because there are many variables in the model and the variables are centered, the plots show the impact on the dependent variable at various levels of the two plotted independent variables when all other variables are held at their mean values.

Plotting interactions involving a binary variable, like organizational proximity, is straightforward, since it is relatively simple to plot a line for each of the two values across all possible values of a continuous variable, like social proximity. With two continuous variables, like social proximity and cognitive proximity, plotting becomes a bit more complex, so two plots are included for each of these cases with a subset of values from one variable as a moderator plotted against all possible values of the other variable. As shown in the results section, interpreting the two plots together provides insights that might not be obvious from just one of the plots.

A complementary relationship can be found when the plots show that higher levels of both interacted dimensions of proximity result in increases in the likelihood of collaboration. If the two dimensions of proximity are complements, both of the plots showing how each variable moderates the other will have positive slopes for each of the two variables, and higher levels of each dimension will have higher likelihoods of collaboration. This demonstrates an

additive relationship indicating an increase in the likelihood of collaboration as the two dimensions increase.

For substitution, proximity in one of the two dimensions compensates for a lack of proximity in another dimension. This can be demonstrated in the plots in two different ways. In one case, the lines for lower levels of one of the dimensions of proximity may appear above the lines for higher levels of that dimension of proximity, which indicates that lower levels have a higher likelihood of collaboration. In this case, the dimension with lower levels of proximity having a higher likelihood of collaboration is being substituted by the other dimension of proximity, which compensates for the substituted dimension. In the other case, the dimension of proximity being substituted would have lines with negative slopes in the plot, thus showing that the other dimension compensates for the substituted dimension with the negative slopes.

Unfortunately, the substitution or complement relationship is not always straightforward, since the two dimensions can be substitutes at some levels of proximity and complements at others, which is why looking beyond the sign and value of the interaction coefficient is important. The advantage of plotting the dimensions of proximity at various levels is that it becomes clear if both relationships are present at different levels of proximity. This can be indicated by having positive slopes of the lines for some values and negative slopes for other values with the negative slopes indicating substitution at those levels and complements at the levels with positive slopes. It can also be demonstrated with an inflection point where on one side of this point, lower levels of proximity have a higher likelihood of collaboration demonstrating substitution at those levels, and at the other side of the inflection point, higher levels of proximity have a higher likelihood of collaboration indicating a complementary relationship at those levels of that dimension.

6.4. Results

Table 9: Relational event model with interactions

Variables	Model 3			Model 4		
Alter Maintainer	-0.059	(0.015)	***	-0.044	(0.014)	**
Either Maintainer	0.218	(0.075)	**	0.161	(0.075)	*
Alter Committer	-0.217	(0.053)	***	-0.253	(0.061)	***
Either Committer	0.637	(0.145)	***	0.638	(0.148)	***
Ego To CC	2.618	(0.688)	***	2.597	(0.685)	***
Geographic Proximity	0.137	(0.074)	.	0.152	(0.088)	.
Organizational Proximity	0.627	(0.133)	***	0.944	(0.176)	***
Institutional Proximity	**-0.131**	**(0.049)**	**	**-0.078**	**(0.048)**	
Social Proximity	1.052	(0.243)	***	1.072	(0.242)	***
Social Proximity Squared	-0.050	(0.013)	***	-0.050	(0.012)	***
Cognitive Proximity	1.856	(0.403)	***	4.528	(0.920)	***
Cognitive Proximity Squared	-4.287	(0.888)	***	-1.991	(0.513)	***
Repeated Effect	**0.002**	**(0.001)**		**0.006**	**(0.002)**	**
Participation Shift	-0.163	(0.151)		-0.088	(0.156)	
Recency Effect	0.882	(0.227)	***	0.760	(0.217)	***
Transitive Closure	0.036	(0.010)	***	0.025	(0.008)	**
Cyclic Closure	0.086	(0.019)	***	0.078	(0.018)	***
Shared Partnership Inbound	-0.077	(0.018)	***	-0.066	(0.016)	***
Shared Partnership Outbound	-0.127	(0.029)	***	-0.126	(0.031)	***
Social x Cognitive Prox				-0.613	(0.134)	***
Organizational x Social Prox				-0.269	(0.056)	***
Organizational x Cog Prox				0.582	(0.286)	*
Geographic x Social Prox				0.153	(0.048)	**
Geographic x Cog Prox				-2.725	(0.738)	***
BIC	18033			17681		
Observations (realized events + controls)	63072			63072		
Realized Events	10512			10512		
Significance p < 0.001 '***' p < 0.01 '**' p < 0.05 '*' p < 0.1 '.' Robust Standard Errors are shown in parentheses						

Model 4, shown in Table 9, builds on the previous 3 nested models described in Phase 2 (see Table 6) with the addition of interaction effects implemented as products between the interacted variables. The Bayesian Information Criterion (BIC) scores indicate that Model 4 is an improvement over Model 3 as demonstrated by the lower BIC score for Model 4.

When the interaction effects were added in Model 4, most of the coefficients and significance remained similar to what was described in the previous chapter for Models 1 - 3; however, there were a few changes in the results when compared to Model 3. As expected, there were some changes to the coefficients for the effects, but the largest changes related to changes in significance for *Institutional proximity* and *Repeated effect*. First, *Institutional proximity* lost significance indicating that there is no evidence that working for the same type of institution has an independent effect on the likelihood of collaboration between two

individuals, which is consistent with the qualitative results from Phase 1. *Repeated effect* was positive and became significant, but with a coefficient of 0.006, indicating only a 0.6% increase in the likelihood of collaboration for each repeated event, the effect is quite small.

6.4.1. Social and Cognitive Proximities as Complements

Hypothesis 1 stated that two individuals who have participated in the same mailing list threads (social proximity) and have contributed to the same areas of the source code (cognitive proximity) are more likely to collaborate in the future. The interaction between social proximity and cognitive proximity was negative and significant, and Hypothesis 1 is supported. Increasing levels of cognitive proximity and higher levels of social proximity work together in a complementary fashion to increase the likelihood of collaboration as seen in Figure 9 and Figure 10.

Inspecting how social proximity moderates cognitive proximity in Figure 9 shows that at lower levels of cognitive proximity, small increases in the level of social proximity result in relatively large increases in the likelihood of collaboration. However, there are diminishing returns at higher levels of cognitive proximity where there is no additional increase in the likelihood of collaboration as social proximity increases. The positive slopes of the lines combined with higher levels of social proximity indicating higher likelihood of collaboration clearly indicates a complementary relationship.

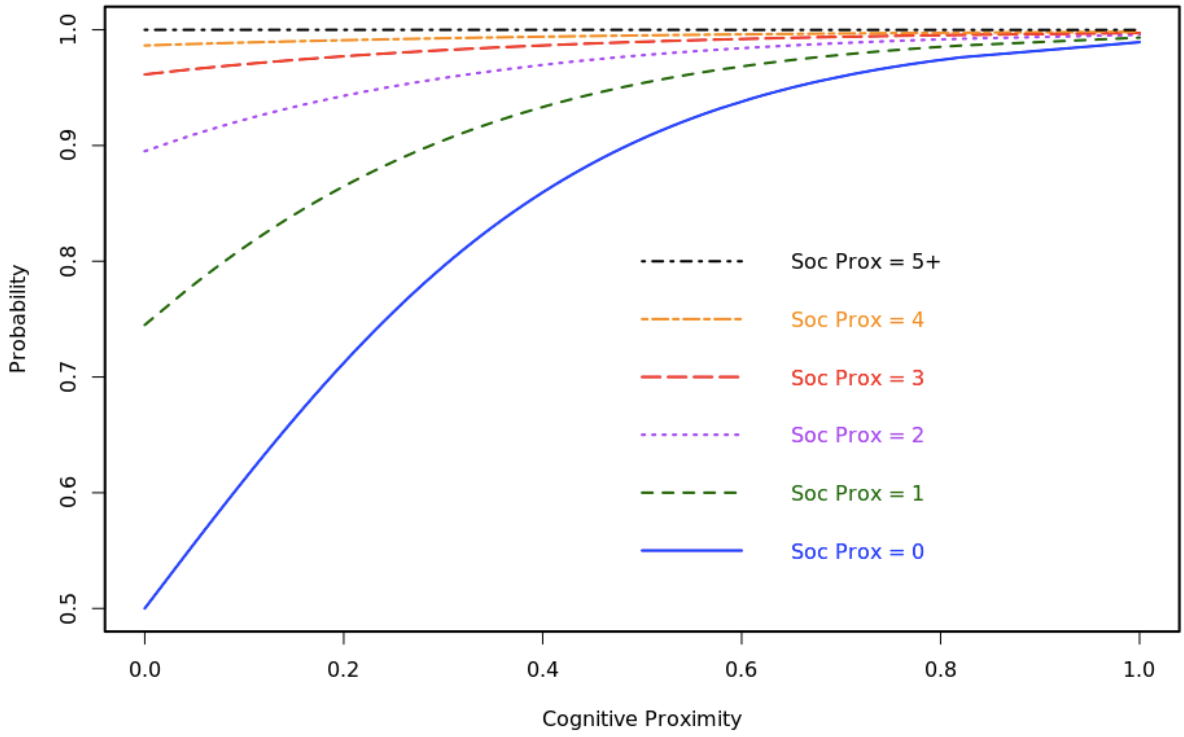

Figure 9: Social moderating cognitive

This is confirmed by looking at how cognitive proximity moderates social proximity in Figure 10. The likelihood of collaboration increases as the levels of social proximity and cognitive proximity increase together; however, there are diminishing returns at higher levels of social proximity with no further increase in the likelihood of collaboration from higher levels of social proximity. Again, the positive slopes of the lines combined with higher levels of cognitive proximity indicating higher likelihood of collaboration clearly indicates a complementary relationship.

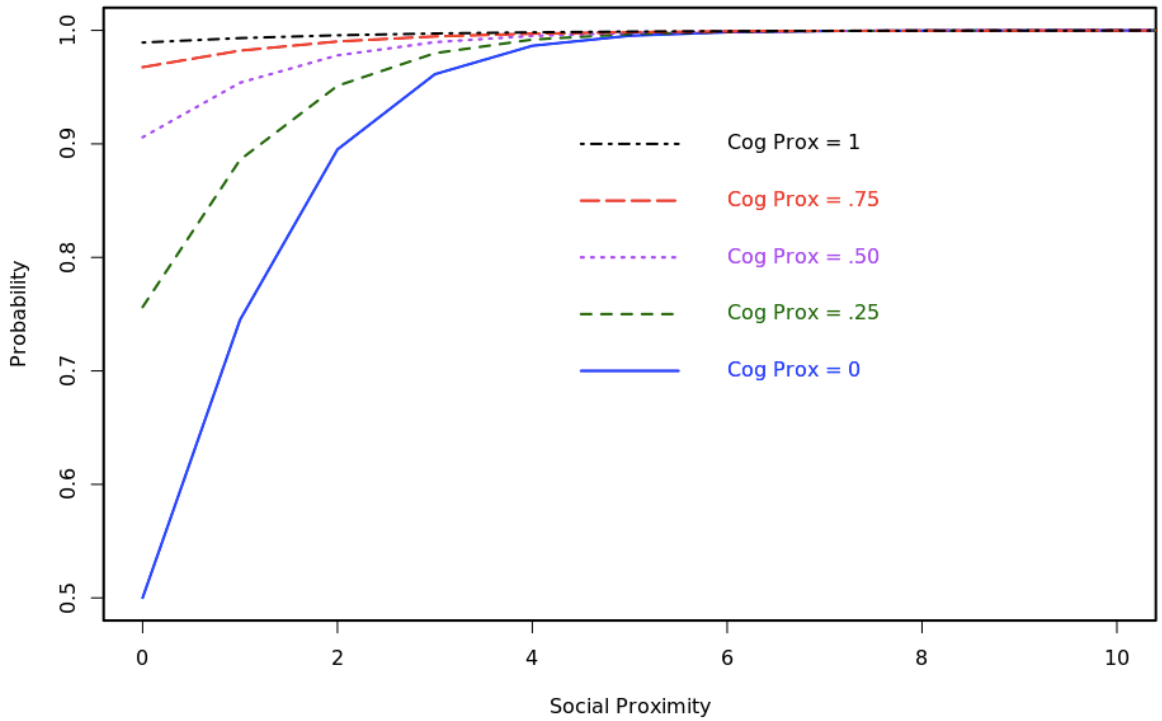

Figure 10: Cognitive moderating social

Based on the earlier mentioned proximity literature for substitutes and complements, a negative coefficient for the interaction between social and cognitive proximities would be expected to indicate a substitution effect. However, in this case, Figure 9 and Figure 10 clearly show that the relationship is complementary, which highlights the importance of plotting the interaction and looking beyond the coefficient sign of the interaction for interpretation of variable interactions.

This result has implications for real-world outcomes. Participants who have little to no social proximity and little to no cognitive proximity are much less likely to collaborate, but even small increases in either of these dimensions of proximity can result in relatively large increases in the likelihood of collaboration. By increasing both cognitive proximity and social proximity together in a complementary fashion, participants in a fluid organization can

quickly increase the likelihood of collaboration with small increases in these dimensions of proximity. Third party organizations seeking to increase collaboration between an employee and other members of the fluid organization should focus on increasing social proximity and cognitive proximity.

6.4.2. Organizational and Cognitive Proximities as Complements

Hypothesis 2 states that two individuals working for the same employer (organizational proximity) and having contributed to the same areas of the source code (cognitive proximity) become more likely to collaborate in the future. The interaction between organizational and cognitive proximities was positive and significant, and Hypothesis 2 is supported. As seen in Figure 11, two people with organizational proximity are more likely to collaborate at all levels of cognitive proximity. Furthermore, organizational proximity and cognitive proximity work together in a complementary fashion to increase the likelihood of collaboration; however, this starts to show diminishing returns for higher levels of cognitive proximity. The positive slopes of the lines combined with the higher level of organizational proximity indicating a higher likelihood of collaboration clearly indicates a complementary relationship.

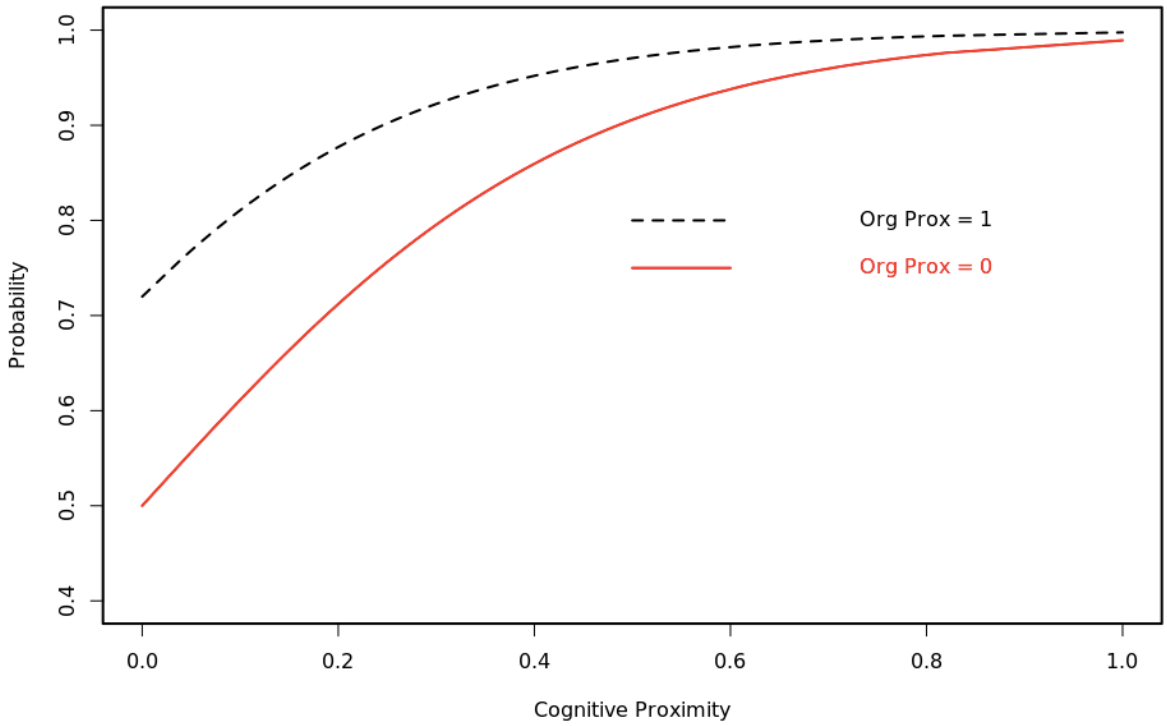

Figure 11: Organizational and cognitive

In practice, participants who are employed by the same third party organization are more likely to collaborate; however, by achieving very high levels of cognitive proximity,

participants who are not employed by the same third party organization become effectively just as likely to collaborate. In other words, when considering likelihood of collaboration, participants employed by the same third party organization have a collaboration advantage, but this advantage can be overcome for participants with very high levels of cognitive proximity. Third party organizations seeking to increase collaboration between employees and other participants within the fluid organization should focus on increasing cognitive proximity, which reinforces the implications in the previous section.

6.4.3. Social and Geographical Proximity as Complements

Hypothesis 3 states that two individuals who have participated in the same mailing list threads (social proximity) will have an increased likelihood of collaboration even when they are not in similar time zones (geographical proximity). The interaction between social and cognitive proximities was significant; however, it was positive, and the impact of geographical proximity was negligible compared to social proximity; therefore, the third hypothesis is rejected. The expected outcome described in Hypothesis 3 was that social and geographical proximities would have a substitute relationship. While there is a relationship between social and geographical proximities, these two dimensions of proximity do not act as substitutes, but instead are complementary; however, because the role of geographical proximity is negligible compared to social proximity, social proximity is the primary influence, so the interaction between these two dimensions likely has little to no effect on real-world outcomes.

When looking at how geographical proximity moderates social proximity in Figure 12, as social proximity increases, the likelihood of collaborating increases sharply before showing diminishing returns and leveling off at higher levels of social proximity. The positive slopes of the lines combined with higher levels of geographical proximity indicating higher likelihood of collaboration indicates a complementary relationship. This confirms that social and geographical proximity work together as complements to increase the likelihood of collaboration, but most of the effect is a result of social proximity with geographical proximity contributing only a negligible increase, thus showing that the interaction has little to no effect on outcomes beyond the influence of social proximity alone.

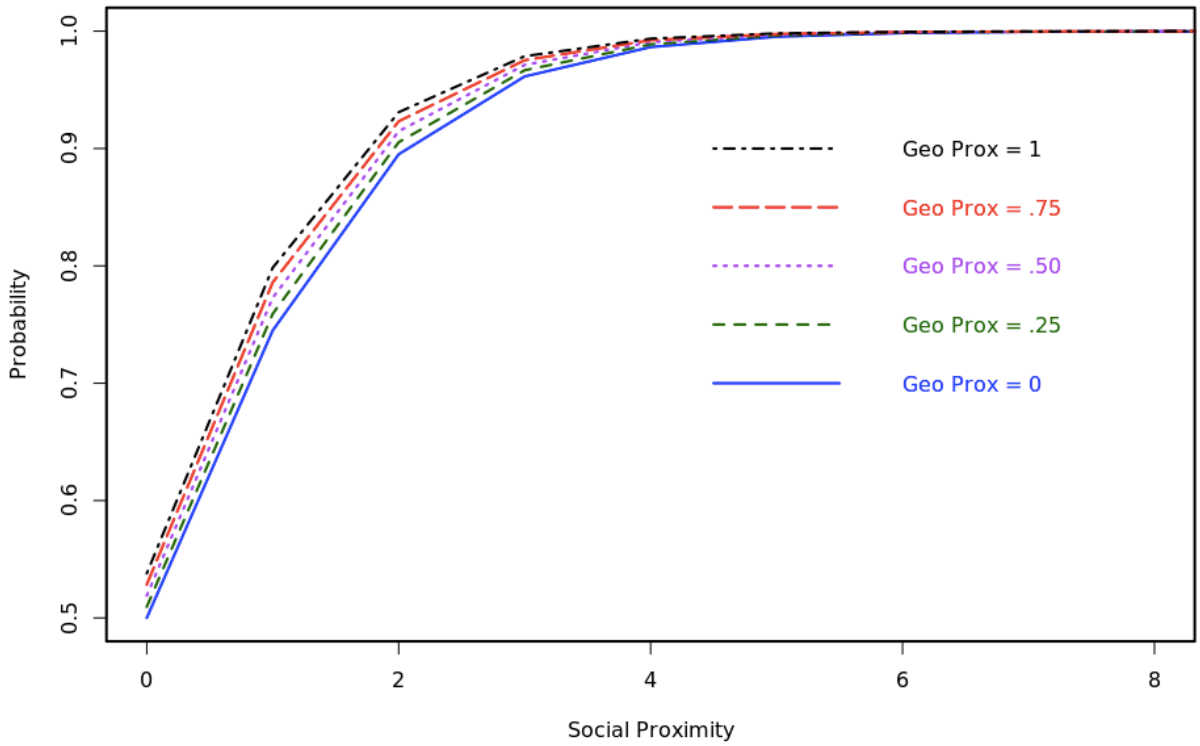

Figure 12: Geographical moderating social

When looking at how social proximity moderates geographical proximity, Figure 13 shows that as levels of geographical proximity increase, there is a slight increase in social proximity. The slight positive slopes of the lines combined with higher levels of social proximity indicating higher likelihood of collaboration indicates a complementary relationship. However, the slope of the social proximity lines across various levels of geographical proximity are almost flat showing again that while they are complementary, geographical proximity contributes only a negligible increase in the likelihood of collaboration for this interaction. Thus the interaction between social proximity and geographical proximity has little to no effect on practical outcomes beyond the effect of social proximity alone.

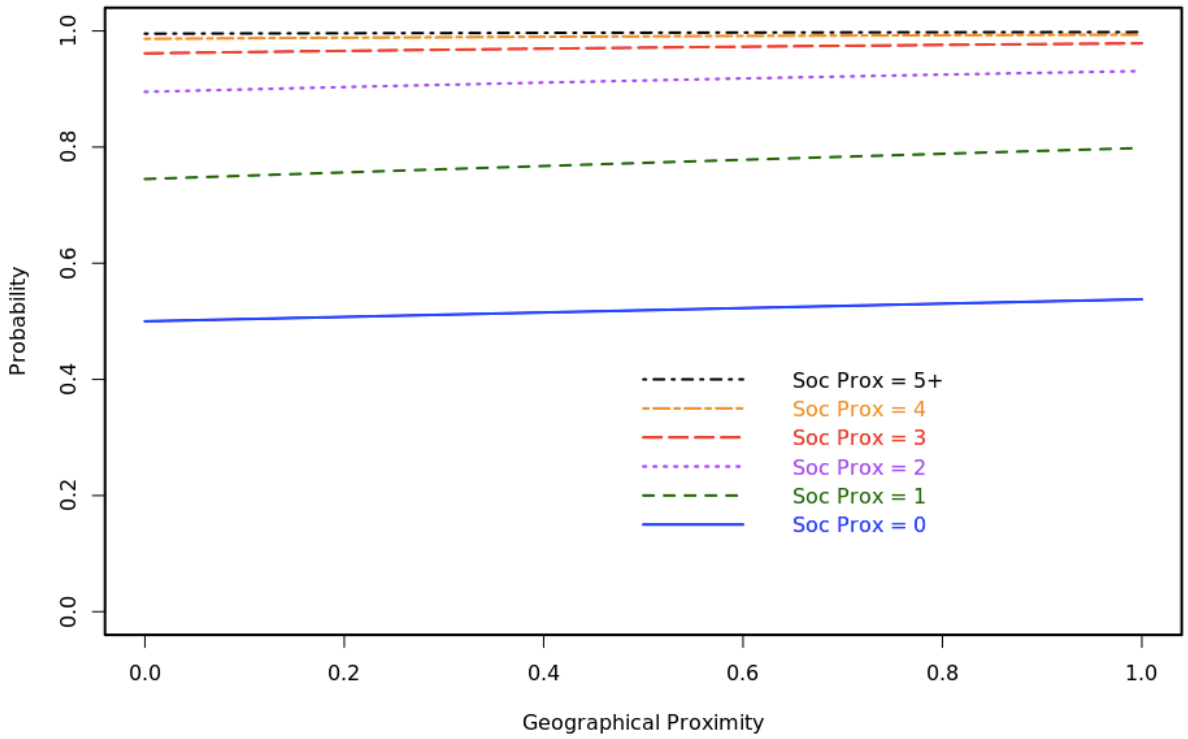

Figure 13: Social moderating geographical

The implication of this result is that increases in collaboration can be gained by focusing on increasing social proximity and that increasing geographic proximity adds little to no effect on practical outcomes. Again, this reinforces the implication from section 6.4.1 showing that third party organizations seeking to increase collaboration between employees and other fluid organization participants should focus on increasing social proximity. This also implies that geographical location of employees may not be important for those employees whose primary responsibility is to collaborate within a fluid organization.

6.4.4. Cognitive and Geographical as Substitutes

Hypothesis 4 states that two individuals who contribute to the same areas of the code (cognitive proximity) will have an increased likelihood of collaboration even when they are not in similar time zones (geographical proximity). The interaction effect between cognitive and geographical proximities was negative and significant. Hypothesis 4 is partially supported with cognitive proximity acting as a substitute for geographical proximity at most levels of cognitive proximity.

Looking at how cognitive proximity moderates geographical proximity, Figure 14 shows that as geographical proximity increases, the likelihood of collaboration only increases at the lowest level of cognitive proximity. As geographical proximity increases, there is a decline in the likelihood of collaboration indicated by negative slopes at higher levels of

cognitive proximity, thus indicating that cognitive proximity can act as a substitute for geographical proximity at most levels of cognitive proximity. However, when two individuals have very low levels of cognitive proximity, the increase in slope for the "cognitive proximity = 0" line shows that the relationship is complementary with increases in the likelihood of collaboration at higher levels of geographical proximity only in the case where there is no cognitive proximity. Nonetheless, the increase in the slope is very small, which indicates that this would have a negligible effect on outcomes.

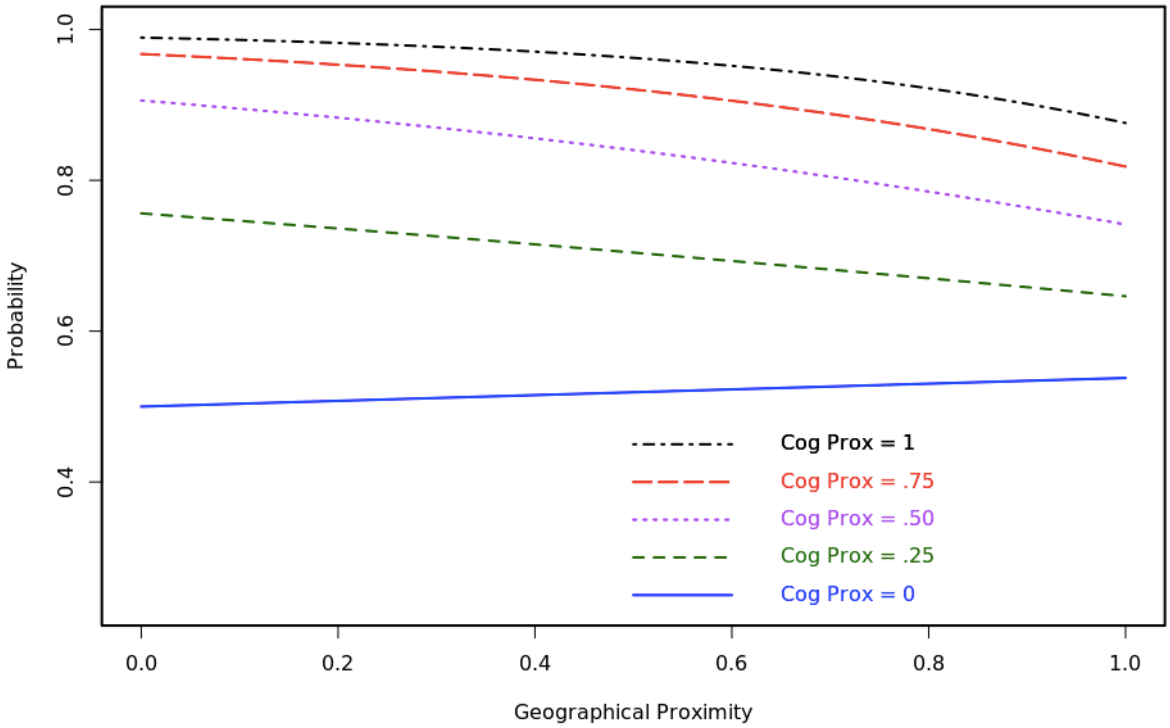

Figure 14: Cognitive moderating geographical

This relationship can be seen even more clearly by looking at how geographical proximity moderates cognitive proximity in Figure 15 to see that the inflection point where the relationship moves from complement to substitute is not just at 0, but also includes very low levels of cognitive proximity just below 0.1. At cognitive proximity levels above this inflection point, lower levels of geographical proximity have a higher likelihood of collaboration, but only as levels of cognitive proximity increase, thus again illustrating that cognitive proximity substitutes for geographical proximity at most levels of cognitive proximity.

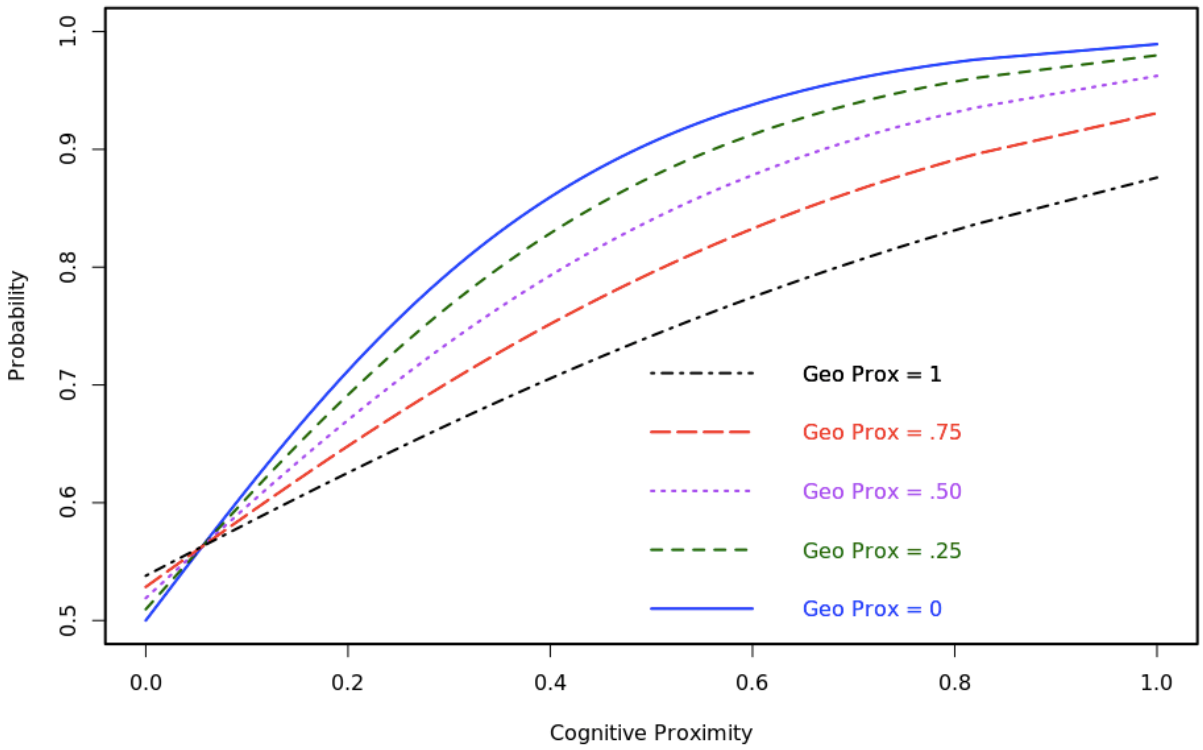

Figure 15: Geographical moderating cognitive

This also highlights the importance of investigating the moderation in both directions by plotting the interaction between two continuous variables both ways. By looking only at how cognitive proximity moderates geographical proximity in Figure 14, it would have been easy to conclude that social and geographical proximity are substitutes only when cognitive proximity is 0 without realizing that they are also substitutes at levels slightly higher than 0.

However, the real-world implications of these results should be carefully considered. While the results indicated that for participants with little to no cognitive proximity, geographical proximity interacts with cognitive proximity in a complementary fashion; nevertheless, the effect is very small, which indicates that the portion of the interaction attributed to geographical proximity would have a negligible effect on outcomes as compared to increases in cognitive proximity. This is consistent with the implications from section 6.4.3. With cognitive proximity substituting for geographical proximity at most levels, this reinforces the implications from sections 6.4.1 and 6.4.2 indicating that third party organizations seeking to increase collaboration between employees and other participants should focus on increasing cognitive proximity.

6.4.5. Organizational and Social Proximity as Substitutes

Hypothesis 5 states that two individuals who work for the same employer (organizational proximity) will have an increased likelihood of collaboration even when they participate in fewer of the same mailing list threads (social proximity). The interaction effect between organizational and social proximities was negative and significant, but Hypothesis 5 is only partially supported depending on the level of social proximity. Because the effect is quite small, the inflection point is not visible in Figure 16, but it can be shown in the exploded section in Figure 17 where two people who work for the same organization will have increased likelihood of collaboration at lower levels of social proximity. At higher levels of social proximity, Figure 17 demonstrates that social proximity can substitute for organizational proximity as indicated by the line representing no organizational proximity appearing above the line indicating that organizational proximity exists; however, the difference in the lines is so small as to be negligible with respect to practical outcomes. It also indicates that most of the increase in the likelihood of collaboration is a result of increases in social proximity with organizational proximity contributing very little to the effect and showing diminishing returns at higher levels of social proximity with no further increase after social proximity reaches higher levels.

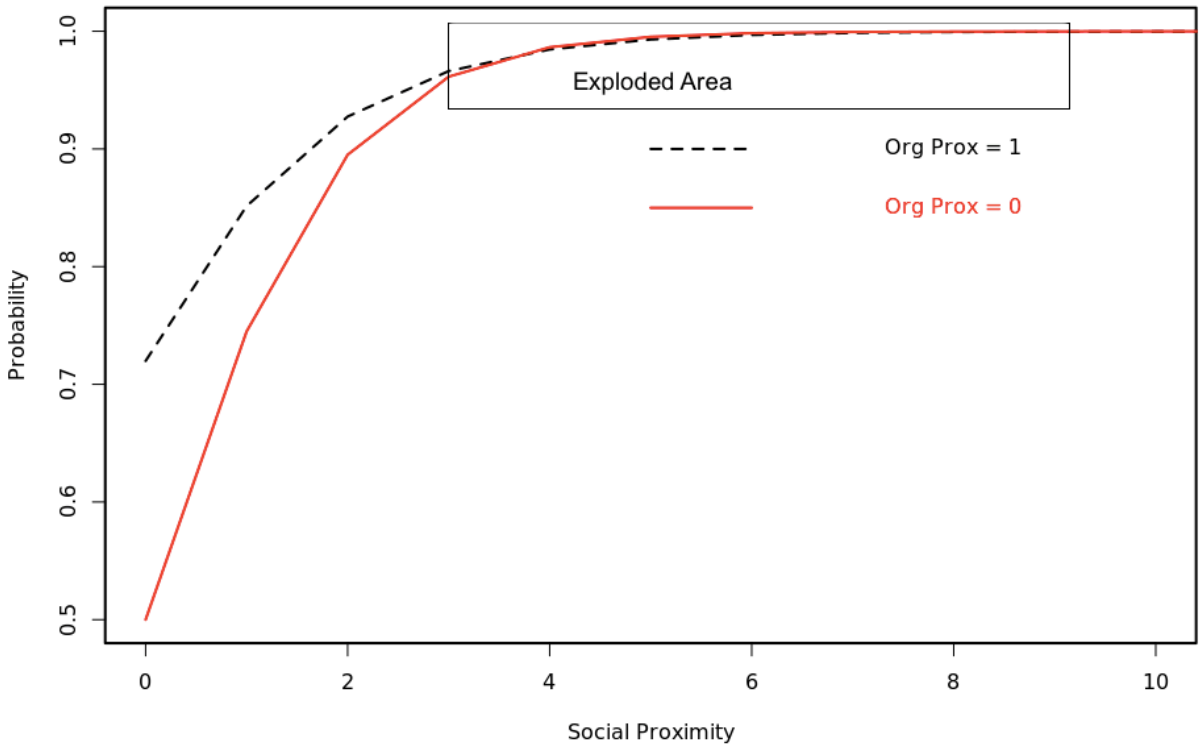

Figure 16: Organizational and social

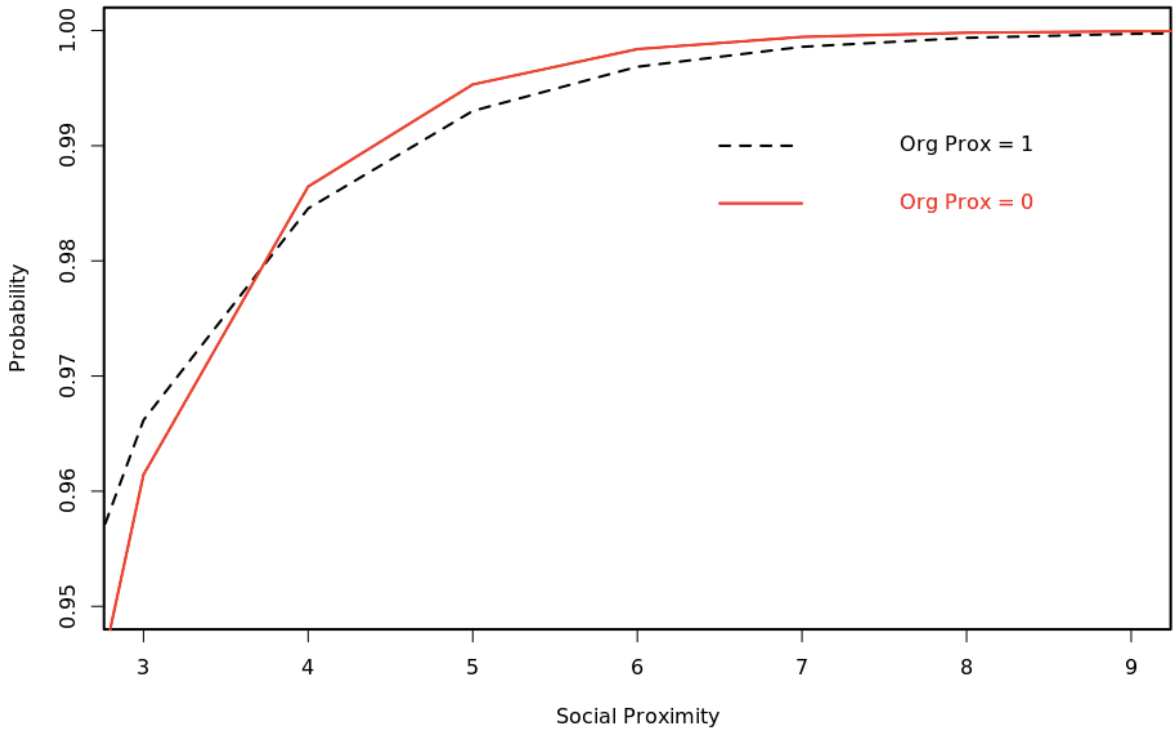

Figure 17: Organizational and social (exploded)

Like with previous interactions, in practice, most of the effect in this interaction comes from increases in social proximity with a small to negligible effect from organizational proximity. Participants who are employed by the same third party organization only have a collaboration advantage at very low levels of social proximity, but this advantage is quickly overcome for participants with even a small amount of social proximity. Third party organizations seeking to increase collaboration between employees and other fluid organization participants should focus on increasing social proximity, which reinforces the implications in the previous sections.

6.5. Discussion

The goal of this chapter is to look beyond the influence of individual dimensions of proximity to understand how proximity dimensions work together to influence the likelihood of collaboration in a fluid organization. The research confirms that dimensions of proximity are interrelated in a variety of ways.

Cognitive proximity and social proximity are shown to be complementary with both of them increasing together to contribute to the likelihood of collaboration. In other words, two individuals who have participated in the same mailing list threads (social proximity) and have contributed to the same areas of the source code (cognitive proximity) are more likely to collaborate in the future; however, there are diminishing returns where additional increases in

the likelihood of collaboration level off as social proximity and cognitive proximity increase to higher levels. It is to be expected that people with similar knowledge resulting from working on similar technologies (cognitive proximity) would be discussing those contributions (social proximity) such that more cognitive proximity and more social proximity would lead to lowered coordination costs between individuals driving additional increases in the likelihood of collaboration over time. The diminishing returns is likely a result of two people becoming increasingly aligned through both social and cognitive dimensions, thus not requiring increased collaboration as they maintain their working relationship. The interaction between social proximity and cognitive proximity highlights the importance of plotting interactions for interpretation. Within the proximity literature, a negative coefficient is often interpreted as a substitution (Cassi and Plunket 2014), but the plots for this interaction clearly showed a complementary relationship with social and cognitive proximity working together to increase the likelihood of collaboration.

The results for the two interactions with geographical proximity were interesting. While the interactions with cognitive proximity and social proximity were significant, geographical proximity contributes very little to the combined effect compared to the other variable in the interaction. There are a couple of potential explanations for this result. Since the Linux kernel is a decades old project, this aligns with Ter Wal's (2014) finding that geographical proximity is most relevant in the early stages becoming less influential over time as network relationships mature. It is also consistent with the results from the interviews in Phase 1 stating that because collaboration occurs online using mailing lists, time zones and location did not matter much in this setting. It is also worth noting that the standalone effect for geographical proximity is not significant in Model 3 or Model 4. This implies that geographical proximity makes only a very small or negligible contribution when combined with social proximity where it is complementary or with cognitive proximity where cognitive proximity can substitute for geographical proximity. This leads to the conclusion that the importance of geographical proximity may be overstated, at least for collaboration within some settings, including fluid organizations. Because the effect of geographic proximity in the interactions is negligible compared to social proximity and cognitive proximity, third party organizations seeking to increase collaboration would have a better outcome by focusing on increasing social proximity and cognitive proximity.

This research shows that cognitive and organizational proximity are complementary with increased alignment on the cognitive dimension and employment at the same organization working together to increase the likelihood of collaboration, but this has

diminishing returns at higher levels of cognitive proximity. Since third party organizations tend to focus on specific technologies, it is expected that working on similar technologies and working for the same employer would lower coordination costs and be complementary. The diminishing returns at higher levels of cognitive proximity demonstrate that two people who have high levels of experience working on the same technologies, there is little to no additional benefit gained from working for the same employer that would drive further increases in collaboration. This has important implications for fluid organizations because while employer affiliation can lower the coordination costs of collaboration, if two individuals have enough common understanding of the same technologies with high levels of cognitive proximity, employer affiliation has a reduced impact on further increases in the likelihood of collaboration. Investing time in fluid organizations and developing cognitive proximity with other people outside of their employer provides a collaborative environment that allows people employed by many third parties organizations to benefit.

Organizational proximity and social proximity are shown to have a complex relationship that is complementary at lower levels of social proximity and substitution for other levels before approaching a point of diminishing returns. Two people who work for the same third party organization will have an increased likelihood of collaboration at lower levels of social proximity as compared to people who do not share the same employer, so there is an advantage for the likelihood of collaboration if people work for the same employer. However, this advantage begins to disappear after two people have participated in several threads together, and the likelihood of collaboration begins to show diminishing returns. For fluid organizations where participants are affiliated with a variety of different employers, the cost of collaboration is reduced when two people work for the same employer and have some social proximity, but higher levels of social proximity can overcome the benefit of having organizational proximity. This reinforces the point in the previous paragraph that investing time in fluid organizations and developing increased proximity along social and cognitive dimensions builds common ground that facilitates collaboration with other people outside of their employer and allows people employed by many third party organizations to benefit from participation in the fluid organization.

This research demonstrates that interrelationships between dimensions of proximity, including substitutes and complementary relationships, influence the likelihood of collaboration within a fluid organization. This shows that for fluid organizations, investigating individual dimensions of proximity provides only a portion of the overall picture and that

relationships between dimensions should also be explored to better understand the likelihood of collaboration.

CHAPTER 7. CONCLUSIONS

The aim of this research is to provide answers to the questions, "How do participants who are employed by third party organizations collaborate within a fluid organization?" and "What is the role of proximity in these collaborations?" These questions are explored across the three phases of this research project.

The first question is answered primarily in Phase 1 of the research, and had implications for the design of the other two phases. One of the reasons this empirical setting was selected is because most of the participants in this fluid organization are employed by third party organizations (Corbet and Kroah-Hartman 2017), which is part of the focus of this first question. Despite being employed by a variety of third party organizations, the participants consider their affiliation with the Linux kernel as a fluid organization to be more important than their employer affiliation. These individuals receive little direction from their employer as they collaborate within the Linux kernel, but they are occasionally asked to do specific work that supports the employer's products or services, and as a result, the employer only has indirect influence on the Linux kernel. People collaborate with other individuals across third party organizations, including their competitors, so they try to collaborate as individuals, rather than focusing on employer affiliations. This collaboration occurs primarily on the Linux kernel mailing lists. In sum, participants who are employed by third party organizations collaborate primarily on mailing lists with other individuals working for a variety of employers that have only an indirect influence. These participants have considerable autonomy for how they participate in this fluid organization, which they identify with more closely than with their employer.

The second question, "What is the role of proximity in these collaborations?" is addressed across all three phases of the research using both qualitative and quantitative methods, and this appears to be the first study using Boschma's (2005) five dimensions of proximity to investigate collaboration within this setting. Across all phases of the research, cognitive and social proximities are consistently shown to increase the likelihood of collaboration; however, both are subject to diminishing returns with the likelihood of collaboration leveling off at a certain point where additional proximity no longer produces the same gains. Individuals also tend to be more likely to collaborate with others who work for the same employer; however, when combined with social or cognitive proximity, the increase in likelihood is higher at lower levels of social or cognitive proximity. The results for geographical proximity were mixed with the qualitative interviews indicating that time zones

and locations do not matter in this setting; however when considered in combination with cognitive or social proximity, geographical proximity may provide a small increase in the likelihood of collaboration, but only at low levels of cognitive and social proximity. In this setting, there is no consistent evidence that institutional proximity influences the likelihood of collaboration. In the final model, Model 4, the effect was insignificant, which is consistent with the qualitative interviews from Phase 1 indicating that the type of institution is not important for collaboration. In sum, cognitive, social, and organizational proximities have the biggest impact on collaboration with possibly a small contribution from geographical proximity and no consistent evidence that institutional proximity influences collaboration within this fluid organization.

7.1. Contributions and Implications for Theory

As organizations become increasingly fluid through the use of new technologies, networked societies, and alternative working arrangements (Dobusch and Schoeneborn 2015), contributing to the literature for fluid organizations becomes increasingly important as approaches to organizational analysis evolve to include more flexibility and fluidity. These results contribute to the literature on fluid organizations in several important ways.

First, this research proposes five criteria to determine whether an organization is a fluid organization and then uses this criteria to demonstrate that the Linux kernel is a fluid organization. The literature on fluid organizations is diverse with a wide variety of names and definitions used to describe fluid organizations. This research reviews some of the more common names and definitions to derive five criteria required for fluid organizations as described in Table 2: organization, flexible hierarchy, flexible boundaries, organic collaboration, and pronounced role of networks. This criteria is then applied to the Linux kernel to establish that the Linux kernel meets the criteria to be defined as a fluid organization. The Linux kernel meets the four criteria for an *organization* described in Table 1: affiliation, collective resources, substitutability, and recorded control. The *hierarchy* is flexible and evolves as needed with people *collaborating* organically across *boundaries* while relying on their *network* for collaboration. Thus, the Linux kernel is shown to be a fluid organization.

Second, this research demonstrates that proximity theory can be used effectively as a theoretical lens to better understand intraorganizational collaboration in fluid organizations. In traditional organizations, collaboration can be enforced by the hierarchy; however, the flexible boundaries and evolving structures of fluid organizations require that participants rely on common ground to facilitate effective collaboration, and this research shows that proximity

theory is one way of understanding this common ground using Boschma's (2005) five dimensions of proximity and their interrelationships.

Third, this research adds to the body of knowledge on fluid organizations by investigating the impact of third party organizations on collaboration. This is directly related to the organizational proximity results, but since it was identified as a gap in the Literature Review, it is highlighted here as a specific contribution. Open source software, for example, has often been used as the setting to understand fluid organizations, but most of the research fails to consider the influences that employer affiliation might have on contributions and collaboration. Because employers can influence how employees spend their time, this affiliation should be factored into the analysis of fluid organizations. During the Phase 1 interviews, participants mentioned that their employers sometimes asked them to work on specific areas of the Linux kernel, thus influencing the areas where they collaborate with others. The research also shows that individuals tend to be more likely to collaborate with others who work for the same employer, but that by increasing social and / or cognitive proximity, they were as likely to collaborate with people outside of their organization. This research shows that employer affiliation influences collaboration and should be included in the analysis of collaboration within fluid organizations.

The research also makes several contributions in addition to the ones within the literature on fluid organizations. First, this research demonstrates that alternative operationalization of proximity measurements can be used to investigate collaboration within a fluid organization. In this study, social, cognitive, and geographical proximity were operationalized in ways that were consistent with conceptual definitions in the proximity literature, but that map more directly to this empirical setting. Geographical proximity is typically measured using physical distances between locations; however, since physical location is somewhat irrelevant when collaboration occurs virtually within online communities (Boschma 2005; Torre 2008), this research used time zones to provide insight into geographical proximity as it relates to when people are more available to collaborate based on their location (O'Leary and Cummings 2007). In this setting, the interviews from Phase 1 indicated that knowledge of particular areas of the source code within subsystems was important for cognitive proximity, so cognitive proximity was operationalized using cosine similarity between the areas within the source code where two people have contributed. Hardeman et al. (2015) used cosine similarity on contributions to similar journals to operationalize cognitive proximity; however, to date, no research has been found using source code contributions as a measure of cognitive proximity, so this would appear to be a new and

unique way to operationalize cognitive proximity. Likewise, at the time of writing, no research has been found using threads to measure social proximity, so this may be the first time that participation in mailing list *threads* (as opposed to individual emails) has been used to operationalize social proximity, instead of more typical measures like using network distance (i.e. shortest path, geodesic distance) to determine social proximity.

Second, in the context of the analysis of interrelationships between proximity dimensions, this research makes additional methodological contributions. These findings demonstrate the importance of looking beyond coefficient signs when interpreting interactions. The negative coefficient of the interaction between the social and cognitive proximities would typically indicate a substitution relationship, but plotting the interaction showed a clear complementary relationship. Also, when considering interactions for two continuous variables, this study shows that it is important to investigate the variables in multiple ways by reversing the moderating variable. This provides valuable insights into the interaction at all levels of each variable to avoid possible misinterpretation at some levels.

7.2. Implications for Practice

Since the primary way for third party organizations to contribute to open source projects is by having employees participate, and software developers who work at third party organizations are increasingly contributing to these projects (Roberts et al. 2006; Jensen and Scacchi 2007), employers who benefit from open source projects should consider how their employees can participate in these fluid organizations. Employers could validate these findings in real-world settings by cautiously applying the following recommendations.

First, the third party organization should find a suitable employee to participate or possibly hire someone who already contributes to the project. Based on the cognitive proximity findings, it is important to find someone with subject matter expertise, knowledge, and experience that is appropriate for the areas of the project where they are expected to contribute, and participating in multiple areas might allow them to generate more innovative ideas. If the primary methods of collaboration are online, similar to the Linux kernel, the employee could be located anywhere, but it might be beneficial to be in a time zone close to other key contributors if they do not yet have much social and cognitive proximity with those key people. Next, the employee should be encouraged to attend relevant conferences and form professional relationships outside of the main project communication channels, since these results suggest that collaboration is facilitated by social (Ter Wal 2014) and temporary geographical proximity (Torre 2008).

Results from several of the empirical setting specific controls also have interesting implications for collaboration within fluid organizations. Considering the results for maintainers and committers together with collaboration events increasing in likelihood for people who commit code or are maintainers, one possible interpretation of these results is that experienced people are more likely to provide feedback on contributions from less experienced participants. This could explain why more experienced people are more likely to create collaboration events, and why those experienced participants are less likely to be the target for collaboration. Third party organizations wanting to have greater influence in the Linux kernel should consider focusing on maintainers and committers when employing developers to work on the Linux kernel or encourage existing employees to gain the experience required to move into these roles. These experienced developers through increased collaboration can provide greater influence and increased visibility on topics of interest to their employer, thus benefiting the third party organization employing these experienced developers.

While it cannot be concluded that geographical proximity, in this case time zones, do not influence collaboration at all, the effect seems to be weak. Third party organizations who employ people to participate in fluid organizations should consider whether there is much benefit to having employees in the same location or time zone. Employers might consider hiring or allowing existing employees to work from a variety of locations or time zones for Linux kernel contributions. This flexibility could make it easier to hire or retain in demand talent, especially for experienced contributors who regularly commit code or maintain portions of the kernel. Employers who recruit and retain these developers might benefit from drawing from a larger and more diverse pool of candidates outside of the regions where they have physical offices without jeopardizing collaboration within the fluid organization if they focus on making sure that these employees are building social and cognitive proximities with key collaborators outside of their employer. Regardless of their physical location, employees working in fluid organizations should be focused on building cognitive and social proximity with key participants as part of the collaboration process to help reduce the costs of collaboration over time.

CHAPTER 8. FUTURE WORK

8.1. Limitations

The primary limitation of this research as a whole is that this is a study of a single fluid organization, the Linux kernel. Because so many of the participants are employed to perform this work (Corbet and Kroah-Hartman 2017), it was a reasonable choice to study this phenomenon. However, because the research is based on a single case, the results should not be broadly generalized to other fluid organizations, or even other open source software projects until similar findings are found in other projects. Further research should be conducted confirm these findings in other fluid organizations. Additionally, the quantitative analysis from Phases 2 and 3 is based on one subsystem mailing list within a single fluid organization, the Linux kernel. Thus, further research should be conducted confirm these findings in other fluid organizations. Caution would be urged even in generalizing these results to other subsystems within the kernel until the results are replicated for other mailing lists.

Another limitation of the quantitative analysis in Phases 2 and 3 is that the social and geographical proximity measurements are subsets of the full phenomenon due to the virtual / online nature of this fluid organization. Ideally, social proximity would be operationalized to include activities beyond participating online in the same threads and could be expanded to include participation at conferences, in-person relationships and other communication between people that occurs outside of the mailing list being studied. Similarly, geographical proximity could include measures of physical location in addition to time zones. The addition of physical locations and real-world social proximity measurements in other types of fluid organizations where these data are available would be an interesting exercise for further study.

The final limitation is that this research only looks at interactions between dimensions of proximity. It would be interesting to look at interactions between proximity and other measures, like network effects to determine whether there are interrelationships between other measures that impact the likelihood of collaboration.

There were also two additional limitations in earlier phases that were addressed as part of the research in a later phase. First, Phase 1 is based solely on interview data, and the results were not confirmed using other types of data. Because open source projects work in the open

with publicly archived conversations in mailing lists and source code repositories, these data sources could be used to validate this research on fluid organizations. This limitation was addressed in Phases 2 and 3. Second, the proximity literature demonstrates that dimensions of proximity are often interrelated (Boschma 2005; Balland et al. 2015; Crescenzi et al. 2016; Heringa et al. 2016), but interactions between dimensions of proximity were not considered in Phase 2. Based on variable correlations and results from Model 3, Phase 2 indicated that there might be some relationships between dimensions of proximity for collaboration within fluid organizations. This limitation was explored and addressed in Phase 3.

8.2. Further Research

While this research makes important contributions to the literature on fluid organizations, several areas highlighted as limitations would benefit from additional research. First, this research could be replicated using additional Linux kernel subsystem mailing lists to better understand whether these results apply to other areas of the Linux kernel outside of the PCI subsystem. Beyond the Linux kernel, this research should also be replicated in other open source projects and other fluid organizations to improve generalizability. Second, it would be interesting to replicate this research using fluid organizations that benefit from having additional proximity data sources that include in-person social proximity measures and physical geographical proximity. Third, interactions could be investigated between proximity variables and other measures, like network effects, to determine whether there are other interrelationships that impact the likelihood of collaboration beyond just the proximity interrelationships.

The biggest challenge for Phases 2 and 3 of this research is that the relational event model approach proved to be much more difficult than anticipated. Initially the plan was to use Butts' (2008) "relevent" software package in R; however, this software is not capable of scaling to the number of events / actors even in a two year dataset for a single Linux kernel mailing list. This required more manual statistics work to implement the approach using a case-control, conditional logit model with sampling for unrealized events as a comparison for each realized event. This is not a commonly used approach, especially in R, and while clogit and coxph tools are available to perform the analysis, many other tools used for plotting or diagnostics do not work with the model output from these tools. Because these statistical methods are not a frequently used, there were few examples and limited expertise within the faculty to provide guidance. With research and time, these challenges were overcome, but

additional verification and validation of these results using more common statistical methods and other approaches would be a welcome addition to build on this research.

There are also several other topics that would be interesting additions to this research in future studies. This research found that more experienced people are more likely to create collaboration events, which leads to questions about leadership within fluid organizations that could be explored. For example, “How do leaders emerge within a fluid organization?” and “How to leaders fit within the overall network structure of a fluid organization?” This study also found that people are more likely to work with others who work for the same employer (organizational proximity), but it would also be interesting to understand other organizational dynamics. For example, “How do participants who are employed by competing third party organizations collaborate?” and “How do participants who are employed by third party organizations with close partnerships collaborate?” It would also be interesting to explore the content of the contributions, rather than just which people collaborated. A sentiment analysis to investigate whether the overall tone of the message influences collaboration could answer the question, “How does the positive, negative, or neutral tone of a message influence collaboration?” Any of these would be worthwhile extensions to this research for future work.

REFERENCES

Agneessens, F. and Wittek, R., 2012. Where do intra-organizational advice relations come from? The role of informal status and social capital in social exchange. *Social networks*, 34 (3), 333–345.

Agrawal, A., Cockburn, I., and McHale, J., 2006. Gone but not forgotten: knowledge flows, labor mobility, and enduring social relationships. *Journal of Economic Geography*, 6 (5), 571–591.

Ahrne, G., 1994. *Social Organizations: Interaction Inside, Outside and Between Organizations*. London: SAGE.

Ahrne, G. and Brunsson, N., 2005. Organizations and meta-organizations. *Scandinavian Journal of Management*, 21 (4), 429–449.

Ahrne, G. and Brunsson, N., 2010. Organization outside organizations: the significance of partial organization. *Organization*, 18 (1), 83–104.

Ahuja, M.K. and Carley, K.M., 1999. Network Structure in Virtual Organizations. *Organization Science*, 10 (6), 741–757.

Alexy, O., Henkel, J., and Wallin, M.W., 2013. From closed to open: Job role changes, individual predispositions, and the adoption of commercial open source software development. *Research policy*, 42 (8), 1325–1340.

Allen, T.J., 1977. *Managing the flow of technology: technology transfer and the dissemination of technological information within the R&D organization*. Cambridge, MA: Massachusetts Institute of Technology.

Allen, T.J. and Henn, G.W., 2007. *The Organization and Architecture of Innovation: Managing the flow of technology*. Amsterdam: Elsevier.

Ashkenas, R., Ulrich, D., Jick, T., and Kerr, S., 2002. *The Boundaryless Organization: Breaking the Chains of Organizational Structure*. 2nd ed. San Francisco: Jossey-Bass.

Balland, P.A., 2012. Proximity and the Evolution of Collaboration Networks: Evidence from Research and Development Projects within the Global Navigation Satellite System (GNSS) Industry. *Regional studies*, 46 (6), 741–756.

Balland, P.A., Boschma, R., and Frenken, K., 2015. Proximity and Innovation: From Statics to Dynamics. *Regional studies*, 49 (6), 907–920.

Balland, P.A., De Vaan, M., and Boschma, R., 2013. The dynamics of interfirm networks along the industry life cycle: The case of the global video game industry, 1987–2007. *Journal of Economic Geography*, 13 (5), 741–765.

Baum, J.A.C. and Rowley, T.J., 2002. Companion to Organizations: An Introduction. *In*: J.A.C. Baum, ed. *The Blackwell Companion to Organizations*. Oxford: Blackwell Publishers, Ltd, 1–34.

Boschma, R., 2005. Proximity and Innovation: A Critical Assessment. *Regional studies*, 39 (1), 61–74.

Boschma, R. and Frenken, K., 2010. The spatial evolution of innovation networks. A proximity perspective. *In*: R. Boschma and R. Martin, eds. *The Handbook of Evolutionary Economic Geography*. Cheltenham, UK: Edward Elgar, 120–135.

Breschi, S. and Lissoni, F., 2009. Mobility of skilled workers and co-invention networks: an anatomy of localized knowledge flows. *Journal of Economic Geography*, 9 (4), 439–468.

Broekel, T. and Boschma, R., 2012. Knowledge networks in the Dutch aviation industry: the proximity paradox. *Journal of Economic Geography*, 12 (2), 409–433.

Bryman, A., 2009. Mixed methods in organizational research. *In*: D. Buchanan and A. Bryman, eds. *The Sage Handbook of Organizational Research Methods*. London: Sage Publications, 516–531.

Burt, R., 1992. *Structural Holes: The Social Structure of Competition*. Cambridge, MA: Harvard University Press.

Butts, C.T., 2008. 4. A Relational Event Framework for Social Action. *Sociological methodology*, 38 (1), 155–200.

Cantner, U. and Graf, H., 2006. The network of innovators in Jena: An application of social network analysis. *Research policy*, 35 (4), 463–480.

Cassi, L. and Plunket, A., 2014. Proximity, network formation and inventive performance: in search of the proximity paradox. *The Annals of regional science*, 53 (2), 395–422.

Cassi, L. and Plunket, A., 2015. Research Collaboration in Co-inventor Networks: Combining Closure, Bridging and Proximities. *Regional studies*, 49 (6), 936–954.

Checkley, M. and Steglich, C., 2007. Partners in power: job mobility and dynamic deal-making. *European Management Review*, 4 (3), 161–171.

Chen, K.K. and O'Mahony, S., 2009. Differentiating Organizational Boundaries. *In*: B.G. King, T. Felin, and D.A. Whetten, eds. *Studying Differences between Organizations: Comparative Approaches to Organizational Research*. Emerald Group Publishing Limited, 183–220.

Cherryholmes, C.H., 1992. Notes on Pragmatism and Scientific Realism. *Educational researcher* , 21 (6), 13–17.

Cohen, M.D., March, J.G., and Olsen, J.P., 1972. A Garbage Can Model of Organizational Choice. *Administrative science quarterly*, 17 (1), 1–25.

Colazo, J.A., 2010. Collaboration Structure And Performance In New Software Development: Findings From The Study Of Open Source Projects. *International Journal of Innovation Management*, 14 (05), 735–758.

Conaldi, G. and Lomi, A., 2013. The dual network structure of organizational problem solving: A case study on open source software development. *Social networks*, 35 (2), 237–250.

Corbet, J. and Kroah-Hartman, G., 2017. *Linux Kernel Development Report*. The Linux Foundation.

Corbet, J., Kroah-Hartman, G., and McPherson, A., 2010. *Linux Kernel Development: How Fast it is Going, Who is Doing It, What They are Doing, and Who is Sponsoring It*. The Linux Foundation.

Crescenzi, R., Nathan, M., and Rodríguez-Pose, A., 2016. Do inventors talk to strangers? On proximity and collaborative knowledge creation. *Research policy*, 45 (1), 177–194.

Creswell, J.W., 2009. *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. Los Angeles: SAGE Publications.

Criscuolo, P., Salter, A., and Ter Wal, A., 2010. The role of proximity in shaping knowledge sharing in professional services firms. *In*: *Opening Up Innovation: Strategy, Organization and Technology*. Presented at the DRUID10.

Cross, R., Borgatti, S.P., and Parker, A., 2002. Making invisible work visible: Using social network analysis to support strategic collaboration. *California management review*, 44 (2), 25–46.

Crowston, K. and Howison, J., 2006. Hierarchy and centralization in free and open source software team communications. *Knowledge, Technology & Policy*, 18 (4), 65–85.

Crowston, K., Wei, K., Howison, J., and Wiggins, A., 2012. Free/Libre open-source software development: What we know and what we do not know. *ACM Computing Surveys (CSUR)*, 44 (2), 7.

Dahlander, L. and O'Mahony, S., 2010. Progressing to the Center: Coordinating Project Work. *Organization Science*, 22 (4), 961–979.

Dawson, J.F., 2014. Moderation in Management Research: What, Why, When, and How. *Journal of business and psychology*, 29 (1), 1–19.

DeSanctis, G. and Monge, P., 1999. Introduction to the Special Issue: Communication Processes for Virtual Organizations. *Organization Science*, 10 (6), 693–703.

De Waal, C., 2005. *On Pragmatism*. Belmont, NJ: Thomson Wadsworth.

Dobusch, L. and Schoeneborn, D., 2015. Fluidity, Identity, and Organizationality: The Communicative Constitution of Anonymous. *Journal of Management Studies*, 52 (8), 1005–1035.

Dougherty, C., 2011. *Introduction to Econometrics*. Oxford University Press.

Ferraro, F. and O'Mahony, S., 2012. Managing the Boundaries of an Open Project. *In*: J.F. Padgett and W.W. Powell, eds. *The emergence of organizations and markets*. Princeton, NJ: Princeton University Press, 545–565.

Gibbs, G., 2007. *Analyzing Qualitative Data*. Los Angeles: SAGE Publications.

Gilsing, V., Nooteboom, B., Vanhaverbeke, W., Duysters, G., and van den Oord, A., 2008. Network embeddedness and the exploration of novel technologies: Technological distance, betweenness centrality and density. *Research policy*, 37 (10), 1717–1731.

Glance, N.S. and Huberman, B.A., 1994. Social dilemmas and fluid organizations. *In*: K.M. Carley and M.J. Prietula, eds. *Computational Organization Theory*. Hillsdale, NJ: L. Erlbaum Associates, 217–239.

Grand, S., von Krogh, G., Leonard, D., and Swap, W., 2004. Resource allocation beyond firm boundaries: A multi-level model for Open Source innovation. *Long range planning*, 37 (6), 591–610.

Granovetter, M., 1985. Economic Action and Social Structure: The Problem of Embeddedness. *The American journal of sociology*, 91 (3), 481–510.

Greene, W.H., 2012. *Econometric Analysis*. Seventh. Upper Saddle River, NJ: Prentice Hall.

Gulati, R. and Gargiulo, M., 1999. Where Do Interorganizational Networks Come From? *The American journal of sociology*, 104 (5), 1439–1493.

Gulati, R., Puranam, P., and Tushman, M., 2012. Meta-organization design: Rethinking design in interorganizational and community contexts. *Strategic Management Journal*, 33 (6), 571–586.

Hanneke, S., Fu, W., and Xing, E.P., 2010. Discrete temporal models of social networks. *Electronic journal of statistics*, 4, 585–605.

Hansen, T., 2015. Substitution or Overlap? The Relations between Geographical and Non-spatial Proximity Dimensions in Collaborative Innovation Projects. *Regional studies*, 49 (10), 1672–1684.

Hardeman, S., Frenken, K., Nomaler, Ö., and Ter Wal, A.L.J., 2015. Characterizing and comparing innovation systems by different 'modes' of knowledge production: A proximity approach. *Science & public policy*, 42 (4), 530–548.

Hars, A. and Ou, S., 2002. Working for free? Motivations of participating in open source projects. *International Journal of Electronic Commerce*, 6 (3), 25–39.

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

Heringa, P.W., Hessels, L.K., and van der Zouwen, M., 2016. The influence of proximity dimensions on international research collaboration: an analysis of European water projects. *Industry and Innovation*, 23 (8), 753–772.

Hertel, G., Niedner, S., and Herrmann, S., 2003. Motivation of software developers in Open Source projects: an Internet-based survey of contributors to the Linux kernel. *Research policy*, 32 (7), 1159–1177.

Huber, F., 2012. On the Role and Interrelationship of Spatial, Social and Cognitive Proximity: Personal Knowledge Relationships of R&D Workers in the Cambridge Information Technology Cluster. *Regional studies*, 46 (9), 1169–1182.

Iivari, N., 2011. Participatory design in OSS development: interpretive case studies in company and community OSS development contexts. *Behaviour & information technology*, 30 (3), 309–323.

Jaccard, J. and Turrisi, R., 2003. *Interaction Effects in Multiple Regression*. Sage.

Jensen, C. and Scacchi, W., 2007. Role Migration and Advancement Processes in OSSD Projects: A Comparative Case Study. *In*: *29th International Conference on Software Engineering (ICSE'07)*. 364–374.

Kernel development community, 2017a. Submitting patches: the essential guide to getting your code into the kernel [online]. Available from: https://www.kernel.org/doc/html/v4.14/process/submitting-patches.html [Accessed 11 Jan 2018].

Kernel development community, 2017b. A guide to the Kernel Development Process: How the development process works [online]. Available from: https://www.kernel.org/doc/html/v4.14/process/2.Process.html [Accessed 6 Feb 2018].

Kilduff, M. and Tsai, W., 2003. *Social Networks and Organizations*. SAGE.

Kleinbaum, D.G., 1994. *Logistic regression: A self-learning text*. New York: Springer-Verlag.

Knoben, J. and Oerlemans, L.A.G., 2006. Proximity and inter-organizational collaboration: A literature review. *International Journal of Management Reviews*, 8 (2), 71–89.

Korpi, K., 2010. Google's Failure to Contribute to Linux Kernel Due to Secrecy and Weird Design [online]. *The Next Web*. Available from: http://thenextweb.com/asia/2010/03/05/googles-failure-contribute-linux-kernel-due-secrecy-weird-design/ [Accessed 1 Sep 2016].

Krackhardt, D., 1999. The Ties that Torture: Simmelian Tie Analysis in Organizations. *Research in the Sociology of Organizations*, 16, 183–210.

Kuzel, A.J., 1999. Sampling in qualitative inquiry. *In*: B.F. Crabtree and W.L. Miller, eds. *Doing qualitative research*. Thousand Oaks: Sage Publications, 33–45.

Kvale, S. and Brinkmann, S., 2009. *InterViews: Learning the Craft of Qualitative Research Interviewing*. Los Angeles: Sage Publications.

Lakhani, K.R. and Wolf, R.G., 2005. Why hackers do what they do: Understanding motivation and effort in free/open source software projects. *In*: J. Feller, B. Fitzgerald, S.A. Hissam, and K.R. Lakhani, eds. *Perspectives on free and open source software*. Cambridge, MA: MIT Press, 3–22.

Lee, G.K. and Cole, R.E., 2003. From a firm-based to a community-based model of knowledge creation: The case of the Linux kernel development. *Organization Science*, 14 (6), 633–649.

Lerner, J. and Lomi, A., 2017. The Third Man: hierarchy formation in Wikipedia. *Applied Network Science*, 2 (1), 24.

Lincoln, Y.S. and Guba, E.G., 1985. *Naturalistic Inquiry*. Beverly Hills: Sage Publications.

Linux Foundation, 2016a. Video Keynote: In conversation with Linux Creator Linus Torvalds and Dirk Hohndel [online]. Available from: https://www.youtube.com/watch?v=_QlFfmHIbLg [Accessed 21 Sep 2016].

Linux Foundation, 2016b. The Linux Foundation Trademarks and Trademark Usage Guidelines [online]. Available from: https://www.linuxfoundation.org/trademark-usage [Accessed 1 Sep 2016].

Linux Kernel Organization, 2017. List of maintainers and how to submit kernel changes [online]. *Linux Kernel Archives*. Available from: https://www.kernel.org/doc/linux/MAINTAINERS [Accessed 11 Jul 2017].

López-Fernández, L., Robles, G., Gonzalez-Barahona, J.M., and Herraiz, I., 2006. Applying Social Network Analysis Techniques to Community-Driven Libre Software Projects. *International Journal of Information Technology and Web Engineering*, 1 (3), 27–48.

Mackenzie, N. and Knipe, S., 2006. Research dilemmas: Paradigms, methods and methodology. *Issues in Educational Research*, 16 (2), 193–205.

MacKinnon, D.P., Coxe, S., and Baraldi, A.N., 2012. Guidelines for the Investigation of Mediating Variables in Business Research. *Journal of business and psychology*, 27 (1), 1–14.

Madey, G., Freeh, V., and Tynan, R., 2002. The open source software development phenomenon: An analysis based on social network theory. *In*: *8th Americas Conference on Information Systems*. 1806–1813.

March, J.G. and Simon, H.A., 1993. *Organizations*. 2nd ed. Malden, MA: Blackwell.

Maxcy, S.J., 2003. Pragmatic threads in mixed methods research in the social sciences: The search for multiple modes of inquiry and the end of the philosophy of formalism. *In*: A. Tashakkori and C. Teddlie, eds. *Handbook of mixed methods in social and behavioral research*. Thousand Oaks: Sage Publications, 51–89.

Miles, M.B., Michael Huberman, A., and Saldana, J., 2013. *Qualitative Data Analysis: A Methods Sourcebook*. 3rd ed. Thousand Oaks: SAGE Publications.

Mockus, A., Fielding, R.T., and Herbsleb, J.D., 2002. Two case studies of open source software development: Apache and Mozilla. *ACM Transactions on Software Engineering and Methodology*, 11 (3), 309–346.

Nooteboom, B., 1999. *Inter-firm Alliances: Analysis and Design*. London: Routledge.

Nooteboom, B., 2000. *Learning and Innovation in Organizations and Economies*. Oxford: Oxford University Press.

Nooteboom, B., 2009. *A Cognitive Theory of the Firm: Learning, Governance and Dynamic Capabilities*. Cheltenham: Edward Elgar Publishing.

Nooteboom, B., Van Haverbeke, W., Duysters, G., Gilsing, V., and van den Oord, A., 2007. Optimal cognitive distance and absorptive capacity. *Research policy*, 36 (7), 1016–1034.

Nurmi, N. and Hinds, P.J., 2016. Job complexity and learning opportunities: A silver lining in the design of global virtual work. *Journal of International Business Studies*, 47 (6), 631–654.

O'Leary, M.B. and Cummings, J.N., 2007. The spatial, temporal, and configurational characteristics of geographic dispersion in teams. *MIS Quarterly*, 31 (3), 433–452.

O'Mahony, S., 2003. Guarding the commons: how community managed software projects protect their work. *Research policy*, 32 (7), 1179–1198.

O'Mahony, S., 2007. The governance of open source initiatives: what does it mean to be community managed? *Journal of Management & Governance*, 11 (2), 139–150.

O'Mahony, S. and Bechky, B.A., 2008. Boundary Organizations: Enabling Collaboration among Unexpected Allies. *Administrative science quarterly*, 53 (3), 422–459.

Patton, M.Q., 2002. *Qualitative research and evaluation methods*. 3rd ed. Thousand Oaks: Sage Publications.

Podolny, J.M. and Page, K.L., 1998. Network Forms of Organization. *Annual review of sociology*, 24 (1), 57–76.

Ponds, R., Van Oort, F., and Frenken, K., 2007. The geographical and institutional proximity of research collaboration*. *Papers in regional science: the journal of the Regional Science Association International*, 86 (3), 423–443.

Powell, W.W., 1990. Neither Market nor Hierarchy: Network forms of organization. *Research in Organizational Behavior*, 12, 295–336.

Puranam, P., Alexy, O., and Reitzig, M., 2014. What's 'New' About New Forms of Organizing? *Academy of Management Review*, 39 (2), 162–180.

Quintane, E., Conaldi, G., Tonellato, M., and Lomi, A., 2014. Modeling Relational Events: A Case Study on an Open Source Software Project. *Organizational Research Methods*, 17 (1), 23–50.

Quintane, E., Pattison, P.E., Robins, G.L., and Mol, J.M., 2013. Short- and long-term stability in organizational networks: Temporal structures of project teams. *Social networks*, 35 (4), 528–540.

Rank, O.N., Robins, G.L., and Pattison, P.E., 2009. Structural Logic of Intraorganizational Networks. *Organization Science*, 21 (3), 745–764.

Roberts, J.A., Hann, I.-H., and Slaughter, S.A., 2006. Understanding the Motivations, Participation, and Performance of Open Source Software Developers: A Longitudinal Study of the Apache Projects. *Management science*, 52 (7), 984–999.

Robins, G., Pattison, P., and Wang, P., 2009. Closure, connectivity and degree distributions: Exponential random graph (p*) models for directed social networks. *Social networks*, 31 (2), 105–117.

Robles, G., González-Barahona, J.M., Izquierdo-Cortazar, D., and Erlandson, B.E., 2009. Tools for the Study of the Usual Data Sources found in Libre Software Projects. *International Journal of Open Source Software & Processes*, 1 (1), 24–45.

Rossman, G.B. and Wilson, B.L., 1985. Numbers And Words - Combining Quantitative And Qualitative Methods In A Single Large-Scale Evaluation Study. *Evaluation review*, 9 (5), 627–643.

Schneider, D., Spurlock, S., and Squire, M., 2016. Differentiating Communication Styles of Leaders on the Linux Kernel Mailing List. *In*: *Proceedings of the 12th International Symposium on Open Collaboration*. ACM, 2.

Shah, S.K., 2006. Motivation, Governance, and the Viability of Hybrid Forms in Open Source Software Development. *Management science*, 52 (7), 1000–1014.

Simmel, G., 1950. *The Sociology of Georg Simmel*. New York: The Free Press of Glencoe.

Singh, J., 2005. Collaborative Networks as Determinants of Knowledge Diffusion Patterns. *Management science*, 51 (5), 756–770.

Snijders, T.A.B., 1996. Stochastic actor-oriented models for network change. *Journal of Mathematical Sociology*, 21, 149–172.

Snow, C.C., Miles, R.E., and Coleman, H., 1992. Managing 21st Century Network Organizations. *Organizational dynamics*, 20 (3), 5–20.

Sorenson, O., Rivkin, J.W., and Fleming, L., 2006. Complexity, networks and knowledge flow. *Research policy*, 35 (7), 994–1017.

Sorenson, O. and Stuart, T.E., 2001. Syndication Networks and the Spatial Distribution of Venture Capital Investments. *American Journal of Sociology*, 106 (6), 1546–88.

Stake, R.E., 1995. *The Art of Case Study Research*. Thousand Oaks: Sage Publications.

Ter Wal, A.L.J., 2013. Cluster Emergence and Network Evolution: A Longitudinal Analysis of the Inventor Network in Sophia-Antipolis. *Regional studies*, 47 (5), 651–668.

Ter Wal, A.L.J., 2014. The dynamics of the inventor network in German biotechnology: geographic proximity versus triadic closure. *Journal of Economic Geography*, 14 (3), 589–620.

Torre, A., 2008. On the Role Played by Temporary Geographical Proximity in Knowledge Transmission. *Regional studies*, 42 (6), 869–889.

Tushman, M.L., 1977. Special Boundary Roles in the Innovation Process. *Administrative Science Quarterly*, 22 (4), 587–605.

Uzzi, B., 1997. Social Structure and Competition in Interfirm Networks: The Paradox of Embeddedness. *Administrative science quarterly*, 42 (1), 35–67.

vger.kernel.org, 2016. Majordomo lists at VGER.KERNEL.ORG [online]. Available from: http://vger.kernel.org/vger-lists.html#linux-pci [Accessed 3 Mar 2016].

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

von Krogh, G., Haefliger, S., Spaeth, S., and Wallin, M.W., 2012. Carrots and rainbows: Motivation and social practice in open source software development. *MIS Quarterly*, 36 (2), 649–676.

von Krogh, G., Spaeth, S., and Lakhani, K.R., 2003. Community, joining, and specialization in open source software innovation: a case study. *Research policy*, 32 (7), 1217–1241.

Wasserman, S. and Faust, K., 1994. *Social Network Analysis: Methods and Applications*. Cambridge: Cambridge University Press.

West, J. and O'Mahony, S., 2008. The Role of Participation Architecture in Growing Sponsored Open Source Communities. *Industry and Innovation*, 15 (2), 145–168.

White, H.C., 1981. Where Do Markets Come From? *The American journal of sociology*, 87 (3), 517–547.

Yin, R.K., 2009. *Case Study Research: Design and Methods*. Los Angeles: Sage Publications.

Zammuto, R.F., Griffith, T.L., Majchrzak, A., Dougherty, D.J., and Faraj, S., 2007. Information Technology and the Changing Fabric of Organization. *Organization Science*, 18 (5), 749–762.

APPENDIX A: PHASE 1 INTERVIEW GUIDES

Interview Guide - Pilot Study

Semi-Structured Interview Guide for Pilot Study
Company Contributions in Open Source Software Communities:
Collaboration, Competition and Productivity
Dawn M. Foster

Research Questions and Focus Areas

Primary research question: How do software developers, who are paid by organizations for their work, collaborate within an open source software community?

- Sub question: How does the competition between these organizations influence individual software developers' contributions?
- Sub question: How productive are paid software developers?

Interview Topics (Focus the questions to get responses that achieve these goals):

- Gain a better understanding of how paid software developers participate and the role that their employer plays in their work, especially with respect to collaboration, competition and productivity.
- Learn more about the interactions of paid software developers with a focus on the dynamic between collaboration and competition. Do they interact with or collaborate with competitors? Do they interact differently with their competitors as compared to other participants who do not work for competing companies?
- Begin to understand the productivity of kernel developers especially as it relates to paid vs. unpaid developers and productivity associated with collaborating with the software developers working for competing organizations. Note: this is a lower priority goal than the first two and might be sacrificed for the sake of time.

Additional goals for pilot study only:

- Get feedback on the interview process to better understand what worked well, what didn't, and what could be improved before we conduct the main body of the interviews.
- Validate some early thinking about definitions of collaboration to see if it resonates with kernel developers.

Reminders for throughout the interview process:

- Good qualitative questions should be open-ended, neutral, singular and clear.
- Provide reinforcement / encouragement for concise, but in-depth answers (verbal and non-verbal).
- Thank them periodically throughout the interview.
- Active listening is important: listen not just to what is said, but how it's said and follow-up appropriately to achieve above goals.
- Maintain control and keep the interview on track: non-verbal clues if they are off-track, interrupt nicely if needed.

Briefing and Setting the Stage

Deliver basic information about the interview, make a good impression and put the subject at ease.

___ Confirm that they have received the participant information sheet and signed the consent form. As a reminder, your information will be anonymized and confidential.
___ Talk about how interview is being recorded and let them know that they can stop the recording at any time.
___ Mention that I will also be taking a few notes to go along with the recording.
___ Ask if they have any questions for me about the process.
___ Transition into the first question

Introduction

___ Put the subject at ease with an introductory question.
- Q: Please tell me about how you first got involved in Linux kernel development.

Paid Software Development

___ Employment situation – current / past (may have been covered in intro question)
- Q: Would you tell me about the first time you were paid to do kernel development?
- Q: Would you tell me more about your role at …?
- Q: How many hours per week would you say that you spend working on the Linux kernel?

___ Reasons for employer to pay kernel developers
- Q: What would you say is the primary reason that your current employer pays you to do this work?
    - Q: What are some of the other reasons?
    - Q: What are some of the other benefits to the company?

___ Company involvement in day-to-day work
- Q: How does your current (or most recent) employer get involved in providing direction for your Linux work?
    - If yes, Q: How much of the work is at your own discretion vs. at your employer's request?
    - If yes, Q: To what extent does this vary based on the type of work you are doing?
    - If no, Q: Tell me more about how this works?
    - If no, Q: They pay you to work on the Linux kernel. Do you have an agreement or understanding with them on what type of work you should be doing? Maybe you can tell me a little more about this agreement / understanding?

___ Differences between paid and unpaid developers (collaboration & productivity)
- Q: What are some of the differences between people within the kernel community who are paid to do their work versus people who contribute on a purely voluntary basis?
    - Q: Does one group tend to be more productive than the other?
        - What would you say makes a kernel developer productive? OR How do you define productivity in the case of Linux kernel developers? Note: make sure that I get their definition of productivity.

Interactions: Collaboration and Competition

___ General interactions and collaboration

- Q: Please tell me more about how you interact with other people within the kernel community in your day-to-day kernel work?
    - Q: It seems like sometimes it might be difficult to accomplish what your employer asks you to do. If it is, how does this impact your interactions with other developers?
    - Q: Are there areas or subsystems within the kernel where you tend to interact with more people? Or areas where you tend to work alone more of the time?
    - Q: Who do you interact with most closely (look for names of individuals and companies)?
    - Notes: Make sure that they defined how they interact. Probe into the areas listed in the Appendix if they do not spontaneously come up in their answer.

___ Which competitors

- Q: Which of your employer's competitors also work on the kernel?
    - Look for specific names.

___ Competition interactions – differences from interactions with non-competitors

- Q: How do you interact with employees from competing companies?
    - Q: How is this different from how you interact with other people who don't work for your competitors?
    - Q: Would you call this a collaborative relationship? If so, why?
    - Q: Do you think you are more or less productive when you are interacting with competitors versus other contributors? Or is it the same?
        - Earlier, you defined productivity as …, how would your company define productivity?
    - Q: Are there any competitors that you interact with more often (look for names of individuals and companies)?
    - Notes: Make sure that they define how they interact with employees of competitors. Probe into the areas listed in the Appendix if they do not spontaneously come up in their answer.

___ Employer guidelines for competitor interactions

- Q: What sort of guidelines or rules does your employer have that specify how you are or are not allowed to interact with employees from competing companies?
- Q: How do you balance what you know about your company's confidential, proprietary data with your daily open source work on the Linux kernel?
    - Would you describe the tension that exists between what you know, but can't discuss with your open source participation in the kernel?

Debriefing and Wrap-up

___ Final insights

As a reminder, the overall goal of this research is to learn more about collaboration, competition and productivity of kernel developers who are paid by organizations,
Q: Would you like to add anything else?
Q: What should I have asked you that I didn't think to ask about?

___ Thank them for taking the time and providing insights into [mention a couple of things I learned].

___ A few reminders:

- Your answers are confidential. The interviews will be collected together and the anonymized analysis will be published as part of my PhD dissertation in a few years,

but it is also possible that some of it will be published online, in journal articles or as part of a conference presentation over the next 3 years. You can also request to see the transcript from your interview along with any other research outcomes.
- I also wanted to remind you now that we've finished the interview that you can change your mind any time before October 30th, and I'll delete your responses if you decide that you don't want to participate.

Pilot Study Feedback

___ Get feedback on interview (improvements, strengths, advice)
- Since this is a pilot study before I do more interviews at LinuxCon Europe and LinuxCon North America, I wanted to get a little feedback about the interview process.
- Q: If you were me, what would you change or improve about this interview?
- Q: What worked well and shouldn't be changed?
- Q: Is there any additional advice you would like to offer me before I do more interviews with kernel developers?

___ Validate early thinking about how to define collaboration
- For a follow up study, we are also looking at ways to quantify some of the interactions and collaboration between kernel developers, and I wanted to run a few ideas by you to see if they make sense.
- Review collaboration methods in appendix and talk about whether each one is important / relevant and how it might be best measured.

End Interview

___ A final thank you.

Reflection

Set aside 10-20 min after each interview to reflect on what was learned.
___Document anything that might not have come through via voice
- facial expressions, excitement, body language, setting, mood, voice, etc.
- This can provide valuable context for later analysis of transcripts.

___ Re-read and clean up notes.
- Add additional information for anything that is a bit light and might not make sense later.

Appendix: Interaction and Collaboration Types and Definitions

Note: part of the pilot will include validating what kernel developers think of these methods of collaboration. The order is currently based on my best guess, and it will likely be re-ordered after the pilot.

Collaboration: Roughly ordered in order of relevance to this research
- Code review / test as designated by the addition of Acked-by:, Tested-by:, or Reviewed-by lines.
- Provide feedback on a person's patch (usually via a mailing list, but could be feedback offered in other ways).
- Working on the same file or subsystem – note: this may or may not be considered collaboration; I want to make sure to ask pilot participants what they think.

- Providing feedback or comments on a bug (via Bugzilla or mailing list).
- Mailing list discussions of a general nature.
- Real-time discussions and other collaboration in person at events (LinuxCon, Kernel Summit, etc.), video in hangouts / Skype, audio over the phone, or online text chat via IRC / IM.
- Other / What did I miss?

Interactions: other non-collaborative interactions that are not relevant to this research

- Watching / keeping track of contributions or communications from another person / company.
- Socializing with other developers when conversation doesn't include discussions about Linux kernel contributions.
- Watching videos and presentations or reading documents about the Linux kernel.

Interview Guide - Full Phase 1 Study

Semi-Structured Interview Guide
Understanding Collaboration in Fluid Organizations, a Proximity Approach
Dawn M. Foster

Research Questions and Focus Areas

Primary research question: "How do participants who are paid by firms collaborate within a fluid organization?"

- Sub question: "How do participants collaborate with people who work for competing firms vs. other participants, and what is the role of proximity in these collaborations?"
- Sub question: "What is the role of the employer in participation within a fluid organization?"

Interview Topics (Focus the questions to get responses that achieve these goals):

- Gain a better understanding of how paid software developers participate and the role that their employer plays in their work, especially with respect to collaboration.
- Learn more about the interactions of paid software developers with a focus on the dynamic between collaboration and competition. Do they interact with or collaborate with competitors? Do they interact differently with their competitors as compared to other participants who do not work for competing companies?
- Gain a better understanding of the role that the major dimensions of proximity have on collaboration.

Briefing and Setting the Stage

Deliver basic information about the interview, make a good impression and put the subject at ease.

___ Introduce myself
___ Confirm that they have received the participant information sheet and signed the consent form. As a reminder, your information will be anonymized and confidential.
___ Talk about how interview is being recorded and let them know that they can stop the recording at any time.
___ Mention that I will also be taking a few notes to go along with the recording.
___ Ask if they have any questions for me about the process.
___ Transition into the first question

Introduction

___ Put the subject at ease with an introductory question.

- Q: Please tell me about how you first got involved in Linux kernel development.

Paid Software Development

___ Employment situation – current / past (may have been covered in intro question)

- Q: Would you tell me about the first time you were paid to do kernel development?
- Q: Would you tell me more about your role at …?

- Q: In the past 30 days, how many hours per week would you say that you spent working on the Linux kernel?
- Q: Over the past 90 days, how many hours per week would you say that you spent working on the Linux kernel?

___ Reasons for employer to pay kernel developers

- Q: What would you say is the primary reason that your current employer pays you to do this work?
  Probes:
  - Q: What are some of the other reasons?
  - Q: What are some of the other benefits to the company?
  - Q: Would you talk a little about how your company recruits kernel developers?

___ Company involvement in day-to-day work

- Q: How does your current (or most recent) employer get involved in providing direction for your Linux work?
  Probes:
  - If yes, Q: How much of the work is at your own discretion vs. at your employer's request?
  - If yes, Q: To what extent does this vary based on the type of work you are doing?
  - If no, Q: They pay you to work on the Linux kernel. Do you have an agreement or understanding with them on what type of work you should be doing? Maybe you can tell me a little more about this agreement / understanding?
  - Q: Tell me more about how this works?
- Q: Has your employer's involvement in your work been consistent or has it changed over time?
  - Q: How has it changed over time?

___ Institutional Proximity

- Q: Does whether a person works for a company, non-profit, university or is an unpaid contributor affect how you interact with them in the Linux kernel?
  Probes:
  - Q: What do you typically know about the affiliation of other participants?
  - Look for examples - maybe ask them to think about the developers they interact with, or some of them, and then to think about the affiliations of those individuals.
  - Need to understand how much they really know about the institutional affiliation for the people they interact with – do they typically know?

Interactions: Collaboration and Competition

___ General interactions and collaboration

- Q: Please tell me more about how you interact with other people within the kernel community in your day-to-day kernel work?
  Probes:
  - Q: It seems like sometimes it might be difficult to accomplish what your employer asks you to do. If it is, how does this impact your interactions with other developers?
  - Q: Are there areas or subsystems within the kernel where you tend to interact with more people? Or areas where you tend to work alone more of the time?
    Probes:
    - Q: Why do you interact with more people in certain areas?
    - Look for reasons.

- Q: Who do you interact with most closely (look for names of individuals and companies)?
- Q: What is the role of IRC within the kernel community?
    - Need to confirm whether it is less important than mailing lists and better understand how / how widely it is used.
- Notes: Make sure that they defined how they interact. Probe into the areas listed in the Appendix if they do not spontaneously come up in their answer.

___ Social Proximity

- Q: Can you tell me about the relationships that you develop with other kernel developers?

  Probes:
    - Q: How do those relationships develop?
    - Q: Which online channels do you use to interact with those people in the context of the Linux kernel community?
    - Q: Do you develop friendships with other kernel developers, or are these strictly professional relationships?

      Probes:
        - Q: How do the friendships differ from the professional relationships?
        - Look for examples of actions that are proxy for friendships
- Q: What role does trust play in these relationships?
    - Make sure that they specify their definition and give examples of actions that are proxy for trust.
- Notes: Make sure that they define how they interact. Probe into the areas listed in the Appendix if they do not spontaneously come up in their answer.

___ Organizational Proximity (competition / company interactions)

- Q: How do you identify with the kernel community as a whole? Is this the same or different from how you identify with your employer?

  Probes:
    - Q: Which is most important – your affiliation with your employer or with the kernel?
    - What are the similarities and differences between how you collaborate with people who work for your employer vs. those who don't?
    - Q: What are the similarities and differences between how you work with people who are employed by competitors vs. those who are not?

  Probes:
    - Q: How is this different from how you interact with other people who don't work for your competitors?
    - Q: Would you call this a collaborative relationship? If so, why?
    - Notes: Make sure that they define how they interact with employees of competitors. Probe into the areas listed in the Appendix if they do not spontaneously come up in their answer.

___ Which competitors

- Q: Which of your employer's competitors also work on Linux kernel development?
    - Look for specific names.
- Q: Are there any competitors that you interact with more often (look for names of individuals and companies)?

___ Cognitive Proximity

- Q: For kernel developers that you collaborate with, can you talk about whether or not they tend to have similar backgrounds or similar knowledge to yours?

Probes:

- Q: Is it important for people to have similar knowledge and backgrounds?
- Q: Is there a downside to having frames of reference or knowledge that are too similar?

___ Geographic / Time Zone proximity

- Q: What role does physical location have on your interactions with other people in the Linux kernel?
  Probes:
  - Q: Do you work more closely with any people who are located near you?
- Q: What about people who are in similar time zones?
  Probes
  - Q: Do you tend to work more closely with people who are online at similar times as you?
- May need to probe on whether they are talking about people in similar time zones vs. physical proximity. Does online presence specifically make someone proximate to the interviewee?

Debriefing and Wrap-up

___ Final insights

As a reminder, the overall goal of this research is to learn more about collaboration, and competition of kernel developers who are paid by organizations,

- Q: Would you like to add anything else?
- Q: What should I have asked you that I didn't think to ask about?

___ Thank them for taking the time and providing insights into [mention a couple of things I learned].

___ A few reminders:

- Your answers are confidential. The interviews will be collected together and the anonymized analysis will be published as part of my PhD dissertation in a few years, but it is also possible that some of it will be published online, in journal articles or as part of a conference presentation over the next 3 years. You can also request to see the transcript from your interview along with any other research outcomes.
- I also wanted to remind you now that we've finished the interview that you can change your mind any time before November 30th, and I'll delete your responses if you decide that you don't want to participate.

End Interview

___ A final thank you.

Reflection

Set aside 10-20 min after each interview to reflect on what was learned.

___Document anything that might not have come through via voice

- facial expressions, excitement, body language, setting, mood, voice, etc.
- This can provide valuable context for later analysis of transcripts.

___ Re-read and clean up notes.

- Add additional information for anything that is a bit light and might not make sense later.

Appendix: Interaction and Collaboration Types and Definitions

Collaboration:

- Mailing list discussions of a general nature.
- Provide feedback on a person's patch (usually via a mailing list, but could be feedback offered in other ways).
- Providing feedback or comments on a bug.
- Working on the same file or subsystem
- Code review / test as designated by the addition of Acked-by:, Tested-by:, or Reviewed-by lines.
- Real-time discussions and other collaboration in person at events (LinuxCon, Kernel Summit, etc.), video in hangouts / Skype, audio over the phone, or online text chat via IRC / IM.

Interactions: other non-collaborative interactions **not** relevant to this research

- Watching / keeping track of contributions or communications from another person / company.
- Socializing with other developers when conversation doesn't include discussions about Linux kernel contributions.
- Watching videos and presentations or reading documents about the Linux kernel.

APPENDIX B: PHASE 1 QUALITATIVE CODES

Codes Used - Pilot Study

Changing Jobs		
	Recruiting kernel devs	
Company Impressions within community		
	Negative Impressions	
	Positive Impressions	
Roles and Types of Work		
	Architectural / High-level	
	Project / Community Mgmt	
	Justify reasoning after the fact	
	Fixing Bugs	
	Meetings	
	Hobby contributions	
	Time - hours spent contributing	
	Product work	
	Lack of clarity around role	
	Porting to new hardware	
	Driver work	
	Maintainer	
	Submitting Patches and Upstream Dev	
	Providing Advice and Training	
	Managing people	
	Getting Started with Kernel	
		Hobby start
		Paid start
Paid Development Company Reasons		
	Testing / maintenance	
	Visibility / Marketing	
	Platform / Product enabling	
	Legitimization / Credibility	
	Giving back to community	
	Information back to company	
	Advising others	
	Influence	
	Prestige	
Direction from Company to Developer		
	Trust their judgment	

	Pulling back - re-guide	
	Encourage existing interests	
	Little direction	
	Work on something specific	
	Take an interest in a project	
Paid vs. Unpaid Dynamics		
	Balancing corp vs. community interests	
	Give unpaid devs more leeway	
	Longer-term commitment / Lack of	
	Unpaid more productive	
	Paid more productive	
	Hard to distinguish	
	Most people are paid	
	Enthusiasm / Lack of	
	Quality / Lack of	
	Responsiveness / Lack of	
	Differences	
	Similarities	
Productivity		
	Non-measurement inputs	
	Difficulties in measuring	
	Output	
	Time component	
Collaboration		
	Private Collaboration	
	Kernel Bugzilla	
	ack-ed by, tested-by, reviewed-by lines	
	Bug collab	
	Lack of collaboration	
	with specific people	
	sharing info / Q&A	
	Existing relationships	
	Challenges	
	In-Person Collab	
	Code Collab	
		Same file or subsystem
	IRC Collab	
	Mailing List Collab	
	First Mention of Collaboration	
Competition		

	Confidential Info	
		Company processes and policies
		Share without being obvious
		Be careful about sharing
		Don't have much of it
	Collaboration with competitors	
	Encouragement of competitors	
	Treat them as individuals	
	Specific competitors	
Pilot Feedback		

Codes Used - Full Phase 1 Study

Proximity		
	Social	
		Work Together / Alone
		Trust
		Professional
		Friendship
	Institutional	
		don't care about affiliation
		corporate email
		impact on work
		known vs unknown
	Organizational	
		cultural differences
		other employees
		both employer and kernel
		employer first
		kernel first
	Cognitive	
		experience
		importance
		knowledge - similar/diff
	Geographic	
		temporary geo
		physical location
		Language
		Time Zone
Changing Jobs		
	Recruiting kernel devs	
Company Impressions within community		
	Negative Impressions	
	Positive Impressions	
Roles and Types of Work		
	Specific Subsystems / Areas	
	Architectural / High-level / New Features	
	Project / Community Mgmt	
	Justify reasoning after the fact	
	Fixing Bugs	
	Meetings	

	Hobby contributions	
	Time - hours spent contributing	
	Product work	
	Lack of clarity around role	
	Porting to new hardware	
	Driver work	
	Maintainer	
	Submitting Patches and Upstream Dev	
	Providing Advice and Training	
	Managing people	
	Getting Started with Kernel	
		Hobby start
		Paid start
Paid Development Company Reasons		
	Easier to Hire	
	Faster / More Efficient	
	Testing / maintenance	
	Visibility / Marketing	
	Platform / Product enabling	
	Legitimization / Credibility	
	Giving back to community	
	Information back to company	
	Advising others	
	Influence	
	Prestige / Reputation	
Direction from Company to Developer		
	too technical for mgmt to provide direction	
	How much direction	
	Trust their judgment	
	Pulling back - re-guide	
	Encourage existing interests	
	Little direction	
	Work on something specific	
	Take an interest in a project	
Paid vs. Unpaid Dynamics		
	Balancing corp vs. community interests	
	Give unpaid devs more leeway	
	Longer-term commitment / Lack of	
	Unpaid more productive	
	Paid more productive	

	Hard to distinguish	
	Most people are paid	
	Enthusiasm / Lack of	
	Quality / Lack of	
	Responsiveness / Lack of	
	Differences	
	Similarities	
Productivity		
	Non-measurement inputs	
	Difficulties in measuring	
	Output	
	Time component	
Collaboration		
	incorporating feedback	
	Private Collaboration	
	Kernel Bugzilla	
	ack-ed by, tested-by, reviewed-by lines	
	Bug collab	
	Lack of collaboration	
	with specific people	
	sharing info / Q&A	
	Existing relationships	
	Challenges	
	In-Person Collab	
	Code Collab	
		Same file or subsystem
	IRC Collab	
	Mailing List Collab	
	First Mention of Collaboration	
Competition		
	Confidential Info	
		Company processes and policies
		Share without being obvious
		Be careful about sharing
		Don't have much of it
	Collaboration with competitors	
	Encouragement of competitors	
	Treat them as individuals	
	Specific competitors	
Pilot Feedback		

APPENDIX C: VARIABLES

Table 10: Variable operationalization summary

Dependent Variable	Collaboration event operationalized as a reply to a message on the mailing list
Control Variables:	
Alter maintainer	1 if the alter is a maintainer, otherwise 0
Either maintainer	1 if the ego and/or the alter are maintainers, otherwise 0
Alter committer	1 if the alter has committed code within the moving window, otherwise 0
Either committer	1 if the ego and/or the alter have committed code within the moving window, otherwise 0
Ego to cc	1 if the ego was explicitly included in the “to” or “cc” field of the email that was replied to, otherwise 0
Proximity Variables	
Geographical	1 minus the normalized geographical distance calculated as the time zone offsets in seconds for a measure of Geographical proximity that ranges from 0 (maximum time zone distance) and 1 (same time zone)
Organizational	1 if both work for the same employer, otherwise 0
Institutional	1 if both work for the same type of third party organization, otherwise 0
Social	Number of times ego and alter participated in same thread within the moving window
Cognitive	Cosine similarity on contributions to areas of the source code with 0 indicating no overlap and 1 if both have contributed to exactly the same areas in the moving window
Network Variables:	
Repeated events	Number of times the ego replied to messages from the alter within the moving window
Participation shift	1 if the ego was the last person the alter replied to on the mailing list within the moving window
Recency effect	1/n with n defined as the number of people the alter emailed on the mailing list before the ego within the moving window
Transitive closure	Number of third parties that an ego has replied to where those third parties have also replied to the alter within the moving window
Cyclic closure	Number of third parties an alter has replied to where that third party has also replied to the ego within the moving window
Shared partnership inbound	Number of third parties who have replied to both the ego and the alter within the moving window
Shared partnership outbound	Number of times the ego and the alter have replied to messages by the same third party

Table 11: Variable correlations and descriptive statistics

	Variables	Median	Mean	SD	Variable Correlations 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	Alter Maintainer	0	0.093	0.290																
2	Either Maintainer	0	0.185	0.388	0.67															
3	Alter Committer	1	0.823	0.381	0.09	0.08														
4	Either Committer	1	0.970	0.170	0.03	0.05	0.38													
5	Ego To CC	0	0.179	0.383	0.00	-0.02	0.02	0.04												
6	Geographic Proximity	0.714	0.706	0.233	0.03	0.03	0.00	-0.03	-0.01											
7	Organizational Proximity	0	0.090	0.286	0.01	-0.01	0.07	-0.02	0.00	0.27										
8	Institutional Proximity	1	0.738	0.440	0.02	0.04	0.11	0.04	0.03	-0.06	0.19									
9	Social Proximity	0	4.780	18.268	-0.05	-0.09	0.10	0.03	-0.02	0.21	0.55	0.11								
10	Cognitive Proximity	0	0.131	0.237	-0.01	-0.05	0.26	0.10	0.04	0.20	0.61	0.19	0.66							
11	Repeated Effect	3	13.730	29.001	-0.08	-0.12	0.14	0.07	0.11	0.02	0.31	0.14	0.70	0.52						
12	Participation Shift	0	0.090	0.286	0.00	-0.02	0.02	0.02	0.32	0.03	0.10	0.04	0.10	0.13	0.16					
13	Recency Effect	0.025	0.167	0.291	0.00	-0.05	0.04	0.04	0.38	0.06	0.20	0.08	0.23	0.27	0.28	0.90				
14	Transitive Closure	12	18.822	19.201	-0.09	-0.14	0.20	0.10	0.08	0.02	0.27	0.16	0.62	0.52	0.83	0.14	0.29			
15	Cyclic Closure	10	16.479	18.832	-0.07	-0.13	0.20	0.10	0.11	0.05	0.30	0.15	0.66	0.55	0.75	0.15	0.31	0.92		
16	Shared Partnership In	11	19.084	22.776	-0.08	-0.14	0.18	0.09	0.08	0.05	0.36	0.19	0.67	0.61	0.80	0.15	0.31	0.96	0.95	
17	Shared Partnership Out	13	19.896	22.563	-0.07	-0.13	0.20	0.09	0.07	0.11	0.43	0.14	0.79	0.65	0.79	0.14	0.31	0.92	0.95	0.93