R&D MANAGEMENT

Virtual collaborative spaces: a case study on the antecedents of collaboration in an open-source software community

Guido Conaldi[1,*], Riccardo De Vita[1], Stefano Ghinoi[1,2,3] and Dawn Marie Foster[1,*]

[1]School of Business, Operations and Strategy, University of Greenwich, Park Row, SE10 9LS, London, UK. g.conaldi@greenwich.ac.uk, r.devita@greenwich.ac.uk, s.ghinoi@greenwich.ac.uk dawn@dawnfoster.com
[2]Department of Economics and Management, University of Helsinki, Helsinki, Finland.
[3]Trieste Laboratory on Quantitative Sustainability, Trieste, Italy. s.ghinoi@greenwich.ac.uk
*VMware, Inc, Palo Alto, CA USA

Collaboration enables the sharing amongst individuals of resources and knowledge required to innovate. In recent years, this phenomenon has increasingly manifested in virtual collaborative spaces such as open-source software communities because of the advancement in the use of online technologies and the heightened need for distance work. However, it is still unclear which underlying mechanisms foster collaboration in these spaces. By using the Linux kernel open-source software community as a case study, we analyze data from the linux-pci@vger.kernel.org mailing list to model the influence of proximity on the likelihood of collaboration between individuals. Our dataset is composed of 10,513 message replies to the PCI mailing list posted by its 654 active members in the years 2013 to 2015. Our results show that geographical proximity does not have a direct impact on collaboration, while organizational features defined by institutional and organizational proximity do significantly affect collaboration. Cognitive and social proximity also significantly, and positively, affects collaboration, but these relationships show an inverted u-shaped form. Our results confirm the need to develop specific theorizing about virtual spaces, as they present unique features when compared to traditional physical environments.

1. Introduction

Collaboration is the cornerstone of competitive advantage (Chesbrough, 2003; Nelson, 2018). By promoting interactions between multiple actors, collaboration enables the generation and implementation of innovation to address specific problems and improve the use of resources (Antikainen et al., 2010; Huang and Yu, 2011; Fjeldstad et al., 2012).

The benefits of collaborative innovation have been widely discussed in the literature (e.g., Dodgson et al., 2014; Najafi-Tavani et al., 2018). The coordination of collaborative relationships is, however, a challenging task with a very high risk of failure (van

de Vrande et al., 2009; Ollila and Elmquist, 2011). Success or failure of collaboration depends on environmental, organizational, and individual factors (see McNamara et al., 2020): prominent among them is *space* in its many dimensions, i.e., cognitive, physical, or virtual (Leminen et al., 2020; Ollila and Yström, 2020).

Previous studies focused on how physical spaces such as accelerators, fab labs, incubators, and living labs support collaborative innovation (e.g., Leminen and Westerlund, 2012; Capdevila, 2015; Caccamo, 2020), since shared space can enable the cognitive processes fostering knowledge creation (Peschl and Fundneider, 2012; Mathisen and Jørgensen, 2021). However, this cannot be directly applied to virtual spaces where interactions do not require a shared physical space. Virtual collaborative spaces – such as open-source communities or digital teaching environments – enable the transfer of knowledge using information technologies and an open innovation approach (Aslesen et al., 2019); previous studies on virtual collaborative spaces have looked at the antecedents of innovation in such spaces, but results are mixed (Usoro et al., 2007; De Maggio et al., 2009; Liu et al., 2017; Zhou et al., 2022). Scholars have demonstrated that proximity can enable collaboration and thus innovation also in the virtual space (Aslesen et al., 2019); however, the effects produced by its different dimensions (e.g., Boschma, 2005; Knoben and Oerlemans, 2006) are still unclear, since the virtual space remains overlooked when analyzing collaborative innovation (Bogers et al., 2017). Scholars have highlighted the need for additional studies exploring the antecedents, nature and mechanisms of collaboration in virtual spaces (e.g., Liu et al., 2017; Enkel et al., 2020). This study aims to fill this research gap by addressing the following research question: what are the effects of multiple dimensions of proximity on collaborative innovation in a virtual collaborative space?

By employing a proximity framework and modeling collaboration as a relational event occurring at the individual level (similarly to Brunswicker and Schecter, 2019; Lerner and Lomi, 2020), our work explores the impact of collaborative innovation using an instrumental case study approach (Stake, 1995). Our empirical setting is the open-source Linux kernel community, which is considered a successful case of virtual collaborative space (Lee and Cole, 2003) and has proven to be an interesting context for analyzing virtual communities and open innovation (Nguyen and Ignat, 2018; Dalle et al., 2022; Schaarschmidt, 2022). This community's virtual space can be considered an interaction space where collaboration happens *via* mailing lists (Toral, Martínez Torres, et al., 2009). Our results show that geographical proximity is not key for collaborative innovation, while organizational proximity positively affects collaboration in virtual spaces and institutional proximity negatively influences it; moreover, cognitive proximity and social proximity show an inverted u-shaped curve, i.e., identical cognitive skills and the maintenance of too many contacts can weaken the collaborative process. These effects can be explained in light of the presence of a mix of competitive dynamics and strategies adopted by developers for improving their effectiveness and innovativeness.

The reminder of this paper is structured as follows: the next section reviews current literature on collaborative innovation and virtual spaces and describes the proximity framework. The third section illustrates the research design. The fourth section presents the main results of the analysis, while the last section presents propositions for further studies and discusses managerial implications to support collaborative innovation in virtual spaces.

2. Literature review

2.1. *Innovation in virtual collaborative spaces*

Innovation is a social phenomenon emerging from the interaction of different actors, and it can be seen as a process aimed toward the creation of something that did not exist before (Peschl and Fundneider, 2014). Thanks to the rapid development of information and communication technologies (ICT), organizations are increasingly exploiting innovative solutions to complex problems by promoting collaboration in virtual spaces (Provan and Lemaire, 2015; Liu et al., 2017). Virtual collaborative spaces are interaction spaces 'for individuals, groups and organizations, mediated through ICT' (Aslesen et al., 2019, p. 669). Virtual collaborative spaces have been examined from different perspectives. Teaching-related synchronous environments where students can cooperate remotely using virtual reality (Nishide, 2011; Philippe et al., 2020); virtual communities operating in online platforms used as co-creation spaces (Elia et al., 2020); innovation networks connecting firms, institutions, and inventors *via* asynchronous online communication systems (Aalbers and Whelan, 2021); and open-source software projects in which individuals interact for revising and co-editing software code using public channels of communication (O'Mahony, 2007). Overall, the role of technology to support collaboration is the key element to

understand these spaces. Technologies, such as the virtual reality or virtual forums, have changed traditional workspaces and the way individuals interact, fostering the re-definition of collaborative environments in light of interactions that are not physical anymore, but virtual (Nassiri et al., 2010).

Scholars have discussed the benefits associated with virtual collaborative spaces and the positive effects of accessing different sources of knowledge (Faraj et al., 2011; Germonprez et al., 2013; Akman et al., 2019; Nohutlu et al., 2022). Indeed, in virtual collaborative spaces, the support of digital infrastructures allows for the quick spread of ideas. *Space* enables individuals to contribute to specific projects and stimulates the open innovation process – thanks to the ICT support (Peschl and Fundneider, 2014). Moreover, virtual collaborative spaces facilitate the execution of collaborative tasks by spanning geographical boundaries. However, the notion of space in this context cannot be reduced to its geographical dimension: geographical proximity is, in fact, not a sufficient condition to promote innovation (Boschma, 2005; Mattes, 2012), and other measures of proximity need to be considered. As Amin and Roberts (2008) pointed out, relational proximity can emerge also in virtual contexts lacking in geographical proximity, thus challenging the role originally attributed to the idea of physical distance between actors.

2.2. *The multidimensional concept of proximity in virtual collaborative spaces*

In 2005, Boschma proposed one of the most widely used conceptualization of proximity, decomposing it in the following five dimensions (see Table 1): geographical (spatial closeness), organizational (due to organizational arrangements), institutional (similar rules or cultural norms), social (trust-based relationships), and cognitive (same knowledge base and expertise). Yet, for several years, research on innovation in virtual spaces has followed the view of Morgan (2004, p. 5): the virtual space 'may well be a surrogate for physical proximity in the context of standardized transactions, but not in the context of transactions which are high in complexity, ambiguity and tacitness' – in Morgan's view, physical proximity is equivalent to geographical proximity. Studies in this vein assume that if individuals are geographically close, they are more likely to have physical contacts in addition to the virtual ones and thus collaborate. The virtual environment is primarily used to overcome geographical boundaries, especially when supporting collaborative innovation (Sawhney et al., 2005; De Maggio et al., 2009) and promoting knowledge exchange within organizations (Hwang et al., 2015). However, more recently, Capdevila and Mérindol (2022, p. 15) demonstrated that 'the starting point of the collaborative practices for innovation is not always the physical space', and 'virtual networks of practices can act as precursors of local communities in collaborative spaces'. Their findings have reversed the traditional perspective on the relationships between physical and virtual, but they have not examined in detail the multidimensional concept of proximity in virtual contexts.

Table 1. Definitions of proximity dimensions retrieved from Boschma (2005).

Dimension	Definition
Geographical	Spatial distance between actors, in absolute and relative meaning
Organizational	The extent to which relations are shared in an organizational arrangement, either within or between organizations
Institutional	Actors sharing the same institutional rules of the game, cultural habits and values
Social	Socially embedded relations between actors
Cognitive	Similarity in terms of knowledge base and expertise, which allow to communicate, understand and process new knowledge

This research problem – how multiple dimensions of proximity influence virtual collaborative spaces – is still poorly investigated, even if its importance has emerged as crucial (see Huang et al., 2013; Capdevila and Mérindol, 2022; Clifton et al., 2022). Studies on collaborative innovation in physical spaces found that geographical proximity and institutional proximity have a positive effect on collaboration; organizational proximity is also supporting collaborative interactions; while cognitive and social proximity are key but tend also to show an inverted U-shape curve – i.e., they foster collaboration but only up to a certain point, after which similarities in terms of actors' cognition and trust have a detrimental effect on collaborative innovation (Ponds et al., 2007; Gilsing et al., 2008; D'Este et al., 2012; Steinmo and Rasmussen, 2016; Chen and Xie, 2018).

In virtual spaces, spatial and temporal boundaries are blurred by the digitization of innovation, with increased importance of socio-cognitive sensemaking (Nambisan et al., 2017). Digital technologies facilitate the creation of shared understanding, shaping, and being embedded in, social relationships between organizational actors (Zammuto et al., 2007; Kostis and Ritala, 2020); indeed, 'digitalization has substantially reduced the cost of testing ideas and incorporating feedback, which are crucial in collaborative innovation' (Caccamo, 2020,

p. 188). In virtual collaborative spaces, users and companies create communities to contribute to the development of innovative products and solutions, such as for Threadless, Wikipedia, and Yahoo! Answers (Antikainen, 2011). Aslesen et al. (2019) pointed out that knowledge exchange generated in such spaces is enabled by multiple proximity dimensions. According to these authors, organizational, cognitive and social proximity can be seen as enablers of collaboration in virtual spaces, even if it is unclear what their ultimate impact on collaboration is.

Other studies highlight that social and organizational proximity are important for individuals and organizations in virtual spaces (e.g., Pallot, 2011; Korbi and Chouki, 2017) because maintaining formal and informal relational tie allows to strengthen collaboration. Still, these studies do not concentrate specifically on virtual collaborative spaces, but virtual spaces in general. Moreover, Liu et al. (2017) suggest that individual factors such as trust – which is linked to social proximity – have a positive impact on collaboration, while institutional factors such as norms and regulations – linked to institutional proximity – are relevant as well but potentially less impactful.

The above studies highlight different shortcomings in the research field of virtual collaborative spaces. First, as innovation is influenced by the characteristics of the space in which it takes place (Corsaro and Cantù, 2015; Enkel et al., 2020), findings from studies on physical environments should not be expected to automatically extend to virtual ones. There are differences in the organization and functioning of these environments, and such differences call for new approaches to advance theories of digital innovation management (Nambisan et al., 2017). Second, while proximity as an antecedent of collaboration in physical spaces has been disentangled in its multiple dimensions, in virtual spaces it remains mostly confined to its geographical dimension. Such a view is rooted in the idea that the primary goal of virtual spaces is to span geographical boundaries, even if the existing literature (Nambisan et al., 2017; Kostis and Ritala, 2020) has discussed how digital technologies redefine the spaces for interaction completely, with different social interactions being brought forward by collaboration in virtual spaces.

3. Research design

3.1. Empirical setting and data

The Linux kernel is an established and large open-source software. Linux Software developers routinely collaborate on the source code while being scattered across the globe. Developers can work for private companies, research organizations, or even be nonprofessional contributors (Dalle et al., 2022; Schaarschmidt, 2022). Already at the beginning of the 2000s, companies from different countries were paying their employees to work on the development of the kernel (Hertel et al., 2003). Nowadays, 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). This subsystem collaboration occurs online over more than 240 separate mailing lists.

The Linux kernel documentation helps define this collaboration space by stating 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, 2023). Comments lead to changes and the collaborative editing of these contributions. Indeed, mailing lists have been identified as a primary tool for collaboration in empirical research on open-source software (e.g., Toral, Martínez Torres, et al., 2009).

The fact that the Linux kernel mailing lists constitute the virtual space for collaboration regardless of physical location, employer, specific areas of technical expertise, or other factors was also independently confirmed by collecting primary data: 16 semi-structured qualitative interviews were conducted with Linux software developers chosen using purposive and strategic sample selection methods. The interviews lasted between 30 and 80 minutes and were mostly conducted *via* online video chat, with one in-person interview and two conducted *via* email. All interviews were conducted between May 12, 2015 and May 19, 2017. Intensity and maximum variation sampling strategies were adopted when setting out the sample selection criteria. Interviewees were all experienced Linux contributors who were at the time – or had been in the recent past – employed to work on the Linux kernel by a variety of third-party organizations. Interviews stopped when data saturation had been reached. The full interview guide is presented in Appendix B.

When asked about where they thought collaboration happened, the interviewees consistently mentioned the official mailing lists, where they iterated on new code contributions and general ideas for the Linux kernel. This is best summarized by the following quotes for two interviewees: ‘The 24/7 collaboration that happens is on the mailing

list discussions. That is 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'. Given the convergence of official documentation and interview evidence, for the purpose of this study the Linux kernel mailing lists were identified as the virtual spaces where to investigate the impact of proximity on collaboration.

Specifically, the linux-pci@vger.kernel.org was chosen amongst the 240 existing Linux kernel mailing lists for the purpose of data collection and modeling. This is the virtual space where Peripheral Component Interconnect (PCI) drivers for the Linux kernel are developed, and it was selected for two primary reasons. First, the PCI mailing is one of the top 20 mailing lists as measured by the number of times it is listed in the maintainers file and has 350 active subscribers as of February 2023 (vger.kernel.org, n.d.). Second, the PCI mailing list is a typical example of a top Linux kernel mailing list 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.

To operationalize both collaboration and proximity dimensions a dataset was created combining mailing list data, source code data, and affiliation data on all Linux kernel developers. Both the Linux mailing lists and source code are publicly available. The source code was downloaded and stored into a database using the CVSAnalY software. The PCI mailing list was imported into a database using the MailingListStats software (Robles et al., 2009).

Affiliation data of kernel developers are not public information. An initial dataset containing employer affiliations was obtained from The Linux Foundation directly. This snapshot captured information for the 2013–2015 period approximately. However, this was incomplete for many mailing list members and lacked dates for job changes for those that had changed jobs during the observation period. The missing information was found accessing other online resources. As a result, a mostly complete dataset tracking the job affiliations of the PCI mailing list members during the observation period was obtained. Only in cases where people changed jobs and there were gaps or overlaps that did not provide reliable dates, the midpoint between dates of posts from employer email addresses was taken as the date of the job change. In almost all cases the resolution of gaps and overlaps was straightforward and univocal upon manual data inspection and cleaning. Furthermore, the choice of midpoint dates *versus* any other date in between email activity with two different affiliations does not affect our analytical strategy (see Appendix A.3 for more details). The identity of actors was matched across mailing list, source code contributions, and affiliations using custom-made scripts and manual checks.

Linux kernel development happens in cycles with regular releases. To align with these release cycles, the observation period was set with 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, 12 Linux kernel releases happened.

Our final dataset contains 10,513 message replies to the PCI mailing list by a total of 654 members active during the observation period, all their code contributions to the Linux kernel, and their employment affiliation data.

3.2. Variables

3.2.1. Dependent variable: collaboration events

In our setting, collaboration happens as replies to emails are exchanged by software developers on a mailing list. Thus, we define a single *collaboration event* between two developers as the email reply that a developer (ego) sends to a message previously posted by a fellow developer (alter). A similar approach for measuring collaboration events was used by Quintane et al. (2014, 2022). As alter shares, for example, a new proposal or some piece of code, ego collaborates on them by offering comments, advice, and code changes in their email reply. Our dataset identifies 10,513 collaboration events, each between pairs of the identified 654 PCI system developers. This chronologically ordered sequence of collaboration events is our dependent variable.

3.2.2. Independent variables

We construct a series of independent variables to capture the factors that might make a collaboration event significantly more – or less – likely. These variables capture a characteristic of either ego (the sender in a collaboration event), alter (the receiver), or of both as a pair in each collaboration event. The variables we construct for proximity all capture characteristics of the ego-alter pair. Proximity variables are defined following Boschma's (2005) five dimensions introduced earlier.

Geographical proximity is operationalized using time zone similarity (O'Leary and Cummings, 2007; Chen et al., 2020), since in the case of online communities such as the Linux kernel, there is no spatial dimension to measure (Boschma, 2005; Torre, 2008; Gulati et al., 2012). This measure is normalized to

a value between 0 and 1, and its reciprocal is used for estimating the *Geographical Proximity* variable. Organizational proximity measures whether both ego and alter work for the same employer. An *Organizational Proximity* variable is calculated as a dummy with a value of 1 indicating that both ego and alter work for the same employer or 0 otherwise in a method similar to several proximity studies (Cassi and Plunket, 2015; Crescenzi et al., 2016). Institutional proximity uses the employer affiliation data with a mapping that matches employers to four types of institutions: corporation, non-profit, academic, and hobbyist (unaffiliated). If both actors are employed by the same type of institution, the *Institutional Proximity* variable is set to 1, otherwise, it is set to 0 (similarly to Cao et al., 2019). Only if an actor's affiliation cannot be determined, it is assumed that the person is unaffiliated and included in the hobbyist category. Participation on mailing lists occurs within threads. Over time, developers participating in the same threads develop a sense of increased familiarity and closeness because of this shared experience – which increases the presence of trust. Thus, we define the *Social Proximity* variable as the number of times prior to the collaboration event ego and alter participated in the same mailing list threads. Finally, cognitive proximity is operationalized by considering the similarity between sections of the Linux kernel code where two individuals have contributed. Our interviews with kernel developers confirms the fact that substantially different knowledge and expertise is required to contribute to the various sections. We determine 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 (Hardeman et al., 2015). As a result, the *Cognitive Proximity* variable takes a value between 0 and 1. The variable is also set to 0 if either person has not committed code at all. 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 of the effect of 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). To account for this finding, quadratic versions of the respective variables are also calculated.

Besides proximity, future collaboration events are also bound to be significantly more – or less – likely depending on what collaboration events ego and alter have been part of in the past. This is potentially true also when ego and alter becomes linked by chains of collaboration events involving other developers. These factors can affect collaboration events independently of the levels of proximity between ego and alter, since any interaction between two actors builds over time a structural embeddedness that is known to affect future interaction (Gulati and Gargiulo, 1999). Quintaine et al., (2014) have documented the impact of these structural factors on online interactions of opens source developers. To distinguish between these structural factors and proximity dimension, we build a series of independent structural variables to be used as controls in our analysis following Butts (2008).

Three variables are constructed to capture the effect that past collaboration events between ego and alter might have on future collaboration events between the same ego and alter. *Repeated Collaboration* occurs when a developer might be more likely to reply again to an email sent by a developer they replied to in the past. *Participation shift* captures the fact that an email reply might make *ipso facto* an immediate reply back more likely than any other message to the mailing list. The *Recency effect* captures the diminishing chance of a reply triggering a reply back as messages newer than the initial reply arrive to the mailing list (i.e., the initial reply being 'recent', but not necessarily the latest message sent to the mailing list).

Four variables are constructed to capture the effect that past collaboration events might have on future collaboration events between ego and alter. *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 Collaboration Partners Inbound* and *Shared Collaboration Partners Outbound* instead capture the possible effect on future collaboration events of two developers having been repeatedly involved in collaboration events with the same group of other developers – respectively as the initiators or receivers of those events. These two variables can be conceived to capture the tendency for some of the developers to form closer-knitted, localized collaboration groups – see Robins et al. (2009) for a technical presentation of this type of relational variables.

Finally, independent variables are also constructed to account for three characteristics of ego that might make them more likely to reply to emails to the mailing list: (a) being a mailing list maintainer to capture the role they have in the community; (b) prior code commits to capture individual contribution to the Linux kernel code; (c) being in CC of the mail thread as adding someone in the CC field of a message to

a kernel mailing list is a recognized practice used to increase the chances of a reply (Kernel Development Community, 2023).

Most of the independent variables presented so far are calculated by stratification. For each reply sent to the PCI mailing list, we look back at the past sequence of email replies sent, past code contribution, and affiliation history up to that moment. Going back to the first event can however be computationally intensive and unnecessary in this context. It is common practice to define a time limit for how far in the past these calculations will go based on the empirical context (e.g., Butts, 2008; Quintane et al., 2013, 2014). The Linux kernel development happens in cycles with regular releases and collaboration also follows this pattern. Thus, the median kernel release timing of 63 days for the 12 cycles in our observation period was selected by approximation as our time limit and a moving window of 63 days is used in our calculations.

All operational definitions for the calculated variables are reported in Table 2. More details on how the variables are constructed are reported in Appendix A.2. Descriptive statistics for all calculated variables and correlations are reported in Table 3.

Table 2. Variable operationalization summary.

Variable type	Variable operational definition	References
Dependent variable	Collaboration event operationalized as a reply to a message on the mailing list	Quintane et al. (2014, 2022)
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)	O'Leary and Cummings (2007); Chen et al. (2020)
Organizational	1 if both work for the same employer, otherwise 0	Cassi and Plunket (2015); Crescenzi et al. (2016)
Institutional	1 if both work for the same type of third-party organization, otherwise 0	Cao et al. (2019)
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	Hardeman et al. (2015)
Control variables		
Repeated Collaboration	Number of times the ego replied to messages from the alter within the moving window	Butts (2008)
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 Collaboration Partners Inbound	Number of third parties who have replied to both the ego and the alter within the moving window	
Shared Collaboration Partners Outbound	Number of times the ego and the alter have replied to messages by the same third party	
Alter Maintainer	1 if the alter is a maintainer, otherwise 0	
Either Maintainer	1 if the ego or the alter, or both 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 or the alter or both have committed code within the moving window, otherwise 0	
Ego in Copy	1 if the ego was explicitly included in the 'to' or 'cc' field of the email that was replied to, otherwise 0	

Table 3. Descriptive statistics and correlations.

	Variable	Mean	SD	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	Alter Maintainer	0.093	0.290																
2	Either Maintainer	0.185	0.388	0.67															
3	Alter Committer	0.823	0.381	0.09	0.08														
4	Either Committer	0.970	0.170	0.03	0.05	0.38													
5	Ego in Copy	0.179	0.383	0.00	–0.02	0.02	0.04												
6	Geographic Proximity	0.706	0.233	0.03	0.03	0.00	–0.03	–0.01											
7	Organizational Proximity	0.090	0.286	0.01	–0.01	0.07	–0.02	0.00	0.27										
8	Institutional Proximity	0.738	0.440	0.02	0.04	0.11	0.04	0.03	–0.06	0.19									
9	Social Proximity	4.780	18.268	–0.05	–0.09	0.10	0.03	–0.02	0.21	0.55	0.11								
10	Cognitive Proximity	0.131	0.237	–0.01	–0.05	0.26	0.10	0.04	0.20	0.61	0.19	0.66							
11	Repeated Collaboration	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	0.090	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	0.167	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	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	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 Collab. Partners Inbound	19.084	22.776	–0.08	–0.14	0.18	0.09	0.08	0.05	0.36	0.19	0.67	0.62	0.80	0.15	0.31	0.96	0.95	
17	Shared Collab. Partners Outbound	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

3.3. Methods

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 (REM) are based on relational events, defined as events generated by a sender directed toward a receiver and are represented by sender, receiver, action type, and time (Butts, 2008). REM use a sequence of actions generated by egos and directed toward alters to directly estimate what variables have a significant effect on the likelihood of future events (Butts, 2008).

Mailing list replies with a sender, and time stamp for each message like the *collaboration events* we defined provide the ideal data structure for relational event models. Here we use REM to test if proximity with other developers makes a developer more or less likely to initiate a collaboration event with them. REM are chosen because they model sequence data without losing information through aggregation (Quintane et al., 2014) and allow for the effect of each proximity variable on collaboration events to be estimated independently. They offer a multivariate statistical framework where the effects of structural and other control variables can also be controlled for.

The ordinal version of REM 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). 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):

$$P_{ij} = \frac{exp\left(x'_{ij}\beta\right)}{\sum_{j=i}^{J} x'_{ij}\beta} \qquad \text{(eq 1)}$$

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

The models we report are estimated in R (R Core Team, 2022) using the function *clogit* within the *survival* package (Therneau, 2021), which makes use of *coxph* function internally. The estimation procedure scales and centers the raw variables, which leads to more numerical stability without changing the results of the regression analysis.

See Appendix A for further details on the chosen estimation procedure and variables construction.

4. Results

The estimated models are reported in Table 4. In the Baseline Model the chances of a collaboration event happening can only be affected by the individual history of past collaboration events between the same ego and alter (*Repeated Collaboration*). Heuristically, the goodness-of-fit diagnostics listed in Table 4 indicate that the Full Model improves significantly on the Baseline Model after accounting for the difference in degrees of freedom.

In the Benchmark Model *Repeated Collaboration* is positive and significant, although relatively small, indicating that an increase of one standard deviation in *Repeated Collaboration* generates an increase of 0.6% (exp(0.006) = 1.006) in the hazard of a future collaboration event happening (i.e., an increase in the chances of it happening) for each repeated event happened in the past. The effect becomes nonsignificant in the Full Model. This means that once both proximity and structural control variables are included in the model, the simple count of past collaboration events between ego and alter is not a good predictor of future collaboration events between the same ego and alter.

Continuing with the Full Model, *Institutional Proximity* is negative and significant, indicating that a collaboration event is 12.3% (exp(−0.131) = 0.877) less likely if ego and alter are from the same type of institution (company, non-profit, academic, or unaffiliated). 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. In contrast, *Organizational Proximity* is positive and significant, which indicates that a collaboration event is 87.2% (exp(0.627) = 1.872) more likely if both people are employed by the same third-party organization. In the Linux kernel, if a specific 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.

The *Cognitive Proximity* effect is positive, significant, and very strong, indicating that a collaboration event is 539.8% (exp(1.856) = 6.398) more likely between two people who have contributed to the same sections of the Linux kernel source code during the moving window. In combination with the negative and significant squared effect, the results indicate that cognitive proximity has an inverted u-shaped curve (see Figure 1), leading to the conclusion that

Table 4. Partial likelihood estimates of relational event models.

Variables	Baseline model			Full model		
Repeated Collaboration	0.006	(0.001)	***	0.002	(0.001)	
Institutional Proximity				−0.131	(0.049)	**
Organizational Proximity				0.627	(0.133)	***
Cognitive Proximity				1.856	(0.403)	***
Cognitive Proximity Squared				−4.287	(0.888)	***
Social Proximity				1.052	(0.243)	***
Social Proximity Squared				−0.050	(0.013)	***
Participation Shift				−0.163	(0.151)	
Geographic Proximity				0.137	(0.074)	
Recency Effect				0.882	(0.227)	***
Transitive Closure				0.036	(0.010)	***
Cyclic Closure				0.086	(0.019)	***
Shared Collaboration Partners Inbound				−0.077	(0.018)	***
Shared Collaboration Partners Outbound				−0.127	(0.029)	***
Alter Maintainer				−0.059	(0.015)	***
Either Maintainer				0.218	(0.075)	**
Alter Committer				−0.217	(0.053)	***
Either Committer				0.637	(0.145)	***
Ego in Copy				2.618	(0.688)	***
BIC	37,549.35			18,032.80		
Log-likelihood	−18,770.04			−8,928.42		
LR test				19,683		
Observations (realized events + controls)	63,072			63,072		
Realized events	10,512			10,512		

Significance ***$P<.001$, **$P<.01$, *$P<.05$; Robust standard errors shown in parentheses.

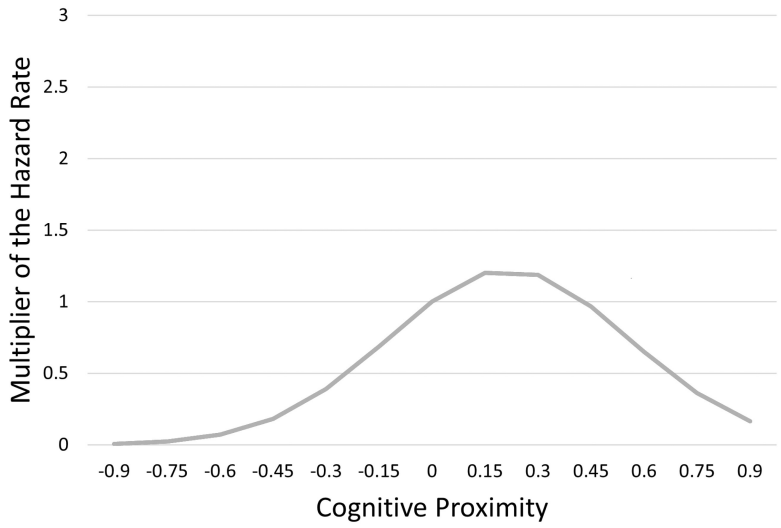

Figure 1. Effect of cognitive proximity on collaboration events.

the likelihood of collaboration increases strongly and quickly as the cognitive proximity between two developers increases, but only up to a point. After that, 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 *Social Proximity* effect is also positive, significant, and very strong, indicating that a collaboration event is 186.3% (exp(1.052) = 2.863) more likely

between two people who have participated in the same threads on the mailing list during the moving window. This indicates the presence of trust, strongly related to social proximity (Boschma, 2005). This is also in line with evidence that trust – built over time – is a key element in online communities (Toral, Rocío Martínez-Torres, et al., 2009). Like with cognitive proximity, the squared effect for social proximity is 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.

The *Geographical Proximity* effect is nonsignificant; thus, the model provides no evidence that being in similar time zones influences the likelihood of collaboration. Further analyses show that *Geographical Proximity* becomes nonsignificant when the structural control variables are added to the model. What could have been mistaken for an independent effect of geographical proximity on the likelihood of collaboration in a virtual space, is instead explained by the structural patterns that collaboration follows – once they are accounted for in the model.

Despite not affecting the ability to estimate the model correctly, some of the correlations between proximity variables are relatively high (see Table 3). This cannot be explained by the variables measuring proximity between contributors along the same axis since the variables are all operationalized without overlap, using data coming from different sources. Further analysis would be required, but the correlations reported at least suggest that the variables are indeed capturing related dimensions of the same general concept of proximity.

When looking at the effect of structural variables, we see that the *Recency effect* is significant – and positive – but the *Participation Shift* effect is not. This indicates that a collaboration event is 141.6% (exp(0.882) = 2.416) more likely if the alter has recently emailed the ego, without the need for it to be exactly the most recent email ego has received. These results suggest that collaboration happens in bursts between pairs of developers, with relatively recent collaboration events having a cumulative effect on the likelihood of further collaboration happening short-term.

The effects for *Transitive Closure* and *Cyclic Closure* are both positive and significant. These results indicate that each collaboration event that leads to the formation of collaboration ties between triplets of developers increases the likelihood of collaboration by – respectively – 3.7% (exp(0.036) = 1.037) and 9% (exp(0.086) = 1.090). Together with the negative and significant effects for the *Shared Partnership Inbound* and *Outbound,* these results suggest that very localized clusters of developers tend to form chains of collaboration events significantly increasing the likelihood of future collaboration.

The *Alter maintainer* effect is negative and significant, indicating that collaboration events are 5.7% (exp(−0.059) = 0.943) less likely to occur if alter is a maintainer. 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 ego is a maintainer – and more so if both ego and alter are. The *Alter Committer* effect is negative and significant indicating that a collaboration event is 19.5% (exp(−0.217) = 0.805) less likely if alter has had source code committed. Combining the negative *Alter Committer* effect with the positive effect of the *Either Committer* variable indicates that a collaboration event becomes more likely if ego has committed code – and more so if both ego and alter have. The similarity with the results for *Committer* variables is explained by the fact that developers with more prior code contributions to the kernel also become more selective and less available for general collaboration, even if they have not become maintainers. Finally, the *Ego in Copy* effect is positive, significant, and extremely strong, indicating that a collaboration event is 1270.8% (exp(2.618) = 13.708) more likely to happen if ego's email address is included in either the 'To' or 'CC' fields of the initial email. This result is not surprising, because as described previously, this is a documented practice when submitting software patches for the Linux kernel that signals a much higher priority to be given to the message by the community.

5. Discussion and conclusion

This study examines the importance of different dimensions of proximity on collaboration between members of a virtual collaborative space. By focusing also on the structural patterns established by actors in this virtual space, the paper makes important contributions to the literature. A recurrent assumption from previous studies is that these spaces are created to span geographical boundaries; while other dimensions of proximity are at play in influencing collaboration, current studies provide limited empirical evidence about their effects. Our work contributes to the literature on innovation and virtual collaborative spaces by adding novel insights on the influence of proximity as an antecedent of innovation. It demonstrates that virtual spaces are characterized by complex intra and inter-organizational interactions, confirming therefore the need for deeper

theorizing. In this vein, we position our research in line with recent studies which are encouraging to move beyond traditional intra-organizational studies (e.g., Provan and Lemaire, 2015) and call for novel approaches to investigate individual and organizational factors influencing collaboration in virtual spaces. Virtual collaborative spaces are characterized by the presence of individual contributors and organizations; the collaborative relationships developed in these spaces have a clear inter-organizational feature – even if intra-organizational aspects are present as well, since collaboration is possible also between individuals employed by the same organization. This research demonstrates that proximity theory can be used effectively as a theoretical lens to better understand collaborative innovation in virtual collaborative spaces. In traditional organizations, collaboration can be enforced by hierarchy; however, the flexible boundaries and evolving structures of these communities require that participants rely on common ground to facilitate effective collaboration. Proximity theory is one way of understanding such common ground and the results from our study about the impact of different forms of proximity on collaboration provide fertile ground to develop propositions to be tested in future research.

Results of our analysis suggest that geographical proximity is not the primary antecedent for collaboration in virtual collaborative spaces. This finding is partially contradicting those reported by Morgan (2004) that virtual spaces cannot fully replace geographical proximity, especially when developing innovation. Morgan's idea (2004) has been supported by the works of Stephens and Poorthuis (2015) and Takhteyev et al. (2012), which found that virtual environments can reduce the constraints imposed by the physical space, but they cannot completely remove them. Instead, our study demonstrates that virtual spaces are characterized by complex intra and inter-organizational interactions, confirming therefore the need for deeper theorizing.

Our results corroborate the idea that effect of cognitive proximity and social proximity on collaboration take the form of an inverted u-shaped curve. This is consistent with previous studies (Boschma, 2005; Sorenson et al., 2006; Nooteboom et al., 2007) and suggests that in this respect virtual collaborative spaces do not differ from physical spaces. To use knowledge effectively and reach successful innovation targets, individuals must consider that similar cognitive skills reduce innovativeness past a certain threshold, because heterogeneous knowledge is needed for creativity; moreover, too much social proximity can weaken the collaborative process because of the efforts needed to maintain multiple contacts, and because of the challenges emerging from managing large groups of people – especially in the context of virtual collaborative spaces, which are supposed to reduce the constraints imposed by hierarchical structures. The fact that some of our results corroborate existing knowledge, while others differentiate virtual collaborative spaces from physical ones, lead to the following proposition:

> **Proposition 1** Proximity – in its various dimensions – influences collaboration in virtual and physical spaces differently.

A second proposition can be developed looking at the combined results for organizational and institutional proximity. Organizational proximity – working for the same employer in our context – produces an increased likelihood of collaboration, which confirms Provan and Lemaire's results (2015): the more individuals have an easy access to others, the more they can establish strong collaboration ties – and this ease of access can be offered by the company for whom these individuals work. However, this result together with the negative coefficient for institutional proximity can be interpreted as an indication of the existence of competitive dynamics within the community (as highlighted by Germonprez et al., 2013), whereas previous studies (e.g., Teixeira et al., 2021) found that online communities support inclusiveness and try to minimize rivalry. The result for institutional proximity also supports the idea that individuals establish connections with other developers to maximize the benefits of being in contact with people with different expertise – therefore improving their own effectiveness and innovativeness of their own contributions (Faraj et al., 2011). These considerations support the development of a second proposition:

> **Proposition 2** Virtual collaborative spaces are characterized by the simultaneous interplay of competitive and collaborative dynamics.

Our findings have managerial implications. As highlighted by Liu et al. (2017, p. 664), organizations – as well as individuals – need to understand 'the importance of sharing similar values with the development partners in a network if they are to join it'. Our results suggest that institutional proximity has a negative effect on collaboration: managers need to take this into account when deciding to promote collaborations and the engagement of their affiliates in open innovation projects that cross institutional boundaries. Moreover, promoting interactions in virtual collaborative spaces requires finding a suitable employee to participate, or possibly

involving someone who already contributes to the project. Based on the cognitive proximity findings, it is important to select individuals with skills that are appropriate for the areas of the project where they are expected to contribute; participating in multiple areas might allow them to generate more innovative ideas. However, a balance of consolidated experience and novel expertise is preferable for establishing collaborative ties, as indicated by the u-shaped cognitive proximity curve: managers should focus on understanding how individuals with shared competences can be allocated to different project areas and thus create synergies useful to address problems requiring multifaceted perspectives. Potential opportunities for collaboration in virtual spaces should therefore consider: (a) the importance of being exposed to a variety of knowledge and information; (b) the constraints imposed by the number of collaborations that can be established; (c) the issues raised by competitive behaviors and how to address them.

Finally, we need to acknowledge the limitations of our study. First, we investigated a case study of software developers in a single open-source software community, which implies that our work might lack in generalizability. Second, while mailing lists have been widely used to study collaboration in open-source software communities (Toral, Martínez Torres, et al., 2009, Toral et al., 2010), we did not explore the content of the messages shared. This information could allow us to identify key themes or problems discussed, or the sense of belonging to the community by the developers. Future studies could look at this aspects and determine if the content of the conversation can be considered a driver of collaboration itself, and if it is associated with developers' proximity. Also, the conceptualization of proximity measures – and the moderation effect of one (or more) measures on others – can be the subject of further discussion. We have followed Boschma's approach (2005) and adapted it to the virtual environment. However, other approaches to measuring proximity can be used: in particular, cognitive and social proximity can be measured by collecting primary data on shared interests, experiences and skills, friendship – and potentially looking at proximity in a dynamic perspective (see Öberg, 2018). Another research avenue that deserves further investigation relates to the influence of the organizational environment in which developers work. Some developers work for large corporations, while others for small companies; it would be interesting to understand if company size and structure are relevant to explain collaboration propensity. Finally, researchers should investigate the intrinsic motivations behind the contribution of developers and test if different motivations affect the way they collaborate in a virtual space.

Data availability statement

The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

REFERENCES

Aalbers, R. and Whelan, E. (2021) Implementing digitally enabled collaborative innovation: a case study of online and offline interaction in the German automotive industry. *Creativity and Innovation Management*, **30**, 2, 368–383.

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.

Akman, H., Plewa, C., and Conduit, J. (2019) Co-creating value in online innovation communities. *European Journal of Marketing*, **53**, 6, 1205–1233.

Amin, A. and Roberts, J. (2008) Knowing in action: beyond communities of practice. *Research Policy*, **37**, 2, 353–369.

Antikainen, M. (2011) *Facilitating customer involvement in collaborative online innovation communities*. PhD dissertation, VTT publications 760, Helsinki, Finland.

Antikainen, M., Mäkipää, M., and Ahonen, M. (2010) Motivating and supporting collaboration in open innovation. *European Journal of Innovation Management*, **13**, 1, 100–119.

Aslesen, H.W., Martin, R., and Sardo, S. (2019) The virtual is reality! On physical and virtual space in software firms' knowledge formation. *Entrepreneurship & Regional Development*, **31**, 9–10, 669–682.

Bogers, M., Zobel, A.-K., Afuah, A., Almirall, E., Brunswicker, S., Dahlander, L., Frederiksen, L., Gawer, A., Gruber, M., Haefliger, S., Hagedoorn, J., Hilgers, D., Laursen, K., Magnusson, M.G., Majchrzak, A., McCarthy, I.P., Moeslein, K.M., Nambisan, A., Piller, F.T., Radziwon, A., RossiLamastra, C., Sims, J., and Ter Wal, A.L.J. (2017) The open innovation research landscape: established perspectives and emerging themes across different levels of analysis. *Industry and Innovation*, **24**, 1, 8–40.

Boschma, R.A. (2005) Proximity and innovation: a critical assessment. *Regional Studies*, **39**, 1, 61–74.

Brunswicker, S. and Schecter, A. (2019) Coherence or flexibility? The paradox of change for developers' digital innovation trajectory on open platforms. *Research Policy*, **48**, 103771.

Butts, C.T. (2008) A relational event framework for social action. *Sociological Methodology*, **38**, 1, 155–200.

Caccamo, M. (2020) Leveraging innovation spaces to foster collaborative innovation. *Creativity and Innovation Management*, **29**, 178–191. https://doi.org/10.1111/caim.12357.

Cao, Z., Derudder, B., and Peng, Z. (2019) Interaction between different forms of proximity in inter-organizational

scientific collaboration: the case of medical sciences research network in the Yangtze River Delta region. *Papers in Regional Science*, **98**, 1903–1924.
Capdevila, I. (2015) Co-working spaces and the localised dynamics of innovation in Barcelona. *International Journal of Innovation Management*, **19**, 3, 1540004.
Capdevila, I. and Mérindol, V. (2022) Emergence of communities through interdependent dynamics of physical, cognitive and virtual contexts: The case of collaborative spaces. *R&D Management*. https://doi.org/10.1111/radm.12561.
Cassi, L. and Plunket, A. (2015) Research collaboration in co-inventor networks: combining closure, bridging and proximities. *Regional Studies*, **49**, 6, 936–954.
Chen, H. and Xie, F. (2018) How technological proximity affect collaborative innovation? An empirical study of China's Beijing–Tianjin–Hebei region. *Journal of Management Analytics*, **5**, 4, 287–308.
Chen, L., Ye, Y., Zheng, A., Xie, F., Zheng, Z., and Lyu, M.R. (2020) Incorporating geographical location for team formation in social coding sites. *World Wide Web*, **23**, 1, 153–174.
Chesbrough, H.W. (2003) *Open Innovation: The New Imperative for Creating and Profiting from Technology*. Boston, MA: Harvard Business School Press.
Clifton, N., Carroll, F., and Wheeler, R. (2022) Proximity, innovation, collaboration; developing the 4th "extended reality" space. *Dialogue and Universalism*, **32**, 2, 61–82.
Corbet, J. and Kroah-Hartman, G. (2017) *Linux Kernel Development Report*. San Francisco, CA: The Linux Foundation.
Corsaro, D. and Cantù, C. (2015) Actors' heterogeneity and the context of interaction in affecting innovation networks. *Journal of Business & Industrial Marketing*, **30**, 3/4, 246–258.
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.
Dahlander, L. and O'Mahony, S. (2010) Progressing to the center: coordinating project work. *Organization Science*, **22**, 4, 961–979.
Dalle, J.-M., David, P.A., Rullani, F., and Bolici, F. (2022) The interplay between volunteers and firm's employees in distributed innovation: emergent architectures and stigmergy in open source software. *Industrial and Corporate Change*, **31**, 1358–1386. https://doi.org/10.1093/icc/dtac037.
De Maggio, M., Gloor, P.A., and Passiante, G. (2009) Collaborative innovation networks, virtual communities and geographical clustering. *International Journal of Innovation and Regional Development*, **1**, 4, 387–404.
D'este, P., Guy, F., and Iammarino, S. (2012) Shaping the formation of university–industry research collaborations: what type of proximity does really matter? *Journal of Economic Geography*, **13**, 4, 537–558.
Dodgson, M., Gann, D.M., and Phillips, N. (2014) *The Oxford Handbook of Innovation Management*. Oxford: Oxford University Press.
Elia, G., Messeni Petruzzelli, A., and Urbinati, A. (2020) Implementing open innovation through virtual brand communities: a case study analysis in the semiconductor industry. *Technological Forecasting and Social Change*, **155**, 119994.
Enkel, E., Bogers, M., and Chesbrough, H. (2020) Exploring open innovation in the digital age: a maturity model and future research directions. *R&D Management*, **50**, 1, 161–168.
Faraj, S., Jarvenpaa, S.L., and Majchrzak, A. (2011) Knowledge collaboration in online communities. *Organization Science*, **22**, 5, 1224–1239.
Fjeldstad, Ø.D., Snow, C.C., Miles, R.E., and Lettl, C. (2012) The architecture of collaboration. *Strategic Management Journal*, **33**, 734–750.
Germonprez, M., Allen, J.P., Warner, B., Hill, J., and McClements, G. (2013) Open source communities of competitors. *Interactions*, **20**, 6, 54–59.
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.
Greene, W.H. (2012) *Econometric analysis*, 7th edn. 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.
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.
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**, 1159–1177.
Huang, Y., Shen, C., and Contractor, N.S. (2013) Distance matters: exploring proximity and homophily in virtual world networks. *Decision Support Systems*, **55**, 4, 969–977.
Huang, K. and Yu, C.J. (2011) The effect of competitive and non-competitive R&D collaboration on firm innovation. *Journal of Technology Transfer*, **36**, 4, 383–403.
Hwang, E.H., Singh, P.V., and Argote, L. (2015) Knowledge sharing in online communities: learning to cross geographic and hierarchical boundaries. *Organization Science*, **26**, 6, 1593–1611.
Kernel Development Community (2023) The Linux Kernel documentation [online] Available from: https://www.kernel.org/doc/html/latest/ [Accessed 10 Feb. 2023].
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.
Korbi, F.B. and Chouki, M. (2017) Knowledge transfer in international asymmetric alliances: the key role

of translation, artifacts, and proximity. *Journal of Knowledge Management*, **21**, 5, 1272–1291.
Kostis, A. and Ritala, P. (2020) Digital artifacts in industrial co-creation: how to use VR technology to bridge the provider-customer boundary. *California Management Review*, **62**, 4, 125–147.
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.
Lee, G. and Cole, R. (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.
Leminen, S., Nyström, A.-G., and Westerlund, M. (2020) Change processes in open innovation networks – exploring living labs. *Industrial Marketing Management*, **91**, 701–718.
Leminen, S. and Westerlund, M. (2012) Towards innovation in living labs networks. *International Journal of Product Development*, **17**, 1–2, 43–49.
Lerner, J. and Lomi, A. (2020) Reliability of relational event model estimates under sampling: how to fit a relational event model to 360 million dyadic events. *Network Science*, **8**, 1, 97–135.
Liu, M., Hull, C.E., and Hung, Y.-T.C. (2017) Starting open source collaborative innovation: the antecedents of network formation in community source. *Information Systems Journal*, **27**, 5, 643–670.
Mathisen, L. and Jørgensen, E.J.B. (2021) The significance of knowledge readiness for co-creation in university industry collaborations. *Innovation: Organization and Management*, **23**, 4, 534–551. https://doi.org/10.1080/14479338.2021.1882862.
Mattes, J. (2012) Dimensions of proximity and knowledge bases: innovation between spatial and non-spatial factors. *Regional Studies*, **46**, 8, 1085–1099.
McNamara, M.W., Miller-Stevens, K., and Morris, J.C. (2020) Exploring the determinants of collaboration failure. *International Journal of Public Administration*, **43**, 1, 49–59.
Morgan, K. (2004) The exaggerated death of geography: learning, proximity and territorial innovation systems. *Journal of Economic Geography*, **4**, 3–21.
Najafi-Tavani, S., Najafi-Tavani, Z., Naudé, P., Oghazi, P., and Zeynaloo, E. (2018) How collaborative innovation networks affect new product performance: product innovation capability, process innovation capability, and absorptive capacity. *Industrial Marketing Management*, **73**, 193–205.
Nambisan, S., Lyytinen, K., Majchrzak, A., and Song, M. (2017) Digital innovation management: reinventing innovation management research in a digital world. *MIS Quarterly*, **41**, 1, 223–238.
Nassiri, N., Powell, N., and Moore, D. (2010) Human interactions and personal space in collaborative virtual environments. *Virtual Reality*, **14**, 4, 229–240.
Nelson, R.R. (2018) Economics from an evolutionary perspective. In: Nelson, R.R., Dosi, G., Helfat, C.E., Pyka, A., Saviotti, P.P., Lee, K., Dopfer, K., Malerba, F., and Winter, S.G. (eds.), *Modern Evolutionary Economics: An Overview*. Cambridge: Cambridge University Press, pp. 1–34.
Nguyen, H.L. and Ignat, C.-L. (2018) An analysis of merge conflicts and resolutions in Git-based open source projects. *Computer Supported Cooperative Work*, **27**, 741–765.
Nishide, R. (2011) Prospects for digital campus with extensive applications of virtual collaborative space. *Journal of Interactive Learning Research*, **22**, 3, 421–443.
Nohutlu, Z.D., Englis, B.G., Groen, A.J., and Constantinides, E. (2022) Customer cocreation experience in online communities: antecedents and outcomes. *European Journal of Innovation Management*, **25**, 2, 630–659.
Nooteboom, B. (1999) *Inter-Firm Alliances: Analysis and Design*. London: Routledge.
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.
Öberg, C. (2018) The dynamics of proximity in multiple-party innovation processes. *IMP Journal*, **12**, 2, 296–312.
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.
Ollila, S. and Elmquist, M. (2011) Managing open innovation: exploring challenges at the interfaces of an open innovation arena. *Creativity and Innovation Management*, **20**, 273–283.
Ollila, S. and Yström, A. (2020) Open laboratories as "in-between spaces". In: Fritzsche, A., Jonas, J.M., Roth, A., and Möslein, K.M. (eds), *Innovating in the Open Lab: The New Potential for Interactive Value Creation Across Organizational Boundaries*. Oldenbourg: De Gruyter, pp. 203–212.
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.
Pallot, M.A. (2011) *Collaborative Distance: Investigating Issues Related to Distance Factors Affecting Collaboration Performance*. PhD Thesis. Nottingham University Business School, Nottingham, UK.
Peschl, M.F. and Fundneider, T. (2012) Spaces enabling game-changing and sustaining innovations: why space matters for knowledge creation and innovation. *Journal of Organisational Transformation & Social Change*, **9**, 1, 41–61.
Peschl, M.F. and Fundneider, T. (2014) Designing and enabling spaces for collaborative knowledge creation and innovation: from managing to enabling innovation as socio-epistemological technology. *Computers in Human Behavior*, **37**, 346–359.
Philippe, S., Souchet, A.D., Lameras, P., Petridis, P., Caporal, J., Coldeboeuf, G., and Duzan, H. (2020) Multimodal teaching, learning and training in virtual reality: a review and case study. *Virtual Reality and Intelligent Hardware*, **2**, 5, 421–442.
Ponds, R., van Oort, F., and Frenken, K. (2007) The geographical and institutional proximity of research

collaboration. *Papers in Regional Science*, **86**, 3, 423–443.
Provan, K.G. and Lemaire, R.H. (2015) Positional embeddedness in a community source software development project network: the importance of relationship intensity. *R&D Management*, **45**, 5, 440–457.
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.
Quintane, E., Wood, M., Dunn, J., and Falzon, L. (2022) Temporal brokering: a measure of brokerage as a behavioral process. *Organizational Research Methods*, **25**, 3, 459–489.
R Core Team (2022) *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.
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.
Sawhney, M., Verona, G., and Prandelli, E. (2005) Collaborating to create: the internet as a platform for customer engagement in product innovation. *Journal of Interactive Marketing*, **19**, 4, 4–17.
Schaarschmidt, M. (2022) Innovating beyond firm boundaries: resource deployment control in open source software development. *Information Technology and People*. https://doi.org/10.1108/ITP-08-2021-0624.
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*. Berlin: ACM. p. 2.
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, 1546–1588.
Stake, R.E. (1995) *The art of case study research*. Thousand Oaks, CA: Sage Publications.
Steinmo, M. and Rasmussen, E. (2016) How firms collaborate with public research organizations: the evolution of proximity dimensions in successful innovation projects. *Journal of Business Research*, **69**, 3, 1250–1259.
Stephens, M. and Poorthuis, A. (2015) Follow thy neighbor: connecting the social and the spatial networks on twitter. *Computers Environment and Urban Systems*, **53**, 87–95.
Takhteyev, Y., Gruzd, A., and Wellman, B. (2012) Geography of twitter networks. *Social Networks*, **34**, 73–81.
Teixeira, J.A., Leppänen, V., and Hyrynsalmi, S. (2021) *Network Science, Homophily and Who Reviews Who in the Linux Kernel?* ArXiv preprint. Available at: https://arxiv.org/abs/2106.09329
Therneau, T. (2021) *A Package for Survival Analysis in R*. R package version 3.5-0. https://CRAN.R-project.org/package=survival
Toral, S.L., Martínez Torres, M.R., and Barrero, F. (2009) Modelling mailing list behaviour in open source projects: the case of ARM embedded Linux. *Journal of Universal Computer Science*, **15**, 3, 648–664.
Toral, S.L., Martínez-Torres, M.R., and Barrero, F. (2010) Analysis of virtual communities supporting OSS projects using social network analysis. *Information and Software Technology*, **52**, 3, 296–303.
Toral, S.L., Rocío Martínez-Torres, M., Cortés, F., and Barrero, F. (2009) An empirical study of the driving forces behind online communities. *Internet Research*, **19**, 4, 378–392.
Torre, A. (2008) On the role played by temporary geographical proximity in knowledge transmission. *Regional Studies*, **42**, 6, 869–889.
Usoro, A., Sharratt, M.W., Tsui, E., and Shekhar, S. (2007) Trust as an antecedent to knowledge sharing in virtual communities of practice. *Knowledge Management Research and Practice*, **5**, 3, 199–212.
vger.kernel.org (n.d.) Majordomo Lists at VGER. KERNEL.ORG. [online] Available at: http://vger.kernel.org/vger-lists.html#linux-pci [Accessed 10 Feb. 2023].
van de Vrande, V., de Jong, J.P.J., Vanhaverbeke, W., and de Rochemont, M. (2009) Open innovation in SME's: trends, motives and management challenges. *Technovation*, **29**, 423–437.
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.
Zhou, J., Kishore, R., Zuo, M., Liao, R., and Tang, X. (2022) Older adults in virtual communities: understanding the antecedents of knowledge contribution and knowledge seeking through the lens of socioemotional selectivity and social cognitive theories. *Journal of Knowledge Management*, **26**, 4, 972–992.

Guido Conaldi is Senior Lecturer in Economic Sociology at the School of Business, Operations and Strategy, University of Greenwich. He holds an MSc in Sociology from the London School of Economics and a PhD in Social Sciences from the Sant'Anna School of Advanced Studies in Pisa. His research interests lie in the area of interpersonal and organizational social networks. His current research investigates the social mechanisms contributing to the endogenous formation of structure and hierarchy in self-managing teams.

Riccardo De Vita is Professor of Innovation Management and Head of School, Business, Operations and Strategy, University of Greenwich. Previously he worked at the Università Carlo Cattaneo – LIUC in Italy, where he obtained his PhD. In his research, Riccardo applies social network analysis to a wide range of business issues. His recent work focuses on the study of innovation networks as well as applications of social network analysis to the higher education management field.

Stefano Ghinoi is Senior Lecturer in Economic Sociology at the School of Business, Operations and Strategy, University of Greenwich, and visiting scholar at the Department of Economics and Management, University of Helsinki. He holds a PhD in Economic Statistics from the University of Bologna; he has worked as a consultant for private companies and public authorities in several European countries, and his main research interests include innovation and sustainability, organizational networks, and policy evaluation.

Dawn Marie Foster currently leads the open-source community strategy efforts within VMware's Open Source Program Office. Dawn is member of the OpenUK board, the TODO steering committee, and the CHAOSS board. She received her PhD from the University of Greenwich with a thesis on collaboration in fluid organizations.

APPENDIX A

Sampling approach

In REM independent variables are calculated for each 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. This comparison is needed to determine which variables influence the likelihood of a collaboration event. However, our dataset is composed of 10,513 realized events and it would be computationally prohibitive to calculate all the variables for every possible unrealized event. Therefore, a case–control approach (Sorenson and Stuart, 2001; Sorenson et al., 2006; Cassi and Plunket, 2015) is used with a sampling strategy where each realized event 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 messages posted in the previous seven days that could have been replied to as alternatives to the realized event. The seven-day cutoff is chosen to control for the temporal variation characterizing our dataset. As shown in Figure 2, the PCI mailing list experiences anywhere from only a few posts to over 140 posts per day. Mailing list replies are also not equally likely over the entire dataset: 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. 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 com-

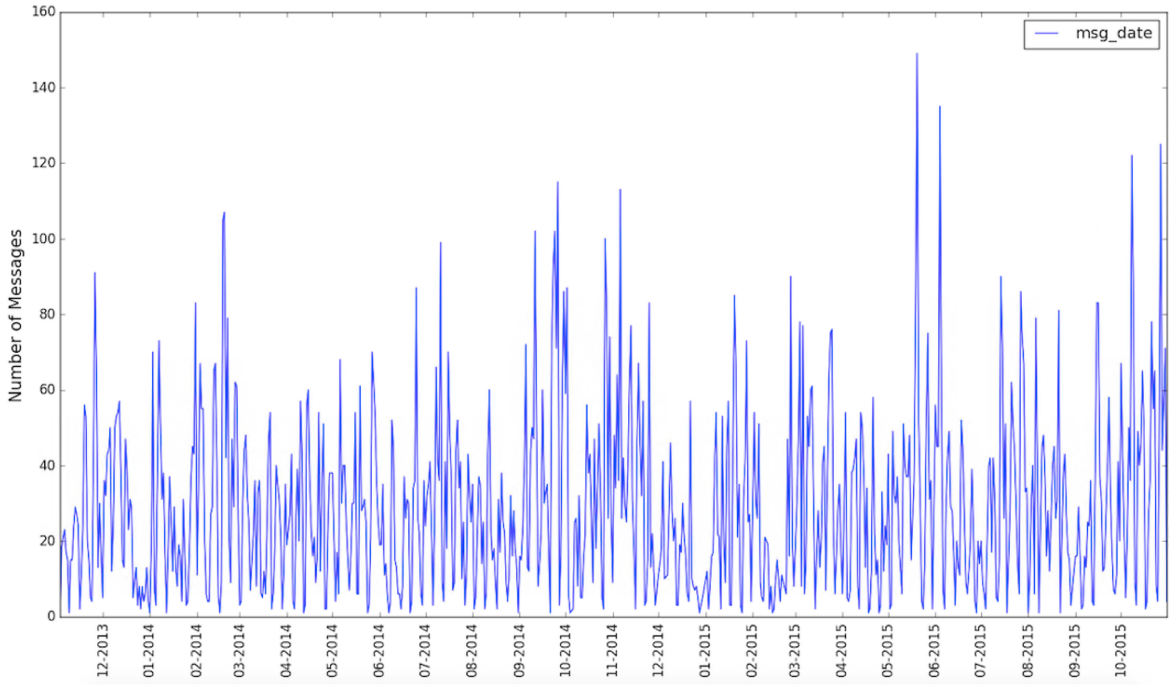

Figure 2. Messages sent to the PCI Linux kernel mailing list per day (12/2013–10/2015).

pared to a weekend day (see Figure 3), 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 hr and 3rd quartile if 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 characteristics of our empirical setting.

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 controlis 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, in our case the five unrealized control events are 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

Moving window

Some of the independent variables presented here 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.

Proximity variables

Proximity variables are presented in detail in the Research Design section of the article (see section 3.2). Here more details on the cosine similarity approach used when constructing *Cognitive proximity* are presented. *Cognitive proximity* is operationalized by determining similarity in contributions to different sections of the source code using a cosine similarity formula that 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

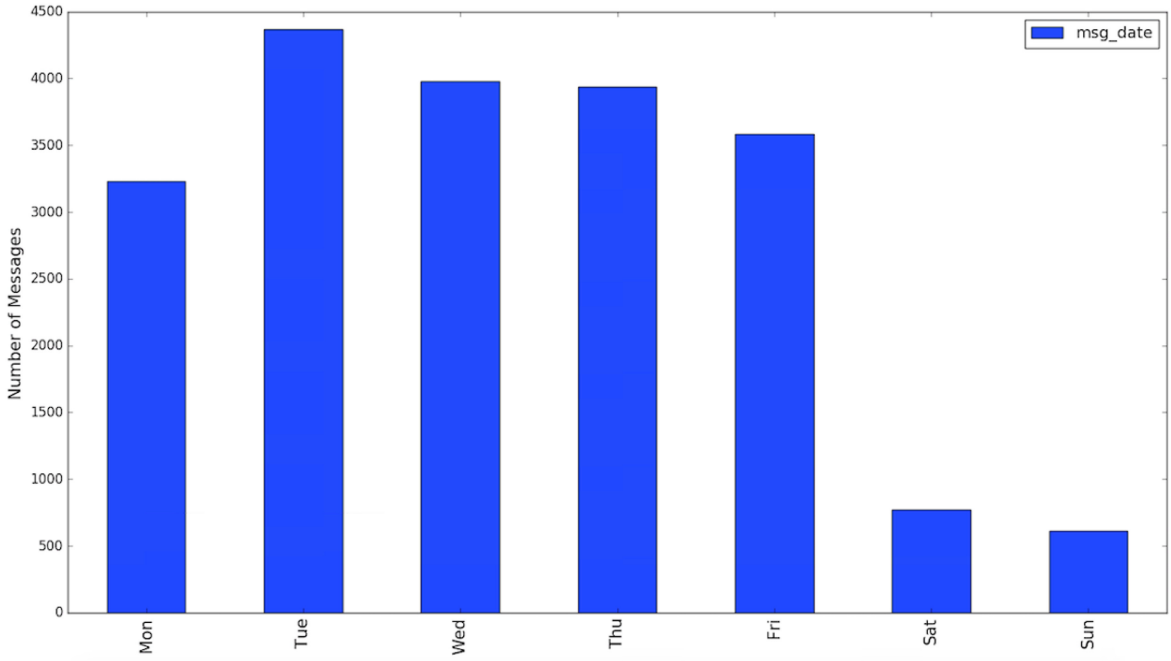

Figure 3. Messages sent to the PCI Linux kernel mailing list by day of week (12/2013–10/2015).

ego (A) and the alter (B) is divided by the product of the square root of sums squared for the ego and the alter.

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

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.

Structural control variables

Structural variables are presented in detail in the Research Design section of the article (see section 3.2). Here the calculations required for their construction is exemplified in Figure 4. Starting at the bottom, at time $t = 0$ is the realized collaboration event in the sequence of all realized events for which the structural variables are currently being calculated (for estimation purposes this procedure is repeated for all realized events in the dataset). This event at the bottom of the figure is our current 'target' – i.e., the reference point for the calculations required to construct the structural control variables.

In Figure 4 this target event represents a message sent by developer *e* (ego, blue circle) to developer *a* (alter, pink circle) and is shown an arrow going from ego to alter. The colors in the figure are used to help distinguish ego, alter, and other developers (*i*, *j*, *k* green circles). All the structural variables are calculated relative to this target event by going backwards and checking all of the collaboration events (i.e., mailing list replies) existing in the sequence during the past 63-days moving window we previously set. In Figure 4 the sequence is exemplified with 11 realized prior events (solid black circle outlines and arrows). Five sampled unrealized events for each realized event are also represented (dashed black circle outlines and arrow). For example, the event immediately prior in the sequence happens to be another reply from the current ego (*e*) to the current alter (*a*). Since this sequencing is captured by the *Repeated Collaboration* variable, its score is increased by one. If we keep going back we might find more events with the same characteristics in the sequence that would also contribute to the *Repeated Collaboration* variable. The third event prior happens to be instead a reply from the current alter (*a*) to the current ego (*e*). The event involves the same two developers, but the direction of the reply is reversed. This sequence is captured by the *Recency Effect* variable. Its score is set to 0.5 because alter (*a*) has sent two messages to other developers in between the *developer a to developer e, then developer e to developer a* sequence that the *Recency Effect* variable captures. Events involving other developers (*i, j, k*) are also used in the calculations of the two *Closure* and *Shared Collaboration Partners* variables.

In Figure 4 some events involving other developers (*i, j, k*) and the current ego and alter are included in the sequence for illustration. In Figure 5 the sequences that – if found – would add to the counts for the two *Closure* and *Shared Collaboration Partners* are presented separately for further clarification. The same target event used in the example illustrated in Figure 4 is at the bottom of the four sequences. The arrows joining other developers with ego and alter represents the events in the sequence that would contribute to each of the variables individually. For example, all events in the sequence prior to the target event where ego has messaged other developers that then have messaged alter (notice the direction of the arrows in Figure 5) would contribute to the calculation of the *Transitive Closure* structural control variable specifically.

Other control variables

The remaining control variables are briefly introduced in the Research Design section of the article (see section 3.2) and are presented in detail here. 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 REM, so the independent variables are focused on alter effects and dyadic covariates (Cassi and Plunket, 2015).

Three variables are used to take into account three factors specific to our empirical setting that may influence collaboration events. 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 (i.e., 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 being a maintainer. *Either Maintainer* is a dummy variable is set to 1 if the ego or the alter, or both 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

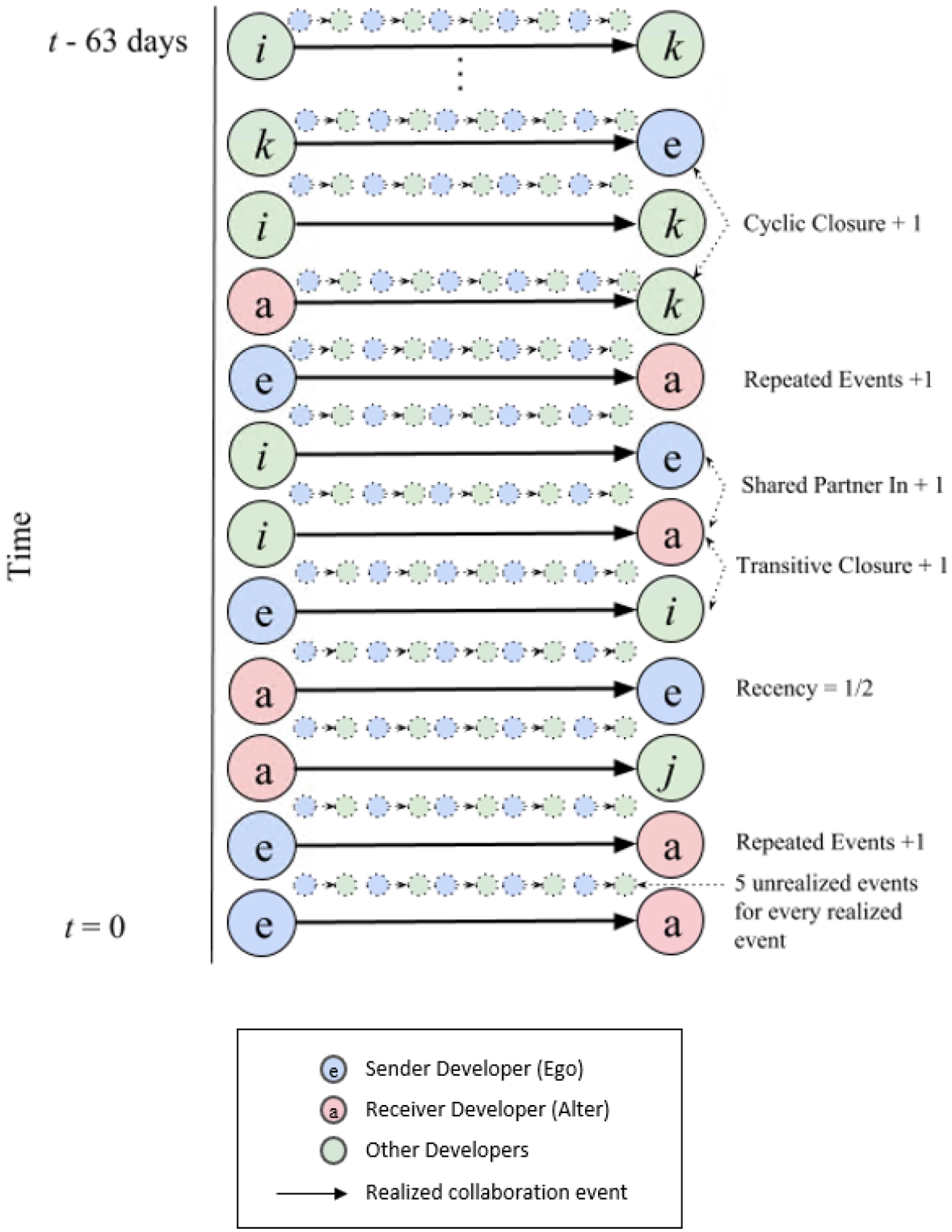

Figure 4. Structural variables calculations example.

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

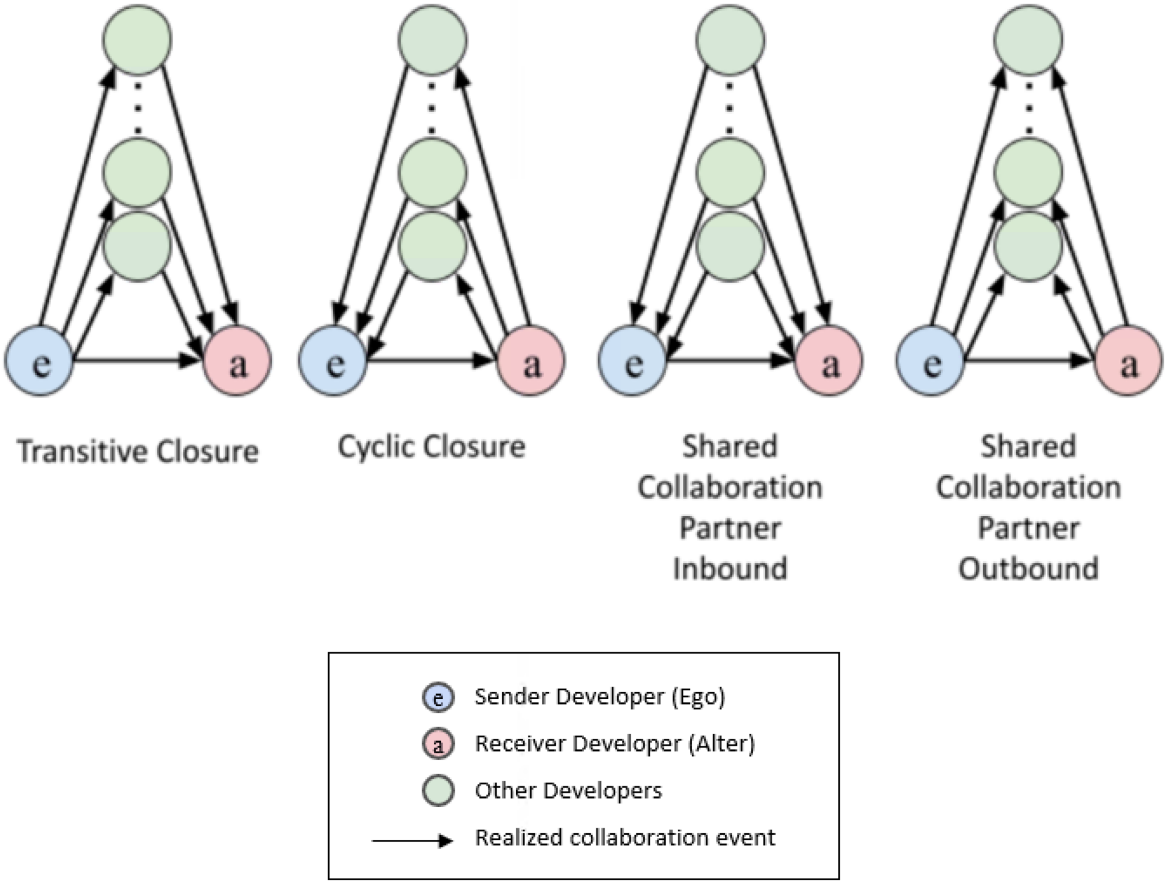

Figure 5. Triadic structural variables patterns illustration.

maintainer variables, a second variable measuring whether either the ego or the alter, or both have committed code can help in understanding the ego effect. *Either Committer* is a dummy variable set to 1 if the ego or the alter, or both 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, 2023). This is often done to get the attention of the maintainer when submitting Linux kernel patches. 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, 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 In Copy* 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 is otherwise set to 0.

Affiliation data cleaning

When testing for the effect of proximity dimensions on collaboration events we want to control for the structural embeddedness of the developers. In other words, we do not want structural factors influencing how they interact to be confounded with the effect of proximity dimensions. To do so we build structural control variables and adopt the REM framework. However, this modeling framework implicitly interprets the absence of an actor – or of an event between two actors – as meaningful information. Because of that it is important to keep missing or incomplete data about the developers to a minimum.

This includes affiliation data for which careful data cleaning was needed, because to allow for the calculation of proximity measures, it is assumed that a developer only has one employer affiliation at a time. After the initial data collection, approximately 22% of developers in the mailing list had overlapping affiliations. The vast majority of these overlaps were very straightforward to sort out. Two scenarios occurred. In some cases, one last reply was sent from the previous company's email address and the next one from the same developer was sent from the new company's email address. In this scenario the mid-point was picked as the 'job change date' when there was no email activity between the 2 dates. Since we use the sequence of events and not exact dates in the REM estimation,

exactly when the change appended between those two dates does not affect estimation at all. In other cases, the overlap was a fluke only due to a developer sending an email to the mailing list from their old account from the previous employer in between a series of emails from the new employer's account. Very likely this happened by mistake, for example using a computer where they were still log in with their old account early on in their new job. In our discussions with kernel developers this was confirmed informally to be a not uncommon occurrence. In those cases, the overlap is not real, and data were cleaned accordingly to reflect the correct affiliation of the developer at the time.

APPENDIX B

Interview guide

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 do not 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 cannot 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 did not think to ask about?