

Reconstructing Open Source Software Ecosystems: Finding Structure in Digital Traces

Research-in-Progress

Alexander Eck

University of St.Gallen
St.Gallen, Switzerland
alexander.eck@unisg.ch

Falk Uebernickel

University of St.Gallen
St.Gallen, Switzerland
falk.uebernickel@unisg.ch

Abstract

We report on the computational reconstruction of 273 open source software ecosystems, consisting of 41,388 artifacts and couplings between them, extracted from digital traces of 34.4 million software artifacts. We argue that digital traces are a new kind of data source, and propose ‘exploratory data loops’ to exploit the benefits of digital trace data in early stages of a research program. We apply this schema to systematically assess data quality, inform sample selection, and detect patterns. Empirically, we show that highly distributed networks are unlikely to follow a hierarchically modular structure, despite popular belief. As is shown visually with two examples, very distinct structures can emerge from autonomous behavior. The results indicate that different, yet similarly effective, strategies may exist to organize for distributed innovation in digital ecosystems. The paper is concluded by outlining how follow-up work will harness the reconstructed ecosystems for detecting behavioral patterns in distributed networks.

Keywords: doubly distributed networks, digital ecosystems, open source software, digital trace data, computational social science, data-driven research

Introduction

Software developers like to quip that their best programming skills include cleverly googling code snippets and stitching together frameworks of various sorts (e.g., Ali 2016; Haney 2016). As with most jokes, there is truth to be found. Driven by widespread success of the open source software model, designers of digital artifacts can freely draw upon a wealth of useful and diverse resources. As a result, individual artifacts are coupled with many other artifacts, thus creating complex *digital ecosystems* (Selander *et al.* 2013). It has been argued that digitized economies will increasingly produce innovations within highly distributed networks (von Hippel 2005; Yoo *et al.* 2010). Yet, empirical research that shows how distributed innovation plays out is scarce, as is research that investigates several ecosystems simultaneously (Manikas 2016). Arguably this is because digital ecosystems are heterogeneous and complex (Lyytinen *et al.* 2016), and thus their elements, structures, and behaviors are hard to identify and to describe. Therefore, to enable empirical research on distributed innovation, a systematic approach for identifying digital ecosystems is beneficial. This paper lays the foundations for an endeavor that aims to uncover innovation dynamics in distributed networks by asking:

How can we reconstruct open source software ecosystems from digital trace data?

Regarded individually, *digital artifacts*, or artifacts that are created, assembled, accessed or otherwise manipulated through digital technology (Kallinikos *et al.* 2013; Leonardi 2010), are already complex objects that are hard to manage and to study (cf. Brooks 1986). Investigating ecosystems of artifacts, and social actors related to them, pushes the limits of what can be achieved with traditional approaches (Ma and

Muelder 2013). But with challenges come opportunities. By definition, *open source software ecosystems* – digital ecosystems whose constituting artifacts are open source software artifacts – emerge and evolve in the public domain, because the source code of these artifacts is available to the public (Raymond 1999; von Hippel and von Krogh 2003). They also generate troves of highly granular data that information systems (IS) researchers can tap into (Agarwal *et al.* 2008). Consequently, computational approaches to make sense of *digital trace data* started to emerge in the IS discipline under the label of *computational social science* (e.g., Berente and Seidel 2015; Gaskin *et al.* 2014; Howison *et al.* 2011). They share the principles of moving between multiple levels of aggregation, and of iterating between exploratory and confirmatory phases.

Here we present the results of an exploratory phase, structured as follows. First, an overview of open source software ecosystems and distributed innovation is given. We identify a lack of research that investigates digital ecosystems of real-world scale, taking into consideration how the individual elements influence each other and how their interdependencies shape the respective ecosystems they inhabit. Next, by discussing the characteristics of digital trace data, we propose three *‘exploratory data loops’* for appropriating digital traces systematically in early stages of research. We then present a dataset of 144,311 open source software artifacts and relationships between them, which were reconstructed from over 58 million data points. We apply the proposed exploratory data loops to this dataset, which yields three main results. First, we find data quality to be sufficient for the intended use. Second, we select 273 open source software ecosystems out of the available population. Third, we present empirical evidence that highly distributed ecosystems are unlikely to have *small-world* structure (Watts and Strogatz 1998), despite the prevalence of said structure throughout many socially constructed complex networks (Albert and Barabási 2002). We briefly discuss the significance of these results, and conclude the paper by outlining how we plan to harness the presented dataset in further research.

Distributed Innovation in Open Source Software Ecosystems

Open Source Software Ecosystems

Rare is the software developer who sets out to implement her design with nothing more than a text editor and a compiler. Digital artifacts seldom come into existence and operate in isolation. Rather, they rely and depend on other digital artifacts for their functioning. For example, to implement a modern web application, designers marshal a multitude of resources over which they have no direct control, such as scripting languages and external data sources (Jazayeri 2007). Couplings among digital artifacts induce interference: as they are indefinitely malleable and thus amenable to change (cf. Kallinikos *et al.* 2013), digital artifacts co-evolve with their environment, causing further need for changes (Boland *et al.* 2007). Such coupled elements share the same ecosystem (Iansiti and Levien 2004). IS researchers regard an ecosystem as a socio-technical system comprising of interdependent digital artifacts and social actors (e.g., Selander *et al.* 2013). Hence, we define a *digital ecosystem* as a collection of digital artifacts that co-evolve through mutual interference, and the social actors related to these artifacts that are linked by a common interest (Eck and Uebernickel 2016; Selander *et al.* 2013).

The ecosystem metaphor carries notions of complexity and emergence, wherein distributed social actors and digital artifacts interact in non-trivial ways (e.g., El Sawy *et al.* 2010). Studying digital ecosystems raises the question of how to detect and trace structure and dynamics in light of a large number of constituting elements. Case study approaches (e.g., Boland *et al.* 2007) risk concerns of generalizability, whereas simulations (e.g., Woodard and Clemons 2014) might be perceived as overly stylized or reductionist. *Digital trace data*, or data generated by actors interacting with digital artifacts, are promising data sources for investigations that aim to match the complexity and size of the systems under study (Watts 2007). The widespread diffusion of open source software (Raymond 1999) and of open source development activities happening in the public domain (von Krogh and von Hippel 2006) has generated – and continues to generate – large amounts of digital trace data. Despite valid concerns not to jump on large data pools simply on the merit of accessibility (Rai 2016), *open source software ecosystems* are destined for studying complex digital ecosystems, precisely because they are so accessible to outsiders, and thus enable distributed innovation (Kogut and Metiu 2001).

In order to study open source software ecosystems, we must identify them first, that is reconstruct their elemental parts and the couplings between them from empirical data. While the majority of research has been confined to studying individual open source software artifacts (Crowston *et al.* 2012), there is some

existing IS research which pursued ecosystem reconstruction. Two distinct strategies emerged from this research. First, in the tradition of social network analysis (Wasserman *et al.* 2005), it is possible to reconstruct ecosystems based on activity of social actors. Two open source software artifacts are assumed to be related if at least one actor contributes to both artifacts, that is links between digital artifacts are established via joint affiliation of social actors (cf. Faust 2005). This approach lends itself to studying how people collaborate, whereas the co-evolution of digital artifacts is less of a concern (e.g., Grewal *et al.* 2006; Singh *et al.* 2011; Zhang *et al.* 2014). The second strategy captures known relationships between artifacts, based on a priori knowledge of the ecosystem structure. More specifically, a focal *digital platform* and its various complementing *modules* (cf. Tiwana *et al.* 2010) are selected for further investigation, like the ecosystem around Wordpress, a content management system (Um *et al.* 2015). While this approach is suitable to reconstruct an open source software ecosystem, it is restricted to the digital platform kind, which excludes more heterogeneous topographies (cf. Hanseth and Lyytinen 2010). Furthermore, it is a top-down approach, which is somewhat at odds with the complex and emergent nature of ecosystems (cf. Woodard and Clemons 2014). Recently, a bottom-up strategy has been established in discourses of software evolution (Mens 2008): ecosystem reconstruction based on design interdependencies, called *artifact couplings* in the context of this paper. Through inspection of source code (e.g., Lungu 2009) or of metadata associated with a digital artifact (e.g., Blincoe *et al.* 2015; Gonzalez-Barahona *et al.* 2009; Santana and Werner 2013), artifact couplings are systematically detected. In the research reported here, we apply this strategy to reconstruct open source software ecosystems.

Distributed Innovation

Yoo *et al.* (2008; 2010) and Lyytinen *et al.* (2016) theorize that most innovation in the digital realm will emanate from *doubly distributed networks* characterized by *distribution of resources* and *distribution of control*. The first dimension denotes that digital artifacts stem from disparate development trajectories, conceived by actors with varying capabilities. The second dimension refers to the absence of centralized control and authority. Thus, in doubly distributed networks, resources and activities are scattered across the ecosystem heterogeneously. A dynamic ensues wherein ecosystem components co-evolve, which creates opportunities for unbounded, distributed innovation. This seems to be a fair description of at least some open source software ecosystems, which makes them a reasonable fit for inquiries of distributed innovation (von Krogh and von Hippel 2006).

Despite some promising empirical results (e.g., Berente *et al.* 2008), the idea of doubly distributed networks remains underexplored. Instead, the concept of *generativity* has seen wider adoption among IS researchers keen to study *distributed innovation* in complex networks, or processes of creating innovative outcomes from collective behavior that is distributed along key dimensions (von Hippel 1988). While a number of different definitions for generativity exist, arguably the most salient one was formulated by Zittrain (2008: 70): “*a system’s capacity to produce unanticipated change through unfiltered contributions from broad and varied audiences*”. It is worth noticing how close this definition is to the notion of doubly distributed networks, as it outlines the very same characteristics of distributed innovation: first, an invitation to participate irrespective of actors’ individual capabilities (*broad and varied audiences*); second, a lack of central control or oversight (*unfiltered contributions*); and third, innovative outcomes from collective behavior (*unanticipated change*).

Generativity research in IS has been mostly conceptual to date (Eck and Uebernickel 2016; Tilson *et al.* 2010; Yoo 2013; Yoo *et al.* 2010). While a number of empirical contributions have adopted the generativity concept, it has usually not been their primary concern (e.g., Eaton *et al.* 2015; Ghazawneh and Henfridsson 2013; Wareham *et al.* 2014). To the best of our knowledge, Um *et al.* (2015) are the first to empirically examine generativity in an open source software ecosystem. The authors draw three main conclusions: First, they infer that a large ecosystem tends to evolve generatively. Second, they observe that ecosystem expansion does not predict innovation within the ecosystem. Third, they detect widely varying activity between the individual components; while a selected few are highly influential, many others are rather esoteric. These are fascinating results, yet they are derived from examining merely *one* open source software ecosystem. Clearly, an examination across *many* such ecosystems is required to acquire generalizable knowledge on how the unfiltered contributions of broad and varied audiences produce innovations. The research results presented here report on preparations for such a larger-scale empirical study, which draws upon digital trace data for empirical grounding.

Digital Trace Data as New Kind of Data Source

Digital Trace Data in Information Systems Research

Most events of people interacting with digital artifacts leave *digital trace data* (Agarwal *et al.* 2008; Howison *et al.* 2011). Data points usually cover intended outcomes (e.g., email message) as well as by-products of the performance (e.g., message size). Some traces are amenable to computational harvesting, which results in an extracted dataset. This dataset may contain traces of activity both in pristine form (e.g., email recipient) and in post-processed form (e.g., detected hyperlinks per message). We can transform this dataset into structured entities following a pre-defined model. For instance, we might want to create a graph of email senders and recipients, and the hyperlinks which were sent around. We can use this populated model to obtain measures, which are indicators for higher-level constructs. Following an approach of this kind, we might answer the question of how internet memes spread virally (e.g., Bauckhage 2011). An illustration of this process (Howison *et al.* 2011) is depicted with *Figure 1*. The bi-directional arrows indicate that the process can be either *construct-driven* or *data-driven*. In the first case, the avid researcher starts with a construct in mind, working her way backwards to specify which kind of digital trace data is required. In the second case, she starts exploring a given set of digital trace data in search for both expected and unexpected patterns (Grover and Lyytinen 2015).

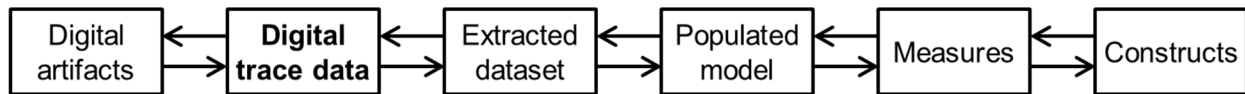


Figure 1. Generic Process to Harvest and Analyze Digital Trace Data

Digital trace data is attractive to researchers, because it differs from traditionally collected data such as observational or survey data in several ways. First, it is *found data*, that is it was not painstakingly created with a specific inquiry in mind (Howison *et al.* 2011). Second, it *captures actual behavior* which is not susceptible to respondent bias (Freelon 2014). Third, digital trace data is *granular*, which lets us extract, aggregate, and interpret it in varied novel ways (Latour 2010). Forth, *access* sets digital trace data apart from preceding attempts subsumed under the *knowledge discovery* label (cf. Chung and Gray 1999). From Github to Twitter and Wikipedia, the open source and social media movements produced large data pools outside of organizational walled gardens. It is not surprising, then, to see appropriation of digital trace data in IS research for studying varied topics, such as creation of coordination mechanisms in open source software development (Howison and Crowston 2014); emergence of routine structures in collaborative work (e.g., Lindberg 2015); and social network formation (e.g., Johnson *et al.* 2014) in online communities, to name a few. In line with Hedman *et al.* (2013) we expect empirical research based on digital trace data to gain further momentum in the years to come.

However, even with digital trace data there is no free lunch. Here, we highlight two fundamental issues that need to be dealt with. First, researchers are confronted with *ethical issues* of hitherto unknown magnitude (cf. Zimmer 2010). Due to the very nature of digital trace data, we can safely assume that people did not consent to any analyses unrelated to their immediate interaction with the digital artifact. Furthermore, due to the amount and granularity of available data, any compromise of personal rights will likely incur high damages. Second, digital trace data typically is *incomplete* along at least two dimensions (Bird *et al.* 2009). Digital artifacts are susceptible to change and unintended use, which makes full coverage, consistent data extraction, and correct interpretation challenging (cf. Howison and Crowston 2014). Moreover, the extraction procedure might not cover all potentially available digital trace data, for example due to technical limitations designed into the digital artifact that the researcher harnesses for extracting data. In light of both the promises and perils of digital trace data, we discuss the implications of incorporating such data in early stages of empirical research in the following.

Implication for IS: In Need of ‘Exploratory Data Loops’

Despite the opportunity to perform both construct-driven and data-driven research, IS scholars seemingly appropriated digital trace data largely as ‘yet another data source’ (cf. Hedman *et al.* 2013). With a few notable exceptions (e.g., Müller *et al.* 2016; Um *et al.* 2015), data-driven research is widely lacking, in the sense that it explores the entirety of an accessible body of digital trace data before narrowing down to a

small sample. Instead, it is common to filter out the majority of available traces a priori, typically based on theoretical considerations. Sampling follows concept-driven approaches, which in traditional empirical work have proven to serve us well. For example, in their study of how knowledge flows within open source software ecosystems, Singh *et al.* (2011) select a specific ecosystem a priori, thus reducing the number of individual software artifacts under consideration from potentially over 100,000 down to about 2,400 (Singh *et al.* 2011: 818). Sure enough, the authors are diligent in their methodology, and test whether their main design decisions are sensible, for instance by presenting some empirical support for their rationales.

We do not mean to imply that a concept-driven approach needs to lessen the methodological soundness, original contribution, and overall quality of such research. Our argument is rather that a substantial opportunity is missed if we do not reap the main advantage of digital trace data over traditionally collected data forms: namely the chance to engage with the empirical context, before diving into the thick of investigation (cf. Tukey 1980). To describe it more vividly, traditional approaches trained us to first decide on the frame and then fill the canvas, whereas the advent of digital traces asks us to explore the pre-filled canvas, challenges us to settle on the most interesting frame, and to continue from there. Digital trace data, in combination with data processing capabilities of typical computers available to researchers, afford a new way to interact with empirical material, particularly in the early stages of a research endeavor. In the following, we discuss three considerations, which leads us to propose three distinct **exploratory data loops**. Very much in the tradition of exploratory data analysis (Tukey 1977), the proposed exploratory data loops are intended to make the researcher familiar with an initially unfamiliar dataset. Their outcomes are descriptive, with subsequent theory building research designs (e.g., Berente and Seidel 2014; Gaskin et al. 2014; Howison and Crowston 2014) required to seek explanations for the phenomenon under study.

First, we should take notice of the possibility to **assess data quality** and gain a macro picture of the domain under study in the process (Naumann 2014). Obtaining an idea on how the entities of interest are distributed; which dimensions in the data correlate and which do not; how complete the dataset is; and generally getting a feeling for the data are crucial steps to assess data quality – after all, it is *found data* that shall be approached with due skepticism (Freelon 2014).

Second, we can explore data before settling on a specific sample, which **informs decision-making**. In empirical research, samples are drawn if studying the full population is not feasible. If the selection criteria are sound, a sample is representative of the whole and thus we can derive valid conclusions from it. Unlike survey or case study data, digital trace data exists before our research starts. Thus, there is little reason not to examine the general shape of it before settling on a sensible extraction protocol. This approach is also in the spirit of Bayesian statistics, which tells us that decisions tend to become better when we take more information into account that is available at a certain point in time (Taroni *et al.* 2014).

Third, and possibly most intricately linked to a researcher's curiosity, digital trace data let us explore and **detect regularities**, trends, and surprises (Grover and Lyytinen 2015). For example, Basole (2016) examines the topology of twelve inter-firm networks and detects systematic similarities among high-performing networks. Arguably, such data-driven, predominantly descriptive approaches are particularly useful when investigating very complex systems, whose many, multi-layered interactions we do not (yet) grasp. Grolemond and Wickham (2014) make the point that any researcher who has obtained a large collection of (digital trace) data is likely to follow a sense-making process that iteratively switches between exploratory and confirmatory procedures. Even more radically, Berente and Seidel (2015) recently questioned the integrity of research results based on digital trace data that make the impression to have been the outcome of a purely confirmatory research design.

Reconstructing Open Source Software Ecosystems from Digital Traces

This article is part of a larger research endeavor that aims to uncover how large open source software ecosystems emerge and change through co-evolution of their constituting elements, and how these dynamics lead to innovative outcomes. Open source software is an apt field for studying distributed innovation. It is known for promoting individual initiative and distributed governance, which in aggregate has been leading to outstanding outcomes (von Hippel and von Krogh 2003). Open source has seen widespread use, which makes the population of all open source software artifacts arguably going into many millions. To capture the unbounded nature of distributed innovation, we find it crucial to be as inclusive as possible, while still acquiring and processing empirical data efficiently. For data collection, we turned to

Github, which is not only a leading collaboration platform and repository for open source software (GitHub 2016d), but also grants access to digital traces via an API, or an application programming interface (GitHub 2016a). As endpoints can call the API only a limited number of times per hour, we mostly relied on the Ghtorrent dataset, which makes a large part of API-provided data available to researchers (Gousios 2013).

We regard an open source software ecosystem to be a set of interacting, coupled, and co-evolving open source software artifacts and associated social actors. In order to limit conceptual complexity in light of the vast amount of data we expected to collect, and in line with prior work (e.g., Carlile 2002; Um *et al.* 2015), we chose to identify ecosystem boundaries based on couplings between the artifacts, instead of couplings between both artifacts and actors. A common bottom-up approach to detect artifact interdependencies is through source code inspection (e.g., Lungu 2009). However, source code inspection does not scale well across a broad range of software artifacts. For example, we would need to devise different techniques for every programming language that we encounter (cf. Baldwin *et al.* 2014). To overcome this hurdle, we traded completeness on artifact level for coverage on population level. Specifically, we harnessed the Github discussion boards. On these boards developers discuss issues, exchange ideas for improvement, and so on. As part of their discussions, developers may reference external software artifacts following a certain syntax (GitHub 2016e). We exploited this feature to detect couplings uniformly across all open source software artifacts hosted on Github.

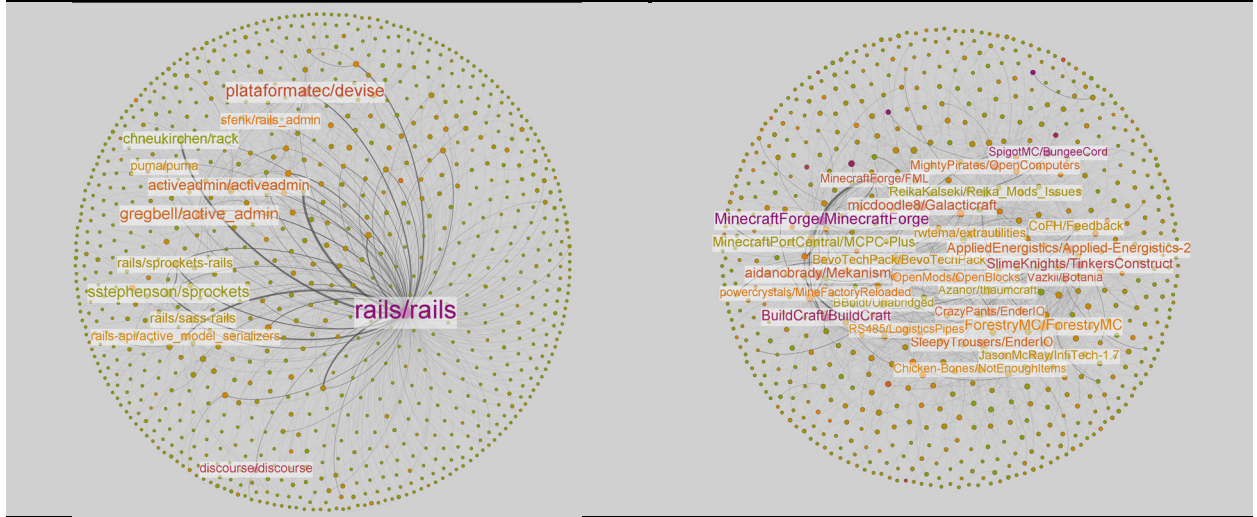
For example, on 05/05/2015 a developer commented with regard to a previously identified issue with ‘msysgit/git’: “It is fixed in curl master (not released yet), commit curl/curl@59f3f92” (GitHub 2016c). This developer referenced ‘curl/curl’, and the context of the discussion clearly shows that there is a coupling between the two artifacts. Blincoe *et al.* (2015) demonstrated that cross-references on Github discussion boards indeed reveal couplings between artifacts, which makes this approach methodologically sound. Conceptually, we argue that a discussion in which developers reference external software indicates that an artifact co-evolves with its environment. Therefore, only extracting artifacts for which discussion entries exist filters out those objects published on Github for purely archival purposes (cf. Kalliamvakou *et al.* 2015). Moreover, this approach also filters out couplings between artifacts which co-exist but do not co-evolve, that is artifacts that are connected technologically, but do not change over time or that change without mutual influence. For example, while a weather forecasting app might rely on an external data provision service, it might well be that the forecasting algorithm improves over time without the interface between the two artifacts ever changing. Lastly, given the well-defined syntax of how to reference external artifacts on Github discussion boards, couplings can be detected via automated pattern matching. This makes the chosen approach efficient, albeit not exhaustive. The data available to us spanned 8 years, from 10/04/2008 (the day Github became available to the public) until 31/03/2016, and encompassed 58,260,994 discussion board entries. The extraction technique yielded 544,979 couplings across 144,311 different open source software artifacts.

From this sample we created a network graph, with the software artifacts as its nodes and couplings between artifacts as its edges. It would be inaccurate to regard this graph as ‘the one’ open source software ecosystem, because of the variety of the open source software field: for instance, the ‘xbmc/xbmc’ media player hardly co-evolves with the data visualization library ‘mbostock/d3’. We assume that an individual ecosystem can be detected by localizing groups of artifacts which share many couplings within their group but few couplings outside of their group. For example, ‘xbmc/xbmc’ is coupled with 210 other artifacts in our sample and ‘mbostock/d3’ is coupled with 155 artifacts, but not with each other. The problem of detecting groups within a network is known as *community detection* in graph theory (Girvan and Newman 2002). We selected the well-performing *infomap* algorithm (Lancichinetti and Fortunato 2009; Rosvall and Bergstrom 2008) to partition the graph into groups. In line with methodical recommendations (Bohlin *et al.* 2014) the non-deterministic algorithm was repeated 500 times to increase the chance of obtaining a global optimum instead of a local one, which resulted in 22,609 individual groups. The largest group had 941 nodes, 14 groups had 500 or more nodes, 360 groups had 50 or more nodes, and 20,480 groups had 10 or fewer nodes. We interpret each group detected via the infomap algorithm as being an individual open source software ecosystem (cf. Blincoe *et al.* 2015).

¹ Artifacts on Github are uniquely identified with the syntax ‘*artifact_owner/artifact_name*’.

From Data Hairball to Surprising Pattern

This paper documents steps towards an empirical study on distributed innovation in open source software ecosystems. As discussed above, we argue that when drawing upon digital trace data, three exploratory data loops are mandated in early stages of research: to assess data quality; to inform sample selection; and to detect intriguing regularities. Following this sequence, we show how an initially overwhelming ‘*data hairball*’ (cf. Nocaj *et al.* 2014) was systematically analyzed to detect an unexpected regularity among doubly distributed ecosystems, which is illustrated in *Table 1* and explained in the following.



Graph layout:	ARF layout algorithm (Geipel 2007)		
Node/label color:	number of forks (logarithmically scaled gradient from green to purple)		
Node/label size:	number of couplings (logarithmic scaling)		
Edge width:	number of times coupling between two nodes was detected in digital trace data (logarithmic scaling)		
Label:	Name of artifact (shown if highly coupled, i.e. coupled with >5% of all artifacts in ecosystem)		
Software artifacts:	899	Software artifacts:	653
Couplings:	1,458	Couplings:	1,732
Highly coupled artifacts:	12	Highly coupled artifacts:	25
Total forks:	84,857	Total forks:	16,464
Scale-free correlation (r ²):	0.73	Scale-free correlation (r ²):	0.88

Table 1: Ecosystems ‘rails/rails’ (Left) and ‘MinecraftForge/MinecraftForge’ (Right)

Exploratory Data Loop I: Assessing Data Quality

We assessed data quality with a number of metrics and approaches, of which we highlight three. First, we verified that the available trace data was sufficiently exhaustive. For example, we counted over *34.4 million* different software artifacts in the dataset available to us, which is close to the *35 million* reported by Github (Github 2016d). Second, we must be confident in the extraction technique which we implemented with custom-written software. Following a comparable technique, Blincoe *et al.* (2015) report to have detected *18,533* artifacts coupled via *89,784* links in a dataset comprising *2.4 million* as of *May 2014*; that is, *0.8%* of all artifacts were coupled. In contrast, we extracted couplings between *0.4%* of all artifacts in the dataset. This is not surprising, given that socially constructed networks tend to grow faster in number of nodes than in number of edges between nodes (Johnson *et al.* 2014). In this context, it means that the population of artifacts increased faster than couplings between artifacts. Third, we could confirm that the constructed network graph topology was as expected. For example, we found evidence that the constructed graph is *scale-free*, which – among others – indicates that it reflects the outcome of complex social activity and is not purely random (Albert and Barabási 2002), which would have been the case if the underlying dataset were systematically flawed. Scale-freeness is defined as the *degree distribution* (in our case: couplings between artifacts) fitting a *power-law distribution* (Barabási and Albert 1999). We tested for scale-freeness as described in Kasthurirathna and Piraveenan (2015), and calculated $r^2=0.86$ for the whole graph, denoting a very strong correlation between a fitted power-law function and the empirical data.

Exploratory Data Loop II: Informing Sample Selection

In prior studies of open source software ecosystems, sample selection relied on rather coarse, a priori criteria such as identical programming language (Howison and Crowston 2014) or affiliation to a particular digital platform (Um *et al.* 2015). In contrast, owing to the many ecosystems reconstructed from digital trace data, we were able to apply more precise selection criteria. This research is guided by an interest to study distributed innovation. Thus, it is essential to select those ecosystems for inquiry that are likely to satisfy the conditions of a doubly distributed network, meaning that they are highly distributed both in resources and in control dimensions.

We operationalized *distribution of resources* as the number of artifacts in an open source software ecosystem. Hence, we assume that an ecosystem consisting of many coupled artifacts exhibits a heterogeneous distribution of resources overall, including knowledge (cf. Lyytinen *et al.* 2016). Moreover, we operationalized *distribution of control* as the sum over the number of *forks* per artifact in an ecosystem. On Github, anybody can ‘fork’ the source code of an open source software artifact, change this copy at will, and propose any change to be integrated with the master version of the source code (GitHub 2016b). Although the original artifact owner may choose to ignore change proposals, previous research identified forking as a mechanism that effectively diverts control into the periphery (e.g., Biazzi and Baudry 2014; Dabbish *et al.* 2012). Hence, we assume that control is distributed in an ecosystem whose constituting artifacts have many forks, and thus many prospects for outside contributions exist.

The 98th percentiles of each dimension were set as thresholds: We considered an ecosystem to be doubly distributed, if and only if it was composed of at least 42 software artifacts and 3,097 total forks. Applying the selection criteria yielded a sample of **273 open source software ecosystems**, representing **41,388 software artifacts** (median: 94 artifacts per ecosystem) and 3,596,202 total forks (median: 7,749 forks per ecosystem).

Exploratory Data Loop III: Detecting Intriguing Regularities

A large number of distributed networks, such as the world wide web and power grids (Solé *et al.* 2002), have been found to be *small-world networks* (Watts and Strogatz 1998). They are sparsely connected, yet information (or any other signal) spreads very fast due to *short-cut nodes* that sit in between otherwise unconnected groups of nodes; furthermore, many nodes are part of well-connected *local clusters* (Albert and Barabási 2002). Small-world networks can be thought as *hierarchically modularized systems*, wherein short-cut nodes (interfaces) connect local clusters (modules) to form other modules. Indeed, small-world structure has been found to be common in complex open source software systems (e.g., Myers 2003).

		doubly distributed		
		moderate	very high	Σ
small world	yes	N = 10 $r^2 = .74$	N = 4 $r^2 = .75$	14
	no	N = 27 $r^2 = .71$	N = 38 $r^2 = .69$	65
Σ		37	42	79

Table 2: Test Results for Small-World and Scale-Free Structure

Consequently, we expected to find many instances of small-world networks in our sample of 273 open source software ecosystems. We applied a test procedure similar to Telesford *et al.* (2011), and found only 67 small-world networks, or 25% of the sample. Even more unexpected, the sample shows that the most highly distributed networks are likely not small-world, as shown in Table 2 to the left. Out of those 42 ecosystems that lie within the 4th quartile of both resources distribution (≥ 163 artifacts per ecosystem) and control distribution ($\geq 14,378$ forks per ecosystem), only 4 (10%) passed the small-world test. In contrast, out of the 37 ecosystems occupying the 1st quartile of each distribution dimension, 10 (27%) ecosystems can be considered small-world. Because small-world networks tend to be scale-free (Valverde *et al.* 2002), we tested each ecosystem also for this property. The average values for r^2 are given in Table 2, and they corroborate the finding that hierarchically modular structure is not a common property of highly distributed ecosystems.

An illustration of the difference in structure is given with Table 1 above. Overall, this example hints at the possible existence of different patterns of how doubly distributed networks produce innovative outcomes. On the left-hand side, the ecosystem we call ‘rails/rails’ is shown, which is not small-world. It is clearly dominated by a single artifact, ‘rails/rails’, which is the Ruby on Rails framework for web development.

Judging by the depicted network graph, we may assume that this artifact is a *digital platform* on which other artifacts rely (cf. Tiwana *et al.* 2010). For instance, we find an artifact that extends Ruby on Rails with authentication capabilities (*devise*); one that adds a data administration interface (*activeadmin*); and an artifact which enables file compression (*sprockets*). The most highly coupled artifact in this ecosystem maintains links to 435 other artifacts, or 48% of the ecosystem population. In follow-up research it will be of interest to investigate to which extent the central artifact unilaterally influences its environment, or whether such influence is mutual, that is how co-evolution of a digital platform ecosystem topology plays out between the central digital platform and its surrounding modules. Research on digital platform ecosystem innovation dynamics is in its infancy, but suggests that both governance (Eaton *et al.* 2015) and functionality (Um *et al.* 2015) of the digital platform jointly shape these dynamics.

In contrast, the ‘*MinecraftForge/MinecraftForge*’ ecosystem, presented on the right-hand side of Table 1, is a small-world network. This ecosystem mostly consists of software artifacts that modify the popular video game *Minecraft*. Among others, we find an artifact (in jargon: a *mod*) that adds virtual oil pipelines (*BuildCraft*); an artifact that facilitates installation of other mods (*MinecraftForge*); and an artifact which connects separate instances of Minecraft servers (*BungeeCord*). The most highly coupled artifact maintains links to 105 other artifacts, or 16% of the ecosystem population. Visually, this ecosystem resembles a network which might have emerged through *wakes of innovation*, wherein autonomous activities in one part of the network lead to repercussions in other parts (cf. Boland *et al.* 2007). The absence of a digital platform as central reference point increases ecosystem complexity (cf. Hanseth and Lyytinen 2010), raising the question how social actors coordinate their disparate activities. Research on individual open source software artifacts suggests that coordination is managed by means of modularization (MacCormack *et al.* 2006), partitioning of tasks until they can be handled by a single person (Howison and Crowston 2014), and incremental as opposed to punctuated changes to the source code (Scacchi 2004). It will be of interest to investigate to which extent these results scale up from the artifact to the ecosystem level.

Instead of a Conclusion: Plans for Further Research

This research is guided by an interest to identify and explain the dynamics of autonomous behavior leading to distributed innovation in open source software ecosystems. This paper reports how a dataset of 273 ecosystems, amassing 41,388 software artifacts between them, was constructed from digital trace data captured on Github. We contribute to the body of literature on distributed innovation in digital ecosystems (Yoo *et al.* 2010) in two ways. Empirically, we show that highly distributed open source software ecosystems cannot be expected to be small-world; a surprising conclusion given previous results (Albert and Barabási 2002). Methodologically, we outline three *exploratory data loops*, which provide useful guidance on how to systematically approach an initially overwhelming set of digital trace data in early stages of research.

Going forward, we plan advances in four major areas. First, we intend to analyze sequences of activities in doubly distributed ecosystems, similar to the approach proposed by Lindberg *et al.* (2013), but on a considerably larger scale. We expect to find significant differences in complexity and variety of action sequences, depending on varying coordination requirements in ecosystems of different topology. Furthermore, we expect to discern diverse strategies that lead to comparable outcomes with regard to relevant success measures (cf. Gresov and Drazin 1997). Second, we intend to considerably refine the ecosystem graph model. Crucially, we plan to add the notions of growth and decay, for instance by modelling the addition/removal of software artifacts due to activity/inactivity (cf. Hanseth and Lyytinen 2010). Third, we aim to explicitly consider social actors and their interdependencies as part of an ecosystem (Woodard and Clemons 2014). In particular, we expect to find explanations for how social and technical subsystems co-evolve (Henfridsson *et al.* 2014). Fourth, we plan to make the dataset, as well as accompanying tools, available to the public at a later stage of the research program. In summary, we expect that further empirical analyses and more substantiated theorizing, supported by the dataset presented in this paper, will lead to considerably better-founded and more fine-grained conclusions than those we were able to present here.

Acknowledgements

The authors would like to thank the three anonymous reviewers for their helpful suggestions. Furthermore, the first author thanks Ola Henfridsson and the ISM Group at Warwick Business School for hosting a research visit during which a large portion of this paper was elaborated.

References

- Agarwal, R., Gupta, A. K., and Kraut, R. 2008. "The Interplay Between Digital and Social Networks," *Information Systems Research* (19:3), pp. 243–252.
- Albert, R., and Barabási, A.-L. 2002. "Statistical Mechanics of Complex Networks," *Reviews of Modern Physics* (74:1), pp. 47–97.
- Ali, T. 2016. *Essential Copying and Pasting From Stack Overflow*.
<https://www.gitbook.com/download/pdf/book/tra38/essential-copying-and-pasting-from-stack-overflow>. Accessed 5 August 2016.
- Baldwin, C., MacCormack, A., and Rusnak, J. 2014. "Hidden Structure: Using Network Methods to Map System Architecture," *Research Policy* (43:8), pp. 1381–1397.
- Barabási, A.-L., and Albert, R. 1999. "Emergence of Scaling in Random Networks," *Science* (286:5439), pp. 509–512.
- Basole, R. C. 2016. "Topological Analysis and Visualization of Interfirm collaboration networks in the electronics industry," *Decision Support Systems* (83), pp. 22–31.
- Bauchhage, C. 2011. "Insights into Internet Memes," (AAAI Conference on Weblogs and Social Media 2011 Proceedings), pp. 42–49.
- Berente, N., and Seidel, S. 2015. "Don't Pretend to Test Hypotheses: Lexicon and the Legitimization of Empirically-Grounded Computational Theory Development," University of Georgia, University of Liechtenstein.
- Berente, N., Srinivasan, N., Lyytinen, K., and Yoo, Y. 2008. "Design Principles for IT in Doubly Distributed Design Networks," *ICIS 2008 Proceedings*.
- Biazzini, M., and Baudry, B. 2014. "May the Fork Be With You: Novel Metrics to Analyze Collaboration on GitHub," *WETSoM 2014 Proceedings*, pp. 37–43.
- Bird, C., Bachmann, A., Aune, E., Duffy, J., Bernstein, A., Filkov, V., and Devanbu, P. 2009. "Fair and Balanced? Bias in Bug-fix Datasets," *ESEC/FSE 2009 Proceedings*, pp. 121–130.
- Blincoe, K., Harrison, F., and Damian, D. 2015. "Ecosystems in GitHub and a Method for Ecosystem Identification using Reference Coupling," *MSR 2015 Proceedings*, pp. 202–211.
- Bohlin, L., Edler, D., Lancichinetti, A., and Rosvall, M. 2014. "Community Detection and Visualization of Networks with the Map Equation Framework," in *Measuring Scholarly Impact: Methods and Practice*, Y. Ding, R. Rousseau, and D. Wolfram (eds.), Cham: Springer International Publishing, pp. 3–34.
- Boland, R. J., Lyytinen, K., and Yoo, Y. 2007. "Wakes of Innovation in Project Networks: The Case of Digital 3-D Representations in Architecture, Engineering, and Construction," *Organization Science* (18:4), pp. 631–647.
- Brooks, F. P. 1986. "No Silver Bullet: Essence and Accidents of Software Engineering," *Information Processing 1986 Proceedings*, pp. 1069–1076.
- Carlile, P. R. 2002. "A Pragmatic View of Knowledge and Boundaries: Boundary Objects in New Product Development," *Organization Science* (13:4), pp. 442–455.
- Chung, H. M., and Gray, P. 1999. "Special Section: Data Mining," *Journal of Management Information Systems* (16:1), pp. 11–16.
- Crowston, K., Wei, K., Howison, J., and Wiggins, A. 2012. "Free/Libre Open-Source Software Development: What We Know and What We Do not know," *ACM Computing Surveys (CSUR)* (44:2), Article 7.
- Dabbish, L., Stuart, C., Tsay, J., and Herbsleb, J. 2012. "Social Coding in GitHub: Transparency and Collaboration in an Open Software Repository," *CSCW 2012 Proceedings*, pp. 1277–1286.
- Eaton, B., Elaluf-Calderwood, S., Sørensen, C., and Yoo, Y. 2015. "Distributed Tuning of Boundary Resources: The Case of Apple's iOS Service System," *MIS Quarterly* (39:1), pp. 217–243.
- Eck, A., and Uebernickel, F. 2016. "Untangling Generativity: Two Perspectives on Unanticipated Change Produced by Diverse Actors," *ECIS 2016 Proceedings*.
- El Sawy, O. A., Malhotra, A., Park, Y., and Pavlou, P. A. 2010. "Seeking the Configurations of Digital Ecodynamics: It Takes Three to Tango," *Information Systems Research* (21:4), pp. 835–848.
- Faust, K. 2005. "Using Correspondence Analysis for Joint Displays of Affiliation Networks," in *Models and methods in social network analysis*, P. J. Carrington, J. Scott, and S. Wasserman (eds.), Cambridge: Cambridge University Press, pp. 117–147.

- Freelon, D. 2014. "On the Interpretation of Digital Trace Data in Communication and Social Computing Research," *Journal of Broadcasting & Electronic Media* (58:1), pp. 59–75.
- Gaskin, J., Berente, N., Lyytinen, K., and Yoo, Y. 2014. "Toward Generalizable Sociomaterial Inquiry: A Computational Approach for Zooming In and Out of Sociomaterial Routines," *MIS Quarterly* (38:3), p. 849.
- Geipel, M. M. 2007. "Self-Organization Applied to Dynamic Network Layout," *International Journal of Modern Physics C* (18:10), pp. 1537–1549.
- Ghazawneh, A., and Henfridsson, O. 2013. "Balancing Platform Control and External Contribution in Third-party Development: The Boundary Resources Model," *Information Systems Journal* (23:2), pp. 173–192.
- Girvan, M., and Newman, M. E. 2002. "Community Structure in Social and Biological Networks," *PNAS* (99:12), pp. 7821–7826.
- GitHub 2016a. *API*. <https://developer.github.com/v3/>. Accessed 5 August 2016.
- GitHub 2016b. *Bootcamp: Fork a Repo*. <https://help.github.com/articles/fork-a-repo/>. Accessed 5 August 2016.
- GitHub 2016c. *Msysgit/git: Push to Gerrit via Https is Broken #332*. <https://github.com/msysgit/git/issues/332#issuecomment-99059176>. Accessed 5 August 2016.
- GitHub 2016d. *Press Highlights*. <https://github.com/about/press>. Accessed 28 June 2016.
- GitHub 2016e. *Writing on GitHub: Autolinked References and URLs*. <https://help.github.com/articles/autolinked-references-and-urls/>. Accessed 5 August 2016.
- Gonzalez-Barahona, J. M., Robles, G., Michlmayr, M., Amor, J. J., and German, D. M. 2009. "Macro-level Software Evolution: A Case Study of a Large Software Compilation," *Empirical Software Engineering* (14:3), pp. 262–285.
- Gousios, G. 2013. "The GHTorrent Dataset and Tool Suite," *MSR 2013 Proceedings*, pp. 233–236.
- Gresov, C., and Drazin, R. 1997. "Equifinality: Functional Equivalence in Organization Design," *Academy of Management Review* (22:2), pp. 403–428.
- Grewal, R., Lilien, G. L., and Mallapragada, G. 2006. "Location, Location, Location: How Network Embeddedness Affects Project Success in Open Source Systems," *Management Science* (52:7), pp. 1043–1056.
- Grolemund, G., and Wickham, H. 2014. "A Cognitive Interpretation of Data Analysis," *International Statistical Review* (82:2), pp. 184–204.
- Grover, V., and Lyytinen, K. J. 2015. "New State of Play in Information Systems Research: The Push to the Edges," *MIS Quarterly* (39:2), pp. 271–296.
- Haney, D. 2016. *NPM & Left-Pad: Have We Forgotten How to Program?* http://www.haneycodes.net/npm-left-pad-have-we-forgotten-how-to-program/?utm_source=hackernewsletter&utm_medium=email&utm_term=featured. Accessed 5 August 2016.
- Hanseth, O., and Lyytinen, K. 2010. "Design Theory for Dynamic Complexity in Information Infrastructures: The Case of Building Internet," *Journal of Information Technology* (25:1), pp. 1–19.
- Hedman, J., Srinivasan, N., and Lindgren, R. 2013. "Digital Traces of Information Systems: Sociomateriality Made Researchable," *ICIS 2013 Proceedings*.
- Henfridsson, O., Mathiassen, L., and Svahn, F. 2014. "Managing Technological Change in the Digital Age: The Role of Architectural Frames," *Journal of Information Technology* (29:1), pp. 27–43.
- Howison, J., and Crowston, K. 2014. "Collaboration Through Open Superposition: A Theory of the Open Source Way," *MIS Quarterly* (38:1), pp. 29–50.
- Howison, J., Wiggins, A., and Crowston, K. 2011. "Validity Issues in the Use of Social Network Analysis with Digital Trace Data," *Journal of the Association for Information Systems* (12:12), pp. 767–797.
- Iansiti, M., and Levien, R. 2004. "Strategy as Ecology," *Harvard Business Review* (82:3), pp. 68–78.
- Jazayeri, M. 2007. "Some Trends in Web Application Development," *FOSE 2007 Proceedings*, pp. 199–213.
- Johnson, S., Faraj, S., and Kudaravalli, S. 2014. "Emergence of Power Laws in Online Communities: The Role of Social Mechanisms and Preferential Attachment," *MIS Quarterly* (38:3), pp. 795–808.
- Kalliamvakou, E., Gousios, G., Blincoe, K., Singer, L., German, D. M., and Damian, D. 2015. "An In-Depth Study of the Promises and Perils of Mining GitHub," *Empirical Software Engineering* (in Print).
- Kallinikos, J., Aaltonen, A., and Marton, A. 2013. "The Ambivalent Ontology of Digital Artifacts," *MIS Quarterly* (37:2), pp. 357–370.

- Kasthurirathna, D., and Piraveenan, M. 2015. "Emergence of Scale-Free Characteristics in Socio-Ecological Systems with Bounded Rationality," *Scientific Reports* (5), 10448.
- Kogut, B., and Metiu, A. 2001. "Open-Source Software Development and Distributed Innovation," *Oxford Review of Economic Policy* (17:2), pp. 248–264.
- Lancichinetti, A., and Fortunato, S. 2009. "Community Detection Algorithms: A Comparative Analysis," *Physical Review E* (80:5), 56117.
- Latour, B. 2010. "Tarde's Idea of Quantification," in *The Social after Gabriel Tarde: Debates and Assessments*, M. Candea (ed.), London, New York: Routledge, pp. 145–162.
- Leonardi, P. M. 2010. "Digital Materiality? How Artifacts Without Matter, Matter," *First Monday* (15:6).
- Lindberg, A. 2015. *The Origin, Evolution, and Variation of Routine Structures in Open Source Software Development: Three Mixed Computational-Qualitative Studies*, Cleveland, Ohio.
- Lindberg, A., Berente, N., Gaskin, J., Lyytinen, K., and Yoo, Y. 2013. "Computational Approaches for Analyzing Latent Social Structures in Open Source Organizing," *ICIS 2013 Proceedings*.
- Lungu, M. F. 2009. *Reverse Engineering Software Ecosystems*, Lugano.
- Lyytinen, K., Yoo, Y., and Boland, R. J. 2016. "Digital Product Innovation Within Four Classes of Innovation Networks," *Information Systems Journal* (25:1), 47–75.
- Ma, K.-L., and Muelder, C. W. 2013. "Large-Scale Graph Visualization and Analytics," *Computer* (46:7), pp. 39–46.
- MacCormack, A., Rusnak, J., and Baldwin, C. Y. 2006. "Exploring the Structure of Complex Software Designs: An Empirical Study of Open Source and Proprietary Code," *Management Science* (52:7), pp. 1015–1030.
- Manikas, K. 2016. "Revisiting Software Ecosystems Research: A Longitudinal Literature Study," *Journal of Systems and Software* (117), pp. 84–103.
- Mens, T. 2008. "Introduction and Roadmap: History and Challenges of Software Evolution," in *Software Evolution*, T. Mens, and S. Demeyer (eds.), Berlin, Heidelberg: Springer-Verlag, pp. 1–11.
- Müller, O., Junglas, I., vom Brocke, J., and Debortoli, J. 2016. "Utilizing Big Data Analytics for Information Systems Research: Challenges, Promises and Guidelines," *European Journal of Information Systems* (25:4), pp. 289–302.
- Myers, C. R. 2003. "Software Systems as Complex Networks: Structure, Function, and Evolvability of Software Collaboration Graphs," *Physical Review E* (68:4), 46116.
- Naumann, F. 2014. "Data Profiling Revisited," *ACM SIGMOD Record* (42:4), pp. 40–49.
- Nocaj, A., Ortmann, M., and Brandes, U. 2014. "Untangling Hairballs: From 3 to 14 Degrees of Separation," *Symposium on Graph Drawing 2014 Proceedings*, pp. 101–112.
- Rai, A. 2016. "Editor's Comments: Synergies Between Big Data and Theory," *MIS Quarterly* (40:2), pp. iii–ix.
- Raymond, E. 1999. "The Cathedral and the Bazaar," *Knowledge, Technology & Policy* (12:3), pp. 23–49.
- Rosvall, M., and Bergstrom, C. T. 2008. "Maps of Random Walks on Complex Networks Reveal Community Structure," *PNAS* (105:4), pp. 1118–1123.
- Santana, F. W., and Werner, C. M. L. 2013. "Towards the Analysis of Software Projects Dependencies: An Exploratory Visual Study of Software Ecosystems," *IWSECO 2013 Proceedings*, pp. 1–12.
- Scacchi, W. 2004. "Free and Open Source Development Practices in the Game Community," *IEEE Software* (21:1), pp. 59–66.
- Selander, L., Henfridsson, O., and Svahn, F. 2013. "Capability Search and Redeem Across Digital Ecosystems," *Journal of Information Technology* (28:3), pp. 183–197.
- Singh, P. V., Tan, Y., and Mookerjee, V. 2011. "Network Effects: The Influence of Structural Capital on Open Source Project Success," *MIS Quarterly* (35:4), pp. 813–830.
- Solé, R. V., Ferrer-Cancho, R., Montoya, J. M., and Valverde, S. 2002. "Selection, Tinkering, and Emergence in Complex Networks," *Complexity* (8:1), pp. 20–33.
- Taroni, F., Biedermann, A., Bozza, S., Garbolino, P., and Aitken, C. 2014. *Bayesian Networks for Probabilistic Inference and Decision Analysis in Forensic Science*, Hoboken: Wiley.
- Telesford, Q. K., Joyce, K. E., Hayasaka, S., Burdette, J. H., and Laurienti, P. J. 2011. "The Ubiquity of Small-World Networks," *Brain Connectivity* (1:5), pp. 367–375.
- Tilson, D., Lyytinen, K., and Sørensen, C. 2010. "Digital Infrastructures: The Missing IS Research Agenda," *Information Systems Research* (21:4), pp. 748–759.
- Tiwana, A., Konsynski, B., and Bush, A. A. 2010. "Platform Evolution: Coevolution of Platform Architecture, Governance, and Environmental Dynamics," *Information Systems Research* (21:4), pp. 675–687.

- Tukey, J. W. 1980. "We Need Both Exploratory and Confirmatory," *The American Statistician* (34:1), pp. 23–25.
- Um, S., Yoo, Y., and Wattal, S. 2015. "The Evolution of Digital Ecosystems: A Case of WordPress from 2004 to 2014," *ICIS 2015 Proceedings*.
- Valverde, S., Cancho, R. F., and Solé, R. V. 2002. "Scale-Free Networks from Optimal Design," *Europhysics Letters* (60:4), pp. 512–517.
- von Hippel, E. 1988. *The Sources of Innovation*, New York NY u.a.: Oxford Univ. Press.
- von Hippel, E. 2005. *Democratizing Innovation*, Cambridge, Mass.: MIT Press.
- von Hippel, E., and von Krogh, G. 2003. "Open Source Software and the "Private-Collective" Innovation Model: Issues for Organization Science," *Organization Science* (14:2), pp. 209–223.
- von Krogh, G., and von Hippel, E. 2006. "The Promise of Research on Open Source Software," *Management Science* (52:7), pp. 975–983.
- Wareham, J., Fox, P. B., and Cano Giner, J. L. 2014. "Technology Ecosystem Governance," *Organization Science* (25:4), pp. 1195–1215.
- Wasserman, S., Scott, J., and Carrington, P. J. 2005. "Introduction," in *Models and methods in social network analysis*, P. J. Carrington, J. Scott, and S. Wasserman (eds.), Cambridge: Cambridge University Press, pp. 1–7.
- Watts, D. J. 2007. "A Twenty-First Century Science," *Nature* (445:7127), p. 489.
- Watts, D. J., and Strogatz, S. H. 1998. "Collective Dynamics of 'Small-World' Networks," *Nature* (393:6684), pp. 440–442.
- Woodard, C. J., and Clemons, E. K. 2014. "Modeling the Evolution of Generativity and the Emergence of Digital Ecosystems," *ICIS 2014 Proceedings*.
- Yoo, Y. 2013. "The Tables Have Turned: How Can the Information Systems Field Contribute to Technology and Innovation Management Research?" *Journal of the Association for Information Systems* (14:5), pp. 227–236.
- Yoo, Y., Henfridsson, O., and Lyytinen, K. 2010. "The New Organizing Logic of Digital Innovation: An Agenda for Information Systems Research," *Information Systems Research* (21:4), pp. 724–735.
- Yoo, Y., Lyytinen, K., and Jr., R. J. B. 2008. "Distributed Innovation in Classes of Networks," *HICSS 2008 Proceedings*.
- Zhang, Z., Yoo, Y., Wattal, S., Zhang, B., and Kulathinal, R. 2014. "Generative Diffusion of Innovations and Knowledge Networks in Open Source Projects," *ICIS 2014 Proceedings*.
- Zimmer, M. 2010. "But the Data is Already Public!: On the Ethics of Research in Facebook," *Ethics and Information Technology* (12:4), pp. 313–325.
- Zittrain, J. L. 2008. *The Future of the Internet and How to Stop It*, New Haven, CT: Yale University Press.