

GENERATIVE DIFFUSION OF INNOVATIONS: AN ORGANIZATIONAL GENETICS APPROACH¹

Research-in-Progress

Zhewei Zhang

Temple University
Philadelphia, PA, USA
zhang@temple.edu

Rob Kulathinal

Temple University
Philadelphia, PA, USA
robkulathinal@temple.edu

Sunil Wattal

Temple University
Philadelphia, PA, USA
swattal@temple.edu

Youngjin Yoo

Temple University
Philadelphia, PA, USA
youngjin.yoo@temple.edu

Abstract

Innovation in open ecosystems such as open source software is characterized by generative diffusion, the property of such ecosystems to evolve and change over time through the actions of uncoordinated participants. In this research, we contend that existing models of diffusion are not adequate to capture the multi-faceted nature of generative diffusion. To address this challenge, we use concepts from biological sciences to propose a multi-dimensional perspective to study generative diffusion, and construct three metrics: proliferation, evolvability, and temporality. Further, we use techniques inspired by genetics to measure these constructs in the context of open source software. In this research in progress manuscript, we demonstrate the applicability of our work with one example of an open source software project. This study makes an immense contribution not only to the study of open innovation, but also makes a methodological contribution by introducing the use of evolutionary genetics to study digital artifacts.

Keywords: Diffusion of innovation, Open source software development, Phylogenetic analysis, Generativity

¹ This paper is based on work supported, in part, by the National Science Foundation under grants VOSS-0943010 and VOSS-1120966. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Introduction

Open innovation has emerged as an important way to facilitate innovations in organizations. Chesbrough et al (2006) elegantly summarizes the definition of open innovation as follows: “Open innovation is a paradigm which assumes that firms can and should use external ideas as well as internal ideas, and internal and external paths to market, as they look to advance their technology”. Open source communities have emerged as an informal mechanism to facilitate knowledge exchange and rapid diffusion of innovations across conventional organizational boundaries (Chesbrough, Vanhaverbeke, & West, 2006). Furthermore, open innovation allows heterogeneous actors to come together and exchange their ideas, allowing more diverse ideas to emerge (von Hippel, 2005; Yoo, Boland, & Lyytinen, 2008). Open innovation supported by digital infrastructure can facilitate generative and dynamic evolutions of innovations (Benkler, 2006; Tuomi, 2002; Zittrain, 2006). In a way, an open innovation community serves as a “trading zone” (Boland, Lyytinen, & Yoo, 2007; Galison, 1997) that brings together actors with diverse ideas and mashing these ideas to generate novel innovations.

A key characteristic of open innovations, which has not received adequate attention in literature is *generative diffusion*. Generativity refers to an “overall capacity [of technology] to produce unprompted change driven by a large, varied, and uncoordinated audience” (Zittrain 2006). Generative diffusion refers to the ability of open innovations to continue to evolve and change over time through the participation of these uncoordinated participants. Boland et al. (2007) use the image of “wakes of innovations” to capture the ever-changing landscape of the innovations as innovations from heterogeneous communities collide with one another. This produces highly volatile and dynamic patterns of innovations. This type of generative diffusion of innovations is different from the more conventional model of innovation diffusion (Rogers, 1995), where there is a singular inventor whose inventions are diffused and adopted by passive consumers. Generative diffusion is also multi-dimensional because not only is the product adopted, but also modified by a number of uncoordinated participants. Although previous studies have demonstrated that users adopt and change in such contexts (Von Hippel, 1986), the implicit assumption that there is a core idea of innovation remains unchanged.

To address this important gap in the literature, we propose a novel way of measuring the generative diffusion of innovations drawing on analytical methods inspired by the field of evolutionary genetics. We focus our research on open source software communities, the most well-established form of open innovation. Specifically, in this research, we ask the following questions:

1. How can we identify the pattern of innovation diffusing in the open source development context?
2. How can we measure the multiple dimensions of diffusion in case of generative innovations??

The paper is organized into the following sections: Section 2 briefly reviews some prior studies in generativity, diffusion of innovation, and open source development and identifies the research gap in these areas. In Section 3 we propose to understand the diffusion of innovation as a multidimensional concept. Section 4 presents our research context and proposed analysis method inspired from genetic research, followed by discussion on our illustrative analysis result in section 5. Finally, we discuss our goals and future plans, and discuss the various implications of this study.

Literature Review

Previous studies on open source projects have carefully examined the motivations and contributions of the open source community (Lakhani and Wolf, 2003). Recent research on software code reuse among open source projects has studied the reasons for reuse (Haefliger et al 2007), patterns of reuse (Mockus 2007), as well as interactions between adoption and invention (King and Lakhani, 2011). However, existing research does not discuss how open source software evolves over time, and how one project can contribute resources such as code base to subsequent projects. To understand these issues, we borrow from the emerging literature on the generativity of digital artifacts.

Generative digital products are rapidly shaping our world into a ubiquitous computing world. Previous work has discussed the properties of generativity (Zittrain, 2006) and characteristics of digital product which sustain the generativity (Yoo et al. 2010). Those properties and characteristics cannot directly

create generativity by themselves. Rather, they provide a solid basis for the effective and efficient diffusion of innovations into a digital format, generating such strong generativity in digital products.

Diffusion of innovation is initially studied in sociology and communication to understand how ideas and technologies spread. Rogers, in his influential books (1962, 1971), explained how innovation is communicated in a social system with an S-curve. Wejnert (2002) defined the characteristics of innovation, innovator, and environment in his classic diffusion framework. In Information Systems literature, much attention has been placed on technology adoption, particularly on the antecedents and consequences of adoption (Karahanna et al. 1999; Plouffe et al. 2001; Agarwal and Prasad 1998, Venkatesh et al 2004). However, many previous studies consider innovation as an end-product: adoption, as the result of diffusion, is usually operationalized as a binary variable. In contrast, generative diffusion is an ongoing process during which new innovations may emerge. It is diffusion with adoption as well as mutation.

What is particularly lacking is not only the way an idea has spread and diffused, but also how that idea mutates and changes over time. The propagation of ideas through mutation and recombination finds parallels in the biological sciences, where basic genetic building blocks (genotype) can explain the complexity of living organisms. This sentiment is echoed by Kelly (2010) who suggests that digital artifacts evolve through both chance and inevitable ways, just as living organisms do. In fact, linking the origin and spread of ideas to models that evolutionary geneticists have used for the last century or so can provide the organizational sciences with a new approach in characterizing how innovations disperse. Novel ideas can be viewed as novel mutations and their spread across different platforms can be tracked through time. Thus, the entire emergence of new ecosystems of innovative ideas can be seen as a series of mutational events over time.

Multi-dimensional view of Generative Diffusion of Innovations

In contrast to prior diffusion models (Bass, 1969), we will conceptualize our generative diffusion of innovations via a multi-dimensional construct. We develop three dimensions based on our understanding of the world's simplest yet most generative object-DNA. With only four types of nucleoside, numerous species emerge. To this extent, life is the largest diffusion of innovation. Below, we explain three dimensions inspired by genetics in detail and propose ways to measure each of these dimensions.

Proliferation

The first dimension of the diffusion of innovations is the *proliferation*. This dimension is similar to the traditional definition of technology adoption. It mainly concerns whether an innovation has been adopted by other people. In our study, we measure proliferation by the number of people who adopt the innovation. In biology, proliferation of DNA makes reproduction possible, and in our case, it makes spread of innovation possible.

In other words, proliferation represents the total number of innovation adopters. In the context of open source project development, this can be considered as the number of developers who have adopted a project and use it as a separate project-their own project.

Evolvability

The second dimension is the *evolvability*. We argue that diffusion does not simply stop after an innovation is adopted, therefore proliferation (number of adopters), by itself, is not sufficient to measure diffusion. In fact, diffusion will continue as the innovation has been absorbed and mutate after the adoption. Evolvability concerns the degree that adopted ideas can mutate. The more an adopted innovation mutates, the more evolvability diffusion has. Apparently, mutation may not necessarily always be good. A mutation in DNA can create bad trait to organism, and mutation of innovation can also bring bad consequences. However, only by mutation will various new opportunities come out. In the context of open source project development, evolvability refers to the further modification/improvement of project after it been adopted and reused by other developers.

Temporality

Temporality presents a third dimension. Temporality refers to the fact that diffusion is not a static state but rather, it is an ever-changing process. In other words, the pattern of diffusion in terms of proliferation and evolvability may change over time. Novel and immature ideas may have different diffusion patterns from mature and stable ideas. In software development, temporality can be seen from the number of different versions of open source project that have been adopted and further developed.

Degree and Speed of Diffusion

Taking the three dimensions together, we are able to understand the diffusion of innovation with a more comprehensive view by proposing the following composite measures of diffusion.

- The *degree of diffusion* measures the extent to which an innovation is generatively diffused for a time period. It is a function of proliferation and evolvability: the sum of degrees of mutation for each adoption. In open source development, it is calculated as a function of the number of people adopting the open source software as well as the degree of modifications after the adoption.

$$\text{Degree of diffusion} = f(\text{Proliferation, Evolvability})$$

- The *speed of diffusion* is the rate at which an innovation is generatively diffused. In our context, it represents the speed of one open source software diffused. It is calculated as the degree of diffusion divided by time:

$$\text{Speed of diffusion} = f(\text{Degree, Time})$$

Measuring the Generative Diffusion of Innovations: An Organizational Genetics Approach

Data Description: Open Source Projects on Github

We collected our data from Github, one of the largest online hosting repositories for programming code. Git is a widely used distributed revision control and source code management system. It allows users and developers to track every single modification of their software source code, providing a perfect research context to our understanding of how innovations diffuse over time. In addition to tracking the project's own source codes, Github also allow developers to easily “fork” other developers' projects. Fork is an action in Git, which basically means one developer can clone or save one existing project owned by other developer into his/her own repository so that he/she can use and modify it as a separate project. For each project, Github provides the family tree which shows every forked project of that project. In the following sections, we call the original project as the *master* project and forked projects as *branch* projects. In the family tree, all branch projects are ordered by the time that they were created. All generations of projects (each branch) are listed in this tree if it exists.

Since every single modification is tracked in Git, we are able to get one version of source code for each change. However, to simplify our work in this preliminary study, we only use source codes of each major release for master projects and the latest version for branch projects. By focusing on the major releases of master projects, we are able to capture the major changes from the master project as well as the time frame while keep the study manageable. By focusing on branch projects' latest versions, we are able to identify new developments after forking.

Phylogenetic analysis: an Organizational Genetics Approach

As our conceptualization of generative diffusion of innovation is inspired by genetics, we will also employ an organization genetics approach (Gaskin et al, 2011; Yoo, 2012; Zhang et al, 2012) in order to study the generative pattern of digital innovations. The key of this approach is a cross-level analysis with a focus on the complex interactions between the system and its underlying “genetic” elements using analysis techniques from genetics. Such approach will help us understand the complexity of the phenotype (which is the distinguishing characteristic of living organisms) via its genotype (which is the basic genetic building blocks of the artifact). In this study, the code itself represents the genetic building blocks, or the

software DNA. The code from different versions or branches can then be aligned and analyzed in a comparative manner.

This study uses common evolutionary genetic techniques, including sequencing and alignment as well phylogenetic analysis, to better understand our data. To our knowledge, this is the first time that a phylogenetic approach has been used for analyzing digital products in business research. We have chosen this approach because of the similarity between raw source code and DNA sequences. Both of them are basically a string of sequential elements: we consider each letter, number and symbol written in source code equivalent to nucleotides in DNA sequence.

The first step is to align each source code to generate an alignment score as well as alignment matrix. The purpose of the alignment is to standardize the source code across different projects for easy comparison. Alignments are performed using a software called ClustalTXY, which is a derivative software from Clustal, a commonly used alignment software among genetic researchers, but adapted for social science studies. This software performs pairwise alignment of sequences and constructs a pairwise similarity matrix. The result of the alignment analysis provides a foundation to analyze the differences and relationship between sequences. Figure 2 shows the alignment results for a short segment of code. It shows a small portion of this segment is different: only the first five elements are different between top 9 sequences and the bottom 6 sequences.

```

PPORT TRANSITION-END-REMOVEELEMENT--REMOVEELEMENT-
PPORT TRANSITION-END-REMOVEELEMENT--REMOVEELEMENT-
PPORT TRANSITION-END-REMOVEELEMENT--REMOVEELEMENT-
PPORT TRANSITION-END-REMOVEELEMENT--REMOVEELEMENT-
PPORT TRANSITION-END-REMOVEELEMENT--REMOVEELEMENT-
PPORT TRANSITION-END-REMOVEELEMENT--REMOVEELEMENT-
PPORT TRANSITION-END-REMOVEELEMENT--REMOVEELEMENT-
PPORT TRANSITION-END-REMOVEELEMENT--REMOVEELEMENT-
PPORT TRANSITION-END-REMOVEELEMENT--REMOVEELEMENT-
-BIND TRANSITION-END-REMOVEELEMENT--REMOVEELEMENT-
-BIND TRANSITION-END-REMOVEELEMENT--REMOVEELEMENT-
-BIND TRANSITION-END-REMOVEELEMENT--REMOVEELEMENT-
-BIND TRANSITION-END-REMOVEELEMENT--REMOVEELEMENT-
-BIND TRANSITION-END-REMOVEELEMENT--REMOVEELEMENT-
-BIND TRANSITION-END-REMOVEELEMENT--REMOVEELEMENT-
-BIND TRANSITION-END-REMOVEELEMENT--REMOVEELEMENT-
::: ***** ### ***** *****

```

Figure 2. Example of alignment results in one section of code.

After generating the alignment results, ClustalTXY also provide phylogenetic analysis result in newick format, a special file format containing information regarding to branch length and relationship. Phylogenetic analysis result contains information regarding to the relationship among branches. We then import the results into visualization software, which we used Treeview here, another popular software to generate phylogenetic tree. Phylogenetic trees allow us to infer the relationships of each sequence relative each other, where each sequence represents a taxon (e.g., a species in biology) or a version code (this project). A typical phylogenetic tree (shown in Figure 3) has a number of common characteristics. An internal node represents a divergence event in a common ancestor: a single ancestral taxon (or coded sequence) splits or bifurcates at each internal node. External nodes (leafs on the tree) refer to taxa (biology) or a version of source code (this project). The root is a special node, referring to oldest extant node among all taxa in the tree: in biology, it represents the relative to the common ancestor of all taxa from the tree. In this study, it represents the earliest source code. Branches depict the evolutionary distance between two taxa or differences between two versions of source code. Distance scale provides a reference to understand the magnitude of the differences between sequences. For example, a branch length of 0.004 in the example tree means a 0.4% difference in the sequence between two given nodes (difference between n1 and n3 in figure 3 is roughly 0.8%²). Last, the overall structure of tree, or the topology, offers a visual representation of all the taxa or versions.

² There are three branches (as numbered 1,2 and 3 in figure 3) between node n1 and n3

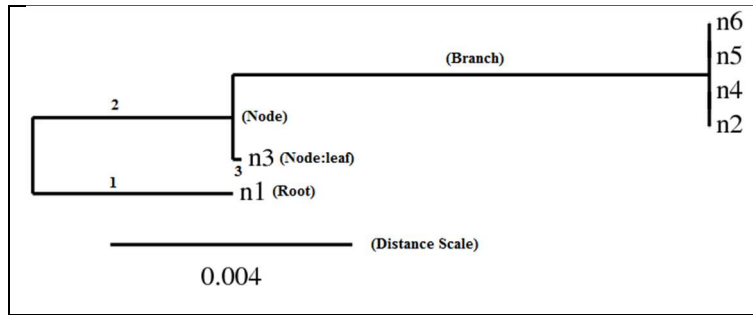


Figure 3. Notations used in phylogenetic tree.

In this example, if n2 to n6 are five different versions of files, all originated from n1. It tells us version n3 is most similar to n1, although it was branched after n2. Version n2, n4, n5, n6 are almost the same, suggesting they have similar changes after branching. The difference between n1 and n3 is 0.8%, suggesting there is 0.8% source code has been changed in n3 when compared with n1.

We believe this method is the best way to answer our research questions. First, in our study, we will perform such analysis at the scale of hundreds of branches across multiple months or even years. In doing so, we can create a tree based their similarity and differences with time information. The overall structure of tree, or the topology, offers a visual representation of all the taxa or versions. Different structures represent different evolution patterns (Kirkpatrick and Slatkin, 1993) as shown in Figure 4. Specifically, from left to right in figure 4, the evolution patterns switch from asymmetric (gradual evolution) to symmetric (punctuated evolution). A gradual evolution pattern suggests differences among branches come out gradually as time goes by whereas a punctuated evolution pattern suggests differences among branches come out in a very short time frame at certain point. Further analysis on such point may reveal new insight on why such phenomenon may occur. Secondly, not only can we create such qualitative tree but also we can calculate quantitatively in terms of percentage change (degree of diffusion) and percentage change in certain time (speed of diffusion) to assess how idea is adopted and modified and diffused. The quantitative measurement then provides us foundation to perform further analysis on how diffusion may be affected by other factors.

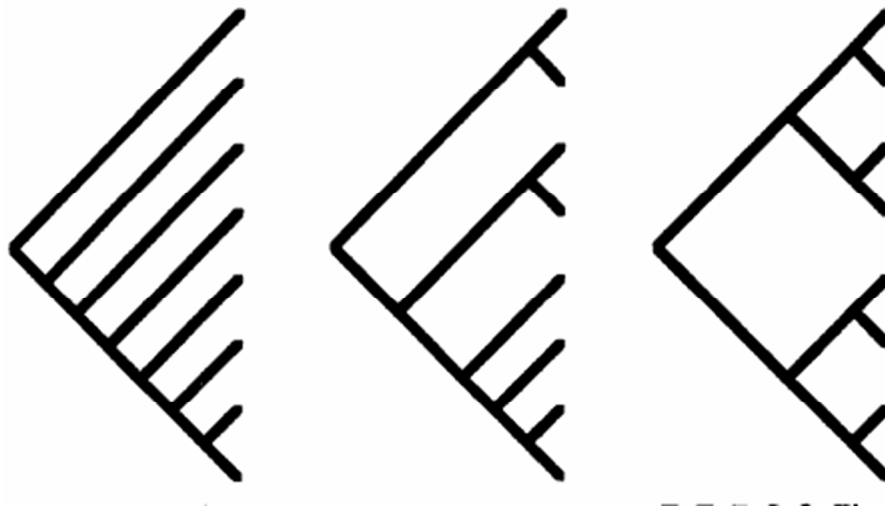


Figure 4 Three phylogenies with topologies

Preliminary Results

Using the results from our phylogenetic analysis, we can measure the diffusion of innovations along three dimensions. To illustrate the power of our method, we present a preliminary analysis using a small set of

sample data from a project called Bootstrap that is created and maintained by Twitter. It is one of most popular projects hosted on Github with more than 5300 forked projects and has been launched for only 12 months with 9 major releases. Our sample dataset includes 7 master project sequences and 9 randomly selected branch project sequences forked during the period of 6 releases. Figure 4 shows the phylogenetic tree of all 16 sequences. Branch projects are labeled as Branch_X, where X is numbered in the same order as they separated from master project. Master projects are labeled as Master_X, where X is the corresponding version number.

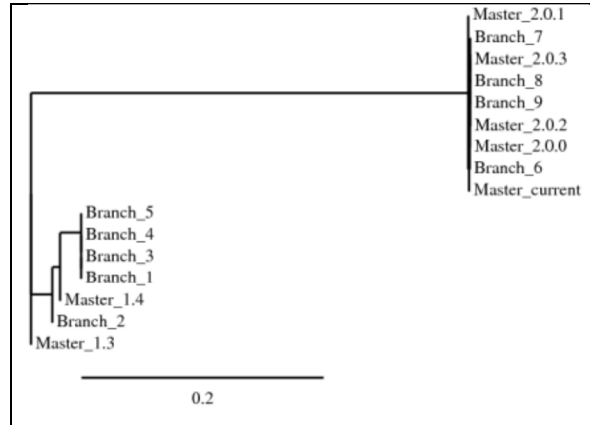


Figure 5. Phylogeny of different versions of “Bootstrap” sequence code.

Figure 4 also enables us to calculate our proposed metrics on different dimensions of innovation diffusion. Proliferation is the total number of tree branches of the phylogenetic tree. This number shows how many developers actually adopted the original ideas in their own way. In the tree in Figure 4, proliferation is 16. To measure the evolvability, we calculate the length for each branch. The length represents the differences among sequences. In our case, it represents how many changes have been made. We consider changes in source code as a physical representation of development in idea. By adding all branch lengths together, we get the sum of length for the tree, which is also the degree of diffusion at the time we collect the data. To precisely measure the diffusion, the actual calculation is based on the numeric phylogenetic matrix. The sum of branch lengths, which is also the degree of diffusion, is 0.37453.

However, this result is not accurate since it also takes the changes in the master project itself into account, which falsely amplified the diffusion. To solve this problem, we create a second tree (Figure 5) only including master branches, and calculate the total length L' as 0.35544. Then, we subtract L' from L and get the actual overall degree of diffusion of 0.01909. After calculating the degree of diffusion, we can further take the temporal factor into consideration to calculate an evolutionary rate. Since it has been 7 months since release 1.3, we then can then estimate the speed of diffusion as 0.0027 ($=0.01909 / 7$) change units per month, which is the overall measurement. Thus, it appears that the temporality of different versions can be quantified using our phylogenetic approach.

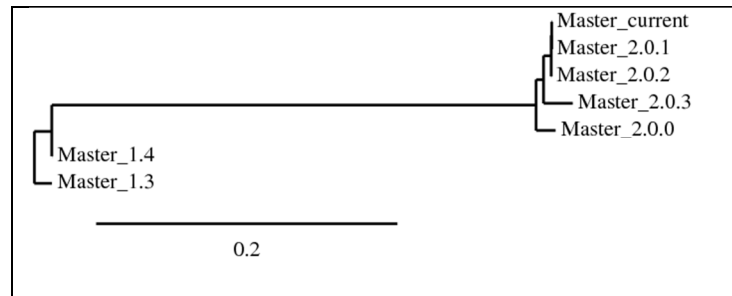


Figure 6. Phylogeny of master project's sequences only.

To further analyze the pattern of diffusion in different stage, we can compare the degree of diffusion and speed of diffusion of projects in the version 1.x (shown on the left in Figure 4) and those within version 2.x (shown on the right in Figure 4). Visually, from the phylogenetic tree (Figure 4), there seems to be a

higher degree of diffusion since version 1.x period (6 months) than that since version 2.x period (2 months): forked projects in 2.x period are basically on the same vertical line. We conduct same analysis for the two periods separately, as represented in Table 1. It confirms our intuition that no diffusion in 2.x period as degree of diffusion is actually 0 at that period. This also seems intuitive because as the master project goes mature, little room is left for generative diffusion. However, the result pertains to a single example, and is mainly for demonstration purposes.

Table 1 Analysis Summary

| Time Period | Degree of Diffusion (changes in percentage) | Speed of Diffusion (changes in percentage per month) |
|-------------|--|---|
| Overall | 1.909 | 0.27 |
| 1.x | 1.909 | 0.318 |
| 2.x | 0 | 0 |

Future Plan

This proposal provides a description of our study, and illustrates the applicability of our measures to study diffusion of open source software. The next step of our study is to collect more data using cyber discovery techniques so that we can create a complete profile of innovation diffusing in different projects. By ICIS 2012, we plan to apply our metrics to all projects posted on GitHub. Then, we will examine the effects of different factors that may serve as antecedents of diffusion, such as designer's or project's characteristics. Moreover, our study can not only provide quantitative data on degree and speed of diffusion, but also offer information regarding to the relationships among those projects, which allows us to construct a network diagram of open source projects, measure the properties of the network, and study the diffusion of innovation through the lens of networks of projects.

Expected Contribution

In this study, we use an organizational genetics approach to extend the traditional concept of diffusion to study the diffusion of generative digital innovations. Using a multi-dimensional view of generative diffusion, we will be able to contribute to research on diffusion by considering the role of evolvability as well as the temporal aspects of diffusion. To the best of our knowledge, this study is also the first to quantitatively measure generative diffusion and create various future research opportunities. Our study also has important implications for managers. Managers can benefit from knowing what characteristics are more likely to make a project generative. Policy makers and governing bodies of open source projects can use this study to formulate best practices such as code reuse which can how to improve the generativity of a platform.

Prior to our study, researchers interested in open source software development, particularly in code reuse, have been using computational techniques, such as file comparison tools, to analyze how certain code may transfer between projects. Many of these researches are conducted by computer scientists where much attention has been put on how to identify and improve the code block reused most. Few studies in management area have also studied code reuse in open source community. Our study differs from previous studies and contributes in two aspects: first, instead of trying to identify what has been used, we focus on how innovative idea, in the form of source code, has been diffused among open source developers. Unlike most prior studies, which start at large scale level, and identify certain reused code from multiple projects, we start from source code in one single file and trace how it has been modified and used through its spreading out. Secondly, in order to facilitate our research, we also propose a novel research method, utilizing software and techniques widely used in genetics, in this study. Traditionally, file comparison tool such as diff or FileMerge for micro level, source code within one file, or even in-house developed tools for macro level, files within or across projects, are often used to detect changes between files. However, the phylogenetic analysis that we propose fits better with our research question for following reason: First, prior tools are mainly designed for the purpose of version control, thus they normally can only compare one file with another file (few can compare with two other files). Also, such tools emphasize on finding changed content, whereas, in our research, we emphasize more on the degree

of changes than the content. Thanks to computational tools used in genetic studies, our method allows us to compare source codes from tens of thousands of files at once and analyze the differences between each two files. In addition, it also allows us to investigate the timeline of changes and create phylogenetic trees to identify the evolutionary pattern. Hereby, this paper does not only provide new understanding of diffusion of innovation but also contributes to the methodology used for conducting Big Data analysis on source code.

Limitations

This study, to the best of our knowledge, represents our field's first exploration into phylogenetic approaches to help us better characterize how innovations evolve. There are a number of limitations that we are presently trying to overcome.

Firstly, the change in the source code may not be necessarily correlated to changes/improvements in the program's functionality. It is possible that changes simply consolidate code or present a non-functional change such as a variable's name. We argue that the impacts of such changes are limited to our study for the following reasons. First, consolidating normally brings improved efficiency which is also one way of improvement. Second, non-functional change usually only involves small amount of codes and should not have significant impact on our result. In addition, we also removed all comments from the source code before analyzing to reduce the impact of improved documentation.

Secondly, our current measurement assumes that all changes are equally important to the whole project, which is not true in most cases. We are considering ways to mitigate this limitation. One possibility is to use discrete measurements instead of continuous measurements to assess the degree of change. Also, a method to estimate line quality of reused code used by King and Lakhani (King and Lakhani, working paper) may provide another approach to overcome this potential limitation.

Finally, in this study, we have utilized one of the most basic phylogenetic algorithms that does not take into account the variation present among source code. We are currently exploring more sophisticated phylogenetic analyses that incorporate probabilistic (e.g., maximum likelihood, Bayesian) and temporally-based models of evolutionary change. The field of phylogenetics has amassed a number of relevant algorithms that may provide a better fit to this particular dataset.

Conclusions

In this study, we argue that diffusion continues after the adoption of novel innovation and should be measured as a multidimensional concept and propose a new way to measure the diffusion of innovation. This new bio-inspired approach allows us to quantitatively compute the degree and speed of this diffusion, enabling us to further address new hypotheses and to generate novel predictions.

References

- Agarwal, R., and Prasad, J. 1998. "The antecedents and consequents of user perceptions in information technology adoption," *Decision Support Systems* (22:1), pp. 15–29.
- Baldwin, C., and Clark, K. 2000. *Design rules the power of modularity*, volume 1, Cambridge, Mass.:: MIT Press,.
- Bass, F. M. 1969. "A New Product Growth for Model Consumer Durables," *Management Science* (15:5), pp. 215–227.
- Benkler, Y. 2006. *The Wealth of Networks: How Social Production Transforms Markets and Freedom*, Yale University Press.
- Boland, R., 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.
- Chesbrough, H., Vanhaverbeke, W., and West, J. 2006. *Open Innovation: Researching a New Paradigm*, Oxford University Press.
- Christakis, N., and Fowler, J. 2007. "The Spread of Obesity in a Large Social Network over 32 Years," *The New England Journal of Medicine* (357), pp. 370–379.

- Galison, P. L. 1997. *Image and logic: a material culture of microphysics*, Chicago: University of Chicago Press.
- Gaskin, J., Lyytinen, K., Thummadi, V., Schutz, D., Yoo, Y., Weiss, A., and Berente, N. 2010. "Design DNA sequencing: A set of methodological artifacts for sequencing socio-technical design routines.," ICIS 2010 Proceedings (Paper 202).
- Godes, D., Mayzlin, D., Chen, Y., Das, S., Dellarocas, C., Pfeiffer, B., Libai, B., Sen, S., Shi, M., and Verlegh, P. 2005. "The Firm's Management of Social Interactions," *Marketing Letters* (16:3/4), pp. 415–428.
- Haefliger, S., von Krogh, G., and Spaeth, S. 2008. "Code Reuse in Open Source Software," *Management Science* (54:1), pp. 180–193.
- von Hippel, E. 1986. "Lead users: A source of novel product concepts," *Management Science* (32:7), pp. 791–805.
- von Hippel, E. 2006. *Democratizing innovation*, Cambridge, Mass.: MIT Press.
- Ho, J. Y. C., and Dempsey, M. 2010. "Viral marketing: Motivations to forward online content," *Journal of Business Research* (63:9-10), pp. 1000–1006.
- Karahanna, E., Straub, D., and Chervany, N. 1999. "Information technology adoption across time: a cross-sectional comparison of pre-adoption and post-adoption beliefs," *MIS Quarterly* (23:2), pp. 183–213.
- King, A., and Lakhani, K. 2011. "The Contingent Effect of Absorptive Capacity: An Open Innovation Analysis," Harvard Business School Technology & Operations Mgt. Unit Working Paper No. 11-102 Working Paper.
- Kirkpatrick, M., and Slatkin, M. 1993. "Searching for Evolutionary Patterns in the Shape of a Phylogenetic Tree," *Evolution* (47:4), pp. 1171–1181.
- Lakhani, K., and Wolf, R. G. 2003. "Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects," In *Perspectives on Free and Open Source Software*.
- Mockus, A. 2007. "Large-scale code reuse in open source software," Presented at the First International Workshop on Emerging Trends in FLOSS Research and Development.
- Morris, M. G., and Venkatesh, V. 2000. "Age differences in technology adoption decisions: Implications for a changing work force," *Personnel Psychology* (53:2), pp. 375–403.
- Plouffe, C. R., Hulland, J. S., and Vandenbosch, M. 2001. "Research Report: Richness Versus Parsimony in Modeling Technology Adoption Decisions--Understanding Merchant Adoption of a Smart Card-Based Payment System," *Information Systems Research* (12:2), pp. 208–222.
- Rogers, E. M. 1962. *Diffusion of innovations*, New York: Free Press.
- Rogers, E. M., and Shoemaker, F. 1971. *Communication of Innovations: A Cross-Cultural Approach*, New York: Free Press.
- Susarla, A., Oh, J.-H., and Tan, Y. 2011. "Social Networks and the Diffusion of User-Generated Content: Evidence from YouTube," *Information Systems Research* (23:1), pp. 23–41.
- Tuomi, I. 2002. *Networks of innovation: change and meaning in the age of the Internet*, Oxford [England]; New York: Oxford University Press.
- Venkatesh, V., Morris, M., Davis, G., and Davis, F. 2004. "User Acceptance of Information Technology: Toward a Unified View," *MIS Quarterly* (27:3), pp. 425–478.
- Wejnert, B. 2002. "Integrating Models of Diffusion of Innovations: A Conceptual Framework," *Annual Review of Sociology* (28:1), pp. 297–326.
- Yoo, Y., Henfridsson, O., and Lyytinen, K. 2010. "Research Commentary--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 Boland, R. 2008. "Distributed Innovation in Classes of Network," Presented at the The 41st Hawaiian International Conference on Systems Science, Big Island, Hawaii.
- Yoo, Y. 2012. "Digital materiality and the emergence of an evolutionary science of the artificial," In *Materiality and Organizing: Social Interaction in a Technological World* Ann Arbor, MI: University of Michigan Press.
- Zhang, Z., Yoo, Y., Sunil, W., and Rob, K. 2012. "Digital Innovation and Organizational Genetics: The Case of Web APIs and Mashups," Presented at the 2012 AOM Annual Meeting.
- Zittrain, J. 2006. "The generative internet," *Harvard Law Review* (119), pp. 1974.

