Herding in Open Source Software Development: An Exploratory Study

Joohee Choi
Yonsei University
134 Shin-chon dong.
Seoul, South Korea.
joohee27@gmail.com

Junghong Choi
Yonsei University
134 Shin-chon dong.
Seoul, South Korea.
junghong@yonsei.ac.kr

Jae Yun Moon
Korea University
An-am dong, Seong-buk Gu
Seoul, South Korea.
jymoon@korea.ac.kr

Jungpil Hahn
National University of Singapore
Singapore, Singapore
jungpil@nus.edu.sg

Jinwoo Kim
Yonsei University
134 Shin-chon dong.
Seoul, South Korea.
jinwoo@yonsei.ac.kr

CSCW '13 Companion, Feb. 23–27, 2013, San Antonio, Texas, USA.
ACM 978-1-4503-1332-2/13/02.

Abstract
In spite of the lack of organizational control, a large number of these self-organized groups have successfully developed high quality software in open source software projects. We examined the process through which coordinated action emerges from the collection of individual developers' choices, i.e., how bottom-up coordination occurs and argue that developer herding on a social coding platform may have a positive impact on OSSD outcomes. As an exploratory study, we analyzed the participation patterns in 10 randomly sampled OSSD projects on a social open source code foundry, Github. Based on the findings we generate theoretical propositions regarding developer herding behavior in OSSD.

Author Keywords
Collaborative software development, coordination; collaboration; open source software development; social computing

ACM Classification Keywords
H5.3. Information interfaces and presentation (e.g., HCI): Group and organization interfaces.

General Terms
Human Factors; Design.

Introduction

Open source software project sustainability depends in large part on its ability to attract a critical mass of developers and to induce them to continue active development [2]. How do developers select the projects on which they will expend development effort? Research to date has identified self-organization as the major process through which developer resources are coordinated within the OSSD communities [1, 7]. Self-organization processes are evident in the project team formation process [8] as well as in the coordination of development activities within the teams thus formed. Crowston [5] found evidence of self-organization within OSSD projects – coding activities were coordinated mostly through developers self-assigning themselves to relevant tasks. Understanding the mechanism of this self-organization process will provide important insight on how to induce more participation in OSSD projects.

Self-organization emerges through indirect interaction between agents who react to and are influenced by the environment and artifacts embedded in the environment that provide cues about the preceding actors' actions [9, 10]. Recent research suggests that the online development environments on which many OSSD projects are hosted may provide an ideal setting within which self-organization emerges [1]. Popular online development environments today provide a persistent trace of information cues regarding participant activities, and thus has the potential to enable stigmergic interaction and coordination amongst OSSD project participants. The goal of this study is to further our understanding of how self-organization emerges through indirect interaction amongst developers in OSSD projects and how this process is affected by characteristics of the online development environment.

In this paper, we report the results of a field study of developers and OSSD projects on Github (http://www.github.com/), a popular online platform for OSSD. We analyzed the patterns of contributions in a sample of 10 OSSD projects on Github. Github is an ideal setting within which to examine how the digital development environment may facilitate the emergence of stigmergic interaction since it implements social media-like features that enable developers to subscribe to feeds that provide real-time updates of both individual developer activities as well as project activities by "watching" projects and "following" developers as one connects to others on social networking sites. These feeds are prominently featured on the developer development dashboards. Recent research has explored how these real-time feeds of ongoing development activities may affect developer activities through increased awareness of development in other projects [6].

Similarly, we propose that developers on Github may exhibit patterns of herding in their contributions to projects. We address the following research questions: 1) how are individual level choices that result in herding behavior triggered on Github? 2) How does group-level herding behavior emerge on Github?

Github

Github is an OSSD project foundry that supports git as the version control system, and thus enables much faster and distributed development. Since it was launched in 2008, Github has grown explosively. In this study, we focus on how the characteristics of Github

affect the emergence of herding with respect to development.

Of the many features that Github provides to support OSSD, we focus on two social features that enable interactions across projects, and distinguish Github from other OSSD platforms: 1) by "following" other developers, one can keep abreast of the developer's coding activities across multiple projects, and 2) by "watching" other projects, one can keep track of all changes to projects that one is interested in, without regard to actual membership in the project. The watch and follow relationships are prominently displayed on the Github developer and project pages. Together, these features enable connections to be made between registered developers on Github and projects hosted on Github, much akin to the links forged within social networking sites, and provide the basis for increased social interactions beyond the boundaries of one's own OSSD projects. Every event generated by a developer or project that one has watched or followed will generate information cues that are prominently displayed in the form of dynamically generated news feeds on Github's landing page.

In this study, we focus on how these characteristics of Github affect the emergence of herding with respect to development.

Method

We conducted exploratory analyses of coding related contributions in a sample of 10 OSSD projects. The procedure is outlined below.

Exploratory Data Analysis (EDA)
We visually explored the patterns of code commits for a sample of 10 OSSD projects.

As we aimed to observe specifically the herding phenomenon, the theoretical sampling method was applied as project selection criteria. Of the 1953 projects filtered in the first phase of excluding projects with no interaction, we identified 195 projects that were within the 90th percentile of social feature use as our theoretical sample pool, and then randomly sampled 10 projects from these for further detailed analysis.

The patterns of code commits were analyzed visually by summarizing the code commit events in graph that displays the date/time of the code commit on the x-axis and the commit location (i.e., module or library) on the y-axis. To explore the effects of the importance of prior commit events, we also augmented the commit event graph as a bubble chart with the size of the bubble representing the number of lines of code altered in the code commit. In sum, a total of 20 graphs for the 10 sample projects (i.e., a commit event graph and a commit event bubble graph for each project) are created and analyzed. Distinctive patterns in the visualizations were identified via group discussion among the 5 authors.

Analysis and Results

In this section, we present the main findings from our analyses of project code commits. The recurring observations will be presented and summarized as theoretical propositions.

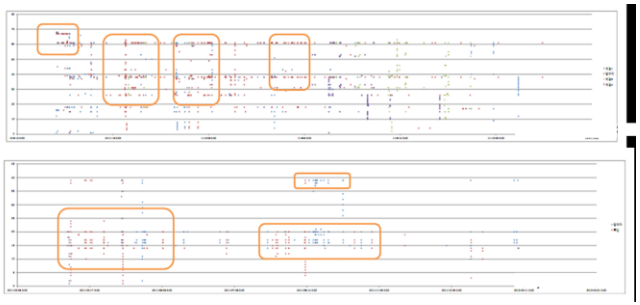

Figure 1. Clustered Contributions

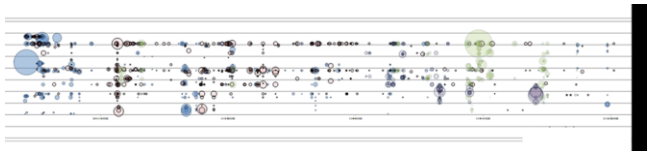

Figure 2. Commit patterns with representations of their sizes

Project-Level Herding Behaviors

Evidence of project-level herding behavior was observed in our analysis of the code commit events. The graphs presented below show representative examples of the identified patterns.

First, we observed clear indication of clustering of commit events as shown on the commit event graph in Figure 1. The highlighted areas (i.e., surrounded by rounded boxes) show that code commits are not distributed evenly over the life-cycle of the project but are concentrated during certain time periods. We therefore propose:

Proposition 1: The patterns of contribution in OSSD projects with more developers who actively make uses of social features will exhibit clustering.

Second, we also observed that contributions were more likely to follow saliently prominent events. For example, we could identify large commits to frequently precede other commits within the clusters of participation, as shown in Figure 2. Recall that the size of the bubbles in the graph indicates the size of a commit in terms of the number of lines of code altered (i.e., deleted and/or added). Larger bubbles tend to be observed at the beginning many clusters of contribution.

According to extant studies that investigated herding behavior in the stock market, extreme price movements were more likely to induce herding behavior among institutional traders [3]. Similar herding behavior was observed here. Several developers mentioned related to this phenomenon in their interview transcripts. For example:

"When watching projects, it's those critical events, for example a project being updated from version 0.1 to 0.12, that gets me going. Events such as a few lines changed here, or a pull request was pushed to the master don't interest me much. But a new release ... that's the type of event I take interest in" (p2)

Based on the above observations, we propose:

Proposition 2: The signals generated by OSSD projects (e.g., extensive code commits, new release etc.) with greater importance are more likely to induce herding behavior from developers.

Finally, we observe that contributions to projects were more likely to occur when the frequency of project contribution events was high. Prior work by Christoffersen and Tang [4] similarly observed that higher trading frequency induced herding behavior among traders. A similar pattern of herding has also been found in our commit event graphs shown in Figure 1. We propose that:

Proposition 3: Signals of events generated with higher frequency by OSSD projects are more likely to induce herding behavior from developers.

Discussion

The results of our interviews and analyses of project commit patterns show that the social features provided by the Github platform does seem to guide individual-level project selection as well as induce herding behaviors from developers who make use of those features. Our study contributes to the literature by providing a (relatively) more definitive account of herding behaviors that may result from stigmergic interactions afforded by Github's social features. We also propose a number of propositions that may (and should) be empirically tested in order to further our understanding of bottom-up collaboration in open collectives such as OSSD.

Acknowledgements

This work was supported by the National Research Foundation of Korea (NRF-20012-8-0334) and National University of Singapore Academic Research Fund (AcRF) Tier 1 Grant.

References

[1] Bolici, F., Howison, J., & Crowston, K. Coordination without discussion? socio-technical congruence and stigmergy in free and open source software projects. Socio-Technical Congruence Workshop in Conj Intl Conf on Software Engineering, (2009). Vancouver, Canada.

[2] Bonaccorsi A, Rossi C. Why open source software can succeed. Research policy. 32(7), (2003) 1243-58.

[3] Cipriani M. & Guarino A. Herd Behavior in Financial Markets: An Experiment with Financial Market Professionals, Journal of the European Economic Association, MIT Press, vol. 7(1), (2009) 206-233, 03.

[4] Christoffersen, S. K., & Tang, Y. Institutional Herding and Information Cascades: Evidence from Daily Trades (2010) Available at SSRN 1572726.

[5] Crowston, K., Li, Q., Wei, K., Eseryel, U. Y., & Howison, J. Self-organization of teams for free/libre open source software development. Information and Software Technology, 49(6), (2007) 564-575.

[6] Dabbish, L., Stuart, C., Tsay, J., & Herbsleb, J. Social coding in GitHub: Transparency and collaboration in an open software repository. Proc. of the ACM 2012 Conference on Computer Supported Cooperative Work, (2012) 1277-1286.

[7] Heylighen, F. Why is open access development so successful? Stigmergic organization and the economics of information. (2006) Arxiv Preprint cs/0612071.

[8] Hahn, J., Moon, J. Y., & Zhang, C. Emergence of new project teams from open source software developer networks: Impact of prior collaboration ties. Information Systems Research, 19(3), (2010) 369.

[9] Holland O, Melhuish C. Stigmergy, self-organization, and sorting in collective robotics. Artif Life. 5(2), (1999) 173-202.

[10] Theraulaz, G., & Bonabeau, E. A brief history of stigmergy. Artificial Life, 5(2), (1999) 97-116.