MODULARITY AND COMMUNITY PARTICIPATION: THE CASE OF TWO FORKS OF AN OPEN SOURCE ERP SYSTEM (IDEMPIERE VS. ADEMPIERE)

Zhengzhong Shi, Charlton College of Business, University of Massachusetts at Dartmouth, 285 Old Westport Road, Dartmouth, MA 02748, 508-910-6513, zshi@umassd.edu

ABSTRACT

This paper is intended to investigate the impact of modularity on community participation in the context of open source complex software systems. In particular, the focus will be on the impact of run time modularity. Community forums for open source ERP systems (i.e., Adempiere and iDempiere which have the same origin) are used as the data source. The empirical test strongly supports the positive impact of run time modularity on community participation. This paper extends the research on modularity and community participation.

INTRODUCTION

Since von Hippel and von Krogh's [9] publication of their research on the private-collective innovation model for open source software systems, community participation has been recognized as the vital force in ensuring the sustainable evolution of open source software project and has thus been widely studied. For example, in Shah's [7] heavily cited study (cited 503 times based on http://scholar.google.com/citations?user=YdcwSQQAAAAJ&hl=en&oi=sra by August 8, 2014), she found that initially, for most participants, individual needs are the dominant motivation for participation in an open source project and over time, among those individuals, some hobbyists' continuous participation plays a critical role in ensuring sustainable growth of the community and the continuous development of open source projects and the "fun and enjoyment derived from the very act of creating and tinkering" [7] are their key motivation for the participation. In addition, she also found the governance structure at the community level such as code control, property rights, fairness, and reciprocity will also significantly influence the extent of community participation. Another widely cited research conducted by Baldwin and Clark [1] (cited 3512 times based on http://scholar.google.com/citations?user=wNN0ka0AAAAJ&hl=en&oi=sra by August 9, 2014) focused on modularity and its impact on product innovation and proposed that modularization can accelerate product innovation through autonomous innovation within each product component and modular mix/match among components . Their study implies that globally distributed teams or individuals may work independently on selected interested component without worrying about the interference among modules due to the autonomy embedded in modular architectural design. These studies contributed significantly to our knowledge about community participation in the context of open source systems.

This current paper will follow these above studies and extend the current research on the impact of modularity in open source systems on community participation. In particular, two forks (Adempiere and iDempiere) of an open source ERP system will be used as data source to investigate the impact of run-time modularity on community participation while keeping the design modularity and the product functionality as equal.

MODULARITY AND COMMUNITY PARTICIPATION

Baldwin and Clark's[1] research points four design problems in modularization. First, it is about the

appropriate number of modules; the second is about the assignment of functions and design elements to each module; the third is about the interactions among design elements within each module; and the fourth is about the interactions between and among modules themselves. With the resolution of these four problems, benefits could be design and development time reduction [8], parallelism in system development and evolution [1] [6], easy recombination of modules and localized modification within each module for design innovations [1]. We believe that these foreseeable benefits due to modular design can make it easier for broader participation from the global developer community because each individual developer does not need to spend time and make efforts to investigate the big picture of the overall design challenge as long as he/she understands the localized functions and design principles for the module he/she wants to get involved. Baldwin and Clark's [2] theoretical finding that systems with more modular designs and more options motivate developers to join and remain in open source system development process and decrease the amount of free riding support this proposition. This current paper is to empirically study the link from modularity (more specifically, run time modularity) to community participation. Further, we believe that with the broader community participation and possibly sustained participation from a subset of the community as hobbyists, community innovations could be more intensive and fruitful. (Note: While we ill not empirically test this proposition, we would like to clarify one condition for this positive relationship between modularity and community innovation. Ethiraj and Levinthal [3] developed a simulation model and studied the relationship between modularity and innovation. One of their key findings is that "given an unknown underlying decomposition of a system, designers are better off erring on the side of integration rather than on the side of greater modularity." To be more elaborative, their finding means that with over-modular design, innovations could be very well hampered and disrupted and the benefits of parallelism due to modular design architecture will be lost due to additional efforts necessary to deal with cross module inter-dependencies. One example for this situation is the experience of the Itanium chip design described in Hamilton [5] where design teams do not really know the underlying modular structure of the chip during their exploration process and with the inappropriate modular architecture, their innovations were indeed hampered. In our study, we are interested in enterprise information system (ERP) that is used to process business transactions such as sales, purchasing, inventory, accounting, manufacturing, and etc. Due to many years of business practice, functional system design, and the available competitive commercial ERP systems, the knowledge of true underlying modular deign of the enterprise system in terms of the number of modules, functional assignments to modules, interaction among elements within each module, and interactions across different modules is comparatively speaking more available in textbooks and research papers than that of technology innovations such as Itanium chip design in Intel corporation. Thus, with the focus on ERP system in our study, we propose that there is a positive impact of modular design on participation for community innovation in the context of open source systems.)

RESEARCH METHODS

A Brief Discussion of Adempiere and iDempiere

Adempiere (adempiere.com/ADempiere_ERP) is an open source ERP systems forked from Compeiere, which was developed by former employees of Oracle. On adempiere.com, a brief description of the background of Adempiere is provided as the following:

> "The ADempiere project was created in September 2006 after a long running disagreement between ComPiere Inc., the developers of Compiere™, and the community that formed around that project. The community believed Compiere Inc. placed too much emphasis on the open

> source *lock-in/commercial* nature of the project, rather than the community *sharing/enriching* nature of the project, and after an impassioned discussion decided to split from Compiere™ giving birth to the ADempiere project."

iDempiere (idempiere.org) is a fork of Adempiere with the same functions but with somewhat different technological architecture. On idempiere.org, a brief description is provided as the following:

> "iDempiere Business Suite ERP/CRM/SCM done the community way. Focus is on the Community that includes Subject Matter Specialists, Implementers and End-Users. iDempiere is based on original Compiere/Adempiere plus a new architecture to use state-of-the-art technologies like OSGi, Buckminster, zk6."

Based on an experienced ERP consultant, the following comparison between Adempiere and iDempiere and some other related ERP systems is provided.

> "The ADempiere, iDempiere, Openbravo and Compiere environments are amazingly similar. iDempiere came from ADempiere. ADempiere and Openbravo came from Compiere. Compiere came from Jorg Janke. Jorg came from Oracle. As a result, iDempiere and ADempiere have much in common with Oracle's ERP in terms of the financial feature set." (http://www.chuckboecking.com/blog/).

Obviously, as we can see from above discussions, there are a quite a lot of similarities/overlaps between Adempiere and iDempiere in terms of source code reuse and functionality. However, a variety of new technologies are used in developing iDempiere. iDempiere uses bitbucket (https://bitbucket.org/idempiere/idempiere) as its source code repository, jira (https://idempiere.atlassian.net/secure/Dashboard.jspa) as its ticketing systems, and Jenkins (http://jenkins.idempiere.com/) as the online compiling and building system. The run time modularity technology used is OSGi (Note: "The OSGi framework defines a dynamic module system for Java. It gives you better control over the structure of your code, the ability to dynamically manage your code's lifecycle, and a loosely coupled approach for code collaboration." [4, p. 2]. "it's a modularity layer for java platform" [4, p. 3].), which enables modules to be dynamically installed, activated, stopped, updated, and uninstalled. To test newly developed/modified modules, for Adempiere, it is necessary to shut down the application server first and then compile, install, and activate the newly developed module to test its performance and impact on the overall system. The efficiency difference between iDempiere and Adempiere when testing new modules with the same functions is estimated to be 2 minutes vs. 20-30 minutes (red1's video https://www.youtube.com/watch?v=Pm0DzUbVhes&list=UUf8hF3GAb7evHwLFiIscufA, accessed on August 11, 2014.). The time difference is due to the run-time modularity of OSGI framework used in iDempiere development. This difference should be in theory motivate participation and facilitate innovations for iDempiere. Further, as mentioned above, bitbucket, jira, and jenkins also support modularity, which enables autonomous code storage, ticket processing, and code compilation.

This current paper is to empirically investigate how these technologies and especially OSGi supported run-time modular design are supporting community participation. The following section is about data collection and analysis.

Data Collection and Analysis

Data for community participation will be collected using community forums. Adempiere and iDempiere communities have their own discussion forums (sourceforge.net/p/adempiere/discussion/ and https://groups.google.com/forum/?hl=en#!forum/idempiere).

For the Adempiere community, the first message in its forum was posted on 09/09/2006. The following table shows the total number of posts monthly for the past year (September, 2013-August 2014).

Items	Sep-13	Oct-13	Nov-13	Dec-13	Jan-14	Feb-14	Mar-14	Apr-14	May-14	Jun-14	Jul-14	8/1/2014 to 08/12/2014	Average	Standard Deviation
Total Number of Posts	117	47	56	41	53	66	35	89	65	77	36	16	58.17	27.19

For the iDempiere community, there are 504 members in the google group for the iDempiere community at the time of data collection (August 12, 2014).

- First message was posted on 05/27 2011.
- Total number of views: 81701,
- Total number of topics: 1635,
- Total number of posts: 6440,
- Average number of views per topic: 81701/1635=49.97
- Average number of posts per topic=3.94

The following table shows the monthly and average posts and topics data over the past year. On average, each month for the past year, there are 308.33 posts and 79.17 topics and for each topic, there are 3.99 posts.

Items	Sep-13	Oct-13	Nov-13	Dec-13	Jan-14	Feb-14	Mar-14	Apr-14	May-14	Jun-14	Jul-14	8/1/2014 to 08/12/2014	Average	Standard Deviation
Total Number of Posts	469	387	433	379	391	278	286	253	338	200	219	67	308.33	113.89
Total Number of Topics	117	99	111	115	105	69	71	67	80	50	52	14	79.17	31.54
Average Number of Posts Per Topic	4.01	3.91	3.90	3.30	3.72	4.03	4.03	3.78	4.23	4.00	4.21	4.79	3.99	

A T-test (type 3-unequal variance samples, 1 tail test) is performed to compare average number of posts per month from September 2013 to August 2014 between iDempiere and Adempiere forums and the p-value=0.0004% (Note: A t-test with the assumption of two-sample with equal variance is also performed and the p-value is 0.00001%, which has the same result as the previous t-test.). Clearly, the community of iDempiere is much more active and more participative than that of the Adempiere.

Further, the following graph shows all the posts in the Adempiere forum since its inception to August 12 2014 when data are collected. As we can see clearly that since middle of 2011 after iDempiere is forked from Adempiere, the number of posts in the forum is decreasing over time. This observation is consistent with the t-test conducted above, demonstrating the influence of run-time modularity on community participation.

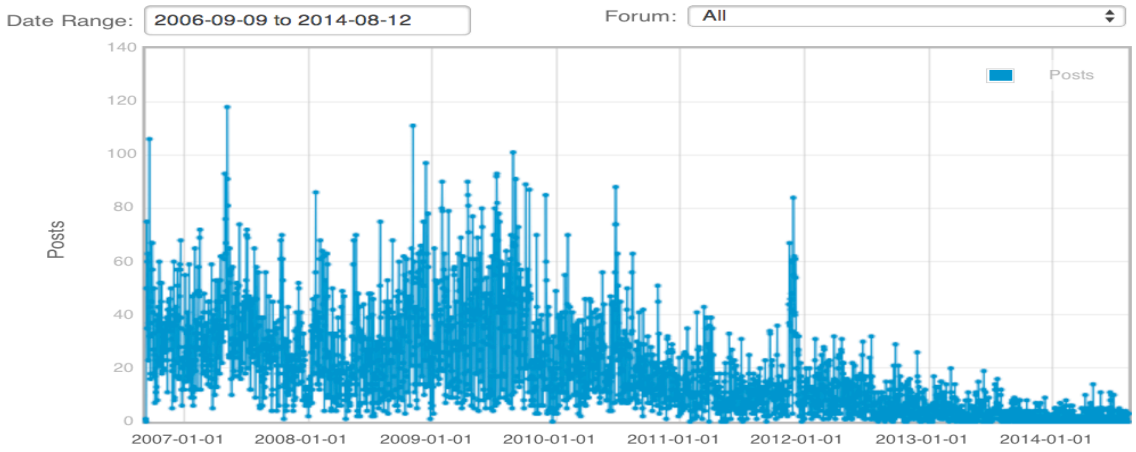

Figure 1: Adempiere Forum Posts Since its Inception

DISCUSSION AND CONCLUSION

This paper reviews a theoretical relationship between modularity and community participation and empirically tests this relationship with focus on the impact of run-time modularity on community participation by using community forum data from two forks of an open source ERP system with the same set of business functions and different technological architecture (iDempiere vs. Adempiere). The test shows strong support of the relationship between run time modularity and community participation.

Adempiere and iDempiere are complex systems and they are two forks of an open source ERP system. By using these two systems as the data source, we believe that the evidence generated in this paper supplements existing research on the relationship between modularity and community participation.

REFERENCES

[1] Baldwin, Carliss Young, and Kim B. Clark. *Design rules: the power of modularity*. Vol. 1. MIT press, 2000.

[2] Baldwin, Carliss Y., and Kim B. Clark. The architecture of participation: does code architecture mitigate free riding in the open source development model? *Management Science* 52.7 (2006): 1116-1127.

[3] Ethiraj, Sendil K., and Daniel Levinthal. Modularity and innovation in complex systems. *Management Science* 50.2 (2004): 159-173.

[4] Hall, Richard, et al. *OSGi in action: Creating modular applications in Java*. Manning Publications Co., 2011.

[5] Hamilton, D. P. (2001). Intel gambles it can move beyond the PC with new microprocessor. in The Wall Street Journal. New York.

[6] Loch, C. H., C. Terwiesch, and S. Thomke (2001). Parallel and sequential testing of design alternatives. *Management Science*, 47 (5), 663-78.

[7] Shah, Sonali K. Motivation, governance, and the viability of hybrid forms in open source software development. *Management Science* 52.7 (2006): 1000-1014.

[8] Ulrich, K. T. and S. D. Eppinger (1999). *Product design and development* (2nd ed.). New York: McGraw-Hill.

[9] Von Hippel, E. G von Krogh. Open source software and the "private-collective" innovation model: Issues for organization science. *Organization Science,* Volume 14 Issue 2, March-April2003, pp. 209-223.