**A Heterogeneous Graph Model to Understand Bug-Fixing Collaboration among Developers**

Jungil Kim, Sunbin Park, Geunho Choi, Eunjoo Lee\*

School of Computer Science And Engineering, Kyungpook National University

80 Daehakro, Bukgu

Daegu, Republic of Korea

[2009307043@knu.ac.kr](mailto:2009307043@knu.ac.kr), [sbp831@naver.com](mailto:sbp831@naver.com), bchoi000@gmail.com, ejlee@knu.ac.kr

**Abstract**

**Recent software project development requires collaboration works of project team members. Thus, understanding of their collaboration relationship in given task is an important key point for successful software project developement. In this paper, we propose a heterogeneous graph model, Bug-Fixing Collaboration Graph (BFCG), which represents bug-fixing collaboration relationship among developers. The BFCG has been constructed based on records of fix and comment activities on the bug reports. In experiment on Jgit open source project, we can identify a bug-fixing collaboration pattern about pairs of developers by investigating visually constructed BFCG.**

Keywords-vizualization; data mining; bug report analysis; software maintenance

**I. Introduction**

As software project system is becoming more and more large and complex, recent software project development requires collaboration works of project team members. For example, developers often discuss and determine which part of files in project should be modified before implementing requirements from end users. In addition, developers often review code change performed by other developers to verify the correctness of the change. Therefore, flexible collaboration work among project term members can lead to reduce further maintenance efforts in project development. As open source project team member are distributed geographically, it is difficult to support and understand the collaborations of members in the open source development.

Bug tracking system[[1]](#footnote-1) is generally used to support bug-fixing works of developers in open source project and industry project development. In the bug tracking system, collaborative works among developers are usually required in bug reports. Developers can contribute other developer's bug-fixing work by leaving their opinions and suggestions as comments in the bug report. The comments leaved by other developers in bug report can provide important hints for fixing the given bug. These collaboration records were often used in prior researches such as developer prioritization, developer recommendation and mentor recommendation.

In this paper, we propose a heterogeneous graph model, Bug-Fixing Collaboration Graph (BFCG) to represent fixed file based bug-fixing collaboration relationship among project team member. The heterogeneous graph model consists of different type of nodes and edges. A developer and a bug-fixed file are represented as a node and a bug fix and comment activities are represented as edge between a developer and bug-fixed file node in the BFCG. The BFCG refers to contributions of developers to each bug-fixed file. Therefore, bug-fixing collaboration pattern among project member can be identified though the BFCG. In experiment on open source project Jgit, we identify a bug-fixing collaboration pattern of a fixed file by performing analysis of a visually constructed BFCG. As the result, we believe that the BFCG could be useful in understanding and predicting further bug-fixing collaboration relationship among project member. Visualization of data is an representation technique which provides complex dataset visually. In previous works, various visualization techniques were proposed [4,5,6]. In [4,5], the authors presented a graph visualization approach which represents communication relationship among project member. They used communication records in e-mail archive and bug reports to construct developer communication graph mode. In [6], the authors showed an visualization of statistic values of code review activities by analzing unrefined code review dataset. The contributions of this paper are as follows :

1. Construction of a novel heterogeneous graph model, Bug-Fixing Collaboration Graph model, by mining bug report fixing history.
2. Identifying an bug-fixing collaboration pattern between a pair of developers of open source project Jgit using a visualization result of the Bug-Fixing Collaboration Graph model.

This paper is organized as follows. The proposed BFCG model is defiend in section 2. In section 3 and 4, we describe visualization based analysis result and conclusion.

**II. The Bug-fixing Collaboration Graph Model**

*A. Bug-fixing collaboration works*

Before we formally define the BFCG model, we first describe bug-fixing collaboration works in bug tracking system. Project team members can contribute to solve a problem of a bug report submitted to bug tracking system. A project manager attempts to find a appropriate developer to the submitted bug report and requests the developer to fix the problem described in the bug report. When it is difficult to find the cause or correct fixing method, the bug fixing process is stoped and the bug fixer try to exchange opinion of the bug with project team member through leaving their comments on the bug report. After significant hints of the bug fix can be obtained through the comments on the bug report, the bug fix process is progressed by incorporating to the given hints. In the aspect of developers, the bug fix contribution is distinguished as two parts, i) a bug fixer who directly fixed the given bug and ii) a bug commenter who offered opinion to find and modify buggy code. The bug-fixing collaboration relationship of a specific bug report can be usefully applied to bug fixing process. For example, when a new bug report is submitted, bug-fixing collaboration relationship of prior bug reports similar to the given bug report can be inferred. The identified relationships enable to identify developer group who can correctly fix the given bug, which makes the bug fix process more flexible.

*B. Definition of a heterogeneous Bug-fixing collaboration graph model*

In this subsection, we define our Bug-Fixing Collaboration Graph model BGCF based on the bug fixing collaboration works in bug tracking system. A bug fixer changes the status of submitted bug report to “fixed” after he correctly fixs the bug report. A bug affects at least one buggy file. Thus, a fixed bug report has obvious links to fixed files [7]. Prior bug-fixed files can be identified through the fixed bug reports in the past. For example, when two source files *Fa*, *Fb* have been modified to fix a bug report *br*, the source files *Fa* and *Fb* are bug-fixed files of the bug report *br*. The knowledge of the contribution of other developers helps a bug fixer handle the buggy file. We propose a graph model which can facilitate to understand buggy files based on bug-fixing collaboration relationship among project member.

The proposed bug-fixing collaboration graph model (BFCG) is comprised of two node types which are developer node and file node and two edge types which are commented edge and fixed edge. The bug-fixing collaboration graph model is defiend as a heterogeneous graph model and it can be formally expressed as , where and mean a finite set of nodes and edges respectively. and mean a finite set of type of nodes and edges , respectively. Each node is mapped to a type of node in by a mapping function . Each edge is mapped to a type of edge in by a mapping function .

*C. Construction of the BFCG*

A BFCG is constructed with a finite set of bug reports . The records of fixer, comments and fixed files in a bug report are required to construct the BFCG. We define a tuple of a bug report data as . Where, fixer, C and F means a fixer of a bug report b, a finite set of commenter and a finite set of fixed files respectively. Given a set of bug reports , the each fixer, commenters and fixed files in given bug reports are involved in a set of nodes of the BFCG. A fixer and a commenter are treated as a developer node. A fixed or commented edge is added between a fixed file node and a developer node. An edge in the BFCG is expressed as a weighted edge. The number of weight of a fixed or commented edge is determined by number of each activity. For example, if a developer commented on a buggy file three time, the edge weight of the commented edge between them is 3.

**III. Application of the Bug-fixing collaboration graph via visualization**

Visualization of data make it possible to easily understand the complex dataset. Visualization based data analysis is desirable to facilitate understanding complex dataset such as graph model [8,9]. In this paper, we visualize our BFCG with JUNG (Java Universial Network Graph) framework and manually analyze the constructed BFCG. Our goal of this analysis is to discover a bug-fixing collaboration pattern between developers in a fixed file.

*A. Experiment dataset*

We select an open source project Jgit[[2]](#footnote-2), an git Java API implementation provide as an eclipse plugin, as our experiment dataset. Jgit project manages its commit history and bug report data with Git repository and bugzilla respectively. Thus, we collect the source code change data from the Git repository and the bug report data from bugzilla. Bug fixing history of source file is required to construct BFCG. However, the bug reports in bugzilla does not specify detailed information of fixed file. Hence, explicit links among bug reports and commit history are needed. we adopt Relink [7], a tool for recovering links between a bug report commit history, to obtain explicit links between collected bug reports and commit history. The collected dataset is summarized in <TABLE 1>.

TABLE 1 The collected dataset of Jgit

|  |  |
| --- | --- |
| Dataset | # |
| Files | 1567 |
| Fixers | 31 |
| Commenters | 301 |
| Commits | 4641 |
| Bug Reports | 472 |
| Links between BugToCommits | 557 |

*B. The result of Visualization analysis*

We choice a set of bug reports which fixed from 1 Jan. 2013 to 28 Mar. 2013 as training set for constructing a BFCG and a set of bug report as evaluation set which fixed after 28 Mar. 2013. In our experiment, the evaluation set is used to verify whether identified bug-fixing collaboration pattern occurred in an bug report involved in the evaluation set. <Fig. 1> shows the overall visualization result constructed with the selected training set.

A great deal of developer nodes in constructed BFCG may make visualization analysis much difficult. Therefore, we determine comment edge weight threshold value with 3 to filter out nodes which have less weight value than 3. A visualization result from the filtered BFCG is depicted in <Fig. 2>. In the visualization result, we can identify a bug-fixing collaboration pattern of two developer nodes, Robin Stocker and Robin Rosenberg who contributed various fixed files with comments together. Their nodes connects to a fixed file node F552 (JGitText.java) with comment edge having weight value 3 and 4 respectively. We confirm their bug-fixing contribution to a bug report 382834 fixed in 17 Oct. 2013. They commented on the bug report 382834 2 times together. The file JGitText.java is also modified to fix the bug report 392934.

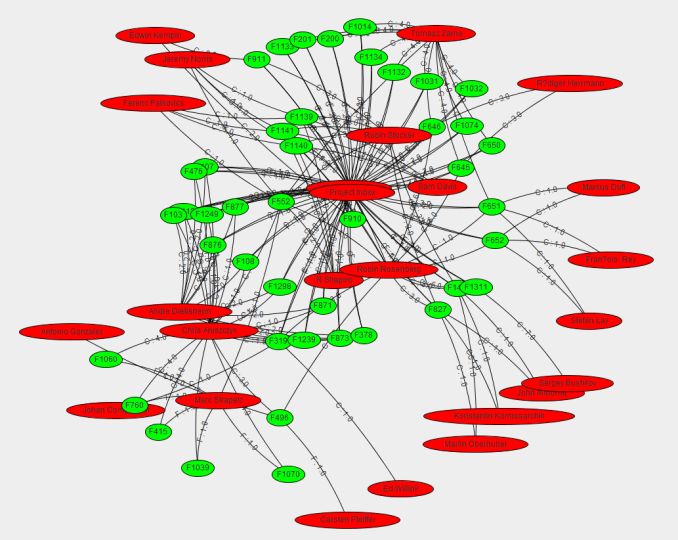


Fig. 1 Overall visualization result of a BFCG constructed with the selected training set.

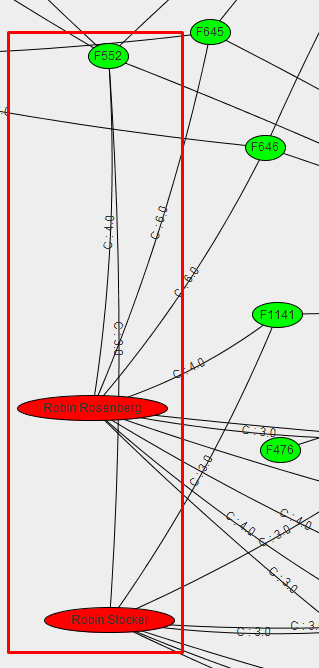


Fig. 2 An bug-fixing collaboration pattern of JGitText.java(F552) between Robin Rosenberg and Robin Stocker

**IV. Conclusion**

In this paper, we propose the Bug-Fixing Collaboration Graph model representing bug fixing contribution of developers. The Bug-Fixing Collaboration Graph model is a heterogeneous graph model which is defined with a finite set of node, *Developer* and *File* node, and a finite set of edge, *Fixed* and *Commented* edges. In experiment on open source projct Jgit, we identify a bug-fixing collaboration pattern between developers in visualized Bug-Fixing Collaboration Graph.

In the near future, we will perform additional visualization based analysis to various open source projects, and will figure out more application of the Bug-Fixing Collaboration Graph Model.

**Acknowledgment**

This study was supported by the BK21 Plus project (SW Human Resource Development Program for Supporting Smart Life) funded by the Ministry of Education, School of Computer Science and Engineering, Kyungpook National University, Korea (21A20131600005), and also supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2012R1A1A3011005).

**References**

[1] J. Xuan, H. Jiang, Z. Ren and W. Zou."Developer prioritization in bug repositories," ICSE, pp.25-25, 2012.

[2] Y. Tian, D. Lo and C. Sun. "Drone: Predicting priority of reported bugs by multi-factor analysis," ICSM, pp.200–209, 2013.

[3] I. Steinmacher, I. S. Wiese and M. A. Gerosa. "Recommending mentors to software project newcomers," RSSE, pp.63-67, 2012.

[4] A. Sarma, L. Maccherone, P.Wagstrom and J. Herbsleb. "Tesseract: Interactive visual exploration of socio-technical relationships in software development," ICSE, pp.23-33, 2009.

[5] J. Wang and A. Sarma. "Which bug should I fix: helping new developers onboard a new project," Int. Work. CHASE pp.76-79, 2011.

[6] P. Thongtanunam et al. "ReDA: A Web-Based Visualization Tool for Analyzing Modern Code Review Dataset," ICSME, pp.606-609, 2014.

[7] R. Wu, H. Zhang, S. Kim and S.C. Cheung. "Relink: recovering links between bugs and changes," FSE pp.15-25 2011.

1. Bugzilla or Jira

   \* Correponding author [↑](#footnote-ref-1)
2. https://eclipse.org/jgit/ [↑](#footnote-ref-2)