Bugarium: 3D Interaction for Supporting Large-Scale Bug Repositories Analysis

Papon Yongpisanpop, Hideaki Hata, Kenichi Matsumoto
Graduate School of Information Science
Nara Institute of Science and Technology, Japan
{papon-y, hata, matumoto}@is.naist.jp

ABSTRACT

Big data became problems not just how to analyze and visualize but also how to interact with the data. In software analysis and maintenance, bug tracking system receive feedbacks of the software project users everyday, which means that the data is increasing everyday. A large-scale bug tracking system that contains large amount of information does not give end users an easy way to analyze bug information because it lacks of good interaction system. We present Bugarium that integrate 3D Motion Controller and data-driven documents to ease both interaction and visualization on a large-scale bug repository. Bugarium leads to a significant increase in terms of using 3D motion controller to operate big data in software visualization. An user study shows that Bugarium made users satisfied while using it to interact and visualize a large-scale bug tracking system.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous ; D.2.8 [**Software Engineering**]: [complexity measures, performance measures]

General Terms

Experimentation, Design, Measurement

Keywords

3D Interaction, software visualization, bug tracking, big data

1. INTRODUCTION

Bug tracking system is a software application that is designed to help keep track of reported software bugs. Bug tracking systems play an important role in software development [7, 8] in these days. They are widely used by open source software projects and industrial software development organizations. It allows developers, testers and end users to provide feedback of an incorrect or error or as a request for enhancement of software. In the other hand, bug tracking systems are also used to analyze and maintain software after release.

Having a bug tracking system is extremely valuable in software development. According to the research [3], for a large-scale, widely-used software system, it could receive a large number of bug reports. In 2010, Eclipse project received 49,422 bug reports which mean that the bugs were reported around 135 bugs everyday.

Big data can be turned into big insights. But most of the time it causes complexity and confusion in visualization. Visualizing a large amount of bug information has always been a big challenge in software engineering. Many researches [1, 3, 6, 8] have tried to solve this visualization problem by developing many tools such as bugmap [3], SourceVis [2], bug database [6], evolution history [5], and vocabulary [1].

According to the research [9,11], using only mouse and keyboard slow sometimes users down while navigating through a large-scale data system. we realized that to perfectly visualize data we need both good data representation and good interaction method.

In this paper, we propose Bugarium, which is a tool that implements 3D motion controller that allows users to naturally use hands and fingers to interact with data in large-scale bug tracking systems.

2. THE STRUGGLE TO INTERACT WITH LARGE-SCALE DATA

The primary challenge in our research is to ease interaction of a visualization-based data discovery tool and seeks to derive more value from large-scale data.

Typically, large-Scale data systems use text-based style to represent the data such as Bugzilla, a bug tracking system that has been using in many open source software projects. The major component of Bugzilla is the database that records information about known bugs such as timestamp, severity, priority, erroneous program behavior, and details on how to reproduce bugs.

Normal interface of Bugzilla represents bug information in text-based style as shown on the left of Figure 1. Many researches that we already mentioned in Section 1 only focused on how to make meaning out of a big data by developed visualization-based data discovery tool.

In Figure 1 on the right, we show the relationships between 1197 committers and 1599 bug reports of Eclipse JDT core project in 2008 using force-directed graph which is one of the most famous visualizing graph to represent multi-relations data structure on a 30 inches display, which has the resolution of 2560x1600. Even we have a large display and new

ICSE '14, May 31 - June 7, 2014, Hyderabad, India

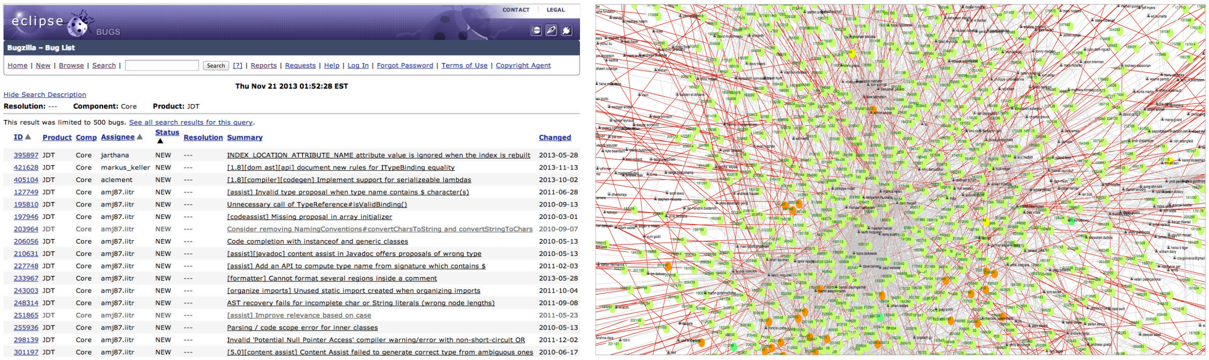

Figure 1: On the left shows Eclipse JDT core project bug reports in Bugzilla user interface. 2796 bug report nodes on the right using force-directed graph to show the relationship between committers and bug reports.

visualization technique to represent large-scale data but yet we still facing the problem that all data can not fit in the display and it is hard to navigate through the data by using only mouse.

3. BUGARIUM

3.1 System Architecture

Figure 2 presents an overview of Bugarium architecture. The system is composed of three layers: view, controller and model. Each layer serves different purposes and works independently from each other.

The top layer is view layer. The purpose of this layer is to display outputs and receive motion inputs from user. Leap Motion is a small USB peripheral device, which is designed to be placed on a physical desktop, facing upward. Using two monochromatic IR cameras and three infrared LEDs, the device observes a roughly hemispherical area. The LEDs generate a 3D pattern of dots of IR light and the cameras generate frames of reflected data, which is then sent to the computer. The motions are analyzed by the Leap Motion controller software.

The middle layer is controller. It has two parallel components: 3D motion interpreter and data controller run together.

Once user interacts with the data in view layer, user interface will pass 3D motion data from Leap Motion to 3D motion interpreter to interpret and send it to data controller. Data controller communicate with data layer to get provided JSON data structure that fed from Bugzilla. Data controller uses D3.js, which is a JavaScript library for manipulating documents based on data to bring data back to user with powerful visualization components.

3.2 User Interaction with LEAP MOTION

We tried to replace mouse by using motion interaction. Bugarium allows users to swipe, zoom-in and out, select, grab or even using the basic hand signs to interact with the data. In this paper we show two interactions, which user can categorize bug reports by selecting priority of the reports using hand signs as shown in Figure 3 and selection comparison for bug reports using fingers shown in Figure 4.

In Figure 3 on the left shows the big picture of the relationships between Eclipse JDT core bug reports and committers in 2008 which have more than 3000 nodes liking together. What if you would like to view only bug reports that have

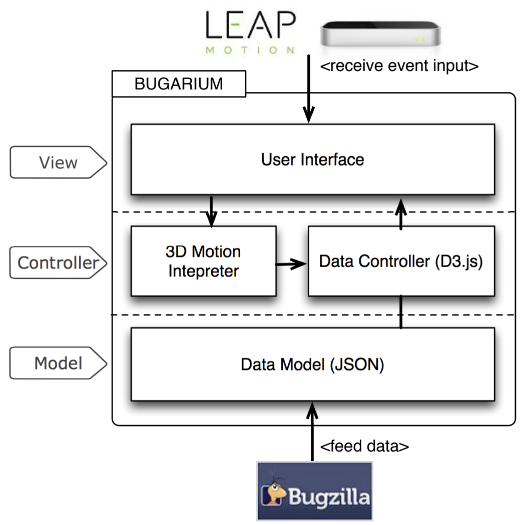

Figure 2: The structure of Bugarium

priority = 5? The problem is even Bugarium has shown a proper graph to represent the multiple relation in bugs but it is not easy to visualize at all when having such a big data.

Normally to select bug reports based on the priority using Bugzilla, users need to click on the selection box to select the level of priority. Or even other visualizing tools need to provide interface option for users to use a mouse to click on it. But Bugarium allows you to use just the basic hand signs to select bug reports as shown in Figure 3 based on the priority without any additional interface component such as checkbox or drop down menu.

Figure 4 shows the benefit of using two hands and fingers to select 2 bug reports for the comparison, which using mouse can not do something like this.

4. AN EXPERIMENT ON ECLIPSE

We sampled 40,000 bug reports from Eclipse JDT core project from 2001 to 2008 and use them as an example to illustrate the usage of our tool. We set up experiments to evaluate the satisfaction of Bugarium overall usage. We selected ten participants from our software engineering lab., which have variety roles in software development process including programmers, testers and end users to use Bugarium to visualize and navigate through the bug reports.

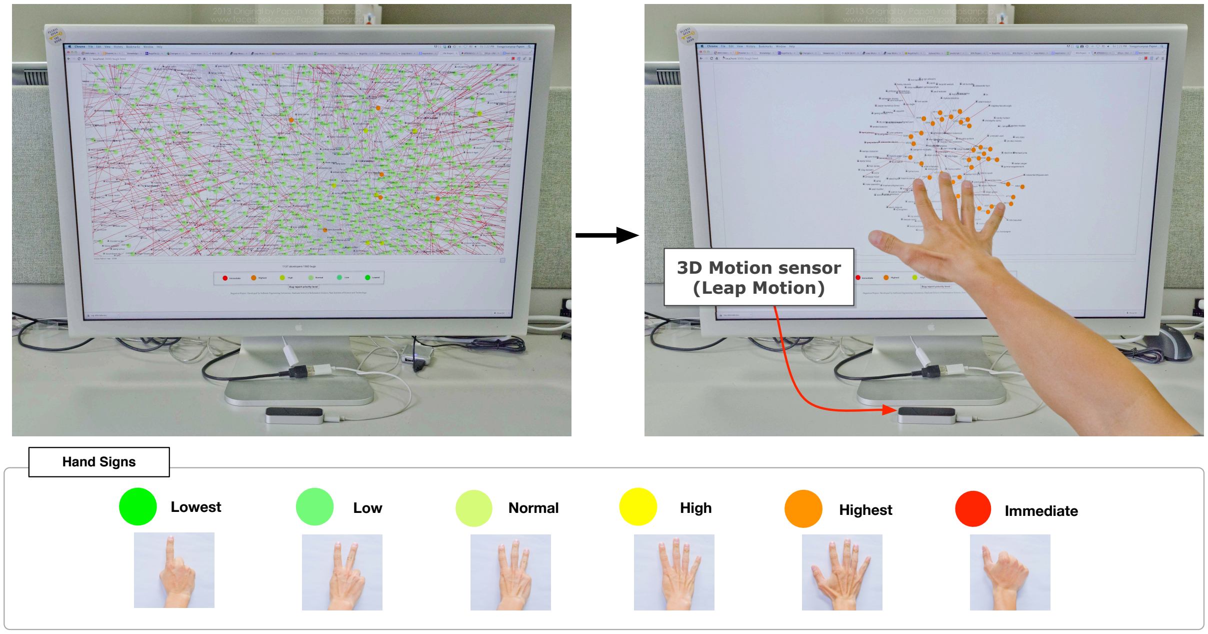

Figure 3: On the left shows the relationships between committer node and bug report nodes of Eclipse JDT core project bug reports in 2008 using force-directed graph. On the right shows how to use hand sign to classify bug reports based on their priority level. We use 5 fingers to represent level 5 of the bug reports. On the bottom shows all the hand signs that use for select bug reports by priority.

We setup two tasks for participants to complete. **1)** *Select bug reports based on priority.* **2)** *Select two bug reports to compare the data inside.* After completed the tasks, we gave a questionnaire for participants to rate our overall usage compare it with Bugzilla. As shown in Figure 5, the questionnaire contains five simple questions. The first question is to see how easy to visualize bug reports. The result shows that using Bugarium is a lot easier to visualize both big picture and specific bug reports.

Second question is to see how quick user can access needed information. The result also shows that Bugarium is better than Bugzilla. participants said that Bugarium allow to use hand signal to interact with data which provide more options than using just mouse to click and keyboard to type to access the information.

Next question is how easy to manipulate data. The result shows that Bugarium and Bugzilla scores are almost the same. "To input data, Bugarium still have to use keyboard to type" one participant said.

The last two questions are how easy to use and learn. Many participants said "It is not hard at all to use Bugarium. The interface and hand signs are based on human common sense."

5. RELATED WORK

In the past, many software visualization techniques have been proposed. For example, Kuhn et al. proposed a tool [1], which in the position of a software artifact reflects its vocabulary, and distance corresponds to similarity of vocabularies. D'Ambros et al. [6] proposed a system to visualize a bug database by using radiography technique to display bug information in the system level and introduced a "Bug Watch" to visualize a specific bug. Wettel et al. proposed Software Systems as Cities [10], which represents large-scale software systems as a city metaphor.

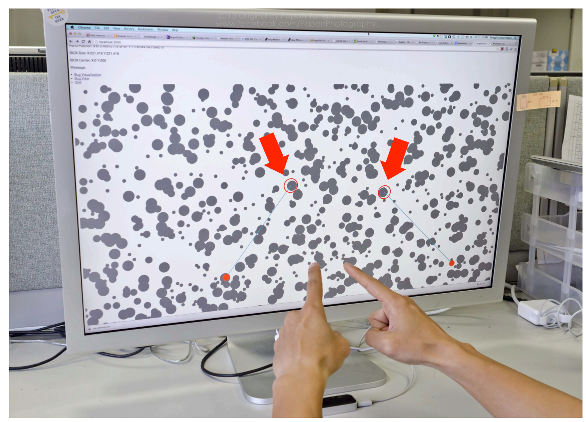

Figure 4: Using two hands to selection bug reports for comparison.

In 2002 the movie called *Minority Report*, in the scene that Tom Cruise manipulates data on a series of large screen with hands has inspired researchers that visualization is not just the way to represent the data but also how we interact and manipulate it. Computer scientist started to develop new type of user interface and input devices that not only existing in science fiction but now in reality [9].

Another research area that tries to use hand gestures is in medical field. Jacob et al. [4] proposed a technique to collaborate with a robotic scrub nurse which allow surgeons use hand gestures and/or voice commands without interrupting the natural flow of a procedure to command rot hand to passes instruments, sutures, and sponges during surgery.

6. DISCUSSIONS

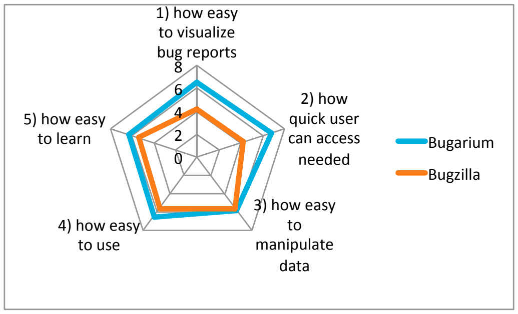

Figure 5: Participant ratings of Bugarium and Bugzilla. The result has shown that participants seemed to satisfy with the overall usage of Bugarium.

- **What is the new idea?** We propose a system that uses 3D Motion Controller to help interact with big data. We implemented a tool called Bugarium which allows users to ease interaction and visualization on large-scale bug repositories.

- **Why is it new idea?** Many researches have tried to propose new techniques to represent a large-scale of data in software engineering field. But none of them did not consider further more on how to interact with big data. To our best knowledge, it is novel to interact and visualize bug reports with 3D interaction.

- **Who gain the benefit from our tool?** Bugarium is designed for everyone who involves with software development process to gain the benefit from new user interface and 3D interaction.

- **Why better than use mouse?** Basically human can use hands and fingers to do things in everyday life such as writing, holding, lifting or even talking with hand signs. Using mouse just like limit users to only have one hand and a finger. In Bugarium, we allow users to use theirs hands and fingers freely to interact with data inside. Users can zoom in, zoom out, grab and select objects with their hands and fingers.

- **What is the limitation?** Bugarium still need to use keyboard to input data into the system. In the future we will try to integrate voice recognition for users to input data by just speak into the microphone.

7. CONCLUSIONS

We have pointed out that to solve problems on a large-scale and complex data by just delivery users visualization tools and techniques are not enough. In this paper, we propose a system called Bugarium, which combine visualization and interaction methods together by using data-driven documents and 3D motion controller to help users fully manipulate data on a large-scale bug repositories. Using Bugarium could benefit everyone in software development process, it allow users to easily interact with the bug reports in big and specific picture by just using both hands and fingers to interact with the data.

In future, we would like to improve the performance of Bugarium by adding these following features.

Remote collaboration We would like to enable remote collaboration in Bugarium for some analysis tasks that users need to collaborate together.

Input technology We think that keyboard is not going to get replaced for a really long time, because it is really good at text. But we would like to provide another text input option by using microphone and voice recognition in Bugarium.

Data representation We will make the most meaning out of large-scale data visualization by research more on visualize technique to represent the specific data structure and appropriate motions that can be used to interact with the data.

8. ACKNOWLEDGMENTS

This study has been supported by JSPS KAKENHI Grant Number 25880015.

9. REFERENCES

[1] P. L. A. Kuhn and O. Nierstrasz. Consistent layout for thematic software maps. In *Proceedings of the Working Conference on Reverse Engineering*, pages 209–218, 2008.

[2] C. Anslow, S. Marshall, and R. Biddle. Sourcevis: Collaborative software visualization for co-located environments. In *Proceedings of the 1th IEEE Working Conference on Software Visualization (VisSoft)*, pages 1–10, 2013.

[3] J. Gong and H. Zhang. Bugmap: A topographic map of bugs. In *Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering*, pages 647–650, 2013.

[4] M. G. Jacob, Y.-T. Li, G. A. Akingba, and J. P. Wachs. Collaboration with a robotic scrub nurse. *Commun. ACM*, 56(5), May 2013.

[5] R. Koschke. Software visualization in software maintenance, reverse engineering, and re-engineering. *Journal of Software Maintenance*, 15(2):87–109, 2003.

[6] M. P. M. D'Ambros, M. Lanza. A bug's life" visualizing a bug database. In *Proceedings of the 4th IEEE International Workshop on Visualizing Software for Understanding and Analysis*, pages 113–120, 2007.

[7] A. Mockus, R. T. Fielding, and J. D. Herbsleb. Two case studies of open source software development: Apache and mozilla. *ACM Trans. Softw. Eng. Methodol.*, 11(3):309–346, July 2002.

[8] C. R. Reis and R. P. de Mattos Fortes. An overview of the software engineering process and tools in the mozilla project. In *In The Open Source Software Development Workshop*, pages 155–175, 2002.

[9] N. Savage. More than a mouse. *Commun. ACM*, 56(11):15–16, Nov. 2013.

[10] R. Wettel, M. Lanza, and R. Robbes. Software systems as cities: A controlled experiment. In *Proceedings of the 33rd International Conference on Software Engineering*, pages 551–560, 2011.

[11] Z. Zhang, Y. Wu, Y. Shan, and S. Shafer. Visual panel: Virtual mouse, keyboard and 3d controller with an ordinary piece of paper. In *Proceedings of the 2001 Workshop on Perceptive User Interfaces*, pages 1–8, 2001.