Interactive Visualization System for Monitoring Issues in Industrial Software Development

Jinwook Bok*
Seoul National University
Soohyun Lee†
Seoul National University
DongHwa Shin‡
Kwangwoon University
Ku Bong Min§
LG Electronics
Gyoseok Chu¶
LG Electronics
Jinwook Seo‖
Seoul National University

ABSTRACT

In order to effectively manage and resolve multiple errors that arise during the development of large software applications, it is a typical approach to record errors in the form of issues. Issues contain crucial information related to the error, such as the symptoms, conditions, and resolution progress, which can be referred to when dealing with the errors. As such information stored in these issues provides insights into patterns on how errors occur and are resolved, it is essential to manage and monitor them effectively. However, handling multiple issues can be a challenging task; in addition to the large number of issues that require management, each issue contains multiple data fields, which may be updated during the resolution process. In response to these difficulties, we designed IssueML, a visualization system for monitoring and analyzing multiple issues. Based on expert interviews, IssueML is equipped with specialized visualization techniques for monitoring issues and their progress over time. With the help of multiple, coordinated views, IssueML enables scalable observation and analysis of multiple issues, following the Visual Information Seeking Mantra.

1 INTRODUCTION

In industrial fields where large projects are collaboratively developed, it is natural for many errors to occur during the process. To efficiently manage errors, symptoms and conditions during the occurrence are organized as issues. Since issues correspond to each errors, project managers utilize them as milestones to be resolved to ensure the progression of the development process. Hence, one of the primary responsibilities of the managers is to monitor and manage ongoing issues and their related developers responsible for resolution. However, monitoring issues is not a simple task. Not only are the number of errors significant, but errors also occur in complicated and unexpected patterns that are proportional to the size of the project. Furthermore, information in issues is frequently updated over time due to newly revealed information during the resolution process, making monitoring even more challenging.

We introduce IssueML, a visualization system that facilitates the monitoring of multiple issues generated from large projects. In designing IssueML, we conducted interviews with industrial insiders to identify domain problems related to managing multiple issues. From the interviews, we define two critical tasks in monitoring issues: managing the related developers and visualizing the progress over time. Based on these tasks, IssueML consists of multiple specialized views that support the visualization of multiple issues in a space-efficient and intuitive manner, with respect to their related developers and progress over time. These views are designed by following the Visual Information Seeking Mantra [2], enabling users to steer their data exploration, starting from an overview of overall issues and progressing to the details of a selected subset of issues.

*e-mail: bok@hcil.snu.ac.kr
†e-mail: shlee@hcil.snu.ac.kr
‡e-mail: dhshin@kw.ac.kr
§e-mail: kubong.min@lge.com
¶e-mail: gyoseok.chu@lge.com
‖e-mail: jseo@snu.ac.kr

2 BACKGROUND INTERVIEW

We conducted interviews with insiders from a large electronics company to gain a deeper understanding of how industry professionals handle multiple issues, and related problems they have. The interviews involved four managers, responsible for managing their subordinate developers, and five developers, responsible for actually resolving the assigned issues. The interviews were conducted in a hybrid (either remote or in person) fashion, depending on the individual conditions. We used a semi-structured interview format, in which we observed their working routines in monitoring or resolving issues, and then asked open-ended questions based on the observation. Based on the interviews, we identified two primary tasks in monitoring multiple issues.

Task 1: Identifying developers' status. One of the primary roles of managers is to review and monitor developers' involvement in the assigned issues. Although managers could refer to the values of selected fields (e.g., the *assignee* field), interviews revealed that solely relying on the values may be misleading. Values of fields may change over time during the resolution process; for example, the assignee field frequently changes over time as the cause of the issue is updated. To address this issue, IssueML provides visual cues that enable managers to monitor developers' status across state changes.

Task 2: Identifying updates in issues. Monitoring issues requires closely observing how the issue progresses over time. In the past, managers had to manually read through most comments left by developers to track changes while understanding the context of the issues, which was inconvenient and required a significant amount of time. Alternatively, they could rely on the internal issue management system [1], which only supported limited tasks for monitoring issues. Log data contained all of the update histories but was often very large and unstructured, making it difficult for users to analyze, and thus was not utilized in the monitoring task. To address these challenges, we designed an effective visualization system that efficiently summarizes the progress of issues over time, following the Visual Information Seeking Mantra. This approach allows managers to quickly and effectively monitor issues, starting from the overview of the whole issues to the details of the focused issues.

3 ISSUEML

Based on the identified tasks from interviews, we designed a visualization system called IssueML (Figure 1) that consists of four coordinated views dedicated to effectively monitoring multiple issues in a scalable manner. In line with the Visual Information Seeking Mantra, users can explore the data starting from the overview of multiple issues to the details of a single issue.

Figure 1: IssueML visualizing a real world data. Some fields are censored due to privacy. (a) From the attribute view, users can observe the distribution of each attribute, and select ranges or values of interest to filter the issues. (b) Each of the selected issues is displayed in the issue list view. (c-1) Status of each developers can be easily monitored from the stacked bar chart of analysis view. (c-2) Upon selecting one developer of interest, progress of each of the corresponding issues is visualized. (d) Detail of one selected issue can be observed in the information view.

The **attribute view** (Figure 1(a)) can be the starting point for the monitoring task of multiple issues. It presents the distribution of each attribute in the issues as either a bar chart (for categorical attributes) or a histogram (for numerical attributes). Users can filter items of interest according to the attribute values by selecting values or ranges in the distributions. When a filter is applied, users can compare the distribution of the overall data to the distribution of the selected issues that are highlighted. The attribute view provides an overview of the issue data by offering distributions of each attribute, where users can select items of interest by applying filters.

In the **issue list view** (Figure 1(b)), selected issues from the attribute view are visualized as a list. Each item in the list reveals essential information about the issue, such as its key value, priority, and assigned organization. Users can further observe the summary by hovering over the list item. Furthermore, users can either click on one of the items to display details of the issue in the information view (Figure 1(d)), or manually select an issue of interest and compare its distributions to the overall distribution, in the same way as selecting items in the attribute view. From the issue list view, users can elaborate on the items selected from the attribute view.

After issues of interest are selected from the attribute and the issue list view, their patterns can be analyzed in the **analysis view** (Figure 1(c)). Users can observe the progress of each developer and further inspect the issues related to a developer from the visualization of each issue log, fulfilling each of the tasks identified from the interviews. Correspondingly, the analysis view consists of two components, the developer component (Figure 1(c-1)) and the issue log component (Figure 1(c-2)).

In the *developer component* (Figure 1(c-1)), the distribution of the selected issue, categorized by the involved developers is displayed as a stacked bar chart. To address the user task of managing developers (Task 1), we categorized the issue into four types, not only reflecting previous practices but also considering changes in fields over time; *Gray* indicates the issues that were previously assigned to the developer, but not currently assigned, and colored bars correspond to the issues currently assigned to the developer, varying by their completion status and due date (*green*: resolved issues, *blue*: ongoing issues that are not overdue, *red*: overdue issues). From the bar charts users can effectively observe the status of each developer and select a developer of interest to be further analyzed.

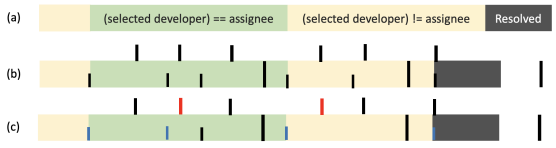

Figure 2: Visualizing a single issue's information over time in the issue log component. (a) Colors indicate changes in the assignee, or the issue's resolution status (b) Short ticks indicate major events (upper: comments, lower: changes in fields), and long ticks correspond to the due dates of the issue. (c) Colors hint special types of events (red: events from the selected developer, blue: changes in major fields)

After a developer is selected, corresponding issues are visualized in the *issue log component*. In the component, progress of each issue over time is displayed, as in Figure 2. To visualize the updates of an issue over time, we render each issue log as a timeline with visual cues that show major progress. With this visualization technique, the previous task of identifying the changes in issues (Task 2), which was not accomplishable unless users read the comments or related logs thoroughly, can be fulfilled by readily observing the visual patterns. In addition, by comparing the patterns between the selected issues, users can group issues of similar patterns, or detect outliers with outstanding patterns. If an item worth further inspection is discovered from the patterns, users can hover over the ticks to retrieve its details or interact with further details in the information view.

Finally, in the **information view**, detailed information on one selected issue can be observed. Details of the selected issue, such as each of the comments and log history, can be examined in inverse chronological order, prioritizing recent updates. This view serves as the final stage of information retrieval, allowing users to inspect the target issue in detail with the provided data.

4 Conclusion

We introduced IssueML, an interactive visualization system for managing multiple issues during the development of large scaled software. Based on interviews with domain users, IssueML consists of multiple views that enable effective monitoring of multiple issues, starting from the overview of multiple issues to the details of selected issues following the Visual Information Seeking Mantra. In the future, we plan to deploy IssueML in an actual industrial software development situation, and further improve the functionalities based on feedback from the domain users.

References

[1] Jira Software. `https://www.atlassian.com/software/jira`, February 2023.

[2] B. Shneiderman. The eyes have it: a task by data type taxonomy for information visualizations. In *Proceedings 1996 IEEE Symposium on Visual Languages*, pp. 336–343, 1996. doi: 10.1109/VL.1996.545307