An Empirical Investigation on Code Debugging and Understanding: An Eye-tracking Perspective
Rachel Turner (Computer Science) - student
Bonita Sharif (Computer Science) – faculty advisor
Alina Lazar (Computer Science) – faculty advisor
Youngstown State University

Introduction

Eye tracking has been used to study cognitive processes involved in the comprehension of visual material for over thirty years [Rayner 1998]. The fundamental design of eye-tracking equipment is based on the physiology of the human visual capability [Duchowski 2003, Jacob 1990]. Most of these systems use cameras to monitor eye movements and eye location with a great deal of accuracy. Eye movement monitoring techniques allow researchers to gather on-line records of processing during higher-order cognitive tasks such as reading and scene perception without disrupting the online process. Eye trackers work by monitoring where the eye is located at any given time, thereby giving researchers information about where participants are looking and how long they spend at any given location.

Fixations and saccades are the two most widely used eye movement measures. Fixations are periods when the eye is at rest on a stimulus (e.g., a phrase), and saccades are quick movements of the eyes from one position to the next. Processing of visual information occurs during fixations [Jacob 1990], and humans use saccades to locate interesting parts in a visual scene to form a mental model of what they are viewing. According to Rayner [Rayner 1998], there is a link between where people look and cognitive processing. This is referred to as visual attention. There is considerable evidence that factors known to influence cognitive processing are reflected in the eye movement record.

The research project proposed here extends earlier work by Sharif concerning the evaluation of how software engineers read and understand software design described using a graphical notation called the UML (Unified Modeling Language). The Unified Modeling Language (UML) [Booch et al. 2005] is widely used in various domains as the defacto standard graphical notation to represent user requirements and design of object oriented software systems. It was adopted as a standard by the Object Management Group (OMG) [OMG] in 1997. UML provides a wide range of diagrams each contributing to a different view of a complex system. The work here focuses on the design view or structural/static view of software that is captured by the class diagram. UML class diagrams contain the main components of an object-oriented system. They present a set of classes, interfaces, collaborations and their relationships. Experimental evaluations conducted on systems using UML diagrams for analysis and design have shown significant improvements in the correctness of changes and design quality versus those that do not use any type of modeling method [Arisholm et al. 2006, Tilley and Huang 2003].
Recently Sharif [Sharif et al. 2012] replicated a study by Uwano [Uwano et al. 2006] on debugging five simple programs using an eye-tracker. In the Uwano study eye movements are used to characterize the performance of individuals in reviewing source code. Their analysis showed that subjects who did not spend enough time initially scanning the code tend to take more time finding defects. The study here follows a similar setup with added eye-tracking measures and analyses on effectiveness and efficiency of finding defects with respect to eye gaze. The subject pool is larger and is comprised of a varied skill level. Results indicate that scanning significantly correlates with defect detection time as well as visual effort on relevant defect lines. Results of the study are compared and contrasted to the Uwano study. Sharif also recently published a journal article on how identifier style affects comprehension [Binkley et al. 2012]. This project directly seeks to extend the previous work [Sharif et al. 2012]. The main tasks this project seeks to investigate is debugging of code.

Research Questions and Hypotheses

There are a number of issues with respect to reading and understanding program source code that have not been well studied with the support of eye-tracking equipment. To date, little or no research has been conducted on these issues in the context of eye-tracking. These studies could have the potential to transform how software developers work on a day-to-day basis by improving their ability to read and comprehend large source code bases. Issues of interest include the following:

- What do developers look at while debugging programs in an IDE setting?
- Do experts differ from novices in the way they approach the debugging process?
- Can we teach novices based on behavior observed from experts?

For each of the research questions, we will create sub-questions and hypotheses to be tested based on the specific debugging task. For example one such hypothesis could read

- Null hypothesis: There is no difference in eye gaze patterns between experts and novices.
- Alternative hypothesis: There is a difference in eye gaze patterns between experts and novices.

Similarly, other hypotheses will be developed.

Methods


Studies conducted to validate certain methods typically form conjectures and/or draw conclusions from the data explicitly collected from subjects’ via a combination of questionnaires, experience reports, and feedback comments after a designated task is completed. This raises a potential threat to the validity of the study, namely the match/disparity between the subjects’ responses on completion of a task and the “reality” they observed while performing that task. For example, a subject may forget to report an observation after a lengthy task. Often times, while on a specific task, subjects’ are hesitant between several choices but only present one of their choices as their answer. Eye-tracking data would allow us to see this type of eye behavior i.e., scan paths (pattern of fixations), between the subjects’ possible choices even though they don’t explicitly say so. This is valuable information, which would aid in developing efficient
methods for the approach at hand. The goal is to reduce the initial effort a software maintainer needs to go through in order to understand software artifacts. This way, he/she can concentrate more on solving the task rather than worrying about correctly finding information in the tool. These have important implications in design of better tools, interfaces, and techniques.

In this research we take a different approach to assessing software artifacts such as source code. We use eye-tracking equipment to implicitly collect a subject’s activity data in a non-obtrusive way as they are interacting with the diagram in performing a given task. These studies will generate and add to the human aspect of empirically evaluating source code reading and comprehension that is missing at this point.

In the presence of an eye-tracker, we can derive concrete eye-tracking measures to create metrics that determine the quality of software artifacts such as source code as well as software tools.

**Eye-tracking Metrics**

The eye-tracking equipment collects three forms of pertinent data including the eye-gazes with respect to the visual presentation and an audio/video recording of the subject during the session. The eye gaze data consists of the following main data:

- **Fixation count**: The number of times a subject fixates on a location on the stimulus.
- **Fixation duration**: How long the subject fixates on a location on the stimulus.
- **Average fixation count**: Total fixation count / number of fixations
- **Average fixation duration**: Total fixation duration / number of fixations

Besides the above metrics, one can also create derived metrics based on the above for specific areas of the stimulus. For e.g., in a debugging task one would like to know the ratio of time spend on part of the screen versus another.

**Impact on the Goal of CREU**

The foremost goal of the CREU project is to encourage females and minorities to pursue graduate work and study in the field of computer science. This project will provide a realistic research experience for the female undergraduate, by active involvement in the planning, execution and interpretation of scientific research. Well-developed research projects can significantly enrich the educational experience for undergraduate students. Working on this research project, the student will be able to enhance her computer and programming skills, apply those skills to investigate scientific problems, learn how to formulate questions and problems and to participate in the discovery of new knowledge. A good research experience can foster an enthusiasm for lifelong learning and a desire to continue education beyond the baccalaureate. Successful scientific instruction should develop in student a sense of wonder and curiosity about the world. The students will be exposed to both sides of the scientific investigation: hypothesis testing and development of theoretical explanations of observations. No science education is complete without research related activities, technical writing and oral presentations.

Rachel Turner considers this project a continuation of the REU program she will attend over the summer. As an undergraduate female with intentions of pursuing graduate work in computer science, this project will give her, a useful introduction to the practical applications of
her studies for research. This is a project that can potentially be a foundation for her senior thesis, and the experimental studies will be a wonderful basis for a research poster.

The student intends to present this project at next year's QUEST at Youngstown State University (a program highlighting student research). The student will likely also present findings at regional and/or national scientific meetings (ICPC and ICSM) within their fields. Results generated by this project will be included in one or more future manuscripts, with participating student afforded full opportunity to share the responsibilities and rewards of authorship.

**Student Activity and Responsibilities**

Specific tasks for the participant student will include: literature search and review, reading and discussing research articles, designing and implementing eye-tracking experimental studies, data processing, data analysis and interpretation, summarizing and preparing results for presentations and publications, YSU QUEST 2013 participation and writing the final report. The student will also be mentored to prepare a conference paper that will be sent to the International Conference on Program Comprehension (ICPC).

The primary responsibility of the student is to participate in all phases of the project: proposal, development, experiments, and dissemination. The student will be required to do weekly independent work and to schedule team meetings with the faculty advisors. The faculty advisors will meet with the student every week. Email and a central repository will be used for questions, announcements and documents interchange.

**Faculty Activity and Responsibilities**

Dr. Sharif has conducted several eye-tracking empirical studies in software engineering. She will be responsible for the developing concrete research hypotheses for each of the research questions mentioned above. She will also mentor the student to develop a good experimental design including selection of variables and methods before the study is implemented. Various aspects of the study design will be discussed with the student. After the data is collected, she will also be responsible for guiding the student to choose appropriate methods of analyzing the data.

As faculty advisor for the proposed project Dr. Alina Lazar will work to actively mentor the student and continuously supervise their progress during the one year period. She will meet with the student on regular basis to guide their activities and answer their questions related to the project. Dr. Lazar has extensive experience data mining, machine learning and artificial intelligence and she has written several papers related to that field. Dr. Lazar will help the student with the data analysis, and with the final report and also with the preparation of a conference paper.

The overall guidance and mentoring will not refer only to this project but it will provided insights about how to apply and how to succeed in graduate school, about being a female scientist and what the options are after graduate school.
Table 1. Project Time Table

<table>
<thead>
<tr>
<th>Task Name</th>
<th>2012</th>
<th>2013</th>
</tr>
</thead>
<tbody>
<tr>
<td>Literature review research about studies that use eye-trackers for</td>
<td></td>
<td></td>
</tr>
<tr>
<td>understanding how programmers perform debugging how they comprehend code.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Analyze existing eye-tracker datasets to understand the dataset format.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Develop and conduct experimental studies using the eye-tracker.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Statistically analyze the eye-tracker data collected.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dissemination of results through papers and communications at specific</td>
<td></td>
<td></td>
</tr>
<tr>
<td>conferences.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Evaluation of the applicability of the performed studies to other areas or</td>
<td></td>
<td></td>
</tr>
<tr>
<td>problems.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Budget**

For the proposed project we are requesting $3000 for the student. The additional $1500 will be used to buy computer media, books, and other materials necessary for the project. While working on the project the student will be encouraged to apply for the Undergraduate Research Grant Award sponsored by the Youngstown State University and other scholarships.

**Role of the CREU project within the larger scope of this research**

The Department of Computer Science and Information Systems has recently established a digital usability lab for undergraduate learning headed by the faculty sponsors. The usability lab (part of the LEGUP grant at YSU) has access to an eye tracker namely, the Mirametrix S2 eye tracker\(^1\). This will be the main eye tracker that will be used for the studies to be conducted in this project. The lab will also have access to three machines to be used for the data analysis and experiments. The eye tracker also comes with an API and eye tracking software that is easy to use and setup.

**Prior results of CREU projects**

Alina Lazar advised two prior CREU projects in 2004-2005 and 2006-2007, and one MROW project in 2007-2008. Darcy Davis the participant student in the 2004-2005 project just finished her PhD in Computer Science and Engineering at Notre Dame University. Louise Popio who participated as a student during 2006-2007, received her Master’s in Information Sciences and Technology from Pennsylvania State University in 2010. Irena Lanc participated in the 2007-2008 MROW project and received her Master’s in Computer Science and Engineering from

\(^1\) http://mirametrix.com/
Notre Dame University this year and is currently pursuing a PhD at the same institution. Erin Pfeil, the other student that did the 2007-2008 MROW project, is working on her PhD in Ecology at University of Pittsburgh.

References


