Usability of Error Messages for Introductory Students

Paul A. Schliep
Division of Science and Mathematics
University of Minnesota, Morris
Morris, Minnesota, USA 56267
schli202@morris.umn.edu

ABSTRACT

Error messages are an important tool for programmers to help find and fix errors in their code. When an error message is unhelpful it can be difficult to understand how to find and fix the mistakes. Error messages are especially critical for introductory programmers in understanding problems with their code. Not all error messages are beneficial for helping novice programmers. This paper discusses how well error messages can help introductory students resolve mistakes in their programs and what aspects make an error message more user-friendly for introductory programmers. After that, we discuss the analyses of syntax, compiler, and exception errors and their results. We will also be discussing several methodologies and programs developed to help improve the experience a novice programmer has when attempting to understand causes for errors and their results.

Will most likely be changing

Keywords

Novice programmers, usability, error messages, usability studies, compiler errors, exception errors, syntax errors, functional-programming

1. INTRODUCTION

One of the most important foundations of computer programming is the communication between the system and the user, specifically in the error messages produced by the system. These error messages are especially important for introductory-level computer science students to help them resolve issues in their program because the error messages are the primary source for understanding what's wrong and according to Marceau, ”[they] lack the experience to decipher complicated or poorly-constructed feedback” [4]. The first rule of good message design is to be sure that the error does not add confusion [3]. When a novice programmer receives an error message that they can not understand, it becomes difficult to fix their program. These difficulties often leads to frustration because the error message was either too complicated to understand or lead them down the wrong path [5], which can sometimes introduce new errors [1].

Several studies have been conducted on modern programming languages' error messages to study the effectiveness in helping novice programmers debug their program and help learn the concepts and programming languages. The results have shown that students struggle with compiler and syntax error messages [1] [7] (which we will discuss in detail in Section 2) and the general vocabulary of the error messages along with IDE-specific features such as source highlighting can be bothersome for introductory computer science students [5].

Several tools and heuristics are being developed to help address issues in error message usability and its development. The goals of these methodologies are to help introductory programmers learn the language and concepts easier. The goal of this paper is to discuss the analyses of error message design and its usability for introductory students in a classroom setting, and how these developed methodologies help improve the user experience with error messages.

This paper is divided into five sections. In Section 2 we discuss usability studies, define compiler, syntax, and exception error messages, and discuss imperative and functional programming. In Section 3 we will focus on analyses of the usability of exception messages and compiler messages for introductory students and how those analyses were performed. In Section 4 we discuss the results of those analyses. In Section 5 we discuss three methodologies developed to help improve the error message usability. The first we will discuss is Traver's heuristics for compiler message design [7]. Then, we will examine Marceau, Fisler, and Krishnamurthi's recommendations for error message design [5]. Lastly, we will explore Denny, Luxton-Reilly, and Carpenter's syntax error enhancement tool [1].

Will most likely be changing

Add image of error message example

2. BACKGROUND

This is where we will describe the concepts such as functional and imperative languages, usability studies, and error message types needed to explain the research done on error message usability. We will use multiple sources from primary and background to describe these concepts.

- Discuss Racket
- Discuss IDEs

UMM CSci Senior Seminar Conference, December 2013 Morris, MN.

- Discuss DrRacket
- Discuss C++
- Discuss Java
- Discuss Compiler errors
- Discuss Syntax errors

TODO

2.1 Runtime and Compiler errors

2.2 Human-computer interaction and usability studies

2.3 Overview of programming languages and tools analyzed

3. ANALYSES

This section will discuss two different studies performed on the usability of error messages and what the results of those studies are. The first analysis will discuss how well the error messages in Racket and DrRacket help introductory students debug their programs. The second analysis will discuss the effectiveness of compiler error messages in the C++ programming language

needs work

3.1 Analysis of error messages in Racket and DrRacket

This will be a combination of two main sources used for this part: Mind Your Language: On Novices' Interactions with Error Messages and Measuring the Effectiveness of Error Messages Designed for Novice Programmers

In the spring of 2010, Marceau, Fisler, and Krishnamurthi ran a usability study on error messages in DrRacket. They had 24 students from an introductory programming course participate in this study. They were interested in which error messages are effective and in what ways. In order to measure effectiveness, the authors developed a rubric which determined whether the student made a reasonable edit in response to the error message [4]. The rubric was meant to distinguish how an error message would fail or succeed. They determined that an error message is effective if a student can read it, understand it, and use that information to figure out how to resolve the issue.

TODO

3.2 Analysis of compiler messages in C++

This will be a combination of two main sources used for this part: On Compiler Error Messages: What They Say and some of Enhancing Syntax Error Messages Appears Ineffectual. (look at appendices - case studies, use examples and code snippets from there)

4. METHODOLOGIES

In this section, we will be discussing three tools and methodologies meant to attempt to improve the usability of error messages or suggest improvements for error message design based on the results of the analyses discussed in section 3. The first methodology we will discuss is a set of recommendations for improving the usability of error messages. The second approach discusses improving compiler error messages through a set of principles meant to increase usability of the messages. For the third approach, we will discuss an attempt on enhancing syntax error messages in Java and the how well these syntax error messages improve over the original.

4.1 Recommendations for error messages

This will use my source, Mind Your Language: On Novices' Interactions

TODO

4.2 Principles of compiler error design

This will use my source, On Compiler Error Messages: What They Say and What They Mean

TODO

4.3 Syntax error message enhancement and results

This will use my source, Enhancing Syntax Error Messages Appears Ineffectual

TODO

5. CONCLUSION

Discussion of the direction usability of error messages will be taken and how the methodologies will be applied for future work.

TODO

6. ACKNOWLEDGMENTS

TODO

7. REFERENCES

[1] Denny, P., Luxton-Reilly, A., and Carpenter, D. Enhancing syntax error messages appears ineffectual. In *Proceedings of the 2014 Conference on Innovation &#38; Technology in Computer Science Education* (New York, NY, USA, 2014), ITiCSE '14, ACM, pp. 273–278. *Discusses syntax errors and helps coincide with compiler source.*

[2] Hartmann, B., MacDougall, D., Brandt, J., and Klemmer, S. R. What would other programmers do: Suggesting solutions to error messages. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2010), CHI '10, ACM, pp. 1019–1028. *Presents a framework for adding hints for error messages, can probably mention hints in methodologies.*

[3] Isa, B. S., Boyle, J. M., Neal, A. S., and Simons, R. M. A methodology for objectively evaluating error messages. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 1983), CHI '83, ACM, pp. 68–71. *can use this in methodologies and probably as a background in several other areas.*

[4] Marceau, G., Fisler, K., and Krishnamurthi, S. Measuring the effectiveness of error messages designed

for novice programmers. In *Proceedings of the 42Nd ACM Technical Symposium on Computer Science Education* (New York, NY, USA, 2011), SIGCSE '11, ACM, pp. 499–504. *Helps with analyses and results of usability of error messages and presents a nice rubric for improving error messages.*

[5] Marceau, G., Fisler, K., and Krishnamurthi, S. Mind your language: On novices' interactions with error messages. In *Proceedings of the 10th SIGPLAN Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software* (New York, NY, USA, 2011), Onward! 2011, ACM, pp. 3–18. *This paper along with the next help with a lot of the analyses and results of error message research.*

[6] Murphy, C., Kim, E., Kaiser, G., and Cannon, A. Backstop: A tool for debugging runtime errors. In *Proceedings of the 39th SIGCSE Technical Symposium on Computer Science Education* (New York, NY, USA, 2008), SIGCSE '08, ACM, pp. 173–177. *Can possibly mention this in the conclusions as a background source.*

[7] Traver, V. J. On compiler error messages: What they say and what they mean. In *Advances in Human-Computer Interaction* (2010).

[7] [5] [4] [1] [2] [6] [3]