

Supporting Multiple Program Comprehension Strategies
During Software Maintenance

by

Susan Elliott Sim

A thesis submitted in conformity with the requirements for the degree of Master of Science
Graduate Department of Computer Science
University of Toronto

© Copyright by Susan Elliott Sim 1998

Abstract

Software maintainers are task-oriented knowledge seekers. They focus on getting the answers they need to complete a task and they use a variety of sources and strategies to do it. This thesis describes the development of a search tool `grug` intended to support program comprehension. This design was based on two user studies and previous work on program comprehension models and tools developed by other researchers. The first study looked at the habits of software maintainers with access to a software visualization tool, the Portable Bookshelf (PBS). The strategies used by subjects to complete maintenance tasks indicated PBS could be improved by adding a search tool, so that information relevant to the immediate task could be more easily located. A second study was undertaken to further characterize programmers' source code searching behaviour to determine what functionality to include in the search tool. Based on these studies and a review of other source code searching and analysis tools, `grug` was designed. This tool supports bottom-up code comprehension strategies by allowing users to search for semantic elements in source code, which they can use to build higher-level concepts. When integrated with PBS, `grug` provides a means of relating program code to the pictorial elements in the software visualization, thereby supporting top-down code comprehension strategies. The suite of tools taken together support multiple comprehension strategies and transitions between them.

Acknowledgements

My first debt of gratitude goes to those who participated in the user studies. Without these people, this research would not have been possible. Standard ethics procedures do not allow me to identify these people for their own protection, so I must thank this faceless cast anonymously.

I would like to thank my supervisors, Ric, for getting me started, and Charlie, for getting me to stop. Thanks also to my “second” reader, Marsha, for her helpful comments.

Finally, many thanks go to my husband, Jeff, for his unfailing support and encouragement.

Table of Contents

Chapter 1: Introduction	1
1.1 Motivation	1
1.2 Starting Points	2
1.3 Organization	4
Chapter 2: The Portable Bookshelf	5
2.1 Overview	5
2.2 The Software Bookshelf Concept.....	5
2.3 Software Landscape.....	7
2.4 Creating a PBS	8
2.4.1 Generating the Factbase	9
2.4.2 Recovering the Software Architecture.....	10
2.5 Populated Bookshelves	11
2.6 Summary	12
Chapter 3: Empirical Studies of Software Maintainers.....	13
3.1 Overview	13
3.2 Methods for Studying Software Maintainers	13
3.3 Code Comprehension Models	15
3.4 Maintenance Tasks	17
3.5 Work Practices	18
3.6 Summary	20
Chapter 4: Studies of PBS Users	21
4.1 Overview	21
4.2 The Software Project	21
4.3 Software Immigrants.....	22
4.3.1 Method.....	23
4.3.1.1 Data Collection	24
4.3.1.2 Data Analysis	25
4.3.2 Results	27

4.3.2.1 Mentoring	28
4.3.2.2 Difficulties Outside of the Software System	30
4.3.2.3 First Assignment.....	32
4.3.2.4 Predictors of Job Fit	33
4.3.3 Application of the Results to PBS	35
4.4 Project Veterans	36
4.4.1 Questions about Edges.....	37
4.4.2 Maintainers' Comments on Anomalies.....	37
4.4.3 Journalism-Style Questions	38
4.4.4 Code Migration.....	39
4.5 Summary	39
Chapter 5: Source Code Searching Survey.....	41
5.1 Overview	41
5.2 Method.....	42
5.2.1 Formulate the Research Questions.....	42
5.2.2 Create a Data Gathering Instrument	43
5.2.3 Define the Population and Sampling Method	47
5.2.4 Administering the Survey	48
5.2.5 Analyze the Data.....	49
5.2.6 Methodological Considerations	49
5.2.6.1 External Validity	49
5.2.6.2 Reliability	50
5.3 Results.....	50
5.4 Participants	51
5.5 Search Tools.....	52
5.6 Situations.....	53
5.6.1 Coding and Analysis of Anecdotes	54
5.6.2 Search Targets	54
5.6.3 Motivations for Searching	55
5.7 Searching Archetypes.....	57

5.7.1 Common Searches	58
5.7.2 Uncommon Searches.....	60
5.8 Respondents' Suggestions for Features	61
5.9 Implications for Tool Design.....	64
5.10 Application of the Results	65
5.11 Summary	66
Chapter 6: Supporting Queries on Source Code.....	68
6.1 Overview	68
6.2 grep.....	69
6.2.1 Strengths of grep	69
6.2.2 Limitations of grep	71
6.2.3 Analysis of grep's Attributes.....	72
6.3 cgrep	73
6.4 sgrep	73
6.5 agrep	74
6.6 tksee	74
6.7 SCRUPLE.....	75
6.8 LSME.....	76
6.9 Comparison of Tools	77
6.10 Lessons Learned	78
6.11 Summary	80
Chapter 7: Design of grug	81
7.1 Overview.....	81
7.2 Platform Requirements.....	81
7.3 Functional Requirements.....	82
7.3.1 Basic grug Functionality.....	82
7.3.2 Requirements for the command-line version	84
7.3.3 Requirements for the Graphical User Interface Version	84
7.3.4 Requirements for Operating Across the World Wide Web	86
7.4 Non-Functional Requirements.....	87

7.5 Specification of <code>grug</code>	88
7.6 The GCL Query Language.....	88
7.7 Markup Schema and Macros for <code>grug</code>	95
7.8 Preliminary Implementation of <code>grug</code> and the Searchable PBS.....	100
7.8.1 Expanding the Factbase.....	101
7.8.2 Augmenting GCL.....	102
7.8.3 Using <code>grug</code>	102
7.8.4 The Searchable Bookshelf	104
7.9 Summary	106
Chapter 8: Conclusion	108
8.1 Observations.....	108
8.2 Future Work.....	109
8.2.1 User Testing	109
8.2.2 Organizational Studies	109
8.2.3 Source Code Searching.....	110
8.2.4 Tool Implementation.....	110
References.....	112
Appendix A: Dictionary of Terms.....	123

List of Tables

Table 4.1: Summary of Respondent Characteristics.....	24
Table 4.2: Summary of Question Set Usage	25
Table 5.1: Origin of Participants By Newsgroup	51
Table 5.2: Origin of Participants by Email Domain.....	52
Table 5.3: Tools Used	52
Table 5.4: "Other" tools used.....	53
Table 5.5: Summary of Common Searches.....	55
Table 5.6: Summary of Motivations for Searching	57
Table 6.1: Comparison of Tool Characteristics.....	78
Table 7.1: Markup and Macros for Function Declarations.....	97
Table 7.2: Markup and Macros for Function Definitions	98
Table 7.3: Markup and Macros of Function Calls.....	98
Table 7.4: Markup and Macros for Variable Declarations	99
Table 7.5: Markup and Macros for Variable Definitions.....	99
Table 7.6: Markup and Macros for Variable References.....	99
Table 7.7: Markup and Macros for Structural References	100
Table 7.8: Available Options in <code>grug</code>	104

List of Figures

Figure 2.1: Portable Bookshelf of the GNU C Compiler.....	6
Figure 2.2: Objc Subsystem of GCC	8
Figure 2.3: Syntax of Tuple Attribute Language in Backus-Naur Format	9
Figure 2.4: Example of TA Language Applied to Software	10
Figure 2.5: Factbase Schema for Software Landscape	11
Figure 4.1: Question Set One.....	26
Figure 4.2: Question Set Two.....	26
Figure 4.3: Question Set Three	26
Figure 4.4: Variables Used During Analysis	27
Figure 5.1: Introductory Page of Questionnaire.....	44
Figure 5.2: Text of Questionnaire	46
Figure 5.3: Message to Solicit Participants.....	48
Figure 5.4: Example of Scenario Anecdote	53
Figure 5.5: Coding Categories	54
Figure 5.6: Usefulness of Searching Source Code by Task	56
Figure 5.7: Example of Wish List for Features	62
Figure 6.1: Call Graph Extractor for C in SCRUPLE from [Griswo96].....	75
Figure 6.2: Call Graph Extractor for C in LSME [Griswo96].....	77
Figure 7.1: Example of grug GUI Search Dialog Box	85
Figure 7.2: Syntax of the GCL Query Language in Backus-Naur Form	89
Figure 7.3: Code Sample from gcc.c of the GNU C Compiler R2.7.2.....	90
Figure 7.4: Line 21 of Code Sample with Hypothetical File Positions Labeled.....	91
Figure 7.5: Markup Index for Line 21 of Code Sample	92
Figure 7.6: Example of Function Declaration	97
Figure 7.7: Example of Function Definition.....	98
Figure 7.8: Example of Function Call.....	98
Figure 7.9: Example of Variable Declaration.....	99
Figure 7.10: Example of Variable Definition	99
Figure 7.11: Example of Variable Reference	99

Figure 7.12: Factbase Schema for GCL Index in TA Format 101

Figure 7.13: The Extended GCL Syntax 103

Figure 7.14: The Searchable Bookshelf..... 105