Exploring software systems
Moonen, L.M.F.

Citation for published version (APA):
Moonen, L. M. F. (2002). Exploring software systems Amsterdam: University of Amsterdam, Faculty of Natural Sciences, Mathematics, and Computer Science

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: http://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.

UvA-DARE is a service provided by the library of the University of Amsterdam (http://dare.uva.nl)
## Contents

**Adventure Starts Here**  

1 Introduction  
1.1 Software Evolution  
1.1.1 Software Immigration  
1.2 Exploring Software Systems  
1.2.1 A Voyage of Discovery  
1.2.2 Urban Exploration  
1.3 Program Comprehension  
1.4 Reverse Engineering  
1.5 Research Questions  
1.5.1 Effective Extraction  
1.5.2 Creating New Knowledge  
1.5.3 Supporting Maintenance  
1.5.4 Software Quality Assurance  
1.6 Organization of this Thesis  
1.7 Origins of the Chapters  

### Part I: Island Grammars  

2 Generating Robust Parsers using Island Grammars  
2.1 Introduction  
2.2 Island Grammars  
2.2.1 Island Grammar Example  
2.2.2 Island Grammar Applications  
2.2.3 Processing  
2.3 MANGROVE  
2.3.1 Syntax Definition in SDF  
2.3.2 MANGROVE/JAVA  
2.3.3 MANGROVE/ASF  

---

ix
## Contents

10.2.2 Classification ........................................... 169  
10.3 Revisiting the Video Store ................................. 171  
10.4 Test-Driven Refactoring ................................... 174  
10.5 Refactoring Test Code ................................. 175  
10.6 Refactoring Sessions .................................. 176  
10.7 Conclusions ............................................ 176  

### Part IV: Epilogue

11 Conclusions ............................................ 181  
11.1 Effective Extraction .................................. 181  
11.2 Creating New Knowledge ............................... 182  
11.3 Supporting Maintenance ................................. 183  
11.4 Software Quality Assurance ............................ 183

Bibliography .............................................. 199  
Summary in Dutch / Samenvatting .......................... 201