Using source-code analysis to help end-user programmers create dependable software p. 3
Software de-pipelining technique p. 7
Abstracting stack to detect obfuscated calls in binaries p. 17
Content-sensitivity matters, but context does not p. 29
Some notes on interprocedural program slicing p. 36
Formalizing executable dynamic and forward slicing p. 43
Control flow reversal for adjoint code generation p. 55
Evolving transformation sequences using genetic algorithms p. 66
Program restructuring through clustering techniques p. 75
Amorphous procedure extraction p. 85
Mining aspectual views using formal concept analysis p. 97
Adding distribution to existing applications by means of aspect oriented programming p. 107
Specifying the law of Demeter and C++ programming guidelines with FCL p. 119
Clone detection in source code by frequent itemset techniques p. 128
Predicting class testability using object-oriented metrics p. 136
Executable source code and non-executable source code analysis and relationships p. 149
A CASE tool platform using an XML representation of Java source code p. 158
Table of Contents provided by Blackwell's Book Services and R.R. Bowker. Used with permission.