Comprehension by derivation p. 3
A qualitative comparison of three aspect mining techniques p. 13
Understanding concerns in software: insights gained from two case studies p. 23
Static techniques for concept location in object-oriented code p. 33
Presenting micro-theories of program comprehension in pattern form p. 45
An investigation into professional programmers’ mental representations of variables p. 55
An investigation of Java abstraction usage for program modifications p. 65
Identifying and addressing problems in framework reuse p. 77
What can programmer questions tell us about frameworks? p. 87
Concise and consistent naming p. 97
Textual views of source code to support comprehension p. 109
On the comprehension of security risk scenarios p. 115
Modelling the information-seeking behaviour of programmers - an empirical approach p. 125

A COTS component comprehension process p. 135
The CodeSurfer software understanding platform p. 147
JRipples: a tool for program comprehension during incremental change p. 149
Visualizing the behavior of dynamically modifiable code p. 337
HyperSoft system: tool demonstration and use example p. 153
SEAT: a usable trace analysis tool p. 157
REGoLive: Web site comprehension with viewpoints p. 161
CHET: checking specifications in Java systems p. 165
JIVE and JOVE: Java as it happens p. 169
NavTracks: supporting navigation in software p. 173
Browsing software architectures with LSEdit p. 176
Theories, methods and tools in program comprehension: past, present and future p. 181
Design and implementation of an extensible and modifiable refactoring tool p. 195
Detecting and visualizing refactorings from software archives p. 205
Understanding object-oriented source code from the behavioural perspective p. 215
An (architecture-centric) approach for tracing, organizing, and understanding events in event-based software architectures p. 227
System evolution tracking through execution trace analysis p. 237
Efficient monitoring and display of thread state in Java p. 247
Clustering software artifacts based on frequent common changes p. 259
Software clustering based on omnipresent object detection p. 269
Achieving a reuse perspective within a component recovery process: an industrial scale case study p. 279
Working session on interoperable reengineering services p. 291
Adding control-flow to a visual data-flow representation p. 297
Supporting the evolution of a software visualization tool through usability studies p. 307

On evaluating the layout of UML class diagrams for program comprehension p. 317