Message from the Chairs

Program Committee

Finding Reusable Software Components in Large Systems p. 2
Reengineering Real-Time Embedded Software onto a Parallel Processing Platform p. 11
Some Experiments Toward Understanding How Program Plan Recognition Algorithms Scale p. 21
On Designing an Experiment to Evaluate a Reverse Engineering Tool p. 31
Rule-Based Detection for Reverse Engineering User Interfaces p. 42
Partial Automation of an Integrated Reverse Engineering Environment of Binary Code p. 50
Mandrake: A Tool for Reverse-Engineering IBM Assembly Code p. 57
Reverse Engineering of Assembler Programs: A Model-Based Approach and its Logical Basis p. 67
A Portable Assembler Reverse Engineering Environment (PARE) p. 76
A Catalog of Object Model Transformations p. 87
Creating a Research Infrastructure for Reengineering p. 98
Encapsulating Legacy Software for Use in Client/Server Systems p. 104
Reverse Engineering a Medical Database p. 121
Structure Elicitation in Database Reverse Engineering p. 131
Combining a Flexible Data Model and Phase Scheme Translation in Data Model Reverse Engineering p. 141
Analysis and Presentation of Recovered Software Architectures p. 153
GASE: Visualizing Software Evolution-in-the-Large p. 163
Object-Oriented Cobol Recycling p. 169
An Overall Process Based on Fusion to Reverse Engineer Legacy Code p. 179
Extracting Reusable Modules from Legacy Code: Considering the Issues of Module Granularity p. 189
Augmenting Pattern-Based Architectural Recovery with Flow Analysis: Mosaic - A Case Study p. 198
Design Recovery by Automated Search for Structural Design Patterns in Object-Oriented Software p. 208
Lessons from a Domain-Based Reengineering Effort p. 217
Identifying Enumeration Types Modeled with Symbolic Constants p. 227
Reverse Engineering Concurrent Programs Using Formal Modelling and Analysis p. 239
Using Informal and Formal Techniques for the Reverse Engineering of C Programs p. 249
Evaluation of a Knowledge-Based Approach to Program Understanding p. 259
Early Field Experience with the Software Reconnaissance Technique for Program Comprehension p. 270
A Cliche-Based Environment to Support Architectural Reverse Engineering p. 277
Preprocessors from an Abstract Point of View p. 287
Author Index p. 297

Table of Contents provided by Blackwell's Book Services and R.R. Bowker. Used with permission.