WCRE Program Committee
Message from the General Chair
Message from the Program Co-Chairs
Some History of CASE for Generating Legacy Systems Documentation
Using Generalized Markup and SGML for Reverse Engineering Graphical Representations
Viewpoint: Explanation on Demand
Viewpoint: Accessible Documentation
Automating Testing by Reverse Engineering of Software Documentation
Recovering Business Rules from Structured Analysis Specifications
Retrieving Information from Data Flow Diagrams
Reengineering Procedural into Data Flow Programs
RESCUE: Legacy Systems Translator
Legacy System Cataloging Facility
Design of a Generic Reverse Engineering Assistant Tool
Customized Tools for Software Quality Assurance and Reengineering
Integrating Reengineering, Reuse and Specification Tool Environments to Enable Reverse Engineering
On Finding Duplication and Near-Duplication in Large Software Systems
Pattern Matching for Design Concept Localization
Analyzing Empirical Data from a Reverse Engineering Project
Observed Idiosyncrasies of Relational Database Designs
Deriving a Logical Data Model for a System Using the RECAST Method
Requirements for Information System Reverse Engineering Support
August-II: A Tool for Step-by-Step Data Model Reverse Engineering
DECODE: A Cooperative Environment for Reverse-Engineering Legacy Software
The Interleaving Problem in Program Understanding
Qualifying Reusable Functions Using Symbolic Execution
Strongest Postcondition Semantics as the Formal Basis for Reverse Engineering
Formal Representation of Reusable Software Modules
Finding Objects in Procedural Programs: An Alternative Approach
Extracting Object-Oriented Specification from Procedurally Oriented Programs
Recovering Abstract Data Types and Object Instances from a Conventional Procedural Language
Reengineering Procedural into Object-Oriented Systems
Recognizers for Extracting Architectural Features from Source Code
Toward Experimental Evaluation of Subsystem Classification Recovery Techniques
Evolution is Essential for Software Tool Development
A Debugging Tool for Software Evolution
Generating User Interfaces from Specifications Produced by a Reverse Engineering Process
Reverse Engineering as a Bridge to CASE
Program Understanding as Constraint Satisfaction