JBOORET: an Automated Tool to Recover
OO Design and Source Models

Hong Mei, Tao Xie, Fuqing Yang

Department of Computer Science & Technology
Peking University, Beijing, China

Oct. 2001
Outline

- Tool Design Principles
- Tool Architecture
  - Data Extractor
  - Knowledge Manager
  - Information Presenter
- Recovered Model Examples
- Conclusions
Design Principles of JBOORET

Jade Bird Object-Oriented Reverse Engineering Tool (JBOORET)

Recovered Models:

• Having different granularities
• Easily being manipulated by users
• Directly being used in forward engineering
• Whose correctness and completeness are ensured
Design Principles of JBOORET (Cont.)

• Data Analysis
  • Light-weight Lexical vs. Heavy-weight Parser-based
  • Incremental analysis

• Knowledge Organization
  • Low-level conceptual model (Stored in advance)
  • High-level design and source models (On demand)

• Information Presentation
  • Different levels or perspectives of models
  • Manipulability of entities in models
JBOORET Architecture Overview

Understanding Task

Version Controller

C++ Programs

Parsing Trees

Extractor

Parser
Lookaheader
Lexer

Data Extractor

Database Server

Incremental Database

Model Extractor

Info Viewer

Info Printer

Knowledge Manager

Info Database

Database Linker
Data Extractor

- Customized Lexer (Lexical analyzer)
  - Comment extraction
  - Physical location association for entities
- Parser based on YACC
  - Token lookahead technique
- Incremental parsing
  - Link incremental databases for each file
Knowledge Manager

JBOORET Conceptual Model
Information Presenter

• Frequently used information
  • Basic symbol information table (Hash table)
  • Loaded in memory in advance

• Complementary infrequently used information
  • A circular cache
  • Loaded on user’s demand

• Export the models to the OO Development Tool (Rational Rose & JBOO)

• Complementary GUI to manipulate the models
Multi-perspective views of recovered models

JBOO
Rational Rose
High-Manipulability GUI
Conclusions

• Model-based (Comprehensive Conceptual model)
• Parser-based (Rather accurate and complete)
• Forward-engineering support
  (Export to Rose & JBOO)
• Multi-perspective models
  (Class diagrams + other models)
• High-Manipulability Model Entities
  • Search, filter and select
  • Compress or expand