ExMAn: A Generic and Customizable Framework for Experimental Mutation Analysis
(MUTATION 2006)

Jeremy S. Bradbury, James R. Cordy, Juergen Dingel
School of Computing, Queen’s University
Kingston, Ontario, Canada
Motivation

- Mutation as a comparative technique has been used traditionally within the *sequential testing* community.
- Our work is based on the idea that mutation can also be used to assess:
  - concurrency testing (e.g., IBM’s ConTest)
  - static analysis (e.g., FindBugs, Jlint, PathInspector)
  - model checking (e.g., Java PathFinder, Bogor)
  - dynamic analysis
Research Goals

1. To compare different techniques using mutation to better understand any complementary relationship that might exist

2. To use the assessment to design improved hybrid techniques to detect bugs.
Experimental Mutation Analysis

- Quality Artifacts
- Original Source
- Mutant Generator
- Quality Assurance Technique/Tool 1
- Mutant Source
- Quality Assurance Technique/Tool 2
- Mutant Analysis Results Generator

Properties

Artifacts
Experimental Mutation Analysis

- Tests
- Original Source
- Properties
- Mutant Generator
- Testing with ConTest
- Mutant Source
- Model Checking with Bogor
- Mutant Analysis Results Generator
The ExMAn Framework

- **ExMAn** = Experimental Mutation Analysis

- What is ExMAn?
  - “ExMAn is a *reusable* implementation for building different *customized* mutation analysis tools for comparing different *quality assurance* techniques.”
ExMAn Architecture

**LEGEND**
- BUILT-IN COMPONENT
- EXTERNAL TOOL COMPONENT

**ExMAn**

- Mutant Generator
- Compiler (Optional)
- Artifact Generator 1 (Optional)
- Artifact Generator n (Optional)
- QA Tool 1
- QA Tool n

- Script Generator & Executor
- Plugin Interface
- Script Generator & Executor

- Source Viewer
- Mutant Viewer
- Compiler Viewer (Optional)
- Quality Artifact Selectors
  - Tool 1
  - Tool n

- QA Tool Viewers
  - Tool 1
  - Tool n

- Results Generator & Viewer
- Hybrid Artifact Set Generator

© J.S. Bradbury • MUTATION 2006 • 7
Video Demo
Related Work

• Mothra
  – a mutation tool for Fortran programs
  – method level mutation operators
    (e.g. relational operator replacement).

• Proteum
  – a mutation analysis tool for C programs

• MuJava
  – the most recent mutation tool
  – designed for use with Java
  – method-level operators
  – class mutation operators to handle object oriented
    issues such as polymorphism and inheritance
Conclusions

- ExMAn is a **generic** and **flexible** framework
- It allows for the automatic comparison of **different** quality assurance techniques
- It allows for the development of **hybrid** quality assurance approaches
Future Work

• Add equivalent mutant identification
• Add ability to automatically specify patterns for the creation of mutation operators
• Expand the artifact selection to allow for the selection of multiple quality artifact sets for each type and thus allow for statistical analysis
Availability

- ExMAn will be available for download in Spring 2007
ExMAn: A Generic and Customizable Framework for Experimental Mutation Analysis

(MUTATION 2006)

Jeremy S. Bradbury, James R. Cordy, Juergen Dingel
School of Computing, Queen’s University
Kingston, Ontario, Canada