Fault Seeding vs. Mutation Operators:
An Empirical Comparison of Testing Techniques for Web Applications

Stacey Ecott

Sara Sprenkle
Lori Pollock

http://www.cis.udel.edu/~hiper

1. General Research Problem

- Experimenters need *faults* to evaluate testing techniques
  - three known methods for obtaining faults

Technique	Advantages	Limitations
manually seeding faults	based on developer's knowledge of application and its implementation	• time consuming • faults may not be realistic
using mutation operators	quickly generates many faulty versions	• equivalent mutants possible • faults may not be realistic
obtaining real faulty versions	**are** real faults	• difficult to obtain

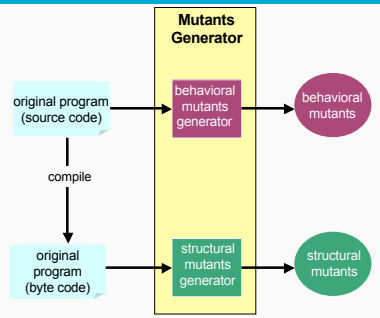
2. Research Focus

- Empirical comparison of fault seeding techniques in a web application
- Manual fault seeding
  - fault categories: data, logic, form, link, appearance
- Mutation operators
  - existing mutation operators implemented in MuJava
- Subject application: MASPLAS
  - conference administration tool (513 lines of code, 5 Java servlet classes, 14 methods, and 4 JSPs with 384 lines of code)

3. Research Questions

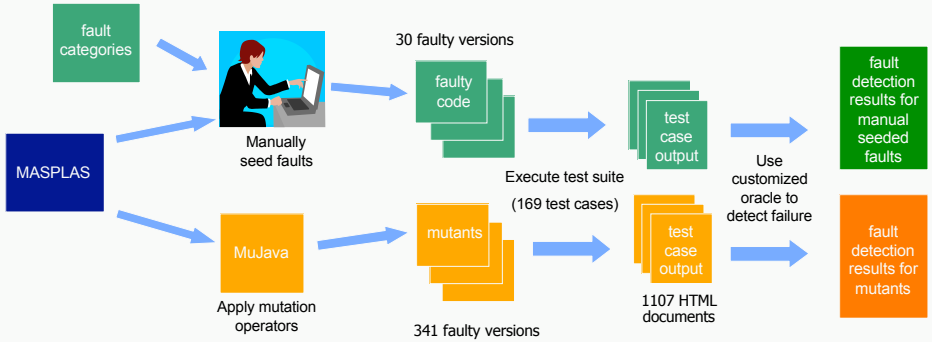
- Empirical
  - E1. What percentage of manually-seeded faults are detected?
  - E2. What percentage of mutants are detected?
- Analytical
  - A1. Which method, manual seeding or mutation operators, provides a more realistic sample of faults?
  - A2. Which characteristics of web applications are not yet covered by existing mutation operators?

4. MuJava: Mutation System for Java



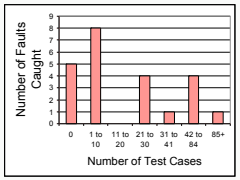
MuJava : An Automated Class Mutation System, Yu-Seung Ma, Jeff Offutt and Yong-Rae Kwon. *Journal of Software Testing, Verification and Reliability*, 15(2):97-133, June 2005.

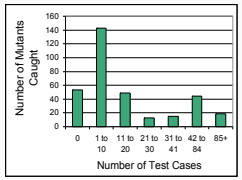
5. Methodology for Generating and Detecting Faults



6. Results

	Total Faults	Faults Detected	Detection Rate
E1. Manual-Seeded	30	18	60%
E2. Mutants	341	288	84.50%





7. Analysis

- Faults caught by 85+/169 test cases are probably unrealistic
- Faults caught by 1 to 10 test cases (0 to 10 for manual seeded) are probably the most realistic
  - not yet empirically proven with real faults
- 56.5% of manual seeded faults fall in the realistic range, compared to 41.9% of mutants
  - A1. With more investigation, this result could show that manual seeding creates more realistic faults than creating mutants
- A2. MuJava does not provide sufficient mutation operators for web application testing (e.g., form, link, and appearance faults)

8. Contributions

- Applied MuJava's mutation operator to web applications
- Empirical comparison of mutants for web applications
- Identified limits of operators for web applications

9. Future Work

- Evaluate approach using other subject web applications
- Compare results with results from real faults
- Apply MuJava to Java code generated from JSPs
- Create web-application-specific mutation operators

Funded by the CRA-W Distributed Mentor Program