

A Fault Detection Framework for Comparing Test Suite Reduction Techniques

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

Emily Gibson
Sreedevi Sampath



Amie Souter



Sara Sprenkle
Lori Pollock



The Fault Detection Comparison Subproject

- Need to compare test suite reduction techniques
- “Best” Reduced suite detects most faults
- Comparison Methodology
 - Seed faults in working application
 - Compare quantity/types of faults detected by each reduced suite

The Larger Project View

- **Goal:** Testing Web Applications for Correctness
- **Problem:** Expensive to replay all test cases
- **Solution:** Reduce number of test cases in test suite
- Several proposed reduction techniques¹

User-Session-Based Testing

- **Test Suite:** set of test cases
- **Test Case:** sequence of URLs + name-value pairs
- **Example:**
`http://www.MyApplication.com/Login.jsp?Login=emily&FormAction=login&Password=gibson`

Challenge: Create a framework to automatically find faults detected by a test suite

Fault Seeding

- Inject faults into working application

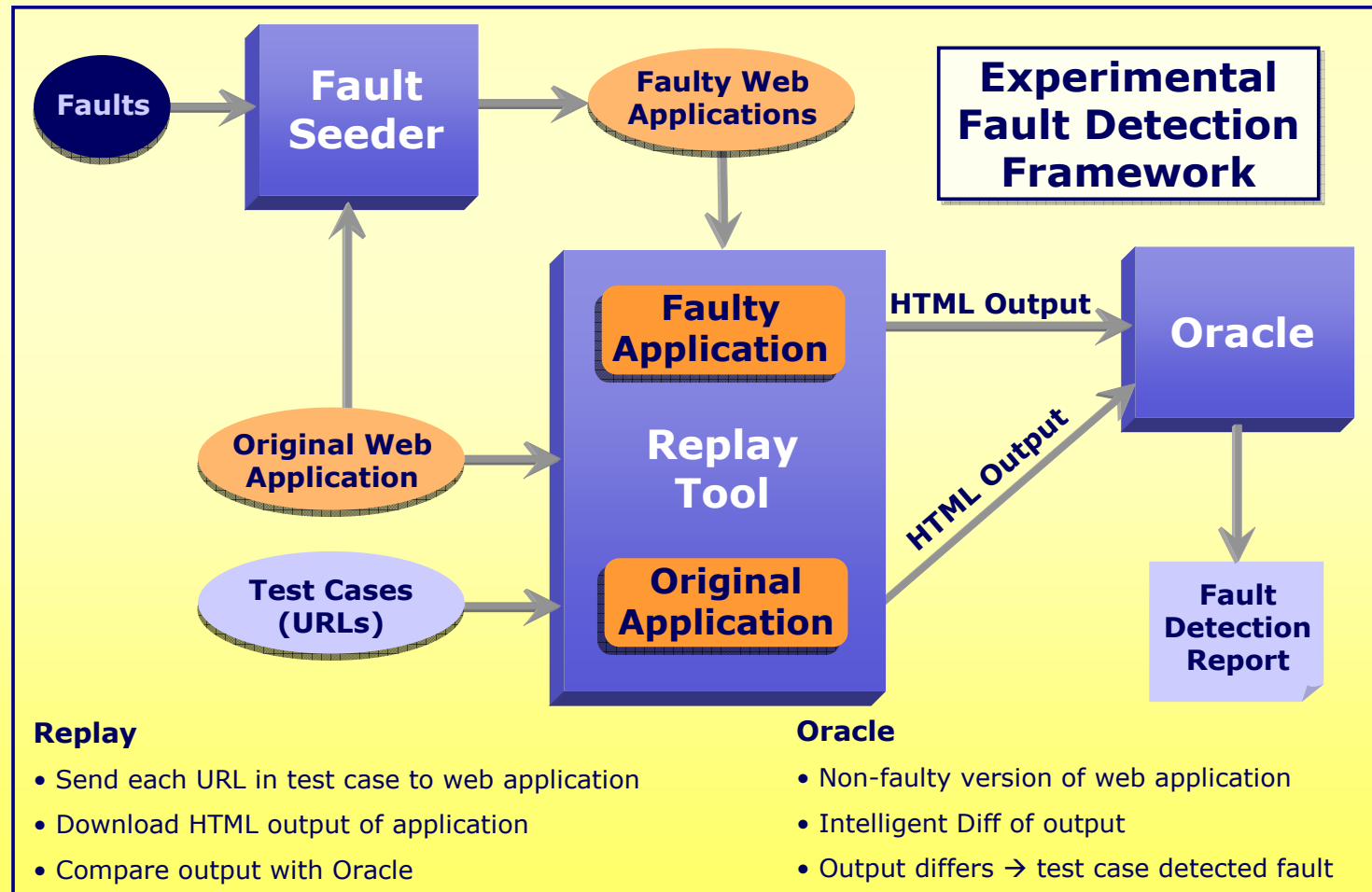
Types of Seeded Faults

- **Scripting:** change control flow
- **Web page:** change HTML, name-value pairs
- **Database:** modify queries, DB structure

Type of Fault	Bookstore	Scheduler
Scripting	28	29
Web page	5	13
Database	7	18
Total	40	60

Achieving Accuracy in Oracle for User-Session-Based Testing

- External data to application
- Time expressions
- Random display order, content same



¹ For details, see Sara Sprenkle’s poster, “An Empirical Comparison of Test Suite Reduction Techniques for User-session-based Testing of Web Applications”