

# An Empirical Study of Statistical Data Models for Effective Automated Testing of Web Applications

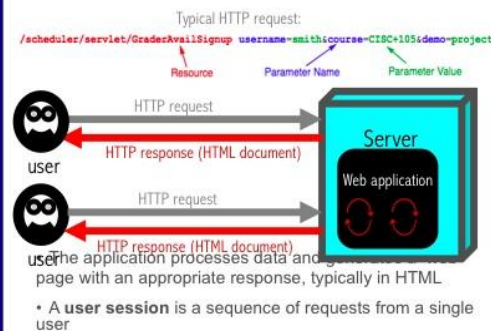


Lucy Simko, Sara Sprenkle, Lori Polk



## WEB APPLICATIONS

- What is a web application?
  - Examples: Amazon.com; Gmail
  - Responds to **dynamic user interaction**
- A web page passes data as parameter name-value pairs to the web application in an HTTP request



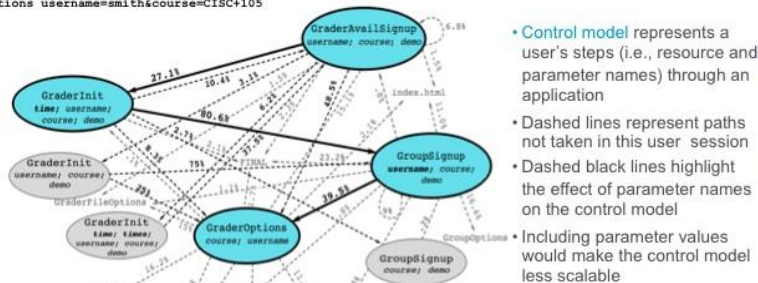
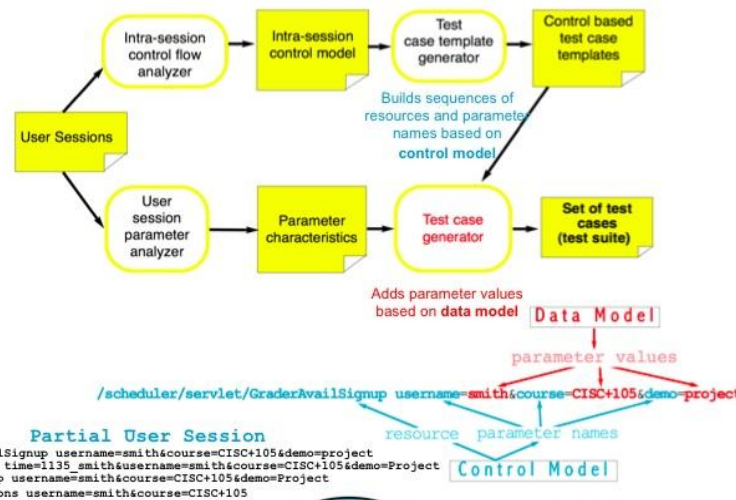
**Problem:** web applications contain millions of lines of complicated code and are **prone to bugs**  
 → *Web applications need to be effectively tested for bugs.*

## CHALLENGES & SOLUTIONS

- One solution is **user session based testing**
  - Goal: **automatically generate test cases** from recorded user requests that emulate real users
  - Pros: Easy to record user requests; Exercises code users frequently access
  - Cons: Testing is limited to what users do; User sessions are not very different from one another, making testing inefficient
- Motivates creating a **model** of user sessions that is smaller than user sessions but generates equally effective test cases
- Sant et al. [SSG] proposed statistical models of user sessions
  - **Control-flow:** models sequence of accessed pages
  - **Data-flow:** models values of parameters

**Key Limitation:** model evaluated on only one application

## TEST CASE GENERATION PROCESS



## OUR FOCUS: DATA MODELS

- SSG Data Models
- **Simple:** Assigns parameter values in groups probabilistically, as found in recorded user requests
  - **Advanced:** Identifies *important* parameters (i.e. sometimes constant across requests); uses important parameters to condition probability
  - Limitation: Restricts combinations of parameter values to those in user requests

- Our Data Models:**
- **Independent:** Assigns parameter values independently  
 → more **variation** in combinations of parameter value
  - **Constant:** Identifies parameters that are always constant across two requests; assigns a constant parameter the value from previously generated request

## RESULTS

**Subjects:** Evaluated on 4 representative web applications and 8 sets of user sessions

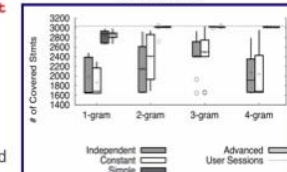
**Methodology:** For each application/set of user sessions, generated 10 test case template suites, each containing 100 test case templates

- For each test case template, generated test cases using each data model
- Executed each test suite and measured effectiveness as the *number of statements* the suite covered

**Graph Legend:**

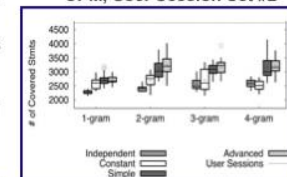
- Box: 50% of data (IQR)
- Whiskers: 1.5 \* IQR
- Center line: median
- + : mean
- o : outlier

### Bookstore



- For  $n > 1$ , coverage is the same as user sessions for Advanced and Simple

### CPM, User Session Set #2



- Even for best test suites, coverage is much less than user sessions
- Problem: insufficient resource coverage

## FUTURE WORK

- Analyze similarity of test cases
- Explore effect of control flow on test case effectiveness
- Generate *unlikely* test cases
- Create alternative data models, e.g., parameters that occur in pairs