Leveraging User-Privilege Classification to Customize Usage-based Statistical Models of Web Applications

Sara Sprenkle, Camille Cobb

Lori Pollock

WASHINGTON AND LEE UNIVERSITY

Test Cases

User Sessions

Classified by User Privilege

User-Privilege Specific Navigation Model

Test Cases
Approach: Usage-based Statistical Navigation Models

1. Record user accesses to a deployed version of the application
   • User session: a sequence of requests from one user
Approach: Usage-based Statistical Navigation Models

1. Record user accesses to a deployed version of the application
   • User session: a sequence of requests from one user

2. Generate usage model from user sessions
Approach: Usage-based Statistical Navigation Models

1. Record user accesses to a deployed version of the application
   • User session: a sequence of requests from one user

2. Generate usage model from user sessions
Approach: Usage-based Statistical Navigation Models

1. Record user accesses to a deployed version of the application
   • User session: a sequence of requests from one user

2. Generate usage model from user sessions
**Approach: Usage-based Statistical Navigation Models**

1. Record user accesses to a deployed version of the application
   - User session: a sequence of requests from one user
2. Generate usage model from user sessions
3. Generate test cases from model

**Benefits:** Compactly represents what users do and what users *could* do
Issues with Generated Navigation Models and Test Cases

- Page requiring Access Privilege A
- Page with no access privilege
- Page requiring Access Privilege B

From a user session for user with access privilege A
From a user session for user with access privilege B

• Can generate a test case with this sequence
• Should test authorization code, but how often?
• Need to generate valid test cases – to test all functionality
The Problem

- All user sessions are treated equally
- Groups of users may interact with application differently
  - May want to test groups separately, differently
Alternatives for Input to Test-Case Generation Process

- All user sessions: state of the art
- User sessions grouped by access privileges
- User sessions grouped by user
- Individual user sessions
Alternatives for Input to Test-Case Generation Process

- All user sessions: state of the art
- User sessions grouped by access privileges
- User sessions grouped by user
- Individual user sessions

Tradeoffs:
- More representative smaller models
- More total space

ICST 2012 - User-Privilege Classification
Sara Sprenkle
Alternatives for Input to Test-Case Generation Process

- All user sessions: state of the art
- User sessions grouped by access privileges
- User sessions grouped by user
- Individual user sessions

Tradeoffs:
- More representative
- Smaller models
- More total space

ICST 2012 - User-Privilege Classification

Sara Sprenkle
Today’s Focus: User Sessions Grouped by Access Privilege

- Can we automatically categorize user sessions by their access privilege?
- What are the tradeoffs of user-privilege specific models compared to the general usage model?
- How can a tester determine if a user-specific model is useful?
Experimental Study

Designed to study effectiveness of categorizing user sessions by user privilege in testing
Subjects

- 4 publicly deployed web applications written in Java using servlets and JSPs

<table>
<thead>
<tr>
<th>Applications</th>
<th># LOC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Book</td>
<td>5279</td>
</tr>
<tr>
<td>CPM</td>
<td>7430</td>
</tr>
<tr>
<td>Logic</td>
<td>10704</td>
</tr>
<tr>
<td>Logicv2</td>
<td>16491</td>
</tr>
<tr>
<td>DSpace</td>
<td>29430</td>
</tr>
</tbody>
</table>
Subjects

- 4 publicly deployed web applications written in Java using servlets and JSPs
- 8 sets of subject user sessions

<table>
<thead>
<tr>
<th>Applications</th>
<th># LOC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Book</td>
<td>5279</td>
</tr>
<tr>
<td>CPM</td>
<td>7430</td>
</tr>
<tr>
<td>Logic</td>
<td>10704</td>
</tr>
<tr>
<td>Logicv2</td>
<td>16491</td>
</tr>
<tr>
<td>DSpace</td>
<td>29430</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>User Session Set</th>
<th># User Sessions</th>
<th># Requests</th>
<th>% Lines Covered</th>
</tr>
</thead>
<tbody>
<tr>
<td>Book</td>
<td>125</td>
<td>3564</td>
<td>61%</td>
</tr>
<tr>
<td>CPM</td>
<td>890</td>
<td>12352</td>
<td>78%</td>
</tr>
<tr>
<td>Logic</td>
<td>497</td>
<td>16179</td>
<td>80%</td>
</tr>
<tr>
<td>Logicv2</td>
<td>374</td>
<td>16052</td>
<td>78%</td>
</tr>
<tr>
<td>DSpace1</td>
<td>1800</td>
<td>22129</td>
<td>74%</td>
</tr>
<tr>
<td>DSpace2</td>
<td>5012</td>
<td>14110</td>
<td>46%</td>
</tr>
<tr>
<td>DSpace3</td>
<td>3853</td>
<td>15126</td>
<td>45%</td>
</tr>
<tr>
<td>DSpace4</td>
<td>7687</td>
<td>38155</td>
<td>49%</td>
</tr>
<tr>
<td>Total</td>
<td>19,525</td>
<td>127,827</td>
<td>-</td>
</tr>
</tbody>
</table>
Open Question:
Can we automatically identify user privileges?
Open Question:
Can we automatically identify user privileges? Yes!

- Each of our subject applications has a resource used for logging in
  - If user session includes a request for that resource → privileged user

- Multiple distinct roles/privileges
  - We manually identified a key resource that classifies a user session as having some privilege

- Scripts run on the order of a few minutes
Results: Classification by User Privilege (CbUP)

- Book and DSpace – only privileged users and not privileged
- CPM: Graders, Groups, NoLogin, MultLogins
- Logic, Logicv2: Professors, Students, Admins, NoLogin, MultLogins
  - Not able to classify all user sessions
    - Logins but no access
Results: Classification by User Privilege (CbUP)

<table>
<thead>
<tr>
<th>Subject</th>
<th>Privilege</th>
<th>% of Original User Sessions</th>
<th>% of Original Requests</th>
</tr>
</thead>
<tbody>
<tr>
<td>Book</td>
<td>Login</td>
<td>91.2%</td>
<td>94.4%</td>
</tr>
<tr>
<td></td>
<td>No Login</td>
<td>8.8%</td>
<td>5.6%</td>
</tr>
<tr>
<td>CPM</td>
<td>Group</td>
<td>57.0%</td>
<td>43.0%</td>
</tr>
<tr>
<td></td>
<td>Grader</td>
<td>24.4%</td>
<td>37.5%</td>
</tr>
<tr>
<td></td>
<td>No Login</td>
<td>17.1%</td>
<td>6.3%</td>
</tr>
<tr>
<td>Logic</td>
<td>Student</td>
<td>77.5%</td>
<td>79.5%</td>
</tr>
<tr>
<td></td>
<td>Professor</td>
<td>15.7%</td>
<td>6.6%</td>
</tr>
<tr>
<td></td>
<td>Admin</td>
<td>2.4%</td>
<td>1.3%</td>
</tr>
<tr>
<td>DSpace1</td>
<td>No Login</td>
<td>90.3%</td>
<td>51.2%</td>
</tr>
<tr>
<td></td>
<td>Login</td>
<td>9.75%</td>
<td>48.8%</td>
</tr>
</tbody>
</table>

All results, Analysis of classified user sessions in paper
Background:
Test Case Generation Process

Test Case: sequence of pages that is accessed; independent of parameter values

Navigation model specification

Intra-session navigation analyzer

Intra-session navigation model

Abstract test case generator

Abstract test case suite

Generation criteria

Data model specification

User Sessions

User session parameter analyzer

Data Model

Test-case generator

Test suite

Sara Sprenkle
Intra-session Navigation Model
Configuration Parameters

- **Parameter: Amount of history (n)**
  - Number (-1) of previous requests on which to base next request
  - **Tradeoffs:** As n increases
    - More representative of usage
    - Model size increases (may become too large)
    - Restricts generated test cases to sequences in collected USs

[Sprenkle et al., ICST2011]
Intra-session Navigation Model

Configuration Parameters

Parameter: Amount of abstraction to represent request

- **RRN**: request type, resource, and parameter names
  
  ✓ Represents parameter names, often part of navigation
  
  - need to design additional models of parameter values

- **RRNV**: include parameter values
  
  ✓ Represent requests using one model
  
  - model may become too large, restricts generated test cases to data in collected USs

---

**Sprenkle et al., ICST2011**

[Sara Sprenkle](#)
Open Questions: Effect of CbUP on Models

- Compare general-usage models to user-privilege specific models

- Expect smaller models, but how much smaller?
  - Fewer user sessions, less diversity in what users do

- What is the impact of the smaller models on model configuration decisions?
Answering Question: How much smaller are the models?
Answering Question: How much smaller are the models?

In general, reduction of RRN and RRRN models was similar. No general trends in terms of $n$. RRNV models are flat as $n$ increases.
Implications: How much smaller are the models?

- In many cases, CbUP resulted in significant (>50%) reduction in RRN and RRNV model size
  - Not necessarily in relation to reduction in original sessions or requests

- If sufficiently small, a tester could choose to use larger $n$, RRNV to generate test cases
  - RRNV tends to be a larger model, but does not have the cost of an additional data model
Open Question

- How similar are the models generated from the different privileges?
  - If similar enough, no need to create different model
Answering Question:
How similar are the models?

Q2: Similar model, Accessed frequently – less incentive for user-privilege specific models

A point represents comparing two user-privilege specific models
Answering Question: How similar are the models?

Each point represents comparing two user-privilege specific models

May not make sense to create user-privilege-specific models for Book
What About the Generated Abstract Test Cases?

Navigation model specification

Intra-session navigation analyzer

Intra-session navigation model

Abstract test case generator

Abstract test case suite

Generation criteria

User Sessions

Classified by user privilege

User session parameter analyzer

Data Model

Test-case generator

Test suite

Data model specification

User Sessions

Classified by user privilege

User session parameter analyzer

Data Model

Test-case generator

Test suite

Sara Sprenkle
Analyzing the resulting abstract test cases

- Are the models too small to generate “interesting” test cases?
  - Short answer: in most cases, no
    - Models weren’t too small
    - Generated sequences not seen in the original user sessions
Conclusions

- Classifying user sessions by user privilege
  - Possible, easy to automate
  - Results in smaller models and test cases that represent different groups of users
    - Reduced size may mean that a tester can use a less abstract model

- Testers can use some simple metrics to determine if classifying by user privilege makes sense for them
Future Work

Navigation model specification

User Sessions

Intra-session navigation analyzer

Intra-session navigation model

Abstract test case generator

Abstract test case suite

Generation criteria

• Alternative user session partitions
• Apply to other usage-based testing approaches

Recommend number of test cases generated for each user privilege, all sessions

Data model specification

User session parameter analyzer

Data Model

Test-case generator

Test suite