EventBreak: Analyzing the Responsiveness of Web Applications

Michael Pradel, Parker Schuh, George Necula, Koushik Sen
University of California, Berkeley
Motivation

Event-based UI applications should be responsive

single thread of execution
Event-based UI applications should be responsive
Motivation

Event-based UI applications should be responsive
Motivation

Event-based UI applications should be responsive

Single thread of execution
Motivation

Event-based UI applications should be responsive
Real-World Example

My great site

Getting Started

It's easy to get started creating your website. Knowing some of the basics will help.

What is a Content Management System?

A content management system is software that allows you to create and manage webpages easily by separating the creation of your content from the mechanics required to present it on the web.

In this site, the content is stored in a database. The look and feel are created by a template. Joomla! brings together the template and your content to create web pages.
Real-World Example

My great site

Getting Started

What is a Content Management System?

A content management system is software that allows you to create and manage webpages easily by separating the creation of your content from the mechanics required to present it on the web.

In this site, the content is stored in a database. The look and feel are created by a template. Joomla brings together the template and your content to create web pages.
Real-World Example
Real-World Example

Joomla bug 30274
Real-World Example

Joomla bug 30274
Real-World Example

Joomla bug 30274
Real-World Example

Joomla bug 30274
Real-World Example

Unresponsive

Joomla bug 30274
Real-World Example

Cost plot for responsiveness problem

Cost of saving menu

Number of menu items

Joomla bug 30274
Real-World Example

Cost plot for responsiveness problem

Number of menu items
Cost of saving menu

Unbounded growth: Unresponsive application

Joomla bug 30274
Goal

Analyze responsiveness of web applications through automated testing

Focus: Slowdown pairs

Event $E_{cause}$ increases cost of event $E_{effect}$
Overview

Dynamic analysis of application

Event-cost history

Infer potential slowdown pairs

Infer finite state model of application

Targeted test generation:
Verify slowdown pairs

Slowdown pairs with cost plots
Overview

Event-cost history

Dynamic analysis of application

Infer potential slowdown pairs

Infer finite state model of application

Targeted test generation:
Verify slowdown pairs

Slowdown pairs with cost plots
Event-Cost History

Sequence of **event-cost** pairs

- DOM element
- Type of event
- Pre-state
- Post-state

Number of conditionals evaluated in event handler
# Potential Slowdown Pairs

Does A increase cost of B?

<table>
<thead>
<tr>
<th>Event</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>5</td>
</tr>
<tr>
<td>A</td>
<td>3</td>
</tr>
<tr>
<td>B</td>
<td>10</td>
</tr>
<tr>
<td>B</td>
<td>12</td>
</tr>
<tr>
<td>A</td>
<td>3</td>
</tr>
<tr>
<td>B</td>
<td>12</td>
</tr>
</tbody>
</table>

Supporting evidence $S$

Refuting evidence $R$

$Supp = |S| = 1$

$Conf = \frac{|S|}{|S| + |R|} = 33\%$
Targeted Test Generation

Confirm or reject slowdown pair:
Alternate between $E_{effect}$ and $E_{cause}$
Targeted Test Generation

Confirm or reject slowdown pair:
Alternate between $E_{effect}$ and $E_{cause}$

$E_{cause} = \text{Save new item}$

$E_{effect} = \text{Save menu}$

Costs of $E_{effect}$:

- Current state
- Target event
Targeted Test Generation

Confirm or reject slowdown pair:
Alternate between $E_{effect}$ and $E_{cause}$

$E_{cause} = \text{Save new item}$

$E_{effect} = \text{Save menu}$

Costs of $E_{effect}$:

- Current state
- Target event
Targeted Test Generation

Confirm or reject slowdown pair:
Alternate between $E_{\text{effect}}$ and $E_{\text{cause}}$

$E_{\text{cause}} = \text{Save new item}$

$E_{\text{effect}} = \text{Save menu}$

Costs of $E_{\text{effect}}$:

- Current state
- Target event
Targeted Test Generation

Confirm or reject slowdown pair:
Alternate between $E_{effect}$ and $E_{cause}$

$E_{cause} = \text{Save new item}$

$E_{effect} = \text{Save menu}$

Costs of $E_{effect}$:
5

current state → target event
Targeted Test Generation

Confirm or reject slowdown pair:
Alternate between $E_{\text{effect}}$ and $E_{\text{cause}}$

$E_{\text{cause}} = \text{Save new item}$

$E_{\text{effect}} = \text{Save menu}$

Costs of $E_{\text{effect}}$: 5
Targeted Test Generation

Confirm or reject slowdown pair:
Alternate between $E_{effect}$ and $E_{cause}$

$E_{cause} = \text{Save new item}$

$E_{effect} = \text{Save menu}$

Costs of $E_{effect}$:
5

current state —- target event
Targeted Test Generation

Confirm or reject slowdown pair:
Alternate between $E_{\text{effect}}$ and $E_{\text{cause}}$

$E_{\text{cause}} = \text{Save new item}$

$E_{\text{effect}} = \text{Save menu}$

Costs of $E_{\text{effect}}$:
5

current state

target event
Targeted Test Generation

Confirm or reject slowdown pair:
Alternate between $E_{effect}$ and $E_{cause}$

$E_{cause} = \text{Save new item}$

$E_{effect} = \text{Save menu}$

Costs of $E_{effect}$:
5

current state \quad target event
Targeted Test Generation

Confirm or reject slowdown pair:
Alternate between $E_{effect}$ and $E_{cause}$

$E_{cause} = \text{Save new item}$

$E_{effect} = \text{Save menu}$

Costs of $E_{effect}$: 5, 10, 15, etc.
Navigation: Challenge

Shortest event sequence in model may be infeasible in application
Navigation: Challenge

Shortest event sequence in model may be infeasible in application

\[ E_{\text{cause}} = \text{Save new item} \]

\[ E_{\text{effect}} = \text{Save menu} \]
Navigation: Challenge

Shortest event sequence in model may be infeasible in application

$E_{cause} = \text{Save new item}$

$E_{effect} = \text{Save menu}$

Insert text, Save new item

current state target event
Navigation: Challenge

Shortest event sequence in model may be infeasible in application

$E_{cause} = \text{Save new item}$

$E_{effect} = \text{Save menu}$

Insert text, Save new item
Navigation: Approach

- Compute sequences to target event
- Randomly pick from set of first steps

\[ E_{cause} = \text{Save new item} \]

\[ E_{effect} = \begin{cases} \text{Save menu} \\ \text{Insert text, Save new item} \end{cases} \]
Evaluation

Find responsiveness problems?
- 2 known + 4 previously unknown

Effectiveness of targeted test generation?
- Reaches 89% of all target events
- Invalidates > 99% of all potential slowdown pairs
Results: Joomla

Joomla bug 30274
Results: Joomla

Several similar examples:

- $E_{cause}$ accumulates data items
- $E_{effect}$ processes all of them and has unbounded cost

Joomla bug 30274
Results: Drupal
Results: Drupal

MySQL configuration bug: Crash
Results: Drupal (2)

Bounded cost of event handler
Limitations

- False negatives
  - Bounded by initial execution
  - Focus on pairs of events

But: No false positives

- Explores event space, ignores input space
Conclusion

Automated analysis of web application responsiveness

- Slowdown pairs
- Targeted test generation for event-driven applications

https://github.com/michaelpradel/WebAppWalker
EventBreak:
Analyzing the Responsiveness of Web Applications

Michael Pradel, Parker Schuh, George Necula, Koushik Sen

University of California, Berkeley

I’m looking for students to join my group at TU Darmstadt!