Understanding JavaScript
Event-Based Interactions

Saba Alimadadi
Sheldon Sequeira
Ali Mesbah
Karthik Pattabiraman
Motivation

• JavaScript
  – Widely used, very popular
  – Event driven, dynamic, asynchronous

• Difficult to understand the dynamic behavior and the control flow
  – Lower level events
  – Their interactions
Challenge 1: Event Propagation
Challenge 2: Asynchronous Events

Timeout for page expiry
Server request for login
Server response for login
Challenge 2: Asynchronous Events

Timeout for page expiry
Server request for login
Server response for login
Server request
Server request
Server response
Server response
Challenge 2: Asynchronous Events

Timeout for page expiry
Server request for login
Server response for login
Server request
Server request
Server response
Server response
Timeout for next image
Challenge 2: Asynchronous Events

- **Timeout for page expiry**
- Server request for login
- Server response for login
- Server request
- Server request
- Server response
- Server response
- Timeout for next image
- Server request image
- Server response
- Timeout callback
- Timeout callback page expiry
function submissionHandler(e) {
    $('#regMsg').html("Submitted!");
    var email = $('#email').val();
    if (isEmailValid(email)) {
        informServer(email);
        $('#submitBtn').attr("disabled", true);
    }
}

function informServer(email) {
    $.get('/register/', { email }
        , function(data) {
            $('#srvrMsg').append(data);
        });
}
Summary of Challenges

• Event propagation

• Asynchronous events

• Implications of events
Approach

- JavaScript Transformation
- Trace Collection
- Model Visualization
- Behavioral Model Creation
JavaScript Transformation

- Interposing on DOM events
- Capturing timeouts and XHRs
- Recording function traces
- Extracting DOM mutations
Trace Collection

- Interposing on DOM events
- Capturing timeouts and XHRs
- Recording function traces
- Extracting DOM mutations

=> Detailed Trace

DOM events
functions
timeouts
XHRs
DOM mutations
Behavioral Model Creation

- Customized graph
- Nodes: episodes
- Links: temporal and causal
Model: Episodes

- A period of JavaScript execution
- Start and end points
Model: Links

Temporal
Causal
Model: Story
Visualization: Overview
Visualization: Zoom Level 1

Source
*click*

Trace
- Event type: click
- ss_update()
- hideElem()
- inlineElem()
- ss_updateNumOfLoads()
- storeUserInformation()
- sendStatsToServer()
- onload()

Dom Mutations
- "text" "removed"
- "text" "removed"
- "text" "added"
- "text" "added"

Episode #3 Event

Source
- TO: 0

Trace
- hideElem()
- ss_slideshow()
- ss_update()

Episode #7 Event

Source
- TO: 0

Trace
- ss_slideshow()
- ss_update()
- hideElem()
- onload()
- ss_updateNumOfLoads()
- storeUserInformation()
- sendStatsToServer()
- onload()
Visualization: Zoom Level 2

```javascript
function ss_update() {
    ss_cur = Math.max(ss_cur, 0);
    if (ss_cur >= ss_date.length) {
        hideElem('ss_link2');
        showElem('ss_theend');
        ss_cur = ss_date.length;
        var a = dg('ss_n');
        a.innerHTML = "Final";
        if (ss_play)
            ss_playpause();
    }
}
```
Implementation

• **Clematis**
  [https://github.com/saltlab/clematis](https://github.com/saltlab/clematis)

• Languages: Java, JavaScript
• Transform JavaScript & inject toolbar via proxy
• Provide a RESTful API for retrieving data
• Render a web-based visualization
Usage Scenario

![My Photo Gallery](image)
Usage Scenario
Usage Scenario
Usage Scenario
Evaluation

RQ1) Does using Clematis decrease the task completion duration for web application comprehension?

RQ2) Does using Clematis increase the task completion accuracy for web application comprehension?

RQ3) Are there any certain categories of tasks for which Clematis improves the performance most?
Industrial Controlled Experiment

• Participants
  – 20 software developers (from a large SW company)
  – Experimental group: Clematis
  – Control group: Chrome, Firefox & Firebug

• Procedure
  – 5 minute tutorial on Clematis
  – Tasks: control flow, feature location, DOM mutations, ...

• Data collection
  – Task completion duration & accuracy
## Results: Duration

### Average Time (mm:ss) Per Task

<table>
<thead>
<tr>
<th>Task</th>
<th>Clematis</th>
<th>Other</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>7:00</td>
<td>&lt;&lt;</td>
<td>11:27</td>
</tr>
<tr>
<td>T2</td>
<td>3:51</td>
<td>&lt;&lt;</td>
<td>7:27</td>
</tr>
<tr>
<td>T3</td>
<td>2:02</td>
<td>&lt;&lt;</td>
<td>6:18</td>
</tr>
<tr>
<td>T4</td>
<td>2:44</td>
<td>&lt;</td>
<td>4:00</td>
</tr>
</tbody>
</table>

### Average Time (mm:ss) in Total

<table>
<thead>
<tr>
<th>Task</th>
<th>Clematis</th>
<th>Other</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>15:37</td>
<td>&lt;&lt;</td>
<td>29:12</td>
</tr>
</tbody>
</table>
Results: Accuracy

Average Accuracy (%) Per Task

<table>
<thead>
<tr>
<th>Task</th>
<th>Clematis</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>84</td>
<td>28</td>
</tr>
<tr>
<td>T2</td>
<td>97</td>
<td>57</td>
</tr>
<tr>
<td>T3</td>
<td>100</td>
<td>80</td>
</tr>
<tr>
<td>T4</td>
<td>95</td>
<td>30</td>
</tr>
</tbody>
</table>

Average Accuracy (%) in Total

<table>
<thead>
<tr>
<th>Task</th>
<th>Clematis</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>90</td>
<td>35</td>
</tr>
</tbody>
</table>
## Results

### Duration

<table>
<thead>
<tr>
<th>Task</th>
<th>Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>(39%↑)</td>
</tr>
<tr>
<td>T2</td>
<td>(48%↑)</td>
</tr>
<tr>
<td>T3</td>
<td>(68%↑)</td>
</tr>
<tr>
<td>T4</td>
<td>(32%↑)</td>
</tr>
</tbody>
</table>

### Accuracy

<table>
<thead>
<tr>
<th>Task</th>
<th>Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>(67%↑)</td>
</tr>
<tr>
<td>T2</td>
<td>(41%↑)</td>
</tr>
<tr>
<td>T3</td>
<td>(20%↑)</td>
</tr>
<tr>
<td>T4</td>
<td>(68%↑)</td>
</tr>
</tbody>
</table>

### Task Description

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>Following control flow in presence of asynchronous events</td>
</tr>
<tr>
<td>T2</td>
<td>Finding DOM mutations caused by a DOM event</td>
</tr>
<tr>
<td>T3</td>
<td>Locating the implementation of a malfunctioning feature</td>
</tr>
<tr>
<td>T4</td>
<td>Detecting control flow in presence of event propagation</td>
</tr>
</tbody>
</table>
Consistent Performance

Duration (mm:ss)

Accuracy (%)

T1 – Ctrl
T1 – Exp
T2 – Ctrl
T2 – Exp
T3 – Ctrl
T3 – Exp
T4 – Ctrl
T4 – Exp
Total – Ctrl
Total – Exp

T7 – Ctrl
T7 – Exp
T8 – Ctrl
T8 – Exp
T9 – Ctrl
T9 – Exp
T10 – Ctrl
T10 – Exp
Total – Ctrl
Total – Exp

0:00
8:20
16:40
25:00
33:20
41:40
50:00

0
20
40
60
80
100

SALT LAB
Understanding JavaScript Event-Based Interactions

Saba Alimadadi Sheldon Sequeira Ali Mesbah Karthik Pattabiraman

Electrical and Computer Engineering
University of British Columbia
Vancouver, BC, Canada
{saba, sheldon, amesbah, karthikp}@ece.ubc.ca

Approach

JavaScript Transformation → Trace Collection

Model Visualization ← Behavioral Model Creation

Model: Episodes

- A period of JavaScript execution
- Start and end points

Visualization: Zoom Level 1

Consistent Performance

Duration (mm:ss) Accuracy (%)