Hybrid DOM-Sensitive Change Impact Analysis for JavaScript

Saba Alimadadi, Ali Mesbah and Karthik Pattabiraman
ECOOP 2015
saba@ece.ubc.ca
Change Impact Analysis (CIA)

• Software must continually change to adapt to the changing environment.

• Goal: identifying parts of the program that are potentially affected by a change.
Challenges of CIA for JavaScript
Challenge 1: Impact through the DOM

```javascript
function calculateTax() {
  $('.price').each(function(index) {
    $(this).text(addTaxToPrice(
      $(this).text());
  });
}

function checkPrice() {
  . . .
  var cad_price = $('#price_ca').innerText();
  . . .
}
```
Challenge 2: Impact through Event Propagation

```
function checkPrice() {
    ...
}

function calculateTax() {
    ...
}

$('#price_ca').bind('click', checkPrice);
$('prices').bind('click', calculateTax);
```
### Challenge 3: Impact through XHRs

```javascript
function checkPrice() {
    var itemName = extractName($('item231'));
    $.ajax({
        url: 'prices/latest.php',
        type: 'POST',
        data: itemName,
        success: eval(getAction() + 'item')
    });
}

function updateItem(xhr) {
    var updatedInfo = getUpdatedPrice(xhr.responseText);
    suggestItem.apply(this, updatedInfo);
}
```
Challenges of CIA for Client-Side JavaScript

1. JavaScript and Document Object Model (DOM)

2. Events and event propagation

3. JavaScript and XMLHttpRequests (XHRs)

+ High dynamism of JavaScript
Exploratory Study: DOM-related and Event-based Impacts

• Subject: 10 web apps (online contests, GitHub trending, etc.)

• Gathered data:
  – JavaScript-DOM interactions (write-read pairs)
  – Event propagation

• Further analysis of the structure of graphs.
  – Measured metrics: fan-in and fan-out of functions and DOM elements, and average path lengths
Exploratory Study: Results

- **W-R DOM elements:** 42%
- **Propagated handlers:** 14%

![DOM Elements](chart)

```
foo() -> elem -> bar()
```
Hybrid Analysis

Static Analysis

Dynamic Analysis

Hybrid Model

Ranked Impact Sets
Static Analysis

Control (and data) dependencies

Partial data-flow analysis
Example: Static Model

Function
XHR object
DOM element
Directed and labeled edge

checkPrice()

XHR

getUpdatedPrice()  updateItem()  suggestItem()  calculateTax()  addTaxToPrice()
Dynamic Analysis

Impact through events
XHR relations (open, send, response)

Impact through DOM
Dynamic call graph
JavaScript dynamism (eval(), function variadicity)
Example: Dynamic Model

- checkPrice()
- XHR
- updateItem()
- suggestItem()
- getUpdatedPrice()
- calculateTax()
- displaySuggestion()
- #item231
- #price-ca.price

Diagram:
- Function
- XHR object
- DOM element
- Directed and labeled edge
Static Analysis

Dynamic Analysis

Hybrid Model

Ranked Impact Sets
Example: Hybrid Analysis

Vertices

- Function
- XHR object
- DOM element
- Directed and labeled edge

Edges

- checkPrice()
- #item231
- XHR
- #price-ca .price
- calculateTax()
- addItem()
- suggestItem()
- addTaxToPrice()
- getUpdatedPrice()
- displaySuggestion()
Static Analysis → Hybrid Model → Ranked Impact Sets
Dynamic Analysis → Hybrid Model → Ranked Impact Sets
Impact Metrics and Impact Set Ranking

• Problem: size of impact sets
• Solutions: impact ranks, based on impact metrics
  – $f_{in}(d)$: Number of functions $f$ such that $f \ W d$
  – $f_{in}(f)$: Number of elements $d$ such that $f \ R d$
  – $f_{out}(f)$: Number of elements $d$ such that $f \ W d$
  – $L_{avg}(P)$: Average length of impact paths in the app
  – $D_m(e)$: Minimum distance of $e$ from the change set
  – $IR_{pr}(e)$: Impact of previous entity
Tool Implementation: Tochal

- Tochal: open source
  - [https://github.com/saltlab/tochal](https://github.com/saltlab/tochal)
- Proxy (Java, JavaScript)
  - Esprima, Estraverse, Escodegen, Mutation Summary, WALA
- Client-side (Google Chrome extension)
  - Chrome DevTools
Research Question 1

Does Tochal outperform static and dynamic analysis methods in terms of the completeness of the results obtained?
Study: Static vs. Dynamic vs. Tochal

- 10 web applications
- 3 random functions as change sets
- Comparing:
  - Size of impact sets
  - Number of functions in dependency graphs
Results: Impact Sets

- Comparing size of impact sets

<table>
<thead>
<tr>
<th>Type</th>
<th>Static Hybrid</th>
<th>Dynamic Hybrid</th>
</tr>
</thead>
<tbody>
<tr>
<td>Static</td>
<td>26%</td>
<td></td>
</tr>
<tr>
<td>Dynamic</td>
<td>80%</td>
<td></td>
</tr>
</tbody>
</table>

[Bar chart showing comparison of Static, Dynamic, and Hybrid impact sets, with 'Static' at 26%, 'Dynamic' at 80%, and 'Hybrid' at 26% for Static and 80% for Dynamic]
Results: Graphs

• Comparing size of model graphs

<table>
<thead>
<tr>
<th></th>
<th>Static Hybrid</th>
<th>Dynamic Hybrid</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>59%</td>
<td>84%</td>
</tr>
</tbody>
</table>
Results: Graphs

• Comparing size of model graphs

Pure Static

Hybrid : 15%

Pure Dynamic

Hybrid : 42%
Research Question 2

Does Tochal help developers in practice to perform change impact analysis?
Experiment: Design

- 12 participants from industry

- Performed 4 tasks

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>Finding the potential impact of a DOM element</td>
</tr>
<tr>
<td>T2</td>
<td>Finding the potential impact of a JavaScript function</td>
</tr>
<tr>
<td>T3</td>
<td>Finding a conflict after making a new change (no ranking)</td>
</tr>
<tr>
<td>T4</td>
<td>Finding a bug in JavaScript code</td>
</tr>
</tbody>
</table>

- We measured task completion duration and accuracy
User Experiment: Results

Duration

Accuracy

80% faster

2 times more accurate
Results: Ranking

Duration

Accuracy

2~3 times faster

25% more accurate
Challenges of CIA for Client-Side JavaScript

1. JavaScript and Document Object Model (DOM)
2. Events and event propagation
3. JavaScript and XMLHttpRequests (XHRs)
   + High dynamism of JavaScript

Exploratory Study: Results

- W-R DOM elements: 42%
- Propagated handlers: 14%

User Experiment: Results

- Accuracy: 2 times more accurate
- Duration: 80% faster