Auto-Locating and Fix-Propagating for HTML Validation Errors to PHP Server-side Code

Hung Viet Nguyen, Hoan Anh Nguyen, Tung Thanh Nguyen, Tien N. Nguyen
Electrical and Computer Engineering Department
Iowa State University
{hungnv,hoan,tung,tien}@iastate.edu
Motivating Example
Example of a Buggy Web Page

Server Side

Client Side

ASE 2011
Submission is now open.
Hung Nguyen: Great news!

Post a new comment
Example of a Buggy Web Page

One bug found: Comment is placed below the Comment button.
Locating the Bug in the Client Code

Client Side

View source

HTML validation error: Missing </div>.
Locating the Bug in the Client Code

Client Side

```
11 <div id='divComments' class='out'>
12   <div class='inComment'>Hung Nguyen:
13   <div class='inComment'></div>
14   </div>
15 <div class='out'>
16   <input id='txtComment' type='text'>
```
Locating the Bug in the Server Code

Client Side

Server Side
Locating the Bug in the Server Code

Client Side

Server Side

Root cause in PHP code: Missing </div>.
Our Framework for Fixing Client-Side Errors

1. Locating & Fixing Errors in Client Code
3. Locating & Fixing Errors in Server Code
PhpSync for HTML Validation Errors

HTML Validation Tool

Error Auto-Locator & Fix Propagator
Challenges in Client-Server Mapping

Client code is

- embedded in PHP strings
- scattered in different locations/files/functions
- generated from different execution paths
PhpSync’s Solution to Client-Server Mapping

Server Side

Client Side

Client-Server Mapping

Mapping to server code

Symbolic Execution

D-model

Mapping to D-model

Represents all possible client pages
D-model Representation
D-model

• Is a tree-based structure
• Represents the string values of PHP variables
  – D-model of a server program represents all possible versions of its client page.
• Contains 5 types of nodes:
  - Literal Node
  - Symbolic Node
  - CONCAT
  - SELECT
  - REPEAT
if (...) 
  $color = 'blue';
else 
  $color = 'green';
if (...) {
    $color = 'blue';
} else {
    $color = 'green';
}
PHP code

```php
if (...) {
    $color = 'blue';
} else {
    $color = 'green';
}
```

Diagram:

- SELECT
- $color
- blue
- green
- Select Node
### D-model Example

**PHP code**

```php
if (...) {
    $color = 'blue';
} else {
    $color = 'green';
}

echo '<div class="'. $color . '"></div>

echo ...

echo '</div>';```

**Version 1**

```html
<div class="blue">...
</div>
```

**Version 2**

```html
<div class="green">...
</div>
```
D-model Example

```php
for ($i = 0; $i < $rowCount; $i++) {
    if ($i % 2 == 0)
        $color = 'blue';
    else
        $color = 'green';
    echo '<div class="' . $color . '">' . getRow($i) . '</div>;
}
```
Client-Server Mapping
Client-Server Mapping via D-model

**PHP code**

```php
if (...) {
    $color = 'blue';
} else {
    $color = 'green';
}
```

**Client page**

```html
<div class="blue">
    ...
</div>
```

**Diagram**

- **SELECT**
- **CONCAT**
- **Mapping**
- Blue
- Green

The PHP code checks if a condition is true. If it is true, `$color` is set to 'blue'. Otherwise, it is set to 'green'. The code then uses `echo` to print the color in a div element with the appropriate class.
Mapping Client Code to D-model

Client code

Expected mapping

```
<body>  <div>  blue  </div>  <div>  green  </div>  </body>
```

```
<body>  REPEAT  CONCAT  </body>
```

```
<body>  SELECT  CONCAT  blue  green  </body>
```
2 Strategies

1. Divide-and-conquer
2. Local best-matching
Divide-and-Conquer
A pivot is the string value of a D-model literal node that appears exactly once in the client code.
Map the fragments split by pivots to the corresponding D-models

Divide-and-Conquer: Splitting into fragments
When there is no pivot to divide the client code, mapping is done from left to right.
Mapping Client Code to D-model
Local best-matching

Select the better mapping (the one with more mapped text)
Local best-matching
Mapping Client Code to D-model

... <body> <div> blue </div> <div> green </div> </body> ...

REPEAT

CONCAT

SELECT

blue

green
Mapping Client Code to D-model
Mapping Client Code to D-model
Mapping Client Code to Symbolic Nodes

Unmapped text

Data from DB: ASE 2011

CONCAT

Symbolic Node

Data from DB: dbQuery()
 CLIENT-SERVER MAPPING REVISITED

PHP code:

```php
if (...) {
    $color = 'blue';
} else {
    $color = 'green';
}
```

Client page:

```html
<div class="blue">
    ...  
</div>
```

Diagram:

- SELECT
- CONCAT
- <div class="blue">
- else
- $color = 'green';
- echo '<div class="' . $color . '"'>';
- echo ...'
- echo '</div>';
- blue
- green
Auto-Locating Errors and Propagating Fixes

Client Side

```
13 <div id='divComments' class='out'>
12   <div class='inComment'>Hung Nguyen:
13     <div class='inComment'></div>
14   </div>
15 <div class='out'>
16     <input id='txtComment' type='text'>
```

Server Side

```
function addComments($comments) {
    $output = "<div id='divComments' class='out'>";
    foreach ($comments as $comment) {
        $output .= "\n" . addComment($comment);
    }
    $output .= "\n";  // Missing </div>
    return $output;
}
```

 Locating the faulty tag

```
\n</div>
```

Propagating the fix

```
Evaluation Results
Goals of Evaluation

• Accuracy in Client-Server Mapping
• Accuracy in Fix-Propagation
• Time Efficiency
## Subject Systems

<table>
<thead>
<tr>
<th>Subject System</th>
<th>Files</th>
<th>KLOC</th>
</tr>
</thead>
<tbody>
<tr>
<td>SchoolMate-1.5.4</td>
<td>63</td>
<td>8</td>
</tr>
<tr>
<td>AddressBook-6.2.12</td>
<td>103</td>
<td>19</td>
</tr>
<tr>
<td>TimeClock-1.4</td>
<td>69</td>
<td>23</td>
</tr>
<tr>
<td>Manhali-1.3.2</td>
<td>299</td>
<td>52</td>
</tr>
<tr>
<td>UPB-2.2.7</td>
<td>400</td>
<td>105</td>
</tr>
<tr>
<td>WebERP-4.0.2</td>
<td>654</td>
<td>220</td>
</tr>
</tbody>
</table>
# Mapping Results on SchoolMate-1.5.4

<table>
<thead>
<tr>
<th>#</th>
<th>Test Page</th>
<th># Characters</th>
<th>Correct</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Terms</td>
<td>12,200</td>
<td>12,100</td>
<td>99%</td>
</tr>
<tr>
<td>2</td>
<td>School</td>
<td>11,8000</td>
<td>11,800</td>
<td>100%</td>
</tr>
<tr>
<td>3</td>
<td>Login</td>
<td>7,000</td>
<td>7,000</td>
<td>100%</td>
</tr>
<tr>
<td>4</td>
<td>Semester</td>
<td>12,500</td>
<td>12,400</td>
<td>99%</td>
</tr>
<tr>
<td>5</td>
<td>Classes</td>
<td>12,900</td>
<td>12,900</td>
<td>100%</td>
</tr>
<tr>
<td>6</td>
<td>Users</td>
<td>13,100</td>
<td>12,900</td>
<td>99%</td>
</tr>
<tr>
<td>7</td>
<td>Teachers</td>
<td>12,000</td>
<td>12,000</td>
<td>100%</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>Users/Edit</td>
<td>10,600</td>
<td>10,300</td>
<td>98%</td>
</tr>
<tr>
<td></td>
<td><strong>Sum</strong></td>
<td><strong>236,900</strong></td>
<td><strong>235,800</strong></td>
<td><strong>99.5%</strong></td>
</tr>
</tbody>
</table>
## Mapping Results on All Systems

<table>
<thead>
<tr>
<th>System</th>
<th>Tests</th>
<th>All</th>
<th>Correct</th>
<th>Accuracy</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SchoolMate-1.5.4</td>
<td>21</td>
<td>236.9</td>
<td>235.8</td>
<td>99.5%</td>
<td>4.4</td>
</tr>
<tr>
<td>AddressBook-6.2.12</td>
<td>10</td>
<td>78.9</td>
<td>77.9</td>
<td>98.1%</td>
<td>0.3</td>
</tr>
<tr>
<td>TimeClock-1.4</td>
<td>14</td>
<td>164.1</td>
<td>162.7</td>
<td>99.1%</td>
<td>1.2</td>
</tr>
<tr>
<td>Manhali-1.3.2</td>
<td>10</td>
<td>115.1</td>
<td>105.9</td>
<td>92.0%</td>
<td>3.8</td>
</tr>
<tr>
<td>UPB-2.2.7</td>
<td>10</td>
<td>56.7</td>
<td>53.9</td>
<td>95.0%</td>
<td>0.6</td>
</tr>
<tr>
<td>WebERP-4.0.2</td>
<td>10</td>
<td>78.5</td>
<td>75.8</td>
<td>96.7%</td>
<td>8.0</td>
</tr>
</tbody>
</table>
## Fix-Propagation Results on SchoolMate-1.5.4

<table>
<thead>
<tr>
<th>#</th>
<th>Test Page</th>
<th># Errors by Tidy</th>
<th># Fixes by Tidy</th>
<th># Fixes by PhpSync</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Terms</td>
<td>52</td>
<td>22</td>
<td>22</td>
</tr>
<tr>
<td>2</td>
<td>School</td>
<td>50</td>
<td>19</td>
<td>19</td>
</tr>
<tr>
<td>3</td>
<td>Login</td>
<td>38</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>Semester</td>
<td>50</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>5</td>
<td>Classes</td>
<td>57</td>
<td>27</td>
<td>27</td>
</tr>
<tr>
<td>6</td>
<td>Users</td>
<td>64</td>
<td>34</td>
<td>34</td>
</tr>
<tr>
<td>7</td>
<td>Teachers</td>
<td>50</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>Users/Edit</td>
<td>48</td>
<td>19</td>
<td>19</td>
</tr>
<tr>
<td>Sum</td>
<td></td>
<td>1041</td>
<td>411</td>
<td>411</td>
</tr>
</tbody>
</table>
## Fix-Propagation Results on All Systems

<table>
<thead>
<tr>
<th>System</th>
<th>Tests</th>
<th># Errors</th>
<th># Fixes by Tidy</th>
<th># Fixes by PhpSync</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SchoolMate-1.5.4</td>
<td>21</td>
<td>1041</td>
<td>411</td>
<td>100.0%</td>
<td>&lt; 0.2</td>
</tr>
<tr>
<td>AddressBook-6.2.12</td>
<td>10</td>
<td>64</td>
<td>51</td>
<td>94.1%</td>
<td>&lt; 0.2</td>
</tr>
<tr>
<td>TimeClock-1.4</td>
<td>14</td>
<td>422</td>
<td>136</td>
<td>100.0%</td>
<td>&lt; 0.2</td>
</tr>
<tr>
<td>Manhali-1.3.2</td>
<td>10</td>
<td>607</td>
<td>189</td>
<td>86.8%</td>
<td>&lt; 0.2</td>
</tr>
<tr>
<td>UPB-2.2.7</td>
<td>10</td>
<td>129</td>
<td>49</td>
<td>95.9%</td>
<td>&lt; 0.2</td>
</tr>
<tr>
<td>WebERP-4.0.2</td>
<td>10</td>
<td>284</td>
<td>188</td>
<td>93.6%</td>
<td>&lt; 1.0</td>
</tr>
</tbody>
</table>
Related Work

**Apollo**
*(Shay Artzi et al., ICSE 2010)*

- Uses dynamic instrumentation
- Requires user inputs to reproduce errors
- Maps client code to echo/print statements

**PhpSync**

- Uses symbolic execution
- Does not require user inputs to reproduce errors
- Maps client code to their original locations
Comparison with Apollo

50% of the client’s contents are mapped to locations other than `echo/print` statements.

The misspelled word “blu” can be corrected using PhpSync.
Conclusions

1. D-model
   - Representation of client pages

2. Client-Server Mapping
   - via D-model

3. Auto-Locating Errors
   - Propagating Fixes