# A Framework for Static Bug Detection in JavaScript Web Applications in the Wild

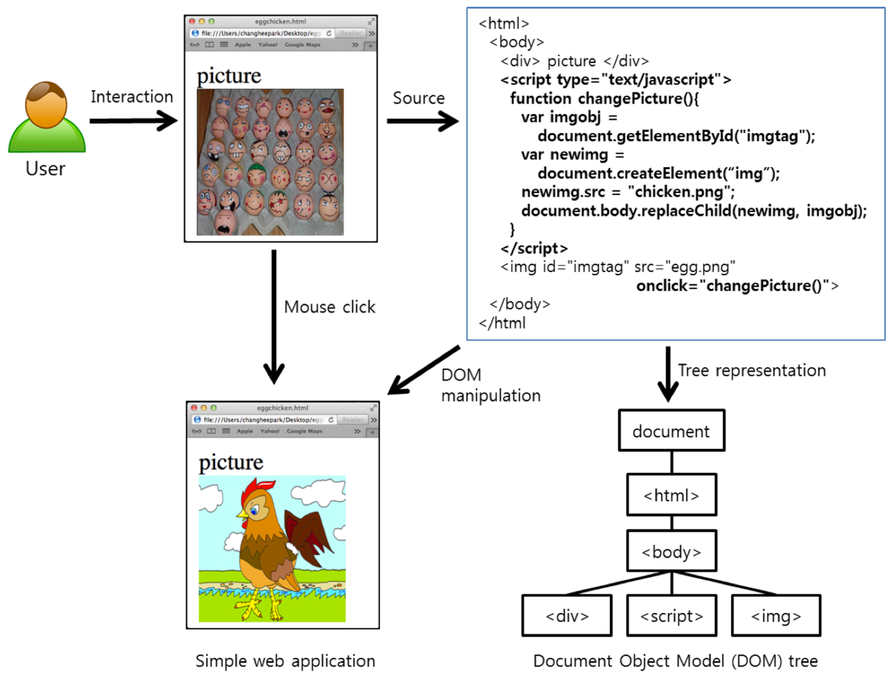
Changhee Park (KAIST) Sooncheol Won (KAIST) Joonho Jin (KAIST) Jaejoon Choi (S-Core., Ltd.) Sukyoung Ryu (KAIST)

## Overview

• $SAFE_{Wapp}$ : a static analysis framework for bug detection in real-world JavaScript web applications built on the SAFE framework

• Features of $SAFE_{Wapp}$
- Browser environment modeling based on extensive empirical data including a precise tree structure of the DOM, the event system, and almost all the browser objects and APIs found in more than 1000, web sites
- Capability of detecting 83.2% of all 155 errors defined in the ECMAScript standard

## JavaScript Execution Model in Web Applications




## Static Analysis of JavaScript Web Applications

### Requirement

• `Modeling`
- DOM tree
- Browser APIs : document.getElementById, document.createElement, …
- Event system : mouse events, keyboard events, ...

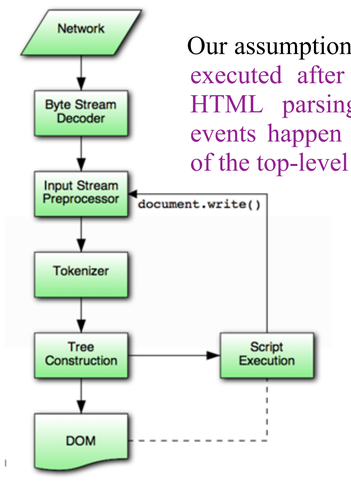
### Challenge 1

• No single standard specification
- W3C DOM recommendations, WHATWG HTML specifications (informal and incomplete)
- Non-standard browser features (`Screen` object, found in more than 70% of the 10,000 most popular websites)
- Inconsistent browser features (`attachEvent` in Internet Explorer and `addEventListener` in Safari and Chrome)

Our solution : modeling based on empirical data

### Challenge 2

• Interleaving HTML parsing and events with JavaScript execution




From http://www.whatwg.org/specs/web-apps/current-work/multipage/parsing.html#parsing/

Our assumption : all scripts are executed after completion of HTML parsing and all the events happen after execution of the top-level code

### Challenge 3

• Multiple JavaScript execution contexts with multiple documents in an application

```
<html> …
  <iframe src = "another.html"> …
  </iframe> …
</html>
```

Our assumption : only a single document is present in a web application

### Challenge 4

• Extensive use of JavaScript libraries such as jQuery, Prototype, and MooTools

Our solution : modeling the core jQuery library

## Practical DOM Modeling

### Method

• Tool : instrumented WebKit browser Engin and MiniBrowser
• Target : 9,465 world-wide most popular web sites by *alexa.com*
• Data : `usage of DOM-related fields and APIs`

### Result 1

• 10 most frequently used DOM properties among all properties defined in the W3C DOM Level 3 Core specification

| rank | interface | field | sites | interface | API | sites |
|---|---|---|---|---|---|---|
| 1 | Document | documentElement | 8,116 | Document | getElementById | 7,441 |
| 2 | NodeList | length | 8,104 | Document | createElement | 7,425 |
| 3 | Node | parentNode* | 8,008 | Node | appendChild* | 7,393 |
| 4 | Node | nodeType | 7,718 | Document | getElementsByTagName | 7,327 |
| 5 | Node | firstChild* | 7,669 | Node | insertBefore* | 7,182 |
| 6 | Node | ownerDocument | 7,497 | Element | getElementsByTagName | 7,134 |
| 7 | Node | childNodes* | 7,354 | Element | getAttribute | 7,060 |
| 8 | Node | nodeName | 7,297 | Node | removeChild* | 7,018 |
| 9 | Node | lastChild* | 6,835 | Element | setAttribute | 6,868 |
| 10 | Node | nextSibling* | 5,319 | Document | createComment | 6,427 |

Properties marked with * indicate that they are related to DOM tree search and manipulation
⇨ It is necessary to model a DOM tree precisely to give precise analysis results for many web applications

### Result 2

• Usage of DOM fields and APIs defined in DOM specifications (W3C DOM Level 3 Core, DOM Level 2 HTML, DOM Level 2 Events)

| spec. | fields def. | fields no use | APIs def. | APIs no use |
|---|---|---|---|---|
| 3 Core | 120 | 67 (55.8%) | 88 | 51 (58%) |
| 2 HTML | 288 | 126 (43.8%) | 36 | 23 (63.9%) |
| 2 Events | 57 | 25 (43.9%) | 14 | 4 (28.6%) |
| total | 465 | 218 (46.9%) | 138 | 78 (56.5%) |

About a half of the fields and APIs were never used
⇨ It is not necessary to model all the fields and APIs defined in standard specifications

### Result 3

• Usage of DOM fields and APIs defined in the main web pages of 9,465 websites

| type | > 1,000 sites | > 100 sites | > 10 sites | > 1 site |
|---|---|---|---|---|
| field | 150 | 327 | 951 | 1,721 |
| API | 50 | 98 | 160 | 267 |
| total | 200 | 425 | 1,111 | 1,988 |

We can minimize the modeling effort by setting priorities among DOM properties

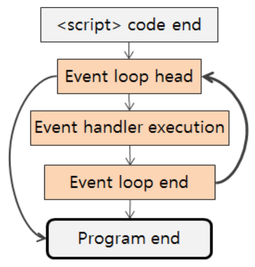
## Added Modules

### HTML Parser

• Combination of two parsers
- Jericho HTML parser 3.3 (script code extraction)
- CyberNeko HTML parser 1.9.17 (DOM tree construction)

### Event CFG Builder

• Build a ever CFG : adopted the TAJS event model




• Event type : load, unload, mouse, keyboard, ready, message, time, other
• No consideration of execution order among events

### Bug Detector

• Capable of detecting 83.2% of all 155 errors defined in the EMCAScript standard
• Provides various configuration options for bug messages
- Example : show all messages or only definite messages
• Bug type

| | Range | Refe-rence | Syntax | Type | URI | Total |
|---|---|---|---|---|---|---|
| ECMAScript | 7 | 7 | 31 | 98 | 12 | 155 |
| BugDetector | 7 | 7 | 31 | 72 | 12 | 129 |
| Coverage | 100% | 100% | 100% | 73.4% | 100% | 83.2% |

**RangeError**
```
var a = new Array(4.5);  // not UInt
```

**ReferenceError**
```
var x = 10 + y;  // y is not declared
```

**SyntaxError**
```
new Function("a, *@", "var x=1;");  // "*@" is not a JavaScript variable
```

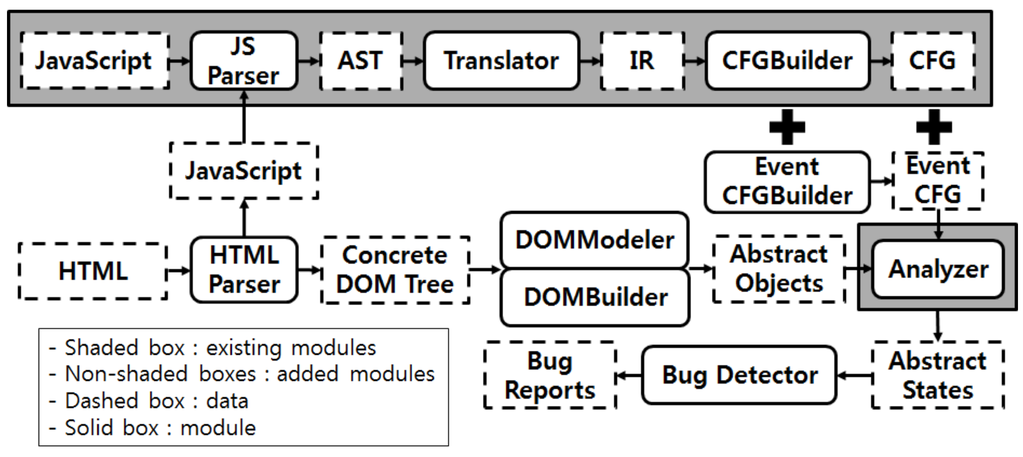
**TypeError**
```
var x = new "aa"; // non-constructor
```

**URIError**
```
decodeURI("%1"); // "%1" not in an URI format
```

## $SAFE_{Wapp}$ Framework

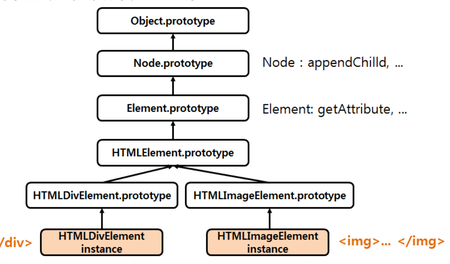
Implementation available at : `http://safe.kaist.ac.kr`

### Overall Structure




### DOMModeler and DOMBuilder

• Build an initial state for the Analyzer module
• DOMModeler : adds DOM prototype objects that have abstract browser APIs




• DOMBuilder : translates a concrete DOM tree to an abstract DOM

## Bug Detection in Websites

• Bug detection in the main web pages of 4 popular web sites

| website | LOC | Time (sec) | CallNon Function | Absent Read | Conditional Branch | Function ArgSize | Shadowing | Primitive ToObject | Total |
|---|---|---|---|---|---|---|---|---|---|
| wikipedia.org | 177 | 32.88 | 0 | 4 | 9 | 1 | 0 | 0 | 14(W) |
| odnoklassniki.ru | 226 | 22.94 | 0 | 1 | 12 | 3 | 1 | 0 | 17(W) |
| soso.com | 2413 | 27.52 | 0 | 9 | 20 | 18 | 2 | 1 | 50(W) |
| directrev.com | 4549 | 42.03 | 1 | 4 | 37 | 7 | 36 | 0 | 1(E)/84(W) |

### CallNonFunction Bug

• directrev.com

```
$('ul.accordion').accordion( ... );
```

The "accordion" function is defined in the jQuery UI library but the site did not import the library!!!

### AbsentRead Bug

• wikipedia.com

```
if (obj.addEventListener) { ... }
else if (obj.attachEvent) { ... }
```

Only the Internet Explorer supports the "attachEvent" method but most browsers do not have the property!!!

### FunctionArgSize Bug

• odnoklassniki.ru

Called the function that have 3 parameters with 2 arguments!!!

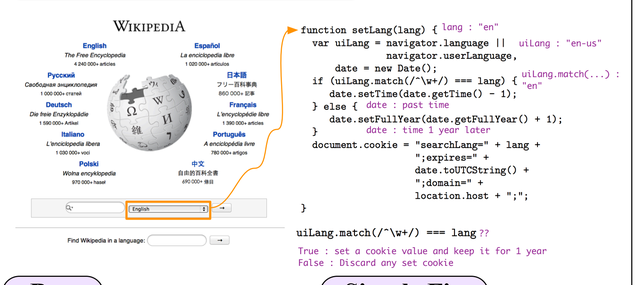
```
function createCookie(name, value, days) {
    if (days) { ... } ...
}
...
createCookie('t', 't');
```

### Shadowing Bug

• directrev.com Duplicate declaration with the same variable!!!

```
if ($cfs.data('cfs_isCarousel')) {
    var starting_position = ...
} else { var starting_position =  ... }
```

### Bug in Wikipedia




**Bug**
```
uiLang.match(/^\w+/) === lang
```
always evaluates to false!!!

**Simple Fix**
```
uiLang.match(/^\w+/) === lang
        ⇩
uiLang.match(/^\w+/) == lang
```