Problems in Software Development

Chenglong Wang
Jan.13.2015
Problem 1: Automatically API migration / API upgrade

• A new version of API may introduce some incompatible changes compared to its old version.
  • Google Calendar v2.0 ➔ v3.0
  • Angular JS v1.2 ➔ v1.3
  • Facebook Platform API v2.4 ➔ v2.5

• Users need to modify their original code to continue their development.

• Developers will provide some guide and examples to demonstrate the modification.
Angular JS migration guide

Due to c8700f04, animation-related callbacks are now fired on `$animate.on` instead of directly being on the element.

```javascript
// < 1.4
element.on('$animate:before', function(e, data) {
  if (data.event === 'enter') { ... }
});
element.off('$animate:before', fn);

// 1.4+
$animate.on('enter', element, function(data) {
  //...
});
$animate.off('enter', element, fn);

myElement.on('$animation:before', function(e, data) {
  if (data.event === 'enter') { dosomething();}
});
myElement.off('$animate:before', fn);

$animate.on('enter', myElement, function(data) {
  dosomething();
});
$animate.off('enter', myElement, fn);
```
Why challenging?
What are the possible solutions?

• Web APIs are commonly used in JavaScript, and most current research focus on Java/C.
  • ‘Infer’ their ‘types’ by learning their names.
• Current research focus more on rule-based transformation (to achieve safety).
  • Automation as a method to help programmers, rather than a substitution of them. (Not necessarily all correct.)
  • Infer the transformation rules through the examples.
• Existing approach to support auto-migration supports only one-to-many substitution. [1]
If enough time...
Problem 2: Data consistency issues in interactive Web applications

• Suppose we have a value *date*.
• *date* may be displayed in different position in the webpage, possibly:
  • Title, caption of a graph, text in a paragraph ...
• *date* can be modified in multiple ways:
  • A slide bar, input box, clock ...
• Making sure all occurrences of *date* are consistent can be challenging.
NFL players: height & weight over time

function updateYear(yr) {
  rects.data(bins[yr]);
  .style("fill", function(d) {
    return color(d.pct);
  });
  year.text(yr);
  $slider.val(yr);
}

$slider.on("slide click", function(e) {
  stop();
  currentYear = +$(this).val();
  updateYear(currentYear);
});
References