



Mocks, Proxies, and Transpilation as Development Strategies for Web Development

Dave Mason & Noury Bouraqadi

Ryerson University École Mines, Douai

Web Development

- Server side
 - Seaside
 - Aidaweb
 - Illiad
 - SST
- Client side
 - Javascript et al
 - SqueakJS
 - Amber, MySmalltalk, S8
 - PharoJS

Development á la PharoJS

1 Native Pharo Testing

- build models
- test/debug in full Pharo

2 App with Remote UI

- open a bridge to a JS engine (web browser)
- from Pharo, in Smalltalk
- access global variables like `document` `window`
- access fields/functions via message sends
- proxies make seamless interaction
- blocks as event handlers

3 Full Client App

- same code from previous step
- export a Javascript file from Smalltalk code

Demo

```
1 testBrowserButton
2   | button flag |
3   button := document createElement: 'button'.
4   button
5     id: 'who';
6     innerHTML: 'Click_on_this_within_10_seconds_or_the_test_fails';
7     addEventListener: 'click' block: [: ev |
8       button innerHTML: 'Click_again_immediately_or_the_test_fails';
9       addEventListener: 'click' block: [: xev |
10        flag := true].
11    ];
12    addEventListener: #mouseover block: [: ev : this |
13      button style backgroundColor: 'yellow'
14    ] close.
15  button style backgroundColor: 'pink';
16        height: '2cm'.
17  document body native_appendChild: button.
18  flag := nil.
19  self assert: (window confirm: 'If_you_can_see_the_pink_button,_accept')
20  JbBridge pollFor: 10 seconds orUntil: [flag].
21  self assert: flag notNil description: 'button_not_clicked'.
22  self assert: (window confirm: 'Did_the_button_turn_yellow?')
```

```
1 Received: document._createElement_("button")
2 Sent: {"proxy": "$_1"}
3 Received: $_1._id_("who")
4 Sent: {"basic": "who"}
5 Received: $_1._innerHTML_("Click on this within 10 seconds or the test fails")
6 Sent: {"basic": "Click on this within 10 seconds or the test fails"}
7 Received: JbLoggingEvaluatorWebSocketDelegate._default()._makeBlockClosureProxy_(false)
8 Sent: {"proxy": "$_2"}
9 Received: JbLoggingEvaluatorWebSocketDelegate._set_callback_to_($_1, "click", $_2);
10 Sent: {"basic": true}
11 Received: $_1._style()
12 Sent: {"proxy": "$_3"}
13 Received: $_3._backgroundColor_("pink")
14 Sent: {"basic": "pink"}
15 Received: $_3._height_("2cm")
16 Sent: {"basic": "2cm"}
17 Received: document._body()
18 Sent: {"proxy": "$_4"}
19 Received: $_4.appendChild($_1)
20 Sent: {"proxy": "$_1"}
21 Received: window._confirm_("If you can see the pink button, accept this and then click on the but
22 Sent: {"basic": true}
23 Received: :-$_4
24 Received: :-$_3
25 Sent: {"cb": [{"$_2"}, {"proxy": "$_5"}, {"proxy": "undefined"}]}
26 Received: $_1._innerHTML_("Click again immediately or the test fails")
27 Sent: {"basic": "Click again immediately or the test fails"}
28 Received: JbLoggingEvaluatorWebSocketDelegate._default()._makeBlockClosureProxy_(false)
29 Sent: {"proxy": "$_6"}
30 Received: JbLoggingEvaluatorWebSocketDelegate._set_callback_to_($_1, "click", $_6);
31 Sent: {"remove": true, "proxy": "$_2"}
32 Received: :-$_2
33 Sent: {"basic": true}
```

Translation Challenges

- DoesNotUnderstand
- `undefined/null` vs. `nil`
- non-local returns
- unfortunate JS/DOM inheritance
- optimizing numeric methods
- optimizing block methods
- minimizing hand-written JS code (39/304 methods)

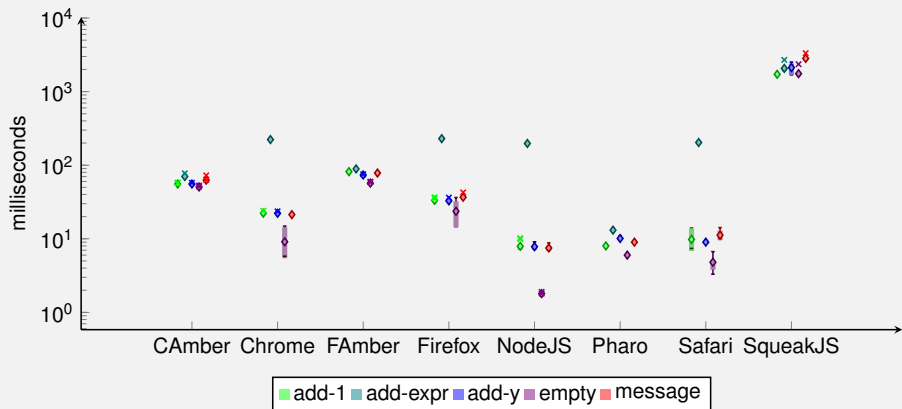
Translation Incompatibilities

Until/Unless implemented in ECMAScript ??

- Number stack
- Characters as String
- `become` :
- weak arrays
- resumable exceptions

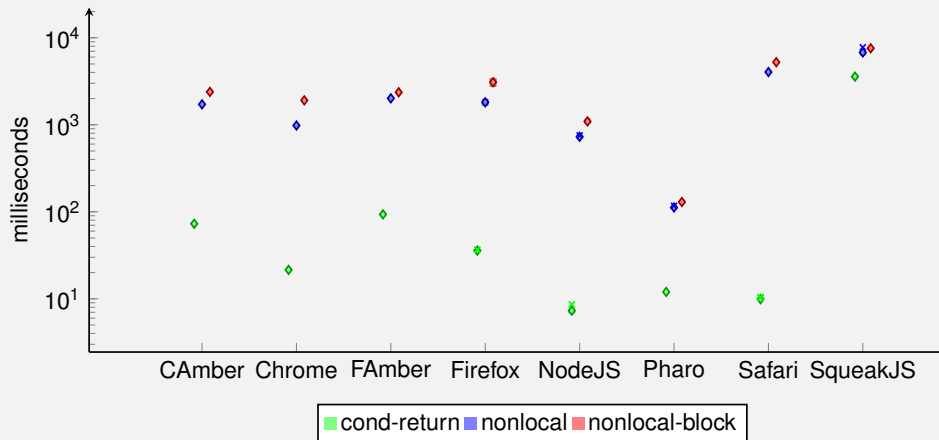
Performance

Simple Expressions



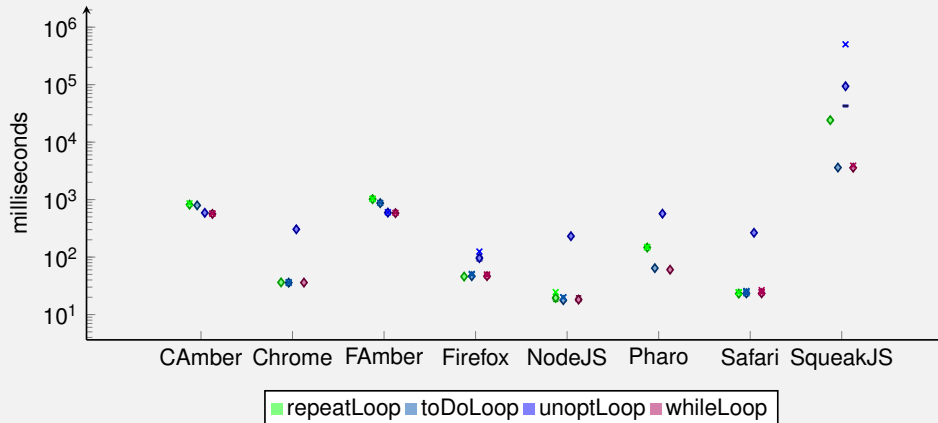
empty	
add-1	<code>x := x + 1</code>
add-y	<code>x := x + y</code>
add-expr	<code>x := x + instance one</code>
message	<code>instance simpleMethod</code>

Return expressions



cond-return	<code>self ten>5 ifTrue: [^#yes]</code>
nonlocal	<code>self ten>5 ifTrue: [aBlock value]</code>
nonlocal-block	<code>self ten>5 ifTrue: aBlock</code>

Loop expressions



repeatLoop	10 timesRepeat: [x:=x+1]
toDoLoop	1 to: 10 do: [:y x:=x+y]
whileLoop	[y<10] whileTrue: [y:=y+1. x:=x+y]
unoptLoop	b1 whileTrue: b2

Conclusions

- if you want perfect semantics, use SqueakJS
- if you want web-browser based development, consider Amber
- if you want high performance and powerful IDE based development, use PharoJS

Questions?



@pharojs

<http://pharojs.org>