

CodeHint: Dynamic and Interactive Synthesis of Code Snippets

Joel Galenson, Philip Reames, Rastislav Bodik,
Björn Hartmann, Koushik Sen
ICSE 2014

Summary by Prof. Thomas LaToza

SWE 795, Spring 2017

Software Engineering Environments

CodeHint Overview

```
1 final JComponent tree = makeTree();
2 tree.addMouseListener(new MouseAdapter() {
3     public void mousePressed(MouseEvent e) {
4         int x = e.getX(), y = e.getY();
5         Object o = null;
6         // Get the menu bar or the clicked element.
7     }
8 });
```

o' instanceof JMenuItemBar

pdspec

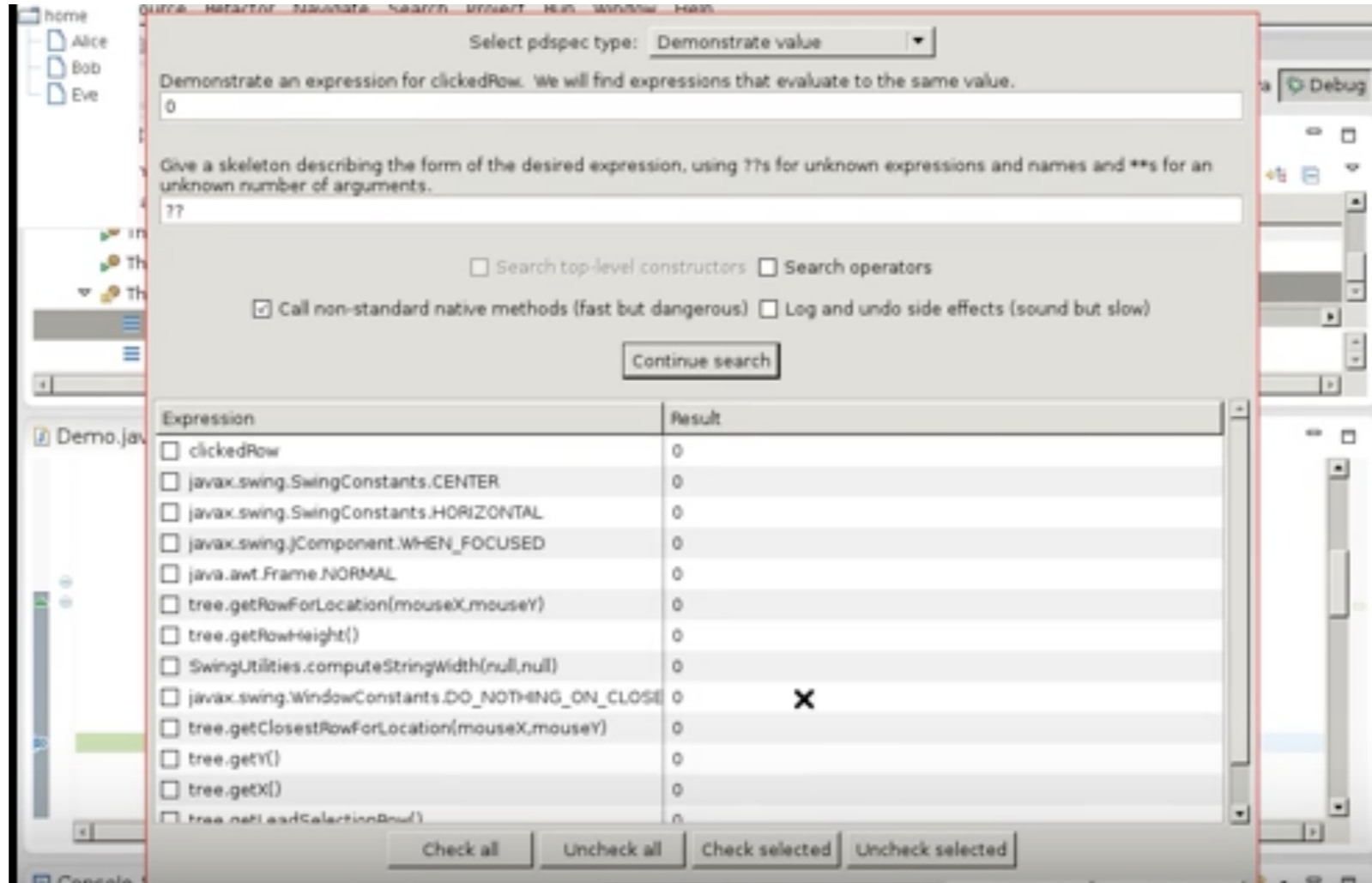


CodeHint

```
((JFrame)SwingUtilities.getWindowAncestor(jtree))
    .getJMenuBar()
((JFrame)tree.getTopLevelAncestor()).getJMenuBar()
((JFrame)SwingUtilities.getRoot(tree)).getJMenuBar()
```

- Uses assertion from developer to guide search for expressions that satisfy assertion

Demo



<https://www.youtube.com/watch?v=qn5yIEe9kks>

<http://jgalenson.github.io/codehint/>

Evaluation: Performance

	Normal algorithm						Side effects Depth 2 Time	No pruning Depth 3 #	Brute force Depth 3 #
	Depth 2		Depth 3		Depth 4				
	#	Time	#	Time	#	Time			
P 1	34	0.4	611	1.1	19259	9.1	0.4	54911	2034829
P 2	57	0.5	912	1.9	35232	7.7	0.8	10953	847418
P 3	124	0.4	1156	1.2	124991	17.8	1.0	157052	6200476
P 4	7	0.2	36	0.3	552	0.8	0.3	36	219
P 5	22	0.3	234	0.4	2565	1.1	0.3	4692	155774
S 1	8	0.9	223	2.0	1401	3.6	4.3	3775	39439
S 2	12	0.6	275	1.4	2043	3.6	4.5	4457	51079
S 3	70	0.7	814	2.8	6645	6.2	4.9	41359	1867350
S 4	103	1.1	842	3.7	22138	10.6	8.5	504018	61246626
S 5	32	0.8	595	2.8	179956	15.8	6.7	24409	272025
R 1	22	0.2	98	0.3	846	0.6	0.6	98	33112
R 2	12	0.2	137	0.3	1090	0.5	0.3	137	1004
R 3	8	0.1	13	0.1	51	0.2	0.5	13	19
R 4	7	0.2	19	0.2	80	0.3	0.3	19	33
R 5	24	0.3	229	0.3	1761	1.9	0.4	609	115410
Avg	36.1	0.5	412.9	1.3	26574	5.3	2.3	53769.2	4857654.2
Med	22	0.4	234	1.1	2043	3.6	0.6	4457	115410

User Study

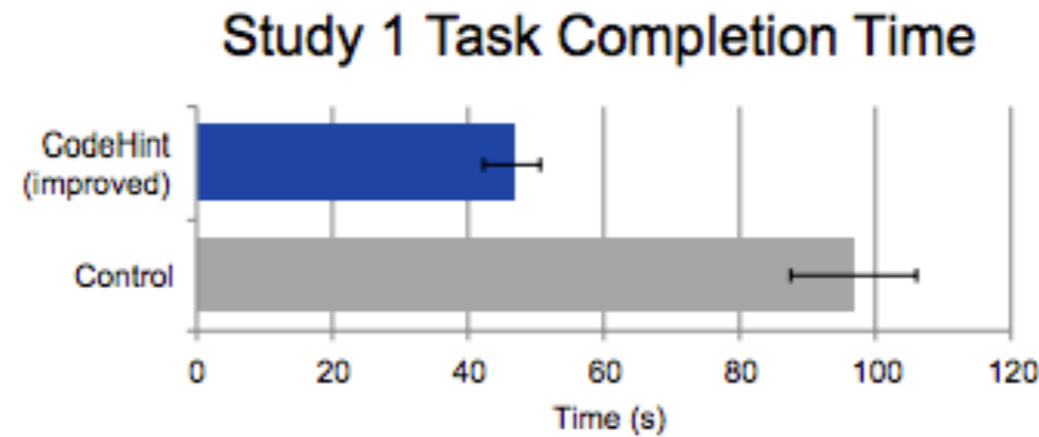


Figure 2: The task completion time of subjects in our first user study. The error bars show the standard error.

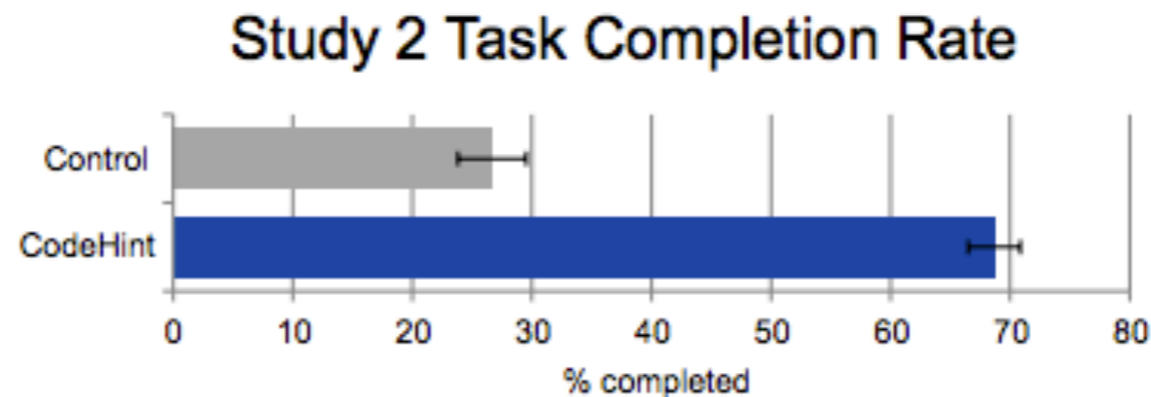


Figure 3: The task completion rate of subjects in our second user study. The error bars show the standard error.

Questions for discussion

- Overall reaction to the paper
- Would you use such a system for your everyday programming?
 - Why or why not?
- In what other programming tasks might this help developers?