Direct Manipulation Programming Systems

Ravi Chugh, Brian Hempel, Justin Lubin
Jacob Albers, Grace Lu, Mitch Spradlin
The original Chicago Ferris Wheel, built for the 1893 World's Columbian Exposition [Wikipedia]
Idea
Idea
Prototype... Repair... Refactor...
<table>
<thead>
<tr>
<th>Programming Language</th>
<th>Direct Manipulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>2D Graphics</td>
<td>![Ai]</td>
</tr>
<tr>
<td>Presentations</td>
<td>![PowerPoint]</td>
</tr>
<tr>
<td>Documents</td>
<td>![Word]</td>
</tr>
<tr>
<td>Spreadsheets</td>
<td>![Excel]</td>
</tr>
<tr>
<td>Web Apps</td>
<td>![Google Web Designer]</td>
</tr>
<tr>
<td>General-Purpose Languages</td>
<td>Domain-Specific Languages or Libraries</td>
</tr>
<tr>
<td>---------------------------</td>
<td>----------------------------------------</td>
</tr>
<tr>
<td>2D Graphics</td>
<td>JS</td>
</tr>
<tr>
<td>Presentations</td>
<td>JS</td>
</tr>
<tr>
<td>Documents</td>
<td>JS</td>
</tr>
<tr>
<td>Spreadsheets</td>
<td>JS</td>
</tr>
<tr>
<td>Web Apps</td>
<td>JS</td>
</tr>
</tbody>
</table>
Programming with:

Less Keyboard.

More Mouse.
\(\lambda\)-Calculus + Direct Manipulation

Reuse effort across domains

Enable experts and library writers to extend built-ins

Smooth spectrum between "Experts" and "End Users"
\(\lambda\)-Calculus + Direct Manipulation
$\lambda$-Calculus + Direct Manipulation

0 : Text-Edit Code
$\lambda$-Calculus + Direct Manipulation of Output

0 : Text-Edit Code

1 : Mouse-Edit Output
$\lambda$-Calculus + Direct Manipulation of Code

0 : Text-Edit Code

1 : Mouse-Edit Output

2 : Mouse-Edit Code
1 : Mouse-Edit Output

2 : Mouse-Edit Code

Sketch-n-Sketch

Current file: Untitled (Ferris Wheel)

(def wheel (cx cy rCenter wCap rCap numSpokes spokeLen rotAngle)
  (let rim [(ring 'darkgray' 6 cx cy spokeLen)]
  (let center [[circle 'black' cx cy rCenter]]
  (let frame [(nStar 'goldenrod' 'darkgray' 3 numSpokes spokeLen]
  (let spokePts (nPointsOnCircle numSpokes rotAngle cx cy spokeLen)
  (let cars (map (i x y) (squareByCenter (if (= i 0) 'pink'
  (let hubcaps (map (x y) (circle 'black' x y rCap)) spokePts)
  (concat [rim cars center frame hubcaps])))

(def [cx cy spokeLen rCenter wCap rCap] [228 241 178 20 30]
(def [spokes angle] [14 [3-25] 0.062 [-3.14 -3.14]])

(svg [Cap spokes spokeLen angle])

Code Tools
Introduce Variables
Make Equal with Single Variable

New variable: num
New variable: angle
Sketch-n-Sketch Demo

```python
(def target ~(red white r anchor[xAnchor yAnchor]~)
  (def [circle1_r circle2_r circle3_r circle4_r]
    (map ~(i * ! r~) (reverse (range 1 4!))))
  (def circle ~(\color r~)
    [~(rowCircle color 360 0 xAnchor yAnchor r)~])
  (def circle1 ~(circle white circle1_r))
  (def circle2 ~(circle red circle2_r))
  (def circle3 ~(circle white circle3_r))
  (def circle4 ~(circle red circle4_r))
  ~(anchoredGroup (concat [ circle1 circle2 circle3 circle4 ]))~)
(blobs ~(withAnchor [170 160] (target 499 14 25.25))
  (withAnchor [101 409] (target 499 238 15.416666666666666))
  (withAnchor [319 353] :Point (target 499 72 25.25)))
```
Sketch-n-Sketch

1 : Mouse-Edit Output

2 : Mouse-Edit Code
Live Synchronization
[PLDI 2016]

Draw, Relate, Group, and Abstract
[UIST 2016] [wip]
Semi-Automated Programming by Manipulating Output
Draw
Relate
Tweak
Group
Draw

Sketch-n-Sketch

Limited, Syntactic Program Updates

Relate

Sketch-n-Sketch ++

Expressive, Trace-Based Program Updates

Group

Tweak
Additional Interaction to Resolve User Intent
Sketch-n-Sketch

1 : Mouse-Edit Output

2 : Mouse-Edit Code
Sketch-n-Sketch

Lightweight Structured Editing
[wip]

2 : Mouse-Edit Code
Plain Text Editing

Structured Editing

Automated Refactoring

Code Editors
Lightweight Structured Editing via Direct Manipulation of (Plain Text) Code
Lightweight Structured Editing via Direct Manipulation

Structural (Multi-)Selection; Context-Sensitive Tool Menu; Hover Preview and Confirm

Traditional UI
Text Selection; Tool Menu; Configuration Wizard

Automated Refactoring
1. Learning Curve for New UI + Tools
2. Faster Tool Invocations using New UI
3. Overwhelming Preference for New UI

Structural (Multi-)Selection; Context-Sensitive Tool Menu; Hover Preview and Confirm

Traditional UI:
Text Selection; Tool Menu; Configuration Wizard

21 users
Sketch-n-Sketch

1: Mouse-Edit Output

2: Mouse-Edit Code
Prototype; Repair; Refactor; Repeat
<table>
<thead>
<tr>
<th>PL</th>
<th>DM</th>
</tr>
</thead>
<tbody>
<tr>
<td>2D Graphics</td>
<td>Sketch-n-Sketch</td>
</tr>
<tr>
<td>Presentations</td>
<td>SnS Slides</td>
</tr>
<tr>
<td>Documents</td>
<td>SnS Docs</td>
</tr>
<tr>
<td>Spreadsheets</td>
<td>SnS Sheets</td>
</tr>
<tr>
<td>Web Apps</td>
<td>SnS Web</td>
</tr>
</tbody>
</table>

General Purpose PL + General Program Synthesis

Domain Specific UI + Domain Specific Program Synthesis

Prototype; Repair; Refactor; Repeat
Prototype; Repair; Refactor; Repeat

<table>
<thead>
<tr>
<th>PL</th>
<th>DM</th>
</tr>
</thead>
<tbody>
<tr>
<td>2D Graphics</td>
<td>Domain Specific UI</td>
</tr>
<tr>
<td>Presentations</td>
<td>Domain Specific Program Synthesis</td>
</tr>
<tr>
<td>Documents</td>
<td></td>
</tr>
<tr>
<td>Spreadsheets</td>
<td></td>
</tr>
<tr>
<td>Web Apps</td>
<td></td>
</tr>
<tr>
<td>Non Visual</td>
<td>DM Code / Traces / Output</td>
</tr>
</tbody>
</table>

General Purpose PL

Domain Specific UI
Programming with: 

Less Keyboard. 

More Mouse.
Related Work

DM + Code
Dynamic Drawing Apparatus

Code + DM
Wang et al. [FSE 2012]
McDirmid's Demos
Sketch-n-Sketch

Constraints
Sketchpad
ThingLab
Juno-2

Program Synthesis
(Sketching, PBE, Repair)
Sketch-n-Sketch
http://ravichugh.github.io/sketch-n-sketch/

DEMO  VIDEOS
CODE  PAPERS

30,000+ LOC Elm

PLDI 2016  UIST 2016
λ-Calculus + Direct Manipulation of Code and Output