Combining Programmatic and Direct Manipulation

Ravi Chugh, Brian Hempel, Mitchell Spradlin, Jacob Albers

What if you could change a program just by changing its output?

1. Programmatic vs. Direct Manipulation

<table>
<thead>
<tr>
<th></th>
<th>Programmatic Manipulation</th>
<th>Direct Manipulation</th>
<th>VISION: &quot;Prodirect&quot; Manipulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Immediate Feedback</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Rapid Prototyping</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Abstraction Capabilities</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
</tr>
</tbody>
</table>

Sketch-n-Sketch supports live synchronization (Goal A) and has growing support for directly drawing shapes (Goal B) and interactively relating shapes (Goal C).

2. Challenges and Goals

**GOAL A: Live Synchronization**
Infer "Small" Program Update During Direct Manipulation

**GOAL B: Infer Program**
Output

**GOAL C: Ad Hoc Synchronization**
Infer "Large" Program Update After Direct Manipulation

3. Applications

Many kinds of programming might benefit from prodirect manipulation. Good first targets are graphical applications such as data visualization, animation, HTML/CSS, and slide show presentation. Our initial tool explores prodirect creation of vector graphics.

4. Prodirect Manipulation for SVG

Users write programs to generate SVG vector graphics and then directly manipulate the graphics output on the right to change the program on the left.

5. The Future

- Novel User Interface Capabilities
- Intelligent, Interactive Program Synthesis
- Domains Beyond Graphics

Sketch-n-Sketch supports live synchronization (Goal A) and has growing support for directly drawing shapes (Goal B) and interactively relating shapes (Goal C).

Live Synchronization

User drags object to new position...

When there is ambiguity about what parts of the program to change (red boxes above), Sketch-n-Sketch uses heuristics to choose which locations to modify.