### TABLE 4-13. Pen/Voice Editor State Diagram Statistics Table 1

<table>
<thead>
<tr>
<th>Subject Number (Task #)</th>
<th>Speech Rec Character Errors</th>
<th>Speech Rec Character Accuracy Rate</th>
<th>Speech Rec Problem Characters (2+ Errors)</th>
<th>Final Wrong Chars</th>
<th>Questions Asked or Comments</th>
<th>Diagram Imperfect</th>
<th>Diagram Drawing Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (Task 2)</td>
<td>3</td>
<td>93%</td>
<td>S</td>
<td>0</td>
<td>6</td>
<td>1</td>
<td>15:14</td>
</tr>
<tr>
<td>2 (Task 1)</td>
<td>0</td>
<td>100%</td>
<td></td>
<td>0</td>
<td>6</td>
<td>2</td>
<td>14:43</td>
</tr>
<tr>
<td>3 (Task 4)</td>
<td>7</td>
<td>85%</td>
<td>S</td>
<td>0</td>
<td>5</td>
<td>3</td>
<td>17:42</td>
</tr>
<tr>
<td>4 (Task 3)</td>
<td>6</td>
<td>87%</td>
<td>S, BKSP</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>12:13</td>
</tr>
<tr>
<td>9 (Task 4)</td>
<td>3</td>
<td>93%</td>
<td></td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>16:04</td>
</tr>
<tr>
<td>10 (Task 3)</td>
<td>2</td>
<td>95%</td>
<td></td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>15:53</td>
</tr>
<tr>
<td>13 (Task 4)</td>
<td>0</td>
<td>100%</td>
<td></td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>11:45</td>
</tr>
<tr>
<td><strong>AVERAGE</strong></td>
<td><strong>3</strong></td>
<td><strong>93%</strong></td>
<td></td>
<td><strong>0</strong></td>
<td><strong>4</strong></td>
<td><strong>3</strong></td>
<td><strong>14:48</strong></td>
</tr>
</tbody>
</table>

### TABLE 4-14. Pen/Voice Editor State Diagram Statistics Table 2

<table>
<thead>
<tr>
<th>Subject Number (Task #)</th>
<th>Speech Command Used</th>
<th>Speech Command Errors</th>
<th>Speech Command Accuracy Rate</th>
<th>Speech Rec Problem Commands (2+ Errors)</th>
<th>Time to Draw Diagram</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (Task 2)</td>
<td>42</td>
<td>2</td>
<td>95%</td>
<td></td>
<td>15:14</td>
</tr>
<tr>
<td>2 (Task 1)</td>
<td>60</td>
<td>2</td>
<td>97%</td>
<td></td>
<td>14:43</td>
</tr>
<tr>
<td>3 (Task 4)</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td></td>
<td>17:42</td>
</tr>
<tr>
<td>4 (Task 3)</td>
<td>48</td>
<td>3</td>
<td>94%</td>
<td>Delete</td>
<td>12:13</td>
</tr>
<tr>
<td>9 (Task 4)</td>
<td>62</td>
<td>3</td>
<td>95%</td>
<td>OK</td>
<td>16:04</td>
</tr>
<tr>
<td>10 (Task 3)</td>
<td>34</td>
<td>4</td>
<td>88%</td>
<td>OK</td>
<td>15:53</td>
</tr>
<tr>
<td>13 (Task 4)</td>
<td>76</td>
<td>7</td>
<td>91%</td>
<td>OK, Select</td>
<td>11:45</td>
</tr>
<tr>
<td><strong>AVERAGE</strong></td>
<td><strong>54</strong></td>
<td><strong>4</strong></td>
<td><strong>93%</strong></td>
<td></td>
<td><strong>14:48</strong></td>
</tr>
</tbody>
</table>
Flowchart Diagram

TABLE 4-15. Pen-only Editor Flowchart Diagram Statistics Table

<table>
<thead>
<tr>
<th>Subject Number (Task #)</th>
<th>Char Rec Errors</th>
<th>Char Rec Accuracy Rate</th>
<th>Char Rec Problem Characters (2+ Errors)</th>
<th>Questions Asked or Comments</th>
<th>Diagram Imperfect</th>
<th>Diagram Drawing Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (Task 3)</td>
<td>7</td>
<td>92%</td>
<td>?, E, T</td>
<td>1</td>
<td>2</td>
<td>10:27</td>
</tr>
<tr>
<td>2 (Task 4)</td>
<td>4</td>
<td>95%</td>
<td>N, P</td>
<td>2</td>
<td>0</td>
<td>10:02</td>
</tr>
<tr>
<td>7 (Task 1)</td>
<td>7</td>
<td>92%</td>
<td>G</td>
<td>2</td>
<td>2</td>
<td>15:16</td>
</tr>
<tr>
<td>8 (Task 2)</td>
<td>12</td>
<td>88%</td>
<td>?, L, T</td>
<td>4</td>
<td>6</td>
<td>13:32</td>
</tr>
<tr>
<td>9 (Task 1)</td>
<td>12</td>
<td>88%</td>
<td>G, T</td>
<td>7</td>
<td>4</td>
<td>22:17</td>
</tr>
<tr>
<td>10 (Task 2)</td>
<td>8</td>
<td>91%</td>
<td>?, N</td>
<td>1</td>
<td>0</td>
<td>12:08</td>
</tr>
<tr>
<td>13 (Task 1)</td>
<td>4</td>
<td>95%</td>
<td>N/A</td>
<td>3</td>
<td>N/A</td>
<td>10:14</td>
</tr>
<tr>
<td>AVERAGE</td>
<td>8</td>
<td>92%</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>13:25</td>
</tr>
</tbody>
</table>

TABLE 4-16. Pen/Voice Editor Flowchart Diagram Statistics Table

<table>
<thead>
<tr>
<th>Subject Number (Task #)</th>
<th>Speech Rec Character Errors</th>
<th>Speech Rec Character Accuracy Rate</th>
<th>Speech Rec Problem Characters (2+ Errors)</th>
<th>Final Wrong Chars</th>
<th>Questions Asked or Comments</th>
<th>Diagram Imperfect</th>
<th>Diagram Drawing Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 (Task 2)</td>
<td>30</td>
<td>77%</td>
<td>A, B, G, D, P, SPACE</td>
<td>6</td>
<td>5</td>
<td>5</td>
<td>24:00</td>
</tr>
<tr>
<td>4 (Task 1)</td>
<td>12</td>
<td>88%</td>
<td>D, T</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>15:06</td>
</tr>
<tr>
<td>5 (Task 4)</td>
<td>17</td>
<td>85%</td>
<td>P, T</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>8:59</td>
</tr>
<tr>
<td>6 (Task 3)</td>
<td>37</td>
<td>74%</td>
<td>A, D, O, P, T</td>
<td>12</td>
<td>0</td>
<td>1</td>
<td>11:06</td>
</tr>
<tr>
<td>11 (Task 4)</td>
<td>22</td>
<td>81%</td>
<td>A, T</td>
<td>5</td>
<td>2</td>
<td>0</td>
<td>9:43</td>
</tr>
<tr>
<td>12 (Task 3)</td>
<td>30</td>
<td>76%</td>
<td>A, T</td>
<td>12</td>
<td>6</td>
<td>2</td>
<td>16:34</td>
</tr>
<tr>
<td>AVERAGE</td>
<td>25</td>
<td>80%</td>
<td>6</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>14:15</td>
</tr>
</tbody>
</table>
### TABLE 4-17. Pen/Voice Editor Flowchart Diagram Statistics Table 2

<table>
<thead>
<tr>
<th>Subject Number (Task #)</th>
<th>Speech Command Used</th>
<th>Speech Command Errors</th>
<th>Speech Command Accuracy Rate</th>
<th>Speech Rec Problem Commands (2+ Errors)</th>
<th>Time to Draw Diagram</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 (Task 2)</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td></td>
<td>24:00</td>
</tr>
<tr>
<td>4 (Task 1)</td>
<td>21</td>
<td>7</td>
<td>67%</td>
<td>Delete</td>
<td>15:06</td>
</tr>
<tr>
<td>5 (Task 4)</td>
<td>16</td>
<td>0</td>
<td>100%</td>
<td></td>
<td>8:59</td>
</tr>
<tr>
<td>6 (Task 3)</td>
<td>28</td>
<td>7</td>
<td>75%</td>
<td>Cut, Delete</td>
<td>11:06</td>
</tr>
<tr>
<td>11 (Task 4)</td>
<td>14</td>
<td>1</td>
<td>93%</td>
<td></td>
<td>9:43</td>
</tr>
<tr>
<td>12 (Task 3)</td>
<td>50</td>
<td>2</td>
<td>96%</td>
<td></td>
<td>16:34</td>
</tr>
<tr>
<td>AVERAGE</td>
<td>26</td>
<td>3</td>
<td>87%</td>
<td></td>
<td>14:15</td>
</tr>
</tbody>
</table>

**Dataflow Diagram**

### TABLE 4-18. Pen-only Editor Dataflow Diagram Statistics Table

<table>
<thead>
<tr>
<th>Subject Number (Task #)</th>
<th>Char Rec Errors</th>
<th>Char Rec Accuracy Rate</th>
<th>Char Rec Problem Characters (2+ Errors)</th>
<th>Questions Asked or Comments</th>
<th>Diagram Imperfect</th>
<th>Diagram Drawing Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 (Task 1)</td>
<td>4</td>
<td>92%</td>
<td></td>
<td>2</td>
<td>1</td>
<td>8:45</td>
</tr>
<tr>
<td>4 (Task 2)</td>
<td>2</td>
<td>95%</td>
<td></td>
<td>2</td>
<td>1</td>
<td>8:17</td>
</tr>
<tr>
<td>5 (Task 3)</td>
<td>5</td>
<td>90%</td>
<td>H</td>
<td>1</td>
<td>0</td>
<td>7:20</td>
</tr>
<tr>
<td>6 (Task 4)</td>
<td>3</td>
<td>93%</td>
<td></td>
<td>0</td>
<td>0</td>
<td>4:39</td>
</tr>
<tr>
<td>9 (Task 3)</td>
<td>13</td>
<td>83%</td>
<td>3, A, I, U</td>
<td>4</td>
<td>0</td>
<td>9:19</td>
</tr>
<tr>
<td>10 (Task 4)</td>
<td>5</td>
<td>90%</td>
<td>V</td>
<td>1</td>
<td>0</td>
<td>5:01</td>
</tr>
<tr>
<td>13 (Task 3)</td>
<td>2</td>
<td>95%</td>
<td></td>
<td>1</td>
<td>0</td>
<td>3:37</td>
</tr>
<tr>
<td>AVERAGE</td>
<td>5</td>
<td>90%</td>
<td></td>
<td>2</td>
<td>0</td>
<td>6:43</td>
</tr>
</tbody>
</table>
### TABLE 4-19. Pen/Voice Editor Dataflow Diagram Statistics Table 1

<table>
<thead>
<tr>
<th>Subject Number (Task #)</th>
<th>Speech Rec Character Errors</th>
<th>Speech Rec Character Accuracy Rate</th>
<th>Speech Rec Problem Characters (2+ Errors)</th>
<th>Final Wrong Chars</th>
<th>Questions Asked or Comments</th>
<th>Diagram Imperfect</th>
<th>Diagram Drawing Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (Task 4)</td>
<td>13</td>
<td>80%</td>
<td>3, 5, E, R</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>6:34</td>
</tr>
<tr>
<td>2 (Task 3)</td>
<td>2</td>
<td>95%</td>
<td></td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>6:16</td>
</tr>
<tr>
<td>7 (Task 2)</td>
<td>2</td>
<td>95%</td>
<td></td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>4:39</td>
</tr>
<tr>
<td>8 (Task 1)</td>
<td>23</td>
<td>72%</td>
<td>A, D, E, BKSP</td>
<td>4</td>
<td>7</td>
<td>3</td>
<td>5:58</td>
</tr>
<tr>
<td>11 (Task 2)</td>
<td>10</td>
<td>83%</td>
<td>T, V</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>5:50</td>
</tr>
<tr>
<td>12 (Task 1)</td>
<td>9</td>
<td>84%</td>
<td>T</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>9:33</td>
</tr>
<tr>
<td>AVERAGE</td>
<td>10</td>
<td>83%</td>
<td></td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>6:28</td>
</tr>
</tbody>
</table>

### TABLE 4-20. Pen/Voice Editor Dataflow Diagram Statistics Table 2

<table>
<thead>
<tr>
<th>Subject Number (Task #)</th>
<th>Speech Command Used</th>
<th>Speech Command Errors</th>
<th>Speech Command Accuracy Rate</th>
<th>Speech Rec Problem Commands (2+ Errors)</th>
<th>Time to Draw Diagram</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (Task 4)</td>
<td>20</td>
<td>4</td>
<td>80%</td>
<td>Shrink</td>
<td>6:34</td>
</tr>
<tr>
<td>2 (Task 3)</td>
<td>24</td>
<td>4</td>
<td>83%</td>
<td>Grid</td>
<td>6:16</td>
</tr>
<tr>
<td>7 (Task 2)</td>
<td>8</td>
<td>0</td>
<td>100%</td>
<td></td>
<td>4:39</td>
</tr>
<tr>
<td>8 (Task 1)</td>
<td>9</td>
<td>0</td>
<td>100%</td>
<td></td>
<td>5:58</td>
</tr>
<tr>
<td>11 (Task 2)</td>
<td>20</td>
<td>0</td>
<td>100%</td>
<td></td>
<td>5:50</td>
</tr>
<tr>
<td>12 (Task 1)</td>
<td>39</td>
<td>1</td>
<td>97%</td>
<td></td>
<td>9:33</td>
</tr>
<tr>
<td>AVERAGE</td>
<td>20</td>
<td>2</td>
<td>93%</td>
<td></td>
<td>6:28</td>
</tr>
</tbody>
</table>
Overall Diagram Statistics

Statistics in Tables 4-21 through 4-23 were calculated from the statistics in Tables 4-9 through 4-20.

TABLE 4-21. Overall Diagram Average Time Statistics

<table>
<thead>
<tr>
<th>Task</th>
<th>Pen-Only</th>
<th>Pen/Voice</th>
<th>Time Reduction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Petri Net Diagram</td>
<td>17:08</td>
<td>16:31</td>
<td>4.6% (PEN/VOICE)</td>
</tr>
<tr>
<td>State Diagram</td>
<td>17:50</td>
<td>14:48</td>
<td>17.0% (PEN/VOICE)</td>
</tr>
<tr>
<td>Flowchart Diagram</td>
<td>13:25</td>
<td>14:15</td>
<td>5.8% (PEN-ONLY)</td>
</tr>
<tr>
<td>Dataflow Diagram</td>
<td>6:43</td>
<td>6:28</td>
<td>3.7% (PEN/VOICE)</td>
</tr>
<tr>
<td>Overall</td>
<td>55:06</td>
<td>52:02</td>
<td>5.6% (PEN/VOICE)</td>
</tr>
</tbody>
</table>

TABLE 4-22. Overall Recognizer Accuracy Rates

<table>
<thead>
<tr>
<th>Recognizer</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Graffiti Handwriting Character</td>
<td>91%</td>
</tr>
<tr>
<td>Speech Character</td>
<td>83%</td>
</tr>
<tr>
<td>Speech Command</td>
<td>90%</td>
</tr>
</tbody>
</table>

TABLE 4-23. Overall Diagram Miscellaneous Statistics

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average Questions Asked per Diagram for the Pen-only Editor</td>
<td>2.5</td>
</tr>
<tr>
<td>Average Questions Asked per Diagram for the Pen/Voice Editor</td>
<td>3.0</td>
</tr>
<tr>
<td>Average Diagram Imperfections for Pen-only Editor</td>
<td>1.6</td>
</tr>
<tr>
<td>Average Diagram Imperfections for the Pen/Voice Editor</td>
<td>2.2</td>
</tr>
</tbody>
</table>
4.7 Results

Figure 4-10 shows that the typical subject used in the experiment has some experience using a graphical editor and a high amount of GUI interface experience. Most the subjects who have had pen computer, shape recognition, and handwriting recognition experience got it from using one of Washington University’s Pen Lab’s graphical editors. These subjects consist of about a third of the experiment subject population. Also, only 2 of the subjects had any previous speech recognition experience. Thus, the statistics indicate that there is a consistent mix of subjects ranging from an average to large amount of computer user interface experience. The interface modalities used in this experiment were new for most of the subjects.

Figure 4-11 shows that 85% of the subjects preferred the pen/voice editor over the pen-only editor. These subjects thought the pen/voice editor was faster and more usable. Only about a half of the subjects thought the pen/voice editor was the more user-friendly editor. Part of the reason behind this might have been the lack of previous experience the subjects have had using speech recognition. A related statistic is that 31% of the subjects preferred the editor that they did not find more user-friendly. It was expected that the preferred editor statistics would match up closer to the user-friendliness statistics rather than the speed and usability statistics. The best possible explanation for this phenomena is that computer users have become accustomed to using interfaces that are not made for the user’s benefit, but for the quickest and easiest way to control a particular application.

Table 4-21 gives statistics that show the pen/voice editor does speed up the pen editor slightly. Three out of the four diagrams were drawn faster on average with the pen/voice editor. The pen/voice editor reduced the overall average task time by 5.6%. The state diagram’s 17.0% average task time reduction shows that the pen/voice editor can significantly reduce the drawing time over the pen-only editor.

The one diagram that was faster to draw with the pen on average was the flowchart diagram. This diagram was chosen to test the text entry methods for the two editors. Table 4-22 shows that the overall character entry rates for the pen/voice editor was a significant 8% less than the pen-only editor. The pen-only editor had it’s best character entry accuracy and the pen/voice editor had it’s worst character entry accuracy in the
flowchart diagram. Tables 4-15 and 4-16 show that the flowchart diagram had a character accuracy difference of 12%. This accounted for 17 more character entry errors for the pen/voice editor while drawing the diagram. If one considers the time it takes to say a character, notice the character is incorrect, and to correct for the incorrect character 17 times, the 50 second difference in average task time is easily accounted for. Therefore, the primary reason why the flowchart was faster with the pen-only editor was because of the poor speech character recognition accuracy.

Based on the handwriting and speech character entry accuracy rate results in Tables 4-9 through 4-22, it is apparent that the speech character recognition accuracy was significantly inferior to the Graffiti handwriting character recognition accuracy. The only diagram in which the speech character recognition had a 90% or higher accuracy rate drawing was the state diagram. The best possible explanation for this is that since state diagram characters are predominantly numbers, numbers can be recognized better than letters by the speech character recognizer. Based on speech recognition problem character statistics in Tables 4-9 through 4-20, the low overall speech recognition accuracy was caused by the inability of the recognizer to distinguish between similar sounding letters.

The average speech command recognition had a 90% accuracy rate, as shown in Table 4-22. Subject responses indicated that having a command being lost on occasion did not bother them. Based on the high task reduction times of the pen/voice editor while competing with low character entry accuracy rates, it is highly probable that it was speech commands that made the pen/voice editor faster than the pen-only editor.

Table 4-23 shows that on average, there were more questions asked and more final diagram imperfections per diagram using the pen with voice editor. The increase is questions is most likely attributed to the subject’s unfamiliarity with the interface. The increase in diagram imperfections could mean that the subjects thought less about aesthetics while drawing with the pen/voice editor.
In summary, the experiment demonstrates the following:

- The addition of speech made an editor that was 5.6% faster overall. The editor was 17.0% faster for one diagram and 5.8% slower for one diagram.
- Subjects preferred the pen/voice editor and thought it was more usable and faster.
- Subjects did not think that the pen/voice editor was more user-friendly.
- The accuracy of the speech character recognizer is significantly less than the handwriting character recognizer.
Chapter 5

Evaluation

The following chapter critically evaluates the results and procedures of the experiment. In addition, comments about the experiment that did not affect the results are contained in Appendix B.8.

5.1 Subject Selection

There was more experience for the subjects using pen technology than speech technology going into the experiment. Five subjects had previous pen experience as opposed to two subjects with speech experience. This extra experience could have slightly decreased the average diagram drawing time for the pen-only editor.

The experiment needed another twelve subjects to distribute all combinations of diagrams and input modes evenly through the four drawing tasks. Because of this, diagram input tasks were not distributed evenly. The addition of the thirteenth subject who repeated subject nine’s experiment trial made the distribution even worse. Because of the uneven distribution of diagram tasks, certain diagram tasks occurred more frequently as the first couple of diagrams that a subject drew. Since the subjects were learning the editor during these tasks, the task time was generally greater. The diagrams that were drawn more often at the beginning of the experiment trials are the pen/voice petri net diagram, the pen-only state diagram, the pen-only flowchart diagram, and the pen/voice dataflow diagram. These tasks should have slower average task times than if the diagram tasks were distributed evenly.
While the unbalanced task distribution problem might effect each individual average task reduction time, two diagrams each for the pen-only and the pen/voice editors benefit overall. Because of this, this overall task time should not be varied significantly.

The subjective benefits of the experiment were determined only from the opinions of computer science graduate students. Thus, the subjective results might not be generalized to be invalid if a computer science graduate students user interface opinions are not typical of an average computer user. This is assumed not to be the case.

The subject selection for this pilot study was too categorically narrow and had too few subjects to collect conclusive data. Broader subject selection and a larger population size are needed in future experiments.

5.2 Tasks

The tasks used in this pilot study were chosen to be categorically familiar to the selected subjects. Overall, the computer science subjects appeared familiar with the tasks and four diagrams seemed to be a reasonable number of tasks. Future experiments should vary the type of tasks based on the subject group. VLSI layouts, floor plans, and employee hierarchy charts could be used for engineers, architects, and business managers respectively.

5.3 Tools

Subjects commented that they did not like having to repeat letters for speech character recognition twice. This user-friendliness data was affected by this. Either a different method of speech character recognition or a more accurate speech recognizer is needed to make speech character recognition more usable.

Pen control issues created some problems for subjects using the pen-only editor. Some subjects had a noteworthy amount of trouble tapping on the tool bar buttons which slowed down command entry. The source of this trouble was partially caused by the poor location of the buttons on the tool bar. Because of it’s awkward location, the side tool bar should have been removed with all its buttons moving to the horizontal tool bar. In
addition, all the buttons on the horizontal tool bar should have been centered on the tool bar for easier access. Since statistics relaying information about tool bar command entry accuracy were not collected, command entry accuracy can’t be compared between the two editors.

There was an inconsistency in the functionality of the two editors concerning the display of state information. The pen-only editor’s tool bar was highlighted when a toggle state was on. The pen/voice editor did not show any state information except for the microphone state. However, the video output of the pen computer made it extremely hard to discern if a button was highlighted. Subjects who were able to discern the state information from the highlighted tool bar buttons had an advantage using the pen-only editor over the pen/voice editor.

The pen-only editor had some advantage entering text labels over the pen/voice editor. The pen-only interface could use the standard Windows edit box functionality in the label entry dialog box while edit box functionality was restricted for the pen/voice editor. The pen/voice editor did not have the ability to overwrite an arbitrary portion of the text label or place the text cursor at an arbitrary position in the edit box.

The problems mentioned affected the task drawing times in this pilot study and are design issues which should be corrected for future experiments.

5.4 Procedures

Different subjects did their experimental trials at different points in the day. This factor could affect how the subjects felt about the experiment and could have biased the drawing task times.

Since the experimental trials were not recorded, some valuable subjective data was lost. Also, having this data would have made the recognizer accuracy figures exact rather than approximate.

Subjects were told the goal of the experiment before doing the experiment. This might have caused the subjects to form opinions that might have affected the experiment trials.
5.5 Statistical Analysis

The average task times and the overall average task time of 5.6% are not significant figures considering the relatively small subject sample size and the wide dispersal of results. Because of this, rigorous statistical analysis is not warranted for this pilot study.

The questionnaire results could be biased by the subjects enthusiasm toward using new interface technology. More accurate subjective results could be obtained if subjects used the interfaces for a period of time before the experiment.

All statistics collected in the experiment are based on experiences using the experimental pen-only and pen/voice editor. The subjective interface data might be partially based on the subjects opinions about these editor tools.
Chapter 6

Future Directions

Possible future work stems into many directions including extending this work further, more multimodal input research and new sound recognition input methods.

I would like to repeat the experiment from Chapter 4 using a pen-only editor that contains a quality gesture recognizer and a pen/voice editor with equivalent functionality. Although not shown in this experiment, I believe speech text entry has the potential for being faster than handwriting text entry. Research in this area, with future speech recognition tools that have a higher accuracy and an infinite vocabulary, could find different results then found in this work.

Expanding the pen/voice editor to have some keypress device as another input modality source might give new ideas how multimodal editors can benefit from three or more input devices. The work done in this thesis with pen and voice multimodal interfaces might give some ideas how other types of multimodal systems can be designed. Some research on graphical editing using new virtual reality systems would be of research interest also. Finally, research examining simultaneous pen and speech input methodology might uncover new ways to speed up multimodal pen/voice applications.

The insight gained from this thesis involving graphical editors could be extended to do research in more advanced graphical programs such as CASE tools. In addition, similar experiments would be worthwhile for other types of application domains.
Speech recognition is not the type of acoustical input modality. Similar research using various non-speech audio input would be beneficial. Research involving musical instrument input could potentially find new and better ways to input information.
Chapter 7

Conclusion

The thesis that speech recognition can increase the speed, accuracy, usability, and user-friendliness of a pen-only interface in a developed system is partially shown. On average, speech recognition slightly increased the speed of drawing diagrams in a pen-based graphical editor. Speech recognition subjectively improves the usability of a pen-based editor, but could not make the same claim for user-friendliness conclusively. Finally, the accuracy of the implemented speech character recognizer has been shown to be worse than the accuracy of the handwriting character recognizer. Because command accuracy was not compared, speech recognition has not been shown to improve the accuracy of a pen-based graphical editor. Further research based on this pilot study should be conducted for more conclusive results.

Most of the subjects had good things to say about speech recognition and seemed to enjoy using the multimodal pen with speech recognition interface. Although the novelty of the pen/voice interface accounts for some of this enthusiasm, my perception was that the subjects would enjoy using the interface for various everyday work.
Appendix A

Picasso User’s Manual

A.1 Development Platform

Picasso was written using Borland\textsuperscript{12} C++ 4.51 using OWL 2.5. The speech recognition engine used for the editor's design was Dragon's VoiceTools 1.1 Windows Supplement. The character handwriting recognition program used was Palm Computing's Graffiti demo version.

A.2 The Fundamentals of Picasso

Introduction

Picasso is a graphical editing application that is designed to quickly draw diagrams that consist of lines, rectangles, ellipses, triangles, and diamonds. The current version of the editor is implemented for use with Microsoft Windows 3.1 or greater. Although Picasso is optimized for use with a pen user interface, the editor will function adequately on a keyboard-mouse based system.\textsuperscript{13} Picasso's main advantage over other editors is it's

\begin{footnotesize}
\begin{itemize}
\item \textsuperscript{12}Borland is a trademark of Borland International, Inc.
\item \textsuperscript{13}A tap is defined as a pen down event followed immediately by a pen up event. A double tap is two taps in rapid succession. A stroke is defined as a pen down event, followed by any number of pen movement events, and concluding with a pen up event. The following documentation assumes that a pen-based user interface is used. Mouse users should substitute a pen down event with a left mouse button down, a pen move event with a mouse move, and a pen up event with a left mouse button up. Also, a pen tap is equivalent to a left button mouse click and a pen double tap translates to a left button mouse double click.
\end{itemize}
\end{footnotesize}
built-in shape recognition algorithms. These algorithms transform sketched single or multiple stroke drawings into perfected symmetrical shapes required for aesthetic graphical diagrams.

**Picasso Functionality**

**Drawing:**

The defaults of Picasso start the editor up in shape recognition mode. In this mode, sketched stroke drawings are transformed into perfected shapes. Figure A-1 shows how typical transformations might look.

![Figure A-1. Shape Recognition Transformations](image)

To start drawing a shape, move the pointer to a location into an area of the drawing window that is not within the selection region of a previously selected shape (see Selection). With the pen draw one or more strokes. For each stroke that is drawn, there is a delay of 200 mS + (20 * Number of Points in the Stroke) mS between a finished
stroke and automatic shape recognition activation. In order to draw multiple strokes, the next stroke must be started before this activation time-out occurs.

After the strokes are processed by the shape recognition algorithms, one of two things will happen. The shape recognition algorithm calculates the parameters for a bounding rectangle which all the points must fit inside. If either the height or the width of the bounding rectangle is 5 pixels or less, a selection event will occur. Otherwise the editor displays the shape recognizer's calculated shape in place of the original freehand drawing.

**Selection:**

Selection of an enclosed shape occurs by tapping inside of the shape. Resize handles will appear on the shape as show in Figure A-2. The corresponding deselection of a shape is achieved by tapping inside the enclosed area of the selected shape. Once the deselection is finished, the resize handles will disappear from the shape. Selection and deselection of lines are achieved by tapping within 10 pixels of the line. If a tap is within the selection region for multiple shapes, the shape with the lowest bounding rectangle area is acted upon. In case of multiple shapes having the same area, the most recently drawn shape is used.

In shape recognition mode, tapping outside the selection region for all shapes has the following effect. If any shapes were previously selected, the tap acts as a deselect all selected shapes. Otherwise, the tap will cause all shapes to be selected. Thus, tapping

![FIGURE A-2. An Example Selected Object](image-url)
outside of all selection regions can be used to toggle between all shapes selected and all shapes deselected.

Labeling:

Double tapping on a shape causes the shape to toggle its selection and causes the text label dialog box, show in Figure A-3, to appear. At this point, enter the text label and tap the OK button. The label will appear centered within the shape.

![Text Label Dialog Box](image)

**FIGURE A-3. The Text Label Dialog Box**

Moving:

To move shapes around the drawing window, select all shapes which are to be moved. Press the pen down on one of the selected shapes and drag to selected shapes to the desired location. All the selected shapes will move simultaneously. Pick the pen up off the drawing surface to finish the move action.

Resizing:

To resize shapes, select all shapes which are to be resized. Press the pen down on one of the selected shapes resize handles and drag to selected shapes until it reaches the desired size. All the selected shapes will resize simultaneously. Pick the pen up off the drawing surface to finish the resize action.

Resizing with the center handle has the following effect. Resizing left will decrease the width of the shape and resizing right will increase the width. Similarly, moving up and
down will shrink and expand the height respectively. Resizing with corner handles will anchor the corner diagonal to the active handle. Resizing with side handles will anchor the opposite side to the active side handle.

### A.3 The Pen-only Version of Picasso

**Introduction**

The pen-only version of Picasso contains all of the functionality discussed in the fundamentals of Picasso section. All other functionality in the editor is implemented as buttons on a button bar. Figure A-4 shows a screen shot of the editor. The menu for the editor contains functionality which will be described in the additional functionality of Picasso section.

![Figure A-4. A Screen Shot of the Pen-only Version of Picasso](image-url)
Functionality

- **Delete Selected Objects:**

  Tapping on this button will permanently removes all selected objects from the drawing window. The drawing window will update to reflect the object’s removal.

- **Cut Selected Objects:**

  Tapping on this button will remove all selected objects from the drawing window and copy them over to the clipboard. The drawing window will update to reflect the object’s removal. Groupings will not be copied over to the clipboard.

- **Copy Selected Objects:**

  Tapping on this button will copy all selected objects on the drawing window to the clipboard. Groupings will not be copied over to the clipboard.

- **Paste Clipboard Objects:**

  Tapping on this button will paste objects on the clipboard onto the drawing window. The drawing window will update to reflect the newly added objects.

- **Group Selected Objects:**

  Tapping on this button will group all selected objects. While objects are grouped, selection or deselection of one of the grouped objects will respectively select or deselect all the other grouped objects.

- **Ungroup Selected Objects:**

  Tapping on this button will ungroup all selected objects. The selected objects will no longer be selected or deselected as a group.
- **Change the Last Drawn Shape to a Rectangle:**

  Tapping on this button will change the last shape that was drawn into a rectangle. The drawing window will update to reflect the change.

- **Change the Last Drawn Shape to an Ellipse:**

  Tapping on this button will change the last shape that was drawn into an ellipse. The drawing window will update to reflect the change.

- **Change the Last Drawn Shape to a Triangle:**

  Tapping on this button will change the last shape that was drawn into a triangle. The drawing window will update to reflect the change.

- **Change the Last Drawn Shape to a Diamond:**

  Tapping on this button will change the last shape that was drawn into a diamond. The drawing window will update to reflect the change.

- **Switch to Shape Mode:**

  Tapping on this button changes the editor into shape mode for shape recognition. This button will highlight when pressed and remove the highlight from the text mode button.

- **Switch to Text Mode:**

  Tapping on this button changes the editor into text mode for text entry. This button will highlight when pressed and remove the highlight from the shape mode button.
- **Toggle Line Arrow Heads:**

  Tapping on this button toggles arrow heads on and off. All future lines that are drawn will contain an arrow head at the end if arrow heads are on. The editor defaults arrow heads off. This button will highlight when arrow heads are on.

- **Toggle Automatic Labeling:**

  Tapping on this button toggles automatic shape labeling on and off. All future shapes that are drawn in shape mode will automatically prompt for a label upon shape recognition if automatic labeling is on. The editor defaults automatic labeling off. This button will highlight when automatic labeling is on.

- **Toggle Grid Display:**

  Tapping on this button toggles the grid display on and off. The editor defaults grid display off. This button will highlight when grid display is on.

- **Toggle Snap to Grid:**

  Tapping on this button toggles grid snapping on and off. While snap to grid is on, drawn objects will snap to the nearest grid line. Moving and resizing objects will move or resize in grid increments. The editor defaults snap to grid on. This button will highlight when snap to grid is on.

- **Expand the Grid Increment Size:**

  Tapping on this button expands the size of the grid increment by a factor of 2.

- **Shrink the Grid Increment Size:**

  Tapping on this button shrinks the size of the grid increment by a factor of 2.
Text Mode Operation

To enter text objects into the editor, Picasso uses a text entry mode rather than the shape recognition mode for shape entry. Switching to text mode is done through the switch to text mode button on the button bar. Switching back to shape mode is done through the switch to shape mode button on the button bar.

Moving, resizing, labeling, and selection of singular objects works identically in text mode as it does in shape mode. However, the select/deselect all functionality is removed.

In order to enter text in text mode, tap outside of the selection area for all objects in the drawing window. A cursor will appear in the location of the tap as shown in Figure A-5.

![Drawing Window 1](image)

**Figure A-5. Picasso's Text Mode Cursor**

Once the cursor is displayed characters are entered through the Graffiti handwriting character recognition program. As the characters are recognized, they will appear in the drawing window. The text mode only allows a single line of text, so the return character will not have any function.

Once the user taps the pen in the drawing window again, a new text object is created and the old text object is considered finished. At this point, the editor does not have the ability to edit previous entered text objects. The text objects may be moved in the standard manor. Resizing text objects has no function in this version of the editor.
A.4 The Pen/Voice Version of Picasso

Introduction

The pen with voice version of Picasso contains all of the functionality discussed in the fundamentals of Picasso section. With the exception of the toggle microphone active button, all other functionality in the editor is implemented as speech commands. Figure A-6 shows a screen shot of the editor. The menu for the editor contains functionality which will be described in the additional functionality of Picasso section. The speech recognition for the editor needs to be trained before it is used. There are 26 speech commands, 26 letters, 10 digits and 4 miscellaneous characters in the speech vocabulary for a total of 66 words.

![Screen Shot of Picasso Editor](image)

FIGURE A-6. A Screen Shot of the Pen/Voice Version of Picasso
System Requirements

In order to use the pen/voice version of Picasso, a run-time license for Dragon VoiceTools is required. In addition, a high quality multimedia sound card and a professional quality high-gain low-noise microphone is necessary for speech recognition.

Functionality

- **Toggle Microphone Active:**

Tapping on this button toggles the microphone on and off. Voice recognition is only active when the microphone is on. The editor defaults microphone active off. This button will highlight when the microphone is on. In addition the text display next to the button will also specify if the microphone is on or off.

“Microphone” - **Turn Microphone Off:**

Speaking “microphone” will turn the microphone off to disable voice recognition input. The text display next to the toggle microphone active button will specify if the microphone is on or off.

“Delete” - **Delete Selected Objects:**

Speaking “delete” will permanently removes all selected objects from the drawing window. The drawing window will update to reflect the removal of the objects.

“Cut” - **Cut Selected Objects:**

Speaking “cut” will remove all selected objects from the drawing window and copy them over to the clipboard. The drawing window will update to reflect the removal of the objects. Groupings will not be copied over to the clipboard.
“Copy” - Copy Selected Objects:

Speaking “copy” will copy all selected objects on the drawing window to the clipboard. Groupings will not be copied over to the clipboard.

“Paste” - Paste Clipboard Objects:

Speaking “paste” will paste objects on the clipboard onto the drawing window. The drawing window will update to reflect the newly added objects.

“Group” - Group Selected Objects:

Speaking “group” will group all selected objects. While objects are grouped, selection or deselection of one of the grouped objects will respectively select or deselect all the other grouped objects.

“Ungroup” - Ungroup Selected Objects:

Speaking “ungroup” will ungroup all selected objects. The selected objects will no longer be selected or deselected as a group.

“Select” - Select the Last Drawn Object:

Speaking “select” will select the last shape that was drawn. The drawing window will update to reflect the selection of the object.

“Deselect” - Deselect the Last Drawn Object:

Speaking “deselect” will deselect the last shape that was drawn. The drawing window will update to reflect the deselection of the object.
“Select All” - Select all Objects:

Speaking “select all” will select all objects in the drawing window. The drawing window will update to reflect the selection of the objects.

“Deselect All” - Deselect all Objects:

Speaking “deselect all” will deselect all objects in the drawing window. The drawing window will update to reflect the deselection of the objects.

“Rectangle” - Change the Last Drawn Shape to a Rectangle:

Speaking “rectangle” will change the last shape that was drawn into a rectangle. The drawing window will update to reflect the change.

“Ellipse” - Change the Last Drawn Shape to an Ellipse:

Speaking “ellipse” will change the last shape that was drawn into an ellipse. The drawing window will update to reflect the change.

“Triangle” - Change the Last Drawn Shape to a Triangle:

Speaking “triangle” will change the last shape that was drawn into a triangle. The drawing window will update to reflect the change.

“Diamond” - Change the Last Drawn Shape to a Diamond:

Speaking “diamond” will change the last shape that was drawn into a diamond. The drawing window will update to reflect the change.

“Shape” - Switch to Shape Mode:

Speaking “shape” changes the editor into shape mode for shape recognition.
“Text” - *Switch to Text Mode:*

Speaking “text” changes the editor into text mode for text entry.

“Arrows” - *Toggle Line Arrow Heads:*

Speaking “arrows” toggles arrow heads on and off. All future lines that are drawn will contain an arrow head at the end if arrow heads are on. The editor defaults arrow heads off. This button will highlight when arrow heads are on.

“Object Labeling” - *Toggle Automatic Labeling:*

Speaking “object labeling” toggles automatic shape labeling on and off. All future shapes that are drawn in shape mode will automatically prompt for a label upon shape recognition if automatic labeling is on. The editor defaults automatic labeling off.

“Grid” - *Toggle Grid Display:*

Speaking “grid” toggles the grid display on and off. The editor defaults grid display off.

“Snap To Grid” - *Toggle Snap to Grid:*

Speaking “snap to grid” toggles grid snapping on and off. While snap to grid is on, drawn objects will snap to the nearest grid line. Moving and resizing objects will move or resize in grid increments. The editor defaults snap to grid on.

“Expand” - *Expand the Grid Increment Size:*

Speaking “expand” expands the size of the grid increment by a factor of 2.

“Shrink” - *Shrink the Grid Increment Size:*

Speaking “shrink” shrinks the size of the grid increment by a factor of 2.
Training

Since the speech recognition engine used in the pen/voice version of Picasso is speaker dependent, the speech recognition must be trained. Figure A-7 shows the voice training dialog box used for training the speech recognition vocabulary. For this version of the editor, training must be initiated by the system manager.

![Voice Training Dialog](image)

**FIGURE A-7. Picasso’s Voice Training Dialog Box**

In order to use the speech recognition to train the system, the microphone button must be tapped to turn the microphone on. Tapping the microphone button again will toggle the microphone off. The status of the microphone is displayed at the bottom of the dialog box.

The user needs to say the word in the black box four times to completely train the word. The word must be trained using a tone of voice, pitch, and volume nearly identical to how it will be used in the editor. The alphabet will be displayed as “AA” through “ZZ” and should be trained by saying “A A” through “Zee Zee”. This was done to increase the accuracy of letter speech recognition.
The repetition counter for the word is displayed immediately below the black box. If the user makes a mistake, the repeat word button allows the user to retrain the word contained in the black box. The repetition count will reset to the first count. The previous word button allows the user to move back to the word trained immediately before the word in the black box.

Pressing the cancel button will cancel the speech recognition training and return the editor to it's state previous to the training. All words that were trained before the cancel button was hit will still be trained after the cancel operation.

**Display Spoken Word Dialog Box**

The display spoken word dialog box is shown in Figure A-8. This dialog box display's the 5 last words recognized by the speech recognizer. For this version of the editor, the dialog box must be initiated by the system manager.

![Voice Words Dialog Box](image)

**FIGURE A-8. Picasso’s Display Spoken Words Dialog Box**

**Entering Text with Speech Recognition**

Text entry is entered with speech recognition for labeling and text mode text entry. The alphabet characters are entered by saying "AA" through "ZZ" for A through Z respectively. Digits are entered by saying the digits "0" through "9". In addition saying "space", "back", "slash", or "question" recognizes to the space character, a backspace, a slash, or a question mark respectively.
Text Mode Operation

To enter text objects into the editor, Picasso uses a text entry mode rather than the shape recognition mode for shape entry. Switching to text mode is done by speaking “text”. Switching back to shape mode is done by speaking “shape”.

Moving, resizing, labeling, and selection of singular objects works identically in text mode as it does in shape mode. However, the select/deselect all functionality is removed.

In order to enter text in text mode, tap outside of the selection area for all objects in the drawing window. A cursor will appear in the location of the tap as shown in figure 4-6.

Once the cursor is displayed characters are entered as explained in the entering text with speech recognition section. As the characters are recognized, they will appear in the drawing window. The text mode only allows a single line of text.

Once the user taps the pen in the drawing window again, a new text object is created and the old text object is considered finished. At this point, the editor does not have the ability to edit previous entered text objects. The text objects may be moved in the standard manor. Resizing the objects is possible, but has no function in this version of the editor.

A.5 Additional Functionality of Picasso

The versions of the Picasso graphical editor that were used in the experiment were scaled down versions of a fully functional editor that we have designed at Washington University. The following text briefly describes the menu functionality from the fully functional editor that was used in the experiment. This functionality was not available to the subjects in the experiment. Only the tester for the experiment could use this functionality.
Experiment Menu Functionality

Save Drawing Window (File-Save)

This action is used to save the diagram in a drawing window to disk. Once this option is selected from the menu, a dialog box will pop up that asks for a filename to save the diagram to.

Save User Speech Recognition Information (Voice-Save User File)

This action is used to save the trained user speech recognition vocabulary to disk. Once this option is selected from the menu, a dialog box will pop up that asks for a filename to save the user information to.

Speech Recognition Filenames (Voice-Voice Filenames)

This action is used to enter the default files used for the speech recognition. Once the speech recognition is trained and saved, the new user file must be specified as the user file. Once this option is selected, a dialog box will appear that lets the user edit the filenames. Enter the new user file in the user file edit control.

Train the Speech Recognizer (Voice-Train)

Selecting this action will pop up the dialog box shown in Figure A-7. Follow the instructions given in Section A.4 to train the system.

Activate the Display Spoken Words Dialog Box (Voice-Display Words)

Selecting this action will pop up the dialog box shown in Figure A-8. The description of this dialog box is given in Section A.4.
A.6 Graffiti

The Graffiti character handwriting recognition program can be used for text entry in Picasso. Figure A-9 shows a screen shot of the program.

![Graffiti Demo](image)

**FIGURE A-9. A Screen Shot of Graffiti**

Graffiti has five buttons. The caps/CAPS button toggles the capitalization mode. The abc/123 button toggles between letter mode and digit mode. The ‘<‘ and ‘>‘ buttons emulate cursor left and cursor right keys respectively. The ‘?’ button shows Figure A-10.

![Graffiti Reference Card](image)

**FIGURE A-10. Graffiti’s Character Entry Reference Card**

In order to input characters, write the appropriate symbol in the Graffiti drawing window, starting from the dot, as shown in Figure A-10.
A.7 Audio Picasso

In order to learn more about how to add speech functionality to a graphical editor, a speech-only version of the Picasso graphical editor called Audio Picasso was developed [33]. In this editor, different methods of command entry with speech were examined. Section A.7.7 contains the command list used for this editor. The functionality was evaluated for usefulness in the pen/voice editor.

A.7.1 Commands

Overall, speech showed to be a quick and convenient way to give commands normally contained on a menu or tool bar to the computer. The one situation where these speech commands seemed awkward was where the same command was performed three or more consecutive times. In this situation, using the tool bar in a multimodal editor should be faster and less tedious.

A.7.2 Drawing

Drawing was accomplished through specifying the shape, location, and size. The possible shapes were lines, squares, circles, triangles, and diamonds. Location was one of the following nine drawing window locations: ‘top’, ‘top-right’, ‘right’, ‘bottom-right’, ‘bottom’, ‘bottom-left’, ‘left’, ‘top-left’, ‘center’. Size was specified as ‘small’, ‘medium’, or ‘large’ and the final size was proportional to the size of the drawing window. Upon specifying a drawn shape, move and resize actions usually needed to be done on the shape which made this drawing method inefficient and monotonous. Drawing with a pen is a much more efficient method of creating shapes.

A.7.3 Moving

Several methods of movement were examined. The first way was to tell the computer to move x units on a grid in a given direction. The amount of grid units could be a number between 1 and 9 and the directions consisted of ‘left’, ‘right’, ‘up’, and ‘down’. This method showed to an effective method for moving approximate distances quickly. The next method tried was to give a command specifying a direction and a phrase
representing how much to move. The directions were the same as for the grid and the
distance phrases consisted of ‘a little’, ‘some’, ‘a lot’, and ‘away’ which corresponded to
a distance proportional to the screen size. Although the language syntax is a little more
intuitive, the inexact amount of movement made this method hard to use.

Another method used for moving objects around the screen was specifying both location
and distance with one word. This was done by varying the speed at which the direction is
uttered. There were three speed levels for each direction. For example, moving an object
up would be accomplished in increasing distance by a ‘move’ ‘up’, a ‘move’ ‘uupp’, and
a ‘move’ ‘uuuppp’ command. This method did not work well as the recognizer doesn’t
pick up the differences in the distance words accurately.

In order to reduce the amount of spoken words necessary for a direct manipulation action
to occur, consecutive movement commands could be entered by just specifying either the
direction or distance. The unspecified parameter defaults to the value used in the last
move command. For example, if an object needed to be move one grid increment up and
one grid increment to the left, the sequence ‘move’ ‘up’ ‘one’ ‘left’ would achieve this.

All the speech methods examined for discrete movement were much harder to use than
direct manipulation with pen interfaces. Since the pen can move objects continuously, a
method for spoken continuous movement was devised. A continuous movement mode
was toggled on/off by speaking the word ‘continuous’. In continuous movement mode,
once a discrete movement command was spoken, the command would be repeated until
the word ‘stop’ was spoken. There were two speeds which continuous commands would
repeat which were specified by speaking ‘slow’ or ‘fast’. Although continuous
movement commands were powerful, it is often easy to loose control of the movements if
a speech word was not recognized or if the objects were moving in large increments.
Even with the continuous movement functionality, speech only movement could not be
controlled well enough to use in place of pen movement functionality.

A.7.4 Resizing

Resizing commands used the same general methodology as movement commands.
However, resizing actions had many more complications. The user needed to specify if
resizing should enlarge or shrink the object. In addition, unlike movement commands, the final results of resize actions are usually determined through trial and error rather than before starting the actions. Because of these two factors, it was almost impossible to control the speech resize functionality precisely for a desired effect.

Some functionality was added for line resizing. The spoken word ‘rotate’ was used to rotate a line 45 degrees about the drawn start point. In addition the phrase ‘other way’ was used to rotate a line 180 degrees about the drawn start point. This functionality significantly helped in resizing lines.

In order to ease resizing, scale up and scale down commands were added for simple scale resizes about the center of an object. This functionality helped in situations where the general structure of the shape didn’t need to be changed, but didn’t ease overall resize functionality.

**A.7.5 Selection**

Several methods of selection were examined. Using the spoken words ‘select’ and ‘select all’ the last drawn object or all objects could be selected. Similar words were used for deselection of objects. These methods of selection worked well, but could not solve the problem of selecting any given object.

One method tried for selection of any given object was to set up a current object pointer which cycled through all the objects. When the pointer was at an object which needed to be selected or deselected, the ‘select current’ or ‘deselect current’ phrases were spoken. This method was difficult to use when there were a large amount of objects to cycle through in a diagram.

Another method tried for selection involved setting up dynamic voice labels for one or more shapes. Upon speaking a voice label, the objects attached to the label toggled their selection. Although voice label selection simplified the selection of objects, there was a significant amount of overhead attached to setting up the functionality. Voice labels could only be set up on the last drawn object, the current pointer object, or the selected
objects. In addition, in order to set up the dynamic voice label, the voice label needed to be trained for speech recognition.

The only selection functionality which seemed appropriate to use outside of a speech only interface was the selection of the last drawn object and the select all objects functionality.

A.7.6 Text Labeling

Specifying an object to text label in the speech-only editor had the same issues as selection of an object. As with selection, the only methodology that seemed to be useful outside of a speech only interface was the label the last drawn object functionality.

A.7.7 Command List

File Voice Commands

"New" - Create a new drawing in a new drawing window.
"Open" - Locate and open a file.
"Insert" - Insert a drawing file into the current drawing window.
"Save" - Save the drawing in the current window.
"Save As" - Save the current drawing, but with a new filename.
"Close" - Close the current drawing window.
"Save Bitmap" - Save the current drawing as a Windows bitmap file.
"Print" - Print the current drawing.
"Print Config" - Select and configure a printer.
"Exit" - Exit Picasso.

Editing Voice Commands

"Delete" - Delete the selected objects.
"Cut" - Remove the selected objects and put them on the clipboard.
"Copy" - Place a copy of the selected objects on the clipboard.
"Paste" - Paste the contents of the clipboard into the current drawing window.
"Clear" - Erase all the objects in the current drawing window.
"Recognize" - Re-recognize the selected strokes.
"Select All" - Select all the objects in the current drawing window.
"Deselect All" - Deselect all the objects in the current drawing window.

Shape Selection Voice Commands

"Line" - Draw lines in draw shape mode.
"Rectangle" - Draw rectangles in draw shape mode.
"Ellipse" - Draw ellipses in draw shape mode.
"Triangle" - Draw triangles in draw shape mode.
"Diamond" - Draw diamonds in draw shape mode.
"Text" - Draw text in draw shape mode.

Line Specific Voice Commands

"Rotate" - Rotate a line 45 degrees.
"Other Way" - Rotate a line 180 degrees.
"Arrows" - Toggles arrows on lines on/off.

Editor Mode Selection Voice Commands

"Shape" - View objects as recognized shapes.
"Ink" - View objects as the scribbles or strokes that compose them.
"Scribble" - When in Ink Mode, treat groups of pen strokes as a single scribble.
"Stroke" - When in Ink Mode, allow manipulation of individual pen strokes.
"Draw Mode" - Draw shapes without recognition.
"Shape Mode" - Recognize the drawings as shapes.

Grid Voice Commands

"Grid" - Toggles grid on/off.
"Shrink" - Halves the size of the grid squares.
"Expand" - Doubles the size of the grid squares.
Window Manipulation Voice Commands

"Cascade" - Cascade the drawing windows.
"Tile Vertically" - Vertically tile the drawing windows.
"Tile Horizontally" - Horizontally tile the drawing windows.
"Arrange Icons" - Line the iconized windows up at the bottom of the desktop window.
"Switch" - Activate the next window.

Help Voice Commands

"About" - Display credits and version information about Picasso.

Voice Label Voice Commands

("Label Name") - Select the objects associated with the voice label.
"Label" - Voice label the last drawn shape object.
"Label Current" - Voice label the current object.
"Label Selected" - Voice label the selected objects.

Current Shape Voice Commands

"Next" - Set the next shape as the current shape.
"No Current" - Clear the current shape iterator.
"Select Current" - Select the current shape.
"Deselect Current" - Deselect the current shape.

Last Drawn Shape Voice Commands

"Select" - Select the last drawn shape.
"Deselect" - Deselect the last drawn shape.
Continuous Movement Voice Commands

"Continuous" - Toggle continuous movement mode on/off.
"Stop" - Stop actions from repeating in continuous movement mode.
"Fast" - Fast repeat command timers in continuous movement mode.
"Slow" - Slow repeat command timers in continuous movement mode.

Voice Recognition Specific Voice Commands

"Train" - Train the vocabulary through the Train Words Dialog.
"Microphone" - Toggle the microphone on/off.

Command Language

Rules for Below Command Language:
(+ = One or More)
(* = Zero or More)
- In text draw mode, because numbers can be entered as characters, once one command is executed, the user must enter a full command to do repeat an action. Ex: “Move Up Left” would now have to be “Move Up Move Left”.
- When using continuous movement mode, current actions are repeated until “Stop” is recognized. “Move Up Left Stop” will move up until “Left” is recognized and then will move left until “Stop” is recognized.
- Emphasized words such as “Uuuppp” have increment arguments larger than their smaller emphasis words such as “Up”.
- The direction words that are emphasized, such as “Uuuppp”, act as de-emphasized if they don’t use their direction as the increment argument also. Ex “Move 9 Uupp” is equivalent to “Move 9 Up”.

Draw Voice Command Language

“Draw” <Size Word> + <Location Word>
“Draw” <Location Word> + <Size Word>
Move Voice Command Language

"Move" <Number Word>+ <Direction Word> <Inc-Dir Word>*
"Move" <Increment Word>+ <Direction Word> <Inc-Dir Word>*
"Move" <EmphasizedDirection Word> <Inc-Dir Word>*

Resize Object Voice Command Language

"Resize Larger"
"Resize Smaller"
"Resize" <Number Word>+ <Direction Word> <Inc-Dir Word>*
"Resize" <Increment Word>+ <Direction Word> <Inc-Dir Word>*
"Resize" <EmphasizedDirection Word> <Inc-Dir Word>*

Scaling Resize Object Voice Command Language

"Scale" <Number Word>+ <Scale Direction Word> <Scale Inc-Dir Word>*
"Scale" <Increment Word>+ <Scale Direction Word> <Scale Inc-Dir Word>*
"Scale" <Scale EmphasizedDirection Word> <Scale Inc-Dir Word>*

Cursor Movement for Text Editing Voice Command Language

"Cursor" <Number Word>+ <Direction Word> <Inc-Dir Word>*
"Cursor" <Increment Word>+ <Direction Word> <Inc-Dir Word>*
"Cursor" <EmphasizedDirection Word> <Inc-Dir Word>*

Text Entry Voice Commands (at Text Cursor)

<Number Word> | <Character Word>

<Inc-Dir Word>

<Inc-Dir Word> = <Number Word> | <Increment Word> | <EmphasizedDirection Word>
<Scale-Dir Word>

<Scale Inc-Dir Word> = <Number Word> | <Increment Word> | <Scale Direction Word>

<Number Word>

"0" - "9"

<Character Word>

"A" - "Z"

<Location Words>

"Top" | "Bottom" | "Left" | "Right" | "Top-Left" | "Top-Right" | "Bottom-Left" | "Bottom-Right" | "Center"

<Size Words>

"Large" | "Medium" | "Small"

<Increment Words>

"A Little" | "Some" | "A Lot" | "Away"

<Scale Direction Words> | <Scale EmphasizedDirection Words>

"Up" | "Uupp" | "Uuuppp" | "Down" | "Ddoowwwnn" | "Dddooowwwnnn"

<Direction Words> | <EmphasizedDirection Words>

"Up" | "Uupp" | "Uuuppp" | "Down" | "Ddoowwwnn" | "Dddooowwwnnn" | "Left" | "Lleeffftt" | "Lleeeefffttt" | "Right" | "Rriiigghhtt" | "Rrriiigghhhttt"
Appendix B

Experiment Procedures and Handouts

Section B.1 contains the handout that subjects were told to read before doing the experiment.

Section B.2 contains the experiment tester’s instructions.

Section B.3 contains the experiment training instructions for the experiment.

Section B.4 contains reference sheets that were available while doing the experiment.

Section B.5 contains the four diagrams and diagram drawing instructions.

Section B.6 contains the questionnaire that was given to subjects at the end of their experiment trial.

Section B.7 contains the testing sheets used by the experiment tester to collect results during the diagram drawing tasks.

Section B.8 contains some comments about the experiment.
B.1 Thesis Experiment Preparation Handout

B.1.1 Purpose of the Experiment

My research is trying to determine if adding of speech recognition to a pen based graphical editor can enhance the speed, usability, and friendliness of the editor. Since the possible enhancements are somewhat subjective, I need to collect data from a set of subjects to make conclusions. The data will be collected from diagrams drawn with two versions of a graphical editor. One version will only accept pen input and the other version will accept only pen and speech input. There will be no keyboard available for either version.

B.1.2 Recognizers in the Experiment

There are three types of recognizers used in this experiment. The first is a shape recognizer which translates freehand pen drawing into shapes. The second is a handwriting character recognizer which translates single stroke patterns, written with the pen, into characters for text entry. The final recognizer is a speech recognizer which translates words and phrases spoken into a microphone into commands. Unlike the previous two recognizers, the speech recognizer needs to be trained before use.

B.1.3 Functionality of the Editors

You will be using both a pen-only version and a pen with voice version of the Picasso graphical editor. Both editors are scaled down to have approximately the same functionality.

The following are the actions that the editor can do on objects: (draw, move, resize, label, delete, cut, copy, paste, group, ungroup, and change the shape of the last drawn object [for shape recognition]).

The following are the options that can be toggled on and off: (arrow heads for lines, automatic text labels for objects, display grid, snap to grid, and microphone [for the
pen/voice version]). There will be commands available to shrink and expand the grid. These grid control commands will be available whether or not view grid is selected since there is a snap to grid option. The only option defaulted on is snap to grid.

There will be two modes for the editor. One is a shape recognition mode in which what you draw with the pen will be translated to be a line, rectangle, ellipse, triangle, or diamond. The other mode is a text entry mode where text objects can be entered.

**B.1.4 Procedure for the Experiment**

There will be three phases to the experiment. The first phase will be an experimental preparation phase in which you setup and learn to use the Picasso graphical editor. During this phase you will learn to use Graffiti (a handwriting character recognition tool), learn and try out the editor’s basic commands, and voice train the words in the vocabulary used by the editor.

In order for Graffiti to recognize a written character, it has to be written in a particular way. During phase 1, I will give you a handout on how to write the characters, and you will write each character that will be used in the experiment at least twice. If you have trouble writing any particular character keep on writing it as many times as necessary. The recognizer is very good, so don’t expect many problems.

In order for speech recognition to be accurate, words must sound the same when trained as when used. This means that you must be careful to use the same tone of voice, pitch, and speed during training as you will use during editing. Use a relaxed voice at a slightly quickened speed and a slightly commanding tone of voice. I recommend practicing a little before the experiment using commands like cut and paste. Remember you want to command the computer to do something.

In addition to saying the words the same for training and use, the acoustics for the microphone must be similar. This means microphone placement is very important. In order to deal with this, you will leave the microphone headset on from the point you train until you draw your last diagram using speech recognition.
If you make a mistake while training, you can redo training for the word. There will be a repeat word button and a previous word button in the voice training dialog box.

Letters will be spoken as two consecutive letters back to back. For instance a T would be spoken as “Tee-Tee” and will be displayed as “TT” during training.

The second phase of the experiment will start by learning how to use either a pen or pen/voice specific version of the editor. This is followed by drawing a diagram with the editor you just learned to use. Next, you will learn how to use the other version of the editor and draw a diagram with it. Finally, you will draw one more diagram with each version of the editor.

All tasks will be done on a 486 pen computer running Microsoft Windows 3.1. The pen computer is a little slow, but performs fairly well.

For each diagram, I will give you a version I drew using Microsoft PowerPoint. Along with the diagram, I will give you a paragraph on what I’m looking for from you when you draw the diagram in Picasso. The paragraph may also give some methodology for drawing the diagrams.

The diagrams that will be drawn will all have shapes connected by lines and some text entry. A typical diagram will have 8 shapes, 14 lines, and 10 text entries.

Try not to use a hurried pace while drawing the diagrams. I’m going to be doing timings, but I’m looking for the most natural time to draw the diagrams.

When using speech recognition, there may be a few letters or digits that just don’t get recognized well. If you can’t get these characters recognized after three attempts, just leave whatever character was recognized in it’s place.

I will be keeping track of all errors that occur in the recognizers. This includes all mis-recognitions and non-recognitions for the shape, handwriting, and speech recognition.
I'm not asking that you duplicate the original diagrams exactly. Unless noted, I'm looking for diagram that you would feel comfortable using in a discussion or presentation.

Text labels have a couple of quirks. One is that labels for lines might be placed on top of the line. Another is that moving and resizing objects with labels draws the label in white (overwriting anything underneath) as the label moves. Don't be disturbed by these quirks.

The final phase of the experiment will be a questionnaire. The questions will be short answer questions geared toward determining subjective conclusions from your experiment results.

It is expected that the experiment can be completed in 2 hours.

**B.1.5 Final Comments**

You may ask questions at any point during the experiment. If the microphone is on, make sure that you turn it off before asking.

Please don't say anything to other subjects, before they do the experiment, that would compromise the objectivity of the experiment. Most importantly, don't discuss your feeling towards about a particular input modality.

If you have about the experiment please feel free to ask me before the experiment. If they are questions that don't compromise the experiment, I will be happy to answer them. (PERSONAL CONTACT INFORMATION WAS LISTED HERE)
B.2 Thesis Experiment Tester Instructions

B.2.1 Introduction Comments

1. Thank the subject for volunteering to do the experiment.

2. Ask the subject if the thesis experiment preparation handout was read. If not, read the handout to the subject.

3. Tell the subject “Only pen and pen/voice input will be used. There will be no keyboard available and menus can’t be used for this experiment.”

4. Tell the subject to move quickly through the tutorial sections of the experiment and not to rush through the diagram drawings.

5. Tell the subject that we are now ready to start the experiment.

B.2.2 Learning Graffiti

1. Tell the subject “You will now learn how to use the Graffiti character recognizer.”

2. Start the pen-only version of Picasso up. Maximize the main window and switch to text mode. Enlarge the drawing window to take up most of the screen. Leave some space at the bottom of the main window for Graffiti. Restore the Graffiti window and place it in an area of the screen that is not obstructed by the drawing window. Turn caps lock on in Graffiti.

3. Tell the subject “A pen tap just means to press the pen against the pen computer’s window and then release. Rewrite any characters that are recognized wrong. ‘X’, ‘?’, and ‘/’ will delete the previous character for now, but don’t worry about it. Don’t worry about capitalization at all for this experiment. Move through writing the characters quickly to conserve time.”
4. Give the subject the learning Graffiti subject instructions. Have the subject draw the Graffiti characters as given in their instructions.

5. Exit Picasso.

**B.2.3 Learning Picasso Basics**

1. Tell the subject “You will now learn how to use the basic functionality of Picasso.”

2. Start the pen-only version of Picasso up. Maximize the main window. Enlarge the drawing window to take up most of the screen.

3. Give the subject the learning Picasso basics subject instructions. Have the subject go through the learning basic Picasso instructions.

4. Tell the subject that there are no scroll bars in Picasso, so watch out for moving objects off the screen. Mention that the center handle of a selected shape is a resize handle. Tell the subject not to use the center handle.

5. Exit Picasso.

**B.2.4 Training the Speech Recognizer**

1. Tell the subject “You will now train the speech recognizer.”

2. Start the pen/voice version of Picasso up.

3. Select the Voice-Voice Filenames menu option and clear out the subject file field using Graffiti. Hit OK and exit out of Picasso.

4. Start the pen/voice version of Picasso up again.

5. Select the Voice-Train Words menu option.
6. Help the subject put the microphone headset on. The microphone should be at
the corner of the subject’s mouth and fairly close to the face.

7. Tell the subject “Hit the microphone button when you are ready to begin. If you make
a mistake, you can either repeat training for the current word or retrain the previous word.
If you have questions, hit the microphone button to turn the microphone off.”

8. Give the subject the training the speech recognizer subject instructions and the Graffity
reference card. Tell the subject to train the system as given in their instructions.

9. When the subject is done training, select the Voice-Save Subject File menu option and
save the training to a specific subject filename using Graffity.

10. Select the Voice-Voice Filenames menu option and enter the newly created subject
filename for the subject file using Graffity.

11. Exit Picasso.

B.2.5 Learning the Pen Version of Picasso

1. Tell the subject “You will now learn how to use the pen version of Picasso.”

2. Start the pen-only version of Picasso up. Maximize the main window and switch to
text mode. Enlarge the drawing window to take up most of the screen. Leave some space
at the bottom of the main window for Graffity. Restore the Graffity window and place it
in an area of the screen that is not obstructed by the drawing window. Turn caps lock on
in Graffity.

3. Give the subject the learning the pen version of Picasso subject instructions and the
pen modality instructions. Have the subject go through the learning the pen instructions.

4. Do this step if the subject is learning the pen editor first. Tell the subject that objects
will move and resize in grid increments rather than on the grid itself. Tell the subject that
starting lines inside an object and drawing slowly might cause a select rather than a drawn line.

5. Exit Picasso.

**B.2.6 Learning the Pen/Voice Version of Picasso**

1. Tell the subject “You will now learn how to use the pen and voice version of Picasso.”

2. Start the pen/voice version of Picasso up. Maximize the main window. Enlarge the drawing window to take up most of the screen. Leave some space at the bottom of the main window for the voice words dialog box. Select Voice-Display Words from the menu. Move the voice words dialog box to the bottom of the main window. Tap on the drawing window.

3. Tell the subject that when the computer beeps, it didn’t recognize the given word.

4. Give the subject the learning the pen/voice version of Picasso subject instructions and the pen/voice modality instructions. Have the subject go through the learning the pen/voice instructions.

5. Do this step if the subject is learning the pen editor first. Tell the subject that objects will move and resize in grid increments rather than on the grid itself. Tell the subject that starting lines inside an object and drawing slowly might cause a select rather than a drawn line.


**B.2.7 Drawing the Diagrams**

1. Give the subject the diagram and diagram instructions.

2. Have the subject read the diagram instructions.
3. If the diagram is a pen-only diagram, do this step. Start the pen-only version of Picasso up. Maximize the main window and switch to text mode. Enlarge the drawing window to take up most of the screen. Leave some space at the bottom of the main window for Graffiti. Restore the Graffiti window and place it in an area of the screen that is not obstructed by the drawing window.

4. If the diagram is a pen/voice diagram, do this step. Start the pen/voice version of Picasso up. Maximize the main window. Enlarge the drawing window to take up most of the screen. Leave some space at the bottom of the main window for the voice words dialog box. Select Voice-Display Words from the menu. Move the voice words dialog box to the bottom of the main window. Tap on the drawing window.

5. Ask if there are any questions about the diagram to be drawn.

6. Tell the subject to let the tester know when he/she is done.

7. Tell the subject to start. Start the stopwatch at this point.

8. While the subject draws the diagram, keep track of how many times the subject uses the speech recognizer for commands. Keep track of how many times every recognizer has a mis-recognition or non-recognition along with what input caused the error. Also, keep track of the number of questions, relevant to using the editor, asked from this point to the point of completion. Also, if the subject is having significant problems with a functionality of the editor, make a comment to the subject and count it as a question asked. All subject experiment information should be written on the subject’s testing sheets. If any bugs appear, pause the stopwatch and do whatever possible to fix them. Write down any useful comments about the subject, the editor, the input modalities, or the experiment.

9. When the subject indicates he/she is finished, look over the diagram. Point out anything that still needs to be done.

10. When the diagram looks to be complete, stop the stopwatch. Save the drawing window by selecting File-Save on the menu. Enter the filename with Graffiti.
11. Exit Picasso.

**B.2.8 Closing the Experiment**

1. Give the subject the questionnaire form and tell the subject to fill it out and get it back to the tester in the next few hours.

2. Tell the subject not to discuss the experiment with future subjects until they have done the experiment.

3. Thank the subject again.
B.3 Thesis Experiment Subject Training Instructions

B.3.1 Learning Graffiti

1. Tap the pen somewhere near the left border of the drawing window. A text cursor should appear.

2. Referring to the given Graffiti Reference Card, you will draw the symbol for A in the Graffiti window. Start the pen stroke where the dot is shown on the reference card. An ‘A’ should appear at the text cursor’s location. Continue writing the rest of the alphabet in order.

3. Tap the “abc” button in Graffiti. “123” should appear in its place. Write the digits ‘0’ - ‘9’ in order.

4. Tap the “123” button so that “abc” reappears. Write the space character followed by the backspace character.

5. Tap inside the Graffiti window. Follow this action by drawing the question mark character in the Graffiti window. Tap again inside the Graffiti window and draw the forward slash character ‘/’.

6. Repeat steps 1-5 again. Because of a bug in the Graffiti Demo program, the letters will appear in lowercase even though the “CAPS” button is on.

B.3.2 Learning Picasso Basics

1. Using a pen, draw a medium size rectangle somewhere in the drawing window. Notice that the shape recognizer transforms your drawing into a perfect shape.

2. Draw a medium size ellipse, triangle, diamond, and line somewhere in the drawing window.
3. Tap the pen inside the rectangle. The displayed handles show that the object is selected.

4. Press the pen down inside the rectangle (not on the handles) and drag the pen around the screen. This is how you move objects.

5. Press the pen down inside one of the rectangle’s handles and drag the pen around the screen. This is how you resize objects.

6. Tap the pen inside the rectangle. Tapping inside a selected object will deselect the object.

7. Select the rectangle and the ellipse. Move and resize the ellipse. Notice that both the selected objects move and resize simultaneously.

8. Tap anywhere in the drawing window that is outside of all the objects. Notice that this performs a deselect all objects action.

9. Tap anywhere in the drawing window that is outside of all the objects. Notice that tapping outside of all the objects, when no objects are selected, performs a select all action.

10. Double tap quickly on one of the objects. A text label dialog box will appear. This is how to label objects. Tap on cancel to exit the dialog box.

**B.3.3 Training the Speech Recognizer**

1. Anytime that a mistake is made, hit either the “Repeat Word” or “Previous Word” button as appropriate to retrain.

2. Letters will be displayed as two consecutive characters and pronounced as two consecutive letters. For example “TT” would be pronounced as “Tee-Tee”. When you want to input a T character in the editor, you must say “TT”.
3. Hit the “Microphone” button on the voice training dialog box.

4. You will need to repeat each word 4 times. Say “zero” for ‘0’. Notice the repetition count will increase.

5. Continue speaking the prompted words until the voice training dialog box closes. “ZZ” will be the last word to train.

**B.3.4 Learning the Pen Version of Picasso**

1. Quickly read through the pen modality instructions.

2. Draw a medium size rectangle somewhere on the screen.

3. Draw a medium size ellipse somewhere on the screen.

4. Tap the rectangle button. Notice that the last drawn shape (the ellipse) now is a rectangle.

5. Tap the ellipse button. Tap the triangle button. Finally, tap the diamond button. The change shape commands are a quick way to fix rectangles, ellipses, triangles, and diamonds that were mis-recognized by the shape recognizer.

6. Select the two objects. Tap the group selected objects button.

7. Tap outside of the objects for a deselect all. Tap inside one of the objects. Notice that both objects were selected as a group.

8. Tap the ungroup selected objects button. Tap inside one of the objects. Notice that only the tapped object was deselected because of the ungroup.

9. Tap the copy button followed by tapping the paste button twice. This demonstrates a copy-paste situation.