Comparing Usability through Computational Cognitive Modeling

IE team
11 Jan 2006

Process

Old Interface - SAMI
- Develop Java-based Prototype
- Develop GOMSL Model
- Integrated Execution of Java Prototype & GOMSL
- Output: Total Execution Time, Learning Analysis, Method Analysis

New Interface – Cookbook
- Develop Java-based Prototype
- Develop GOMSL Model
- Integrated Execution of Java Prototype & GOMSL
- Output: Total Execution Time, Learning Analysis, Method Analysis

Usability Comparison!!

Same Tasks
Developing a Prototype for the Barcoder Dialog

Original Barcoder Interface

Note: The Java-based prototype has similar functions to the original interface and provides for similar interactivity.

New Cookbook Barcoder Interface

Developing a Prototype for the Method Editor

Original SAMI Interface

Note: The Java-based prototype is an almost identical replicate of the original SAMI Method editor.

Java Prototype of SAMI Interface
Developing a Prototype for the New Method Editor

Note: The HTML prototype was used for usability testing with CELISCA personnel. The Java-based prototype was necessary for cognitive model testing.

Integration between Java device and GOMSL

EGLEAN

GOMSL Model - SAMI & Cookbook

Scenario event

xml

Control Event

JAVA

Device Representation

- Java Application
- SAMI & Cookbook interfaces

Output

- Work Sequence
- Total Execution Time
- Learning Analysis
- Workload Report
Time parameters for each operator in GOMSL

- Look_for_object_whose property is value...and_store_under <tag> : 1200msec.
- Keystroke key_name: 280msec.
- Type_in String: 280msec./character
- Click mouse_button: 200msec.
- Double_click: 400msec.
- Hold_down mouse_button: 100msec.
- Release mouse_button: 100msec
- Point_to target_object: 1100msec
- Think_of: 1200msec.

Developing the GOMSL Models

- Used “Task_item”(s) for handling all of the tasks in the original scenario script used for usability test with human subjects.
- To ensure plausibility and to resemble human trials, we used two time consuming “Think_of” operators for representing the time to read each task on in the printed scenario script.
- We applied same task strategies with Java-based prototypes of both interfaces (SAMI and Cookbook interfaces)
Sample GOMSL Model (for SAMI interface)

- TaskItem T38
  - Number is "38".
  - Type is Drag_into_Center.
  - Label is "Home".
  - Next is "39".
  - Target is "Center".

- TaskItem T39
  - Number is "39".
  - Type is Draw_Line.
  - Next is "End".
  - Target1 is "Biomek_Center".
  - Target2 is "Home2_Center".

Method for goal Manipulating SAMI
  - Step 1: Store "1" under current_task_number.
  - Step 2: Decide: If current_task_number is "End", Then Return_with_goal_accomplished.
  - Step 3: Accomplish: goal Read Scenario.
  - Step 4: Get_task_item_whose Number is current_task_number and_store_under current_task.
  - Step 5: Accomplish: Perform Unit_Task.
  - Step 6: Store Next of current_task under current_task_number.
  - Step 7: Goto 2.

Method for goal Read Scenario
  - Step 1: Think_of "Read a line on scenario sheet".
  - Step 2: Think_of "Read a line on scenario sheet".
  - Step 3: Return_with_goal_accomplished.

Method for goal Perform_Unit_Task
  - If Type of current_task is Drag_into_Center, then Accomplish: goal Drag Icon using Label of current_task.
  - If Type of current_task is Draw_Line, then Accomplish: goal Draw Line using Target1 of current_task.
  - If Type of current_task is Click_for_Popup, then Accomplish: goal Configure Select using Label of current_task.
  - If Type of current_task is Select_Drop_down_Menu, then Accomplish: goal Configure Select using Target of current_task.
  - Return_with_goal_accomplished.

Comparison of Old and New Method Editor Output

- Total Execution Time
  - Old SAMI Interface: 317.950 msec.
  - New Cookbook Interface: 365.050 msec.
- Learning Analysis

### SAMI

<table>
<thead>
<tr>
<th>Method Units</th>
<th>Number</th>
<th>Learned to Learn</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manipulating SAMI</td>
<td>8</td>
<td>0 / 8</td>
</tr>
<tr>
<td>Read Scenario</td>
<td>4</td>
<td>0 / 4</td>
</tr>
<tr>
<td>Perform Unit_Task</td>
<td>7</td>
<td>0 / 7</td>
</tr>
<tr>
<td>Clicking Object</td>
<td>5</td>
<td>0 / 5</td>
</tr>
<tr>
<td>Drag Icon</td>
<td>8</td>
<td>0 / 8</td>
</tr>
<tr>
<td>Draw Line</td>
<td>9</td>
<td>0 / 9</td>
</tr>
<tr>
<td>Configure Select</td>
<td>8</td>
<td>0 / 8</td>
</tr>
<tr>
<td>Typing String</td>
<td>6</td>
<td>0 / 6</td>
</tr>
<tr>
<td>Total</td>
<td>55</td>
<td>0 / 55</td>
</tr>
</tbody>
</table>

### Cookbook

<table>
<thead>
<tr>
<th>Method Units</th>
<th>Number</th>
<th>Learned to Learn</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manipulating Method_Edit</td>
<td>8</td>
<td>0 / 8</td>
</tr>
<tr>
<td>Read Scenario</td>
<td>4</td>
<td>0 / 4</td>
</tr>
<tr>
<td>Perform Unit_task</td>
<td>5</td>
<td>0 / 5</td>
</tr>
<tr>
<td>Typing String</td>
<td>6</td>
<td>0 / 6</td>
</tr>
<tr>
<td>Clicking Object</td>
<td>5</td>
<td>0 / 5</td>
</tr>
<tr>
<td>Select_Drop_down_Menu</td>
<td>7</td>
<td>0 / 7</td>
</tr>
<tr>
<td>Total</td>
<td>35</td>
<td>0 / 35</td>
</tr>
</tbody>
</table>

Note: Typing the essay in the Cookbook interface took 189 msec. This is over half the total execution time.

New interface requires far fewer steps in methods to be learned.
### Comparison of Outputs (Continued)

- **Workload Reports**

<table>
<thead>
<tr>
<th>Freq</th>
<th>Subtotal</th>
<th>Avg. Time</th>
<th>% of Total</th>
<th>Method for goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>39</td>
<td>158.100</td>
<td>4.054</td>
<td>49.72</td>
<td>Perform Unit task</td>
</tr>
<tr>
<td>39</td>
<td>99.450</td>
<td>2.550</td>
<td>31.28</td>
<td>Read Scenario</td>
</tr>
<tr>
<td>11</td>
<td>29.950</td>
<td>2.723</td>
<td>9.42</td>
<td>Clicking Object</td>
</tr>
<tr>
<td>4</td>
<td>16.800</td>
<td>4.200</td>
<td>5.28</td>
<td>Configure Select</td>
</tr>
<tr>
<td>11</td>
<td>38.900</td>
<td>3.536</td>
<td>12.23</td>
<td>Drag Icon</td>
</tr>
<tr>
<td>10</td>
<td>55.000</td>
<td>5.500</td>
<td>17.3</td>
<td>Draw Line</td>
</tr>
<tr>
<td>3</td>
<td>13.550</td>
<td>4.517</td>
<td>4.26</td>
<td>Typing String</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Freq</th>
<th>Subtotal</th>
<th>Avg. Time</th>
<th>% of Total</th>
<th>Method for goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td>266.700</td>
<td>11.112</td>
<td>73.06</td>
<td>Perform Unit task</td>
</tr>
<tr>
<td>24</td>
<td>61.200</td>
<td>2.550</td>
<td>16.76</td>
<td>Read Scenario</td>
</tr>
<tr>
<td>18</td>
<td>50.700</td>
<td>2.817</td>
<td>13.89</td>
<td>Clicking Object</td>
</tr>
<tr>
<td>3</td>
<td>12.900</td>
<td>4.300</td>
<td>3.53</td>
<td>Select Drop_down_Menu</td>
</tr>
<tr>
<td>3</td>
<td>200.700</td>
<td>66.900</td>
<td>54.98</td>
<td>Typing String</td>
</tr>
</tbody>
</table>

### User Performance Analysis

- **Performance Analysis**

  - Develop Java-based Prototype
  - Incorporate time and error collection methods
  - Expert and Non-Biologists Recruited

  **Results**
  - Subtask Times
  - Total Execution Time
  - Total Number of Errors

*Note: Remote usability analysis used for Expert Biologists.*
User Performance Times for Barcoder A/C Dialog

<table>
<thead>
<tr>
<th></th>
<th>CELSCLA (ExpertOperator)</th>
<th>NCSU (NonBioOperator)</th>
<th>GOMS</th>
<th>SAMI</th>
<th>Cookbook</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>Std</td>
<td>Mean</td>
<td>Std</td>
<td>Mean</td>
</tr>
<tr>
<td>Determine Side</td>
<td>1.95</td>
<td>0.00</td>
<td>3.59</td>
<td>1.79</td>
<td>1</td>
</tr>
<tr>
<td>Determine label ty</td>
<td>1.55</td>
<td>0.00</td>
<td>3.93</td>
<td>2.36</td>
<td>1</td>
</tr>
<tr>
<td>Determine label content</td>
<td>6.45</td>
<td>0.00</td>
<td>12.86</td>
<td>6.50</td>
<td>1</td>
</tr>
<tr>
<td>Total time on task</td>
<td>14.00</td>
<td>0.00</td>
<td>24.00</td>
<td>10.43</td>
<td>1</td>
</tr>
</tbody>
</table>

**Validation of interface use times.**
- Model output not sign. different from actual human data.
- Expert performance across interfaces.
- No significant difference.
- Experts adapt quickly to new interface.

**Conclusion**

**GOMS Development:**
- The Java-based prototype has similar functions and interactivity as the original application.
- Results are highly generalizable to actual task.

**Usability Analysis:**
- Operators can quickly achieve proficiency with new barcoder interface.
- Experts can adapt to new interface quickly to achieve performance comparable to use of existing software requiring high training.
- Methods to be learned in new interface are fewer, but typing time is a large part of the total task time.