Lecture 5: Designing smart systems

Using statistical methods to anticipate user needs and actions with Bayesian strategies
Overview of the course

• Theory driven approaches to HCI
• Design of visual displays
• Goal-oriented interaction
• Designing efficient systems
• **Designing smart systems**
• Designing meaningful systems (guest lecturer)
• Evaluating interactive system designs
• Designing complex systems
Uniform text entry
Information gain per key press

\[ h(x_i) = \log \frac{1}{p(x_i)} \]
The q?
Information gain per key press

“As you are aware, E is the most common letter in the English alphabet, and it predominates to so marked an extent that even in a short sentence one would expect to find it most often”

The Adventure of the Dancing Men,
Sir Arthur Conan Doyle
<table>
<thead>
<tr>
<th>Letter</th>
<th>Count</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>445.2</td>
<td>12.49%</td>
</tr>
<tr>
<td>T</td>
<td>330.5</td>
<td>9.28%</td>
</tr>
<tr>
<td>A</td>
<td>286.5</td>
<td>8.04%</td>
</tr>
<tr>
<td>O</td>
<td>272.3</td>
<td>7.64%</td>
</tr>
<tr>
<td>I</td>
<td>269.7</td>
<td>7.57%</td>
</tr>
<tr>
<td>N</td>
<td>257.8</td>
<td>7.23%</td>
</tr>
<tr>
<td>S</td>
<td>232.1</td>
<td>6.51%</td>
</tr>
<tr>
<td>R</td>
<td>223.8</td>
<td>6.28%</td>
</tr>
<tr>
<td>H</td>
<td>180.1</td>
<td>5.05%</td>
</tr>
<tr>
<td>L</td>
<td>145.0</td>
<td>4.07%</td>
</tr>
<tr>
<td>D</td>
<td>136.0</td>
<td>3.82%</td>
</tr>
<tr>
<td>C</td>
<td>119.2</td>
<td>3.34%</td>
</tr>
<tr>
<td>U</td>
<td>97.3</td>
<td>2.73%</td>
</tr>
<tr>
<td>M</td>
<td>89.5</td>
<td>2.51%</td>
</tr>
<tr>
<td>F</td>
<td>85.6</td>
<td>2.40%</td>
</tr>
<tr>
<td>P</td>
<td>76.1</td>
<td>2.14%</td>
</tr>
<tr>
<td>G</td>
<td>66.6</td>
<td>1.87%</td>
</tr>
<tr>
<td>W</td>
<td>59.7</td>
<td>1.68%</td>
</tr>
<tr>
<td>Y</td>
<td>59.3</td>
<td>1.66%</td>
</tr>
<tr>
<td>B</td>
<td>52.9</td>
<td>1.48%</td>
</tr>
<tr>
<td>V</td>
<td>37.5</td>
<td>1.05%</td>
</tr>
<tr>
<td>K</td>
<td>19.3</td>
<td>0.54%</td>
</tr>
<tr>
<td>X</td>
<td>8.4</td>
<td>0.23%</td>
</tr>
<tr>
<td>J</td>
<td>5.7</td>
<td>0.16%</td>
</tr>
<tr>
<td>Q</td>
<td>4.3</td>
<td>0.12%</td>
</tr>
<tr>
<td>Z</td>
<td>3.2</td>
<td>0.09%</td>
</tr>
</tbody>
</table>
Hacking Fitt’s Law: “semantic pointing”

Simple application of Fitts Law

What’s wrong with this?
Increasing the depth of the language allows for a further separation...
Building a system based on relative frequencies

Dasher
(https://www.youtube.com/watch?v=FLalNywdHxU)
Some lessons from Dasher

• Turning an information theoretic model into a user interface requires a lot of creativity
  => Interaction with Machine learning course

• In many cases simple models (nGrams + smoothing) are as - or more - effective than complex ones (neural nets)

• Supporting even famous software, useful for marginalised groups is hard

(The Financial Times, February 2002)
Artificial languages
Artificial languages

new Future.
Ordering code completion suggestions

A simple scheme for predicting code completions:

```java
void main() {
    Stopwatch sw = new Stopwatch();
    sw. // <--- What goes here?
}
```

```java
elapsed
elapsedMicroseconds
elapsedMilliseconds
elapsedTicks
Frequency
hashCode
isRunning
noSuchMethod
Reset
runtimeType
Start
Stop
toString
```
Ordering code completion suggestions

We calculate:

\[
P(\text{completion} = \text{"reset"} \mid \text{context} = \text{"void main() \{ Stopwatch sw = new Stopwatch(); sw."}}) \\
P(\text{completion} = \text{"start"} \mid \text{context} = \text{"void main() \{ Stopwatch sw = new Stopwatch(); sw."}})
\]

... And the usual:

\[
P(A \mid B) = \frac{P(B \mid A) P(A)}{P(B)},
\]
Ordering code completion suggestions

\[ P(\text{completion} = ? \mid \text{context} = "\ldots") \propto P(\text{context} = "\ldots" \mid \text{completion} = ?) P(\text{completion} = ?) \]

Feature vector

<table>
<thead>
<tr>
<th>Completion c</th>
<th>Count of seen completions</th>
<th>P(completion)</th>
</tr>
</thead>
<tbody>
<tr>
<td>start</td>
<td>10</td>
<td>0.5</td>
</tr>
<tr>
<td>reset</td>
<td>5</td>
<td>0.25</td>
</tr>
<tr>
<td>elapsed</td>
<td>5</td>
<td>0.25</td>
</tr>
</tbody>
</table>
# Ordering code completion suggestions

\[ P(\text{completion} = c \mid \text{context} = \ldots) \propto P(\text{context} = \ldots \mid \text{completion} = c) \ P(\text{completion} = c) \]

| Completion c | P(completion==c | context) | Order |
|--------------|------------------|-------|
| start        | 0.9 * 0.5 = 0.45 | 0     |
| reset        | 0.4 * 0.25 = 0.1 | 1     |
| elapsed      | 0.2 * 0.25 = 0.06| 2     |

<table>
<thead>
<tr>
<th>Completion c</th>
<th>Feature</th>
<th>Feature value</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>start</td>
<td>“First-Use”</td>
<td>true</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>false</td>
<td>1</td>
</tr>
<tr>
<td>reset</td>
<td>“First-Use”</td>
<td>true</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>false</td>
<td>3</td>
</tr>
<tr>
<td>elapsed</td>
<td>“First-Use”</td>
<td>true</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>false</td>
<td>4</td>
</tr>
</tbody>
</table>
Some progress in information efficient IDEs
Building user interfaces
(from Per Ola’s material)
Building user interfaces
(from Per Ola’s material)

- Entry and error rate
- Learning curve, familiarity and immediate efficacy
- Form factor, presentation, time and comfort
- User engagement
- Visual attention and cognitive resources
- Privacy
- Single vs Multi-character entry
- Specification vs Navigation
- One/Two handed
- Task integration
- Robustness
- Device independence
- Computational demands
- Manufacturing and support cost
- Localisation
- Market acceptance
Building user interfaces: Solution principles
(from Per Ola’s material)

• From closed to open-loop
  • Avoid the need for a visual feedback loop
• Continuous novice-to-expert transition
  • Avoid explicit learning
• Path dependency
  • Avoid redesign the interaction layer
• Flexibility
  • Enable users to compose and edit in a variety of styles without explicit mode switching
• Efficiency
  • Let users’ creativity by the bottle-neck
Airbus autopilot

https://docs.flybywiresim.com/pilots-corner/a32nx-briefing/flight-deck/glareshield/fcu/
Airbus autopilot

Approach Speed

Page 1 of OPS DATA explains how the speed correction is applied. The approach speed increment should be added to the VREF (shown as VLS on the PERF APPR page) for Flap FULL. In addition, provided that the resultant VAPP does not exceed VREF +20kt, one third of the headwind component should be added to this figure.

The resultant speed should be inserted, if possible, in the VAPP field on the PERF APPR page and bugged on the standby airspeed indicator. If the situation requires the speed to be Selected, rather than Managed, then the speed calculated above can be set on the FCU.

Insertion of the calculated VAPP on the PERF APPROACH page will ensure that if Managed speed is available, the correct approach speed will be flown. Also the benefits of GS mini will be available, even though the aircraft is landing in an abnormal configuration.

For example, a DUAL ADR FAULT requires a direct law landing flown in Config 3, using a VAPP of VLS Flap FULL plus 10kt, plus one third of the head wind component, subject to the 20kt limit described above.
Tesla autopilot

Design considerations

- Situational awareness
- Interaction style for configuring future behaviour
- Degree of knowledge of the behaviour of the system
The programming analogy challenge 2023: adding abstraction to a paint program