A Case Study on Model-Based Design - I

Objective

- In the previous lectures, we learned about the idea of models and model-based design
  - We discussed about different types of models used in HCI
  - We learned in details about four models – KLM, (CMN)GOMS, Fitts' law and Hick-Hyman law

Objective

- We have discussed about the broad principles of model-based design
- In this and the following lecture, we shall see a specific case study on model-based design, namely design of virtual keyboards, to understand the idea better

Virtual Keyboards

- Before going into the design, let us first try to understand virtual keyboard (VK)
- We know what a physical keyboard is
  - The input device through which you can input characters
- Although physical keyboards are ubiquitous and familiar, sometimes it is not available or feasible

Virtual Keyboards

- Suppose you want to input characters in a mobile device (e.g., your mobile phone or iPad)
  - Physical keyboards make the system bulky and reduces mobility
- Sometimes the users' may not have the requisite motor control to operate physical keyboards
  - For example, persons with cerebral palsy, paraplegia etc.

Virtual Keyboards

- In such scenario, VKs are useful
  - A VK is an on-screen representation of the physical keyboard (see the below image which shows text input in iPad with a VK)

VK Design Challenge

- The iPad example in the previous slide show a QWERTY layout (i.e., key arrangement)
  - That’s because the typing is two-hand and QWERTY layout is suitable for two-hand typing
- However, in many cases, VK is used with single-hand typing (particularly for small devices where one hand holds the device)

VK Design Challenge

- Since QWERTY layout is good for two-hand typing, we have to find out alternative "efficient" layout
  - Efficiency, in the context of keyboards in general and VK in particular, is measured in terms of character entry speed (characters/sec or CPS, words/min or WPM etc)

VK Design Challenge

- Thus, what we want is a VK layout for single hand typing that allows the user to input characters with high speed and accuracy
- Mathematically, for a N character keyboard, we have to determine the best among N! possible key arrangements

VK Design Challenge

- Thus, it is a typical "search" problem
  - We want to search for a solution in a search space of size N!
  - Note the "huge" size of the search space (for example, if N = 26 letters of English alphabet + 10 numerals = 36, the search space size is 36!)

What We Can Do

- We can apply the standard design life cycle
- Drawbacks
  - We can not check all the alternatives in the search space (that will in fact take millions of years!)
- If the designer is experienced, he can chose a small subset from the search space based on intuition

What We Can Do

- The alternatives in the subset can be put through the standard design life cycle for comparison
  - However, empirical comparison still requires great time and effort
- Alternatively, we can use model-based approach to compare alternatives

GOMS Analysis

- We can compare the designs in the subset using a GOMS analysis (also called CTA or cognitive task analysis)
- In order to do so, we first need to identify one or a set of “representative tasks”

GOMS Analysis

- What is a task here?
  - To input a series (string) of characters with the VK
- Remember, we should have a representative task
  - That means, the string of characters that we chose should represent the language characteristics

GOMS Analysis

- How to characterize a language?
- There are many ways
  – One simple approach is to consider unigram character distribution, which refers to the frequency of occurrence of characters in any arbitrary sample of the language (text)

GOMS Analysis

- How to characterize a language?
- There are many ways
  - Bigram distribution, which refers to the frequency of occurrence of character pairs or bigrams in any arbitrary sample, is another popular way to characterize a language

GOMS Analysis

- In order to perform GOMS analysis, we need to have character string(s) having language characteristics (say, the unigram distribution of characters in the string(s) match(es) to that of the language)
  - How to determine such string(s)?
- We can use a language corpus for the purpose

Corpus

- Corpus (of a language) refers to a collection of texts sampled from different categories (genres)
  - Stories, prose, poem, technical articles, newspaper reports, mails …
- It is assumed that a corpus represents the language (by capturing its idiosyncrasies through proper sampling)

Corpus

- However, corpus development is not trivial (requires great care to be truly representative)
- The good news is, already developed corpora are available for many languages (e.g., British National Corpus or BNC for English)
  - We can make use of those

Corpus-based Approach

- How to use a corpus to extract representative text?
  - Get hold of a corpus
  - Extract a representative text through some statistical means (for example, cross-entropy based similarity measure)

Cross-Entropy Based Similarity Measure

- Let X be a random variable which can take any character as its value
- Further, let P be the probability distribution function of X [i.e., $P(x_i) = P(X = x_i)$]
- We can calculate the "entropy", a statistical measure, of P in the following way

$$H(P) = -\sum_i P(x_i)\log_2 P(x_i)$$

Cross-Entropy Based Similarity Measure

- Now, suppose there are two distributions, P and M
- We can calculate another statistical measure, called “cross-entropy”, of the two distributions

$$H(P, M) = -\sum_i P(x_i) \log_2 M(x_i)$$

Cross-Entropy Based Similarity Measure

- The cross-entropy measure can be used to determine similarity of the two distributions
  - Closer H(P,M) is to H(P), the better approximation M is of P (i.e., M is similar to P)
- We can use this idea to extract representative text from a corpus

Cross-Entropy Based Similarity Measure

- Let P denotes the unigram probability distribution of the language
  - This can be determined from the corpus. Simply calculate the character frequencies in the corpus. Since the corpus is assumed to represent the language, the character frequencies obtained from the corpus can be taken as representative of the language
  - Calculate H(P)

Cross-Entropy Based Similarity Measure

- Take random samples of texts from the corpus and determine the unigram character distribution of the sample text, which is M
- Next, calculate H(P,M)
- The sample text for which H(P,M) is closest to H(P) will be our representative text

Problem with GOMS-based CTA

- Thus, we can perform GOMS analysis
- However, there is a problem
  - The text is usually large (typically >100 characters to make it *reasonably* representative), which makes it tedious to construct GOMS model

Problem with GOMS-based CTA

- We need some other approach, which is not task-based, to address the design challenge
  - Task-based approaches are typically tedious and sometimes infeasible to perform
- In the next lecture, we shall discuss one such approach, which is based on the Fitts’ law and the Hick-Hyman law