Command Languages

- One of earliest computer IO metaphors after cables/toggle switches/meters
- User must
  - recall notation
  - initiate actions
- Unlike other UI approaches before and after, added functionality doesn’t modify UI appearance
- Possibility for excess, baroque complexity
  - !?str?:s/ls/rmdir/:p
  - Does the language have too much functionality?
  - Or does the user have too little training?
Command Languages

- Still relevant because of
  - Prevalence of textual programming languages
  - “Little languages”, “Domain-specific languages”
    - Awk, HTML, CSS, but also…
  - Google
    - \( \sin 89 \) degrees
    - Volume of a cylinder with radius 8 m and height 2 ft
    - 2.64 USD per gallon in INR per litre
    - 0x3a4 + 0xfe3 in roman numerals
    - \( \sin x + \frac{1}{3} \sin 3x + \frac{1}{5} \sin 5x + \frac{1}{7} \sin 7x \)
- “Need for speed” in interactive systems
  - Text editing, with hands already on kbd (avoid using mouse)
  - Keyboard shortcuts
  - Regular expressions: E.g., “mv a*D*.c ~/proj/src”

Organization

- Single command
  - \(^f \ ^b \ ^d \ \text{del} \ \text{delete} \ldots\)
- Command plus args
  - Prefix, postfix, infix
- Command plus args (with options)
  - Optional trailing args
  - Use of option indicator (e.g., “-”)
  - Use of keyword args in arbitrary order (e.g., “<html lang="en-US">”)

Structure

- Consistent argument order
  - strcpy, strcat,…
    - dest=arg1, src=arg2
  - bcopy
    - dest=arg2, src=arg1
- Consistent, hierarchical naming
  \[
  \begin{array}{ll}
  L & RRA \\
  ST & RAR \\
  \text{vs.} & RAA \\
  & RRR
  \end{array}
  \]

Hierarchy and Congruence

J. Carroll 1982

- Hierarchical structure
  - E.g., “Verb object qualifier”
- Congruent (symmetrical) names
  - Meaningful opposites
- Experiment using paper-and-pencil task for small cmd set
  - (congruent, noncongruent) \times
    - (hierarchical, nonhierarchical) cmd sets
- Users
  - liked / performed best with both congruent forms
  - had lowest error rates on hierarchical congruent
- Hierarchical / Congruent
  - Move robot forward
  - Move robot backward
- Nonhierarchical / Congruent
  - Advance
  - Retreat
- Hierarchical / Noncongruent
  - Move robot forward
  - Change robot backward
- Nonhierarchical / Noncongruent
  - Go
  - Back
Interactive Documentation/Completion

- Addresses problem of recall

Assorted command line shells

Menu

- Set of displayed choices from which a user can select
  - Minimizes
    - Training
    - Memorization
    - Syntax errors
  - Good for novices, infrequent users
  - Some overlap with command languages if selection done by keyboard, but no syntax to remember
    - Is it a menu if list of choices is…
      - Post-it'ed on display?
      - Printed on kbd?
      - Displayed on computer, but not complete?

An early example of a menu created by labeling function keys
Types of Matching

- Exact / identity
- Class inclusion / categorical
- Equivalence / fuzzy

Exact Matching

- Alphabetic order > random order
  - Roughly twice as fast on small to medium-size menus
  - Best to be at top of alphabetic list
Exact Matching

Card 1982
- Presented participants with menu of 18 cmds, in three arrangements; they needed to mouse-select a specific target; average time after 43 trials:

<table>
<thead>
<tr>
<th>Alphabetical</th>
<th>Random</th>
<th>Categorical</th>
</tr>
</thead>
<tbody>
<tr>
<td>.81 sec</td>
<td>3.23 secs</td>
<td>1.28 secs</td>
</tr>
</tbody>
</table>

- After > 800 trials, search time is faster, but no significant difference across all conditions; participants have learned exact cmd location.

Exact Matching

- But, will user really know the target?
  - Months, days, states will work, but,…
    - Postal state abbreviations?
    - Editing commands?
      - Conventional names help!
    - Domain-specific commands?
Choosing a Menu Organization

Menu Tree Breadth vs. Depth
T. Landauer & D. Nachbar 85

- Task: Search huge ordered tree of either ints (internal nodes are numerically ordered ranges) or words (internal nodes are alphabetically ordered ranges)
- 4096 leaf items
- Varied depth/breadth: 2–16 items/level arranged 2×12 through 16×3
- Measured selection times: 23.4–12.5 secs: Breadth faster than depth!
- $T = c + k \log_2 b$, where
  - $T$ is time for selection within a level
  - $c$ and $k$ are constants ($k$ decreases with practice)
  - $b$ is breadth at that level
- $D = \log_b N$, where
  - $D$ is depth
  - $N$ is total number of leaf items
- Therefore, total time = $DT = \log_b N (c + k \log_2 b) = c (\log_b N) + k (\log_2 N)$
  - Breadth faster than depth!
  - Choices are progressively slower up to penultimate level (harder category match)
  - Choice at last level is relatively fast (exact match)

Hick-Hyman Law: Time to choose among $b$ equally probable choices is proportional to $\log_2 b$

Note: Assuming no need for exhaustive scan!
Sometimes expressed in terms of $(b + 1)$ to account for additional option of not making a choice

$\log_a b \log_b c = \log_a c$
Menu Tree Breadth vs. Depth

- Fewer levels $\rightarrow$ faster
  - but
- More items in a level $\rightarrow$
  - more screen space
  - or
  - need to use scrolling (requiring more time)

Menu Tree Breadth vs. Depth

- Breadth faster than depth
- Still need to avoid getting lost
  - Cascade
    - Maintains context
    - Quick return to top
    - Relatively easy to back up one level at a time
    - Dexterity issues
      - Can address with delay
Menu Design

- Group items meaningfully
  - Logically similar (cohesion)
    - Categorical organization advantages are lost with poor categorization
  - Cover all possibilities (complete)
  - No overlap (partition)
  - Users should be familiar with item meanings

- Phrasing
  - Familiar
  - Consistent
  - Distinctive
  - Concise
  - Keyword to the left

Split Menus

- Menus divided into sections (often two) with more frequent items in the top section
- Can be more efficient than conventional alphabetic menus [Sears & Shneiderman 94]
  - Short first section (e.g., ≤ 4 items)
  - Sections ordered the same way
Static, Adaptive, and Adaptable Menus L. Findlater & J. McGrenere, CHI 2004

- Compared three types of split menus with fixed split sizes (four on top)
  - Static (top items are the most frequent prior to study)
  - Adaptive (top items chosen dynamically by algorithms during study based on frequency/recency)
  - Adaptable (top items chosen by user prior to timed study block)

- Within-subject, counterbalanced for order
- Static faster than adaptive
- Adaptable faster than adaptive, except when adaptable used first (was slower then)
- Adaptable not slower than static except when adaptable used first (was slower then)

  Why? Top item list initially empty in adaptable menus and participants who used adaptable menu first were less likely to customize. (Were asked to customize without seeing the advantages of customization.)

Subjective: Majority of participants preferred adaptable and thought they were most efficient with adaptable menus.
**Menu Map**

- Visual representation of graph structure of a system’s menus
- D. Parton et al. 1985 experiment
  - Use of menu maps for 12 mins better than practicing for 12 mins, when followed by 10 mins working with 3×3 menu
  - “Big picture” of a menu map has advantages for learning over “peephole” experience of real navigation with a menu

**Site Map**

- Representation of web site structure
  - Often hand curated, showing only the top level

See [http://www.nngroup.com/articles/site-map-usability](http://www.nngroup.com/articles/site-map-usability)
Site Map

- Representation of web site structure
  - Often hand curated, showing only the top level

  Mostly superfluous in this case, courtesy of this “mega menu”

  See http://www.nngroup.com/articles/site-map-usability

“Mega menu” displayed by hovering over “SHOWS” (Note different categorization of items)

Site Map Variation: Showing Link Structure and User Location

- Can be
  - Created automatically
  - Included on all pages

http://www.nngroup.com/articles/site-map-usability

http://www.bricklin.com/history/intro.htm