

Dialog Design 1

Command languages and WIMP

The topic :
How does a user interact with the interface?

Dialog Style

1. Command languages
2. WIMP - Window, Icon, Menu, Pointer
3. Direct manipulation
4. Speech/Natural language
5. Gesture, pen, VR

Dialog Style

1. Command Languages

- Earliest UI interaction paradigms
- Examples
 - MS-DOS shell
 - UNIX shell
 - dBase
 - GPSS
- CL Attributes
 - Work primarily by recall, not recognition
 - Heavy memory load
 - Little or nothing is visible so...
 - Poor choice for novices but...
 - Advantages for experts ?



CL Advantages

- Advantages for experts
 - Speed, conciseness
 - %ls (hard to beat)
 - Can express actions beyond a limited set
 - Flags, piping one command to another
 - Repetition, extensibility
 - Scripting, macros
 - Easier implementation, less overhead
 - Power
 - Abstraction, wild cards

CL Danger

- With added power, comes added responsibility and danger
 - UNIX
 - % rm -r *
 - Deletes every file that you have, and you can't get them back

CL Reflection

- Command languages are often maligned (for good reason)
- But increased functionality can win out over bad UI (e.g., UNIX)
 - Try to get both
 - Avoid excess functionality (comes at cost)

CL Design Goals

- Consistency
- Good naming and abbreviations
- Doing your homework in design can help alleviate some of the negatives

Consistency

- Provide a consistent syntax
 - In general: Have options and arguments expressed the same way everywhere
 - UNIX fails here because commands were developed by lots of different people at different organizations
 - No guidelines provided

Order

- English: **SVO** subject verb object
"you" assumed on computer
- **CL: S** assumed (you)
 - Is VO or OV better?
 - % delete file
 - or
 - % file delete
- **V dO iO** vs. **V iO dO**
 - % print file calvin Which is better?
 - % lpr -Pcalvin file

Syntax

- Pick a consistent syntax strategy
 - Simple command list
 - e.g, vi, minimize keystrokes
 - Commands plus arguments
 - realistic, can provide keyword parameters
 - % cp from=foo to=bar
 - Commands plus options plus arguments
 - what you usually see

Terminology

- Keep terminology consistent
 - Same concept expressed with same options
 - Useful to provide symmetric (congruent) pairings
 - forward/backward
 - next/prev
 - control/meta

Example :

- vi text editor
 - w - forward word
 - b - backward word
- Wouldn't 'f' be better for forward?
 - 'f' already used
- How about 'fw' and 'bw'?
 - Extra keystrokes

Ordering

- Keep ordering consistent
 - VO seems to be the most natural
 - Typically need to pick where options go
- Example
 - % ln -s file1 file2 (I can never remember)
 - Think of % cp file1 file2

Names and Abbreviations

- Specificity versus Generality
 - General words
 - More familiar, easier to accept
 - Specific (typically better)
 - More descriptive, meaningful, distinctive
 - (Nonsense does surprisingly well in small set)

Abbreviations

- Abbrevs. allow for faster actions
 - Expert performance begins to be dominated by motor times such as # of keystrokes
 - Not good idea for novices
 - (Allow but don't require)

Picking Good Abbreviations

- Strategies
 - Simple truncation (works best, but conflicts)
 - Vowel drop plus truncation (avoid conflicts)
 - First and last letters
 - First letters of words in a phrase
 - Standard abbrev from other contexts
 - qty, rm, bldg
 - Phonics
 - xqt

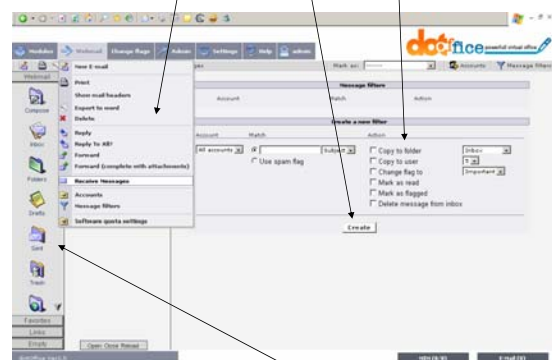
Abbreviation Guidelines

- Use single primary rule (with single fallback for conflicts)
- Use fallback as little as possible
- Mark use of fallback in documentation
- Let user know primary and secondary rules
- Truncation is good but generates conflicts
- Fixed length is better than variable length
- Don't use abbrevs. in system output

Dialog Style

2. WIMP

- Focus: Menus, Buttons, Forms



- icon

Menus

- **Key advantages:**
 - 1 keystroke or mouse operation vs. many
 - No memorization of commands
 - Limited input set
- **Many different types**
 - pop-up
 - pull-down
 - radio buttons
 - pie buttons
 - hierarchies

Menu Items

- **Organization strategies**
 - Create groups of logically similar items
 - Cover all possibilities
 - Ensure that items are non-overlapping
 - Keep wording concise, understandable
- **Bad Example**
 - Travel web page links:
 - Flight page
 - 3 Best Itineraries
 - Flights & Prices
 - Timetables
 - Fares
 - Which do you choose for reservations?

Presentation Sequence

- How does Mac, Netscape, etc, do it?
- Use natural if available
 - Time
 - e.g. Breakfast, Lunch, Dinner
 - Numeric ordering
 - e.g. Point sizes for font
- Choices
 - Alphabetical
 - Group related items
 - Frequently used first
 - Most important first

Presentation Sequence

- **User studies**
 - Novices: alpha > functional > random
 - Experts: categorization
- How would you do it in general?
- One possible methodology (first->last)
 - Natural order (if exists)
 - Frequency of use
 - Order of use
 - Categorical
 - Alphabetical
- Don't change dynamically!