

# Dialog Styles: Command Languages, WIMP, & Direct Manipulation

John Stasko

Spring 2007

This material has been developed by Georgia Tech HCI faculty, and continues to evolve. Contributors include Gregory Abowd, Al Badre, Jim Foley, Elizabeth Mynatt, Jeff Pierce, Colin Potts, Chris Shaw, John Stasko, and Bruce Walker. Permission is granted to use with acknowledgement for non-profit purposes. Last revision: January 2007.

## Dialog Styles

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



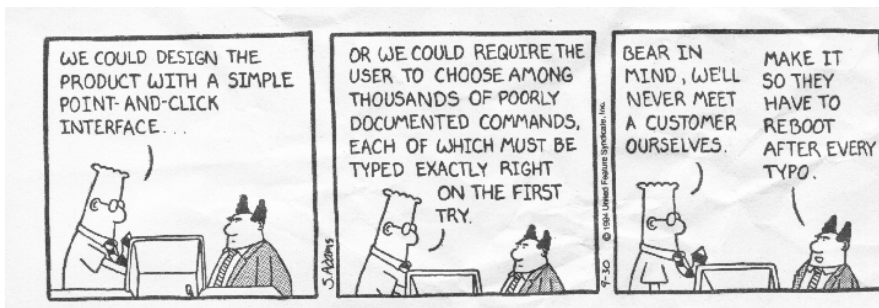
## Agenda

- Command languages
  - Advantages, disadvantages
  - Design guidelines
- WIMP
  - Advantages, disadvantages
  - Design guidelines
- Direct manipulation
  - Definition
  - Advantages & disadvantages
  - Another characterization



## Dialog Design

- How does a user interact with the interface?

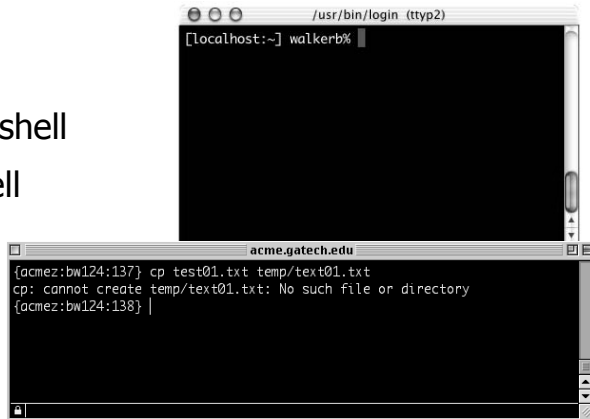


## Command Languages

- Earliest UI interaction paradigms

- Examples

- MS-DOS shell
- UNIX shell
- dBase
- GPSS



## Issues

- CL are easily maligned...

## Unix Shell CL Potential Disadvantages

- Learning takes a long time
- Hard to remember command names
- Some command names don't make sense, so have to memorize
- No in-progress feedback - how much longer?
- System state is invisible, and have to know which commands to use to get which information
- Hard to make sense of outputs, such as with *ls* - no headings, no code interpretations
- No "look"
- No warning if bad things are going to happen
- No universal Undo; to reverse a command, have to know the inverse command (create directory, delete directory)
- Have to use *man* command to find desired command
- How to get help?
- Because commands are short, typos can lead to incorrect command
- Inconsistent flag meanings
- Inconsistent parameter orders
- Have to type a lot - touch typing needed



## CL Attributes

- Work primarily by recall, not recognition
- Heavy memory load
- Little or nothing is visible so...
  
- Poor choice for novices
- But all is not bad...



## CL Attributes

- 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 Dangers

- 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

– % lpr -Pcalvin file

Which is better?



## 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



## Abbreviations Matter...

**DILBERT**

By Scott Adams



## Dialog Design

- 1. Command language
- 2. WIMP ←
- 3. Direct manipulation
- 4. Pen, gesture
- 5. Speech, audio



## WIMP

- Focus: Windows, Menus, Buttons, Forms
- Predominant interface paradigm now  
(with some direct manipulation added)
- Advantages:
  - ?



## Window Pros

- Facilitate multi-tasking, which many people do
- Maps well onto overlapping sheets of paper on our desks, so is a familiar concept
- Makes computer usage easier for many people



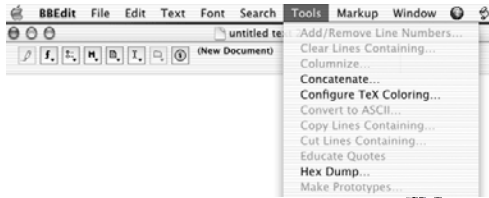
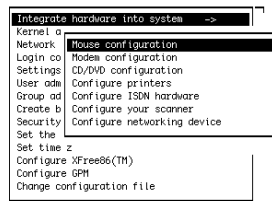
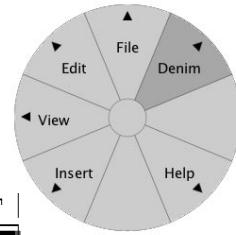
## Window Cons

- Can make concentrating on a single task hard (that incoming mail....)
- An extension of the cluttered desk :)
- May be unnecessary for dedicated-use environments that run a single application

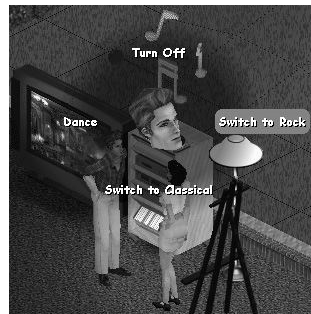
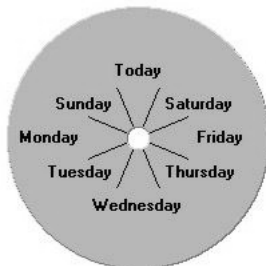


# Menus

- Many different types
  - pop-up
  - pull-down
  - radio buttons
  - pie buttons
  - hierarchies



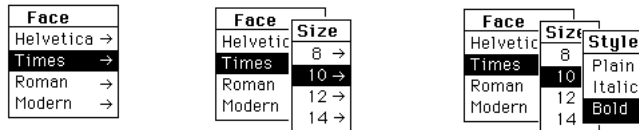
# Pie Menus



From Sim City



## Pop-up Hierarchical



## Menu Pros

- Key advantages:
  - 1 keystroke or mouse operation vs. many
  - No memorization of commands
  - Limited input set





## Menu Cons

- Less direct user control - have to find correct menu / menu item
- Not so readily extensible
- Slower than keyboarding for experienced users, at least without accelerators



## Menu Items

- Organization strategies
  - Create groups of logically similar items
  - Cover all possibilities
  - Ensure that items are non-overlapping
  - Keep wording concise, understandable



## Presentation Sequence

- How does Mac, IE, etc, do it?
- Use natural if available
  - Time
    - e.g. Breakfast, Lunch, Dinner
  - Numeric ordering
    - e.g. Point sizes for font



## Presentation Sequence

- 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?



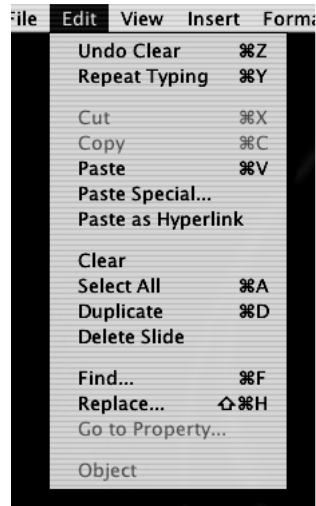
## Presentation Sequence

- One possible methodology (first->last)
  - Natural order (if exists)
  - Frequency of use
  - Order of use
  - Categorical
  - Alphabetical
- Don't change dynamically!



## A Good Menu Example

- Logical grouping
- Visual separation of groups
- Disabled items "grayed out"
- Shortcuts shown
- ... indicates leads to dialogue
- Go forth and find some bad examples!



## Bad Example

- Travel web page links:
  - Flight page
  - 3 Best Itineraries
  - Flights & Prices
  - Timetables
  - Fares



- Which do you choose for reservations?



## Dialog Design

- 1. Command language
- 2. WIMP
- 3. Direct manipulation ←
- 4. Pen, gesture
- 5. Speech, audio



## Definition

- What is direct manipulation?



## Direct Manipulation

- 1) Continuous visibility of the objects and actions of interest
- 2) Rapid, reversible, incremental actions whose effect is immediately noticeable
- 3) Replacement of command language syntax by direct manipulation of object of interest (physical actions, buttons, etc.)

Shneiderman '82



## Direct Manipulation

- Examples
  - WYSIWYG editors and word processors
  - VISICALC - 1<sup>st</sup> electronic spreadsheet
  - CAD
  - Desktop metaphor
  - Video games



## DM Essence

- Representation of reality that can be manipulated
- The user is able to apply intellect directly to the task
- The tool itself seems to disappear



## Direct Manipulation

- Advantages
- Disadvantages



## DM Advantages

- Easier to learn & remember, particularly for novices
- Direct WYSIWYG
- Flexible, easily reversible actions helps reduce anxiety in users



## DM Advantages

- Provides context & instant visual feedback so user can tell if objectives are being achieved
- Exploits human use of visual spatial cues
- Limits types of errors that can be made





## DM Problems

- Screen space intensive (info not all that dense)
- Need to learn meaning of components of visual representation
- Visual representation may be misleading
- Mouse ops may be slower than typing
- Not self-explanatory (no prompts)



## DM Problems

- Not good at
  - Repetition
  - History keeping (harder)
  - Certain tasks (Change all italics to bold)
  - Abstract elements (variables)
  - Macros harder



## What is DM?

- UNIX?
- Word?
- Emacs?
- PowerPoint?



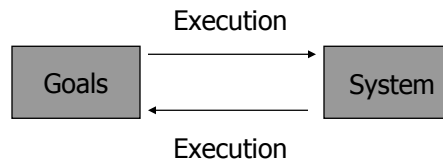
## More Psychological View

- What is directness? (not always done well)
- Related to two things:
  - Distance
  - Engagement



## Distance

- Two gaps or “gulfs” between user’s goals and system image
- Directness partly depends on the distance between these two gulfs
  - Gulf of execution
  - Gulf of evaluation



## Gulfs

- Gulf of execution
  - Distance between user’s goals and means of achieving them in system
    - Does the system allow the user to do what they want?
- Gulf of evaluation
  - Amount of effort person must expend to interpret system state and judge if intention was achieved
    - Can user perceive if progressing favorably?



## Directness and Distance

- Two types
  - Semantic - Relation between what user want to express and what is available in interface
    - Can I say what I want (concisely)?
  - Articulatory - Relation between meanings of expressions and their physical form(s)
    - Is the way to perform an action expected and clear (appropriate)?



## Engagement

- Feeling that you are directly manipulating the objects of interest
- Promoted by
  - Unobtrusive interface
  - Minimizing gulfs of execution and evaluation
  - Appropriately responsive system

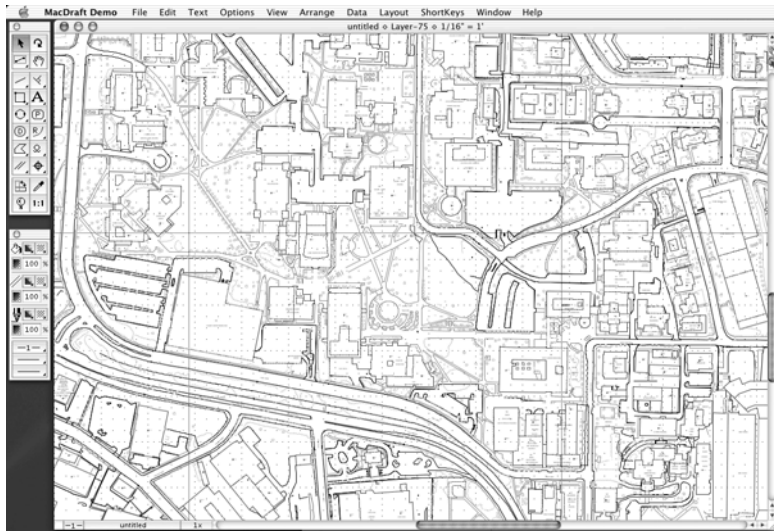


Ultimately...

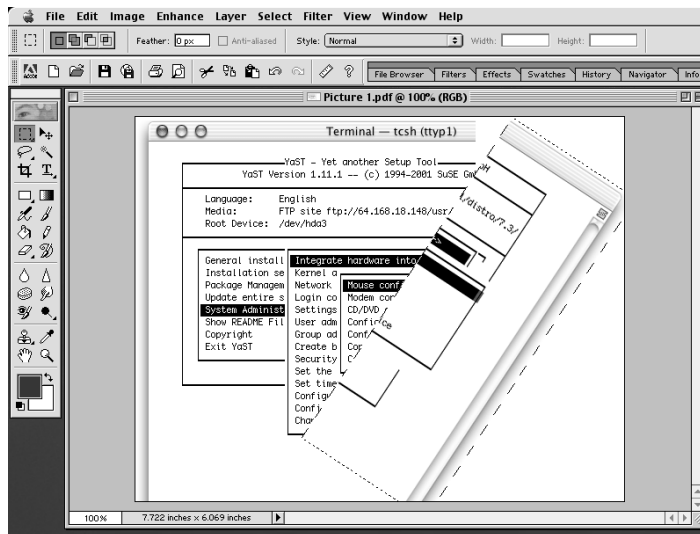
- In end, must characterize direct manipulation by feeling of directness and illusion of manipulating objects at hand



## Example: CAD



## Example: Photoshop



## Reminder

- P2 due Friday
- Show designs, discuss their strengths and weaknesses w.r.t. requirements
- Questions?



## Upcoming

- Dialog
  - Speech & natural language
  - Pen & Gesture
- Predictive Models

