Human-Computer Interaction

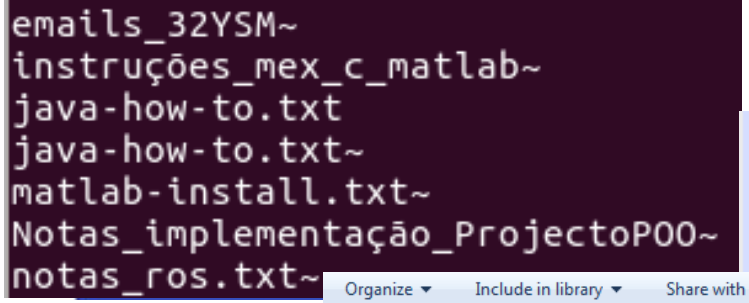
2024/2025

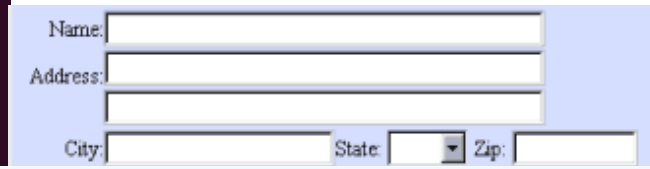
Lecture 11

Interaction styles (cont)

universidade de aveiro
deti departamento de electrónica, telecomunicações e informática

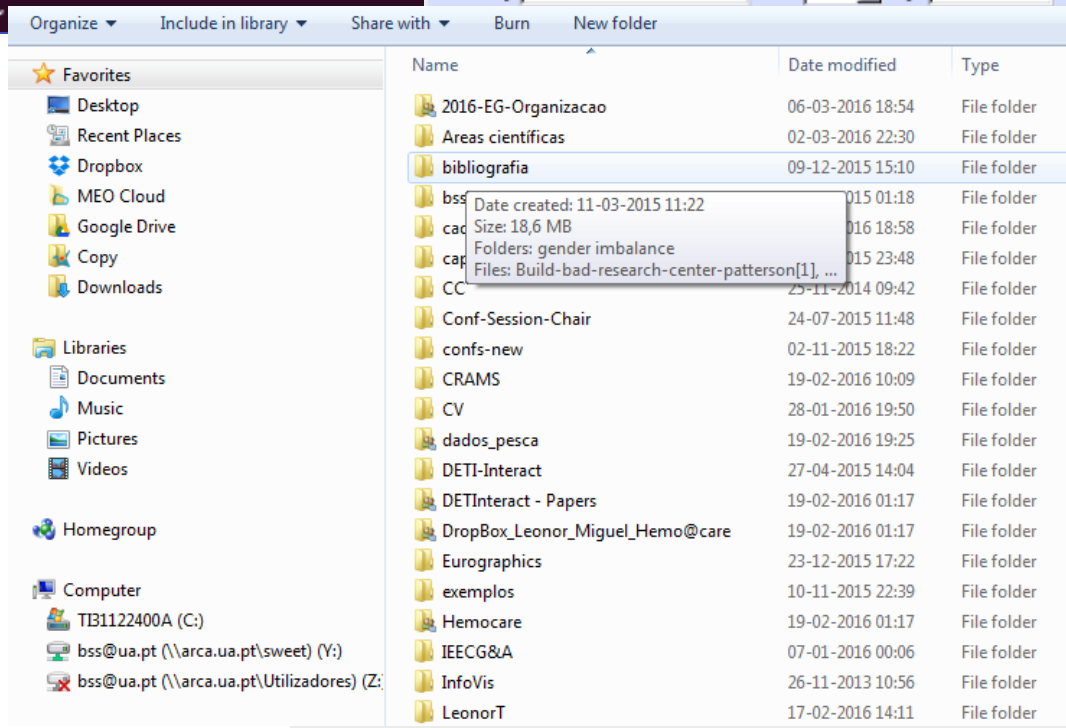
Interaction/ Dialog styles



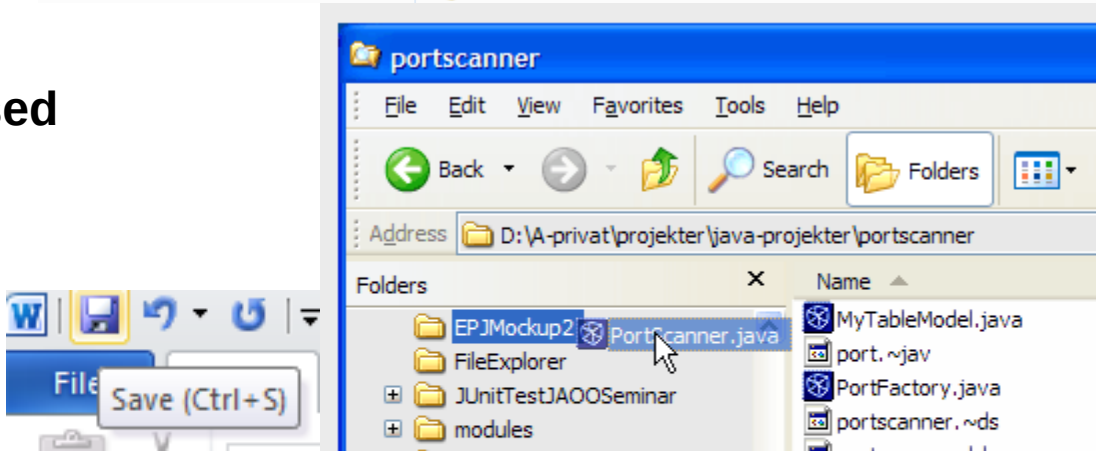


A possible classification:

- Menus
- Fill-in-forms
- Direct manipulation
- Command languages
- Natural languages
- Etc.



Often two or more styles are used simultaneously; why?



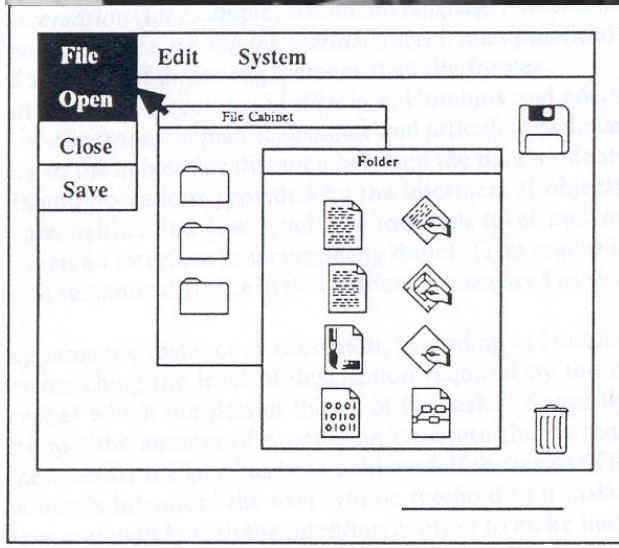
Direct Manipulation

Direct manipulation

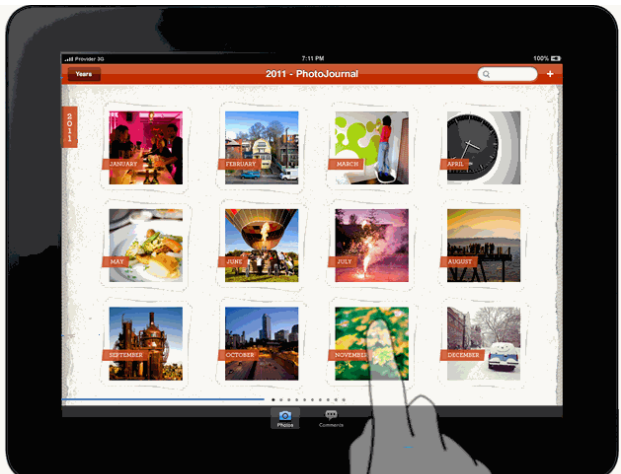
(name coined by Shneiderman, 1982)

- Can be traced down to Sketchpad

[Sketchpad, by Dr. Ivan Sutherland with comments by Alan Kay]



- **Actions are performed directly on visual representations of the objects**



[Interaction Styles | The Glossary of Human Computer Interaction]

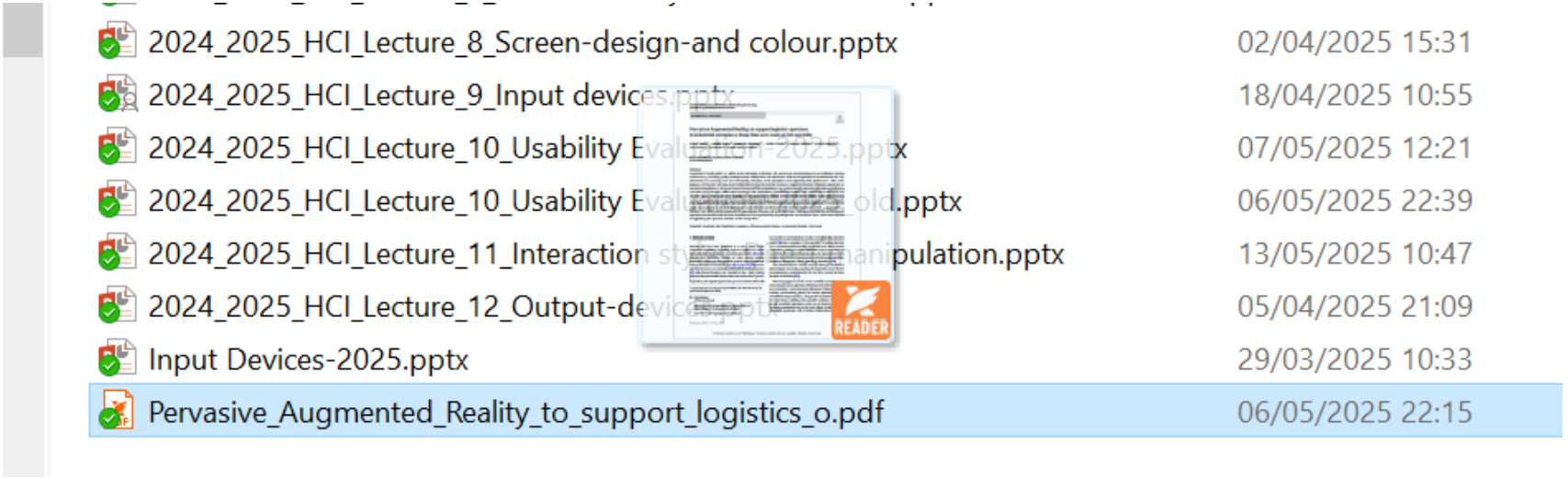
Direct manipulation

- Is characterized by:

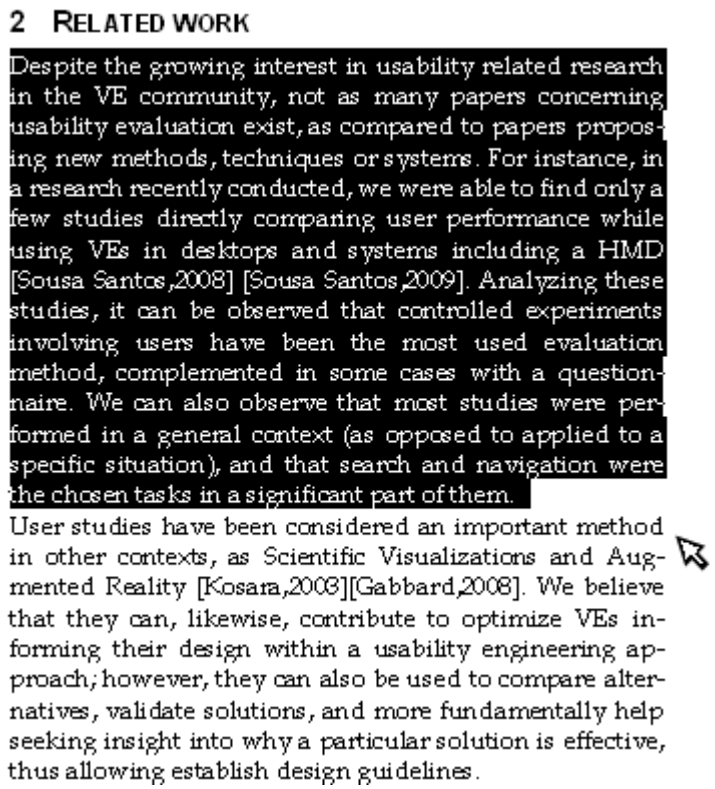
1- Continuous representation of objects

2- Physical actions instead of command languages

3- Fast, incremental, reversible actions with visible results



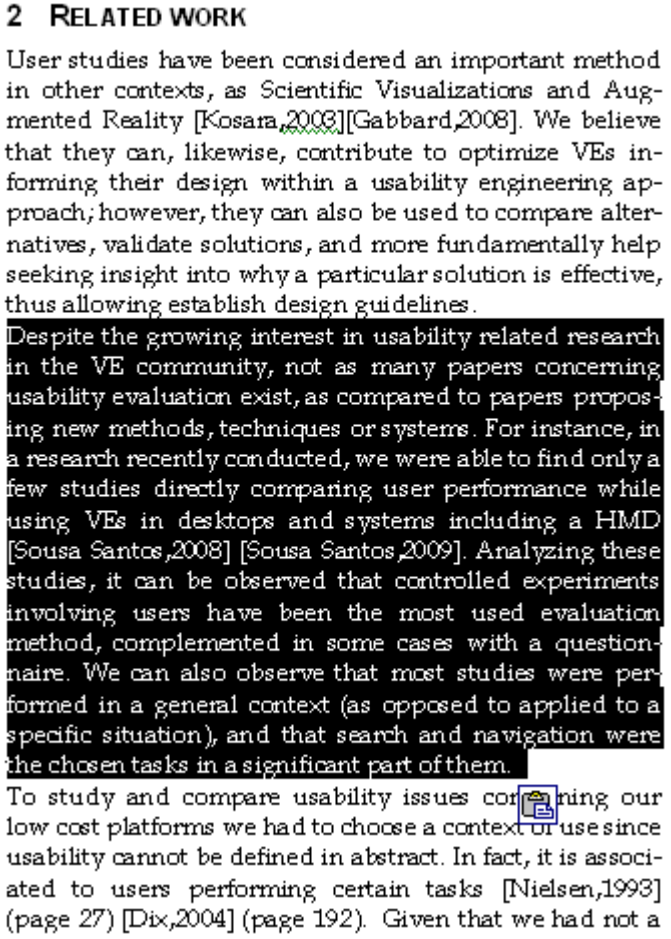
Direct manipulation **does not necessarily imply icons**; however, in most situations they are involved



2 RELATED WORK

Despite the growing interest in usability related research in the VE community, not as many papers concerning usability evaluation exist, as compared to papers proposing new methods, techniques or systems. For instance, in a research recently conducted, we were able to find only a few studies directly comparing user performance while using VEs in desktops and systems including a HMD [Sousa Santos,2008] [Sousa Santos,2009]. Analyzing these studies, it can be observed that controlled experiments involving users have been the most used evaluation method, complemented in some cases with a questionnaire. We can also observe that most studies were performed in a general context (as opposed to applied to a specific situation), and that search and navigation were the chosen tasks in a significant part of them.

User studies have been considered an important method in other contexts, as Scientific Visualizations and Augmented Reality [Kosara,2003][Gabbard,2008]. We believe that they can, likewise, contribute to optimize VEs informing their design within a usability engineering approach; however, they can also be used to compare alternatives, validate solutions, and more fundamentally help seeking insight into why a particular solution is effective, thus allowing establish design guidelines.



2 RELATED WORK

User studies have been considered an important method in other contexts, as Scientific Visualizations and Augmented Reality [Kosara,2003][Gabbard,2008]. We believe that they can, likewise, contribute to optimize VEs informing their design within a usability engineering approach; however, they can also be used to compare alternatives, validate solutions, and more fundamentally help seeking insight into why a particular solution is effective, thus allowing establish design guidelines.

Despite the growing interest in usability related research in the VE community, not as many papers concerning usability evaluation exist, as compared to papers proposing new methods, techniques or systems. For instance, in a research recently conducted, we were able to find only a few studies directly comparing user performance while using VEs in desktops and systems including a HMD [Sousa Santos,2008] [Sousa Santos,2009]. Analyzing these studies, it can be observed that controlled experiments involving users have been the most used evaluation method, complemented in some cases with a questionnaire. We can also observe that most studies were performed in a general context (as opposed to applied to a specific situation), and that search and navigation were the chosen tasks in a significant part of them.

To study and compare usability issues concerning our low cost platforms we had to choose a context of use since usability cannot be defined in abstract. In fact, it is associated to users performing certain tasks [Nielsen,1993] (page 27) [Dix,2004] (page 192). Given that we had not a

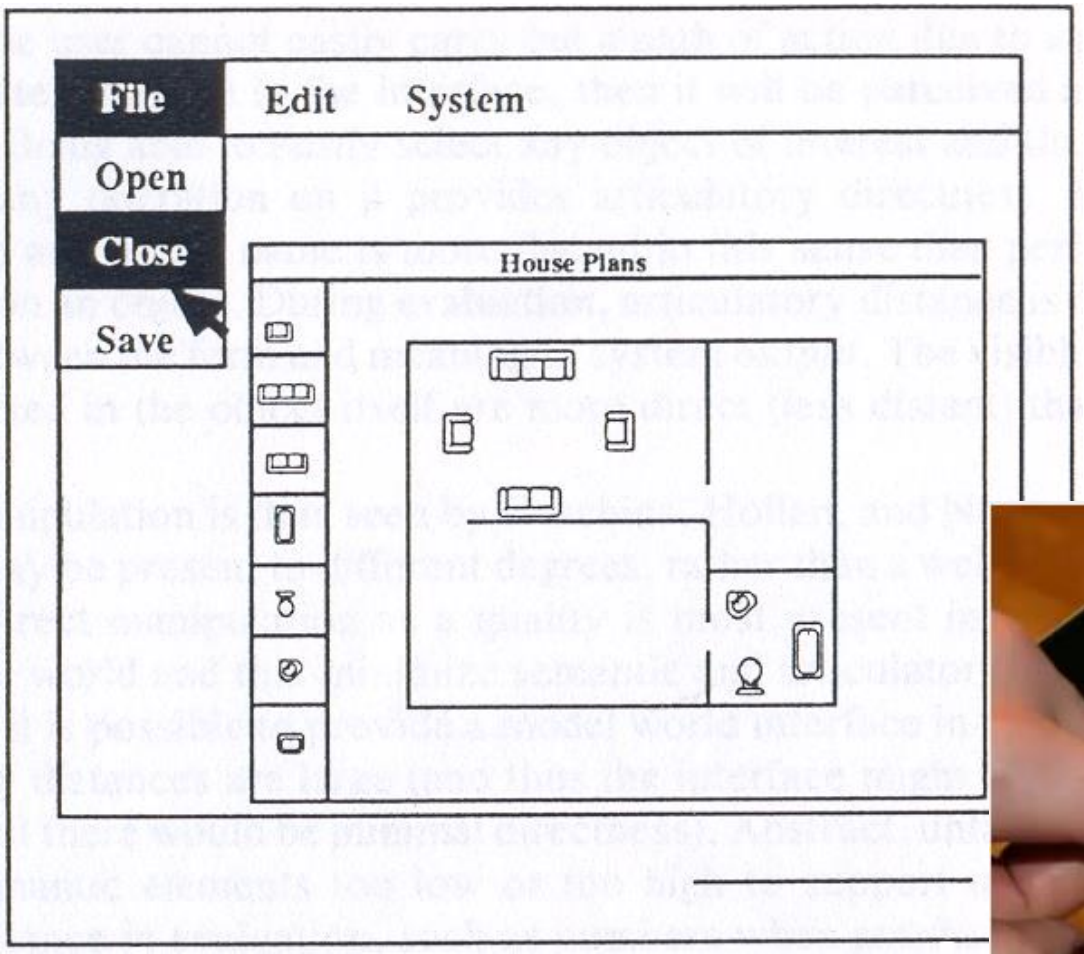
Example: When a section of a text is selected and dragged elsewhere icons are not used, yet an action is performed on a visual representation of an object (text section)

Direct manipulation does not necessarily imply icons; however, in most situations they are involved

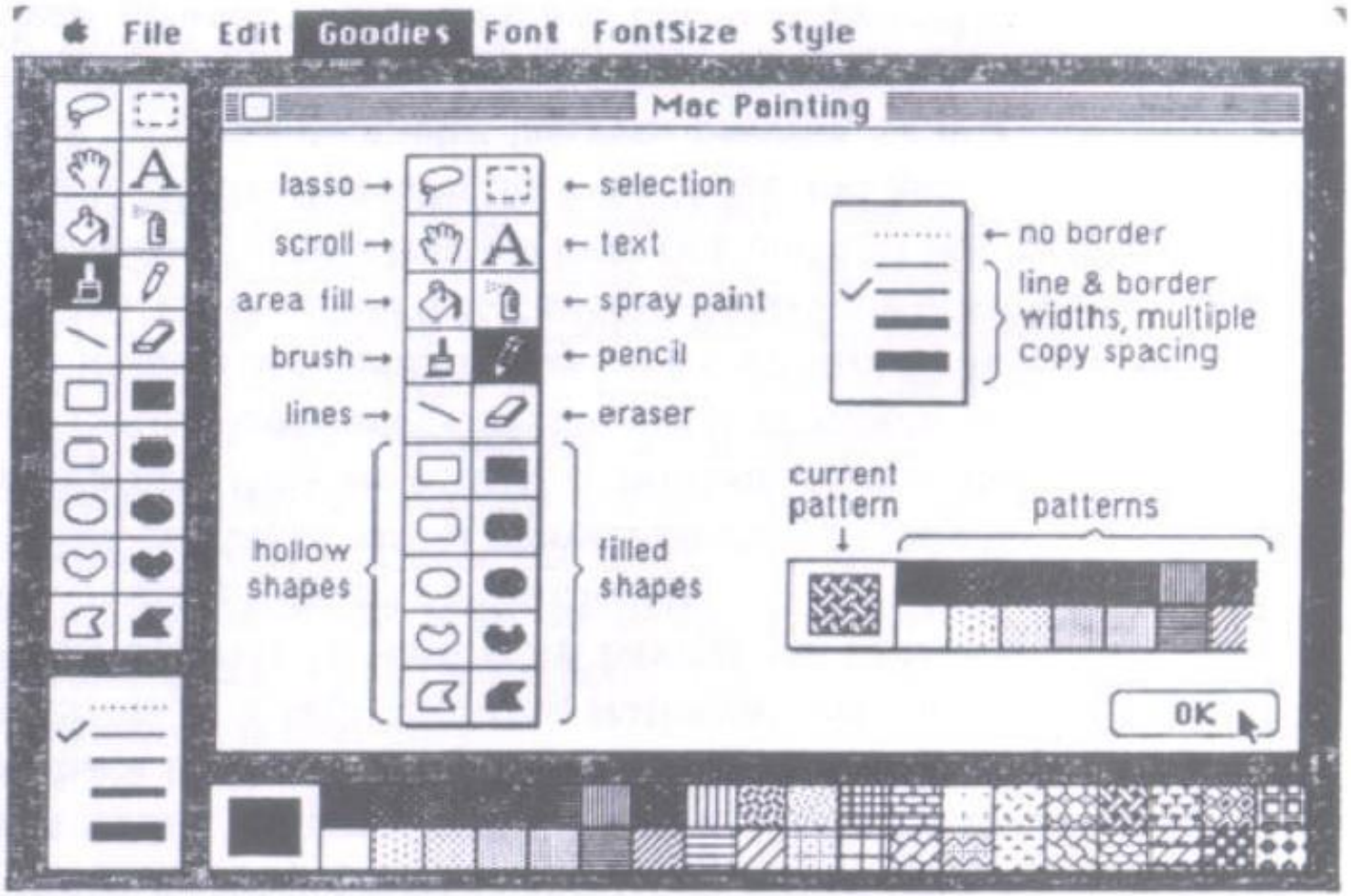
Another example:
On a mobile phone you can pinch out/in to zoom into an image or to zoom out

Direct Manipulation: Definition

Some applications are more adequate to use direct manipulation:

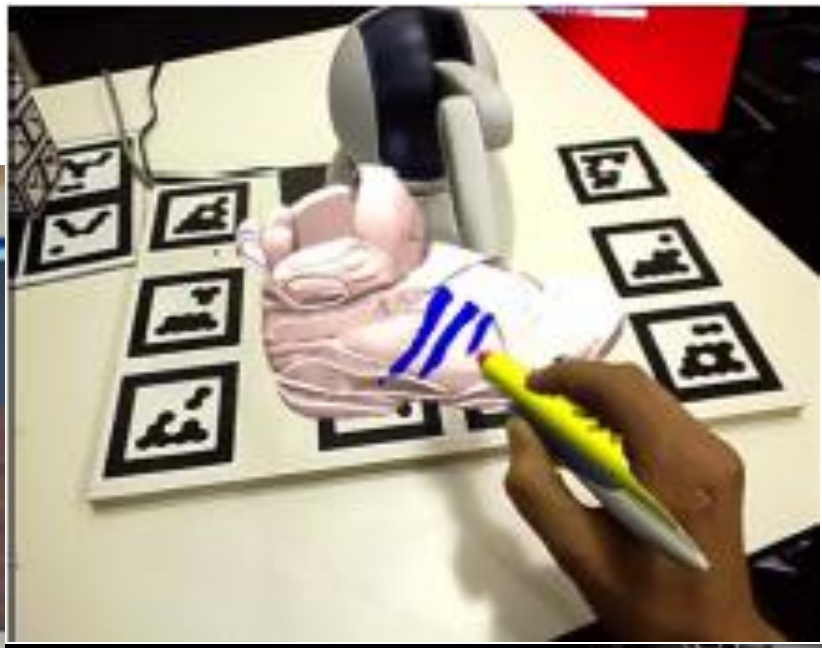


One of the earliest commercially available UI using Direct Manipulation (MacPaint)



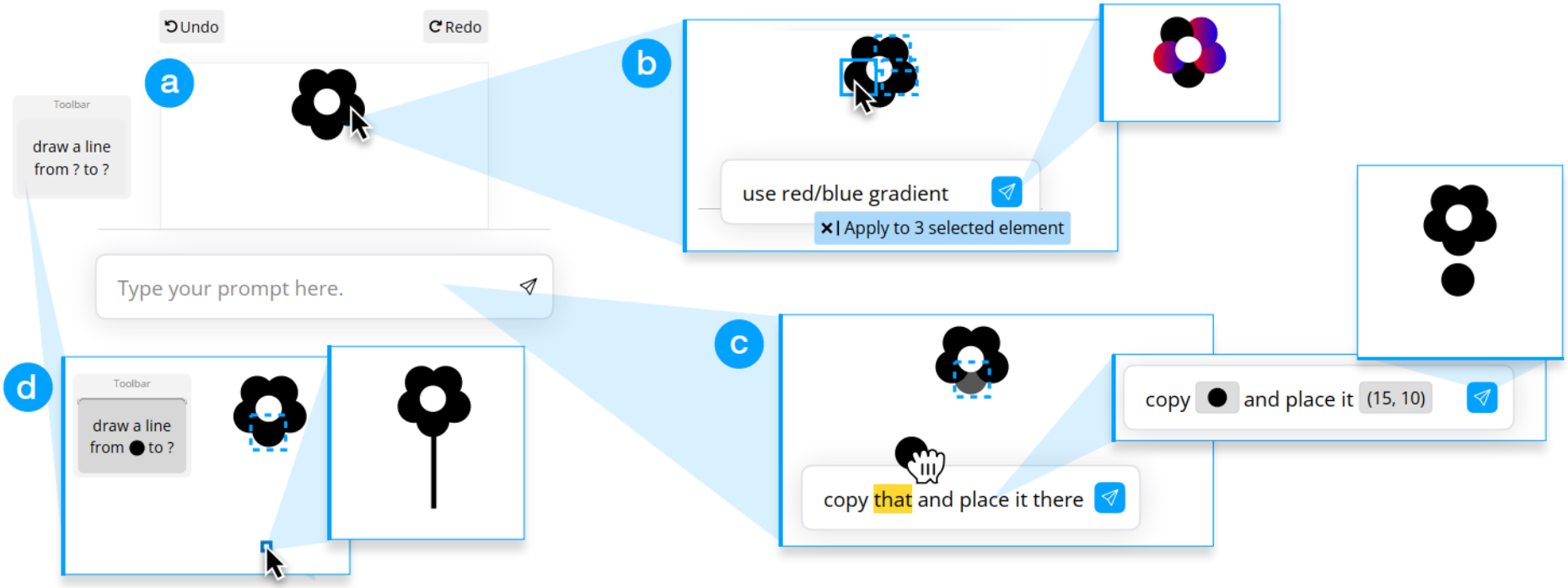
Virtual and augmented reality

Take direct manipulation to another level



[Object Selection and Manipulation in VR Headsets: Research Challenges, Solutions, and Success Measurements | ACM Computing Surveys]

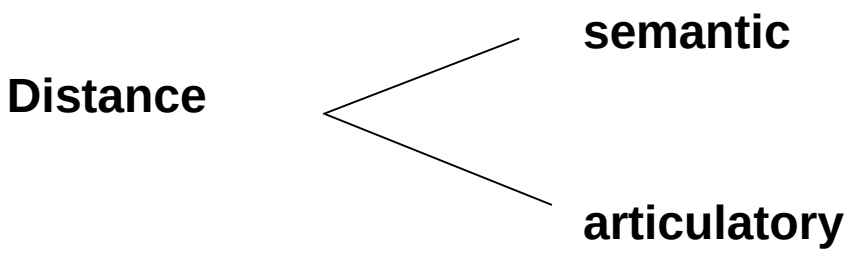
Most recent usage: with LMMs…



DirectGPT used on a vector image to demonstrate direct manipulation principles for LLMs: (a) continuous representation of the objects of interest; physical actions to (b) localize the effect of prompts and (c) refer to objects; (d) reusable prompts in a toolbar of commands; and reversible operations through undo and redo features.

DirectGPT: A Direct Manipulation Interface to Interact with Large Language Models | Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems

- It does not exist a “pure” direct manipulation User Interface (UI)
- Direct manipulation is a **quality which may be present in different degrees**
- According to Hutchins, Hollan e Norman (1986) a UI has the following aspects:



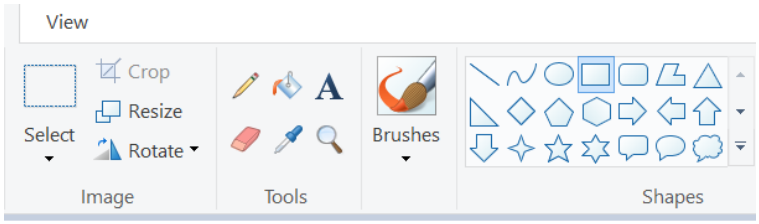
Semantic and articulatory distance

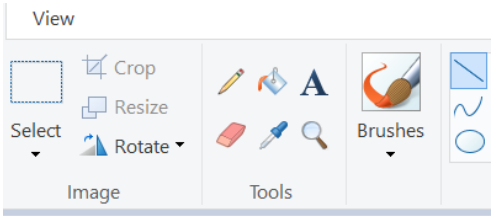
- Semantic Distance – **subjective distance** between the **user's goal** and **interface semantics**

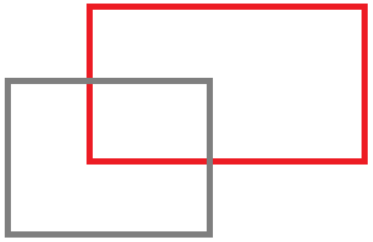
- Articulatory distance – distance between the **meaning of the actions** and their **physical form**

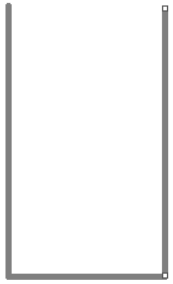
Semantic Distance

If the objects and actions do not support the users' goals, semantic distance is high









If the user wants to draw rectangles this application has a smaller semantic distance

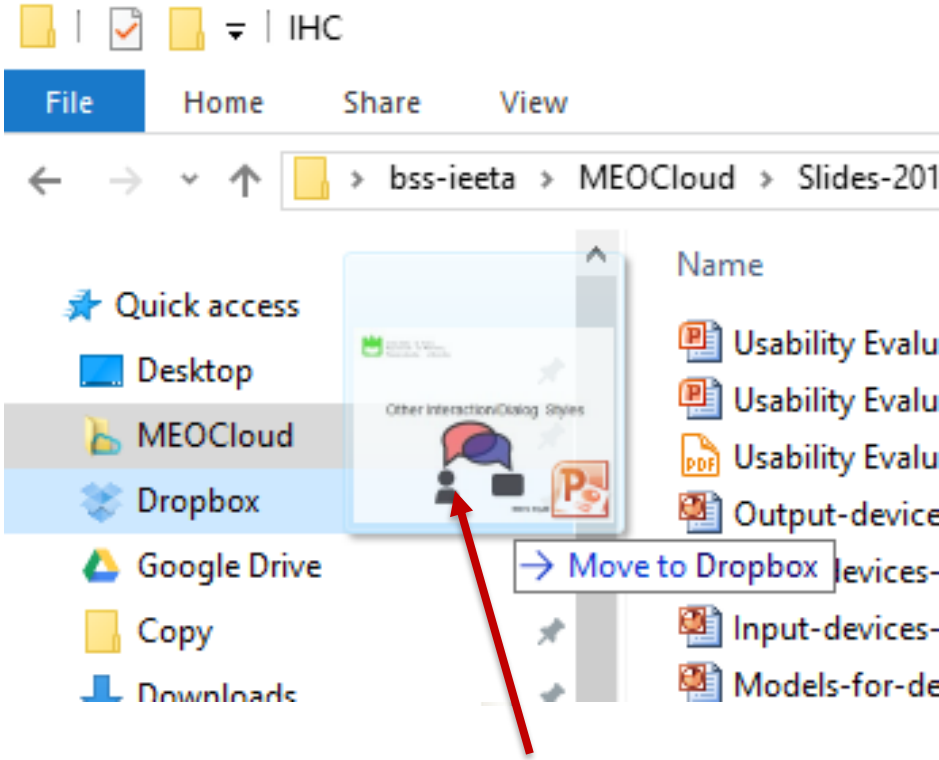
,

In this case the user is still able to draw rectangles, yet the application has a greater semantic distance (feels less direct)

,

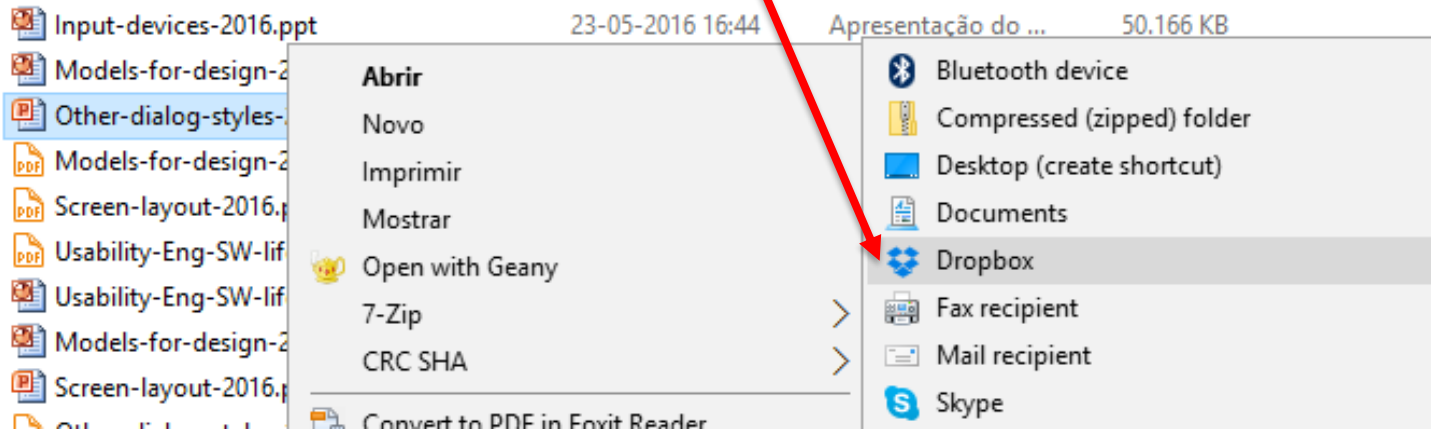
Articulatory Distance

When the physical way actions are performed is more similar to their meaning, articulatory distance is smaller



Selecting an option corresponds to a greater articulatory distance

Dragging an icon corresponds to smaller articulatory distance



- There are two relevant aspects in any user interaction:

Object specification	Action specification
name generation	name generation (write a name)
visual correlation	visual correlation (select)
	gesture generation (draw a symbol)
	analogous action
	coded selection (write a command)

- Specifying objects by **visual correlation** (pointing/selecting) implies the presence of **direct manipulation**

- How **actions** are specified defines the **degree of direct manipulation**

Objects

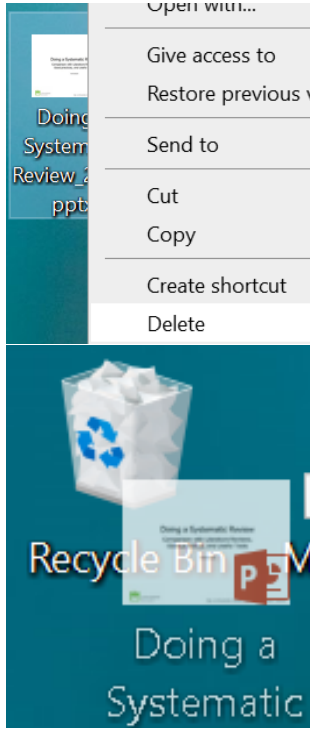
visual correlation +

Actions

name generation
visual correlation
analogous action

- direct UI

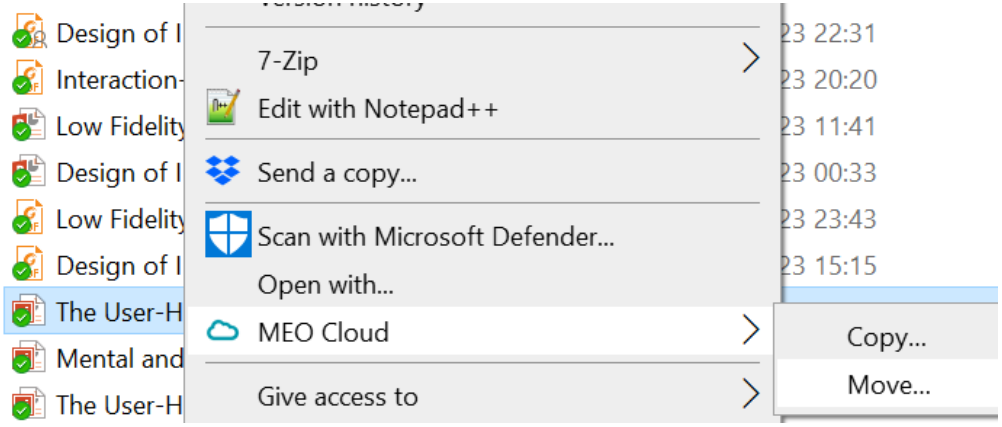
+ direct UI



Examples

```
bi@ub:~/Desktop$
bi@ub:~/Desktop$ mv java-how-to.txt smartbike_paper/
bi@ub:~/Desktop$
bi@ub:~/Desktop$
```

Not direct manipulation UI: name generation + name generation



Direct manipulation UI:
visual correlation + visual correlation

+ Direct manipulation UI:
visual correlation + analogous action

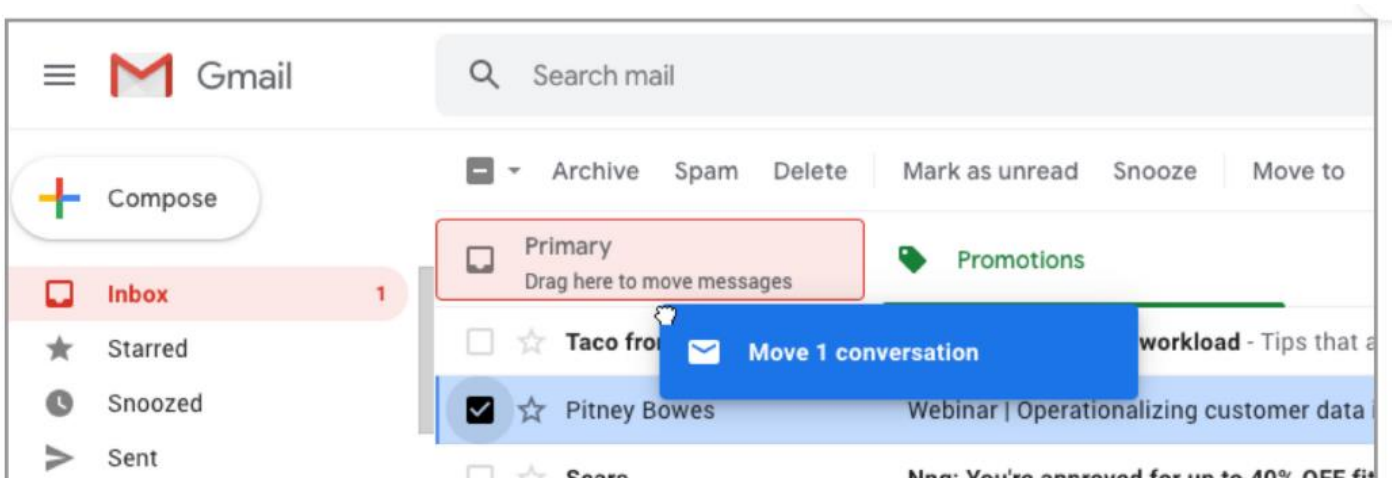
Example

Drag and drop is a type of direct manipulation, particularly useful for:

- grouping, reordering, moving, or resizing objects

Is a core example of direct manipulation

Users interact with an application by selecting an object and moving it by dragging it across a virtual screen or window, dropping it into a location



Drag–and–Drop: How to Design for Ease of Use - NN/g

Main advantages and disadvantages
of direct manipulation UIs

Advantages (potential)

- Easy to learn and remember (are great for novices with good design
- Direct, WYSIWYG (What you see is what you get)
- Flexible, easily reversible actions
- Immediate visual and context feedback
- May be less prone to errors

Disadvantages

- Not auto-explanatory
- May be inefficient
- Repetitive tasks are not well supported
- Some gestures can be more error-prone than typing
- Difficult to draw recognizable icons (particularly for actions)
- Icons occupy more screen real estate then text

Interaction Styles | The Glossary of Human Computer Interaction

User profile to whom direct manipulation is adequate:

Knowledge and experience:

- Moderate system experience
- Moderate to high task experience
- Frequent usage of other systems
- Low computational literacy

Work and task

- Low frequency of use
- Moderate training
- Optional usage
- Low structured tasks

Direct Manipulation design: some guidelines

Minimize articulatory and semantic distance

Use general guidelines to design a usable UI:

- Coherence
- Good conceptual model
- Feedback
- Adequate organization of functionality
- Adequate screen layout
- Adequate colour usage
- Adequate error handling
- Etc.

A Brief History of the Origin of the Computer Icon | IxDF

Icons

Icon in computing is an image that is supposed to represent a function

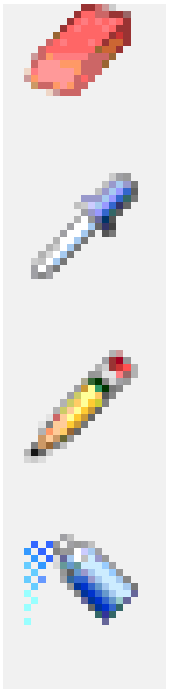
Are not easy to design …

A Brief History of the Origin of the Computer Icon | IxDF

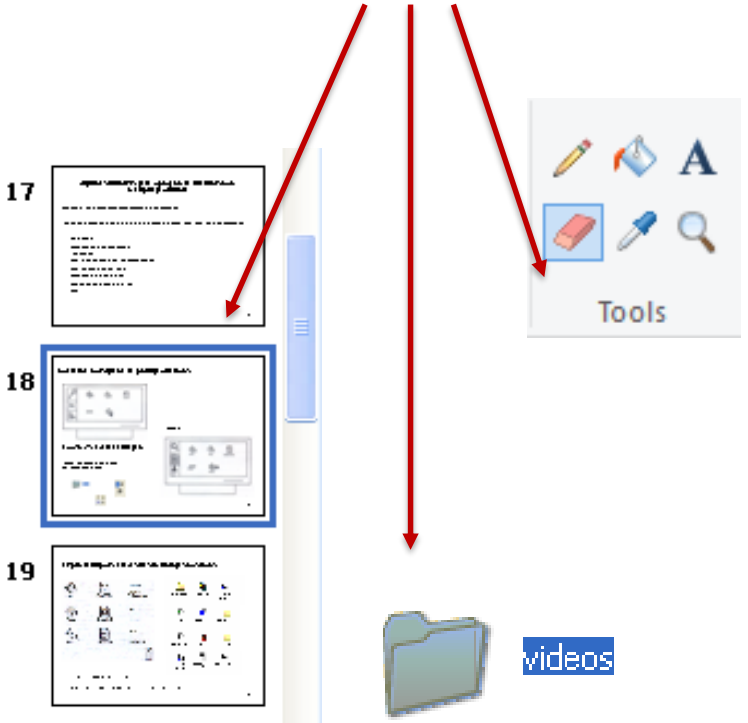
Use a coherent Icon scheme

Different schemes:

better: same scheme

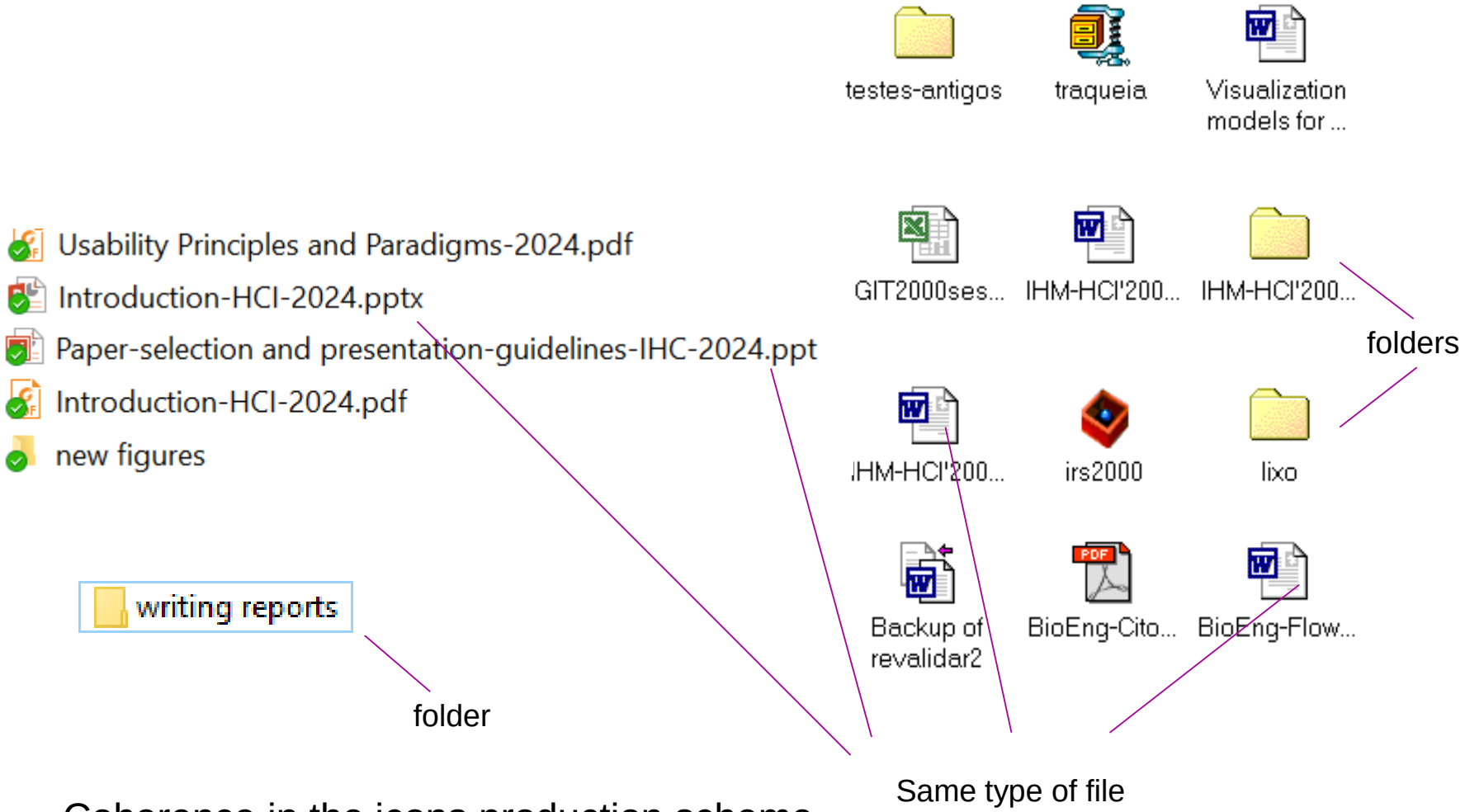


Visual selection feedback



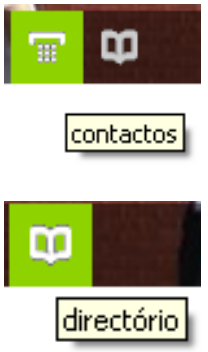
Adding names
(+ recognizable)

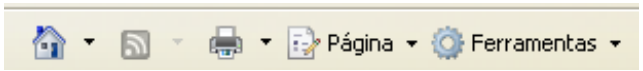
Express relation through icon similarity

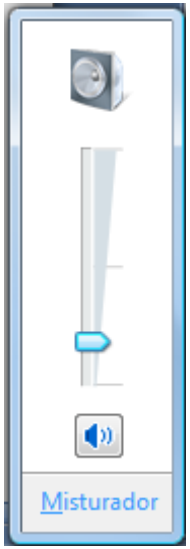


Coherence in the icons production scheme

Add names to icons to make them more recognizable (recognition rather than recall)

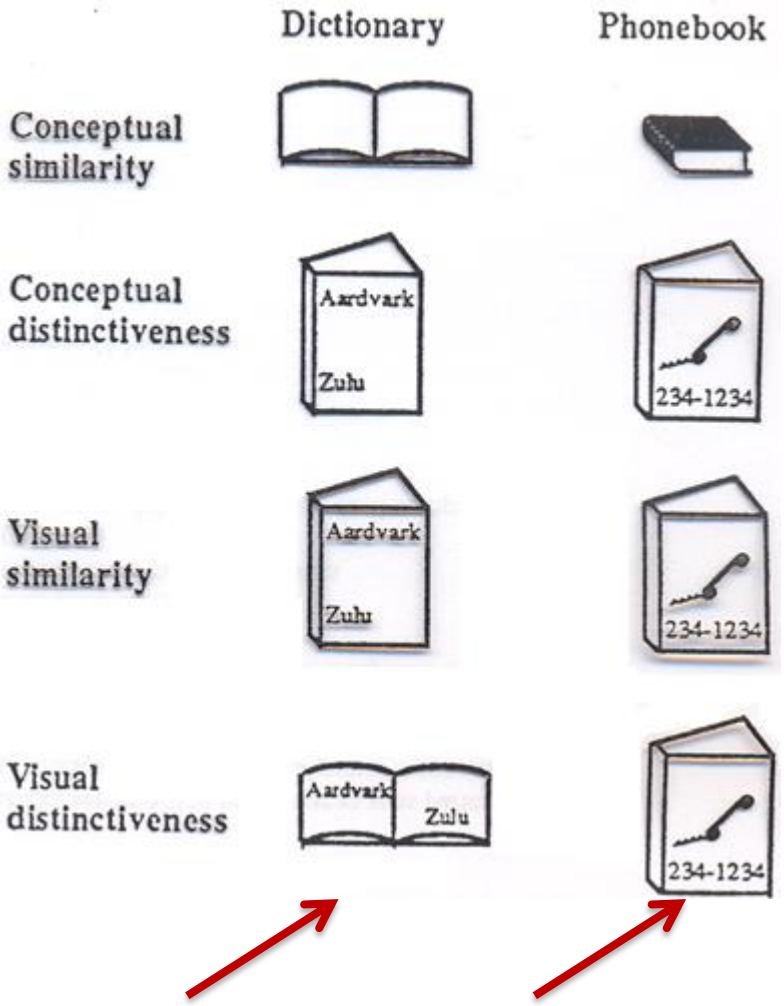


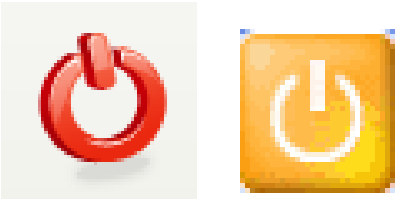




Allow name definition

Icons must be conceptually and visually distinctive
(recognition rather than recall)

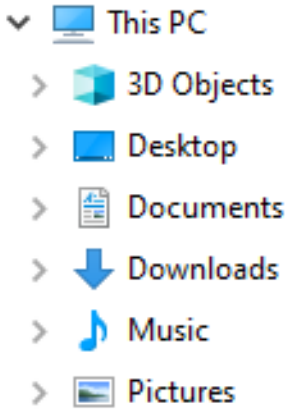




Conceptually similar icons

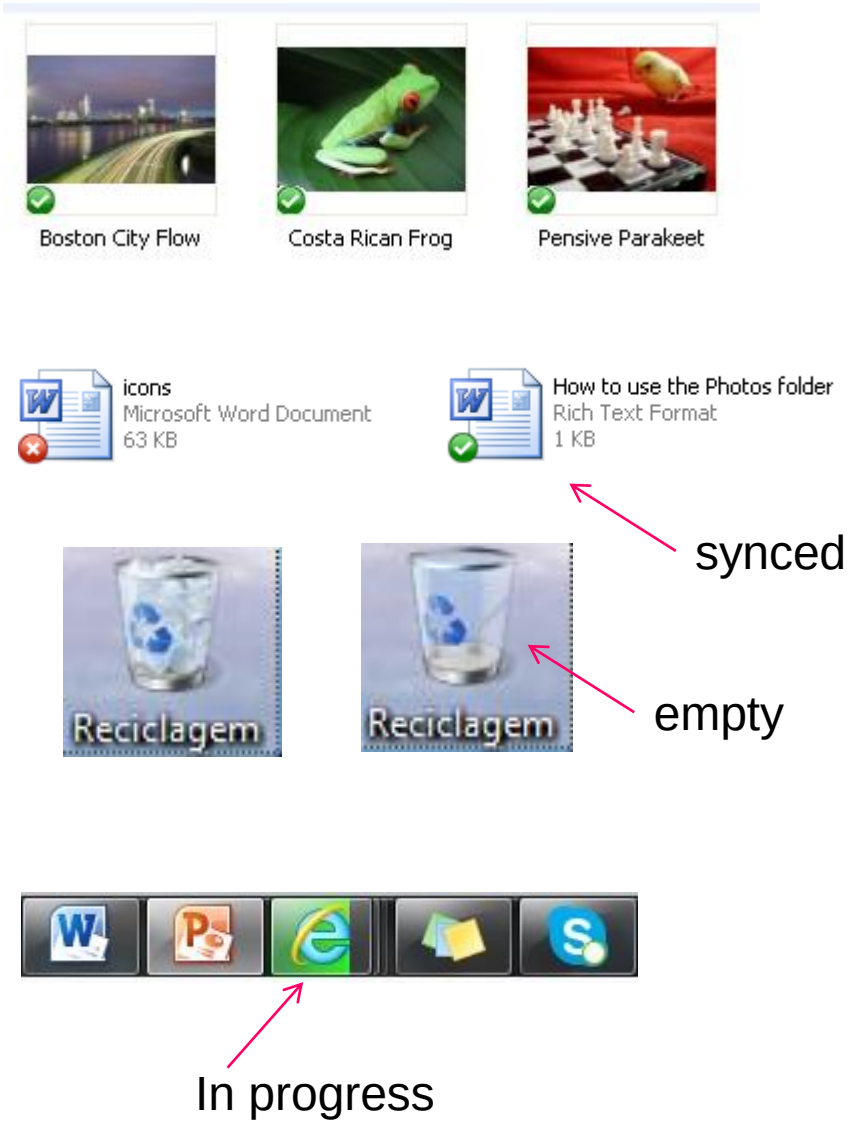
Best solution: conceptually and visually distinct

Icons should be specific/familiar not abstract/non-familiar (familiarity)



Familiar Icons?

Express objects' attributes through icons
(visibility of the system status)



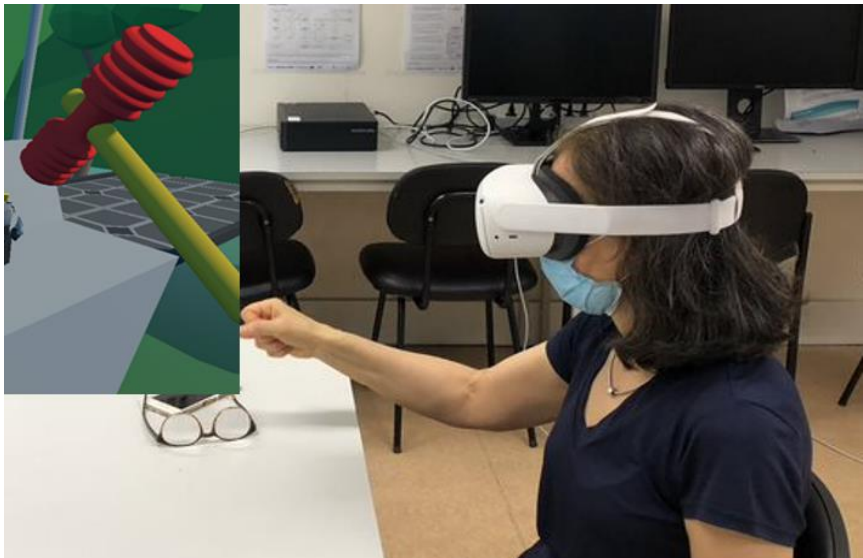
Direct manipulation - Concluding remarks

“It’s hard to imagine modern interfaces without direct manipulation …

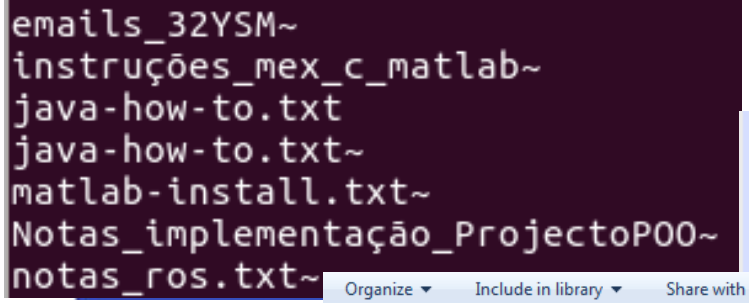
Augmented-reality and virtual-reality systems will push DM to even newer limits …

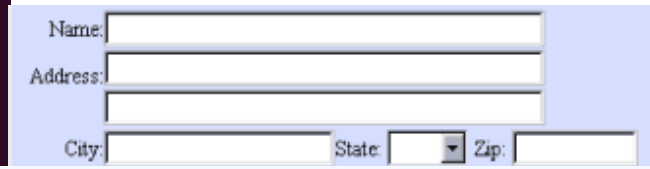
Despite the many downsides, we still recommend a heavy dose of direct manipulation for most UIs”

Direct Manipulation: Definition - NN/g



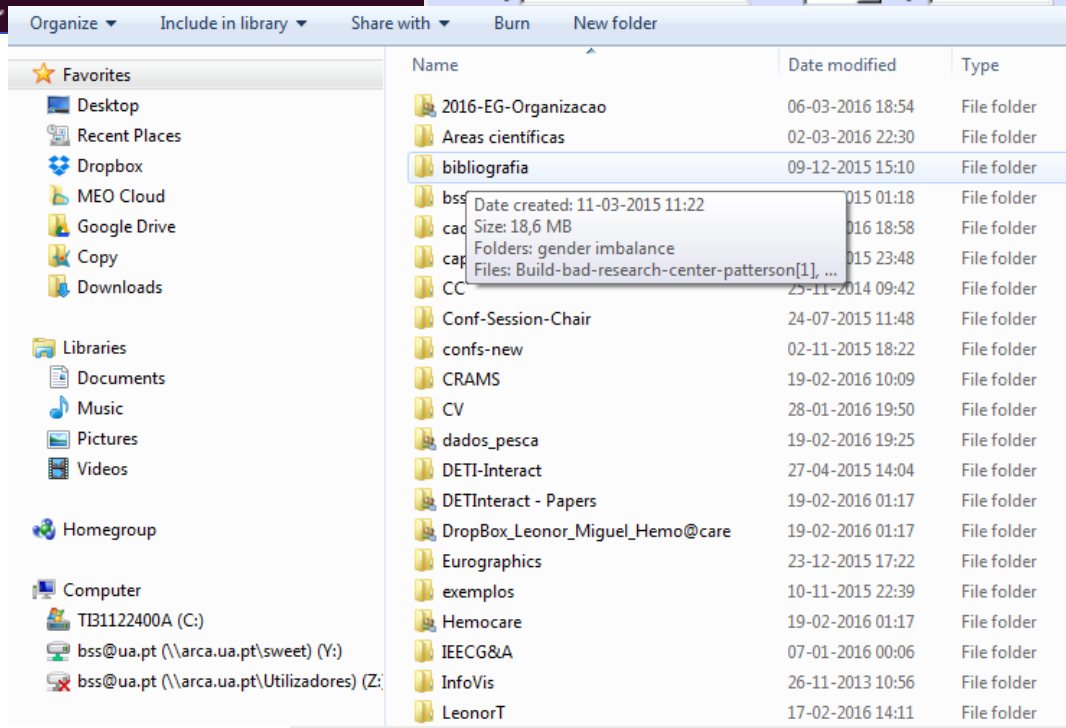
Interaction styles



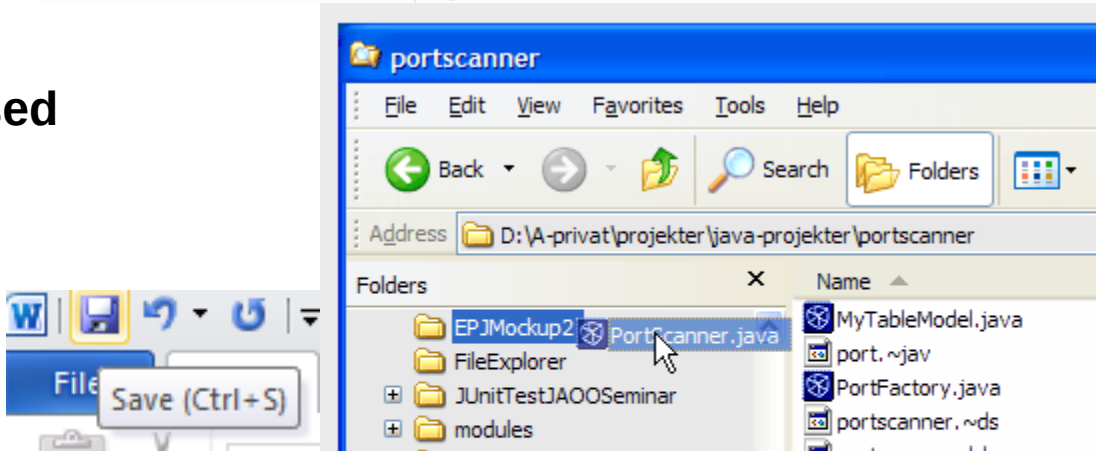


A possible classification:

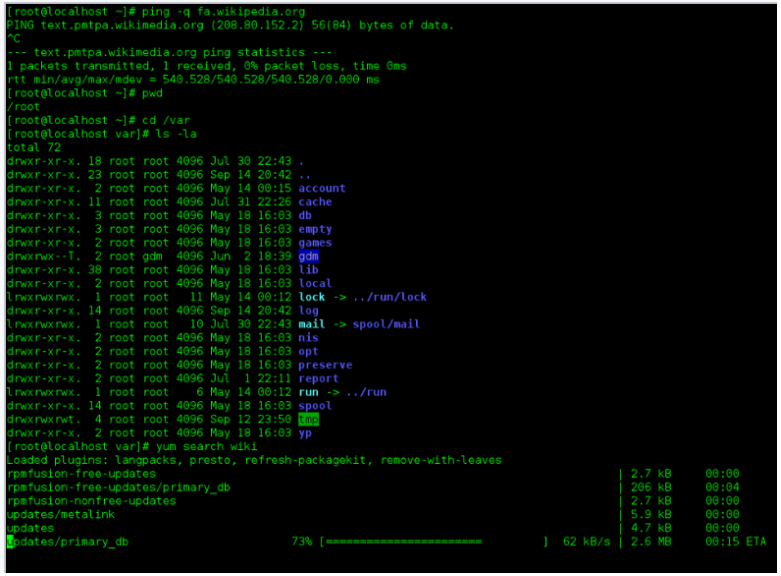
- Menus
- Fill-in-forms
- Direct manipulation
- Command languages
- Natural languages
- Etc.



Often two or more styles are used simultaneously; why?



Command Languages



Command languages

```
cd /tmp
echo "line 1
line 2
line 4" > tmp1$$
echo "line 2
line 3" > tmp2$$
diff tmp1$$ tmp2$$
rm tmp1$$ tmp2$$
```

```
guru99@VirtualBox:~$ history
    1  cat > sample
    2  cat sample
    3  cat sample ^a
    4  cat sample a
    5  cat sample | grep a
    6  cat sample | grep ^a
    7  useradd home
    8  useradd mycomputer
    9  sudo useradd mycomputer
   10  sudo adduser MyLinux
   11  sudo adduser mylinux
   12  vi scriptsample.sh
```

Shall also be designed as to be as usable as possible!

Command-line interface - Wikipedia

Basic Goals of Language Design

- Precision
- Compactness
- Ease in writing and reading
- Speed in learning
- Simplicity to reduce errors
- Ease of retention over time

Usability Questions concerning a command language

- Does the language support necessary functions?
- Is it fast to enter a command?
- Is it easy to recognize what the command might do?
- Is it easy to recall a command?
- Are there few errors when using the language?

Main advantages and disadvantages

Advantages (potential)

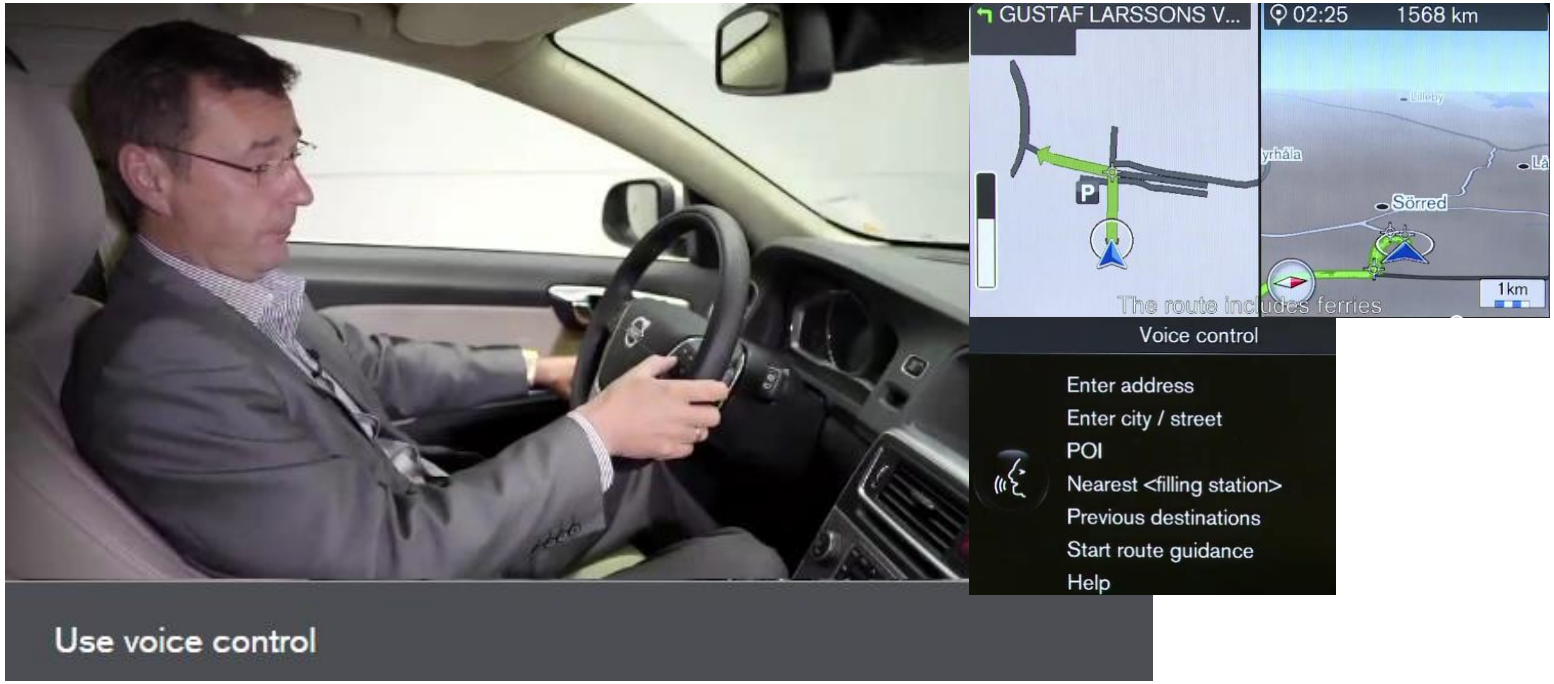
- Powerful
- Flexible
- Efficient
- Do not take much screen real estate

Disadvantages

- Difficult to learn
- Not self-explainable
- Error prone
- Improvements are not visible

Note that:

Command languages may be used
not only through text but also via voice
But they must be very simple …
e.g.
While driving a car to control the media, the phone or navigate



Interaction style: command language;
interaction devices: speech recognition/synthesis

Relevant issues in Command Language design

- Semantics
- Syntax
- Lexicon
- Interaction

Command Languages Design guidelines

Balance richness and minimalism
(similar to semantic distance in direct manipulation)

Examples :

Rich	**Minimal**
Delete	Delete
Insert	Insert
Replace	
Copy	Copy
Move	Delete
Rename	
Delete	

(the functionality is the same)

Use a coherent syntaxe

Use a natural and easy to remember action-object grammar

```
VolB!FileA!D$$
FileA!VolB!ER$L!:KO:!*$$
```

Uncoherent syntax and unfamiliar commands

```
search filea volb.
open filea volb.
list all lines with “KO”.
```

or

```
s filea volb.
o filea volb.
lal “KO”.
```

Command abbreviations should be simple and coherent

Easy to remember (not easy to recognize as for function keys)

	Abbreviations	
Name	**Poor:**	**Improved:**
Move forward	MovF	MovF
Move backward	Mvb	MovB
Insert	I	Ins
Delete	Dl	Del
Replace	Repl	Rep
Search	Srch	Sea
Delete	X	Del
Send	Sn	Sen
Print	Prt	Pri
Search	Srch	Sea
Send	Sn	Sen
Find	Fi	Fin
Choose	Ch	Cho

Allow the following interaction features:

- Defaults
- Command edition
- Intelligent interpretation
- Type-ahead
- Feedback
- Help and documentation
- Make the language “user tailorable”

Example of intelligent interpretation:
“delate”: did you mean “delete”? Y or N

Example of a (complex) command with defaults

ls - Linux man page

Name

ls - list directory contents

Synopsis

ls [*OPTION*]... [*FILE*]...

Description

List information about the FILEs (the current directory by default). Sort entries alphabetically if none of **-cftuvSUX** nor **--sort**.

Mandatory arguments to long options are mandatory for short options too.

-a, **--all**
do not ignore entries starting with .
-A, **--almost-all**
do not list implied . and ..
--author
with **-l**, print the author of each file
-b, **--escape**
print octal escapes for nongraphic characters

You don't need to use all arguments; there are default values

-d, **--directory**
list directory entries instead of contents, and do not dereference symbo
-D, **--dired**
generate output designed for Emacs' dired mode
-f
do not sort, enable **-aU**, disable **-ls --color**
-F, **--classify**
append indicator (one of */=>@|) to entries
--file-type
likewise, except do not append '*'
--format=*WORD*
across **-x**, commas **-m**, horizontal **-x**, long **-l**, single-column **-1**, verbose
--full-time
like **-l --time-style**=*full-iso*
-g
like **-l**, but do not list owner
--group-directories-first
group directories before files.
augment with a **--sort** option, but any
use of **--sort**=*none* (**-U**) disables grouping
-G, **--no-group**
in a long listing, don't print group names
-h, **--human-readable**
with **-l**, print sizes in human readable format (e.g., 1K 234M 2G)
--si
likewise, but use powers of 1000 not 1024
-H, **--dereference-command-line**
follow symbolic links listed on the command line

Etc.., etc., etc.

Main Bibliography

- B. Shneiderman, C. Plaisant, M. Cohen, S. Jacobs, *Designing the User Interface- Strategies for Effective Human–Computer Interaction*, 5th ed., Addison Wesley, 2010

- H. Sharp, J. Preece, and Y. Rogers, *Interaction Design: Beyond Human-Computer Interaction*, 5th Edition Wiley, 2019

- M. Soegaard, Interaction Styles, *Interaction Design Foundation Encyclopedia*, 2nd edition,

  Interaction Styles | The Glossary of Human Computer Interaction