

Administrative Stuff

Get slides for the interim presentation to me by Thursday morning, 30 minutes before class!

How to Use Qualitative Data

Supplement quantitative findings

Explain quantitative findings

Create quantitative data to statistically analyze

Supplementing Quantitative Data

Our results showed that people were faster at using our interface. Users also commented on this:

“I was surprised at how fast I was able to do it!”

“It was way easier to do than I was expecting.”

Explaining Quantitative Data

Our results showed that people were slower at using our interface. We found in our observations that users tended to get distracted by the supplemental information book and enjoyed the extra information we provided:

“Normally I just try to finish as fast as possible, but now I can see that there are all these other options so I wanted to check them out!”

Creating Data to Statistically Analyze

The screenshot displays a software interface for video analysis. The top-left pane shows a video of a child sitting at a desk with a computer monitor. The top-right pane shows a close-up of the child's face with a green, starburst-like visualization. The bottom-left pane is a timeline from 15 to 118 seconds, featuring several tracks: a top track with colored diamond markers (b, d, a, e, b, e, b); a track with horizontal bars and markers (c, f, h); a red line graph; and a blue waveform at the bottom. The right-hand side contains a list of behavioral markers, each with a color-coded label and control buttons:

p Non-Child Audio	In	+Note
h Smiling	Mark	+Note
z Grin	Mark	+Note
j No Face	Mark	+Note
k Oriented @ Scrn	Mark	+Note
l Auditory Focus	Mark	+Note
q Laugh	In	+Note
w Non-Speech Voc	Mark	+Note
e Speech-Like Voc	In	+Note
r BIGmack Switch	Mark	+Note
f Turn Taking	Mark	+Note
s Open Posture	Mark	+Note
d Look Away	Mark	+Note
x Differed Imitation	Mark	+Note
c Immediate Imitation	Mark	+Note
m In Chair	In	+Note

At the bottom right, there are playback controls:

- Interval Playback Mode:
 - Continuous Intervals
 - Interval

Interaction Fundamentals Part I

Irene Rae
Computer Sciences

CS-570 Introduction to Human-computer Interaction



WISCONSIN
UNIVERSITY OF WISCONSIN-MADISON

What is Interaction?

Two-way

Lest we only have one-way or reaction.

Communicative

Information is sent.

Receptive

Information is received.

Effective

The parties are changed as a result.

Thought Experiments

Questions:

Is it two-way? Are all parties acting?

Communicative? Is information sent?

Receptive? Is information received?

Effective? Are the parties changed?

Clicking a Submit button on a web page.

Typing on a computer that is turned off.

Submitting punch cards to a batch processing system.

Picking up the telephone and putting it to your ear.

Typing ESC on a screen that does not allow it.

One computer sending data to another via a network.

Two turned-on televisions facing each other.

Interaction Models

A way of describing interaction and its component parts.

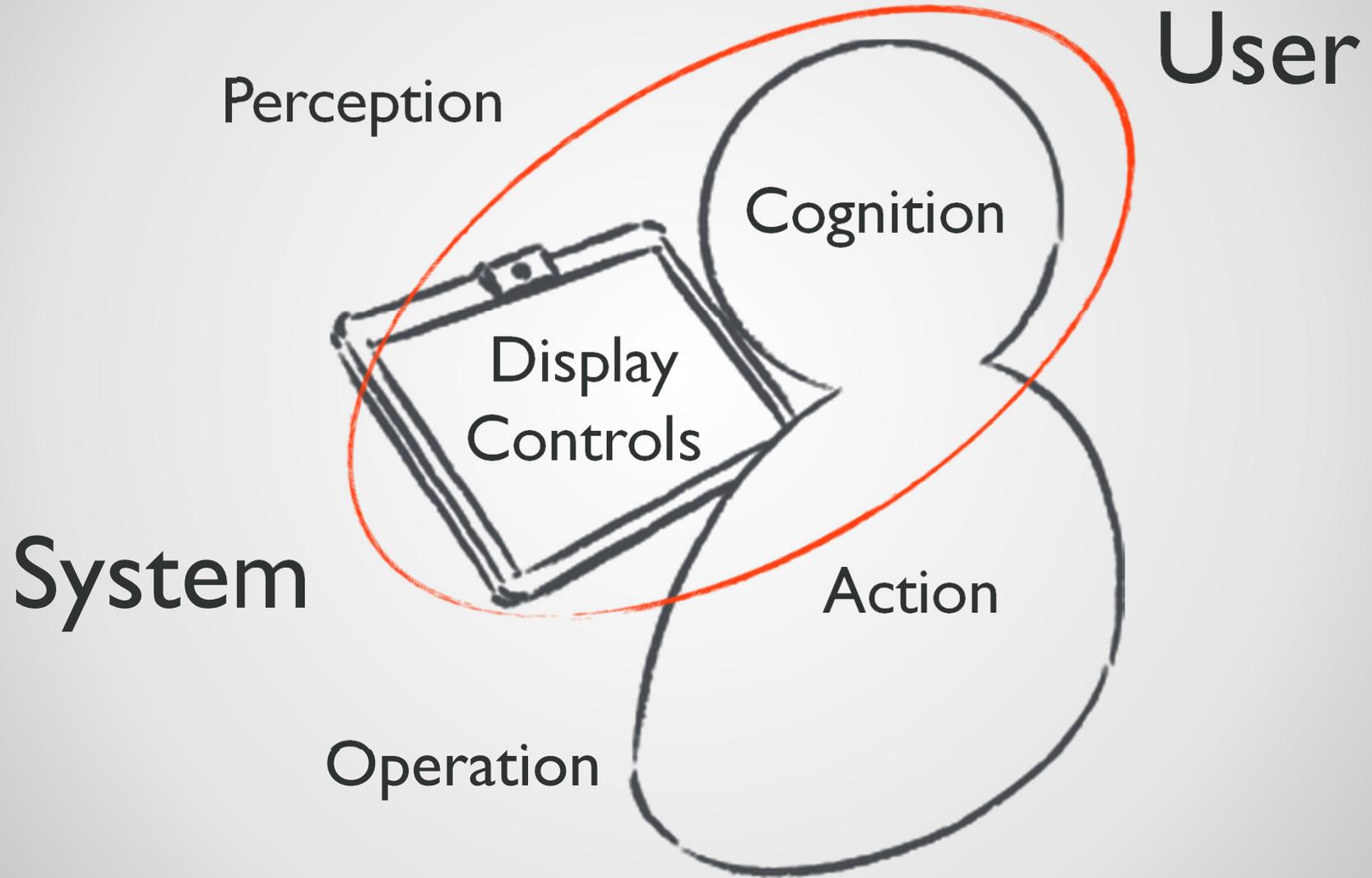
Interaction models

Help us define and describe an interaction.

Help us isolate areas where problems occur.

Help us design new interaction.

Human-Computer System



Linguistics Model

Conceptual

Users' communication and interpretation of meaning within the system.

Semantic

Meanings are conveyed by sequences of users' commands.

Syntactic

Low-level units of action are combined to form commands.

Lexical

Device dependencies and low-level actions a user takes.

Linguistics Model

Conceptual

Dropping a folder in the trash should allow me to remove it from my system.

Semantic

Drag-and-drop a folder into the trash bin to prepare it for deletion.

Syntactic

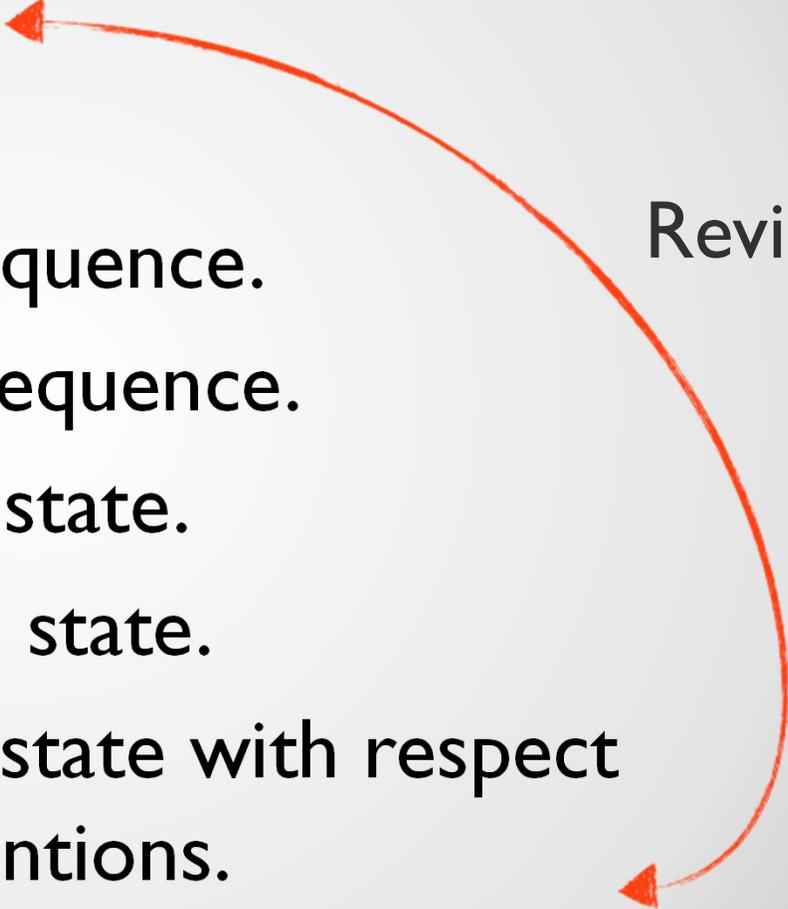
Left mouse button is released over the trashcan icon. Until this moment, the button had been held down over a folder icon and the mouse was moved to the trashcan.

Lexical

Release left mouse button.

Execution-Evaluation

1. Establish the goal.
2. Form the intention.
3. Specify the action sequence.
4. Execute the action sequence.
5. Perceive the system state.
6. Interpret the system state.
7. Evaluate the system state with respect to the goals and intentions.



Revise

* Norman

Turning on the Light

1. Establish the goal

Increase the light in the room.

2. Form the intention.

To turn on the desk lamp.

3. Specify the action sequence.

Walk to the lamp, reach for the knob, twist the knob.

4. Execute the action sequence.

[walk, reach, twist]

5. Perceive the system state.

[hear “click” sound, see light from lamp]

6. Interpret the system state.

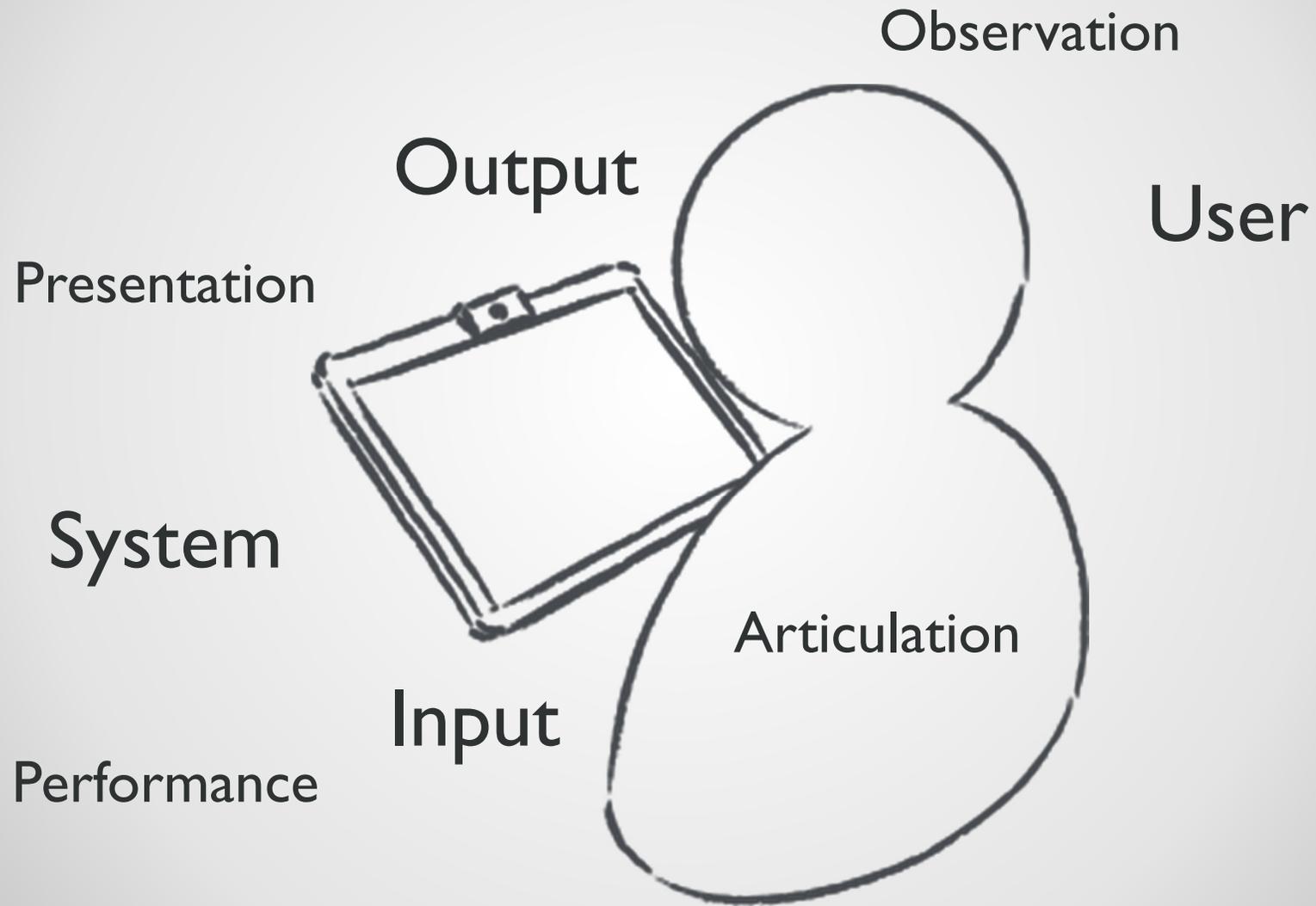
The knob rotated, the lamp is emitting light. The lamp seems to be working.

7. Evaluate the system state with respect to the goals and intentions.

The turned-on desk lamp did indeed increase the light in the room.

[goal satisfied]

Interaction Framework



Interaction Style

Interaction can be thought of as a dialog between the user and the computer.

All dialogs occur in some language as defined by the abilities of the user and the capabilities of the system.

The choice of language is determined by the **interaction style**.

Example Styles

Batch

Command line

Menus

Natural language

Question/answer

Form fill-in

WIMP: windows, icons,
menus, pointers

Point and click

Agent-based, delegation

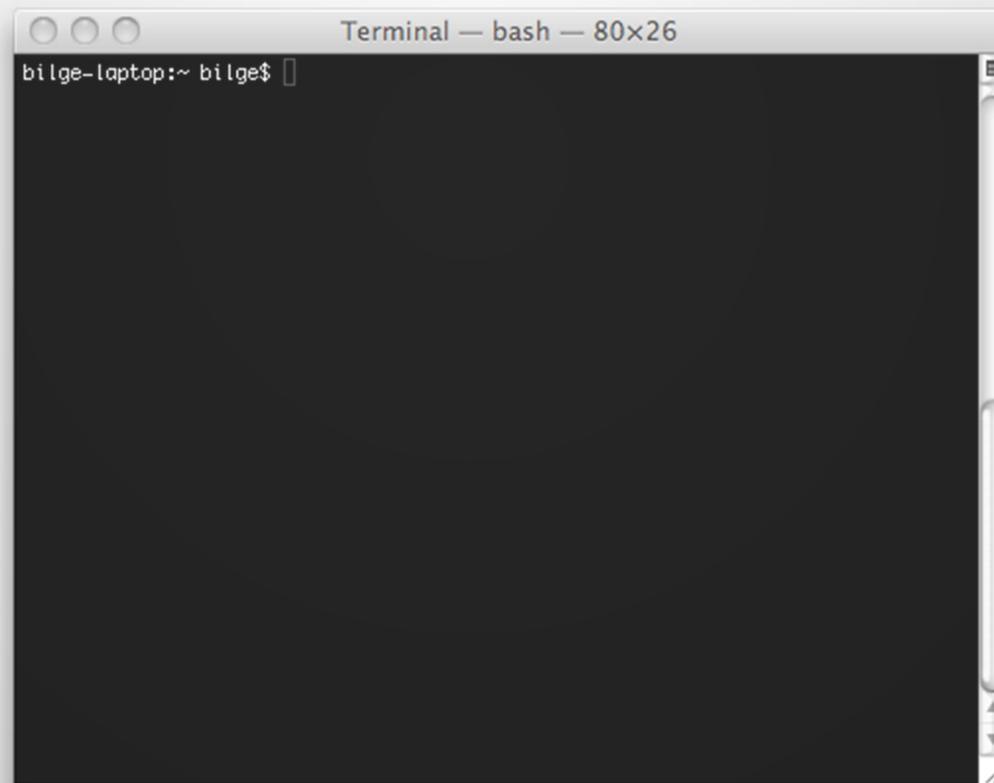
Conversation

Direct manipulation

Command Line

What do they do well?

What do they do poorly?



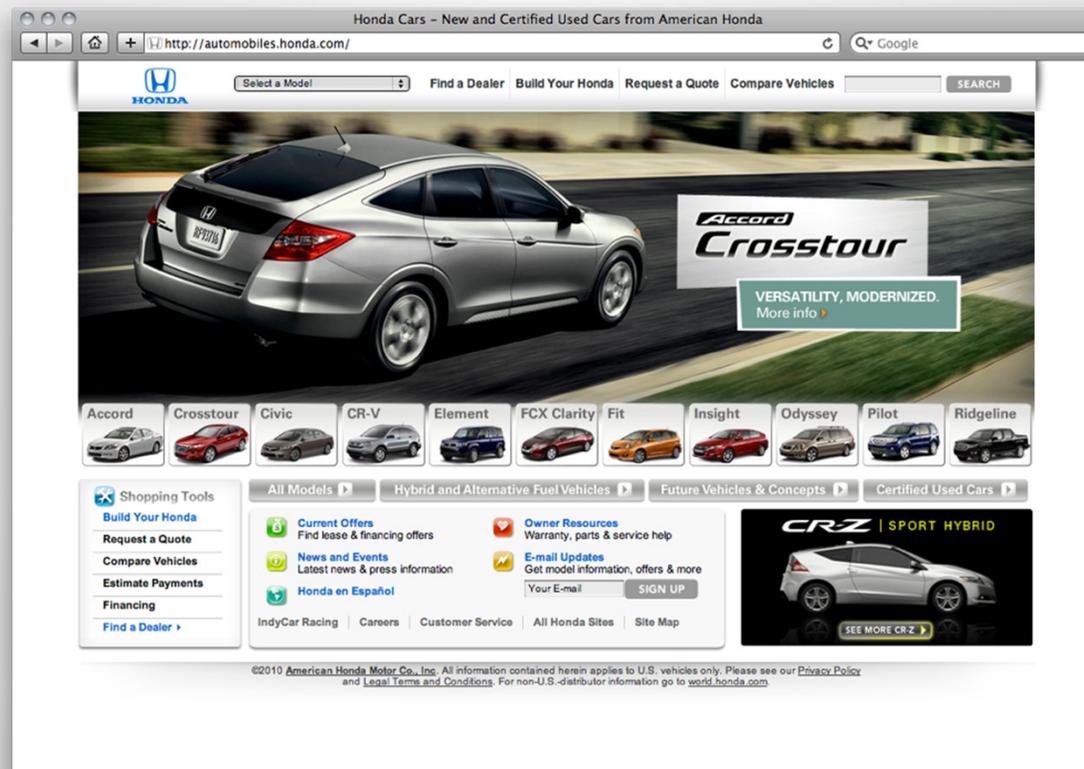
Command Line

- + First interactive style
- + Offer direct access to system functionality
- + Easy for developers to extend the system
- + Often commands can be easily aggregated
- + Good at repetitive tasks
- + Global access to system functionality
- Difficult to learn and use
- Relies on recall, not recognition
- Difficult to see system status
- Commands are often obscure abbreviations

WIMPs: Windows, Icons, Menus, Pointing

What do they do well?

What do they do poorly?



WIMPs

- + Support direct manipulation
- + Support high visibility of objects
- + Relies on recognition, not recall
- + Usually easy to learn
- Do not support repetitive tasks
- Slower than command line interfaces
- Difficult to aggregate commands

Pens

What do they do well?

What do they do poorly?



Pens

+ Devices designed for quick on, quick off

+ Facilitates pointing anywhere quickly

+ Allows for writing like on paper

+ Drawing is natural and fluid

- Usually requires two hands

- Lacks a cursor and related interactions (e.g., hover)

- Text entry slower than a keyboard

- Lack of screen real-estate a significant challenge

Multi-Touch

What do they do well?

What do they do poorly?



Multi-Touch

- + Enables a variety of new interaction styles
- + Intuitive operations are possible
- + Offers high portability
- + Scalable (iPod vs. Microsoft Surface)
- Precise operation is difficult
- Usually requires two hands
- Text entry error-prone
- Hands block screen

Questions?

Conceptual Model

A user's **conceptual model** (also called a **mental model**) is his view of a system as s/he **believes it to be**, especially concerning how s/he can act upon it and what the system's responses mean.



Cooper's Model

Implementation model

How it actually works

Manifest model

How it represents itself to the user

Mental model

How the user thinks it works and what it means

Conceptual/Mental Models

Conceptual models allow users to ask:

What can I do?

How can I do it?

What result will it have?

What is it telling me?

A good conceptual model is crucial for users to have a successful interaction.

The manifest model is responsible for forming an accurate conceptual model within the user.

Styles & Models

Different interaction styles employ different conceptual models.

“How do I execute a command in cmd vs. WIMP?”

A user’s conceptual model is informed by the interaction style, the physical characteristics, and the available actions.

Conceptual models matter!

The Apple Newton and handwriting recognition.

Video!

TopHat Question!

Conceptual/Mental Model Design

Conceptual models are informed by many things:

Metaphors

Modes

Affordances

Constraints

Mappings

Prior experiences

Expectations

Ontology

An ontology is a specification of the “things that exist” or “things in the world.”

We can define an ontology for a conceptual model in terms of:

- Objects

- Properties

- Relationships

- Actions

Paint vs. Draw

What are the

Objects?

Properties?

Relationships?

Actions?

How do the two ontologies reveal important differences?

Ontology Example

Painting Program

Objects: Pixels, Canvas

Properties: Location,
Color

Relationships: Pixels
fixed to a canvas

Actions: Place and
remove pixels on the
canvas using various
tools.

Drawing Program

Objects: Shapes, Art board

Properties: Location, Size

Relationships: Shapes
arranged on a canvas

Actions: Move, rotate,
resize shapes

Text Editor Ontology

Objects: Characters, words, paragraphs, pages

Properties: Font face, font size

Relationships:

Words are made of characters

Paragraphs are made of words

Pages contain characters

Actions: Insert character, insert page, change font...

Gulfs

Gulf of Execution: “How do I do it?”

Gulf of Evaluation: “What does it mean?”



Overcoming the Gulfs

Gulf of Execution

Make the commands and actions for controlling the system match the thoughts, language, and goals of the user.

Gulf of Evaluation

Make the output displays present a clear manifest model of the system that is readily perceived, interpreted, and evaluated, creating an accurate model for the user.

Model Formation

As a user interacts, his/her conceptual model is revised whenever s/he must explicitly revisit it, usually because of a breakdown such as unfulfilled expectations or unclear representations.

During a breakdown, the conceptual model is revisited as the system with which we interact goes from ready-to-hand to present-at-hand.

Model Formation

Ready-to-hand (zuhanden)

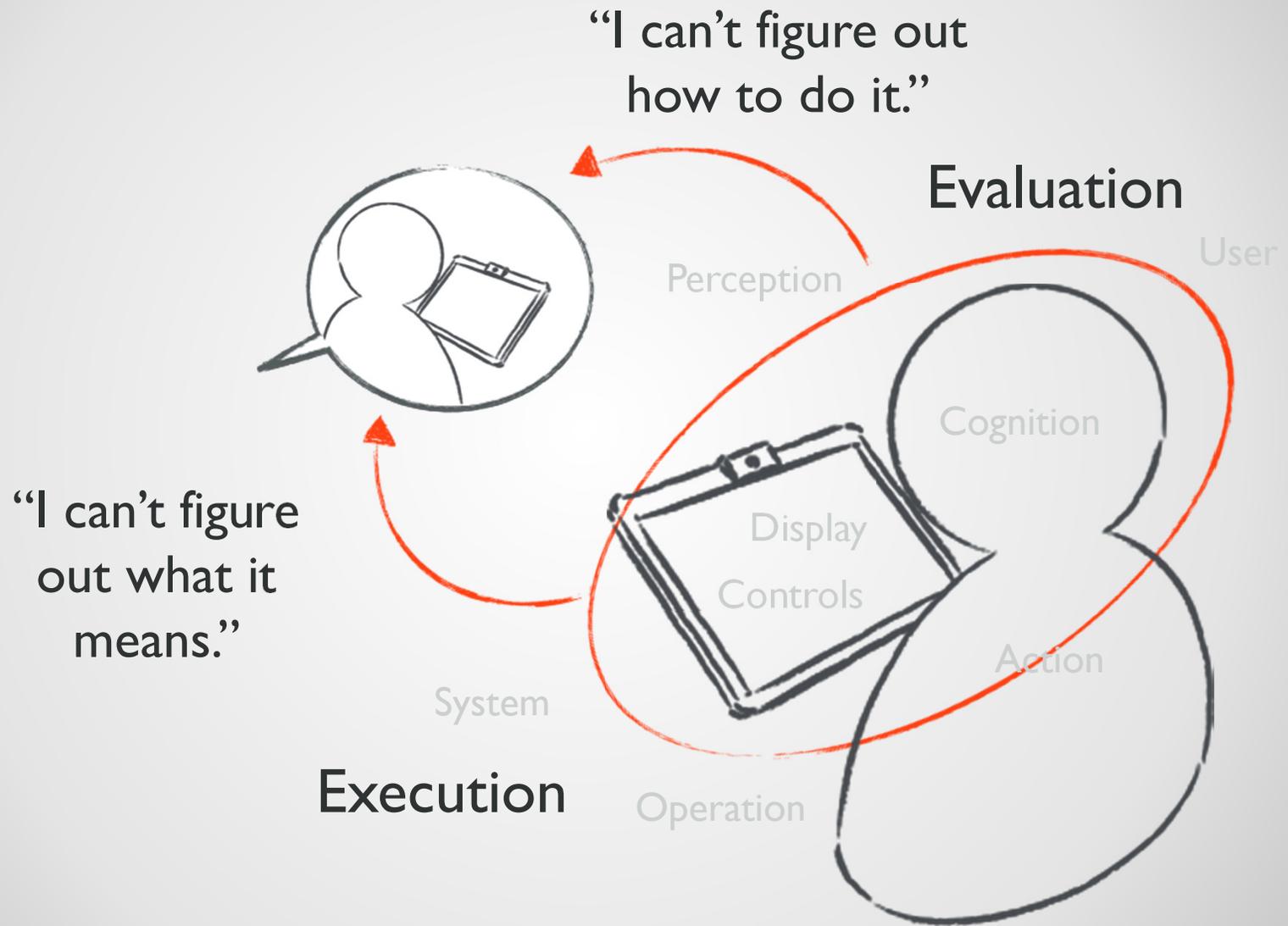
Our normal involvement with tools, where we neither notice them nor theorize about them and instead regard them casually or not at all as we attempt to achieve something.

Present-at-hand (vorhanden)

The moment we truly notice something as it leaps into view, standing apart from any utility or purpose, just as a thing in itself.

This often occurs during a “breakdown” in use.

Revision



TopHat Question!

Struggling Users

Users will struggle until...

Their mental model matches the manifest model of the system.

Does their conceptual model have to match the implementation model?

Constantly revising one's conceptual model can be painful.

Pay close attention the next time you start using a technology that is new to you.

Designing the Conceptual Model

Strive for:

Consistency

Predictability

Visibility

Matching the system to the task & to the way people think about doing a thing.

User-centered concepts, not system-centered ones.

An ontology can help identify areas of weakness.

Questions?

Metaphors

Metaphors help “jump start” users’ conceptual models by pre-populating them with ideas.

How might they do this?

Often new technologies begin as metaphors of old ones. Can you think of some?

Cars were “horseless carriages.”

Trains were “iron horses.”

Radio was called “wire-less.”

Inappropriate Metaphors



Cats don't fetch!

What do you want to search for?

- Pictures, music, or video
- Documents (word processing, spreadsheet, etc.)
- All files and folders
- Computers or people
- Information in Help and Support Center

You may also want to...

- Search the Internet
- Change preferences
- Learn more about Search Companion

Little more than a menu and dialog box.

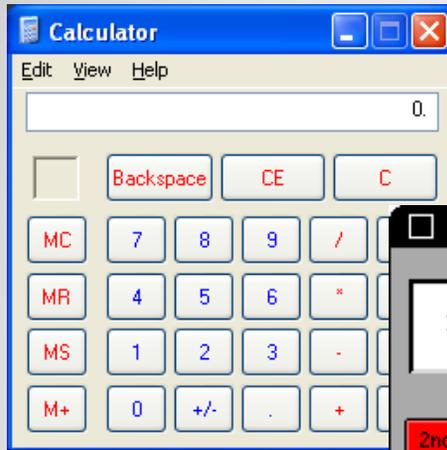


Mixed Metaphor

When two or more different metaphors coexist with some supposed relation.

In Windows, you have both the desktop metaphor (folders, files, etc.) and windows (separate views through which you can see things). But no real desk has windows in it!

Metaphor Examples



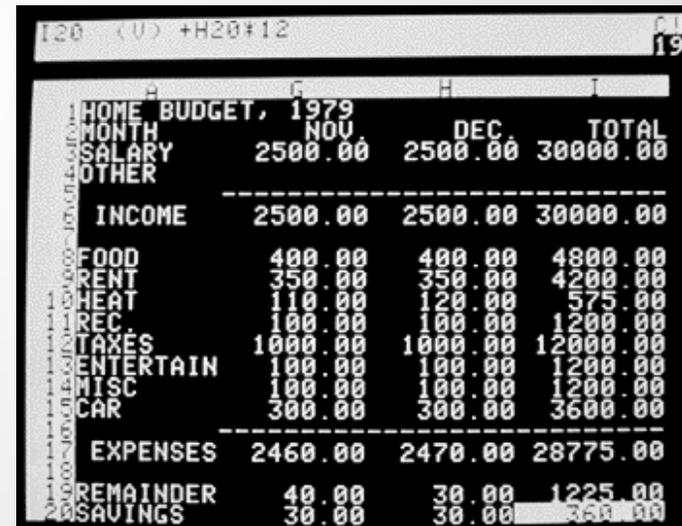
Windows
Calculator



Windows Media Player



SciCalc 2.0



VISICALC (1978)

Other Examples

VCR Controls

Paintbrush

Pencil

Magnifying glass

System tools

Electronic mail

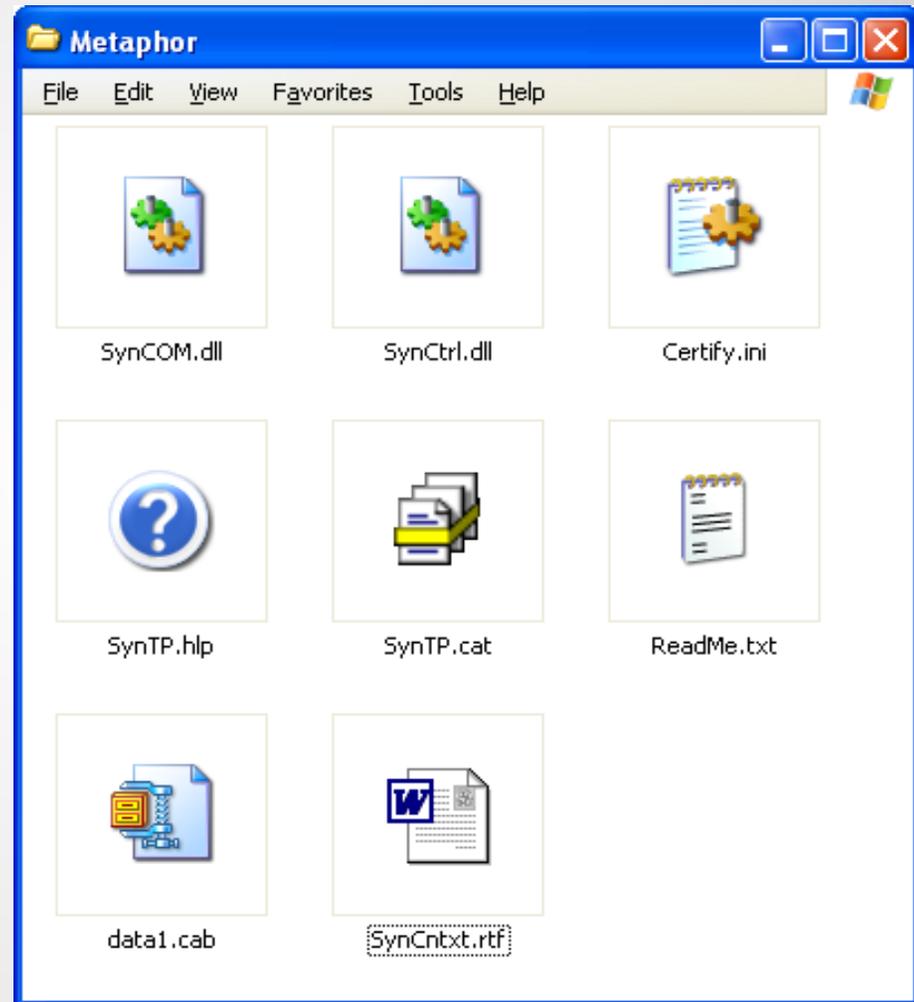
Desktop Metaphor

Files

Documents

Folders

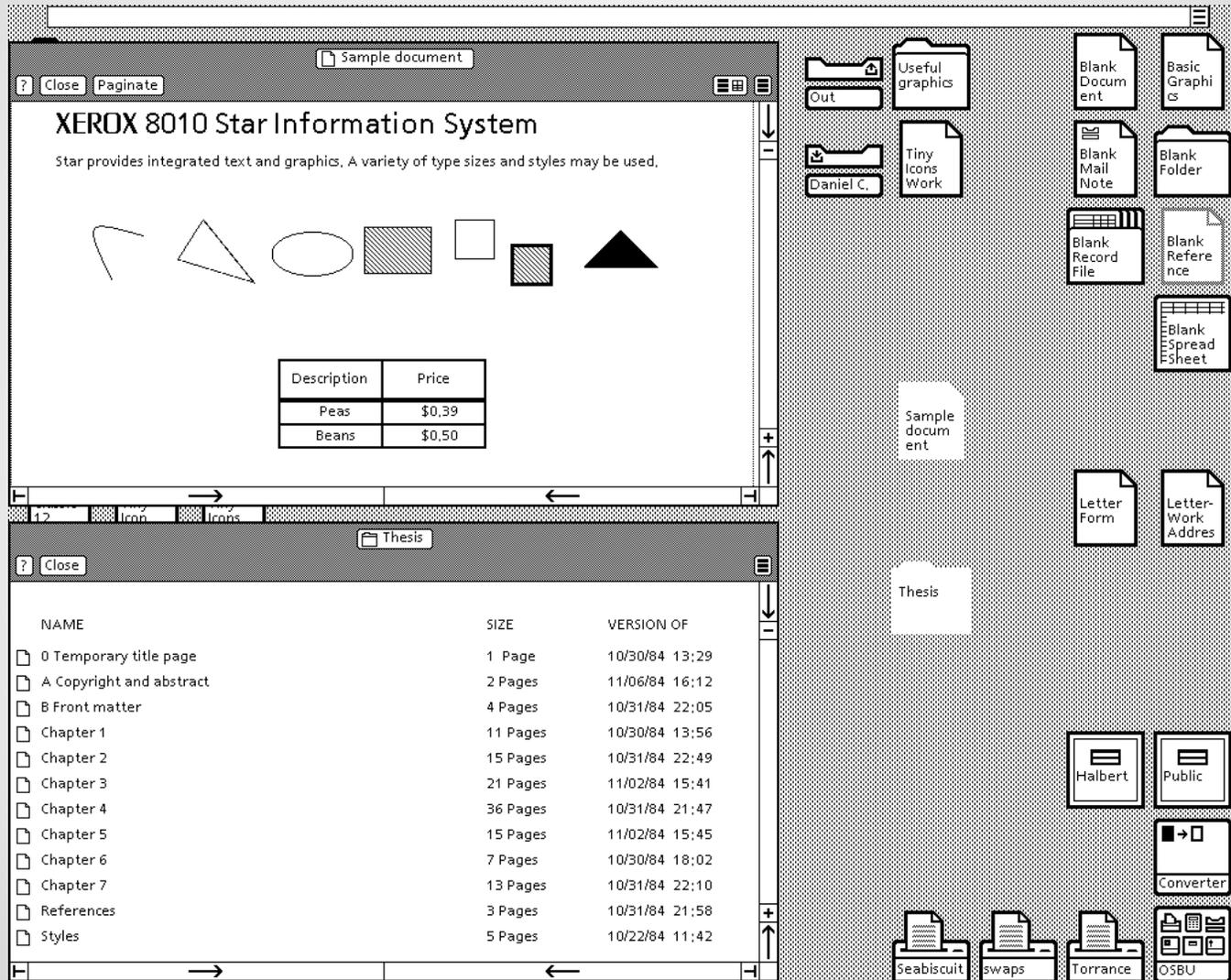
Cabinets



Xerox Star (1979)



Star Screen



Star Screen

XEROX 6085 Workstation
User-Interface Design

To make it easy to compose text and graphics, to do electronic filing, printing, and mailing all at the same workstation, requires a revolutionary user interface design.

Bit-map display - Each of the pixels on the 19" screen is mapped to a bit in memory; thus, arbitrarily complex images can be displayed. The 6085 displays all fonts and graphics as they will be printed. In addition, familiar office objects such as documents, folders, file drawers and in-baskets are portrayed as recognizable images.

The mouse - A unique pointing device that allows the user to quickly select any text, graphic or office object on the display.

See and Point

All functions are visible to the user on the keyboard or on the screen. The user does filing and retrieval by selecting them with the mouse and touching the MOVE, COPY, DELETE or PRINT keys on the keyboard. Text and graphics are edited with the same keys.

Shorter Production Times

Experience at Xerox with prototype workstations has shown shorter production times and thus lower costs, as a function of the percentage of use of the workstations. The following equation can be used to express this:

Table 1: Percentages of use of workstations

Year	Use 6085	Use 6080
1978	95.2	15.8
1980	41.1	59.9
1983	45	55
1984	30	70
1986	10	90
1988	5	95

Activity under the old and the new

Figure 1: Data from Table 1 drive

Workstation usage percentages
Table 1 and illustrated in Figure 1. 6085 users are likely to do the composition and layout, master process including printing and distribution.

Text and Graphics

To replace typesetting, the 6085 offers a choice of type fonts and sizes from 6 point to 36 point.

Here is a sentence of 18 point text.
Here is a sentence of 24 point text.
18-point text.
24-point text.
36-point text.

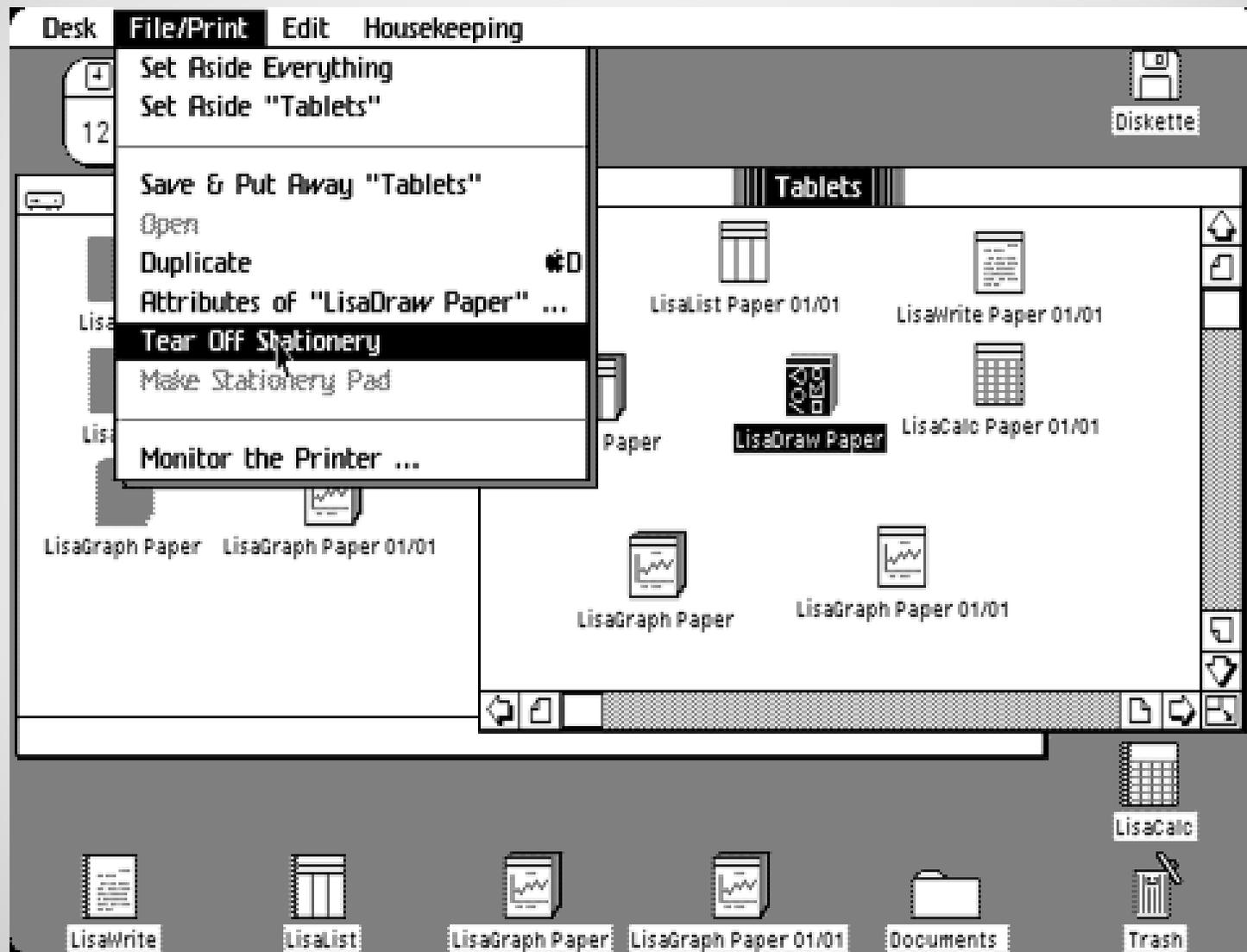
File Manager Window:

NAME	EXTENSION	SIZE	DATE
COMMAND	COM	22677	15-8
AND	SYS	2556	18-8
ASSIGN	COM	984	20-8
ATTRIB	EXE	15093	14-8
BACKUP	COM	17824	20-8
CHKDSK	COM	9435	24-8
CHMOD	COM	6520	27-8
COMP	COM	3018	10-8
DEBUG	EXE	15364	15-8

Apple Lisa (1983)



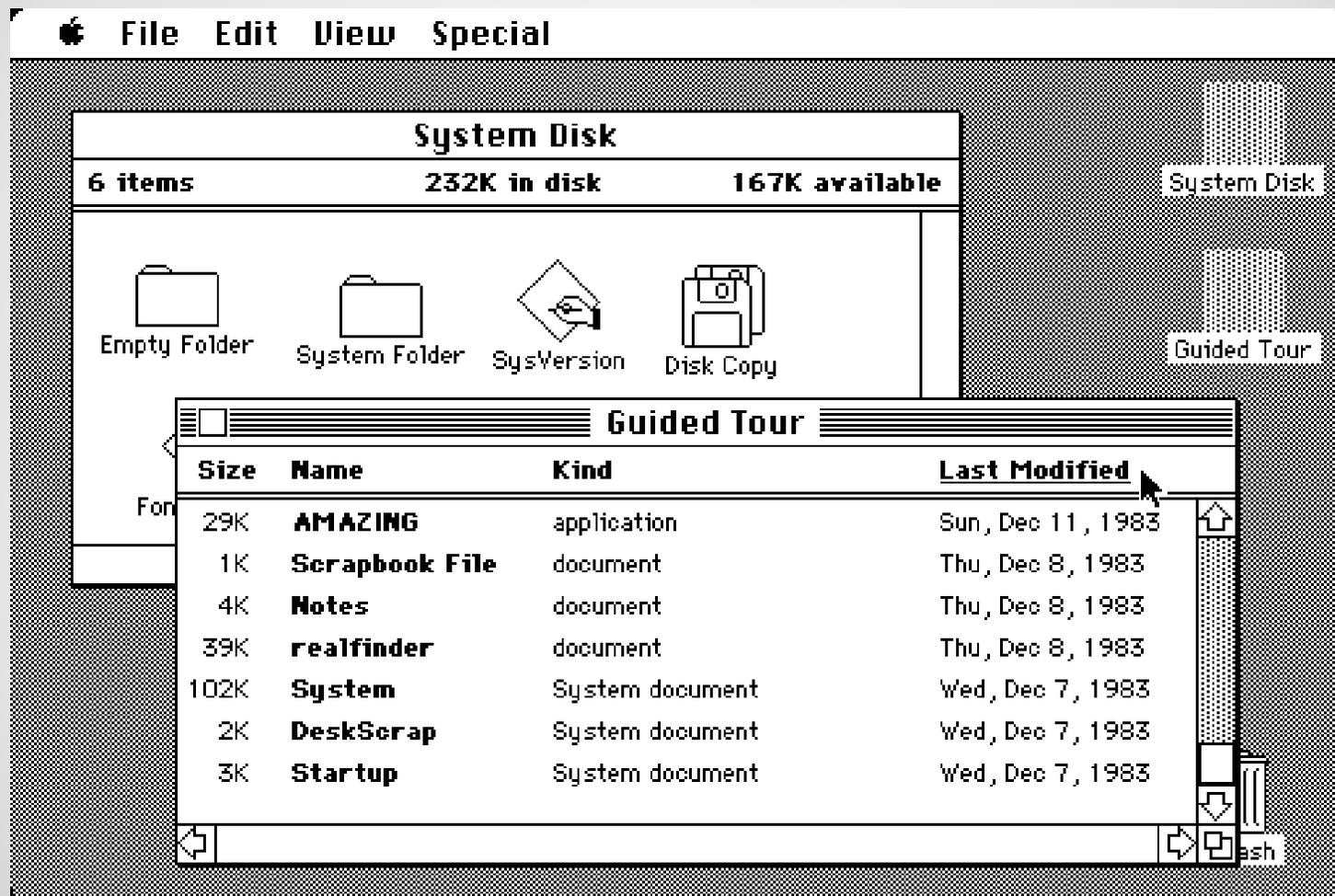
Lisa Screen



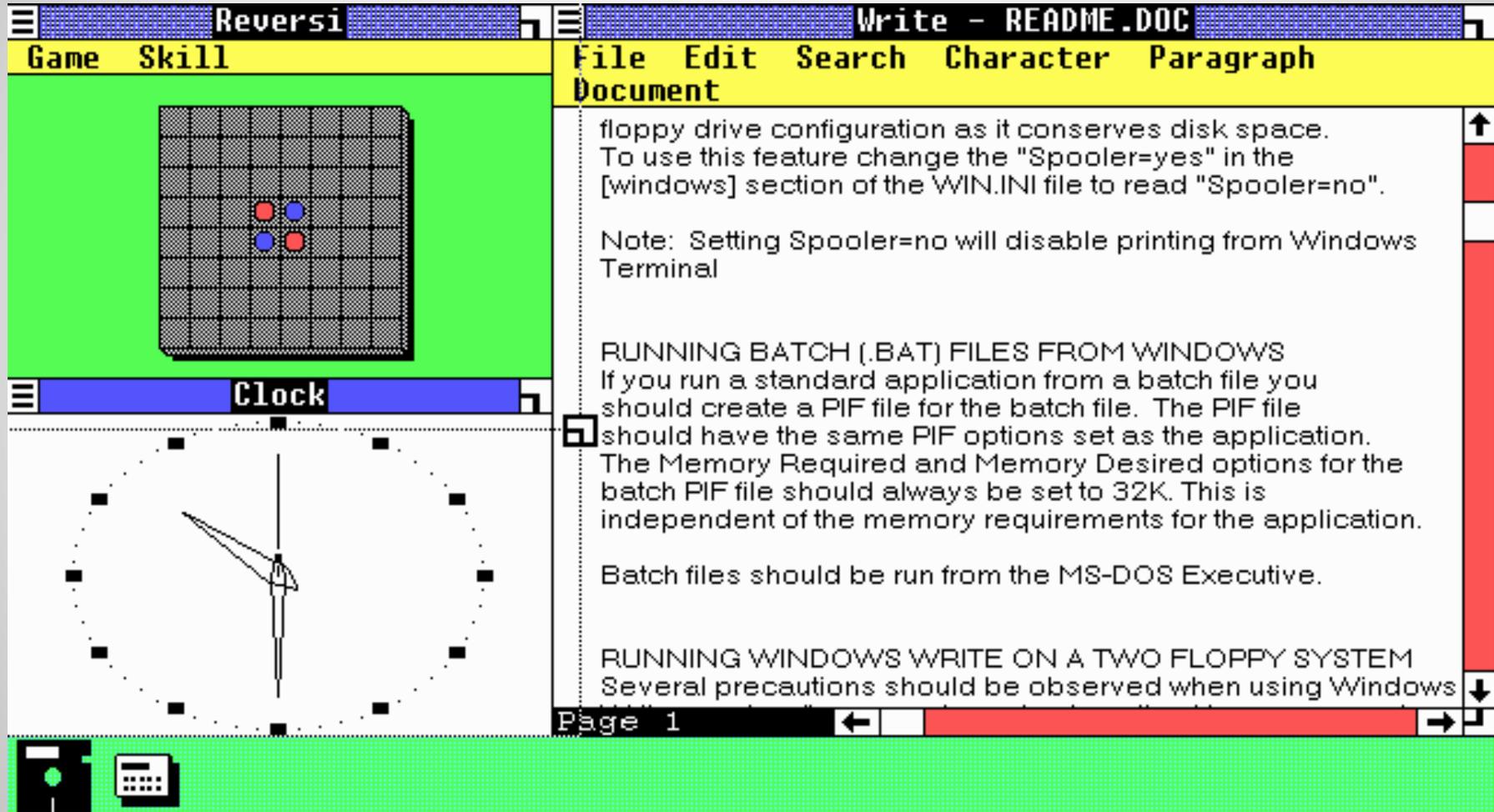
Macintosh (1984)



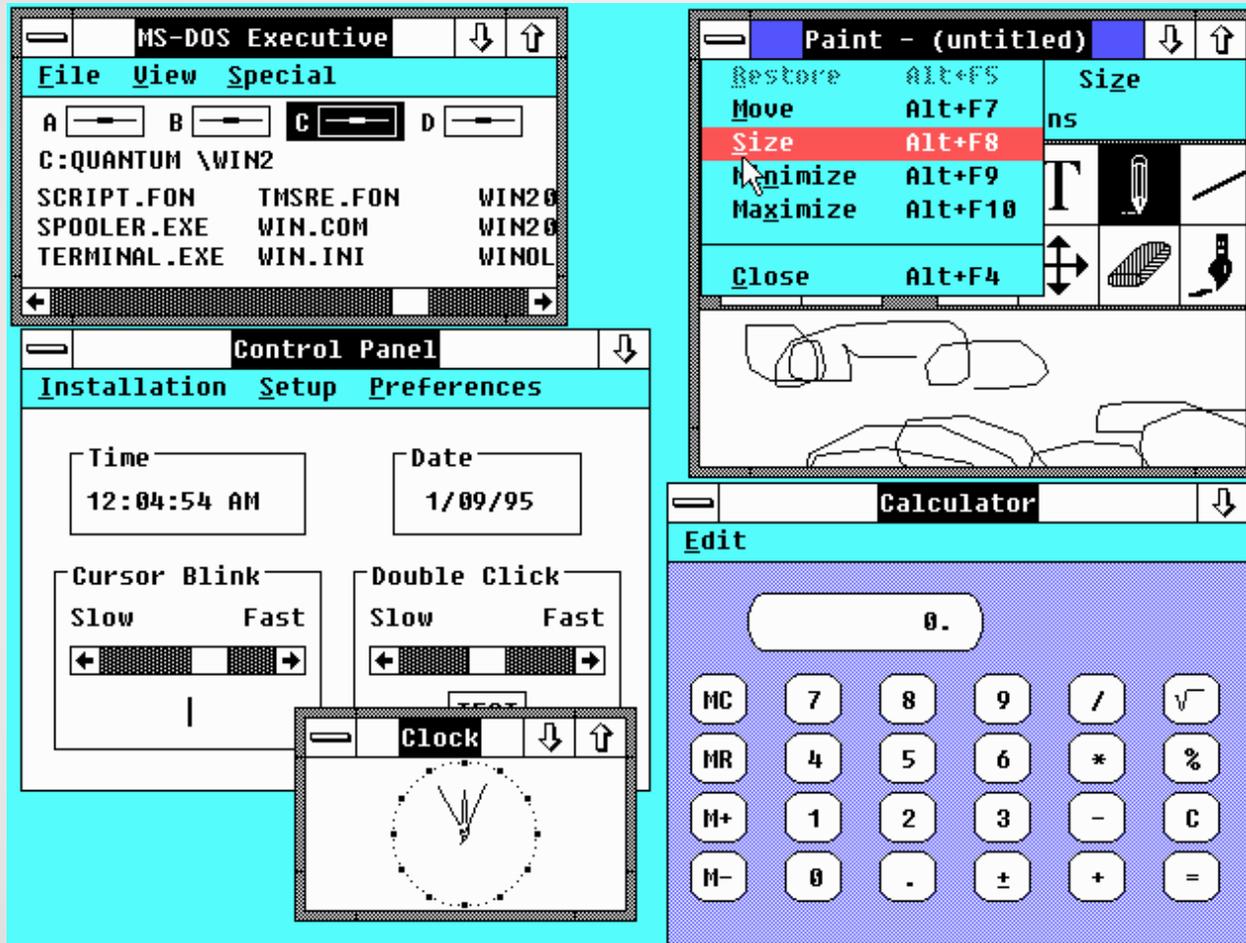
Macintosh Screen



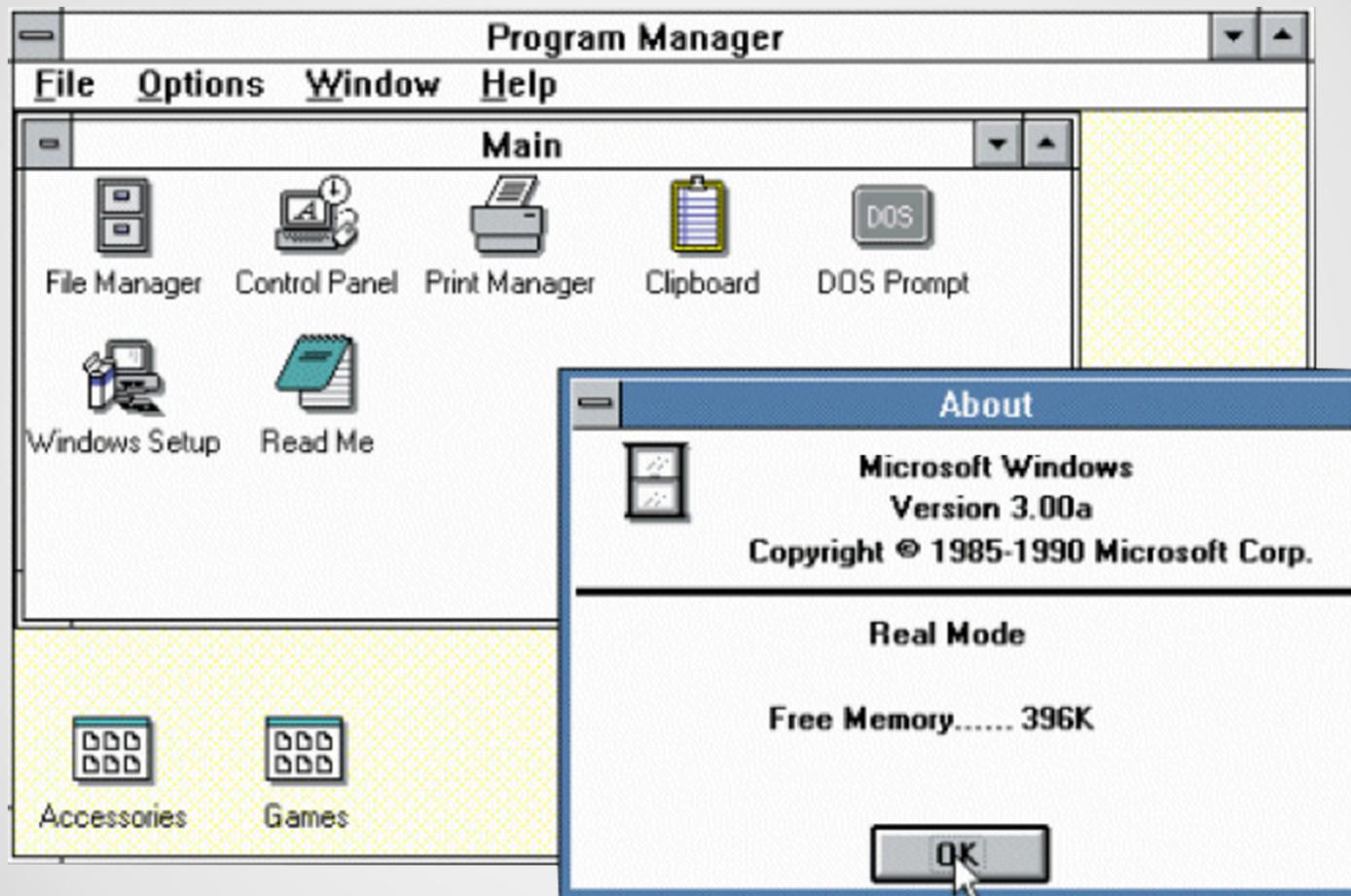
Windows 1.0 (1985)



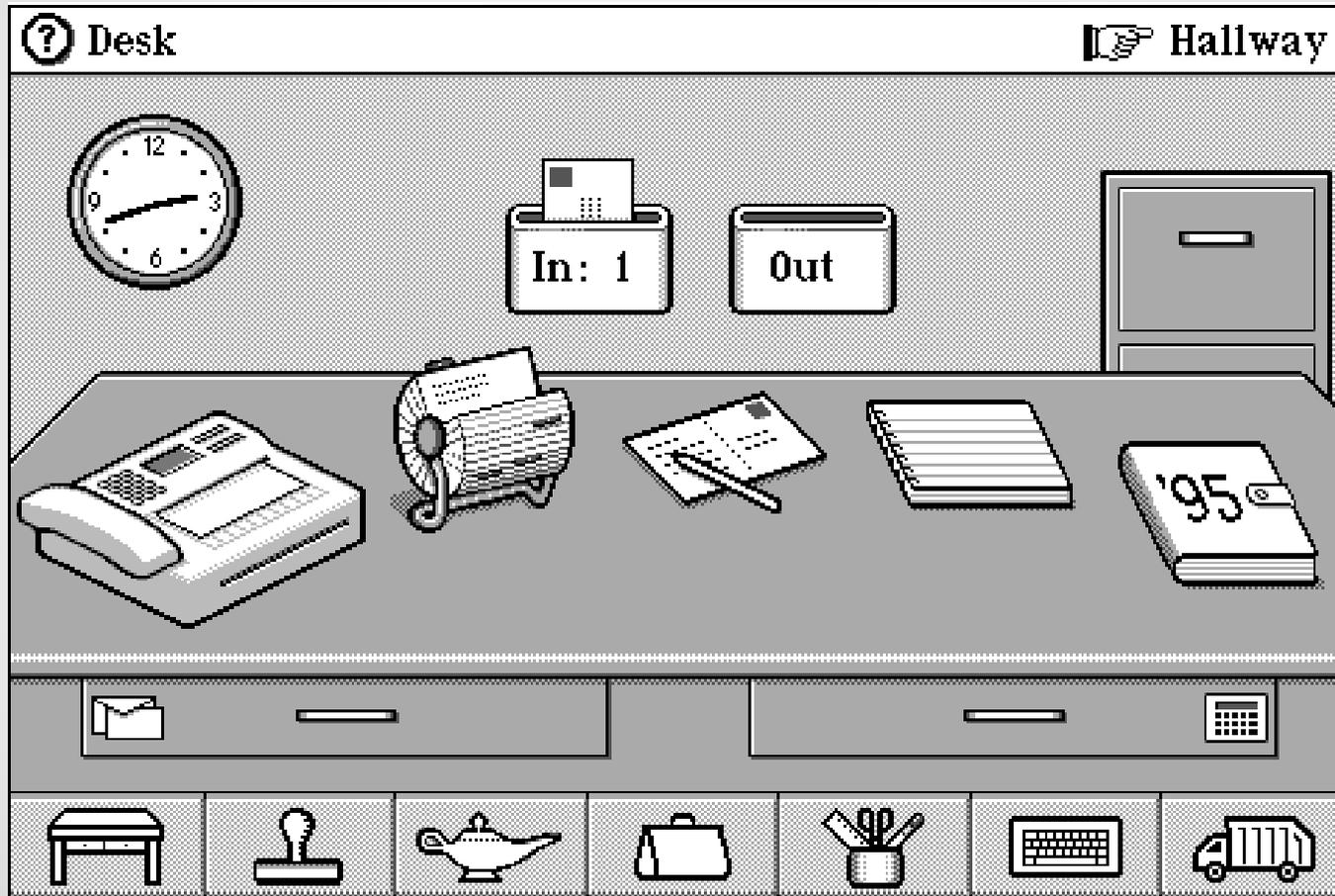
Windows 2.0 (1987)



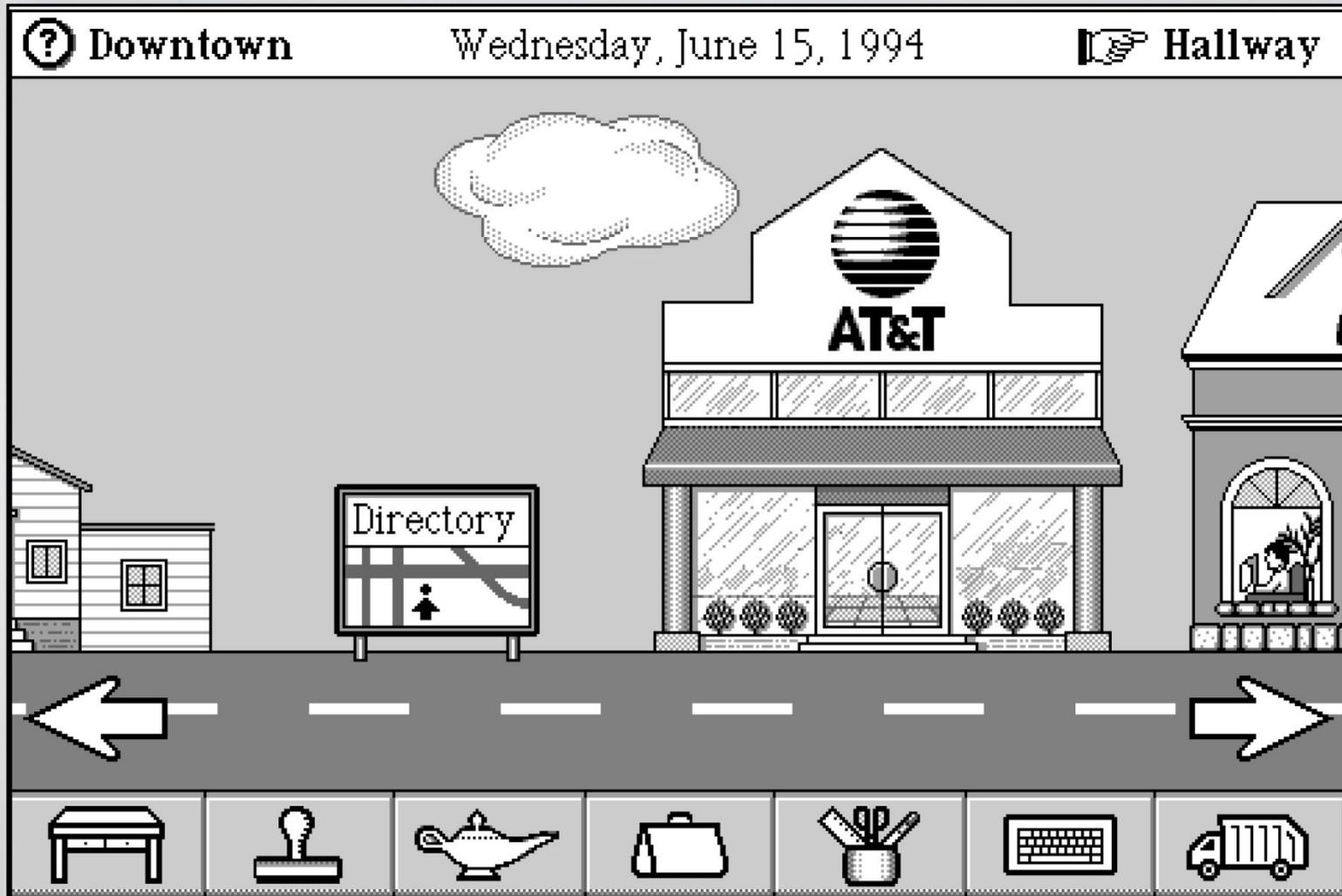
Windows 3.0 (1990)



MagicCap (1994)



MagicCap (1994)



Microsoft Bob (1995)



Problems With Metaphors

They can limit designers' creativity.

They can reduce the advantages of software.

If shallow, inappropriate, or mixed, they can be more misleading than helpful.

They can be overly “cute” at the expense of functionality.

They can be culturally specific.

They can result in slower interactions or performance.

But...

They can make interfaces easier to learn.

Questions?