IE403
Winter 2023
Human-Computer Interaction
Instructor: Dr. Kalyan Sasidhar
(kalyan_sasidhar@daiict.ac.in)
TA:TBA

What is this class about?

- Why are things so hard to use these days?
- Why doesn’t this thing I just bought work?
- Why is this website so hard to use?
- Why are users not liking my design?
- Why is my app not getting popular?

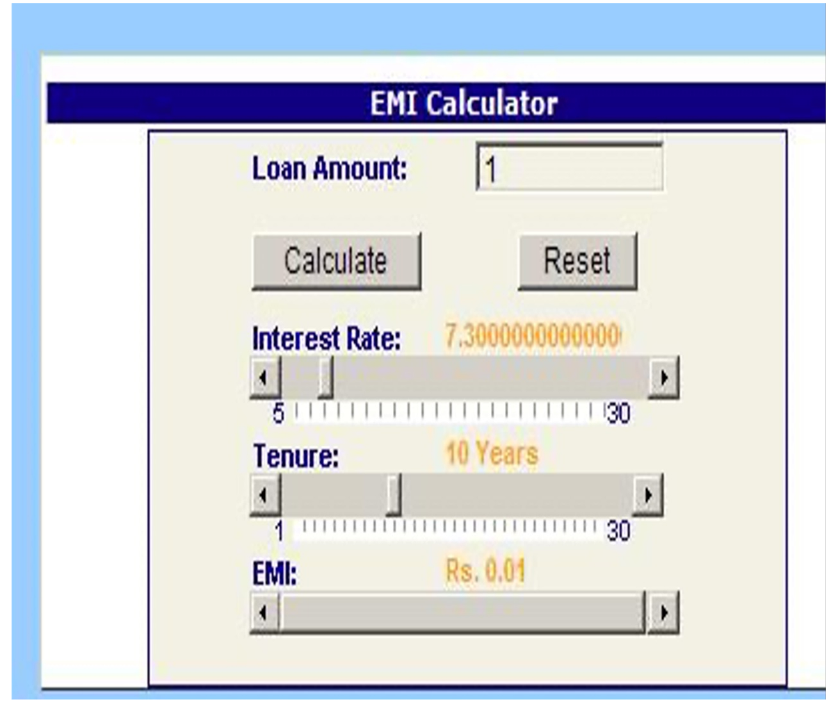
Outcomes

- Understand what makes interfaces more or less usable
- Design and Build usable interfaces
- Scientifically evaluate the usability of those interfaces

Grading Policy

Component	Weightage
Attendance	20%
In-Sem I	15%
In-Sem II	15%
Quizzes	10%
In-class Assignments	10%
Project	30%


HOW GRANDMA SEES THE REMOTE
TV EXPLODES
CAUSE NATIONWIDE BLACKOUT
FORGOT
FORGOT
LOSE SOUND
TIDAL WAVE STARTER
LOSE PICTURE
NO CLUE
NEVER SAW THIS ONE BEFORE
LAUNCH ROCKETSHIP
UTTER MYSTERY
EMIT SPARKS
HOUSE BLOWS UP
OMINOUS SMELL
DROP THE BIG ONE
DZZT


EMI Calculator
Loan Amount:
1
Calculate
Reset
Interest Rate:
7.3000000000000
5
30
Tenure:
10 Years
1
30
EMI:
Rs. 0.01

How can we design human-centered Systems that people find ***useful and usable****?*

This course introduces

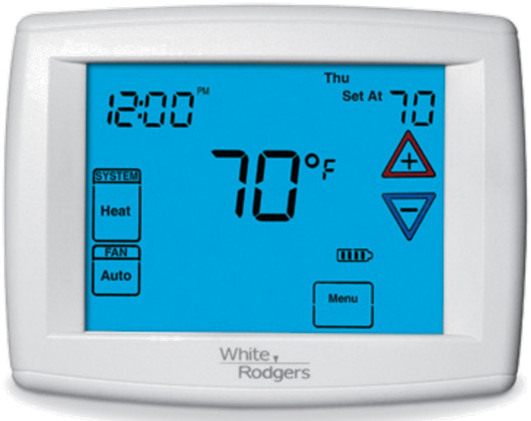
Designing,

Prototyping

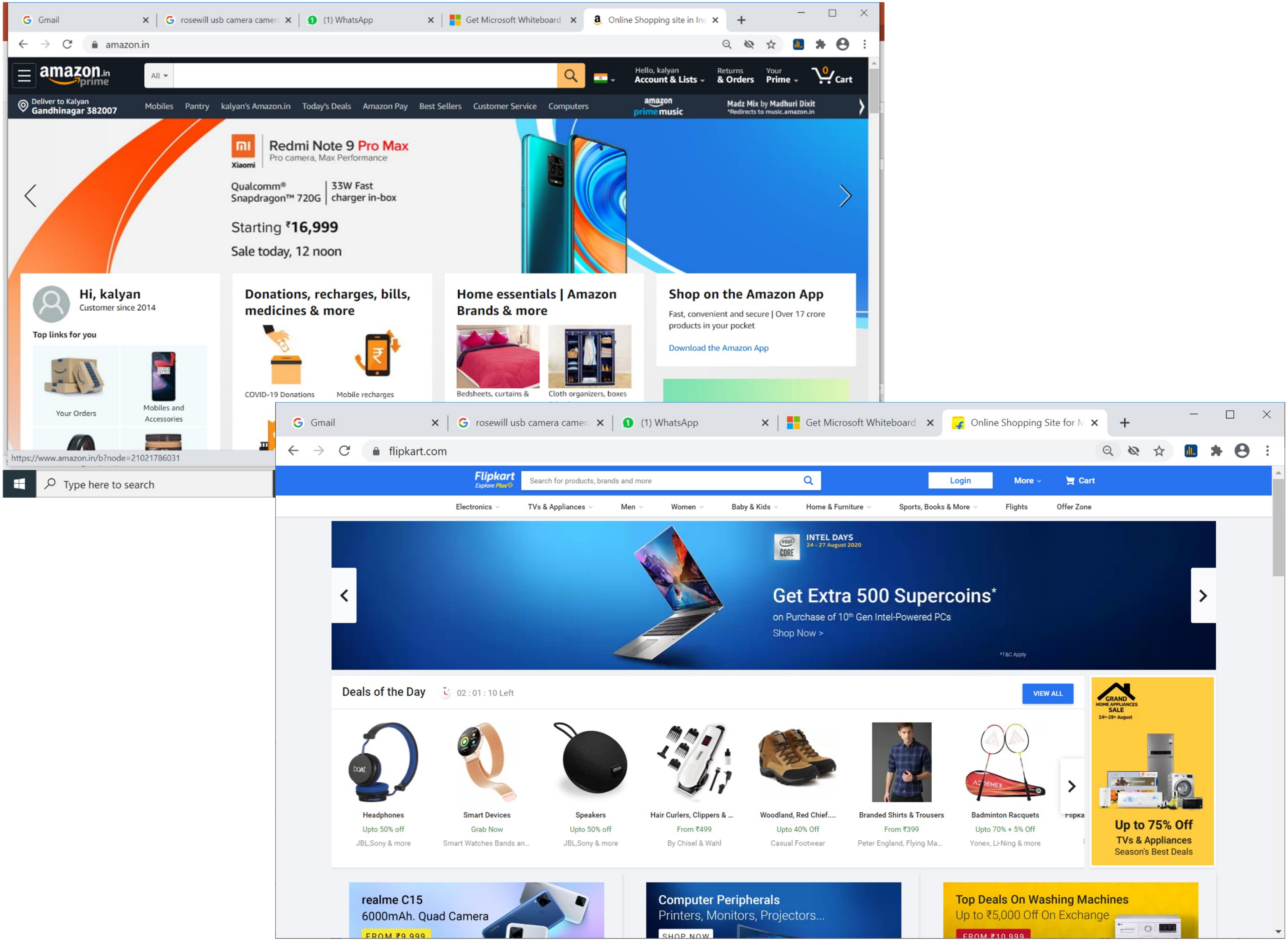
Evaluating

user interfaces.

What do these have in common?

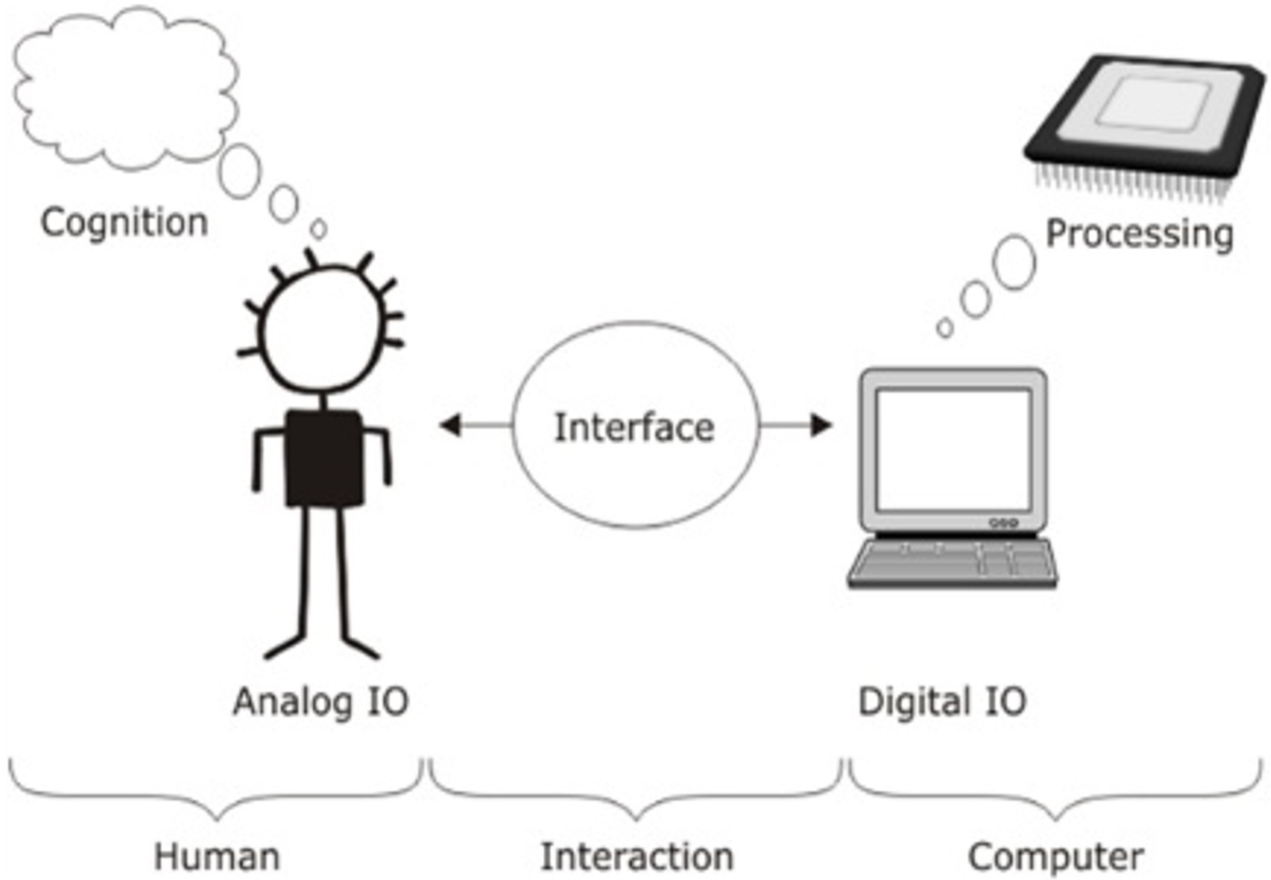


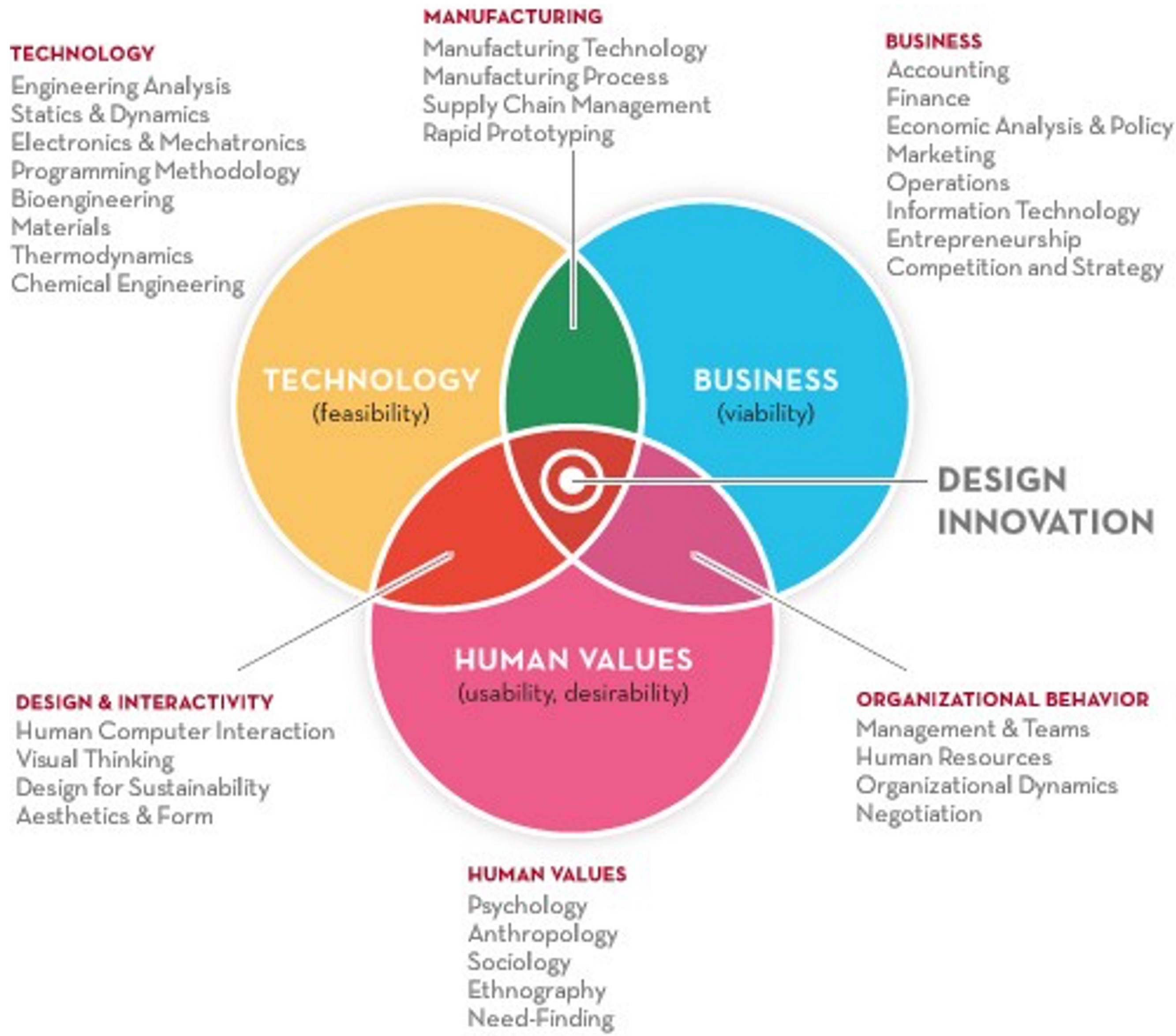
What do these have in common?


Gmail
rosewill usb camera camer
(1) WhatsApp
Get Microsoft Whiteboard
Online Shopping site in Inc
amazon.in
amazon.in prime
All
Hello, kalyan
Account & Lists
Returns
& Orders
Your
Prime
Cart
Deliver to Kalyan
Gandhinagar 382007
Mobiles
Pantry
kalyan's Amazon.in
Today's Deals
Amazon Pay
Best Sellers
Customer Service
Computers
amazon prime music
Madz Mix by Madhuri Dixit
Redmi Note 9 Pro Max
Pro camera, Max Performance
Xiaomi
Qualcomm® Snapdragon™ 720G
33W Fast charger in-box
Starting ₹16,999
Sale today, 12 noon
Hi, kalyan
Customer since 2014
Top links for you
Your Orders
Mobiles and Accessories
Donations, recharges, bills, medicines & more
COVID-19 Donations
Mobile recharges
Home essentials | Amazon Brands & more
Bedsheets, curtains &
Cloth organizers, boxes
Shop on the Amazon App
Fast, convenient and secure | Over 17 crore products in your pocket
Download the Amazon App
https://www.amazon.in/b?node=21021786031
Type here to search
Gmail
rosewill usb camera camer
(1) WhatsApp
Get Microsoft Whiteboard
Online Shopping Site for M
flipkart.com
Flipkart
Explore Plus
Search for products, brands and more
Login
More
Cart
Electronics
TVs & Appliances
Men
Women
Baby & Kids
Home & Furniture
Sports, Books & More
Flights
Offer Zone
INTEL DAYS
24 - 27 August 2020
Get Extra 500 Supercoins*
on Purchase of 10th Gen Intel-Powered PCs
Shop Now >
Deals of the Day
02 : 01 : 10 Left
VIEW ALL
Headphones
Upto 50% off
JBL,Sony & more
Smart Devices
Grab Now
Smart Watches Bands an...
Speakers
Upto 50% off
JBL,Sony & more
Hair Curlers, Clippers & ...
From ₹499
By Chisel & Wahl
Woodland, Red Chief....
Upto 40% Off
Casual Footwear
Branded Shirts & Trousers
From ₹399
Peter England, Flying Ma...
Badminton Racquets
Upto 70% + 5% Off
Yonex, Li-Ning & more
GRAND HOME APPLIANCES SALE
Up to 75% Off
TVs & Appliances
Season's Best Deals
realme C15
6000mAh. Quad Camera
Computer Peripherals
Printers, Monitors, Projectors...
Top Deals On Washing Machines
Up to ₹5,000 Off On Exchange

What is HCI?

HCI is the study of interaction between people (users) and computers




MANUFACTURING
Manufacturing Technology
Manufacturing Process
Supply Chain Management
Rapid Prototyping
TECHNOLOGY
Engineering Analysis
Statics & Dynamics
Electronics & Mechatronics
Programming Methodology
Bioengineering
Materials
Thermodynamics
Chemical Engineering
BUSINESS
Accounting
Finance
Economic Analysis & Policy
Marketing
Operations
Information Technology
Entrepreneurship
Competition and Strategy
TECHNOLOGY
(feasibility)
BUSINESS
(viability)
DESIGN
INNOVATION
HUMAN VALUES
(usability, desirability)
DESIGN & INTERACTIVITY
Human Computer Interaction
Visual Thinking
Design for Sustainability
Aesthetics & Form
ORGANIZATIONAL BEHAVIOR
Management & Teams
Human Resources
Organizational Dynamics
Negotiation
HUMAN VALUES
Psychology
Anthropology
Sociology
Ethnography
Need-Finding

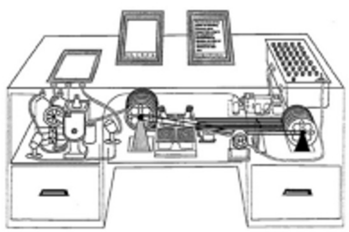
HCI Evolution

- **50s** - Interface at the hardware level for engineers - switch panels
- **60-70s** - interface at the programming level - COBOL, FORTRAN
- **70-90s** - Interface at the terminal level - command languages
- **80s** - Interface at the interaction dialogue level - GUIs, multimedia (http://www.cs.cmu.edu/~amulet/papers/uihistory.tr.html)
- **90s** - Interface at the work setting - networked systems, groupware
- **00s** - Interface becomes pervasive
  - RF tags, Bluetooth, mobile devices, consumer electronics, interactive screens, embedded technology
- 10s -Interface disappears
  - Focus on tasks, experiences, emotions, social connections, beauty

HCI Evolution

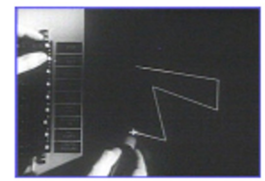
Memex - Vannevar Bush (1945)

Vision for a desktop information management system
Electromechanical system
Seen as the ancestor of the notion of hypertext



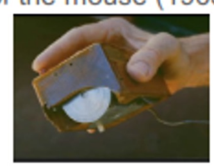
Sketchpad - Ivan Sutherland (1963)

Direct manipulation geometric shapes
Geometric constraints, zoom, click-drag

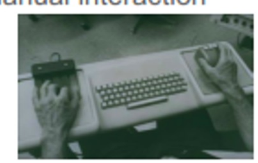


NLS / Augment - Douglas Engelbart (1968)

Inventor of the mouse (1963)



Bimanual interaction

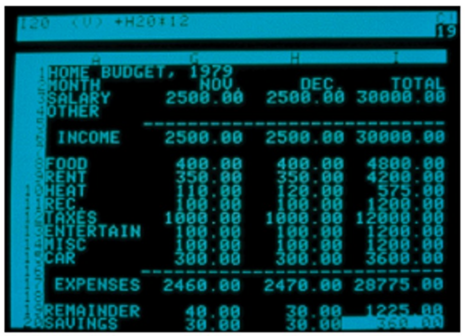


Hypertext, cooperative work,
document sharing, video-conferencing

History

Visicalc - Dan Bricklin (1979)

First spreadsheet (Apple][)

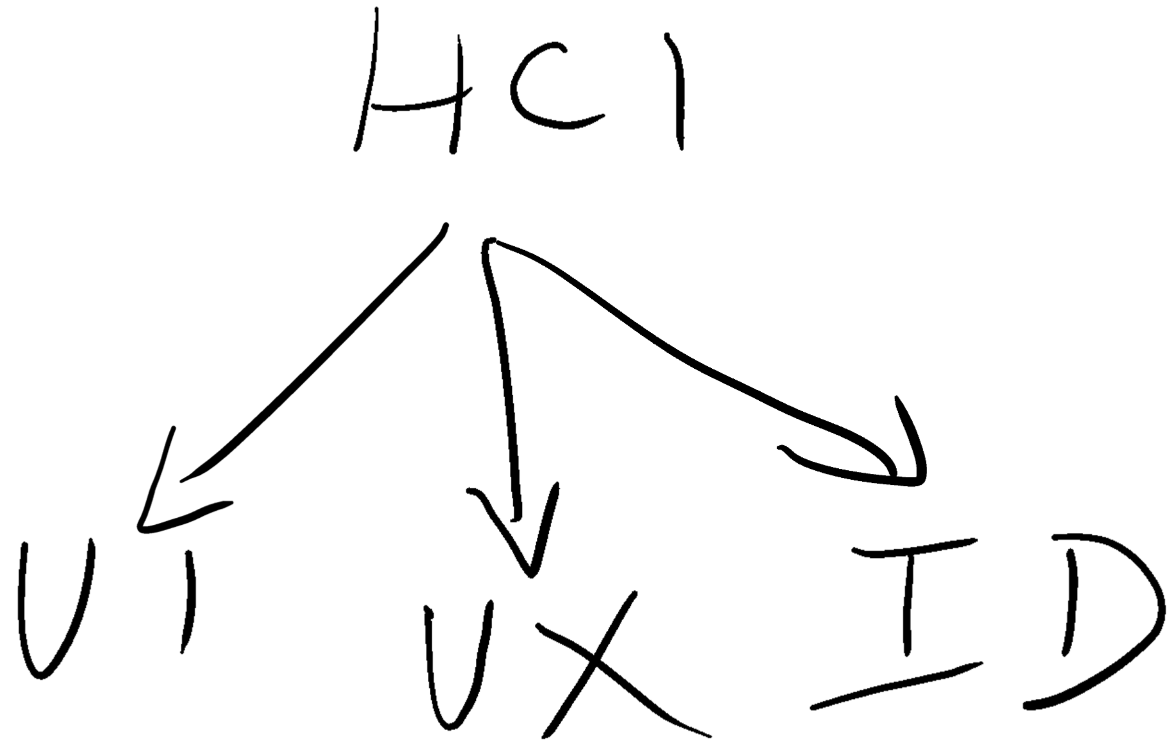


Xerox Star - Xerox PARC (1981)

First commercial graphical workstation
Document-centric approach

Macintosh - Apple (1984)

- Graphical personal computer
  - Finder
  - MacPaint
  - MacWrite
- Hardware + software design



UI (User Interface) refers to the graphical layout and visual appearance of a software application. It includes the buttons, icons, colors, and overall look and feel of the application.

UX (User Experience) refers to the overall experience of a user when interacting with a software application, product or service. It encompasses a wide range of elements such as usability, accessibility, and emotions evoked by the interaction.

ID (Interaction Design) is a discipline concerned with the design of interactive products and services, with the goal of creating seamless, effective and satisfying experiences for users. Interaction design is often used synonymously with UX design.

HCI

- Human
  - The end-user of a program
  - Other people in the organization
  - The surrounding cultural context
- Computer
  - The hardware and the software
  - Microwaves, mobile phones, cars
- Interaction
  - The user tells the computer what they want
  - The computer communicates results

HCI is made up of..

- Theories – learn and apply
- Models – create and use
- Methods – master and apply
- Guidelines – learn and use
- Principles – understand and apply
- Techniques – master and use

Design Goals of the class

- Everything designed has some explicit criteria
  - Attractive
  - Cost effective to manufacture
  - Durable
  - Water tight
  - Can hold hot water effectively
  - Separate tea leaves from rest

But is it usable?

And is it useful to people?

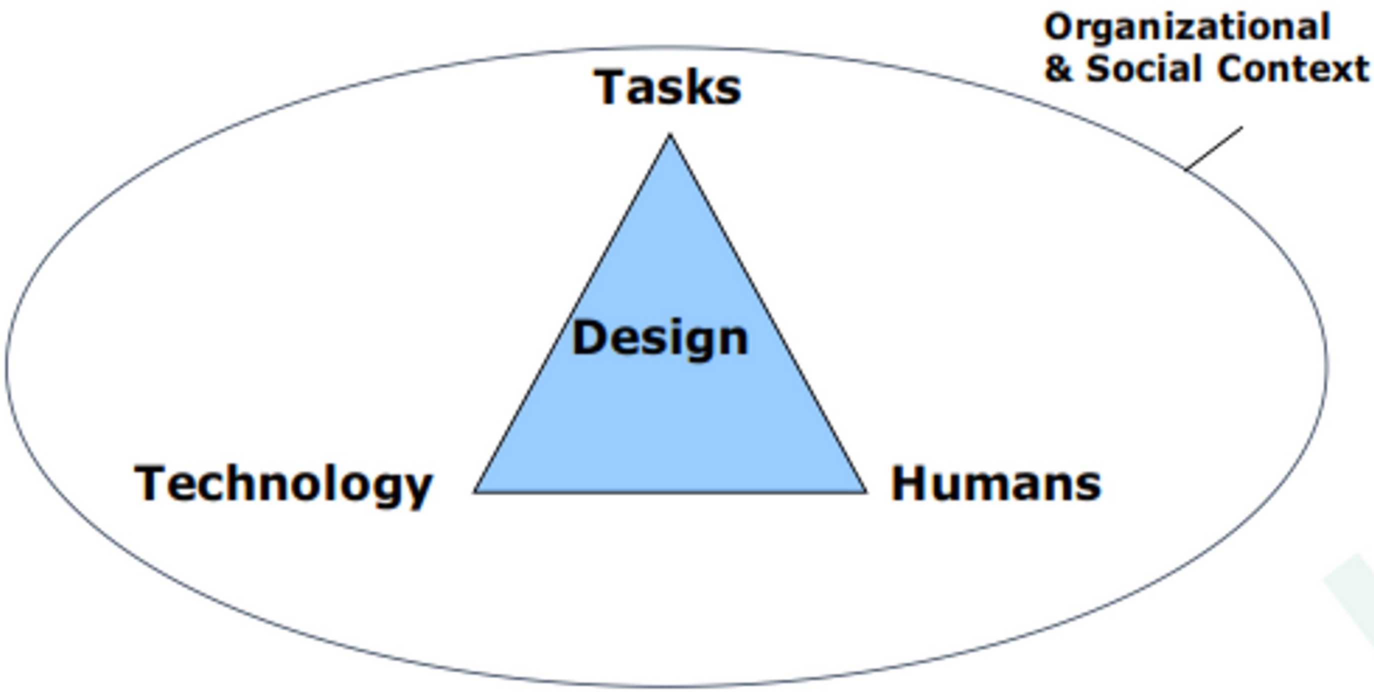
Design and Goals of this class

- Everything designed is done so within a specific context
- Training required
- Conventions
- Laws and regulation
- Competing products
- Human abilities and limitations

Understanding these influencing factors also a key part of this course

Goals of HCI

- Allow users to carry out tasks
  - Safely
  - Effectively
  - Efficiently
  - Enjoyably



- HCI draws computer science, psychology, and design together
- Main focus is on the people using system

To make better Interactive technology…We need to

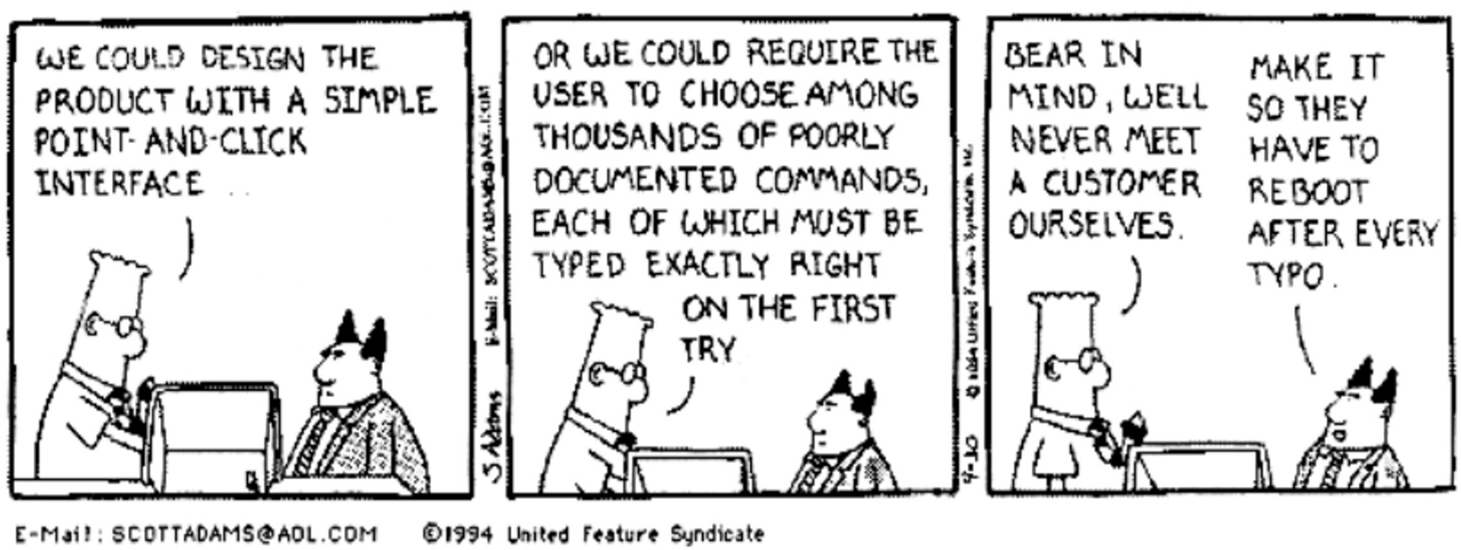
- Know about how people **interact with things**
- Know about what people **can and can’t do**
- Know about the **situations in which people**
- do things
- Know about the **basics of good design**
- Understand people’s **goals**

HCI is changing…

- Physical things
- GUI interfaces
- Collaborative interfaces
- Internet technologies
- Social technologies
- Ubiquitous technologies
- What next??

System Centered Design

- What can I easily build on this platform?
- What can I create from the available tools?
- What do I as a programmer find interesting?



User Centered Design

Design is based upon a user’s

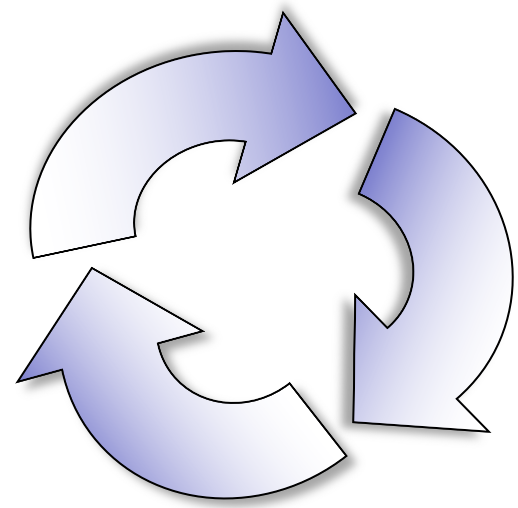
- abilities and real needs
- context
- work
- tasks
- values

User Centered Design

- Focus on the people who will use the system, on their preferences and their requirements
- Building models of the users, tasks and systems
- Iterative process
- Prototyping and Evaluation by users

UI Design/Develop Process

- Analyze user's goals & tasks
- Create design alternatives
- Evaluate options
- Implement prototype
- Test
- Refine



Two Fatal Mistakes

- Assume all users are alike
- Assume all users are like the designer

But Who are the Users?

How do we improve interfaces?

1. Educate software professionals
2. Draw upon fast accumulating body of knowledge regarding H-C interface design
3. Integrate UI design methods & techniques into standard software development methodologies

12:00 PM
Thu
Set At 70
70°F
SYSTEM
Heat
FAN
Auto
Menu
White Rodgers

Honeywell
°C
10
15
20
25

nest
67
IN 25 MIN
72

Group Project

Semester-long team effort

Group Project

- Design and evaluate an interface
  - 0 - Team formation & topic choice
  - 1 - Understand the problem space
  - 2 - Exploring the design space
  - 3 - Prototype
  - 4 - Evaluation

- Main 4 parts worth ~10% each
- Presentation, documentation ~ 10%

Group Project Details

- Part 0 - Topic definition
  - Identify team & general topic
  - Create web notebook
  - Suggestion: Pick a population and pick a technology; check out intersection
- Part 1 - Understanding the problem
  - Describe tasks, users, environment, social context
  - What are implications for design?

Group Project Details

- Part 2 - Design alternatives
  – Storyboards, mock-ups for multiple different designs
  – Explore, push boundaries of design space
  – Explain decisions
- Part 3 - System prototype & eval plan
  – More detailed prototype (semi-working ok)
  – Plan for conducting evaluation

Group Project Details

- Part 4 - Evaluation
  - Conduct formal evaluation with example users
  - Use appropriate methods
  - Analyze results of evaluation
  - Characterize what's working and what's not

Project Teams

- K people
  - You decide (or I will!)
  - Diverse is best!
  - Consider schedules
  - Create a web space:
    - Immediately post ideas for general topics, populations, technologies, etc.

- Cool project and team name

Project Topics

Semester theme: “Innovative and Usable Interfaces in Everyday Life”

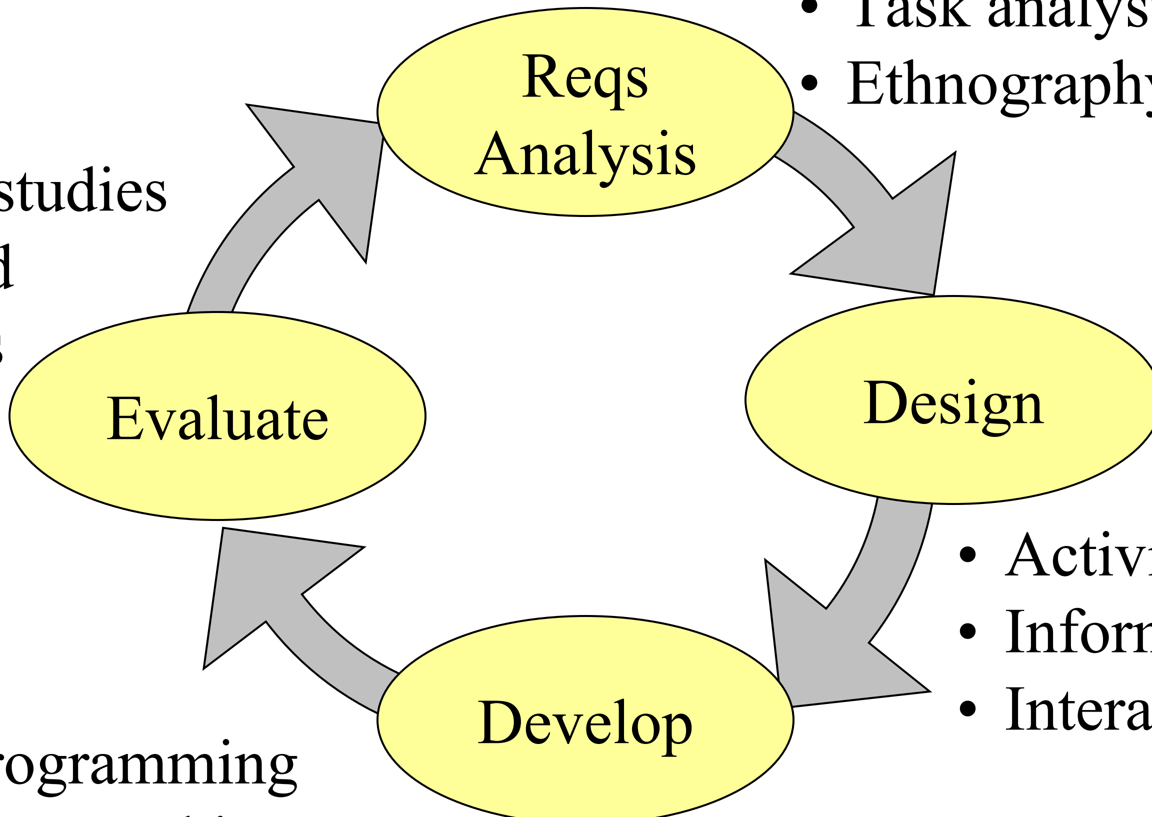
What Makes a Good Project

- Typically:
  - Access to domain experts & users
  - “Real” clients
  - Interesting human issues
  - Rich domain for design

Design Evaluation

- "Looks good to me" isn't good enough!
- Both subjective and objective metrics
- Some things we can measure
  - Time to learn
  - Speed of performance
  - Rate of errors by user
  - Retention over time
  - Subjective satisfaction

The Learning Cycle



- Task analysis
- Ethnography

- Activity design
- Information design
- Interaction design

- GUI programming
- Widgets, graphics, animation

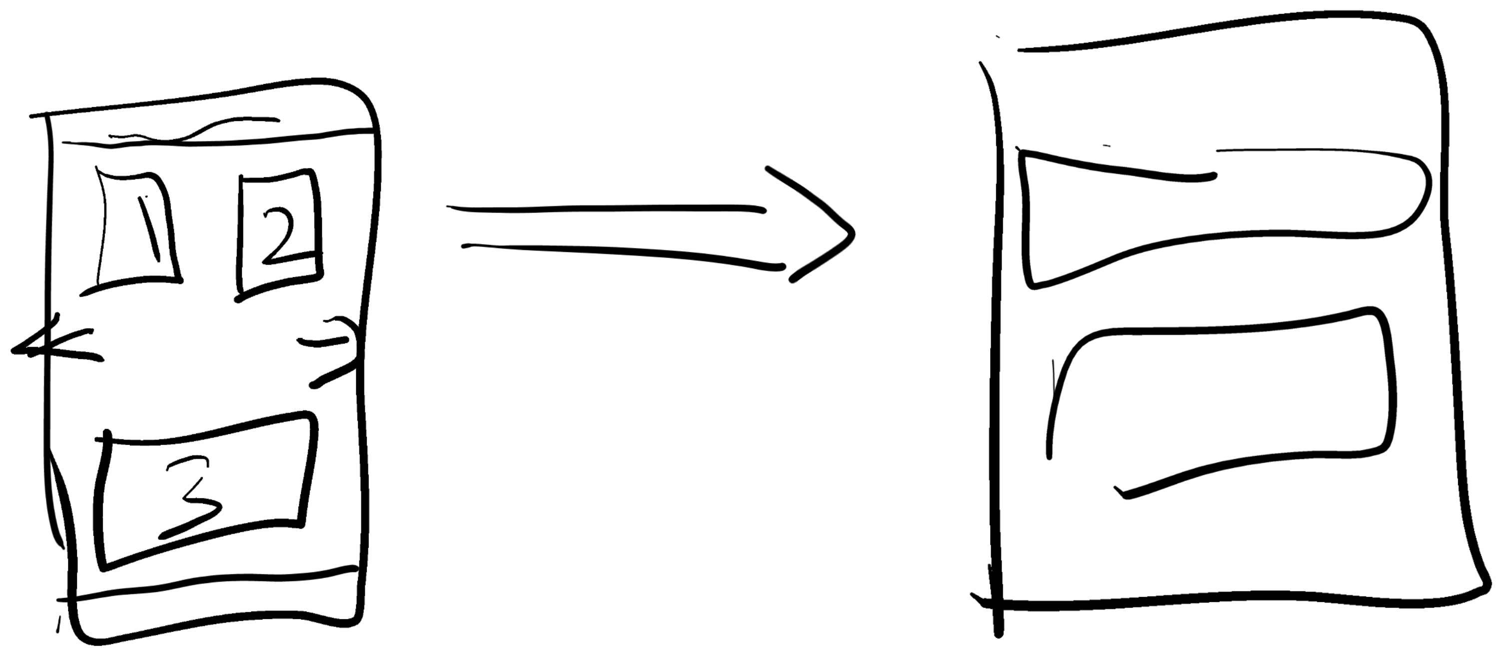
- Usability studies
- Controlled experiments

Case Study sample topics

- Virtual keyboards
- Existing websites and their design patterns, usage behavior, commonalities and differences, improvements
- Mobile applications
- Technology for Senior citizens
  - Usage behavior
  - Problems, difficulties,
  - Usability factors
  - Improvements
- Wearable interfaces
  - smartwatches

Case Study Method

- Objective
- Design Process Followed
  - Requirement Gathering
  - Problem Identification
  - Requirement Specification
- Concept Generation
  - Synthesis
- Prototype Development
- Documentation


1
2
3

IE 403-Winter 2023

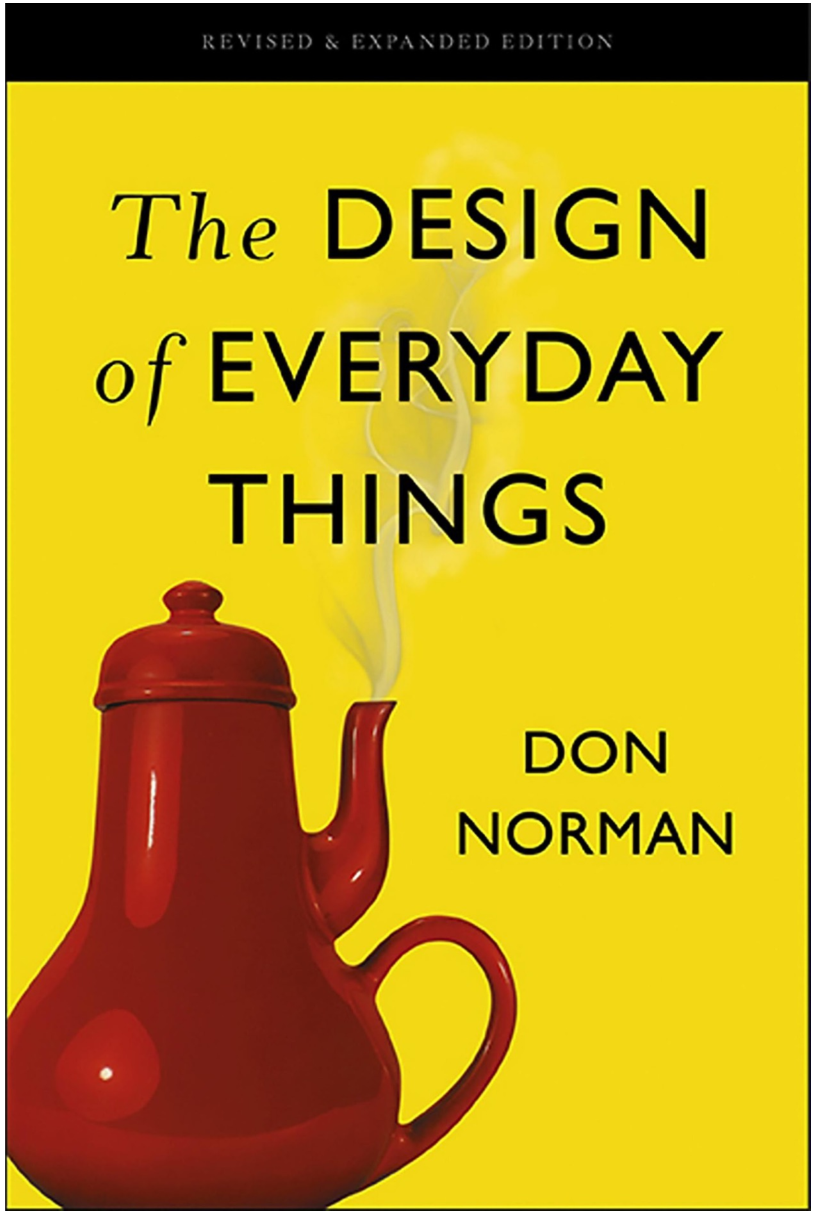
Human-Computer Interaction

Week 2-Lec1

Topics for this week’s classes

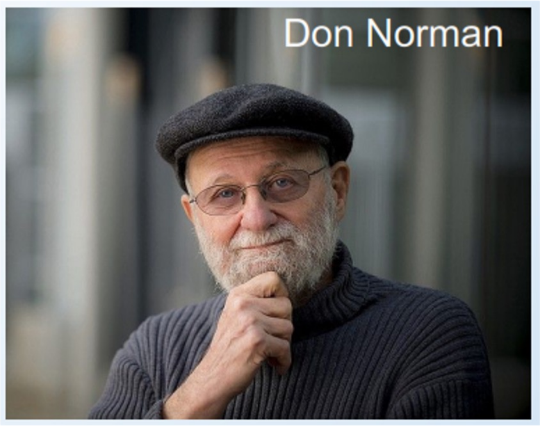
- Discussion on Don Norman’s principles

- Conceptual Models



Overview

- Why are some everyday things difficult to understand and use?
- What are Don Norman's principles and how do they apply to the design of everyday things?
- How can we apply Norman's principles to the design of computer interfaces?



Psychopathology of Everyday Things

- We are surrounded by many everyday things that have poor *usability*
  - Programming a VCR
  - Telephone features we can’t remember how to use
    - How to change the remote access code?
  - Photocopiers and fax machines
    - Face down or face up?
- Many of these things can be difficult to interpret and frustrating to use if they provide no clues or false clues as to how they operate

Norman’s Principles of Design

- Make things visible
- Provide a good conceptual model
  - Affordance
  - Mapping
  - Constraints
  - Feedback

Norman’s philosophy

“…things are designed to be used by people, and without a deep understanding of people, the designs are apt to be faulty, difficult to use, difficult to understand.”

Understanding

- *What does it all mean?*
- *How is the product supposed to be used?*
- *What do all the different controls and settings mean?*

Many products defy understanding simply because they have too many functions and controls.

User/Human Centered Design

Human-centered design (HCD), is an approach that puts human needs, capabilities, and behavior first, then designs to accommodate those needs, capabilities, and ways of behaving.

- Empathize: Understanding YOUR USER and design
  - Human needs, capabilities and behaviour
- Communication: Indicate what actions are possible, what is happening, and what is about to happen.

What is a good User Centered design?

- Has affordances - makes each operation visible;
- Offers obvious mappings - makes the relationship between the actual action of the device and the action of the user obvious;
- Provides feedback on the user's action;
- Provides a good mental model of the underlying behaviour of the device;
- Provides forcing functions - prevents a user from making bad errors;
- Supports automatic learning -- offers consistencies and practice that help the user acquire interface skills.

ma-af-fe-fo-meau

Fundamental Principles of Interaction

Discoverability

Is it possible to even figure out what actions are possible and where and how to perform them?

- Discoverability results by applying 6 fundamental psychological concepts

Affordances

(coined by psychologist James J. Gibson and applied to HCI by Don Norman)

- Affordances determine what actions are possible with an object by an agent(user) Relationship
- Exist even if they are not visible.
- For designers, their visibility is critical
  - visible affordances provide strong clues to the operations of things
- Perceived affordances
  - *Can I figure out what actions are possible ?*
  - If NO,
    - Presence of some SIGNALS is required.

Affordance property
Visible, Discoverable

Glass: AntiAffordance property
Invisible, Not Discoverable

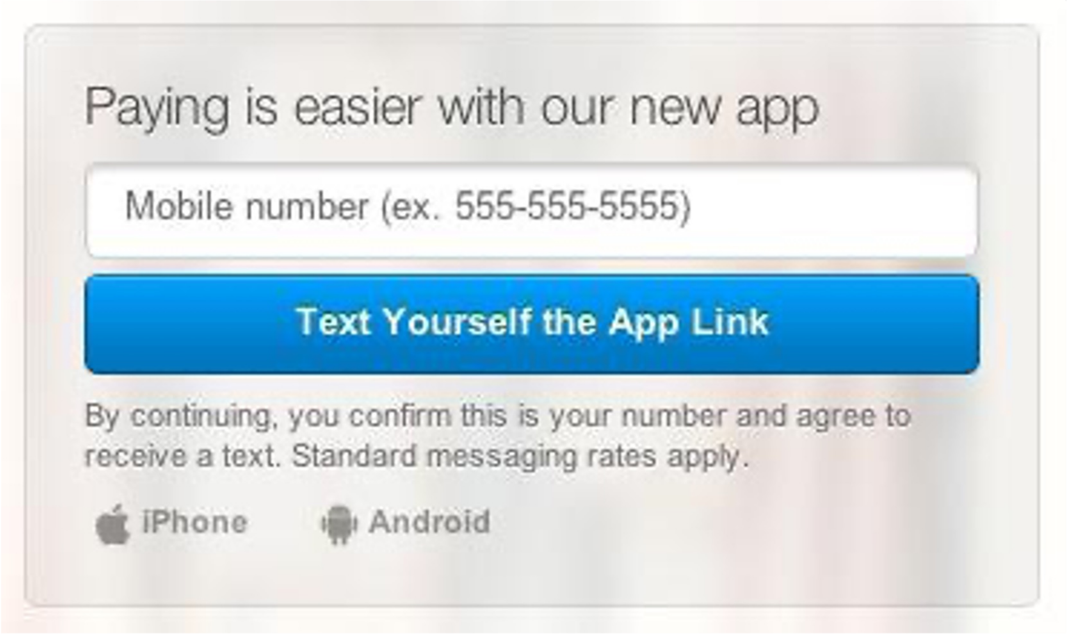
Affordances in Web Interfaces

- What actions does the webpage provide to the user explicitly?

Or

- What does the interface afford to the user?

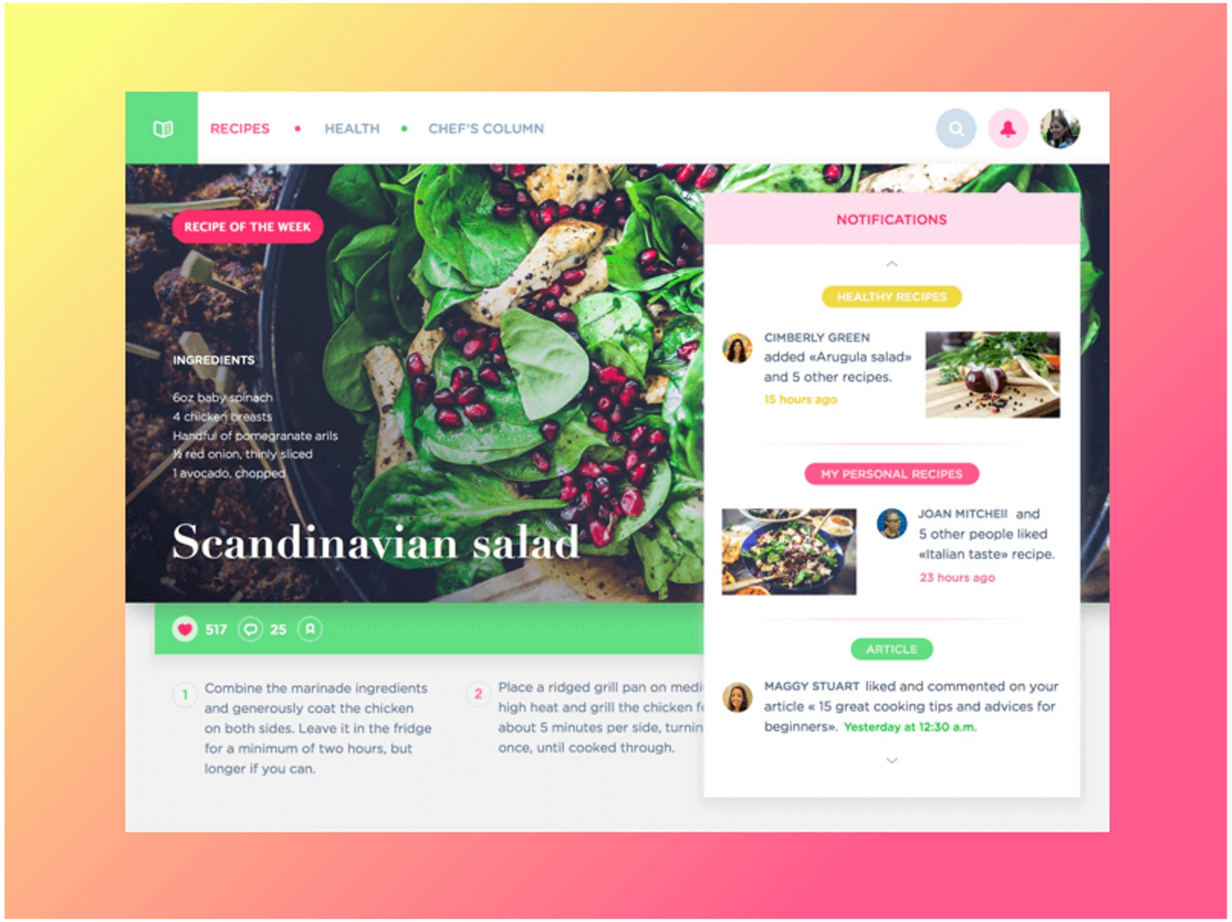
Affordances in Apps



- What Explicit affordance does the interface provide?
  - The raised appearance of the button affords pushing down. The text explicitly signals the result of the action.

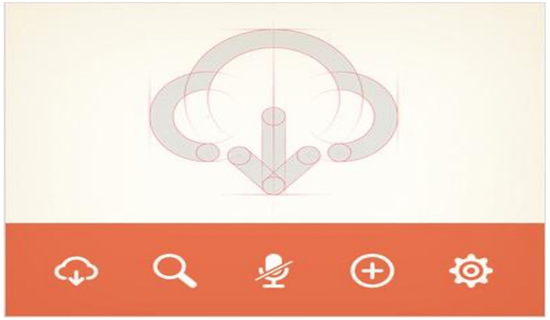
Other types of Affordances

Pattern



Hidden

Metaphorical
(Contextual)



Signifiers

- Communicate where the action should take place
- Signifiers refers to *any mark of sounds, any perceivable indicator that communicates appropriate behaviour to a person*
- Signifiers are signals.
  - signs, labels, and drawings placed in the world

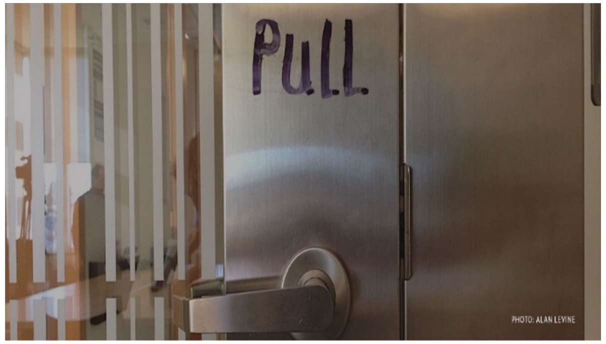


Fig 1. If a door needs a signifier, then it's a bad design

Fig 2. The arrows are signifiers:
- Swiping left or right brings up new restaurant recommendations.
- Swiping up reveals the menu for the restaurant being displayed
- Swiping down, friends who recommend the restaurant

Mapping

- The relationship between the elements of two sets of things
  - Controls and Displays
- Groupings and proximity are map controls to function
  - Controls should be close to the item being controlled.

Fig 1

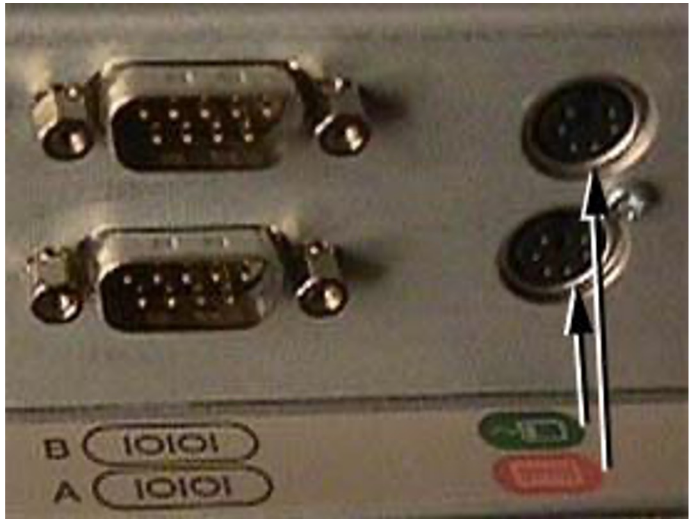


Fig 2

What Knob Goes Where?

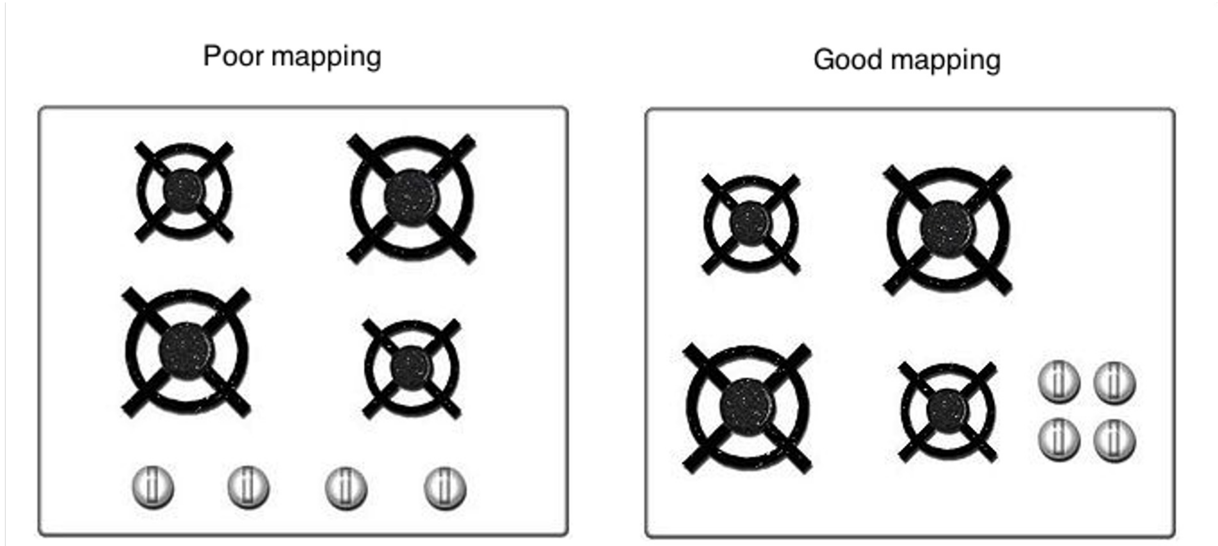


Fig 1

Best mapping: Controls are mounted directly on the item to be controlled.
Second-best mapping: Controls are as close as possible to the object to be controlled.
Third-best mapping: Controls are arranged in the same spatial con- figuration as the objects to be controlled

Exploiting Natural Mapping

Fig 2

Feedback

- Feedback — communicating the results of an action
  - *must be immediate*
  - *must be informative*
  - *Poor feedback can be worse than no feedback at all*

Fig 1

Poor/No Feedback: Pressing elevator buttons repeatedly. Reason: No feedback of lift movement?

Norman’s Principles in Software

- Visibility
  - Visibility of the tasks the interface supports : Can I see it?
  - Communication of system state / mode
- Affordance
  - 

    □ Pressed
  - Here □ Click (web)
- Mapping
  - Where am I and where can I go?
  - Clicking on a particular interface element produces expected effect (under F)ile should be O)pen)

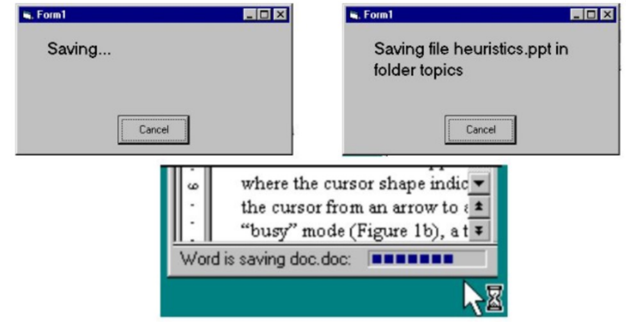


Fig 1

Fig 2

Norman's Principles in Software

- Constraints
  - Why can't I do that?
  - Constraining search criteria, graying out menu items that don't apply in a particular context
- Feedback
  - What is it doing now?
  - Providing clear and immediate feedback for each user action

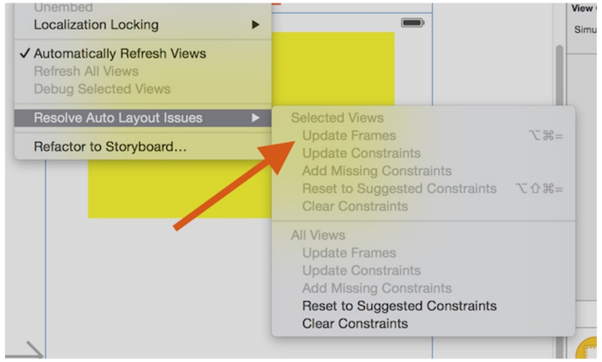


Fig 1

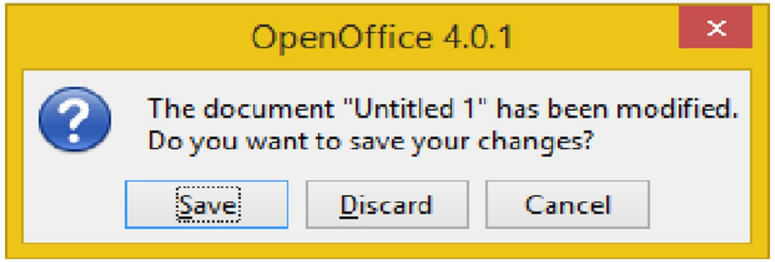


Fig 2

Assignment

I. Major differences between Desktop and Mobile environments
II. Take one real world website or app and start finding affordances, signifiers, mapping and constraints

RECALL OUR
DESIGN CONCEPTS:

the basics:
(elements of these in many of the others)

- affordance
- signifiers
- mapping
- constraints
- feedback

other concepts:

- findability
- transfer effects
- cultural associations
- individual differences

→ all inform a user' mental model

THE DESIGNER'S MODEL, THE USER'S MODEL, AND THE SYSTEM IMAGE.

the **user** also has a **mental model**.
they don't necessarily match.

(No connection)

CONCEPTUAL MODELS & CONCEPTUAL DESIGN:

- **conceptual models** describe how an interactive system is organized
    - the **user** also has a **mental model**. they don't necessarily match.
- conceptual model = the **foundation** of the interface.
    - different user interfaces could be built upon it
    - there are **many ways to represent** a conceptual model
- **goal of conceptual design**, how do conceptual models fit?
- **interface design translates the CM** into things we can see and interact with. It involves design choices, but must stay faithful to the concepts and terminology of the CM.

WHAT DOES A CONCEPTUAL MODEL LOOK LIKE?

however best helps you describe and understand its components:

- lists and tables
- diagrams
- storyboards and sketches
- written descriptions
- mood boards
- physical ‘sketches’

different methods might capture different parts of more effectively than others

➔ you’ll likely use a combination of more than one!

A CONCEPTUAL MODEL CAN INCLUDE:

co-re-ma-te-fa-ac

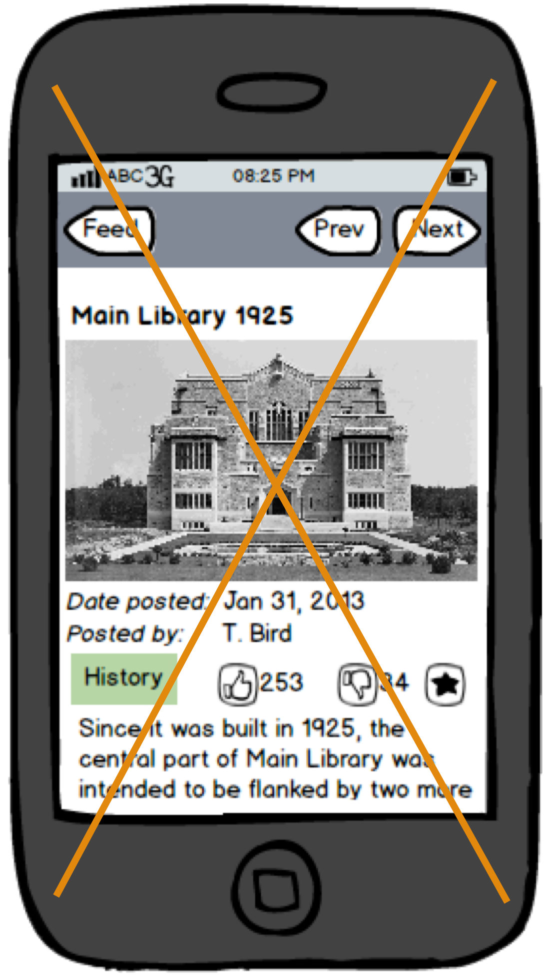
- any central design **metaphors** and analogies
  e.g. the "desktop metaphor"
- **concepts** – objects, actions you can do to them; user roles; attributes of both.
  e.g., files and folders; both can be opened, have names;
- **relationships** among concepts
  e.g., files are *contained* in folders
- **mappings** from concepts to the user experience envisioned**;**
  e.g., the users can *browse* files, and *mark favorites*
- **terminology** that will be used (consistently) to tie it all together
- **interaction** types; how will they interact with it?
  e.g. give commands, perform operations, explore
- **interface** types; is it/should it be constrained? how would different interfaces affect result?

Johnson & henderson

A CONCEPTUAL MODEL EXCLUDES

- low level presentation
- implementation details
- menu and screen designs
- widgets
- etc.

if you started here,
you will probably get into trouble



METAPHORS

well known concepts you can rely on to help users understand and interact with the system

many kinds, e.g.,

interactions

- *swipe to turn page* in an ebook
- *move backwards through time* to explore file backups

ecological, contextual, broader system structure, e.g.

- dropbox: *a box you drop everything into*
- iCloud: *central mother ship to which everything connects*

personal relationships, e.g.,

- siri as a *personal assistant*

EXAMPLE: THE DESKTOP METAPHOR

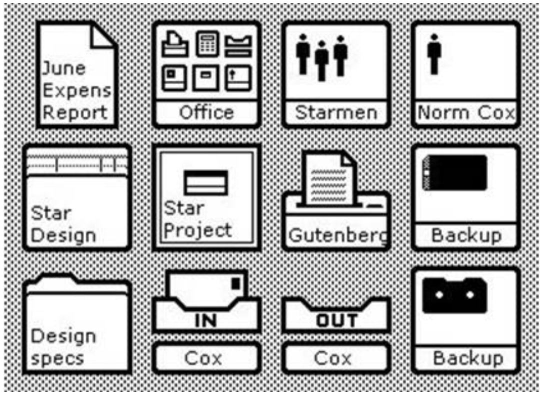
unifying set of concepts employed in graphical user interfaces to help users understand and easily interact with a computer

computer monitor → user's desktop

objects → documents, folders
you can do things with these objects:

- place documents upon desktop
- open documents into a window → paper copy
- organize in folders

extend desktop with desk accessories → calculator, notepad



RELATIONSHIPS AMONG CONCEPTS

what actions or attributes are shared between objects?

- e.g. song, podcast, audiobook all have timelines that users want to navigate (i.e. fast forward, rewind, etc.)

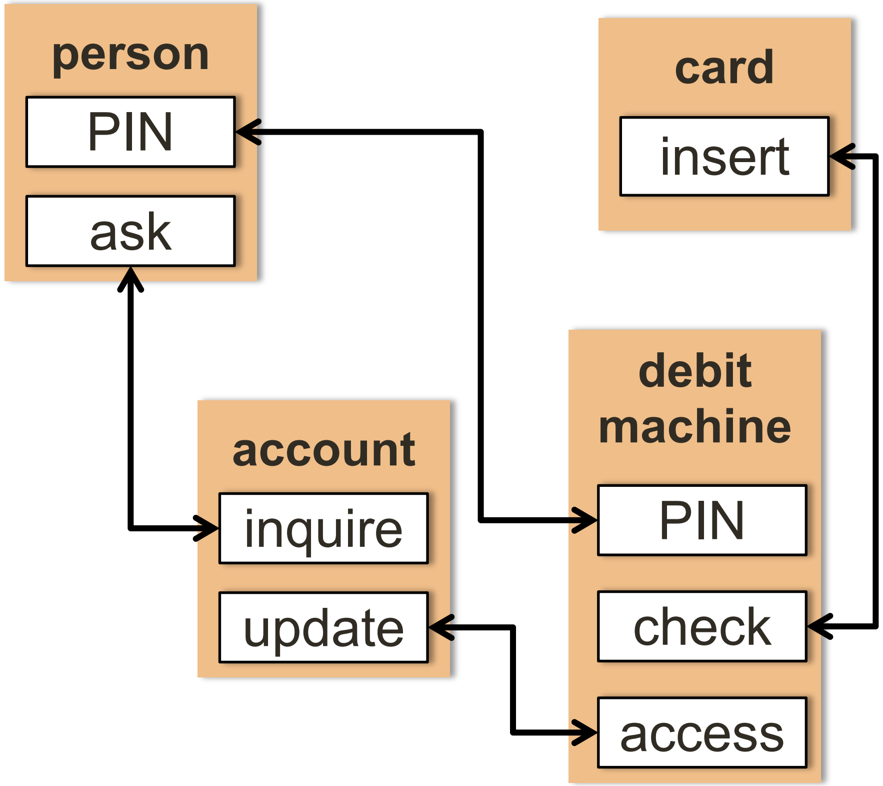
containment and hierarchy

- e.g., a song is contained by an album

MAPPING OF CONCEPTS TO ACTUAL ACTIVITIES

How do the concepts **map** to what people will actually do?

one easy way to tell:
"run" a task example on it



learn:

- are these the right objects?
- can I do all the operations?
- do they match what people want to do?
- can I do them in a consistent way?

example: debit machine

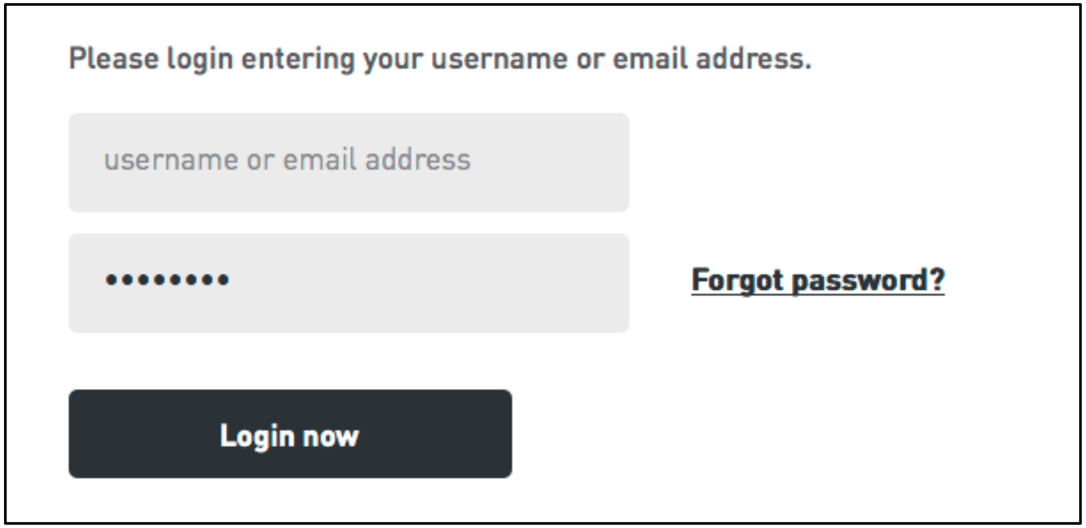
TERMINOLOGY

What terms will you use to communicate concepts?

terminology should match your concepts

choose your terminology and stick to it!

easy to go from planning to interface and minimize confusion



does your user login to a system with a user-id? a username? a member id? or an email address?

CONCEPTUAL MODELS: INTERACTION AND INTERFACE

Interaction type:

what the user is doing when interacting with a system.

- e.g., command line (how you talk to it), intelligent (function), gestural (hardware), touch (both hardware and interaction type)

Interface type:

the kind of interface used to support the mode.

- e.g. speech, menu-based, gesture

INTERACTION TYPES

Instructing

instruct a system and tell it what to do; issuing commands and selecting options (e.g. print a file, save a file)

Conversing

interacting with a system as if having a conversation (e.g. search engines, advice-giving systems, help systems, virtual agents)

Manipulating

interacting with objects in a virtual or physical space by manipulating them (e.g. dragging, selecting, opening, closing and zooming actions on virtual objects)

Exploring

moving through a virtual environment or a physical space (e.g. google maps, GPS)

INTERFACE TYPES

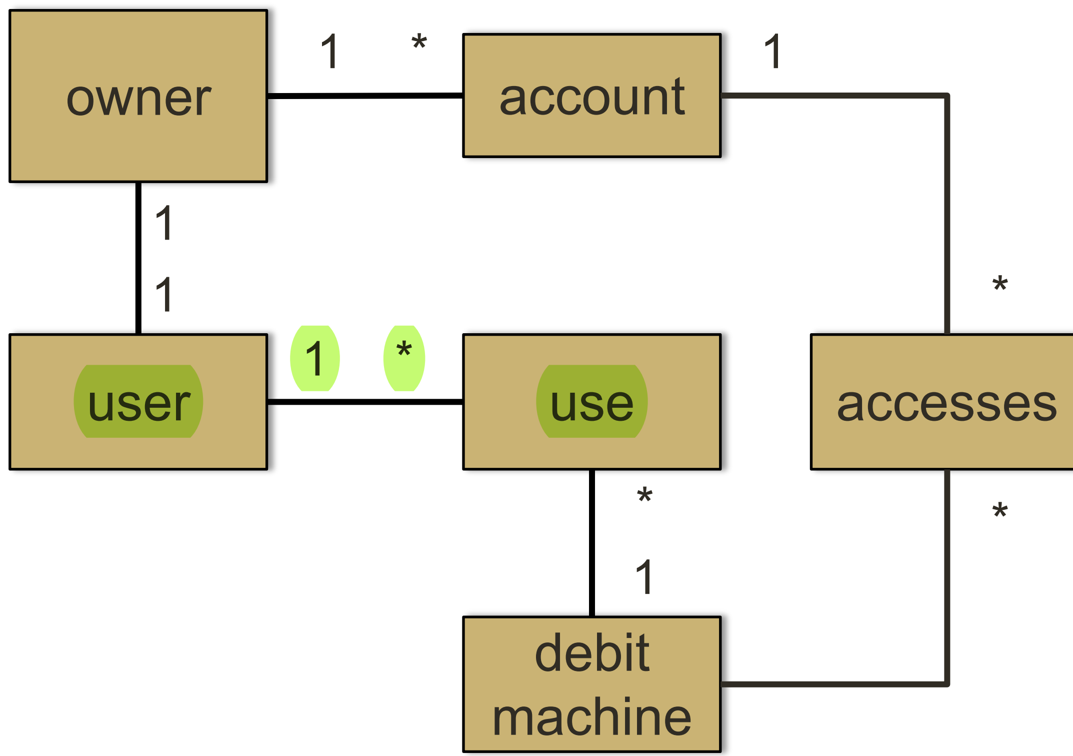
many different kinds (we won't examine each in detail)

- includes: mobile, GUI, touch, tangible, haptic, desktop, command line, data visualizations…

Interface type	See also
1. Command-based	
2. WIMP and GUI	
3. Multimedia	WIMP and web
4. Virtual reality	Augmented and mixed reality
5. Information visualization	Multimedia
6. Web	Mobile and multimedia
7. Consumer electronics and appliances	Mobile
8. Mobile	Augmented and mixed reality
9. Speech	
10. Pen	Shareable, touch
11. Touch	Shareable, air-based gesture
12. Air-based gesture	Tangible
13. Haptic	Multimodal
14. Multimodal	Speech, pen, touch, gesture, and haptic
15. Shareable	Touch
16. Tangible	
17. Augmented and mixed reality	Virtual reality
18. Wearable	
19. Robotic	
20. Brain–computer	

CONCEPTUAL MODEL FOR A DEBIT MACHINE

- *using a diagrammatic approach*
- shows concepts, relationships, terminology



IE 403/476

Human-Computer Interaction

Week 3-Lec2

Conceptual Models

- Need to first think about how the system will appear to users (i.e. how they will understand it)

- A high level description of
  - the proposed system with a set of integrated ideas and concepts about
    - what it should do
    - behave
    - look like
    - that will be understandable by the users in the manner intended

Understanding a Conceptual Model

- How will the user think about the system? Possibly based on:
  - Data or objects
  - Types of operations (activities) done
  - Metaphors (real world analogies/similarities)
- What kind of interface metaphor, if any, will be appropriate?
- What kinds of interaction modes and styles to use?

Fig 1. Word processor Vs a typewriter

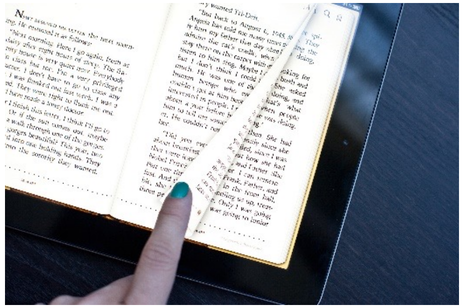


Fig 2. iBooks flipping pages similar to physical books

A desktop Conceptual Model

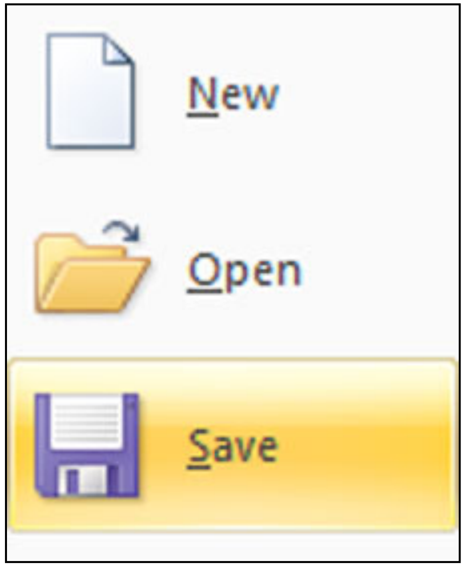
- Interface metaphor
- Items on a desk – Desktop
  - Files, notepad,
- Trash can below the table –
  - Recycle bin (windows)
  - Trash Can in Mac → Easier metaphor
- Files in folders with labels

Developing a CM

A CM is developed from three perspectives

- Interaction mode
- Interface metaphor
- Interface Paradigms

Interaction Modes

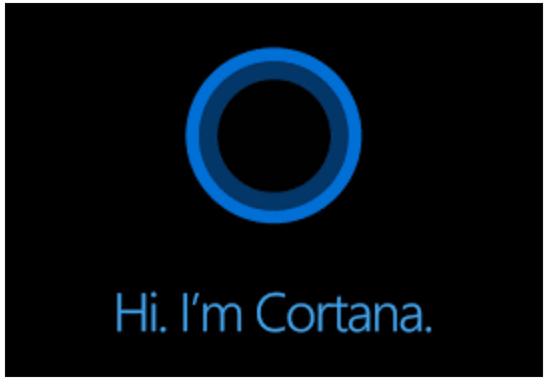


- Activity-based
  - Instructing
  - Conversing
  - Manipulating & Navigating
  - Exploring & Browsing

- Users instruct the system and tell it what to do e.g. tell the time, print a file, save a file
- Implementations: typing, pressing buttons, etc.
- Quick and efficient interaction

Interaction Modes

- Activity-based
  - Instructing
  - Conversing
  - Manipulating & Navigating
  - Exploring & Browsing



Interaction Modes

- Activity-based
  - Instructing
  - Conversing
  - Manipulating & Navigating
  - Exploring & Browsing

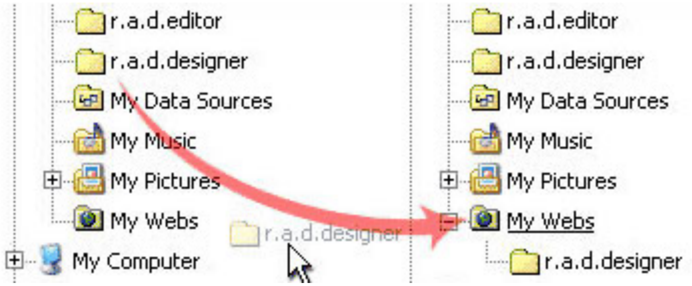


Fig 1. Navigate folders/files similar to real world set up

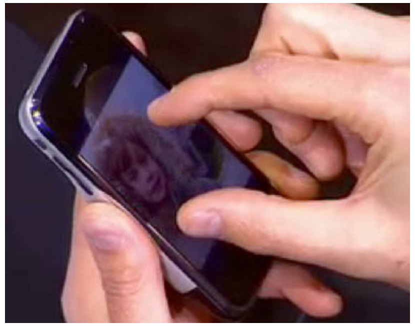
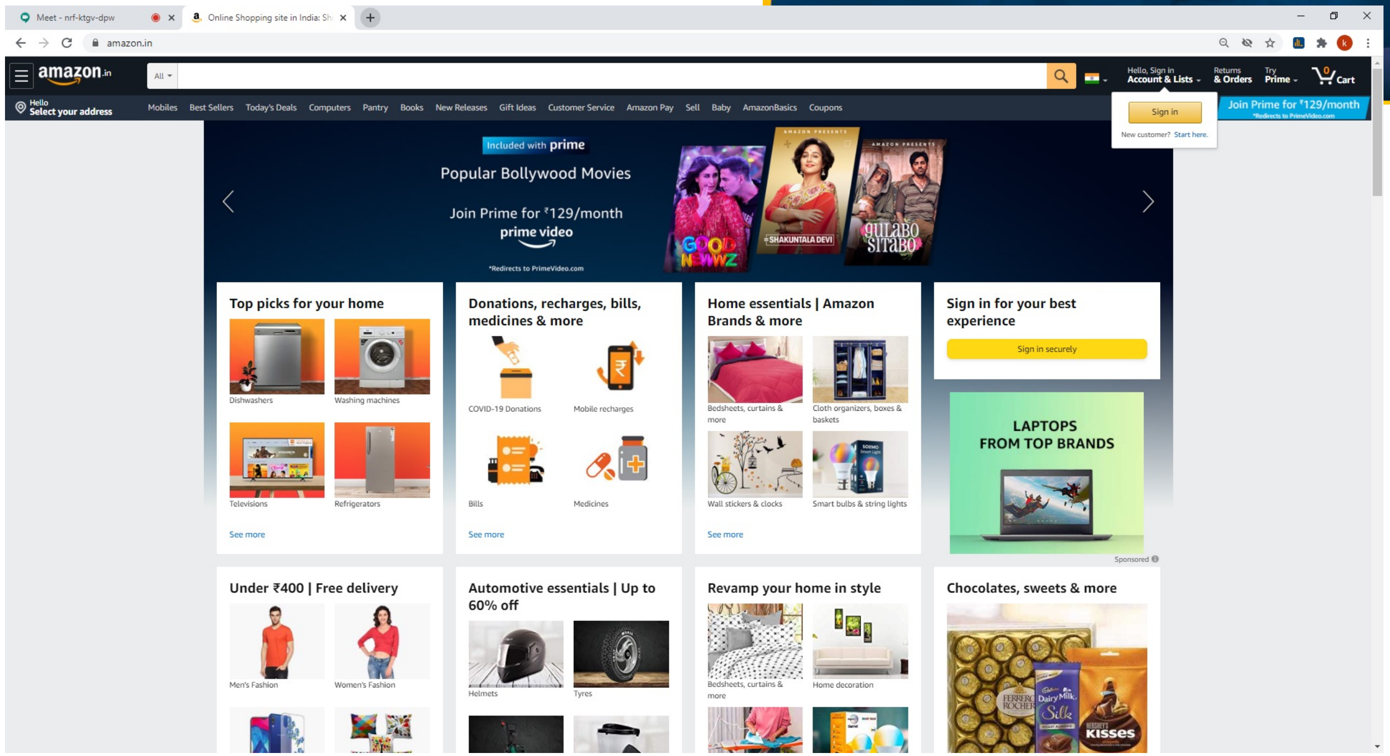


Fig 2. Direct Manipulation: Drag, zoom, pinch etc

Fig 3. Game interface with user experience similar to actual scenario

Interaction Modes

- Activity-based
  - Instructing
  - Conversing
  - Manipulating & Navigating
  - Exploring & Browsing



Interface Metaphors

- Conceptual model similar to some aspects of a physical entity
- Need to be evaluated
  - Structure
  - Relevance
  - Representation
  - Clarity
  - Extensibility

Interface Paradigms

- Desktop
- Ubiquitous
- Pervasive
- Wearable
- …

- Consider user tasks & environmental requirements

What should a CM consist of ?

Object/Action relationships

- Metaphors or analogies
- the (user-level) concepts to be created and manipulated
- the relationships between concepts,
  - Attributes has-a
  - Specialisations is-a
  - Containment contains
- the mappings between concepts and task domain

- Functions performed and by whom
  - Task allocation
  - Relationship between functions
    - Relative position
    - Sequential
    - Parallel
  - Importance
    - How frequent?
  - How data is captured, transformed and output?

Example 1: Online Library

- ❖ Metaphors → Physical card catalogue
- ❖ Concepts → Item, book, periodical, issue, DVD, shelf- mark, user account,
- ❖ object relationships → a book is a type of item; periodicals contain issues
- ❖ Mappings → item corresponds to a physical object; shelf-mark to its physical location
- ❖ functions → issue item, return item, search item
- ❖ Function relationships → issue before return for same item; for different items, in parallel
- ❖ Data → new items added by typing data

Example 2: Bank transactions

- Objects→customer, checking account, savings accnt, cheque
- Actions→withdraw, deposit, open/close, viewing, transfer
- NO→click button, load database, create record etc., these are a) how the action would be enabled (UI part) or no reqd for customer to know DB, record etc.,
- May be → create template, command / action sequence
- Attributes → Properties
  - what will a cheque have: number, date, balance, interest rate, date opened etc., No: transaction memory size,

Understand

- Problem space
- Task domains
- Create story like Scenarios

Mental models

- Users “see” the system through mental models
- Users “rely” on mental models during usage
- Reason about a sys
    - Interact with
    - Infer how it works
    - Figure out how to correct when things do wrong

Why are mental models important?

2016, Chrysler automobiles recalled over 100K vehicles

- Gear shows P, R, N, D/S
- Shift through the gear options
- But returns to center position?
- What is the problem ?

People were getting out of their cars thinking the gear was in Park mode
But it wasn't, so the car drove off without them!!!

100 crashes

40 injuries

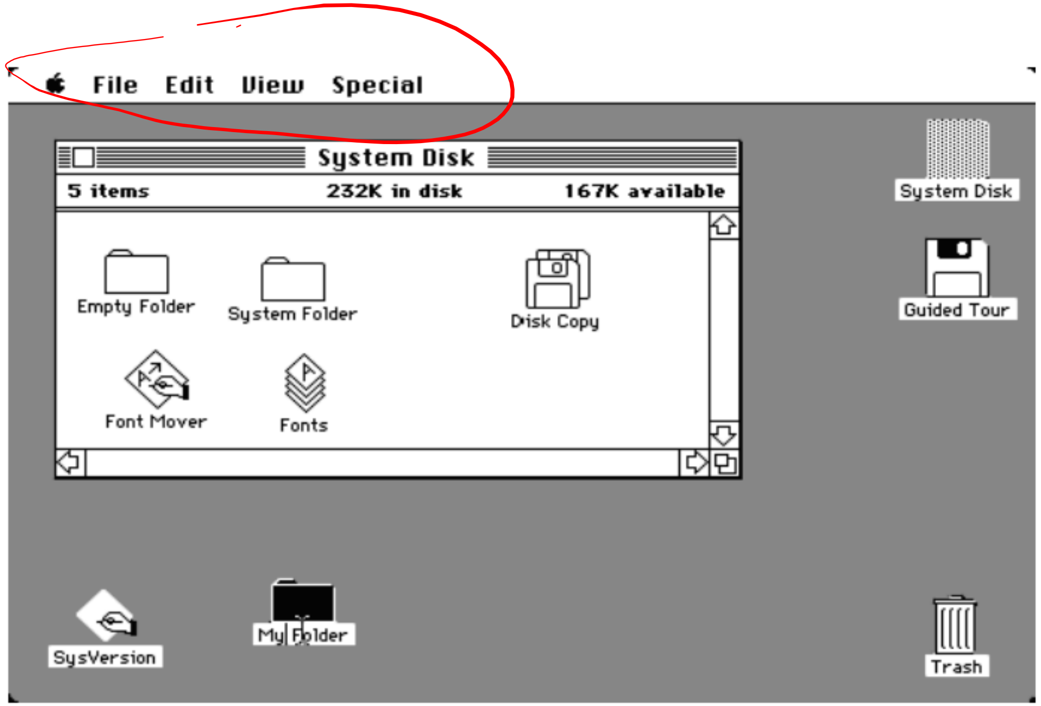
Classic Design Flaw

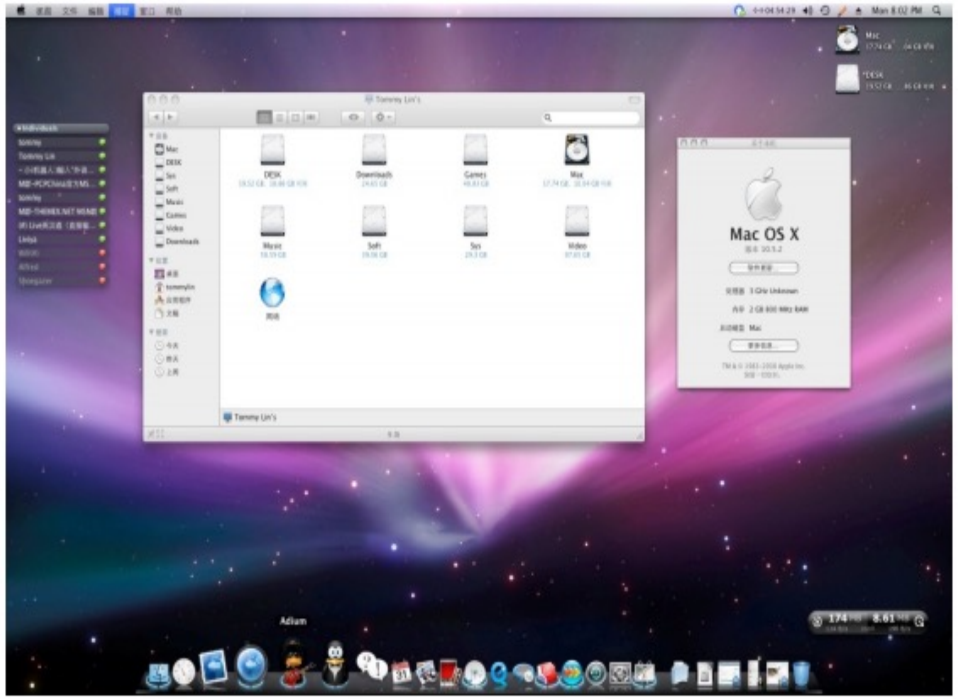
Designing something that people don't understand or making some thing which is totally NEW & Expecting USERS to figure out
User's mental model is not the same as what was designed

https://www.carcomplaints.com/news/2017/lawsuit-chrysler-gear-shift-confusing.shtml

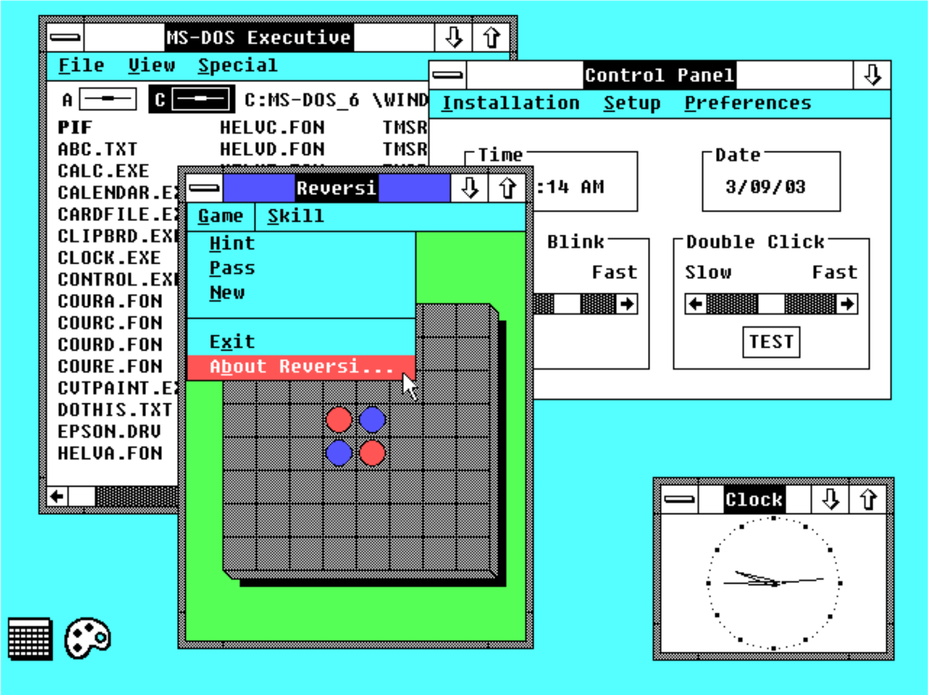
Evolution of OS interface design - Examples

Mac OS


File Edit View Special
System Disk
5 items 232K in disk 167K available
Empty Folder
System Folder
Disk Copy
Font Mover
Fonts
System Disk
Guided Tour
SysVersion
My Folder
Trash

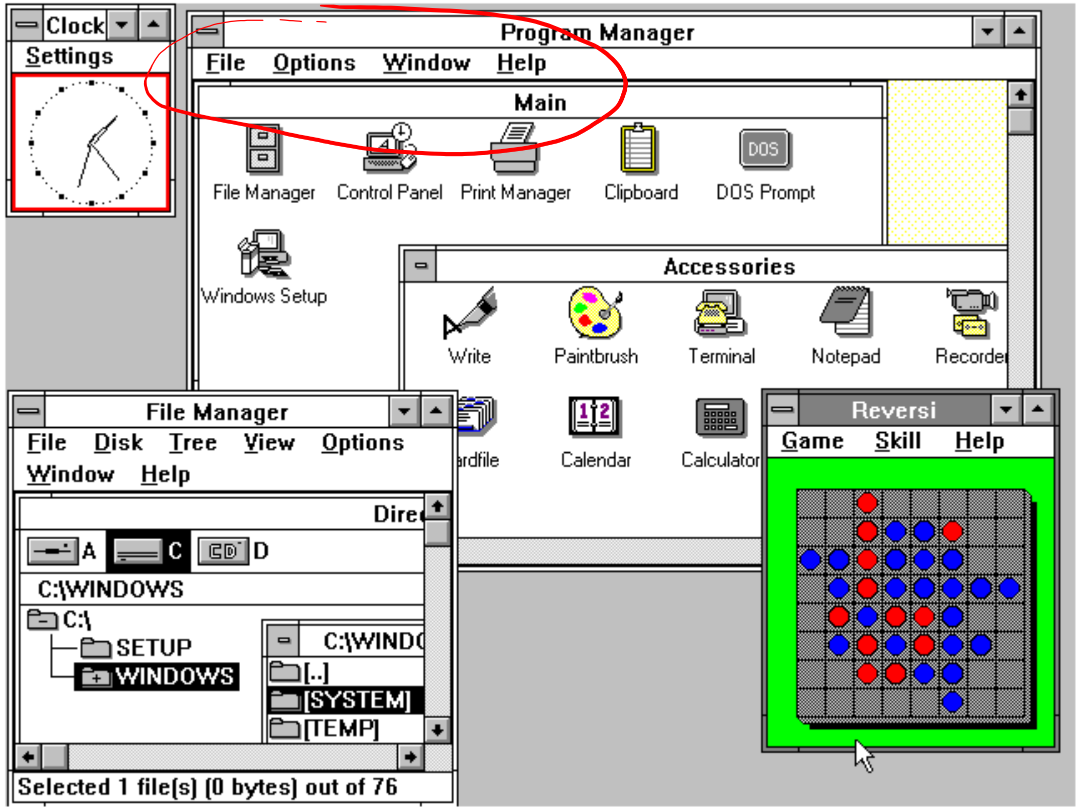

Mac OS X

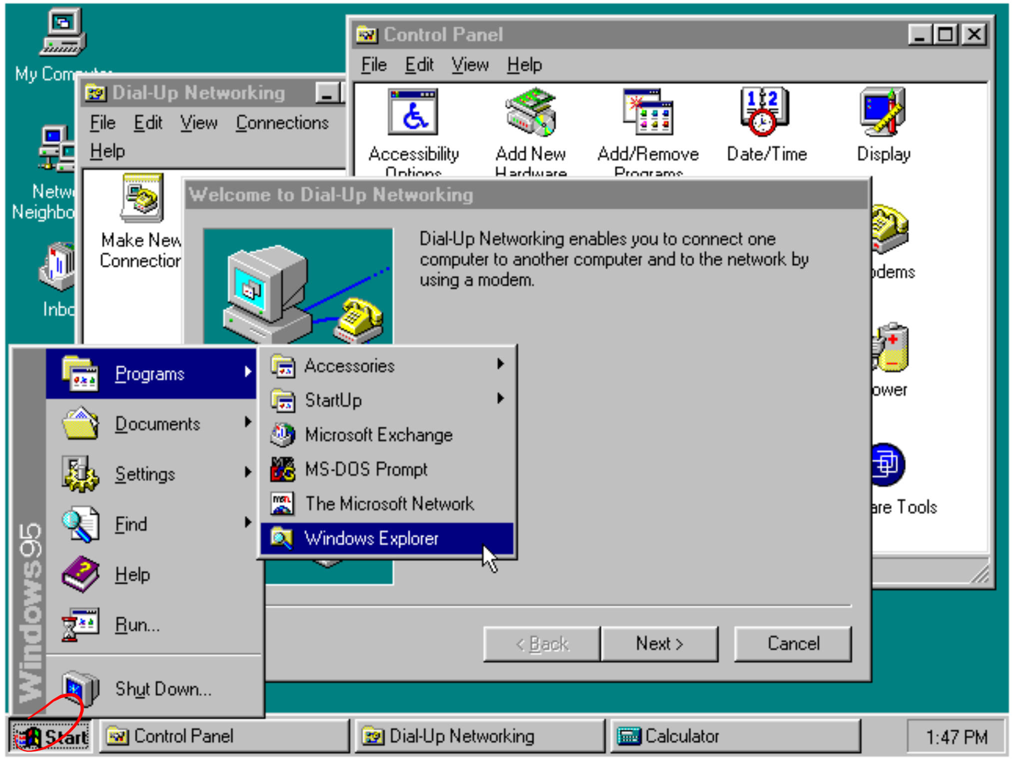
Windows



Win 1.0

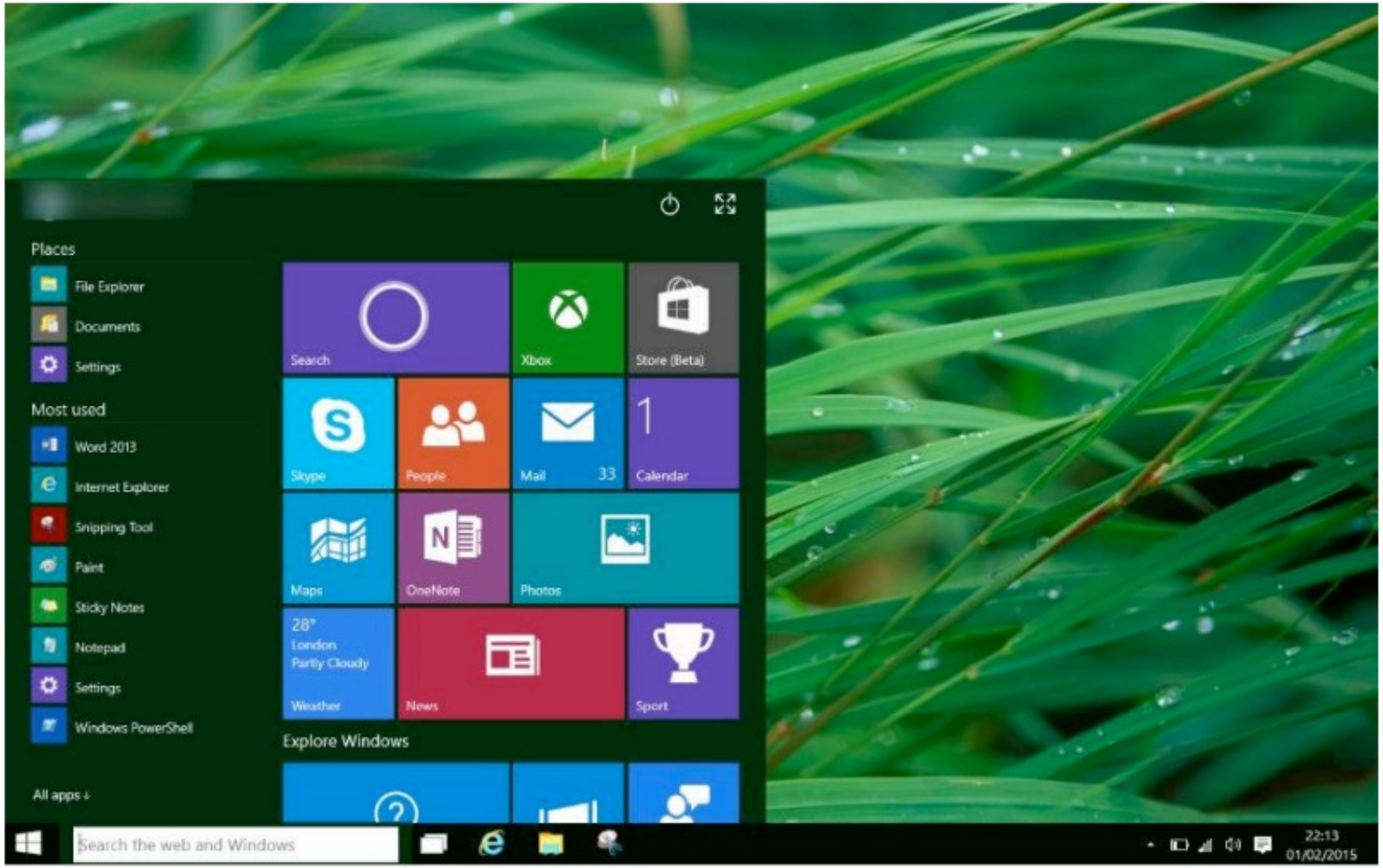
Win 3.0





Win 95

Win Vista


Places
File Explorer
Documents
Settings
Most used
Word 2013
Internet Explorer
Snipping Tool
Paint
Sticky Notes
Notepad
Settings
Windows PowerShell
All apps
Search
Xbox
Store (Beta)
Skype
People
Mail
33
1
Calendar
Maps
OneNote
Photos
28°
London
Partly Cloudy
Weather
News
Sport
Explore Windows
Search the web and Windows
22:13
01/02/2015

Win 10

Assignment 3

- Write down your own mental model of how a cash machine (ATM) works
- Answer the following:
  - What happens to prevent you taking out more than the limit by using several machines in turn?
  - What information is on the card itself, and how is it used?
  - Why are there pauses between steps, and why are they duration they are?
  - What happens to the card while in the machine?
  - Do you count the money? Why or why not?
- Now ask two other people the same questions and compare your mental models.

IE 403/476

Human-Computer Interaction
Conceptual Models-II

Conceptual Models

- Need to first think about how the system will appear to users (i.e. how they will understand it)

- A high level description of
  - the proposed system with a set of integrated ideas and concepts about
    - what it should do
    - behave
    - look like
    - that will be understandable by the users in the manner intended

Understanding a Conceptual Model

- How will the user think about the system? Possibly based on:
  - Data or objects
  - Types of operations (activities) done
  - Metaphors (real world analogies/similarities)
- What kind of interface metaphor, if any, will be appropriate?
- What kinds of interaction modes and styles to use?

Fig 1. Word processor Vs a typewriter

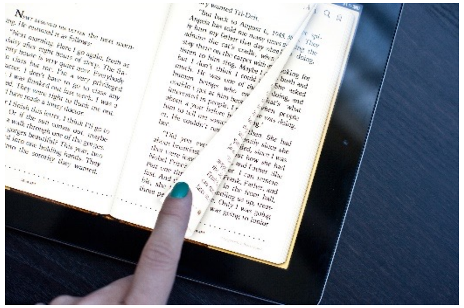


Fig 2. iBooks flipping pages similar to physical books

A desktop Conceptual Model

- Interface metaphor
- Items on a desk – Desktop
  - Files, notepad,
- Trash can below the table –
  - Recycle bin (windows)
  - Trash Can in Mac → Easier metaphor
- Files in folders with labels

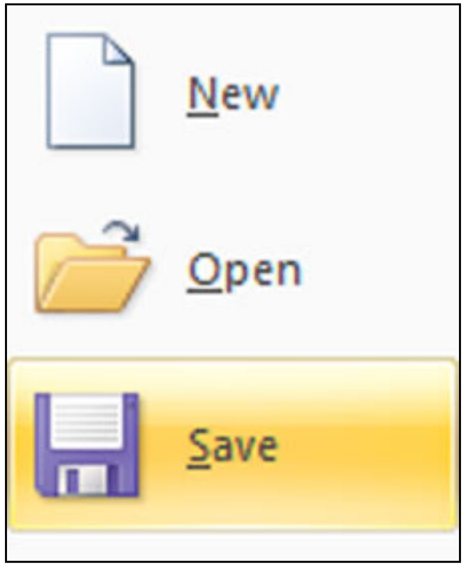
Developing a CM

A CM is developed from three perspectives

- Interaction mode
- Interface metaphor
- Interface Paradigms

Interaction Modes

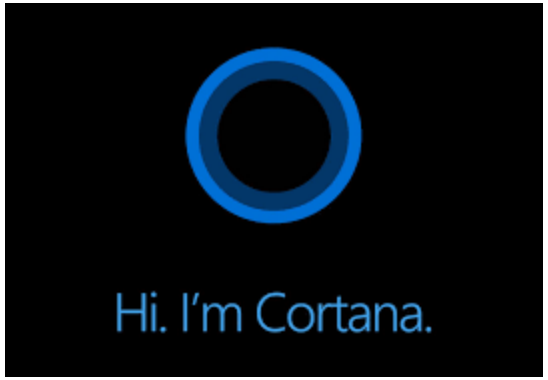
- Activity-based
  - Instructing
  - Conversing
  - Manipulating
    & Navigating
  - Exploring
    & Browsing



- Users instruct the system and tell it what to do e.g. tell the time, print a file, save a file
- Implementations: typing, pressing buttons, etc.
- Quick and efficient interaction

Interaction Modes

- Activity-based
  - Instructing
  - Conversing
  - Manipulating & Navigating
  - Exploring & Browsing



Interaction Modes

- Activity-based
  - Instructing
  - Conversing
  - Manipulating & Navigating
  - Exploring & Browsing

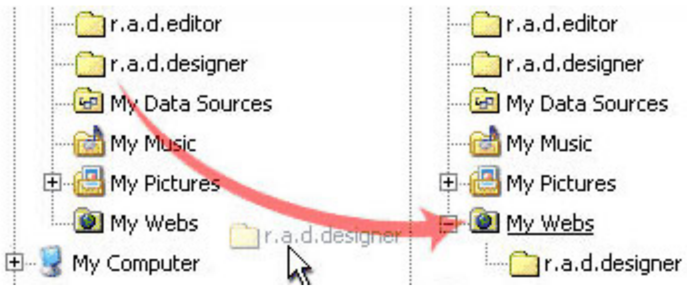


Fig 1. Navigate folders/files similar to real world set up

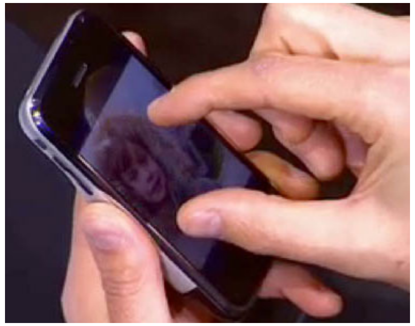
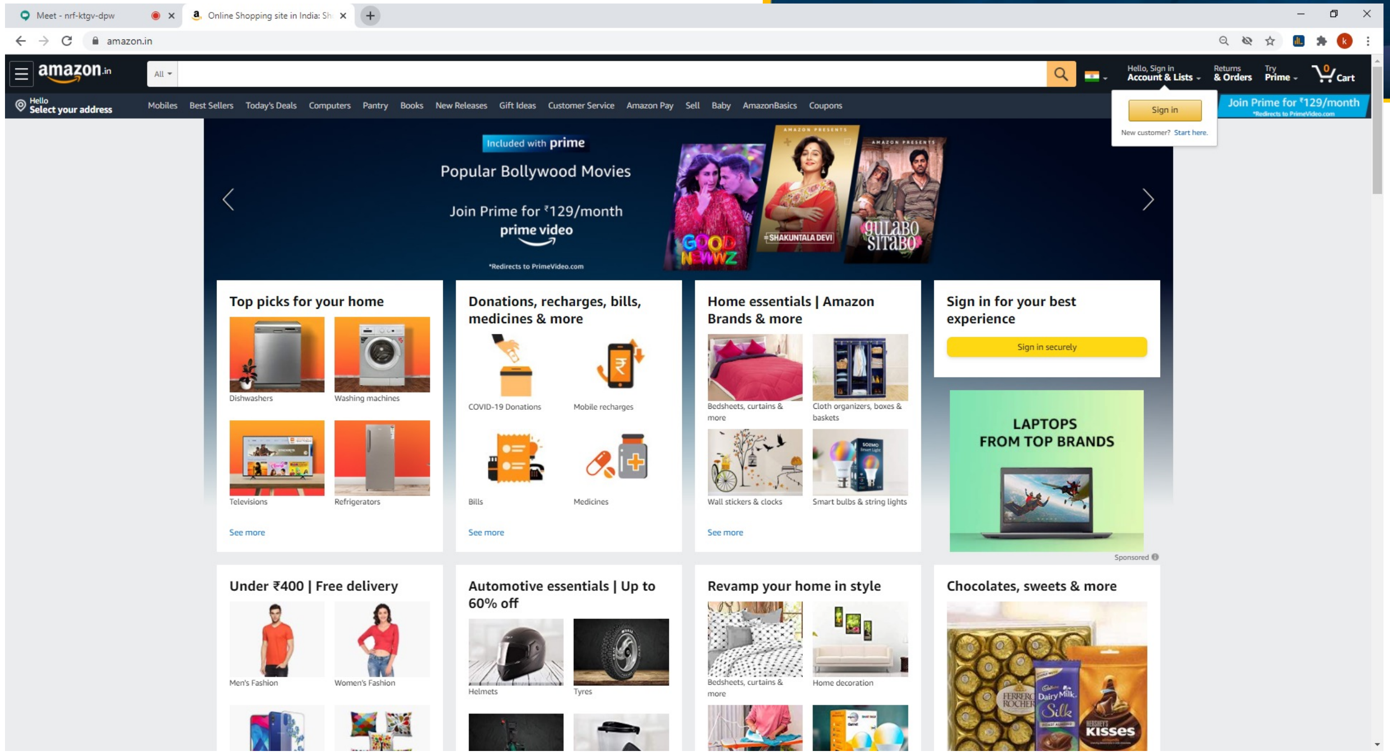


Fig 2. Direct Manipulation: Drag, zoom, pinch etc

Fig 3. Game interface with user experience similar to actual scenario

Interaction Modes

- Activity-based
  - Instructing
  - Conversing
  - Manipulating & Navigating
  - Exploring & Browsing



Interface Metaphors

- Conceptual model similar to some aspects of a physical entity
- Need to be evaluated
  - Structure
  - Relevance
  - Representation
  - Clarity
  - Extensibility

Interface Paradigms

- Desktop
- Ubiquitous
- Pervasive
- Wearable
- …

- Consider user tasks & environmental requirements

What should a CM consist of ?

Object/Action relationships

- Metaphors or analogies
- the (user-level) concepts to be created and manipulated
- the relationships between concepts,
  - Attributes has-a
  - Specialisations is-a
  - Containment contains
- the mappings between concepts and task domain

- Functions performed and by whom
  - Task allocation
  - Relationship between functions
    - Relative position
    - Sequential
    - Parallel
  - Importance
    - How frequent?
  - How data is captured, transformed and output?

Example 1: Online Library

- ❖ Metaphors → Physical card catalogue
- ❖ Concepts → Item, book, periodical, issue, DVD, shelf- mark, user account,
- ❖ object relationships → a book is a type of item; periodicals contain issues
- ❖ Mappings → item corresponds to a physical object; shelf-mark to its physical location
- ❖ functions → issue item, return item, search item
- ❖ Function relationships → issue before return for same item; for different items, in parallel
- ❖ Data → new items added by typing data

Example 2: Bank transactions

- Objects→customer, checking account, savings accnt, cheque
- Actions→withdraw, deposit, open/close, viewing, transfer
- NO→click button, load database, create record etc., these are a) how the action would be enabled (UI part) or no reqd for customer to know DB, record etc.,
- May be → create template, command / action sequence
- Attributes → Properties
  - what will a cheque have: number, date, balance, interest rate, date opened etc., No: transaction memory size,

Understand

- Problem space
- Task domains
- Create story like Scenarios

Mental models

- Users “see” the system through mental models
- Users “rely” on mental models during usage
- Reason about a sys
  - Interact with
  - Infer how it works
  - Figure out how to correct when things do wrong

Why are mental models important?

2016, Chrysler automobiles recalled over 100K vehicles

- Gear shows P, R, N, D/S
- Shift through the gear options
- But returns to center position?
- What is the problem ?

People were getting out of their cars thinking the gear was in Park mode
But it wasn't, so the car drove off without them!!!

100 crashes

40 injuries

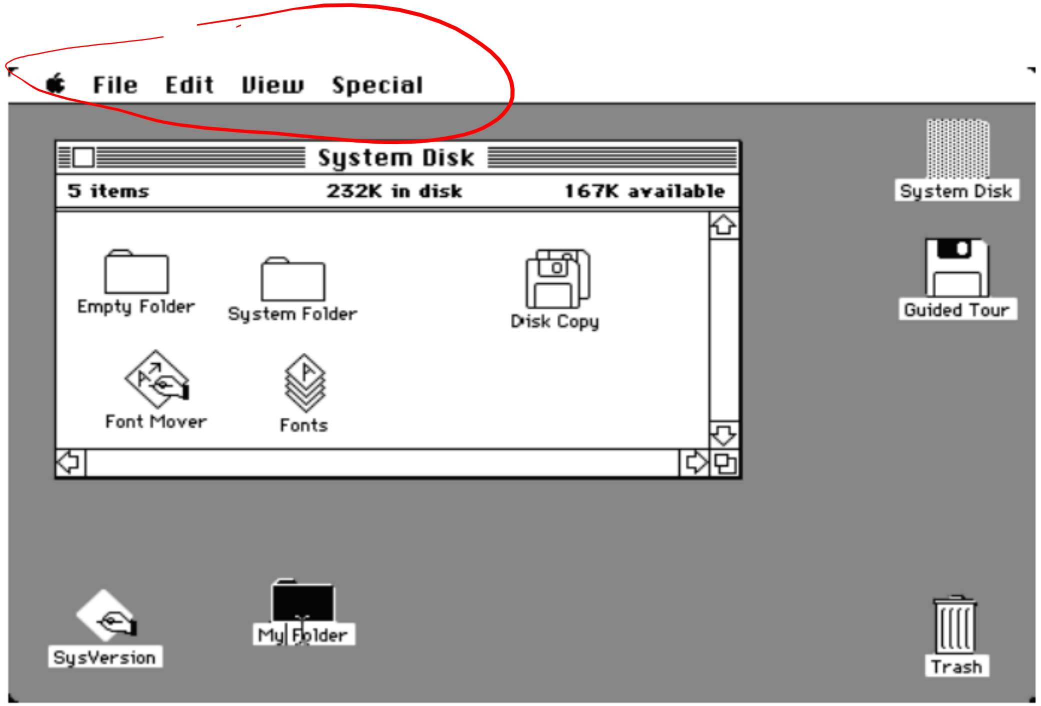
Classic Design Flaw

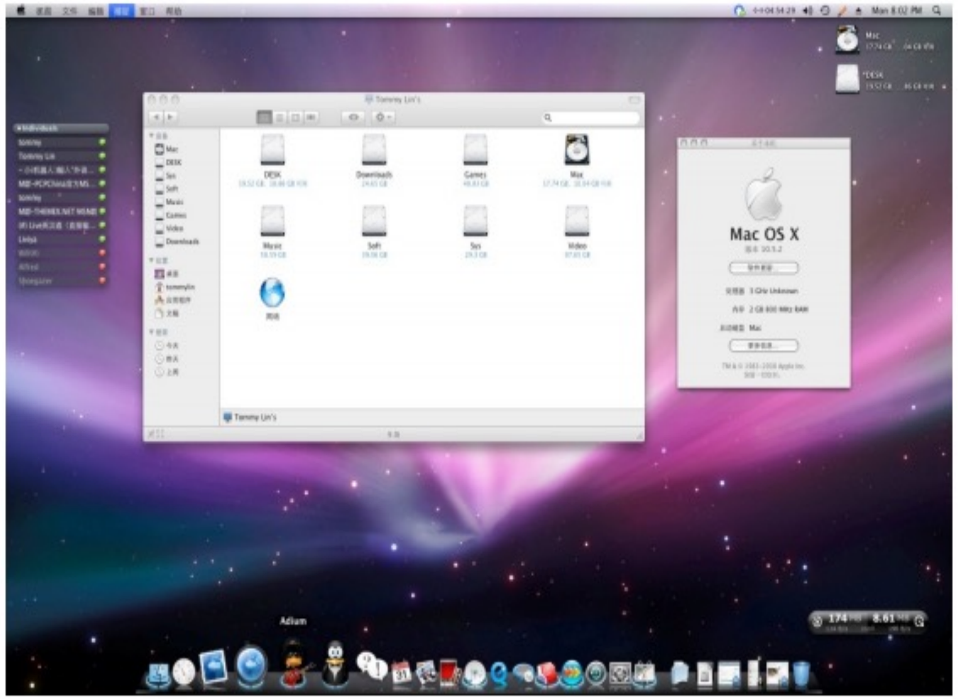
Designing something that people don't understand or making some thing which is totally NEW & Expecting USERS to figure out
User's mental model is not the same as what was designed

https://www.carcomplaints.com/news/2017/lawsuit-chrysler-gear-shift-confusing.shtml

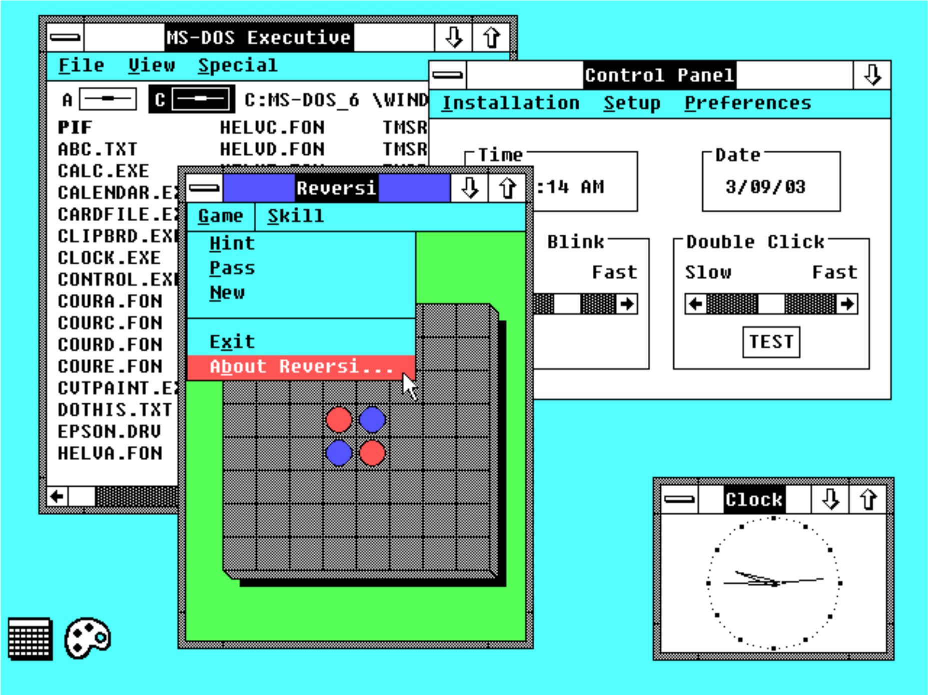
Evolution of OS interface design - Examples

Mac OS


File Edit View Special
System Disk
5 items
232K in disk
167K available
Empty Folder
System Folder
Disk Copy
Font Mover
Fonts
System Disk
Guided Tour
SysVersion
My Folder
Trash

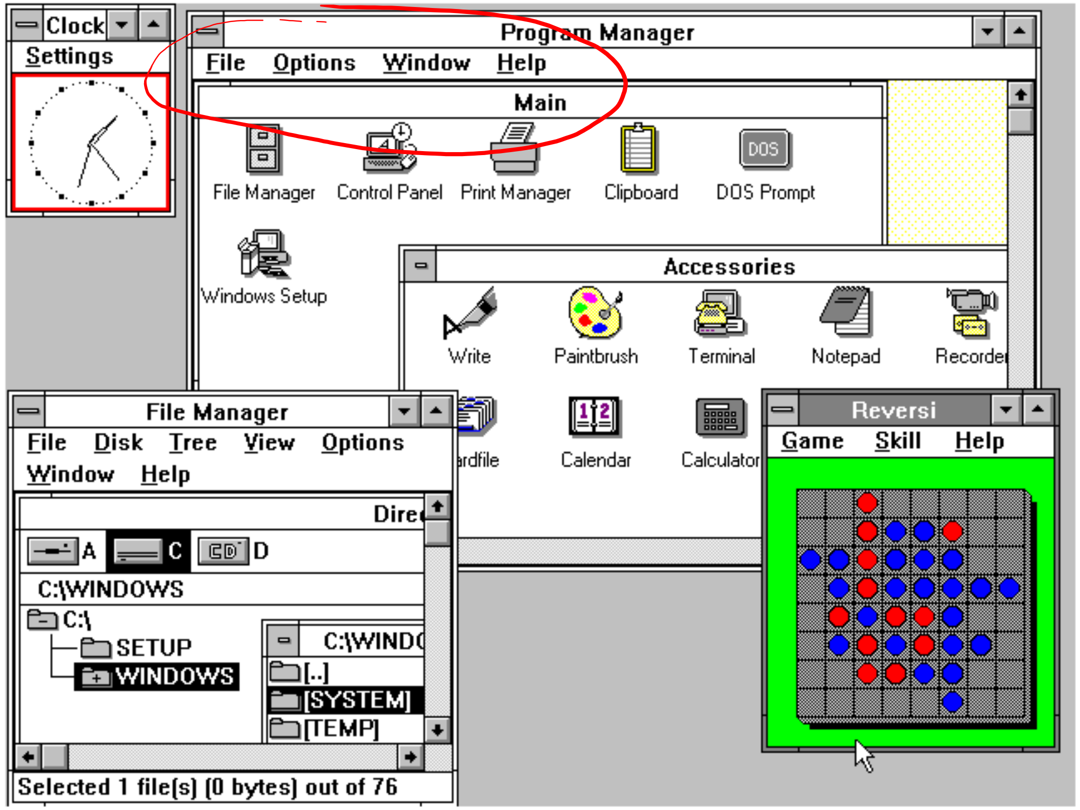

Mac OS X

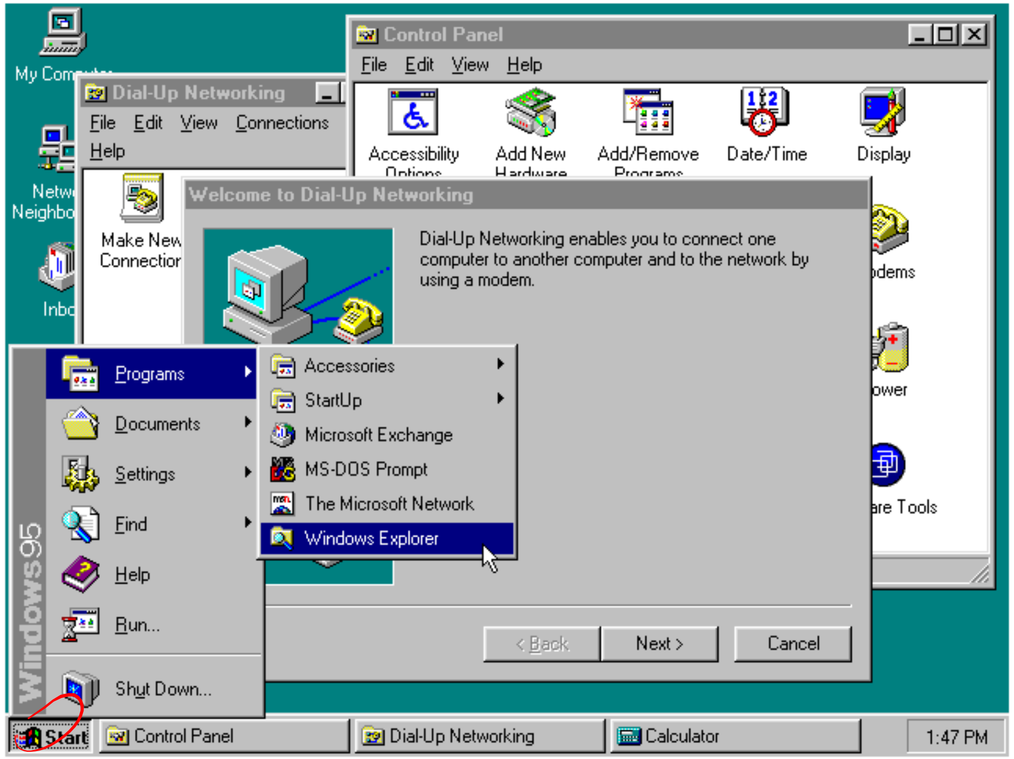
Windows



Win 1.0

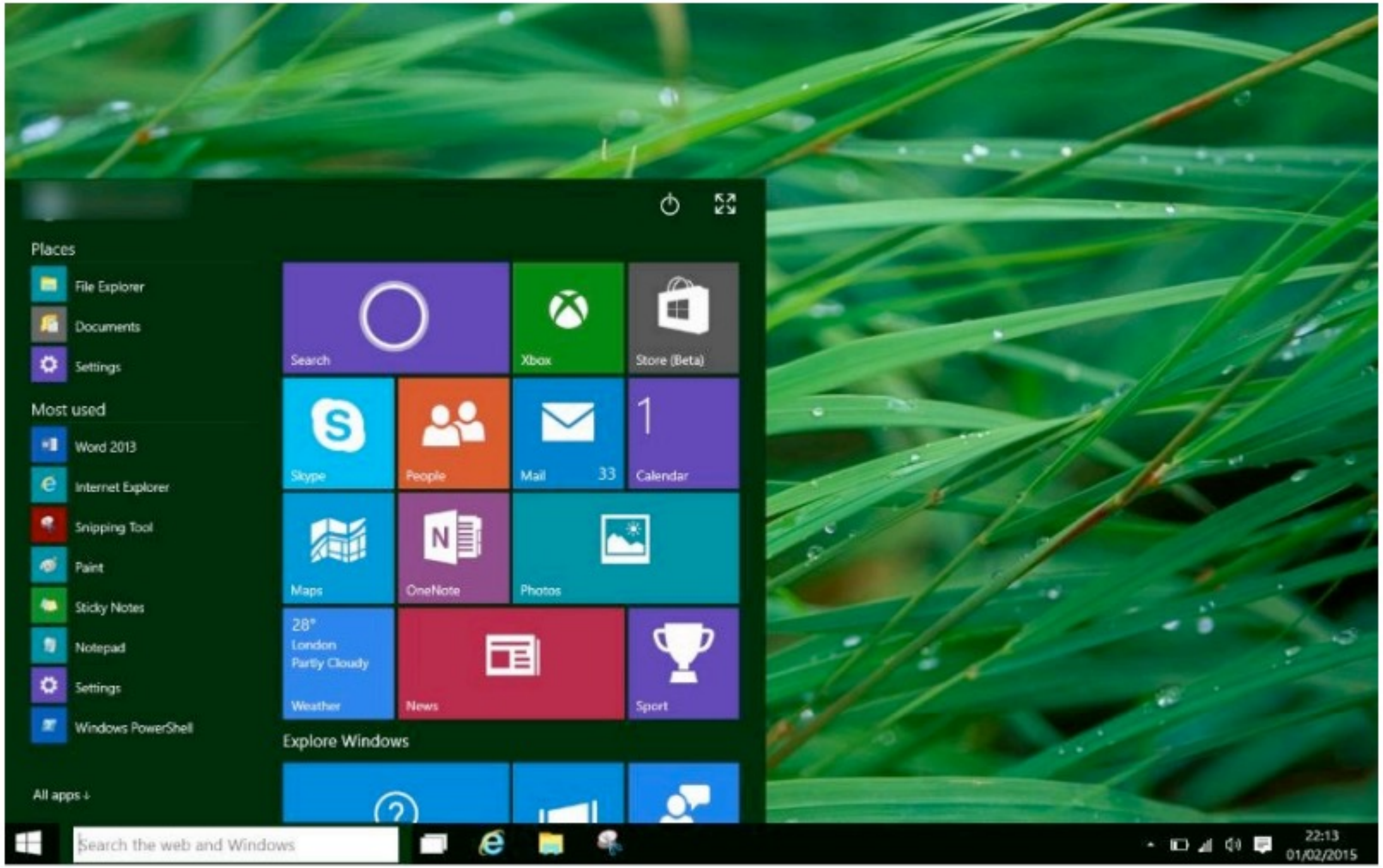
Win 3.0





Win 95

Win Vista


Places
File Explorer
Documents
Settings
Most used
Word 2013
Internet Explorer
Snipping Tool
Paint
Sticky Notes
Notepad
Settings
Windows PowerShell
All apps
Search
Xbox
Store (Beta)
Skype
People
Mail 33
1
Calendar
Maps
OneNote
Photos
28°
London
Partly Cloudy
Weather
News
Sport
Explore Windows
Search the web and Windows
22:13
01/02/2015

Win 10

Assignment 3

- Write down your own mental model of how a cash machine (ATM) works
- Answer the following:
  - What happens to prevent you taking out more than the limit by using several machines in turn?
  - What information is on the card itself, and how is it used?
  - Why are there pauses between steps, and why are they duration they are?
  - What happens to the card while in the machine?
  - Do you count the money? Why or why not?
- Now ask two other people the same questions and compare your mental models.

IE 403

Human-Computer Interaction
GoEx and GoEv

Usability

- ISO 9241 defines usability as effectiveness, efficiency and satisfaction with which users accomplish tasks
- The ability of a User to Use the product/ system / environment as desired
- Usability Engineering: The ‘affordance’ offered by a product that makes it useable

Dimensions of Usability

- Learnability: Easy to learn
- Efficiency: Once learned, is it fast to use?
- Errors: Are errors few and recoverable?
- Visibility: Is the state of the system visible?
- Effectiveness: Can it do the job well and correctly?
- Satisfaction: Is the user happy with the interface?

Building all these dimensions into a product is called Usability Engineering

Rate the usability of each on a scale of 5

A

B

C

Fig 1: Microwave model I

Fig 2: Microwave model 2

Conceptual models

Fig 1. Scissors

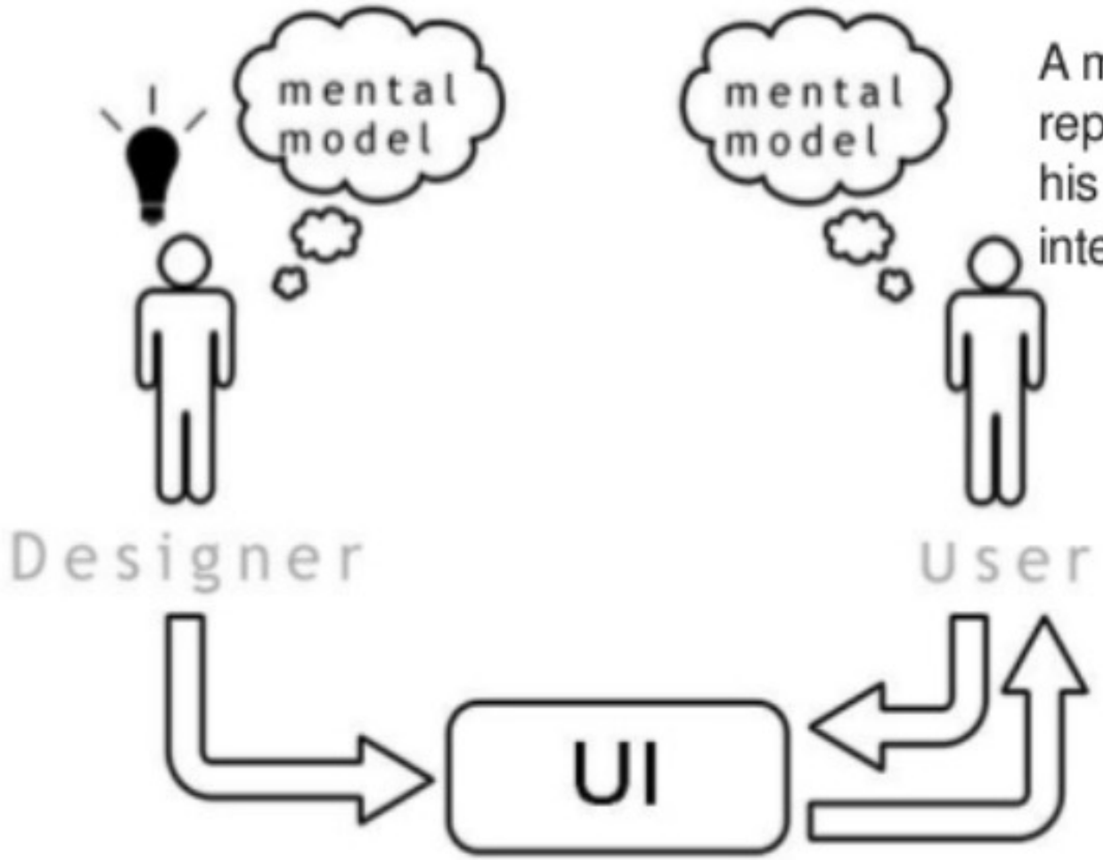
Fig 2. Files in folders Icon

- describe how an interactive system is organized
- Is the foundation of the interface.
- Ask:
  - what users will be able to do
  - what concepts or knowledge users will need, in order to interact
  - how they will interact with system (at a very high level)

Mental models

- Users “see” the system through mental models
- Users “rely” on mental models during usage
- Reason about a sys
  - Interact with
  - Infer how it works
  - Figure out how to correct when things do wrong

Mental Model vs Conceptual Model



A mental model is the representation that a person has in his mind about the object he is interacting with.

A conceptual model is the actual model that is given to the person through the design and interface of the actual product.

https://www.slideshare.net/limdonald/web-design-through-the-lens-of-psychology-23548623

System Image

- The designer's conceptual model
  - How do I want it to be, work and operate?
- The system Model:
  - Actual physical structure, documentation
- User's mental model
  - How is it supposed to be or work?
- DESIGN FLAW
  - Designer's model = User's model

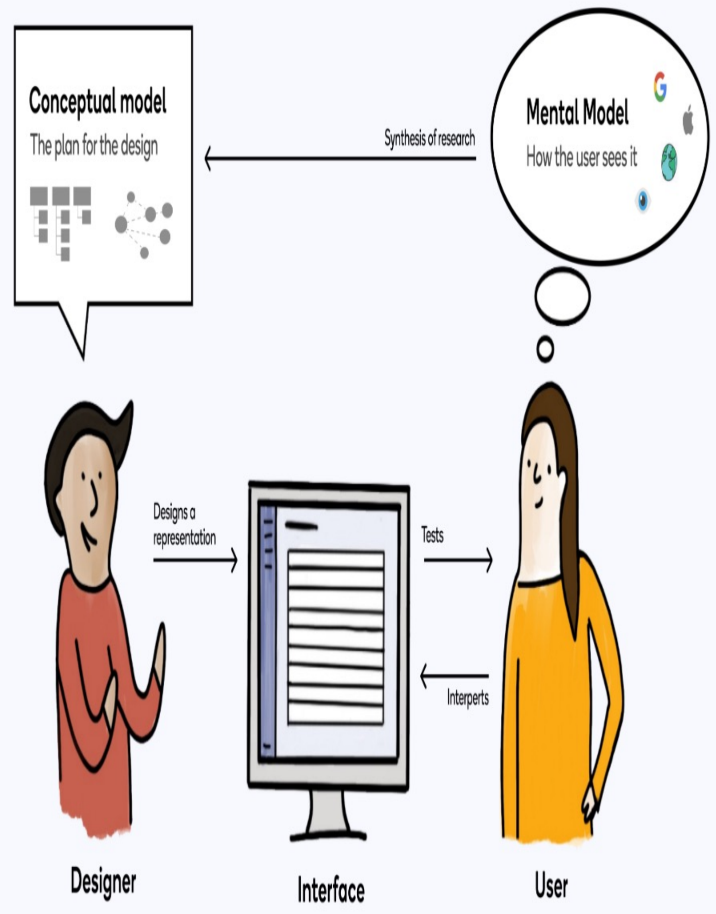


Fig 3. conceptual Vs Mental Model
https://uxdesign.cc/understanding-mental-and-conceptual-models-in-product-design-7d69de3cae26

Learnability & Memorability

Making interfaces easier for

- *new users to learn*
- *casual users to remember.*

Recognition Vs Recall

Recognition:

- remembering with the help of a clue
  - “using knowledge in or of the world

Recall:

- Remembering with no help
  - “using knowledge in the head”
  - How much do we remember?

I hear and I forget.

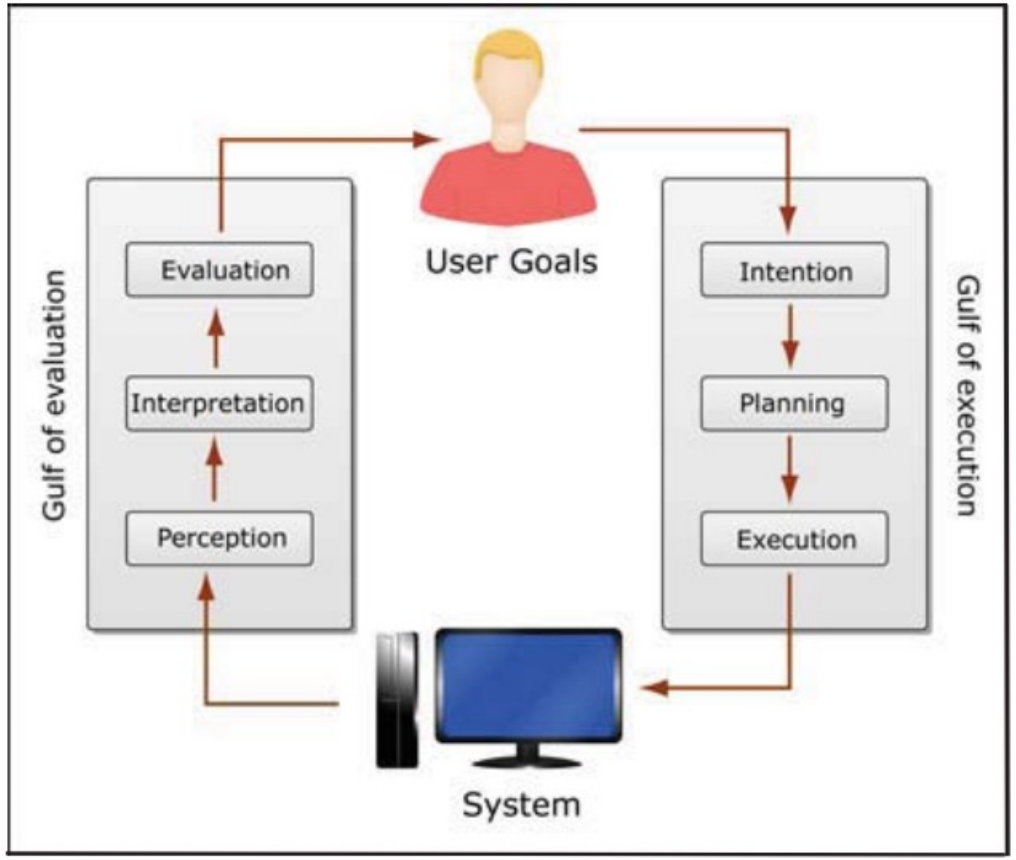
I see and I remember.

I do and I understand.

Confucius

https://www.interaction-design.org/literature/article/simplicity-in-design-4-ways-to-achieve-simplicity-in-your-designs

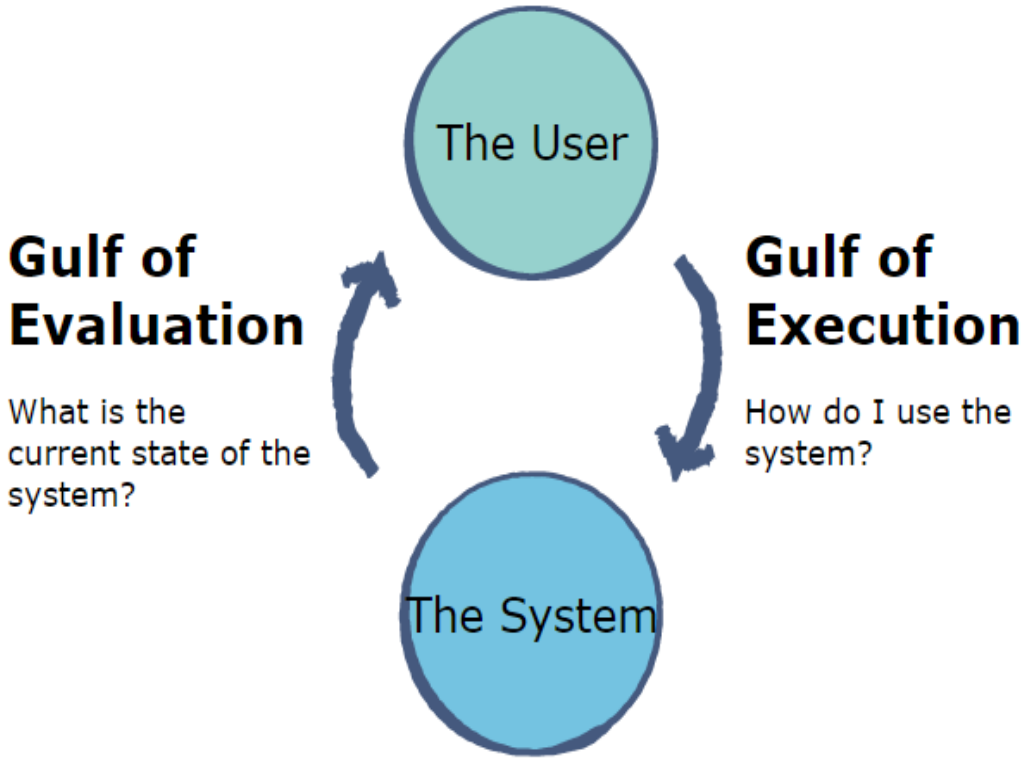
A generalized Cognitive Model



- Learnability: Execution side
- Visibility & Feedback : Evaluation side
- Efficiency: Measure of whole cycle, speed of execution and perception
  - What I did Vs What just happened Vs What is it I am going to see/experience?

GoEv Vs GoEx

The Gulf of Execution (GoEx) is
- *the difference between the intentions of the users and what the system allows them to do*
- *how well the system supports those actions*



https://www.educative.io/edpresso/gulf-of-execution-and-gulf-of-evaluation

The Gulf of Evaluation (GoEv) is
- *the level of difficulty in assessing the state of a system*
- *how well the artifact supports the discovery and interpretation of that state*

Norman’s 7 stages of Actions

Questions	GOAL	EXECUTION	EVALUATION
What do I want to accomplish?	✓		
What are my alternatives?		✓	
What can I do now?		✓	
How do I do it?		✓	
What happened?			✓
What does it mean?			✓
Is it OK? Have I accomplished my goal?			✓

Making Coffee in a Brewer

Questions	GOAL	EXECUTION	EVALUATION
What do I want to accomplish?	Coffee		
What are my alternatives?			
What can I do now?		___	
How do I do it?		___	
What happened?		___	___
What does it mean?		___	___ Coffee
Is it OK? Have I accomplished my goal?			Coffee!

Learnable Interface → *Smaller GoEx & GoEva*

↓

Easier to execute

How designers can bridge the Gulf?

- Bridge gulf of execution by:
- ○ Signifiers, constraints,
- mapping, conceptual model
- ● Bridge gulf of evaluation by:
- ○ Feedback, conceptual model
- ● Help users answer these →
- questions
- ○ 7 questions; 7 stages

IE 403/476

Human-Computer Interaction

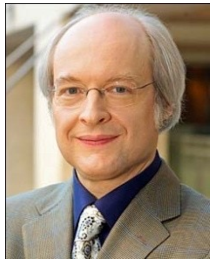
Week 4-Lec1

Agenda

- Jacob Nielsen’s 10 point rules of thumb
- Discussion on a research paper

Recap

- Cognitive Model
- Gulf of execution and Gulf of evaluation
- 7 stages of actions By Norman
- Mental Vs Conceptual Model

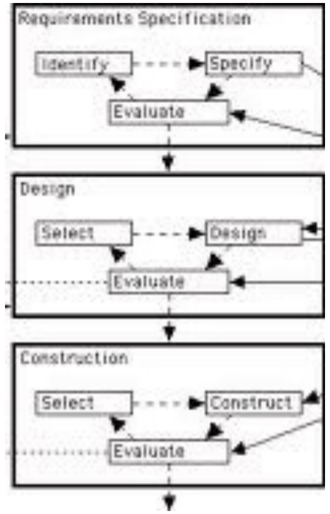


Jakob Nielsen

Usability consultant

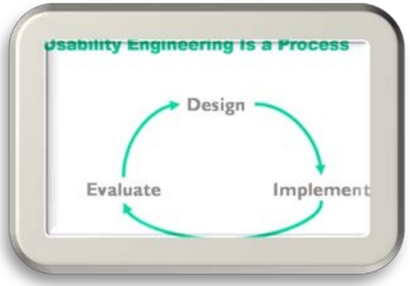
Jakob Nielsen is a leading web usability consultant. He holds a Ph.D. in human–computer interaction from the Technical University of Denmark in Copenhagen.

He has authored many books in Usability, HCI, & Experience design. His book titled " Usability Engineering" 1993 is a textbook on methods to make interfaces easier to use.



Usability Engineering

Usability Engineering involves User Research; Design Research and Validation of Design through Construction & User Testing.
In some institutions it is taught as an independent discipline while in others it is part of HCI discipline.



History of the Heuristics

- Derived
  - Factor analysis of 249 usability problems
- Method are empirically based derivations.
  - Through experiments and observations
- Method suggested by Nielsen is popular because of its
  - simplicity and low cost.
  - It is preferred evaluation technique at the preliminary design stages

The Ten Heuristics as put forth by Nielsen.

1. **Visibility of system status**
2. **Match between system and the real world**
3. **User control and freedom**
4. **Consistency and standards**
5. **Error prevention**
6. **Recognition rather than recall**
7. **Flexibility and efficiency of use**
8. **Aesthetic and minimalist design**
9. **Help users recognize, diagnose, and recover from errors**
10. **Provision of Help and documentation**

Visibility of system status

Users need to be kept informed by the system about what is going on, through appropriate feedback within reasonable time.

Elaboration

constantly made aware of his/her interaction with the interface while interacting.

The control response ratio (input – output time) need to be as small as possible.

Any interface needs to communicate that it is in a ready state to be operated upon

Most important to users is to know "Where am I?" and 'Where can I go next?" Internal reference is a must to feel in control.

Match between system and the real world

The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real- world conventions, making information appear in a natural and logical order.

Elaboration: Technical jargon or using terms like 'Initiate' or 'Load' in pace of ' Start' contributes to initial mismatch between the users cognitive process and machines feed back dialogue.

An interface need to allow smooth transition from contextual 'reality' world to artificial machine world. ….in other words from ' reality' to 'digitality'.

Tendency to use programming language and syntax on the display, while understandable to the software programmer, will certainly be a mismatch to the user.

Users can come from different backgrounds, skills levels , specializations & culture.
The context on the screen needs to match with the context of the user's mental model

User control and freedom

Users often choose system functions which they did not want . (Mouse click due to haste) . This calls for Support undo and redo.
A user need to have to go through tracing too many steps back to regain control.

Elaboration: Sequential thought process in a user that follows a simple everyday human habit need to be reflected in the dialogue between the device and the user. A good interface facilitates this.

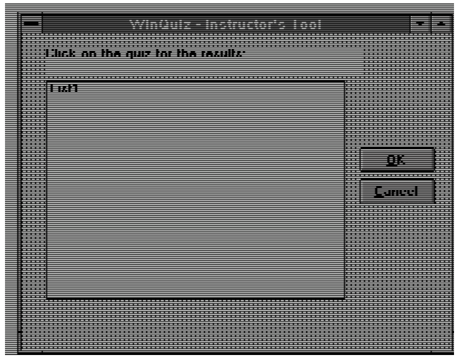
Being in control implies that one can choose to stop interacting an time rather than be forced or trapped by the interface into inaction.
Feeling in the user that he/she is in control at all times must be created.
If the user attempts to gain control and if a message like 404 error occurs the systems is unfriendly & unhelpful !

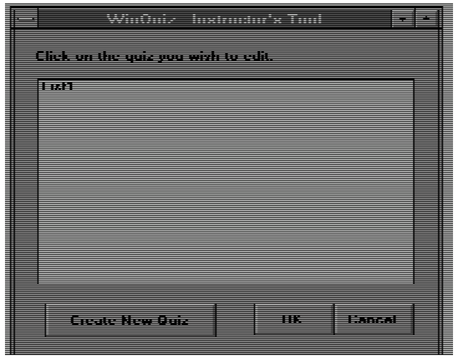
Can users select and sequence tasks? Can they easily re- turn to where they were if they choose an inappropriate /action path? The first example "accuses ' them of committing an error. The second one is much better but does not tell the user what to do next ! The third example is inappropriate !

Consistency and standards

Using different words to mean the same action or using different symbols on different pages can be confusing to the user. Users should not have to wonder whether different words, situations, or actions mean the same thing. They should not be in doubt as to what to do next.

Elaboration: Within an interface if multiple words or actions are used to mean the same thing , it only leads to confusion in the user due to perceived lack of consistency. Interaction pattern gets disrupted. When pattern becomes complex, user's cognitive load increases.
Consistency in dialogue as well as in visual elements is achieved by specifying and adhering to a dictionary of words / labels/ symbols/ colors which together form a 'standard' – a prescribed set – compulsorily to be followed.





Inconsistent wording & windows / buttons can confuse users when the destination page has a different title from the link. The two screens belong to the same software but appear differently at different places within the website.

Strive for Consistency

- There are many form of consistency

Example:

Start **Stop**

Inconsistent Interface Design

Strive for Consistency

- There are many form of consistency

Example:

Consistent interface design

Strive for Consistency

- There are many form of consistency

Example :

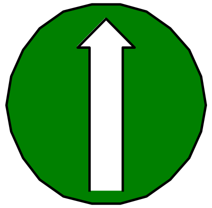
Start **Stop**

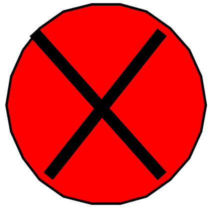
Better consistency in an interface design

Strive for Consistency

- There are many form of consistency

Example :





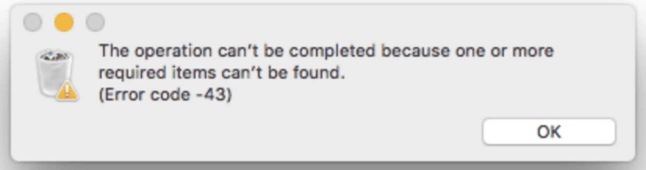
Start **Stop**

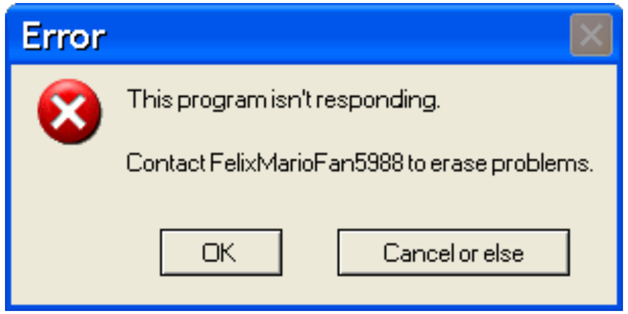
Another better consistency in an interface design

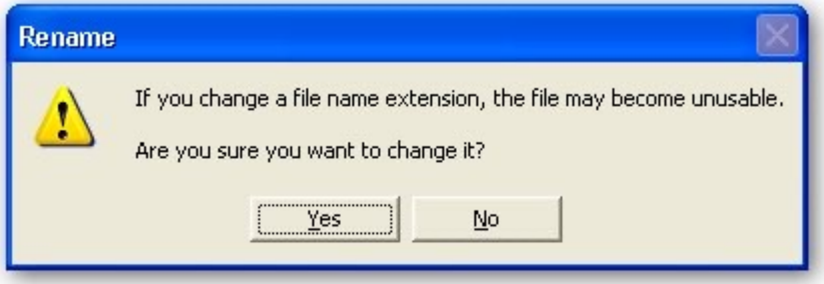
Error prevention

Prevention of error is best approach. However recovery from error prone actions through a well designed error message should be adopted.

Elaboration : To err is human. Errors can happen regardless the level of expertise of the user or familiarity of the interface. A good principle of design is to seek out error prone interactions , build in error prevention within the dialogue. Forewarning, restricting , prompting , retracing or recovery routes , etc are means of addressing errors. Errors lead to a situation wherein users feel subdued by a machine. Anticipating for errors and incorpo
preventive measures ensures fear free and ego free user there
importance to ‘H’ in HCI through ‘I’







GUI-style widgets cut down on the errors but may still have to be double checked before confirmation

Recognition rather than recall

*Loading the STM- short term memory of the user beyond a limit has negative consequences. Given a navigation path, a user need not have to remember or recall all the instructions. Users are better at **recognising** things they have previously experienced. Prompts , visibility , sequential direction, pop-ups etc should come to the aid of the user. Help needs to be easily retrievable.*

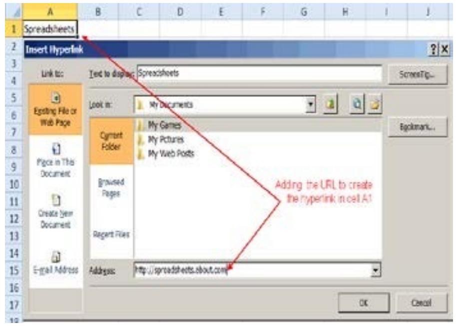
Good labels and descriptive links are crucial for recognition.
The first two icons are difficult to recognise or to recall.

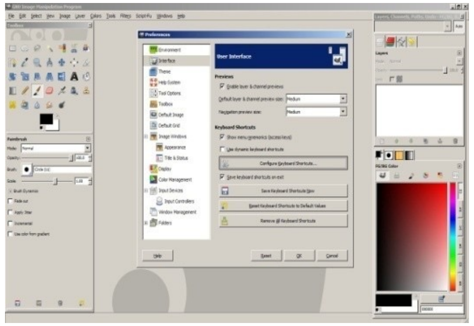
Elaboration : Reduction on cognitive load during the interaction ensures
that the user is not asked to rely on means and methods that extract human cost. If an interface requires specilised training and use of memory to operate - it will be quickly abandoned by the user.
Analogy, metaphor, symbols, sounds, etc are used as design elements in an interface to ease recall thereby eliminating the need for ' thinking while interacting' and memory loads for the user.

Flexibility and efficiency of use

The system can cater to both inexperienced and experienced users. As the user becomes proficient - shortcuts can be encouraged. Thereby increasing the efficiency. Allowing the rearranging of the screen elements by the user can also be adopted.

Elaboration : Once a user becomes adept at using an interface , he/she upgrades into a higher level user from a novice. Such users will always seek to complete the task faster . Such users seek out shortcuts. An interface need to allow this. It needs to be flexible and make it possible for the user to adopt quicker dialogues through shortcuts. The user feels efficient as well as proficient. The feeling of having mastered the software is a flexible sign of being in control thereby.





Advanced users can opt for shortcuts in the spreadsheet example in the first picture.

Flexibility of keeping the required buttons / sections in view or hiding them gives the option to the user to rearrange GUI as needed as shown in the second picture.

Aesthetic and minimalist design

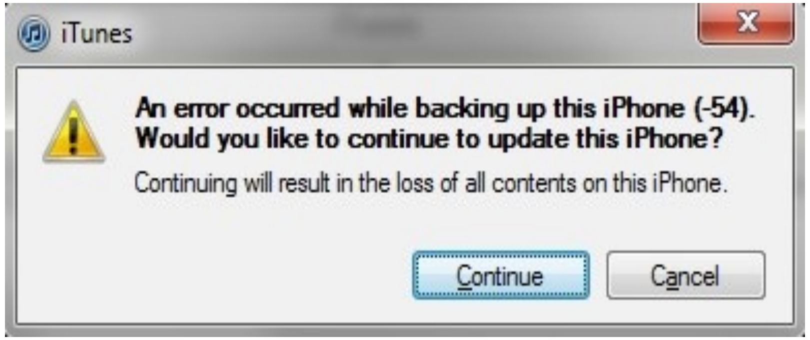
Relevancy, simplicity, minimum amount of labels, un cluttered graphics result in efficient communication dialogue between the user and the interface unnecessary superfluous elements need to be dropped.

Elaboration : Visual clutter in the interface only adds to inefficiency however impressive it is visually. Simplicity is equal to efficiency is equal to elegance is equal to beauty is the aesthetic algorithm in minimalism. Use of least number of elements (minimalism) is more ‘scientific’ rather than ‘artistic’. **Visual noise needs to be completely eliminated.**

Help users recognize, diagnose & recover from errors

Preventing a user who is about to make a error would be a good approach. Gentle wording of error messages, constructive suggestions, reeducating the user- all can contribute to a happy self confident user who is not afraid of being caught unawares or penalized.

Elaboration : No body likes to be loudly informed that he/ she has erred. Error messages need to be disused as suggestions / prompts and precise instructions so as to be able to correct the error and recover. The learning component in errors so that the user recognizes the error as it is being made, or recognizes the reason why the error happened in the first place – helps the user learn.



There is no way to understand the consequences of canceling. The onus seems to be on the user who will be held responsible for what ever is opted for. proper diagnosis & how to possibly recover is not clear. Very unfriendly interface./

Help and documentation

Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Help quarries need to be answered promptly without the user having to go through an elaborate eliminating list.

Elaboration : This again is to assist the user *learn* and understand the dialogue between the user and the machine or understand - where what went wrong - or aid recall during memory-lapses due to long usage time gaps. Adequate 'Help' support system when the user wants and at the point where the user wants it - is a good principle of Interface design.

The screen shots (1&2) attempt to Train the user by o09ffering information on the consequences of their decision

Conclusions:

- These ten heuristics of usability help in refining a potential design into a good design. They ensure that interfaces evolve in the right direction.
- These rules of the thumb act a check list to evaluate a design.
- They also can be used as check list while evaluating any GUI.

Paper discussion

What If Conversational Agents Became Invisible? Comparing Users' Mental Models According to Physical Entity of AI Speaker

SUNOK LEE, Department of Industrial Design, KAIST, Republic of Korea

MINJI CHO, Department of Industrial Design, KAIST, Republic of Korea

SANGSU LEE, Department of Industrial Design, KAIST, Republic of Korea

Proceedings of ACM Interactive Mobile, Wearable Ubiquitous Technology, ver 4, issue 3, Article 88 (September 2020),

Context of the problem

- Invisible intelligence
  - Physical disappearance of computers from users' view
  - Miniaturization of devices and their integration into other everyday artifacts or the home environment
    - to act as intelligent agents that use prediction algorithms to predict behaviors

- How can people perceive and interact with invisible devices?
- What other advantages do invisible CAs have compared to CAs with physical entities?
- How can sensor-based, invisible interfaces be designed for implicit interaction with humans?

Identify users’ mental models and determine the design direction of the invisible CA.

Method

- Conducted a drawing study
  - to understand the differences between the two CAs with different physical entities
  - how their physical visibility affects users' mental models and interactions.

- 30 participants interacted with CAs with a physical presence and an invisible CA
- Differences :
  - Mental models and interactions
- Aim to
  - Understand users' perceptions
  - Expectations of the role of the agent

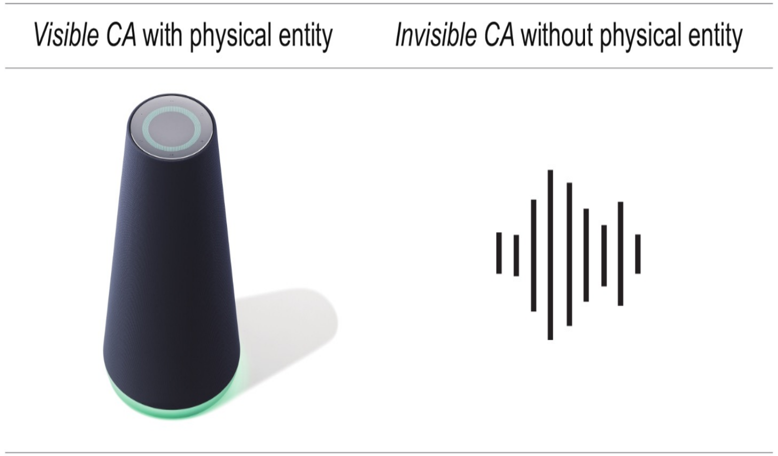


Fig. 1. Equipment used Clova

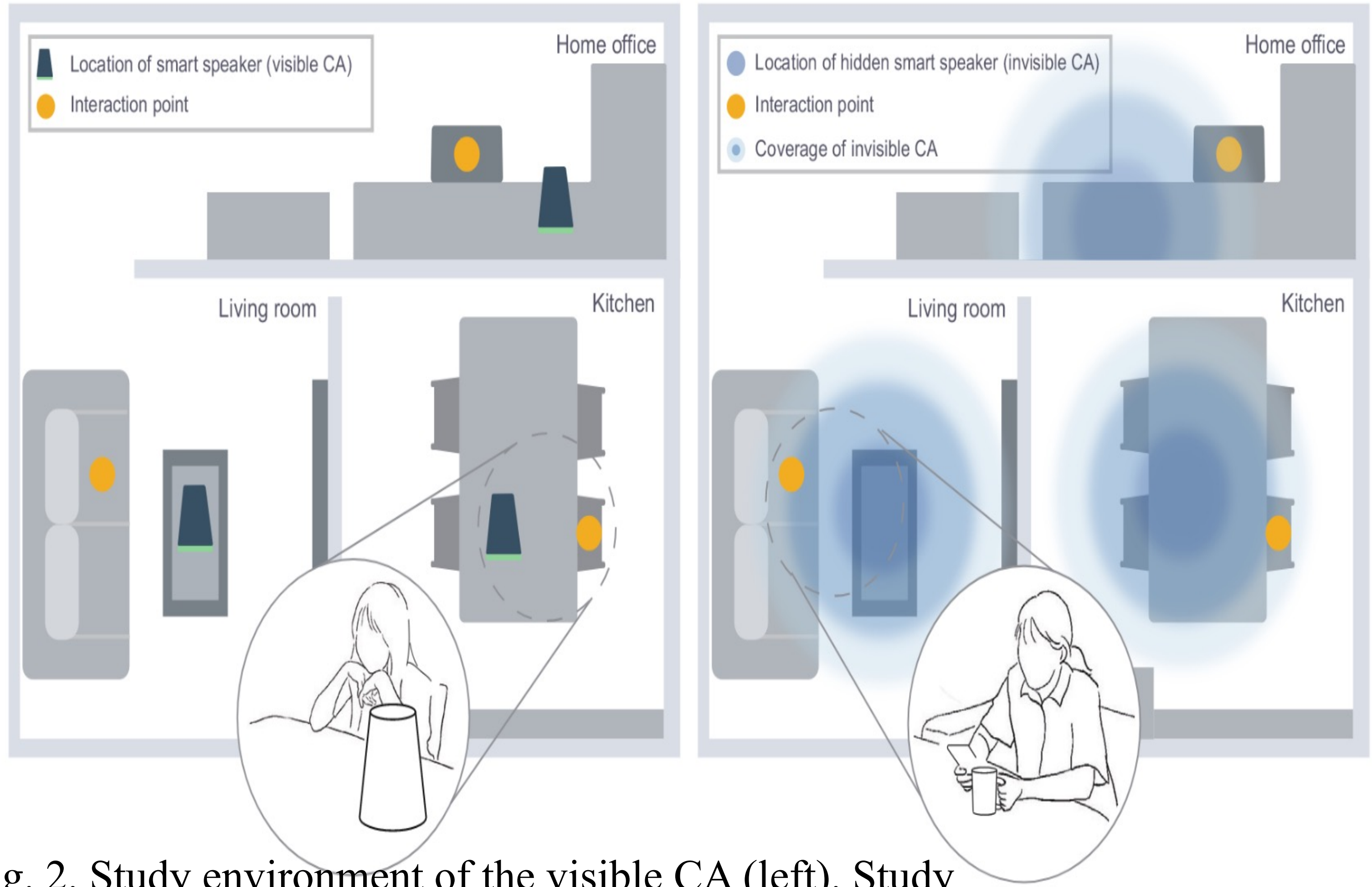


Fig. 2. Study environment of the visible CA (left). Study environment of the invisible CA (right).

Interaction Tasks

- The study consisted of four sessions:
- 5-minute tutorial
- 30-min interaction
- 40-minute drawing,
- Debriefing interview

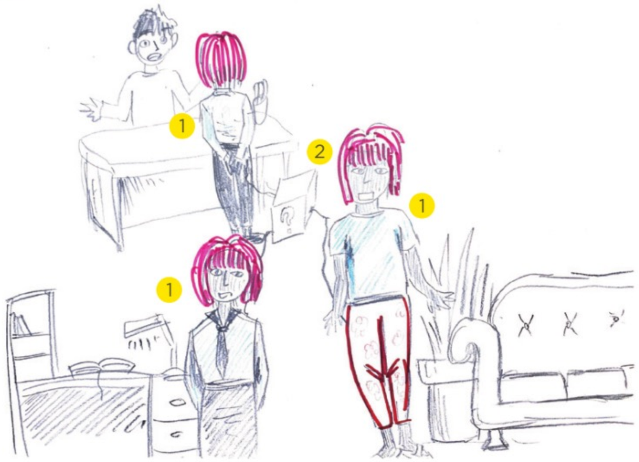
Catergory of instruction	Command examples
Music/Audio	“Play a song that suits on a rainy day” / “Read a book”
Daily information	“How's the weather today?” / “Tell me today's news”
Shopping / Delivery	“Please order paper towel” / “Please deliver pizza”
Schedule management	“Wake me up at 7 in the morning every day” / “Tell me about today's schedule of mine”
Search	“Give me your US country code” / “How old is the earth?”
Smart home	“Turn on or off the TV” / “What is the current temperature of the air conditioner?”
Others	"Hi!" / "Sing me a song"

Table 1. Printed instruction in seven categories

Findings from Drawings – visible CA

- Participants perceived that interacting with the three AI speakers
  - As talking to three individual agents
  - because even though the devices had the same form, wake-up words, and voice in the experimental setting, the users perceived each agent differently

D1 Three agents with different appearances for each space.



1. The triplets are wearing different clothes depends on the each role.
2. The triplets are connected to a box with a question mark.

D8 Three agents with different appearances for each space.

1. Agent in the kitchen is singing bird in the kitchen
2. Agent in home office is Mechanical robot
3. Agent in living room is a person who speaks happily

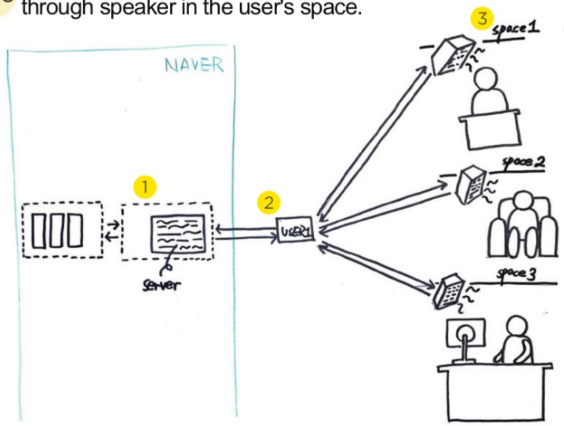
Although there are three devices of the same shape, each one seems to be different depending on where they are placed individually. (P8)

The devices were separate, so I perceived them as different beings. (P10)

Invisible CA

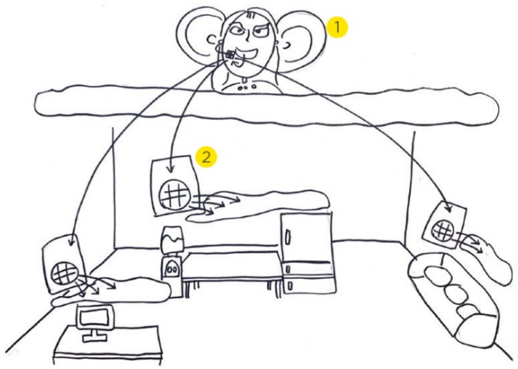
- Perception: Even though 3 diff agents as a single agent
- Drawings depicted One agent controlling the entire space.
  - Because the voice of the agent was the same everywhere
  - Device was not visible in any of the spaces
  - Focus was more on the consistency of the voice regardless of the number and location of the interactions.

D16 The agent is in the headquarter server and interacts through speaker in the user's space.



1. Agent in the server at the headquarter.
2. The agent interacts with the hub containing user data.
3. Users communicate through speakers designated for each space.

D21 Response of agent with large ears is delivered to the speaker through the microphone.



1. Agent is beyond the ceiling of the house and has big ears to listen to the user
2. The answer delivered through the speaker was expressed in a transparent cloud shape.

I feel like there is a single agent connected to the server in the sky. Actually, rather than having three agents, I feel like one agent can go anywhere. (P17)
I think this is one person. Because it doesn't change the tone, it doesn't change the voice, and the personality is not different (....) The reason I drew the big ears is because I feel like she is listening to me very carefully with huge ears. (P21)

Mental Model – Visible CA

- Perceived three agents that each play a role with own expertise depending on the space
- Smart agents in the home office and as more active and bright agents in the living room.
- Users' perceptions of individual agents for each device were naturally connected to different personas, roles, or functionalities
- Negative impressions of interacting with multiple agents.
- If environment were a real home
  - a mental load would be required to talk to several agents in their private spaces.
  - Interaction was similar to communicating with people
    - where more effort is required to get to know several people than it is to know one person.

I feel like I have to try to get to know this [agent] through conversation, but if there are several of them, it is a huge burden that I have to try to get to know three or four people. (P8)

Even if different agents communicate with each other through something like a hub, it's likely to be creepy or less reliable because I'm afraid that these agents will communicate my private information that I don't want to share with other people.

Mental Model – Invisible CA

- Only one agent
  - No change in tone/voice
  - User’s location of communication irrelevant
- Could detect a user’s behavior and status
  - without space limitations
  - Expected the agent to provide the most appropriate information based on its awareness of the user’s context
- Some Users took personalized responses from agents for granted.
  - Contrary to the mental model for the visible agent, where participants were concerned that their conversation records were shared with agents in other spaces.

On the sofa, when I said to the agent, “I was having a hard time today,” and then when I moved and sat at the desk and asked it to play music, the agent said, “I’ll play music that feels good.” That is so touching even though it was a brief moment, I felt like I was emotionally interacting with this agent. (P28)

This [agent] knows all of the things I do, but I don’t want my information to be shared with the next-door neighbor or headquarters like Google. Then, I will never believe this agent and will never use it

Interaction Styles

Visible CA

- Direct – eye contact
  - Device installation location
- Visual Cue – LED
  - Start of device , ready to listen
  - Learnability Increased
    - User knew when to speak
    - Discoverability of the Voice UI

Invisible CA

- ambient interaction without space constraints
- No physical presence
  - Only auditory cues
  - Less affordances
    - Initial interaction issues
- Did other tasks while communicating with agent – Talk and listen anywhere

I felt comfortable because I had someone to talk to. (P9)
This form makes me feel like I'm having a conversation with someone. If this disappears, it would be awkward because I would be talking to nothing, which would definitely not be a natural conversation (P10).

When the light comes on, I can see that the agent is listening to me, and if there is no visual feedback, I wouldn't be really sure. (P8)

Ambient Interaction

- Invisible agent has the advantage of
  - enabling parallel task execution,
    - because in the case of an agent without a device, it is not necessary to look at it, so subtasks can be carried out while the user continues to do his or her previous or main work.

When I can see it, I have to consciously stare and talk, and keep checking to see if the agent is listening to me or not, and I guess this will interfere with usability. But, if I don't need to see the agent, then I can do two things at the same time.

I felt uncomfortable and a sense of cognitive discrepancy with talking to an invisible agent because there's a sound, and something is coming from somewhere, but there is nothing—just empty space. (P21)

This is uncomfortable because Siri lights up but this one does not. I have no idea how loud I should speak and where to speak. If this is visible, I can know how I should talk to the agent, but I don't know where it is... (Is it under there or up there?) I don't know how loud I should speak. (P16)

Conclusions

- the physical entity's visibility influenced
  - the mental model of the user,
  - Played an important role in the User Experience.
- ambient interaction naturally connected to a user's perception
- expectation was
  - Interaction could be with personalized agents everywhere,
  - Similar to the goal of ambient intelligence
    - Example: JARVIS

IE 403

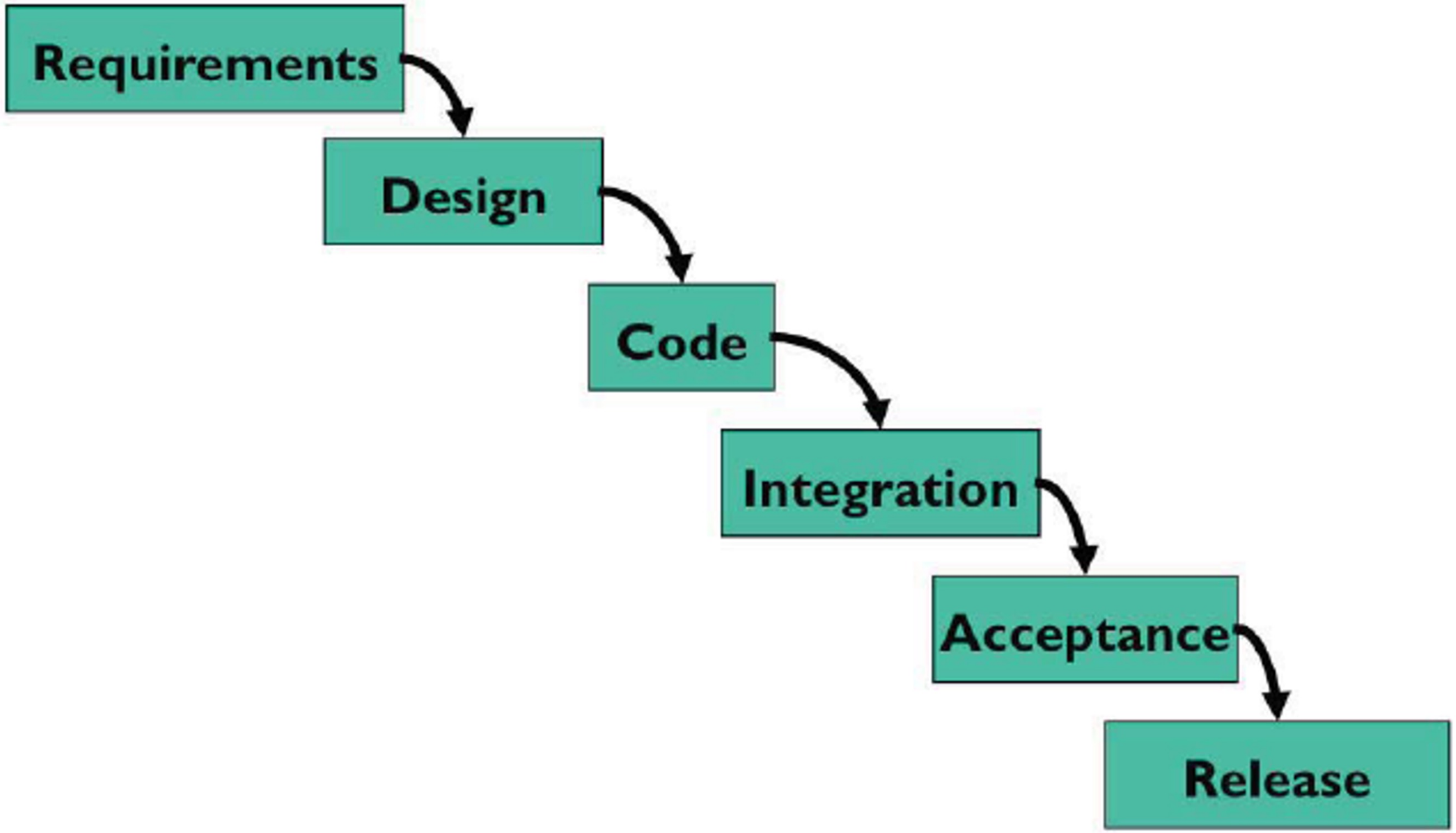
Human-Computer Interaction

Week 4-Lec3

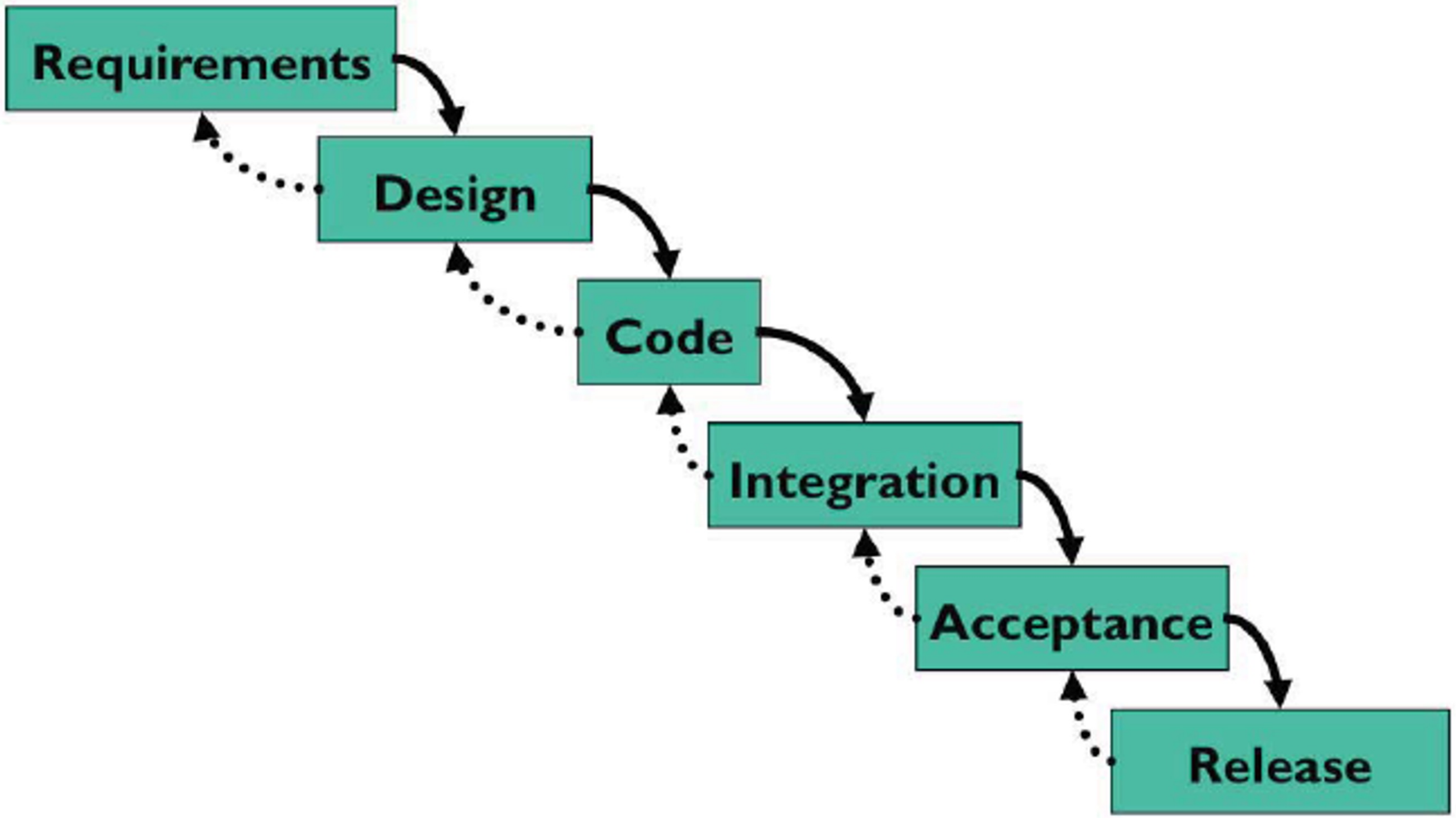
Agenda

- Iterative Design
- User centered Design

Waterfall model



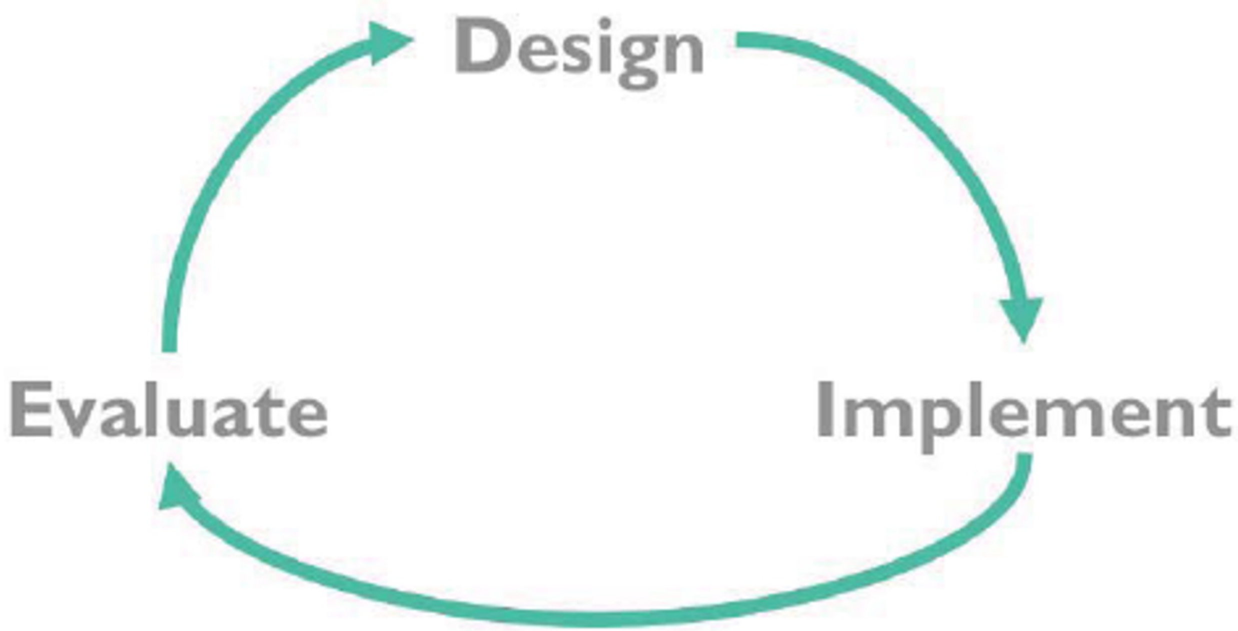
Feedback in Waterfall model



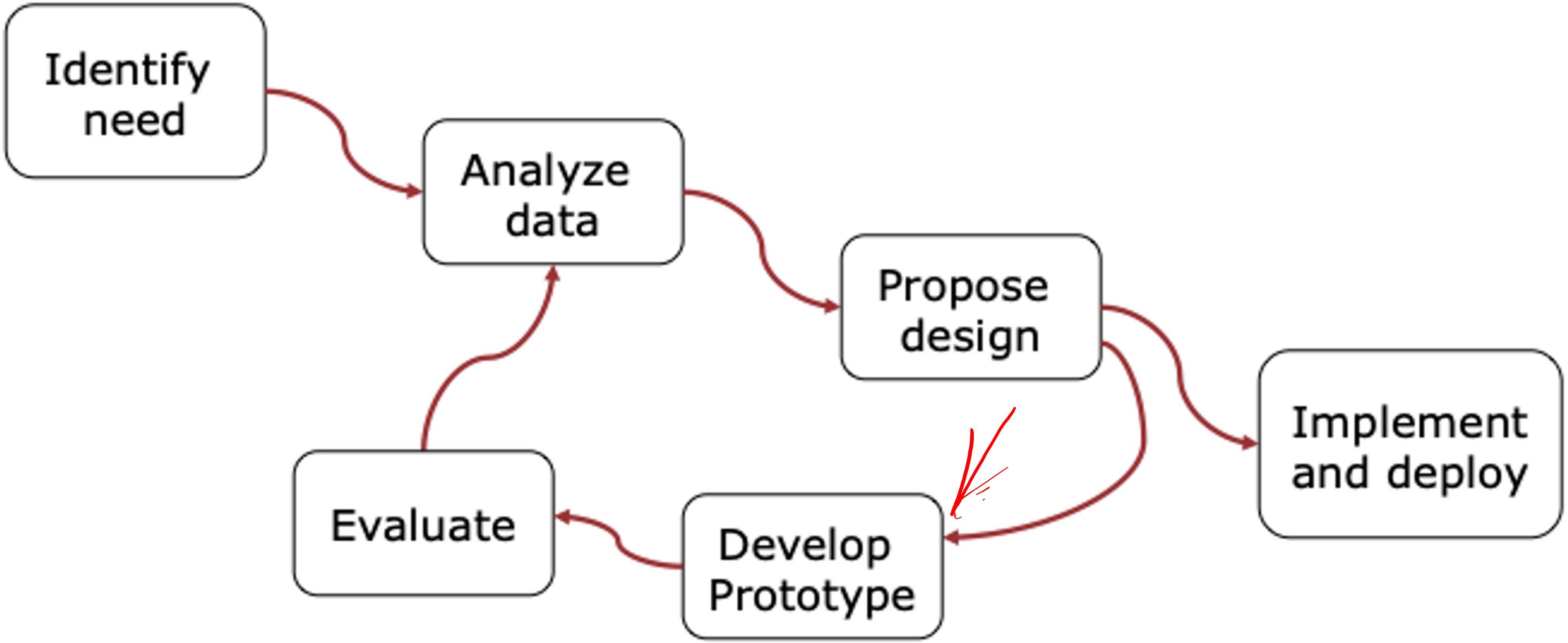
Why waterfall model bad for UI design

- UI is Risky - probability of getting wrong high
- No involvement of users till acceptance testing
- UI makes sudden changes in requirements and design

Usability engineering - Iterative Design



UCD Lifecycle



Identify Need

- Interview (structured, semi-structured, unstructured)
- Contextual inquiry
- Cultural probes
- Ethnography
- User models

Analyze Data

- Scenario analysis : User usage
- Task analysis : tasks reqd to be performed by user

Prototype dev

- Low Fidelity: paper based
- High Fidelity: Tools/software

Evaluate

- Heurisic
- Usability eval
- Full eval

Case Studies examples

Olympic Messaging system (1984)

- Iterative design
  - 200 Iterations for user guide
- Cheap prototypes
  - scenarios
  - user guides
- Absence of UCD
  - language barrier - telephone prompts

https://www.youtube.com/watch?v=W6UYpXc4czM

Microsoft Interactive Program Guide

- Embedded interactive TV application inside set top box
- Objective
  - Simple
  - Low level inputs
  - Familiar and usable interface
  - Accurate
  - Context based - what was entered

User Tasks listing

- Changing channels/channel surfing
- Finding out the name of the TV program currently being watched what it was about
- Searching for a specific TV program
- Setting a list of favorite channels
- Finding child-appropriate cartoons
- Setting parental controls

Procedure

- 13 core tasks to complete by users in lab
- Task performance for benchmarking
- Track progress of UI design over a year

STUDY #1 FINDINGS	ACTION TAKEN
Expected to be able to specify a time rather than a timeframe in Search Three participants wanted the ability to specify specific times rather than a timeframe in the Search by Time feature. These participants also indicated that the labels Morning, Afternoon, Night and Late Night were ambiguous because they weren't certain if 11:00 pm constituted Night or Late Night.	Changed search capabilities from 4 generic categories to specific hours of the day to allow user to specify a specific time. Bug # A21B5

STUDY #2 FINDINGS	FURTHER RECOMMENDATIONS
Able to successfully search using By Day and Time label	
All participants were able to correctly search on a specific day (tomorrow) at a specific time for a specific program using the By Day and Time label. All participants easily picked out the day and time to complete the task.	Continue to allow users to specify a specific day and time to complete a search.
When asked to search for programs on today, six participants selected the Today label within search, while only one participant opted to use the Browse feature.	No recommendations are necessary.

Usability Findings

Benchmark Study A

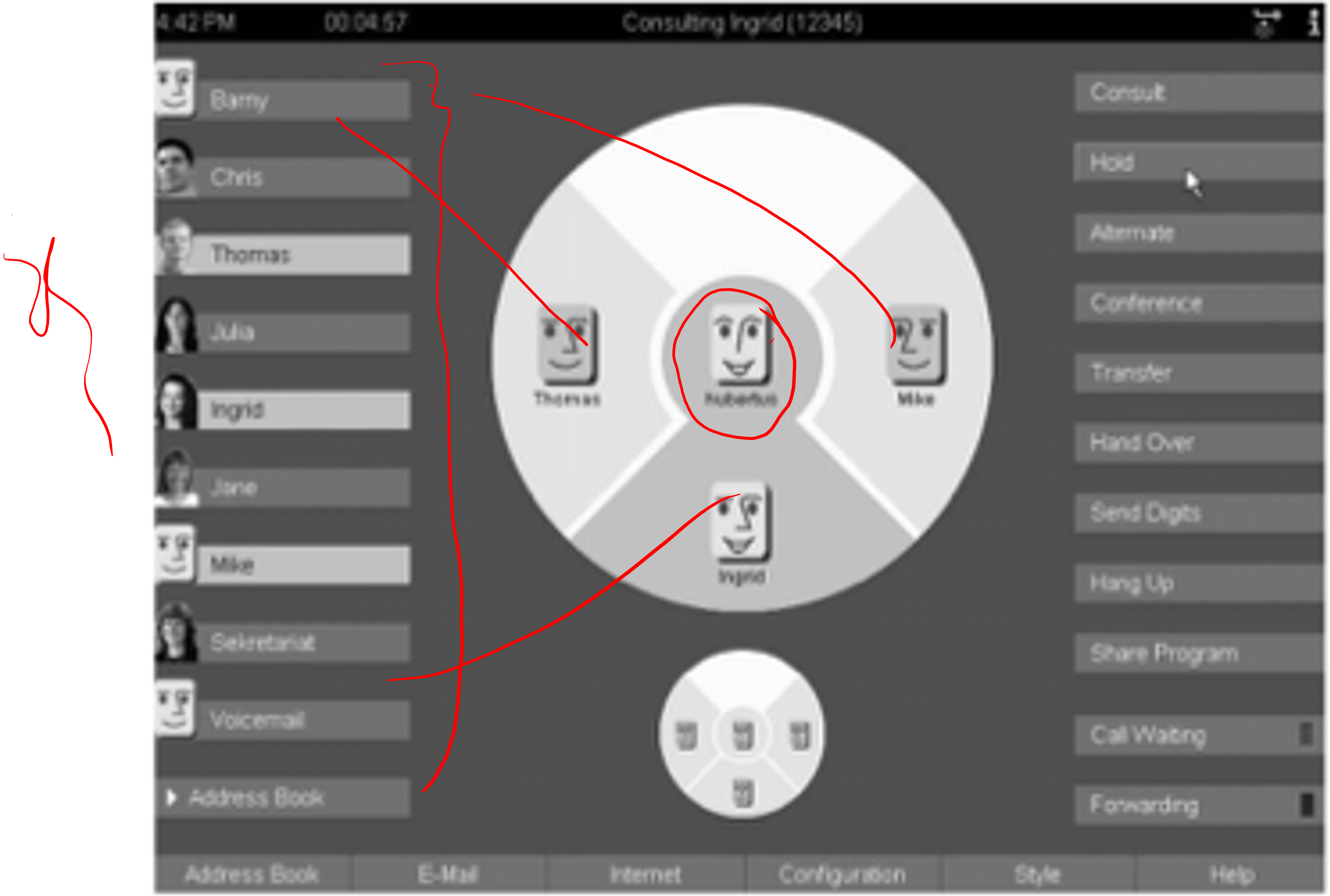
- 33 usability issues were discovered
- 30 of these issues were addressed through a redesign or UI change (to be verified in next study)
- Result – the team is addressing more than 90% of the usability issues found with the user experience

Benchmark Study B

- 15 usability issues were discovered
- 10 of these were new usability issues
- 5 were re-occurring issues – all had dropped in severity from previous study

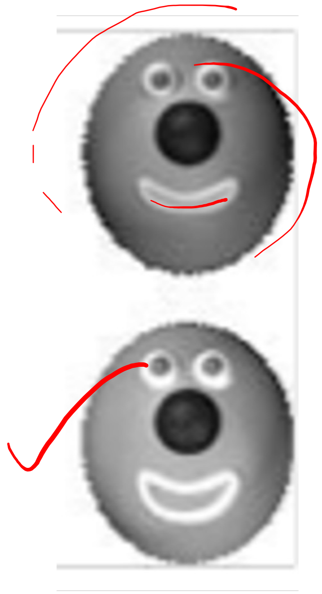
EASYCOM telephony design

- more supplementary features are supported
- the more difficult these phones are to operate
- integrating voice and data communication functions into a single PC-based user interface,
- Efficiency of employee workflows
- Conceptual model
- communication circle
- USer at the center
- drag, drop communication partners
- Direct manipulation



4:42 PM
00:04:57
Consulting Ingrid (12345)
Barry
Chris
Thomas
Julia
Ingrid
Jane
Mike
Sekretariat
Voicemail
Address Book
Thomas
Hubertus
Mike
Ingrid
Consult
Hold
Alternate
Conference
Transfer
Hand Over
Send Digits
Hang Up
Share Program
Call Waiting
Forwarding
Address Book
E-Mail
Internet
Configuration
Style
Help

- improve the handling of complex synchronous communication processes by striving for ease of use
- designing for fun and emotion has become an equally important success factor
- observations are based on user feedback collected in internal and external field trials as well as interviews with customers and sales people.
- reduced Anxiety in making conference calls which were complex
- Pleasant UI
  - innovative
  - USP for Siemens VOIP products



GUI: avatars or talking heads

How to read a research paper

Abstract

What If Conversational Agents Became Invisible? Comparing Users' Mental Models According to Physical Entity of AI Speaker

SUNOK LEE, Department of Industrial Design, KAIST, Republic of Korea
MINJI CHO, Department of Industrial Design, KAIST, Republic of Korea
SANGSU LEE, Department of Industrial Design, KAIST, Republic of Korea

The popularity of conversational agents (CAs) in the form of AI speakers that support ubiquitous smart homes has increased because of their seamless interaction. However, recent studies have revealed that the use of AI speakers decreases over time, which shows that current agents do not fully support smart homes. Because of this problem, the possibility of unobtrusive, invisible intelligence without a physical device has been suggested. To explore CA design direction that enhances the user experience in smart homes, we aimed to understand each feature by comparing an invisible agent with visible ones embedded in stand-alone AI speakers. We conducted a drawing study to examine users' mental models formed through communicating with two different physical entities (i.e., visible and invisible CAs). From the drawings, interviews, and surveys, we identified how users' mental models and interactions differed depending on the presence of a physical entity. We found that a physical entity affected users' perceptions, expectations, and interactions toward the agent.

CCS Concepts: • **Human-centered computing** → **User studies**.

Additional Key Words and Phrases: Ubiquitous smart home, conversational agent, voice user interface, invisible intelligence, drawing study

ACM Reference Format:
Sunok Lee, Minji Cho, and Sangsu Lee. 2020. What If Conversational Agents Became Invisible? Comparing Users' Mental Models According to Physical Entity of AI Speaker. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 4, 3, Article 88 (September 2020), 24 pages. https://doi.org/10.1145/3411840

1 INTRODUCTION

Introduction

1 INTRODUCTION

Due to its seamless and unobtrusive input method, the use of voice user interface (VUI) has increased in the form of AI speakers as hubs to control smart homes [11, 39, 45]. Because commercialized AI speakers consist of a physical device with a speaker and microphone, users need to face the device for better communication, as people do when they communicate face-to-face. Users interact with CAs in stand-alone devices, such as Alexa in Amazon Echo, which makes users look at where the device is installed and listen in the direction of the sound coming from the device. This means that the conversational agent (CA) in the physical device leads to visually and auditorily directed interaction. However, because the directed communication is based on where the device is installed, there are limitations in the agent's immediate interaction with the user, and it is difficult to notice the changing context of the user in real time at home [34].

To find answers of these questions, it is critical to identify users' mental models and determine the design direction of the invisible CA. Similar to previous studies on CAs embedded in AI speakers, we emphasized the importance of understanding users' mental models for designing CAs [7, 11, 16, 28, 30, 39]. These studies attempted to understand users' mental models mainly through verbal approaches [11, 30]. However, because it is unclear how each user will accept the new concept of invisible agents, when forming a mental model toward the unsubstantial system by comparing it with current CAs in physical devices, limitations can hinder discovering users' expectations for a new concept with only a verbal approach. For this reason, it is necessary to explore users' mental models of invisible agents and to find out the difference between existing CAs in physical entities—such as AI speakers—for future design possibilities and development directions combining visual and verbal approaches.

Thus, we conducted a drawing study to understand the differences between the two CAs with different physical entities, as well as how their physical visibility affects users' mental models and interactions. To achieve that goal, 30 participants interacted with CAs with a physical presence and an invisible CA, and we then identified the users' mental models through their drawings, in addition to interviews and a survey. We explored how users' mental models and interactions are different in directed interaction with a visible agent than they are in ambient interaction with an invisible agent. The findings describe users' perceptions, expectations of the role of the agent, and interaction in terms of physical entities to discover the potential of the invisible agent. Based on this finding, physical presence is an important element in CA design, and we found features of invisible agents compared with CAs in AI speakers that were not revealed in previous studies. Finally, this study suggests that invisible agents can be one design direction in VUI to enhance user experience in the ubiquitous environment.

Methodology

3 METHOD

3.1 Participants

To grasp users' first perceptions of visible and invisible CAs, we recruited 30 novice users (avg. age = 22.5 years old, SD = 3.32; 13 females and 17 males) who had never used an AI speaker. Half were to talk to a visible CA and the other half were to talk to an invisible CA. We posted in online communities, such as a university one, to recruit people who wanted to use an AI speaker for their homes. We tried to recruit relatively young participants, even if awareness of users of different ages is important, because 34% of U.S. 18–29-year-olds owned a smart speaker at the beginning of 2019 [51]. This is significantly higher compared to other age groups. Therefore, we thought that this age group would use this type of device in the future. We also focused on first-time users because the initial stage of CA use would affect users' mental models and later stages of use [12, 28, 36], and users in the initial stages of voice interaction need the most assistance to be led to proper use [11]. To compare clearly the new concept of invisible CA with that of the existing visible CA devices, users experienced with neither were more appropriate. Moreover, according to previous research, the first image of an agent rarely changes, being imprinted in users' minds as soon as they hear the agent's voice [36]. Therefore, we would be able to grasp the unbiased perceptions that users formed by the experimental setup we provided. All of our participants were Korean, and we conducted the study in Korean with native Korean speakers. Each participant was compensated with approximately U.S. $20.

3.2 Apparatus

Because the goal of this study was to compare natural user experiences with visible and invisible CAs, we used three commercialized AI speakers and invisible agents in the study environment. Figure 1 shows the apparatus that our study used. A CA without a physical entity—that is, an invisible agent—has not yet been commercialized, so we hid a stand-alone AI speaker in the study environment to make it invisible. Because the goal of the invisible agent is to assist users without visually distracting them from a task, we adjusted the volume and position during the setup process so that the participants did not know where the sound was coming from, causing them to feel as though the invisible agent was ambient. The participants accordingly perceived the sound as coming from the ceiling, rather than from where the agent was actually installed. Three AI speakers were installed in the lab environment in the case of visible devices, and three devices were hidden in the case of invisible agents.

In terms of AI speaker types, we used AI speakers from Naver Clova [38] which are some of the most advanced

3.3 Study Environment

Figure 2 shows the study environment, which includes the location of each device and interaction point. We did not specify the interaction point clearly for the participants but had them interact at whichever point was comfortable in each space. The environment reflected a smart home in which people have multiple speakers dispersed to increase the coverage of a voice agent. Referring to prior research on common locations for Alexa devices [45], we configured the lab environment into three spaces: a kitchen, a living room, and a home office. To provide the sense of a comfortable home and typical environment for AI speaker usage, we placed a sofa and table in the private room, a table and TV in the living room, and a desk and computer in the home office. The

Results

4 FINDINGS

From the qualitative analysis, which mainly focused on comparing each agent, we found that participants perceived CAs with physical entities and invisible CAs differently. In particular, participants' drawings showed how the different physical entities of the CAs affected their perceptions (Figure 3 and 5). The data from observations and interviews allowed us to grasp the mental models and interactions linked to the users' perceptions. In addition, for the survey data, we analyzed the average of the overall CA experience trends for the two physical entities (Table 2). Through findings related to perception, the mental model, and interaction, we can discover the unexplored features of invisible CAs compared with CAs in the stand-alone device. In next section, we refer to each drawing as D and the participant who drew it as P. Both have the same number.

IE 403

Human-Computer Interaction

Week 5-Lec1

Recap

- Cognitive Model
- Gulf of execution and Gulf of evaluation
- 7 stages of actions By Norman
- Mental Vs Conceptual Model

Conceptual Models

- Need to first think about how the system will appear to users (i.e. how they will understand it)

- A high level description of
  - the proposed system with a set of integrated ideas and concepts about
    - what it should do
    - behave
    - look like
    - that will be understandable by the users in the manner intended

Understanding a Conceptual Model

- How will the user think about the system? Possibly based on:
  - Data or objects
  - Types of operations (activities) done
  - Metaphors (real world analogies/similarities)
- What kind of interface metaphor, if any, will be appropriate?
- What kinds of interaction modes and styles to use?

Fig 1. Word processor Vs a typewriter

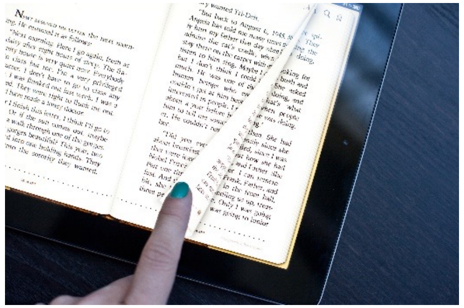


Fig 2. iBooks flipping pages similar to physical books

Mental models

- Users “see” the system through mental models
- Users “rely” on mental models during usage
- Reason about a sys
  - Interact with
  - Infer how it works
  - Figure out how to correct when things do wrong

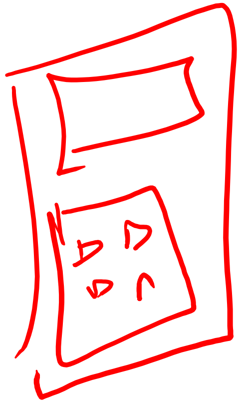
Why are mental models important?

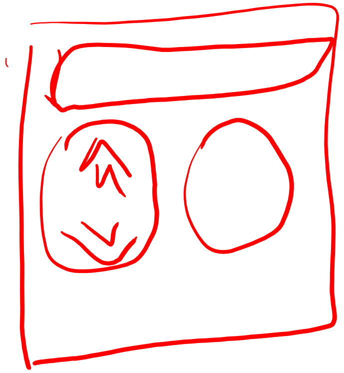
2016, Chrysler automobiles recalled over 100K vehicles

- Gear shows P, R, N, D/S
- Shift through the gear options
- But returns to center position
- What is the problem ?

https://www.endurancewarranty.com/learning-center/news/fiat-chrysler-recall-user-error/

People were getting out of their cars thinking the gear was in Park mode
But it wasn’t, so the car drove off without them!!!





100 crashes

315

40 injuries

$(9+4)\times(3+5)$

Classic Design Flaw

Designing something that people don’t understand or making some thing which is totally NEW & Expecting USERS to figure out
User’s mental model is not the same as what was designed

“1+100%”

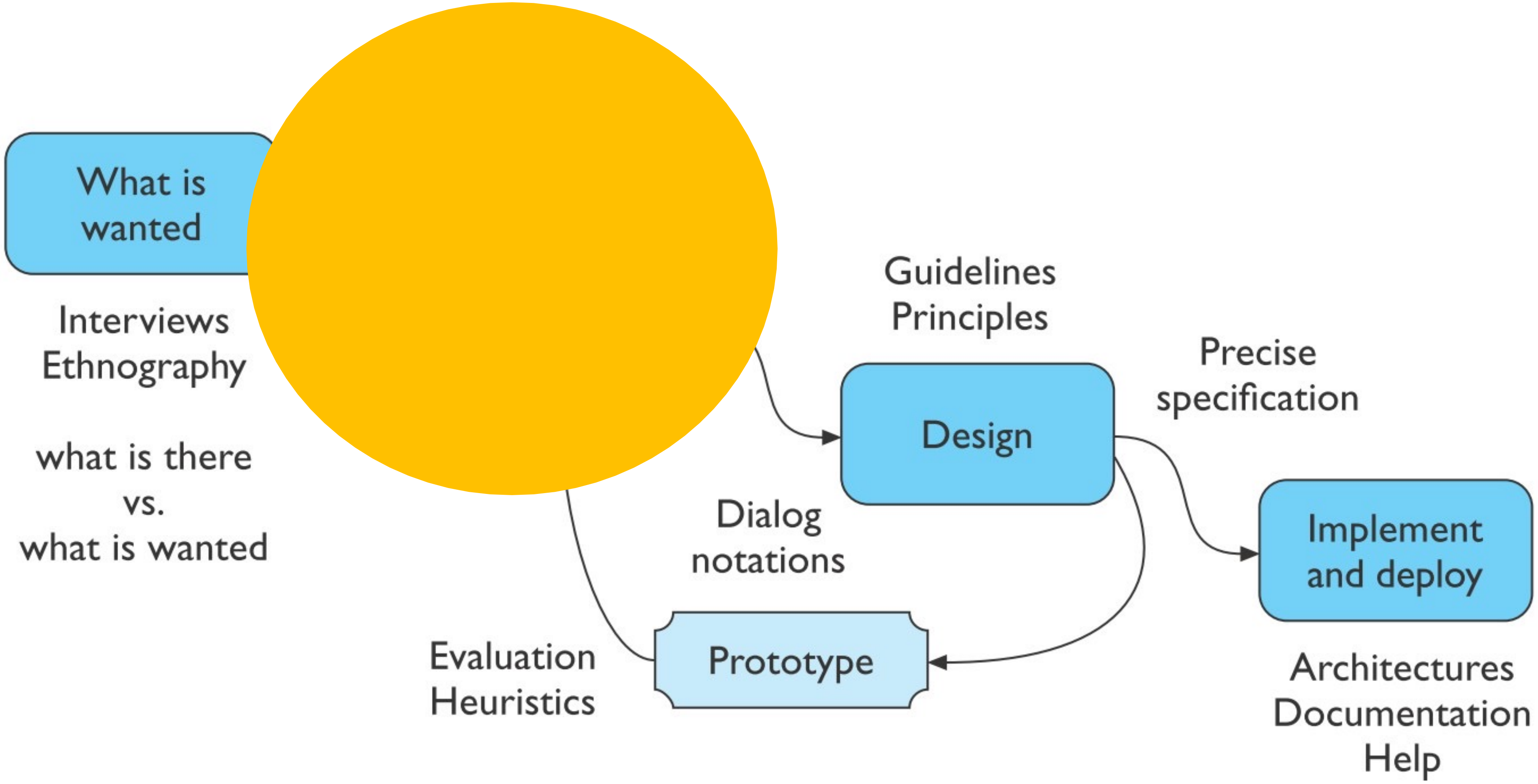
https://www.carcomplaints.com/news/2017/lawsuit-chrysler-gear-shift-confusing.shtml

Task Analysis

Summary

- Task Analysis
  - Purpose and structure
  - Goals, Tasks, Actions
- Hierarchical Task Analysis (HTA)
- Knowledge-based Analysis

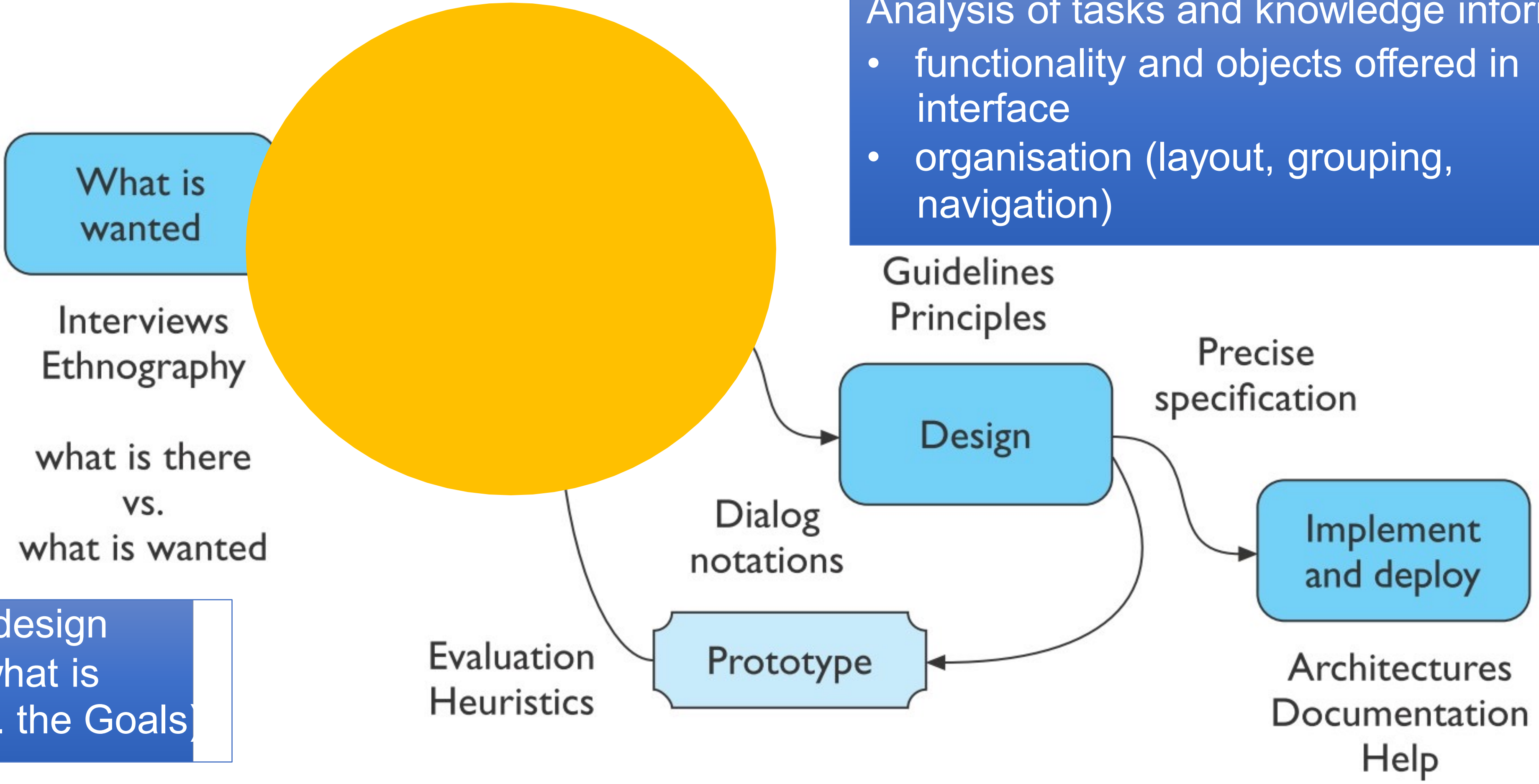
Human-Centered Design Process



Human-Centered Design Process

Analysis of tasks and knowledge inform

- functionality and objects offered in interface
- organisation (layout, grouping, navigation)



Interaction design driven by 'what is wanted' (i.e. the Goals)

Task Analysis

- Task Analysis is the study of the way people perform their jobs.
- Aim is to determine:
  - what they **do**
  - what things they **use**
  - what they must **know**

USER: 1. Clicks button labelled “Start”	SYSTEM RESPONSE: 2. Menu pops up
3. Clicks “shut down”	4. Greys screen and presents dialogue box describing shut down and offering “OK” “cancel” and “Help” options
5. Clicks “OK”	5. Screen blanks and “Please wait” message shown 6. Computer turns off

Example

- Task: to clean the house
  - get the vacuum cleaner out
  - fix the appropriate attachments
  - clean the rooms
  - when the dust bag gets full, empty it
  - put the vacuum cleaner and tools away
- Must know about:
  - vacuum cleaners, their attachments, dust bags,
  - cupboards, rooms etc.

Example

- For example, a person preparing an overhead projector for use would be seen to carry out the following actions :
  1. Plug in to main and switch on supply.
  2. Locate on/off switch on projector
  3. Discover which way to press the switch
  4. Press the switch for power
  5. Put on the slide and orientate correctly
  6. Align the projector on the screen
  7. Focus the slide

What is a Tasks?

- «A **task** is a **goal** together with some ordered set of **actions**.» (Benyon)

Goal

- A state of the application domain that a work system (user+technology) wishes to achieve
- Specified at particular levels of abstraction

Task

- A structured set of activities required, used, or believed to be necessary by an agent (human, machine) to achieve a goal using a particular technology
- The task is broken down into more and more detailed levels of description until it is defined in terms of actions

Action

- An action is a task that has no problem solving associated with it and which does not include any control structure
- Actions are ‘simple tasks’

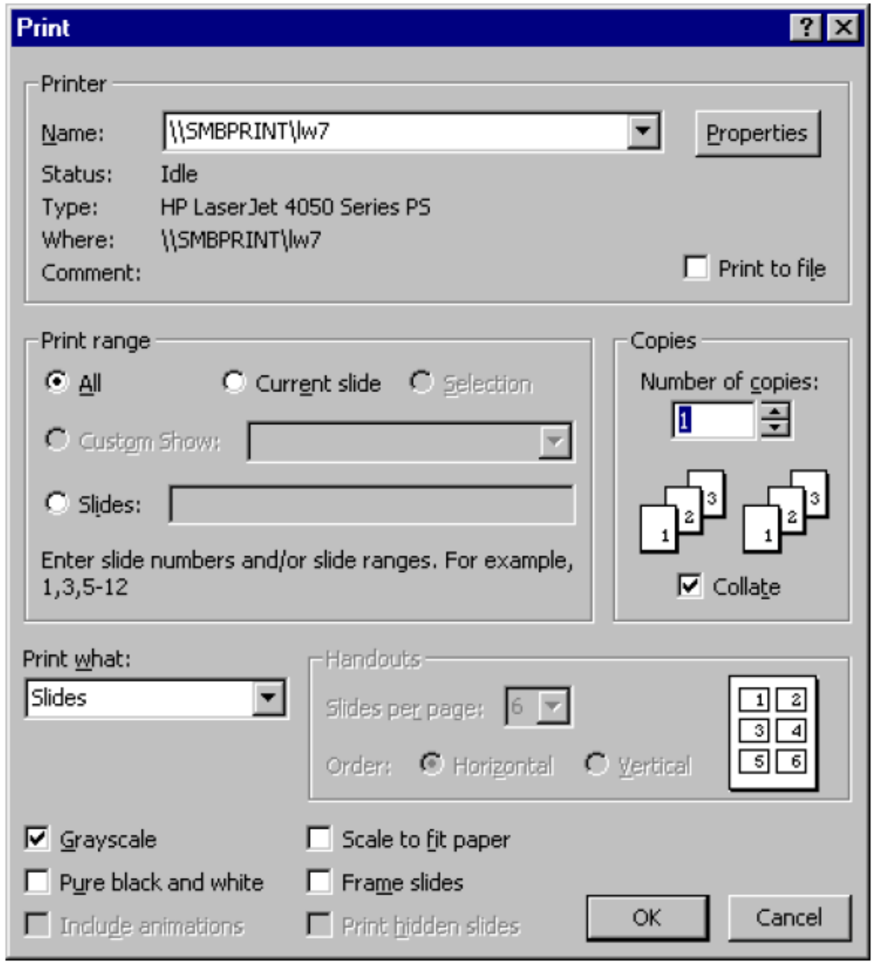
What you learn with Task Analysis

- What your users' goals are; what they are trying to achieve
- What users actually do to achieve those goals
- What experiences (personal, social, and cultural) users bring to the tasks
- How users are influenced by their physical environment
- How users' previous knowledge and experience influence:
  - How they think about their work
  - The workflow they follow to perform their tasks

Why is it useful?

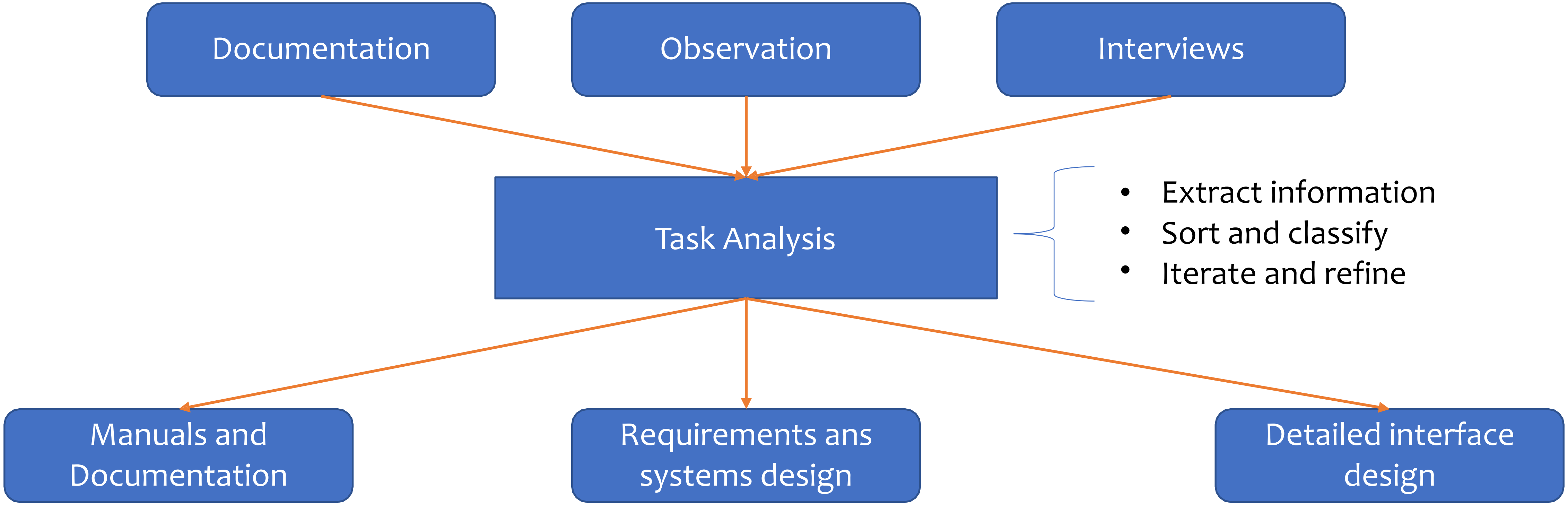
- Task analysis is the process of learning about ordinary users by observing them in action to understand in detail **how** they perform their **tasks** and achieve their intended **goals**.
- Tasks analysis helps in
  - Identifying the tasks that your website and applications **must support**
  - Refining or re-defining your site's navigation or search
  - Website requirements gathering
  - Developing your content strategy and site structure
  - Wireframing and Prototyping
  - Performing usability testing

Example



- Tasks are used to plan for the layout of the application window
- Proximity and Boundaries reflect the decomposition of tasks
- Order of tasks is not mandatory

Where it fits



Manuals and Documentation

- Conceptual Manuals (from KBA)
- Procedural (how-to) Manuals (from HTA)

Requirements ans systems design

- Focus on system usage rather than system features
- Suggests candidates for automation
- Unconvers user's conceptual model

Detailed interface design

- Taxonomies → menu layout
- Objects/actions → interface objects
- Task frequency → default choices

[Some] Techniques for Analysis

- **Task decomposition** – Splitting tasks into sub-tasks and their ordering.
- **Knowledge-based techniques** – Any information and instructions that users need to know, and how that knowledge is organized
- **Entity-relationship-based analysis** – identify actors, objects, relationships and their actions
- **Ethnography** – Observation of users' behavior in the use context.
- **Protocol analysis** – Observation and documentation of actions of the user. This is achieved by authenticating the user's thinking. The user is made to think aloud so that the user's mental logic can be understood.

Hierarchical Task Analysis

Hierarchical Task Analysis (HTA)

- One possible method for Task Decomposition
- Hierarchical Task Analysis is the procedure of **disintegrating tasks into subtasks** that could be analyzed using the logical sequence for execution. This would help in achieving the goal in the best possible way.

> "A hierarchy is an organization of elements that, according to prerequisite relationships, describes the path of experiences a learner must take to achieve any single behavior that appears higher in the hierarchy. (Seels & Glasgow, 1990, p. 94)".

Example HTA: How to clean a house

0. in order to clean the house
    1. get the vacuum cleaner out
    2. fix the appropriate attachment
    3. clean the rooms
        3.1. clean the hall
        3.2. clean the living rooms
        3.3. clean the bedrooms
    4. empty the dust bag
    5. put the vacuum cleaner and attachments away

Plan 0: do 1 – 2 – 3 – 5 in that order.
when the dust bag gets full do 4

Plan 3: do any of 3.1, 3.2 or 3.3 in any order
depending on which rooms need cleaning

- A hierarchy of tasks and sub-tasks
    - Indentation and numbering denote the levels
- A set of **plans** describing in what order and under what conditions subtasks are performed
    - Plans are labeled by the task they describe

Notes

0. in order to clean the house
   1. get the vacuum cleaner out
   2. fix the appropriate attachment
   3. clean the rooms
      3.1. clean the hall
      3.2. clean the living rooms
      3.3. clean the bedrooms
   4. empty the dust bag
   5. put the vacuum cleaner and attachments away

Plan 0: do 1 – 2 – 3 – 5 in that order.
when the dust bag gets full do 4

Plan 3: do any of 3.1, 3.2 or 3.3 in any order
depending on which rooms need cleaning

- Not all tasks are mandatory
  - E.g., task 4 is needed only if the bag is full.
- The order or operations may be free
  - E.g., the rooms may be cleaned in any order
- Could be further refined with additional knowledge or context
  - E.g., Plan 3: do 3.1 every day
    3.2 once a week
    when visitors are due 3.3

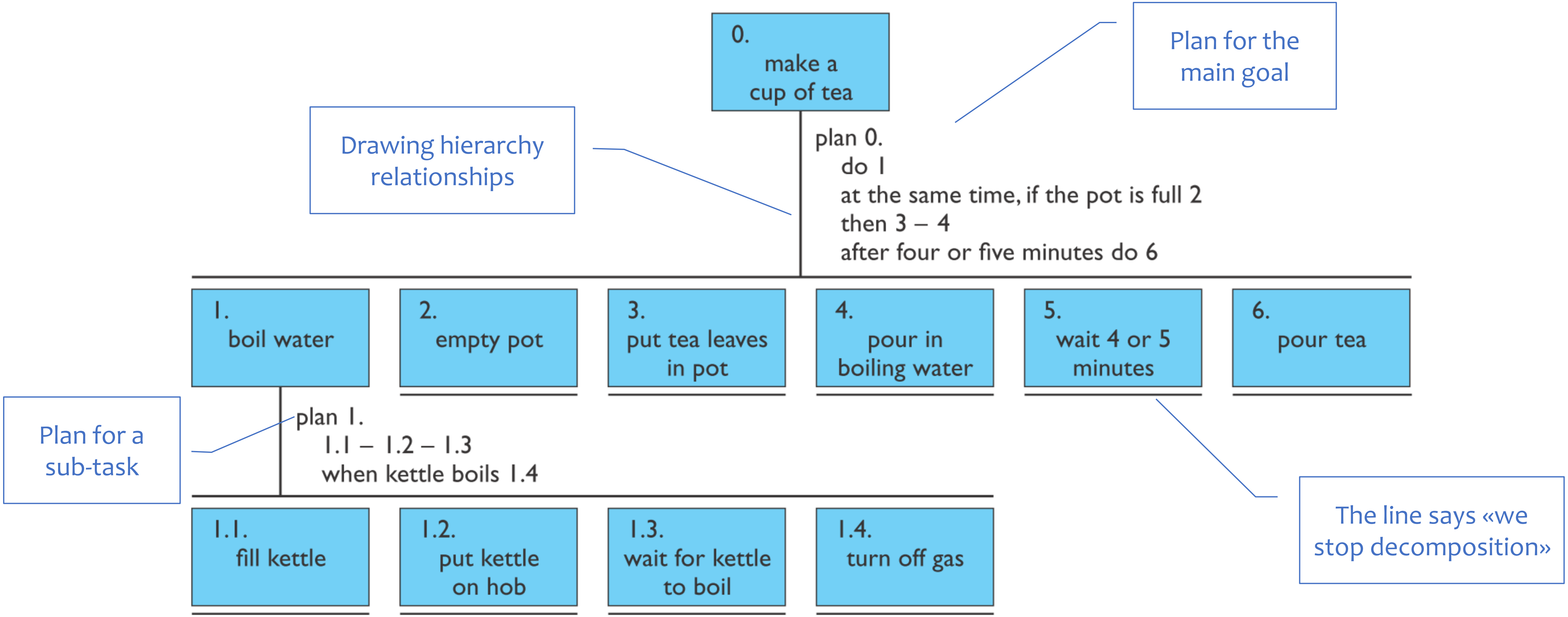
Expanding the hierarchy

- Each task is de-composed in sub-tasks, iteratively and recursively
  - Answer to the question: «what subtasks must be accomplished in order to perform the main task?»
  - The answer will come from direct observation, expert opinion, documentation, …

- Procedural task knowledge elicitation techniques:
  - Observation, re-enactment
  - Ask about procedures and triggers (pre-conditions)
  - “What happens if X goes wrong?”
  - Sorting steps into appropriate orders

- When is this process stopped?
  - Depends on the intended usage of the HTA (design vs documentation vs troubleshooting vs …)
  - Expand only relevant tasks
  - «Simple» tasks should be obvious to the users, and they should not contain hidden risks of failure
  - Motor actions are the lowest level (not always needed)

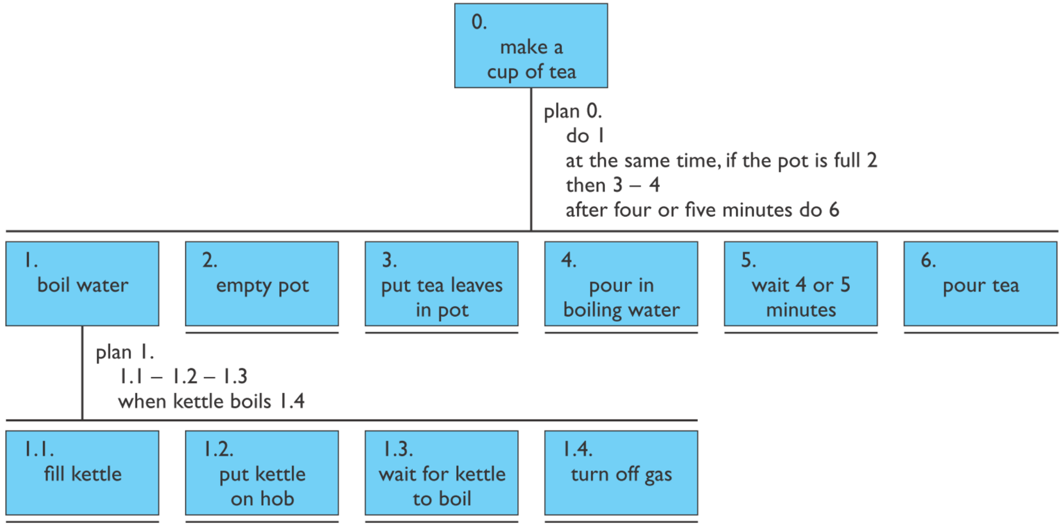
Example


0.
make a
cup of tea
plan 0.
do 1
at the same time, if the pot is full 2
then 3 – 4
after four or five minutes do 6
1.
boil water
2.
empty pot
3.
put tea leaves
in pot
4.
pour in
boiling water
5.
wait 4 or 5
minutes
6.
pour tea
plan 1.
1.1 – 1.2 – 1.3
when kettle boils 1.4
1.1.
fill kettle
1.2.
put kettle
on hob
1.3.
wait for kettle
to boil
1.4.
turn off gas
Drawing hierarchy relationships
Plan for the main goal
Plan for a sub-task
The line says «we stop decomposition»

Tasks as explanation

- Imagine asking the user the question:
  - **What are you doing now?**
- For the same action the answer may be:
  - typing ctrl-B
  - making a word bold
  - emphasising a word
  - editing a document
  - writing a letter
  - preparing a legal case

Refining the HTA



- Checking matched actions
  - Turn “off” without turning “on”?
- Restructuring
  - “Make pot” might be a meaningful task and group related actions
- Balancing complexity
  - Is “pour tea” simpler than “make pot”?
- Generalizing
  - If we want to make one *or more* cups?

References

- Alan Dix, Janet Finlay, Gregory Abowd, Russell Beale: Human Computer Interaction, 3rd Edition, Chapter 15 “Task Analysis”
- David Benyon: Designing Interactive Systems, Chapter 11 “Task Analysis”
- http://www.usabilitybok.org/task-analysis
- https://www.usability.gov/how-to-and-tools/methods/task-analysis.html

Acknowledgements

- Some icons from https://icons8.com

- Some material by
  - http://www.inf.ed.ac.uk/teaching/courses/hci/0708/lecs/tasks.pdf
  - https://www.tutorialspoint.com/human_computer_interface/design_process_and_task_analysis.htm
  - https://www.slideshare.net/alanjohndix/hci-3e-ch-15-task-analysis

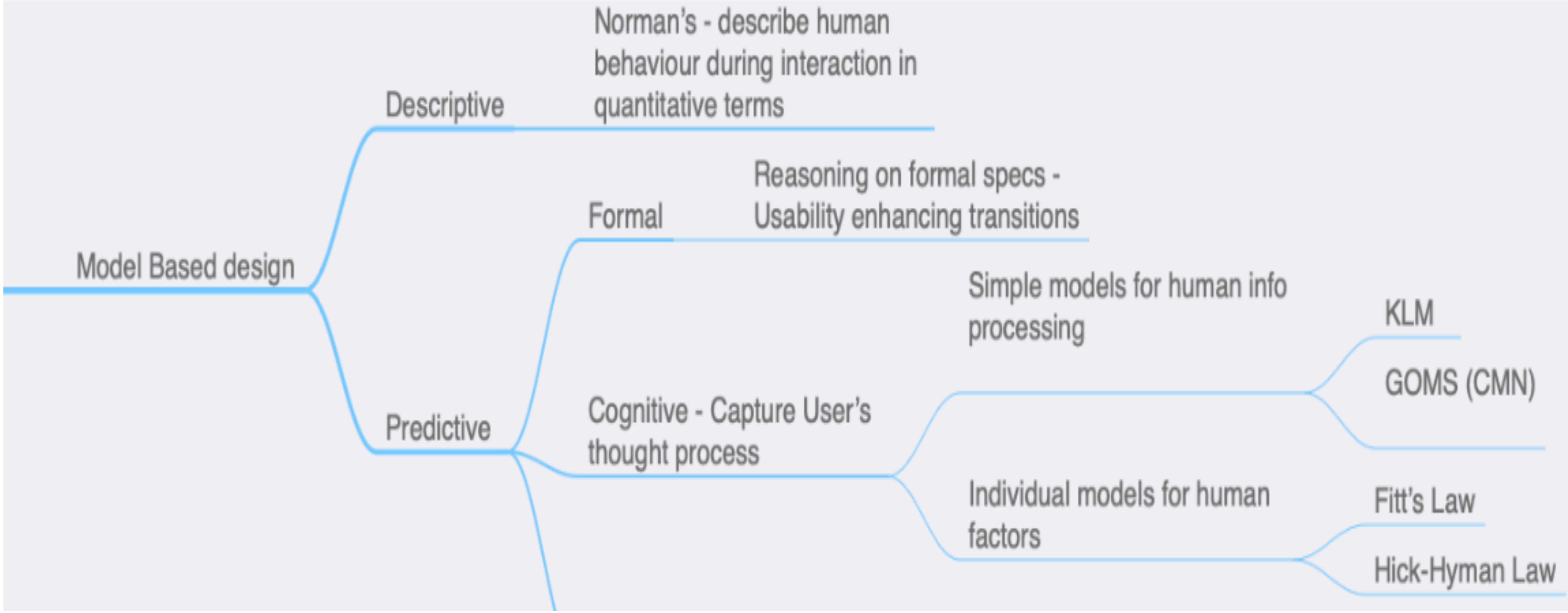
IE 403

Week 6-Lec 1

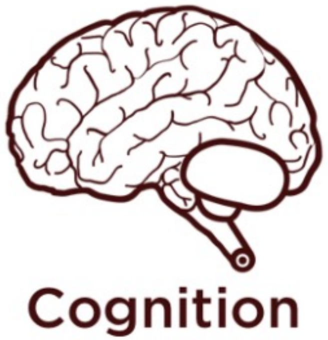
Human Information Processing(HIP) models

HIPs are used to understand and model how user interacts with an interface

Approaches to model cognition



Cognitive (User) models

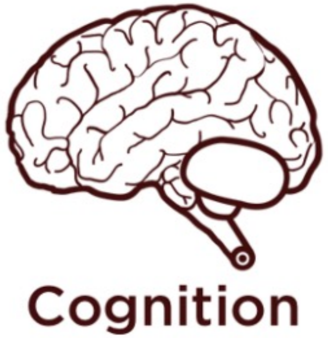


Greater number of steps ↑ Complex Cognitive activities

Higher complexity ↓ Usability

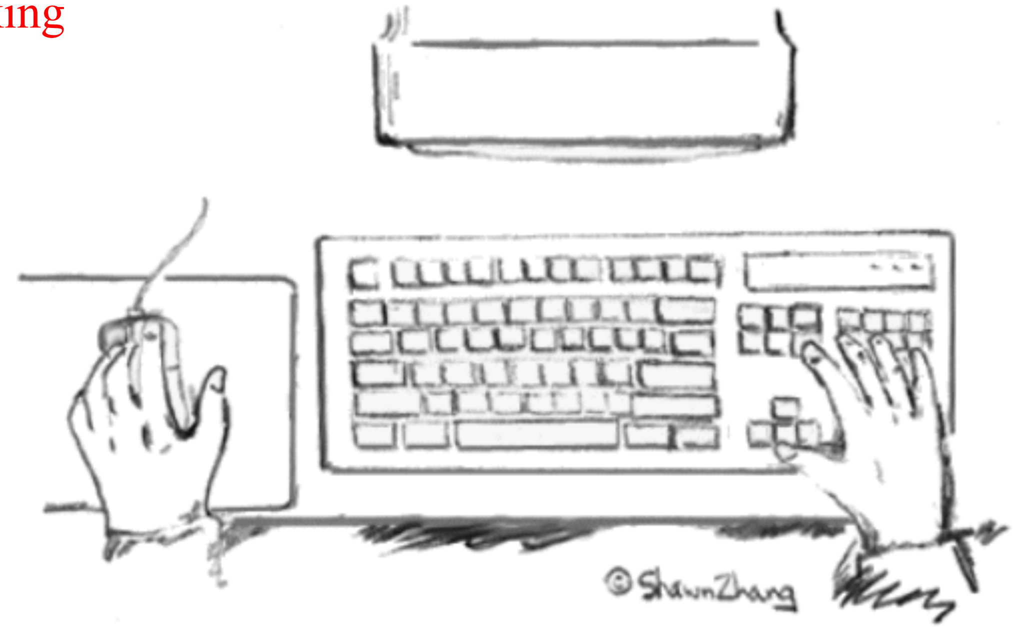
Predict: how many cognitive steps required?

What do cognitive models actually model?



Perceiving the interface

Thinking



Hand
movements

Keystroke Level Model (KLM)

- Predictive model
  - Time taken for avg user to execute through
    - Interface
    - Interaction method
      - Eg: close editor, save file, open document etc.,
- How it works?
  - Break down into simple steps
    - Operators
- How to use it?
  - Focus on task
    - Encode as sequence of steps
  - Calculate time
    - Using operators for each step
    - Sum all time values

The Psychology of Human-Computer Interaction, Card, Moran and Newell, 1983

Physical Motor Operators

Keystroke

Button press or release with mouse

Point with mouse

Draw line with mouse

Home hands between mouse and keyboard

Mental operator

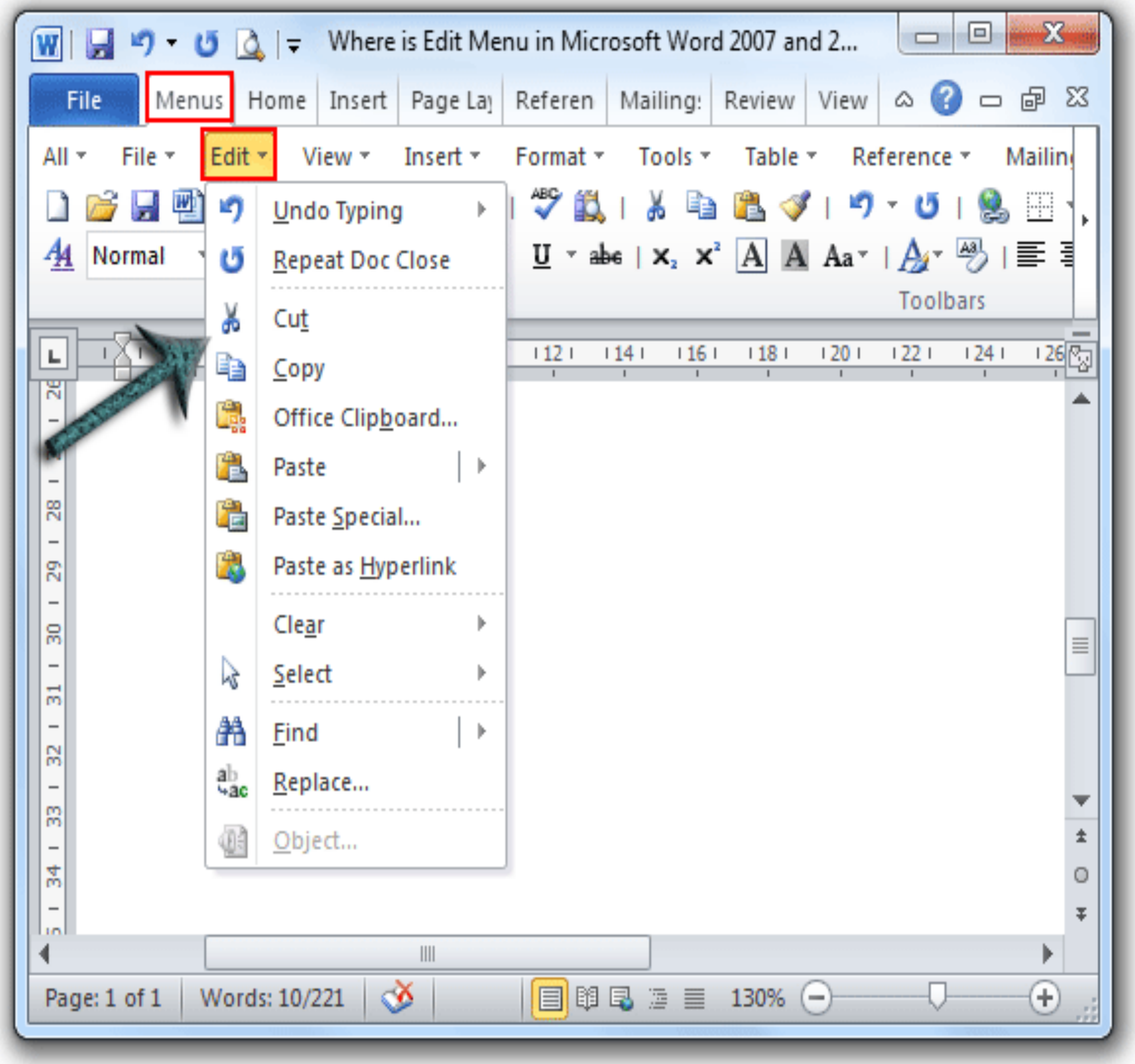
- Core thinking process (M)
- Decision making

Response operator

- Sys response time
  - Key press →output

How to use KLM

- Focus on task
  - Replace a word
  - Encode as sequence of steps
    - K, B, P, H, D (if any)
    - Mental operator time
    - $\sum$ (Operator times)
- Cognitive Task Analysis



Different styles of Keyboard typing

Avg Motor Operator times

Operator	Description	Time (second)
K (The key press operator)	Time to perform K for a good (expert) typist	0.12
	Time to perform K by a poor typist	0.28
	Time to perform K by a non-typist	1.20

Operator	Description	Time (second)
B (The mouse-button press/release operator)	Time to press or release a mouse-button	0.10
	Time to perform a mouse click (involving one press followed by one release)	2×0.10 = 0.20

Operator	Description	Time (second)
P (The pointing operator)	Time to perform a pointing task with mouse	1.10
H (the homing operator)	Time to move hand from/to keyboard to/from mouse	0.40

Mental Operator time

Operator	Description	Time (second)
M (The mental operator)	Time to mentally prepare for a physical action	1.35

Example

Suppose a user is writing some text using a text editor program. At some instant, the user notices a single character error (i.e., a wrong character is typed) in the text. In order to rectify it, the user moves the cursor to the location of the character (using mouse), deletes the character and retypes the correct character. Afterwards, the user returns to the original typing position (by repositioning the cursor using mouse).

Calculate the time taken to perform this task (error rectification) following a KLM analysis.

Building KLM for the Task

- Step 1: user brings cursor to the error location
  - To carry out step 1, user moves mouse to the location and ‘clicks’ to place the cursor there
- Operator level task sequence

Description	Operator
Move hand to mouse	H
Point mouse to the location of the erroneous character	P
Place cursor at the pointed location with mouse click	BB

Building KLM for the Task

- Step 2: user deletes the erroneous character
  - Switches to keyboard (from mouse) and presses a key (say “Del” key)
- Operator level task sequence

Description	Operator
Return to keyboard	H
Press the “Del” key	K

Building KLM for the Task

- Step 3: user types the correct character
  - Presses the corresponding character key
- Operator level task sequence

Description	Operator
Press the correct character key	K

Building KLM for the Task

- Step 4: user returns to previous typing place
  - Moves hand to mouse (from keyboard), brings the mouse pointer to the previous point of typing and places the cursor there with mouse click
- Operator level task sequence

Description	Operator
Move hand to mouse	H
Point mouse to the previous point of typing	P
Place cursor at the pointed location with mouse click	BB

What Is missing ?

- What about M (mental operator) – where to place them in the list?

IE 403/476

Human-Computer Interaction

Week 6-Lec 1

Example

Suppose a user is writing some text using a text editor program. At some instant, the user notices a single character error (i.e., a wrong character is typed) in the text. In order to rectify it, the user moves the cursor to the location of the character (using mouse), deletes the character and retypes the correct character. Afterwards, the user returns to the original typing position (by repositioning the cursor using mouse).

Calculate the time taken to perform this task (error rectification) following a KLM analysis.

- Total execution time (T) = the sum of all the operator times in the component activities

T = HPBBHKKHPBB = 6.20 seconds

Step	Activities	Operator Sequence	Execution Time (sec)
1	Point at the error	HPBB	0.40+1.10+0.20 = 1.70
2	Delete character	HK	0.40+1.20 = 1.60
3	Insert right character	K	1.20
4	Return to the previous typing point	HPBB	1.70

Something Missing!!

- What about M (mental operator) – where to place them in the list?
- It is usually difficult to identify the correct position of M
  - However, we can use some guidelines and heuristics

But what it is this M Operator?

- How was 1.35 arrived at?
  - Modeling variety of methods
  - $t_M = t_{execute} - (rest)$
  - t_M / N_{chunks}
  - *Number of Ms crucial in modeling*

$$\boxed{t_{\mathrm{EXECUTE}} = t_{\mathrm{K}} + t_{\mathrm{P}} + t_{\mathrm{H}} + \cancel{t_{\mathrm{D}}} + t_{\mathrm{M}} + t_{\mathrm{R}}}$$

M Placement Heuristics

- First, use one heuristic rule (Rule 0) to determine
  - Placement of candidate M operators
- Then, apply two heuristic rules (Rules 1–2) to determine
  - whether to delete each M

M placement Heuristics

General Rule:

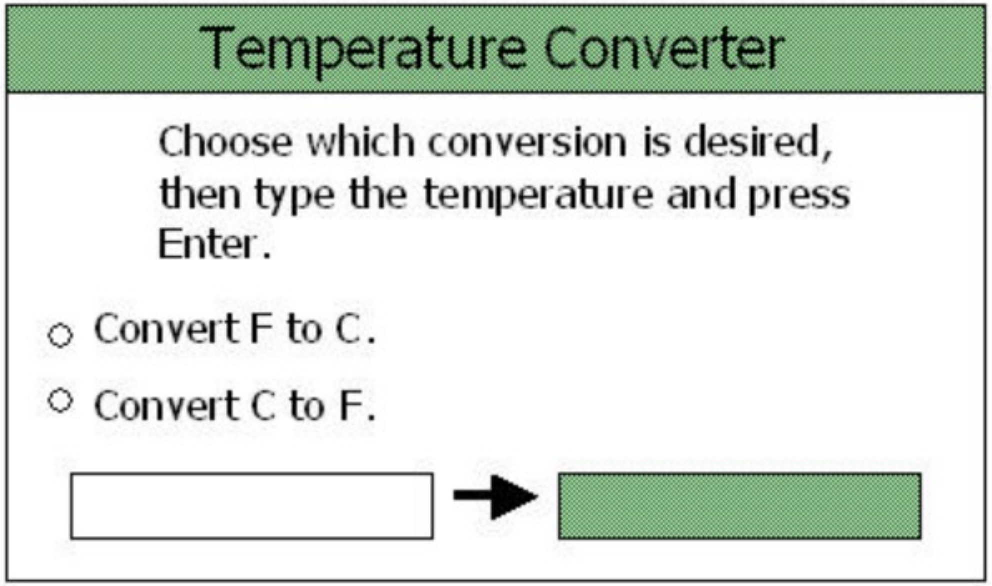
- Initial insertion of candidate Ms
  - Insert Ms in front of all keystrokes (K)
  - Insert Ms in front of all acts of pointing (P) that select commands
- Do not insert Ms before any P that points to an argument

– Mouse-operated widgets (like buttons, check boxes, radio buttons, and links) are considered commands

– Text entry is considered as argument

Rule 0

Rule 0. Insert **M**s in front of all **K**s that are not part of argument strings proper (e.g., text strings or numbers). Place **M**s in front of all **P**s that select commands (not arguments).



M Placement Heuristics

- Rule 1: deletion of anticipated Ms
  - If an operator following an M is *fully anticipated* in an operator immediately preceding that M, then delete the M
  - Example : if user clicks the mouse with the intention of typing at the location, then delete the M as a consequence of rule 0 – YOU already know that you are going to type

M Placement Heuristics

- Rule 2: deletion of Ms within cognitive units
  - If a string of MKs belongs to a *cognitive unit* then delete all Ms except the first
  - A cognitive unit refers to a chunk of cognitive activities which is predetermined
  - Example - if a user is typing “100”, MKMKMK becomes MKKK (since the user decided to type 100 it constitutes a single cognitive unit)

Another Example: deleting a word

Using Mouse to select word

- M
- P [start of word]
- BB [click]
- M
- P [end of word]
- K [shift]
- BB [click]
- H [to keyboard]
- M
- K [Del]
- Total: 3M + 2P + 4B + 1K = 6.93 sec

Pressing Del key *n* times

- M
- P [start of word]
- BB [click]
- H
- M
- K [Del]
- $\times\ n$ [where n = length of word]
- Total: 2M+P+2B+H+nK
- = 4.36 + 0.28n sec

KLM Limitations

- Although KLM provides an easy-to-understand-and-apply predictive tool for interactive system design, it has few significant constraints and limitations
  - It can model only "expert" user behavior
  - User errors can not be modeled

;

GOMS MODEL

GOMS,

- a hierarchical cognitive (thought) process is assumed

KLM

- Elementary (cognitive) steps or operators required to carry out a complex interaction task
  - The listing of operators implies a linear and sequential cognitive behaviour

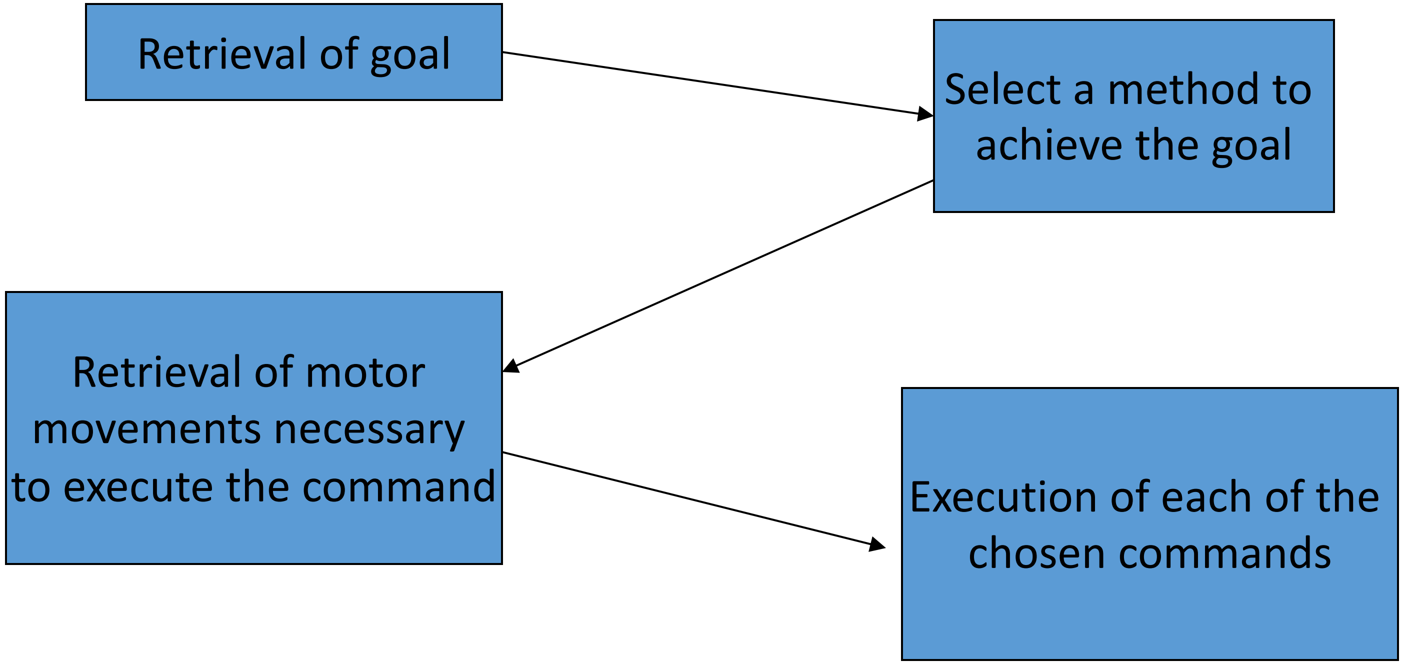
- Both assume error-free and 'logical' behavior
  - A logical behavior implies that we think logically, rather than driven by emotions

GOMS: Models user's behavior in terms of:

- Goals
  - ➢ What the user wants to do.
- Operators
  - ➢ Specific steps a user is able to take and assigned a specific execution time.
  - ➢ the notion of operators is not restricted to those seven
    - The modeler has the freedom to define any "elementary" cognitive operation and use that as operator
- Methods
  - ➢ Well-learned sequences of subgoals and operators that can accomplish a goal.
- Selection Rules
  - ➢ Guidelines for deciding between multiple methods.

Execution of a Mental Step

- GOMS allows explicit representation of mental steps of a task (the “Cognitive Processor”):



General Example

- Goal (the big picture)
  - Go from hotel to the airport
- Operators:
  - Walk, take bus, take taxi, rent car, take train
- Methods
  - Locate bus stop; wait for bus; get on the bus
- Selection rules (choosing among methods)?
  - Example: Walking is cheaper, but tiring and slow
  - Example: Taking a bus is complicated abroad

How to do GOMS analysis

- Generate task description
  - Pick high-level user **Goal**
  - Write **Methods for reaching Goal - may invoke subgoals**
  - Write **Methods for subgoals**
    - This is recursive
    - Stops when **Operators are reached**
- Evaluate description of task
- Apply results to UI
- Iterate!

GOMS: A Family of Models

- Keystroke-Level Model (KLM)
- **Card, Moran, and Newell (CMN-GOMS)**
- Natural GOMS Language (NGOMSL)
- Cognitive-Perceptual-Motor GOMS (CPM-GOMS)

General Example

- Goal: Edit text
- Operators
  - Use arrow keys
  - Use mouse
  - Use other keys
- Method: Delete text (sub-goal)
  - Positioning: 1) arrow key 2) mouse
  - Marking: 1) double click 2) use mouse
  - Delete (and add text): 1) start writing 2) press delete, then write new text
- Selection rules: if close, use arrow key etc.

•Goal: delete text (*n* chars long)
•Select:
  - method 1 if $n > 10$
  - method 2 if $n < 10$
•Method 1: Goal: highlight text & delete
  - Goal: highlight text
    - Point
    - Click
    - Point
    - Shift
    - Click
•Method 2: Goal: delete n char

Description	Duration (sec)
GOAL: MOVE-TEXT	
…….**GOAL: CUT-TEXT**	
……………**GOAL: HIGHLIGHT-TEXT**	
………………..MOVE-CURSOR-TO-BEGINNING	1.10
………………..CLICK-MOUSE-BUTTON	0.20
………………..MOVE-CURSOR-TO-END	1.10
………………..SHIFT-CLICK-MOUSE-BUTTON	0.48
………………..VERIFY-HIGHLIGHT	1.35
……….........**GOAL: ISSUE-CUT-COMMAND**	
…………………MOVE-CURSOR-TO-EDIT-MENU	
…………………PRESS-MOUSE-BUTTON	0.10
…………………MOVE-MOUSE-TO-CUT-ITEM	1.10
…………………VERIFY-HIGHLIGHT	1.35
…………………RELEASE-MOUSE-BUTTON	0.10
………**GOAL: PASTE-TEXT**	
……………**GOAL: POSITION-CURSOR-AT-INSERTION-POINT**	
…………………MOVE-CURSOR-TO-INSERTION-POINT	1.10
…………………CLICK-MOUSE-BUTTON	0.20
…………………VERIFY-POSITION	1.35
……………**GOAL: ISSUE-PASTE-COMMAND**	
…………………MOVE-CURSOR-TO-EDIT-MENU	
…………………PRESS-MOUSE-BUTTON	0.10
…………………MOVE-MOUSE-TO-PASTE-ITEM	
…………………VERIFY-HIGHLIGHT	1.35
…………………RELEASE-MOUSE-BUTTON	0.10
TOTAL PREDICTED TIME	**14.38**

M

Example

- Suppose we want to find out the definition of a word from an online dictionary. How can we model this task with (CMN)GOMS?

Example

- We shall list the goals (high level tasks) first
  - Goal: Access online dictionary (first, we need to access the dictionary)
  - Goal: Lookup definition (then, we have to find out the definition)

Example

- Next, we have to determine the methods (operator or goal-operator sequence) to achieve each of these goals
  - Goal: Access online dictionary
    - Operator: Type URL sequence
    - Operator: Press Enter

Example

- Next, we have to determine the methods (operator or goal-operator sequence) to achieve each of these goals
  - Goal: Lookup definition
    - Operator: Type word in entry field
    - Goal: Submit the word
      - Operator: Move cursor from field to Lookup button
      - Operator: Select Lookup
    - Operator: Read output

Example

- Thus, the complete model for the task is
  - Goal: Access online dictionary
    - Operator: Type URL sequence
    - Operator: Press Enter
  - Goal: Lookup definition
    - Operator: Type word in entry field
    - Goal: Submit the word
      - Operator: Move cursor from field to Lookup button
      - Operator: Select Lookup button
    - Operator: Read output

Example

- Notice the hierarchical nature of the model
- Note the use of operators
  - The operator "type URL sequence" is a high-level operator defined by the modeler
  - "Press Enter" is a keystroke level operator
  - Note how both the low-level and high-level operators co-exist in the same model

Example

- Note the use of methods
  - For the first goal, the method consisted of two operators
  - For the second goal, the method consisted of two operators and a sub-goal (which has a two-operators method for itself)

IE 403

Human-Computer Interaction

Week 7-Lec 2

Cognitive processing

- HCI designer must pay enough attention to human elements
- Must try to understand the human aspects of

computing

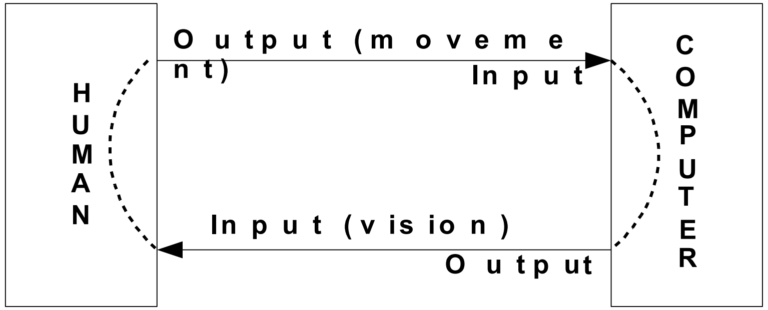
- Usability
- Reliability

Cognitive processing

Cognition refers to the processes by which we become acquired with things

- Cognition → gain knowledge
- Understanding, attending, being aware, acquiring skills and creating new ideas

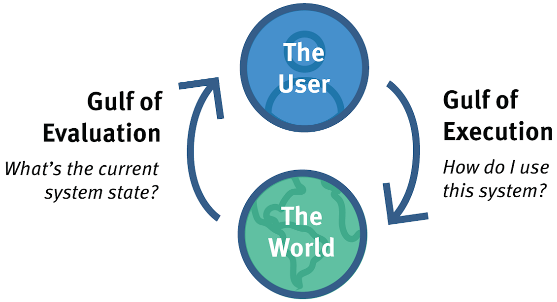
Why Cognition is an Issue?

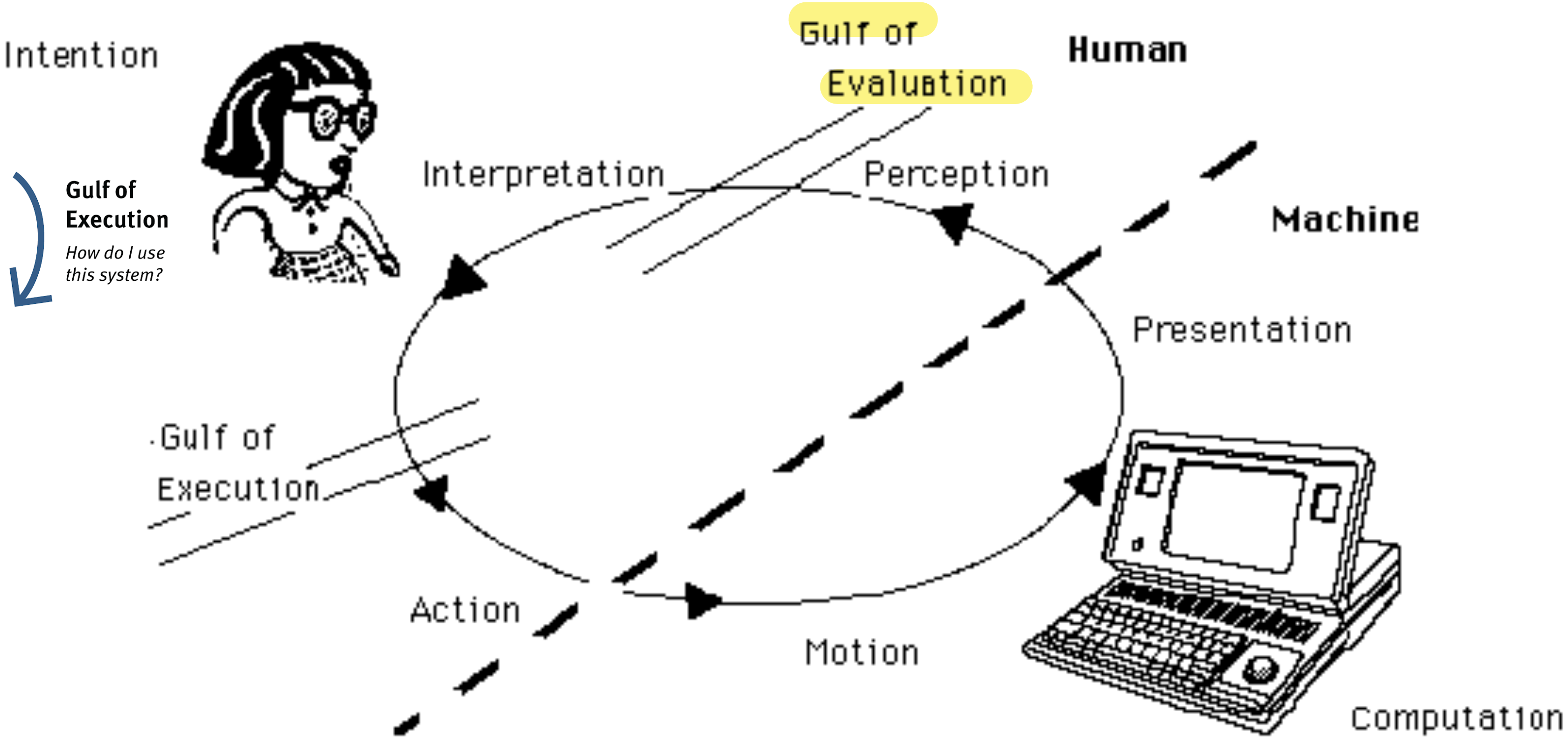


K now ledge transm issiopu
n betw een human and ter
com

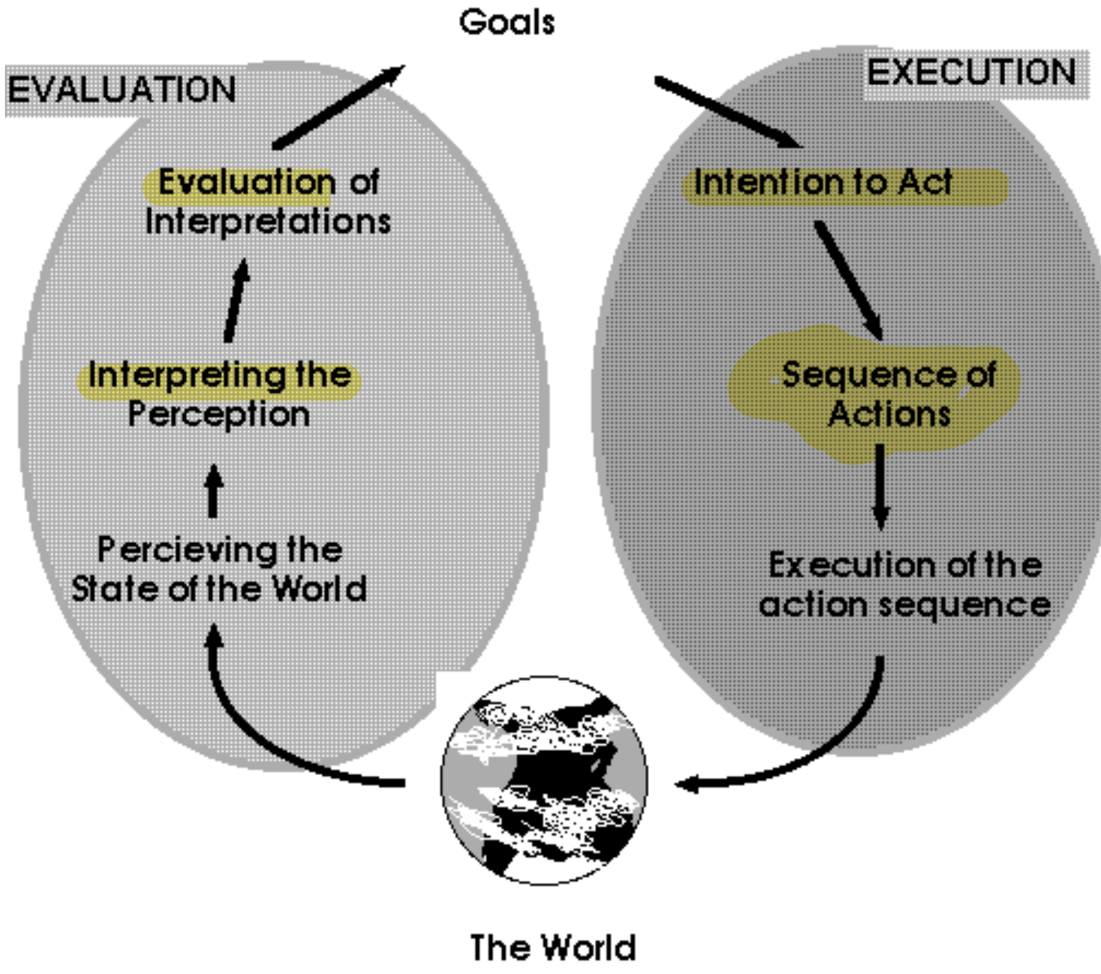
- In order to design good interfaces between humans and computers, the designer must have a basic understanding of
  - How human deals with information
  - How computer deals with similar information

Cognition in Man-Machine Interaction





Norman's Two Gulfs



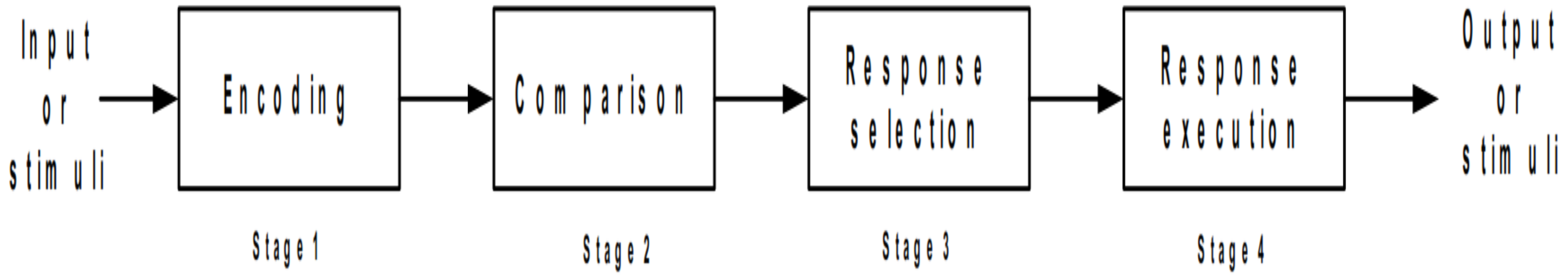
Human Information Processing

- The main paradigm in Cognitive Psychology is to characterize humans as information processor

- Everything that is sensed (sight, hearing, touch, smell and taste) is considered to be information which the mind processes

Human as a processor

Model of Human Information Processor



- Information enters and exits the human mind through a series of ordered processing stages

Human Information Processing

Lets try recalling the phone number of a friend

- Identify the words in this cognitive task and then retrieve their meaning
- Searching our memory for the solution of the problem
- On retrieving the number in the memory we generate a plan and formulate the answer into a representation
- Then we need to recite the digits or write down them down
-

Note: The major aspects in information processing analysis is to trace the mental operations and their for a outcome particular cognitive task

Human Information Processing

- Three aspects in human information processing
  - Perception
    - Visual
    - Audio
    - Other senses (tactile, Gustatory, Smell)
  - Cognition
    - Memory
    - Problem solving
    - Learning
  - Motor behavior
    - Speaking
    - Typing
    - Pointing
    - and others

Role of Cognition in HCI Design

- Quantitative measurement for HCI design
  - The model human processor provides a means of characterizing the various cognitive processes that are assumed to underlie the performance of a task
  - Card, Moran and Newell (1983) proposed a model for predicting the speed with which users could carry out tasks on a computer

Predicting the Cognition by Card et al.

- Essential parameters are

  t_p = time for perceiving a stimulus

  t_c = time for making a decision

  t_m = time for making a tapping motion

- Study reveals that

  t_p = 100 [50 ~ 200] ms

  t_c = 70 [20 ~ 170] ms

  t_m = 70 [30 ~ 100] ms

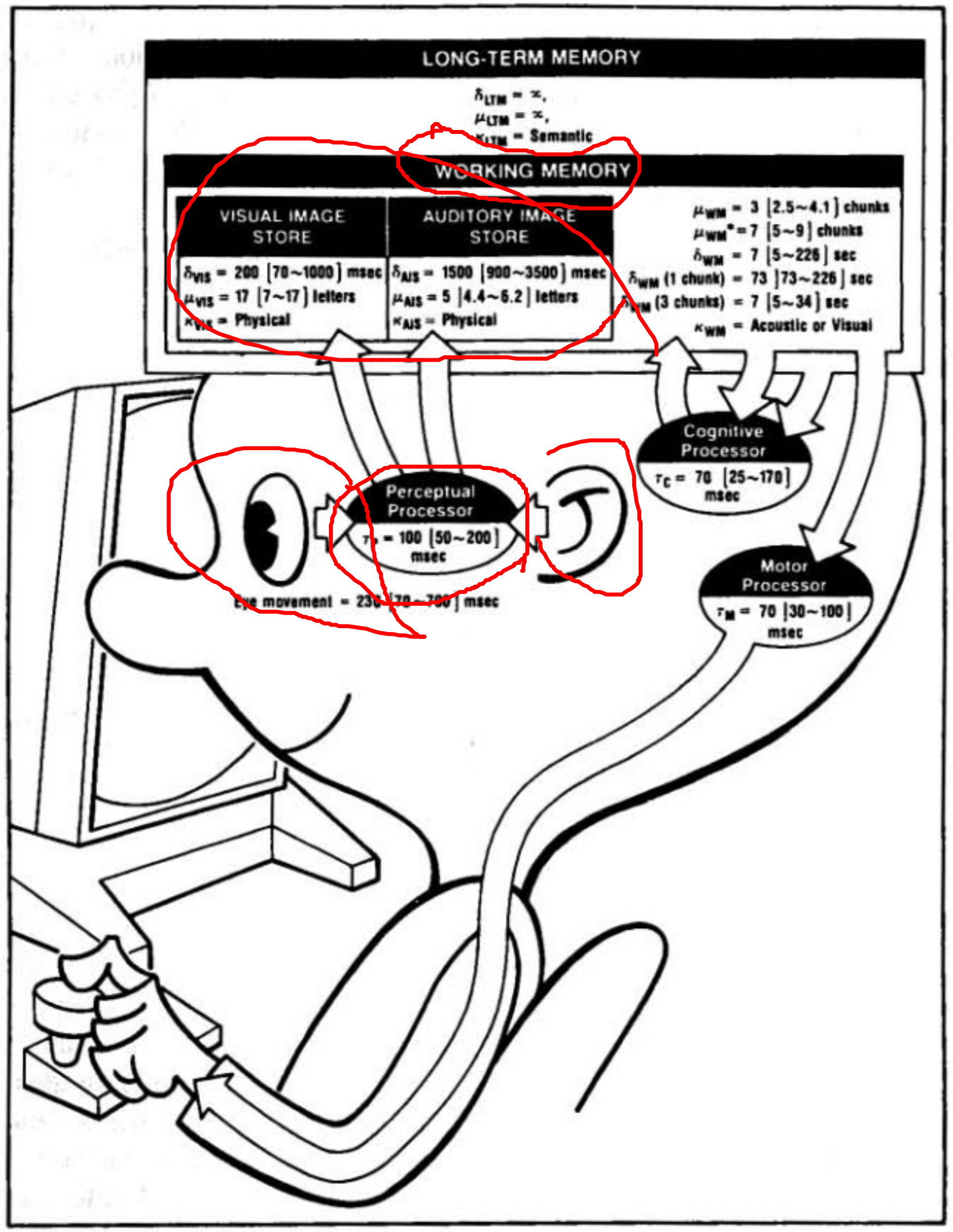
Predicting the Cognition by Card et al.

- Total time required for some user interface action is then predicted by the number of times each type of event must occur in the performance of that action

$$T = n_p t_p + n_c t_c + n_m t_m$$

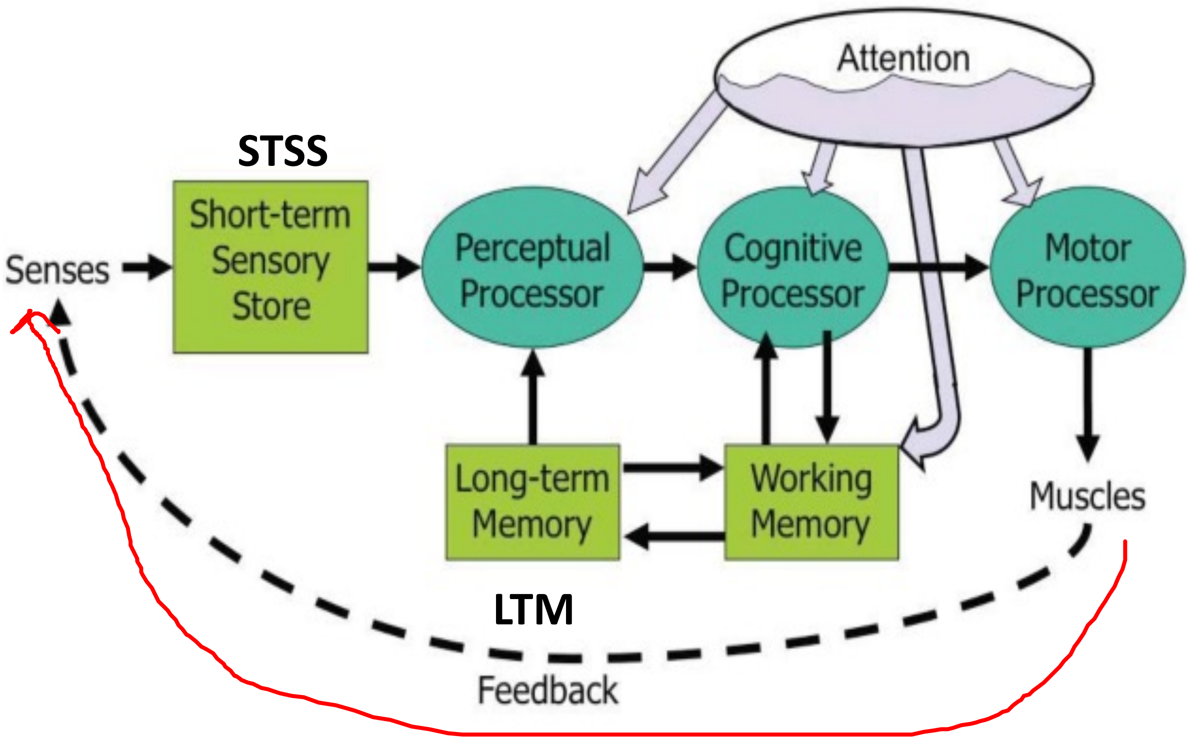
Human Information processing Model (HIP)

Sensory information flows into Working Memory through the Perceptual Processor. The Working Memory consists of activated chunks in Long-Term Memory.



The Psychology of Human-Computer Interaction, Card, Moran and Newell, 1983

Human Information processing Model



- High level
  - Numerical parameters of behavior
- STSS
- Perceptual Proc
  - LTM
- Cognitive Proc
  - WM
- Motor Proc
  - Feedback
- Attention

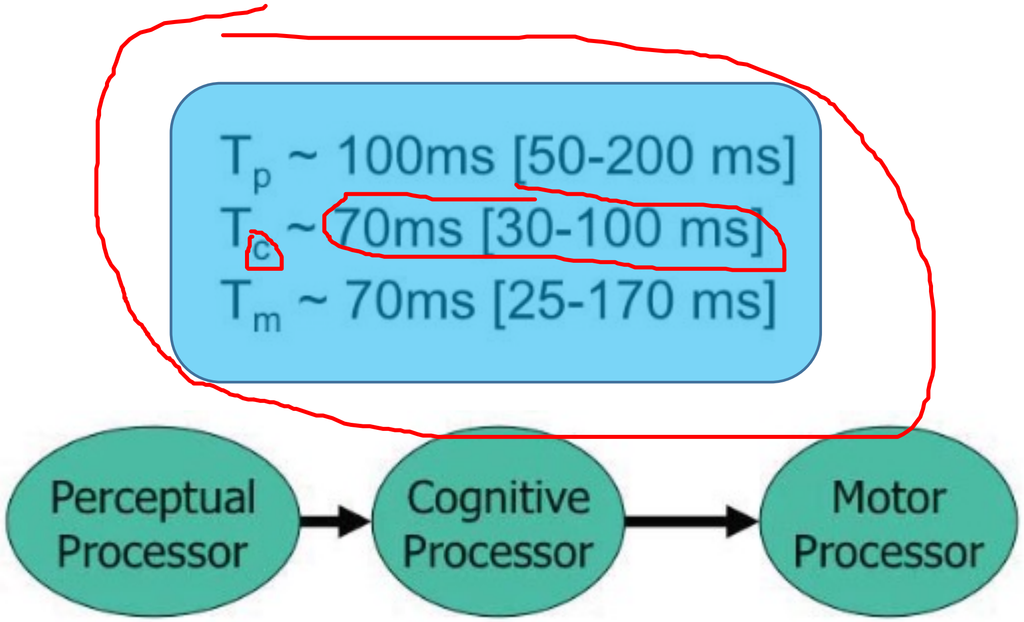
- Perceptual processor
  - senses, detects, and accepts inputs from the external world
  - stores parts of the perceptions in working memory (these are visual or auditory)
- Cognitive processor
  - interprets, manipulates, and makes decisions about the inputs (along with memory)
- Motor processor
  - generates physical actions (such as keying and mouse clicking) based on perceptions and cognitions

Capacity of working memory is roughly 7 ± 2 things
Long term memory is unlimited….but difficult to retrieve

Processors

- Cycle Time : Accept input and produce output
  - Ranges shown vary
    - Perceptual processor
      - Bright Vs dark environment
    - Cognitive processor
      - Driving, playing video games

        Vs
      - Idle or reading



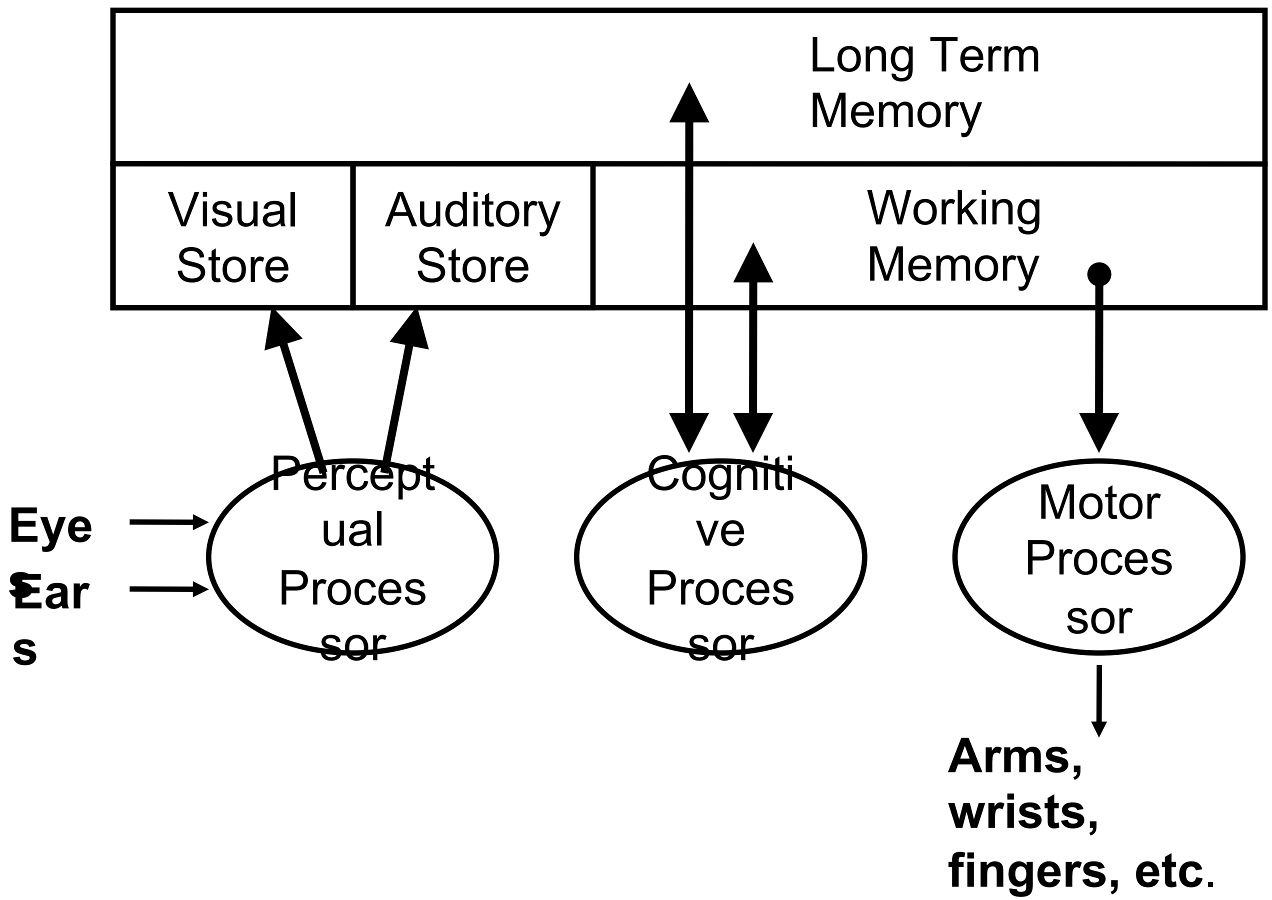
The Psychology of Human-Computer Interaction, Card, Moran and Newell, 1983

HIP used to understand and model how user interacts with an interface

Model Human Processor

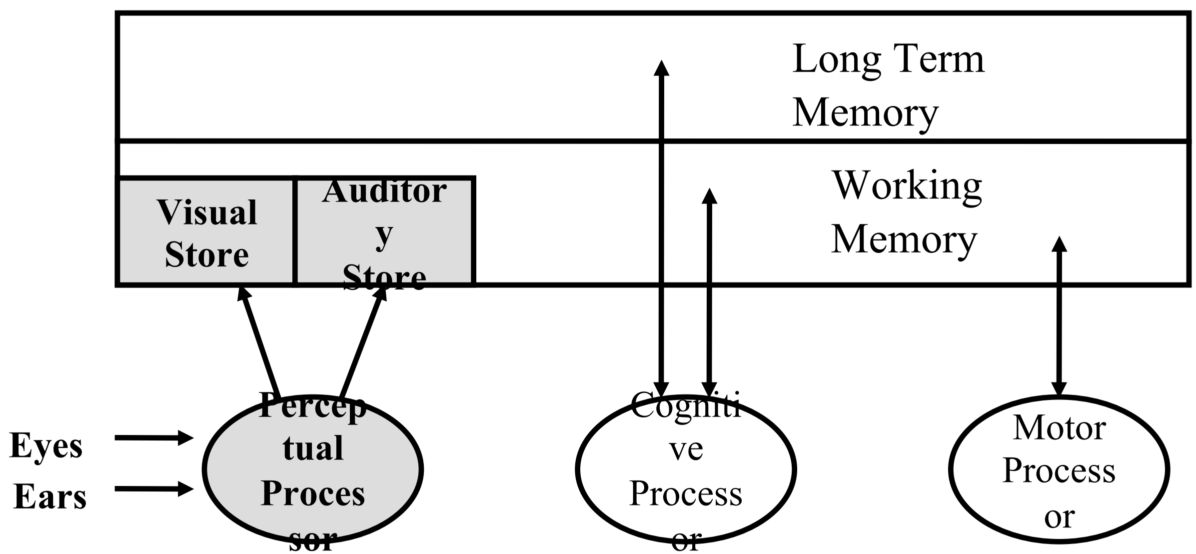
- Contains three interacting systems: perceptual, cognitive, and motor systems
  - For some tasks, systems operate in serial (pressing a key in response to a stimulus)
  - For other tasks, systems operate in parallel (driving, talking to passenger, listening to radio)
- Each system has its own memory and processor
  - Memory: storage capacity and decay time
  - Processor: cycle time (includes access time)
- Each system guided by principles of operation

Model Human Processor



Perceptual System

- Responsible for transforming external environment into a form that cognitive system can process
- Composed of *perceptual memory* and *processor*



Perceptual Memory

- Shortly after onset of stimulus, representation of stimulus appears in perceptual memory
  - Representation is physical (non-symbolic)
  - E.g., “7” is just the pattern, not the recognized digit
- As contents of perceptual memory are symbolically coded, they are passed to WM
  - Which processor does the coding?
- Codes information in perceptual memory for about 100ms and then retrieves next stimulus
- Decay time
  - 200ms for visual store
  - 1500ms for auditory store

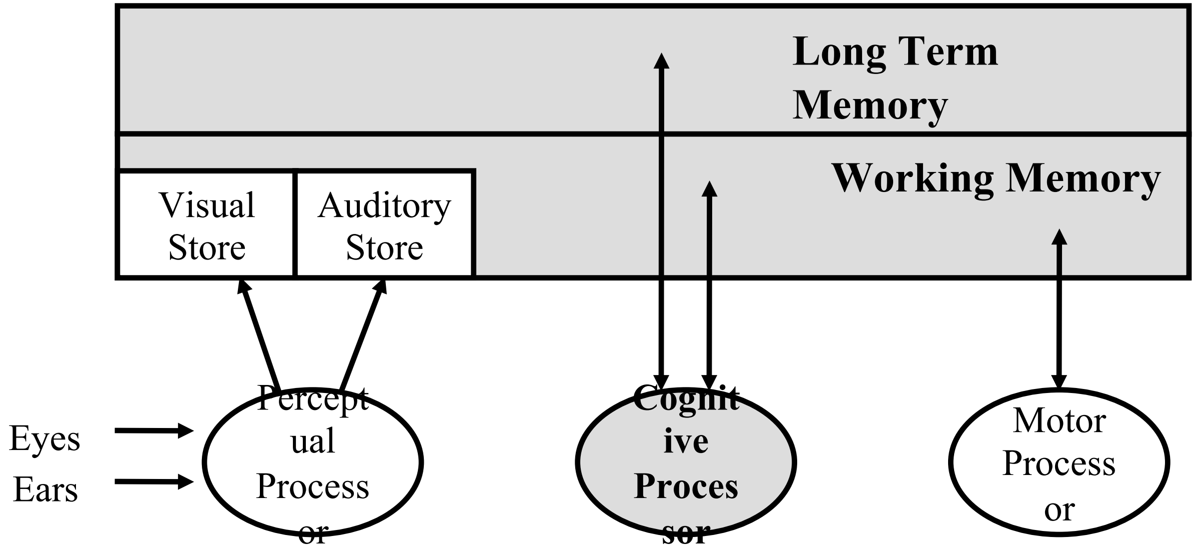
InClass Exercises

- **Assume perceptual cycle time = 100ms**
- If 20 clicks per second are played for 5 seconds, about how many clicks could a person hear?

- If 30 clicks per second are played for 5 seconds, about how many clicks could a person hear?

Cognitive System

- Uses contents of WM and LTM to make decisions and schedule actions with motor system
- Composed of a processor and two memories
  - WM and LTM



M W R C A A O L I B M F B I
B

MWR CAA OLI BMF
BIB

BMW RCA AOL IBM
FBI

Chunking

“Chunk”: unit of perception or memory

Chunking depends on presentation and what you already know

M W R C A A O L I B M F B I B

MWR CAA OLI BMF BIB

BMW RCA AOL IBM FBI

3-4 digit chunking is ideal for encoding unrelated digits

REfer to this URL for further reading

nngroup.com/articles/chunking/

Adapted from Robert Miller

How much of Cognitive processing?

- Manual Vs Calculator

https://www.youtube.com/watch?v=s6Omq
XCsYt8&t=258s

Working Memory

- Holds intermediate products of thinking and representations produced by perceptual system
- Comprised of activated sections of LTM called "chunks"
  - A chunk is a hierarchical symbol structure
  - 7 +/- 2 chunks active at any given time

Working Memory

- Holds intermediate products of thinking and representations produced by perceptual system
- Comprised of activated sections of LTM called “chunks”
  - A chunk is a hierarchical symbol structure
  - 7 +/- 2 chunks active at any given time

XOFVTMCBN

Working Memory

- Holds intermediate products of thinking and representations produced by perceptual system
- Comprised of activated sections of LTM called “chunks”
  - A chunk is a hierarchical symbol structure
  - 7 +/- 2 chunks active at any given time

Working Memory

- Decay caused by:
  - Time: about 7s for three chunks, but high variance
  - Interference: more difficult to recall an item if there are other similar items (activated chunks) in memory
- Discrimination Principle
  - Difficulty of retrieval determined by candidates that exist in memory relative to retrieval cues

Long-Term Memory

- Holds mass of knowledge; facts, procedures, history
- Consists of a network of related chunks where edge in the network is an association (semantic network)
- Fast read, slow write
- Infinite storage capacity, but you may forget because:
  - Cannot find effective retrieval cues
  - Similar associations to other chunks interfere with retrieval of the target chunk (discrimination principle)

Memory Example

- Suppose you are verbally given 10 arbitrary filenames to remember. In which order should you write down the filenames to maximize recall?

1. Quiz1.txt
2. Hw1.txt
3. In-semSols.pdf
4. In-semQP.pdf
5. Assignment2.txt
6. Quiz3.text
7. Week2-Lec2.ppt
8. UserCenteredDesign.ppt
9. NAAC.pdf
10. KLMAnalysis.ppt

- What if you are given 4 sets of filenames, where each set starts with the same characters?

1. Week1.ppt
2. Quiz1.pdf
3. Week2.ppt
4. Week3.ppt
5. HW1.txt
6. Quiz2.pdf
7. Week4.ppt
8. Quiz3.pdf
9. HW2.txt
10. Week4.ppt
11. Quiz4.pdf

Cognitive Processor

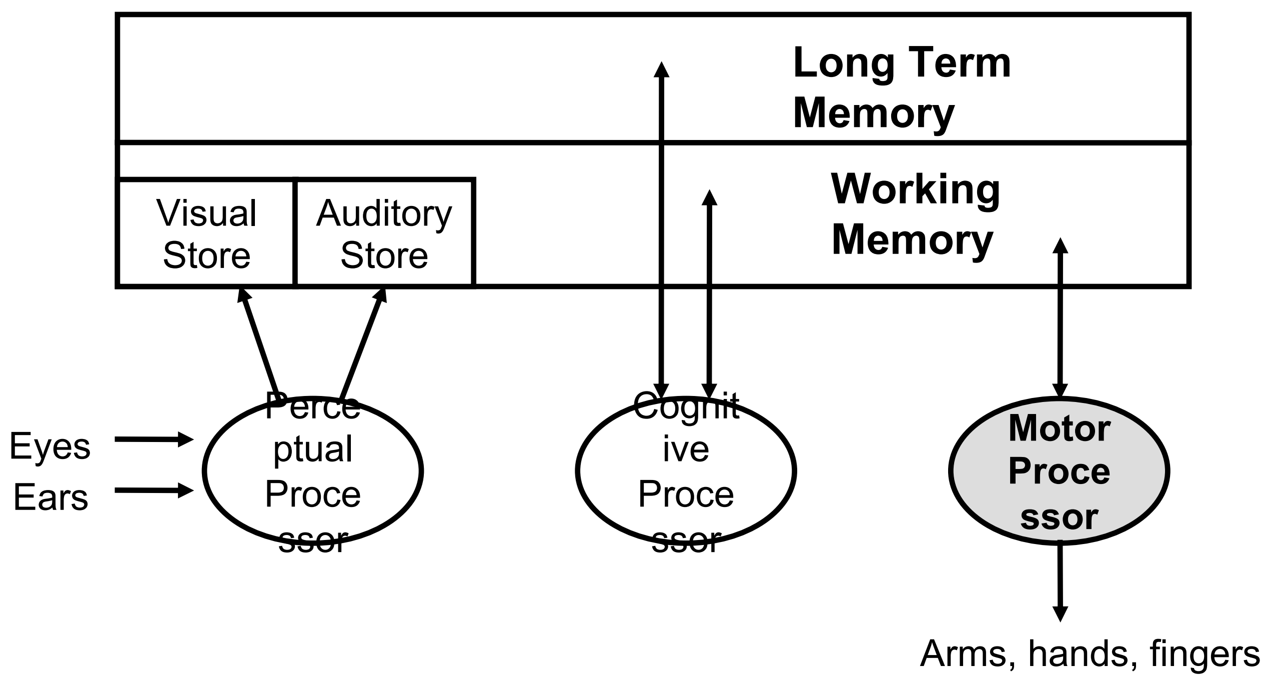
- Based on recognize-act cycle
  - Recognize: activate associatively-linked chunks in LTM
  - Act: modify contents of WM
  - Cycle time = ~70ms

Cognitive System Principles

- Uncertainty Principle
  - Decision time increases with the uncertainty about the judgment to be made, requires more cognitive cycles
- Variable Rate Principle:
  - Cycle time Tc is shorter when greater effort is induced by increased task demands or information loads; it also diminishes with practice.

Motor System

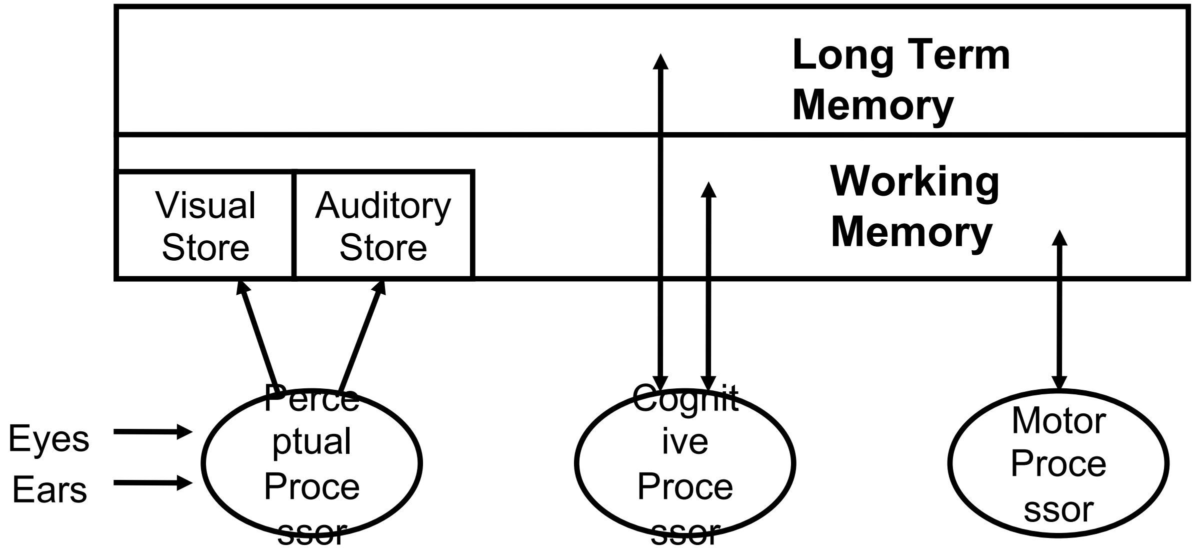
- Translates thoughts into actions
  - Head-neck and arm-hand-finger actions



Motor Processor

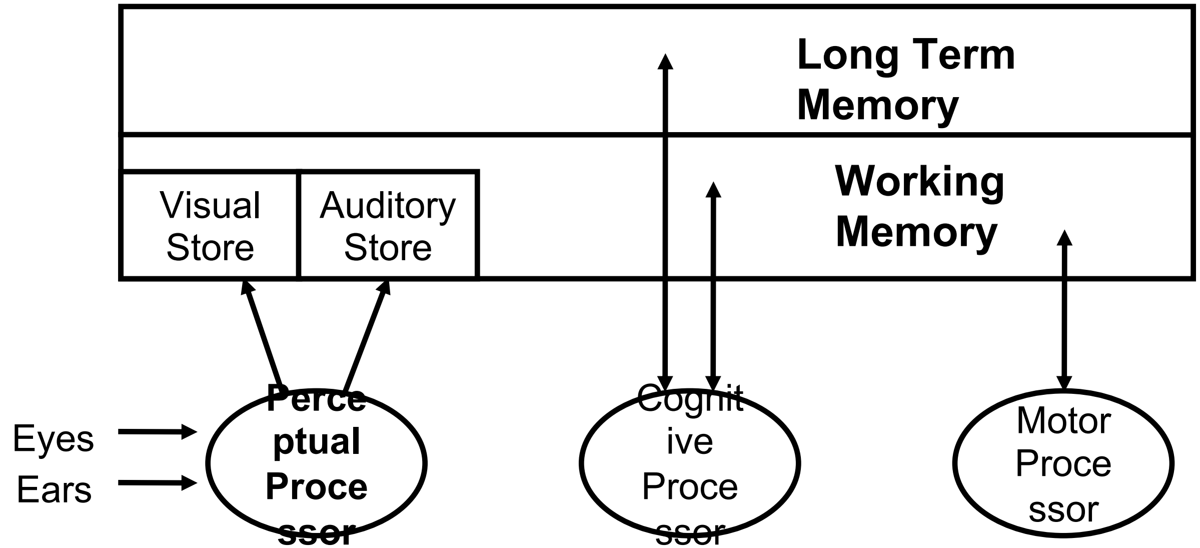
- Controls movements of body
  - Movement composed of discrete micro-movements
  - Micro-movement lasts about 70ms
  - Cycle time of motor processor about 70ms
- Caches common behavioral acts such as typing and speaking
  - No mention of this cache in the model

What We Know So Far



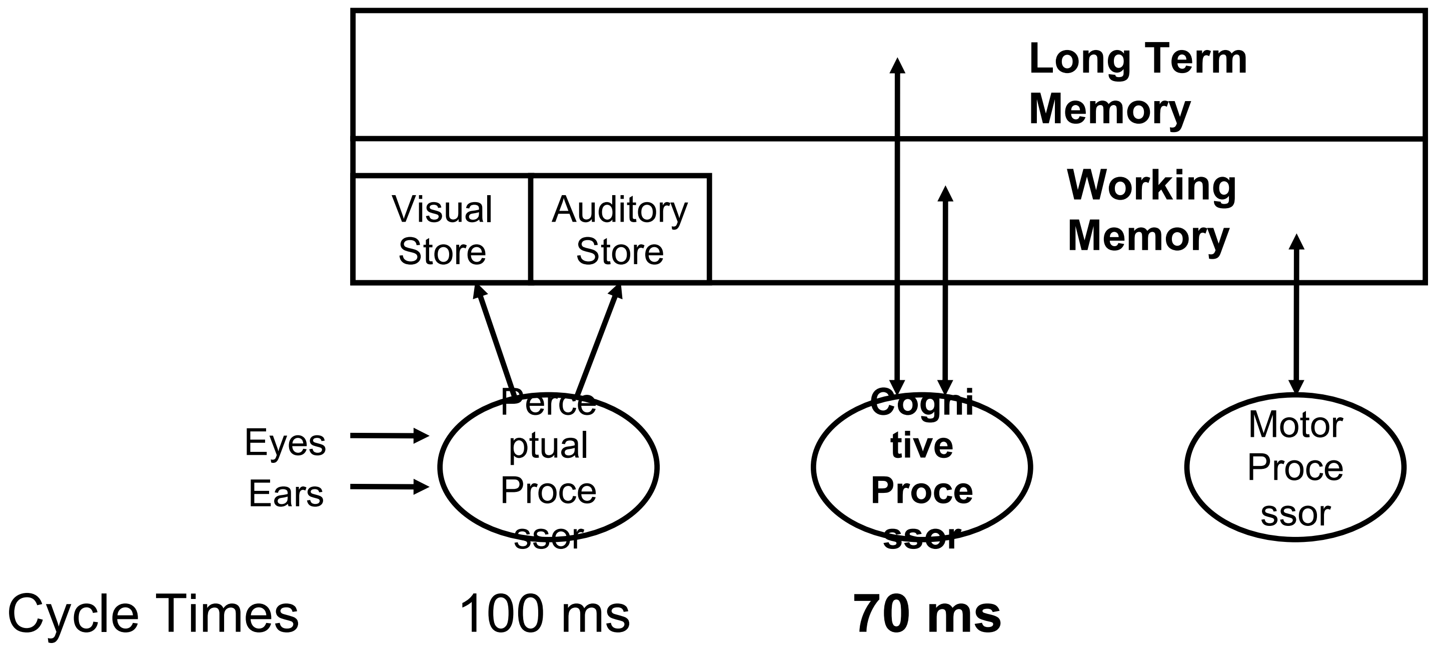
Cycle Times

What We Know So Far

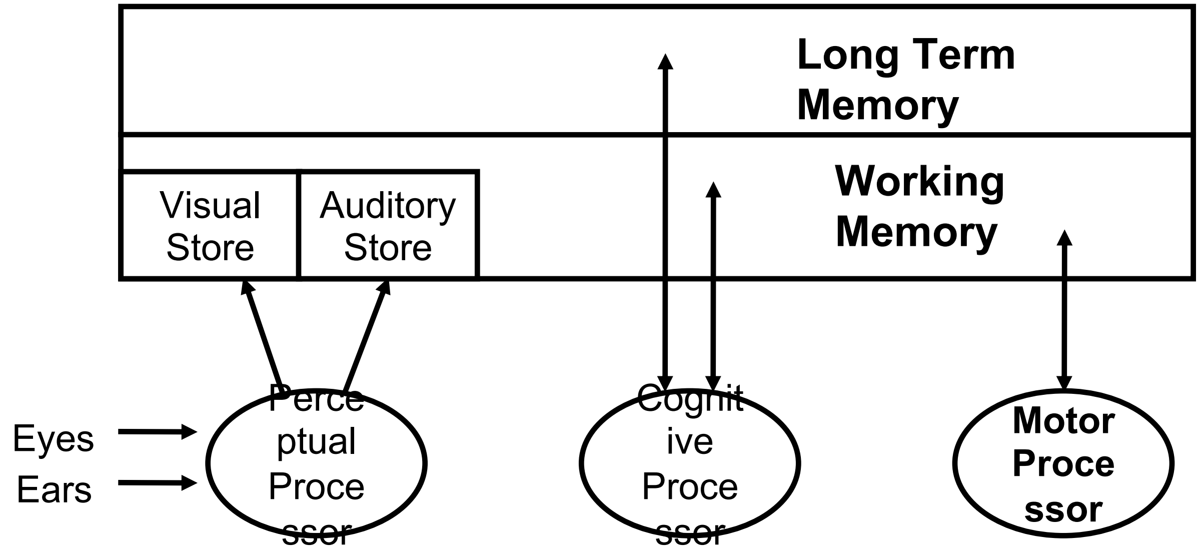


Cycle Times **100 ms**

What We Know So Far

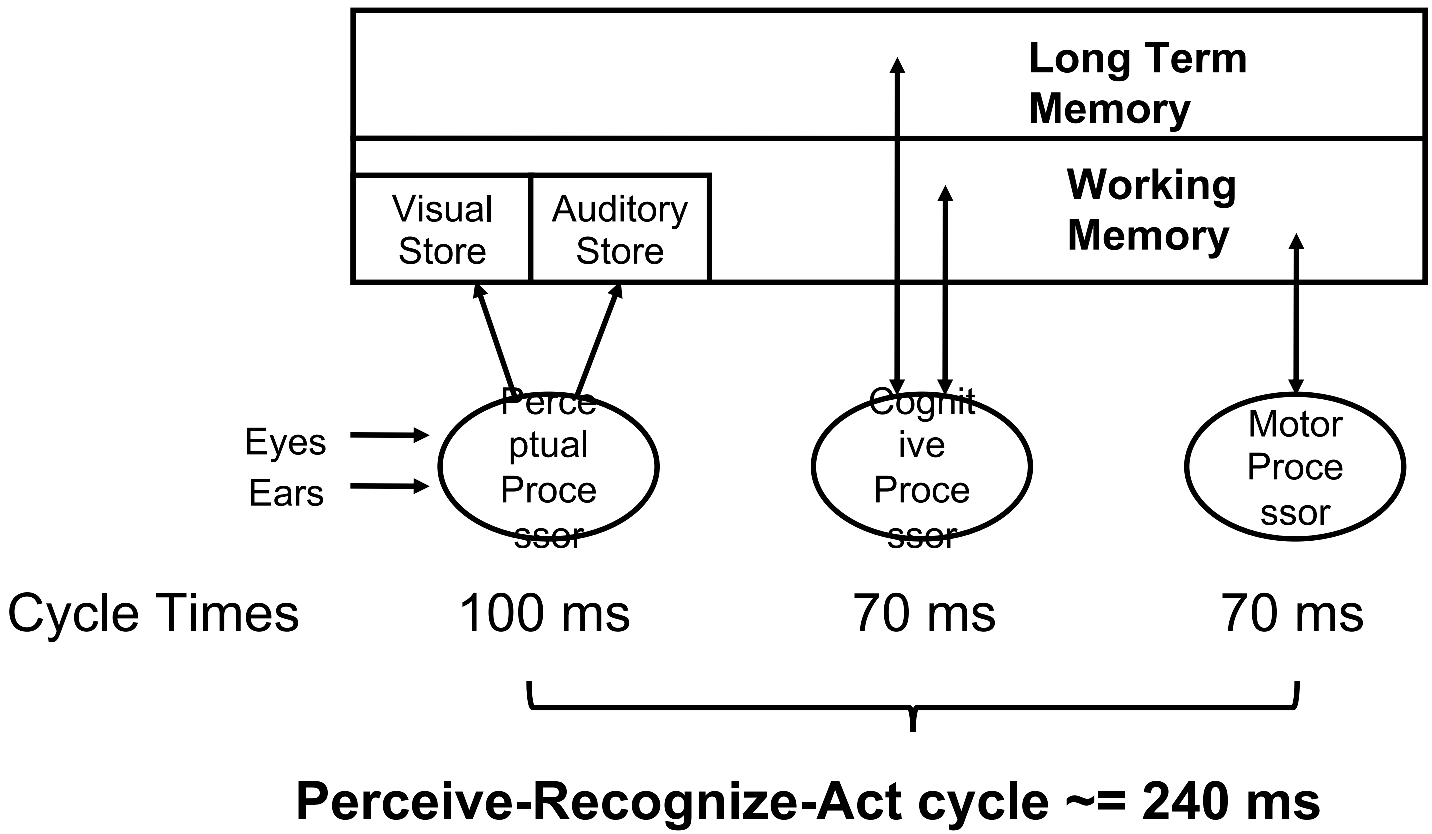


What We Know So Far



Cycle Times 100 ms 70 ms **70 ms**

Model Human Processor



Use Model to Compute Reaction Time for Simple Matching Task

- A user sits before a computer terminal. Whenever a symbol appears, s/he must press the space bar. What is the time between stimulus and response?

Use Model to Compute Reaction Time for Simple Matching Task

- A user sits before a computer terminal. Whenever a symbol appears, s/he must press the space bar. What is the time between stimulus and response?

$$T_p + T_c + T_m = 240 \text{ ms}$$

Use Model to Compute Reaction Time for a Symbol Matching Task

- Two symbols appear on the computer terminal. If the second symbol matches the first, the user presses “Y” and presses “N” otherwise. What is the time between the second signal and response?

Use Model to Compute Reaction Time for a Symbol Matching Task

- Two symbols appear on the computer terminal. If the second symbol matches the first, the user presses “Y” and presses “N” otherwise. What is the time between the second signal and response?

Tp + 2Tc (compare + decide) + Tm = 310 ms

Tc represents the time it takes for the user to compare the two symbols and make a decision. The reason why it is multiplied by 2 is that there are two separate processes involved: comparing the symbols and making a decision. Each of these processes takes time Tc, so the total time for both processes is 2Tc.

In General Case

- Need a bridge from task structure to MHP
  - Enables top down as opposed to bottom up analysis
- Analyze goal structure of the task, then for each step:
  - Analyze user actions required (motor system)
  - Analyze user perception of the output (perceptual system)
  - Analyze mental steps to move from perception to action (cognitive system)
- Sum the processing times from each step to get a reasonably accurate prediction of task performance

GOMS – Advantages

- Enables quantitative comparison of task performance before implementation
  - Empirical data shows model provides a good approximation of actual performance
- Could be embedded in sketch simulation tool
  - Designer provides GOMS model and interface sketch, tool returns performance prediction

GOMS – Disadvantages

- Predicting movement time based on the level of micro-movements not plausible
  - Need a higher-level method for predicting movement time
- Fitt's Law

Take Home Exercises

- How many frames per second must a video be played to give illusion of motion?

- In a talking head video, how far off can the audio and video be before a person perceives the video as unsynchronized?

IE 403

Human-Computer Interaction

Week 7-Lec 1

Recap

- Two popular models belonging to the GOMS family, namely KLM and (CMN)GOMS
  - Those models, as we mentioned before, are simple models of human information processing

Objective

- A second type of cognitive models used in HCI is the individual models of human factors
- To recap, these are models of human factors such as *motor movement, choice-reaction, eye movement* etc.
  - The models provide analytical expressions to compute values associated with the corresponding factors, such as movement time, movement effort etc.

Objective

– **The Fitts' law**: a law governing the manual (motor) movement

– **The Hick-Hyman law**: a law governing the decision making process in the presence of choice

Fitts' Law

- It is one of the earliest predictive models used in HCI (and among the most well-known models in HCI also)
- First proposed by PM Fitts (hence the name) in 1954

Fitts, P. M. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47, 381-391.

Fitts' Law

- Fitts' law is a model of human motor performance
  - It mainly models the way we move our hand and fingers
- Fitts aimed to find the bandwidth of human movement — how many repetitive movements could be performed in a given time interval

Fitts' Law - Characteristics

- The law models human motor performance having the following characteristics
  - The movement is related to some *"target acquisition task"* (i.e., the human wants to acquire some target at some distance from the current hand/finger position)

Fitts' Law - Characteristics

- The law models human motor performance having the following characteristics
  - The movement is *rapid* and *aimed* (i.e., no decision making is involved during movement)
  - The movement is *error-free* (i.e. the target is acquired at the very first attempt)

What does Fitts's law give?

the relationship between the time it takes a **pointer** (such as a mouse cursor, a human finger, or a hand) to move to a particular **target** (e.g., physical or digital button, a physical object) in order to interact with it in some way (e.g., by clicking or tapping it, grasping it, etc.):

$$T = a + b \log_2 \frac{2D}{w}$$

D is distance to TARGET
w is width of TARGET

https://www.nngroup.com/articles/fitts-law/#:~:text=Fitts's%20law%20says%20that%20the,user's%20most%20probable%20prior%20location

Measuring Task Difficulty

$ID = \log_2(D/W+1)$ [unit is *bits*]

Logarithm : as distance increases, does not mean time increases linearly. It increases, slowly.
As target is longer, movement fast but slows down as we approach the target

1.The larger the distance to the target, the longer it will take for the pointer to move to it. In other words, closer targets are faster to acquire.
2.The bigger the target, the shorter the movement time to it. In other words, bigger targets are better.

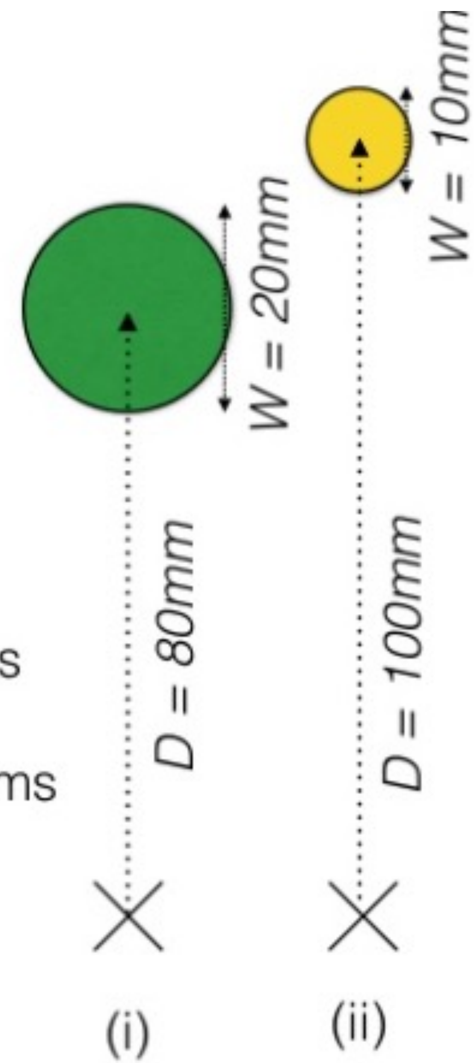
Putting it all together

$$MT = a + b\log_2\left(\frac{D}{W} + 1\right)$$

Assume a = 50ms, and b = 150ms

$MT_i = 50 + 150\log_2(80/20 + 1) = 398$ ms

$MT_{ii} = 50 + 150\log_2(100/10 + 1) = 569$ ms



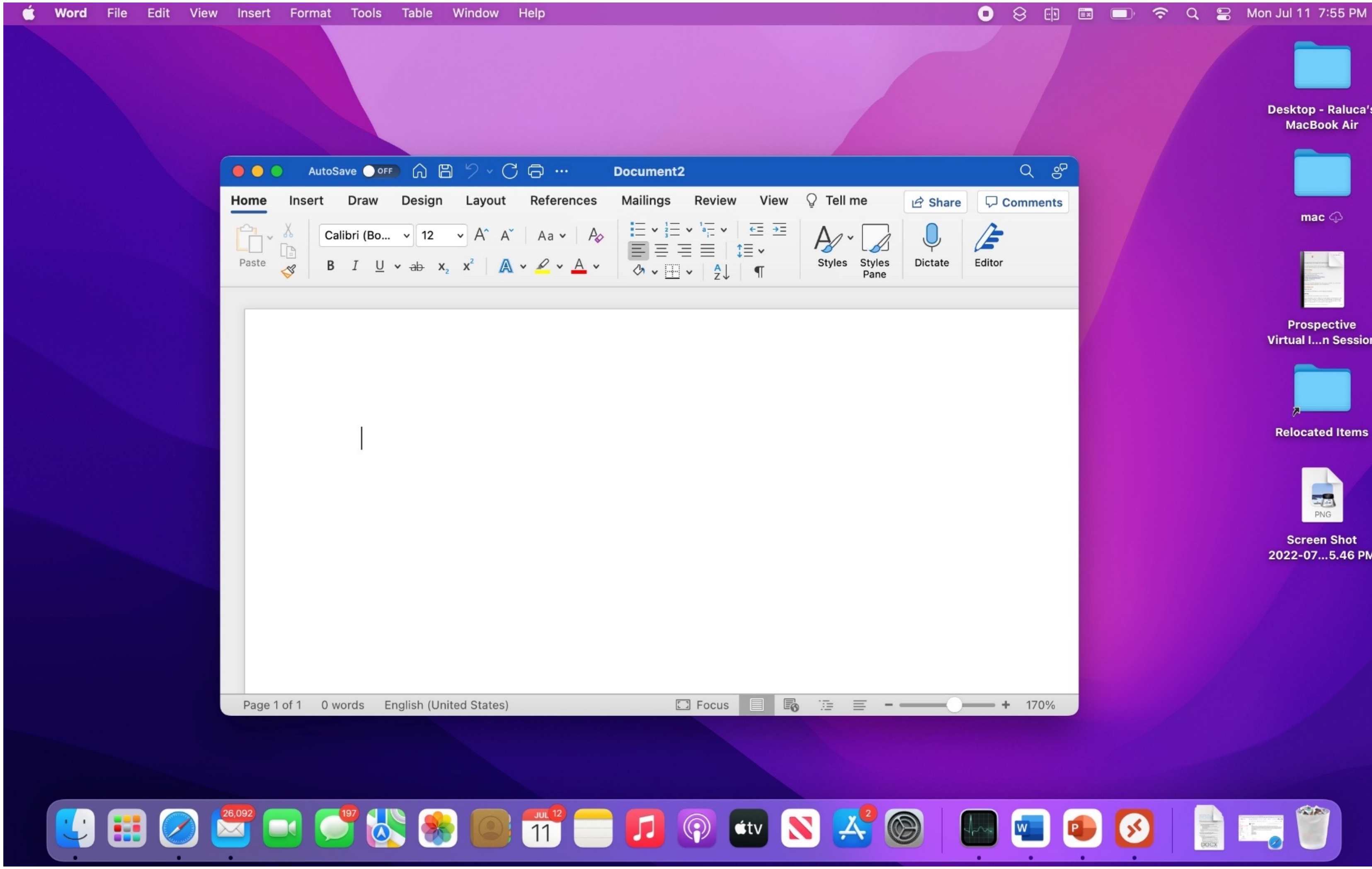
Uses of Fitts' law in UI design

Fitts law uses

- Edge of the screen space
  - controls at the edges of the screen,
  they should be active all the way to the edge to take advantage of this effect

- Pop menus relative to position
  - Nearest pixel pop up opens

- Menu item size
- Icon size
- Scroll bar target size and placement
  - Up / down scroll arrows together or at top and bottom of scroll bar


Word File Edit View Insert Format Tools Table Window Help
Mon Jul 11 7:55 PM
AutoSave OFF
Document2
Home Insert Draw Design Layout References Mailings Review View Tell me
Share
Comments
Paste
Calibri (Bo...
12
Styles
Styles Pane
Dictate
Editor
Page 1 of 1
0 words
English (United States)
Focus
170%
Desktop - Raluca's MacBook Air
mac
Prospective Virtual I...n Session
Relocated Items
Screen Shot 2022-07...5.46 PM

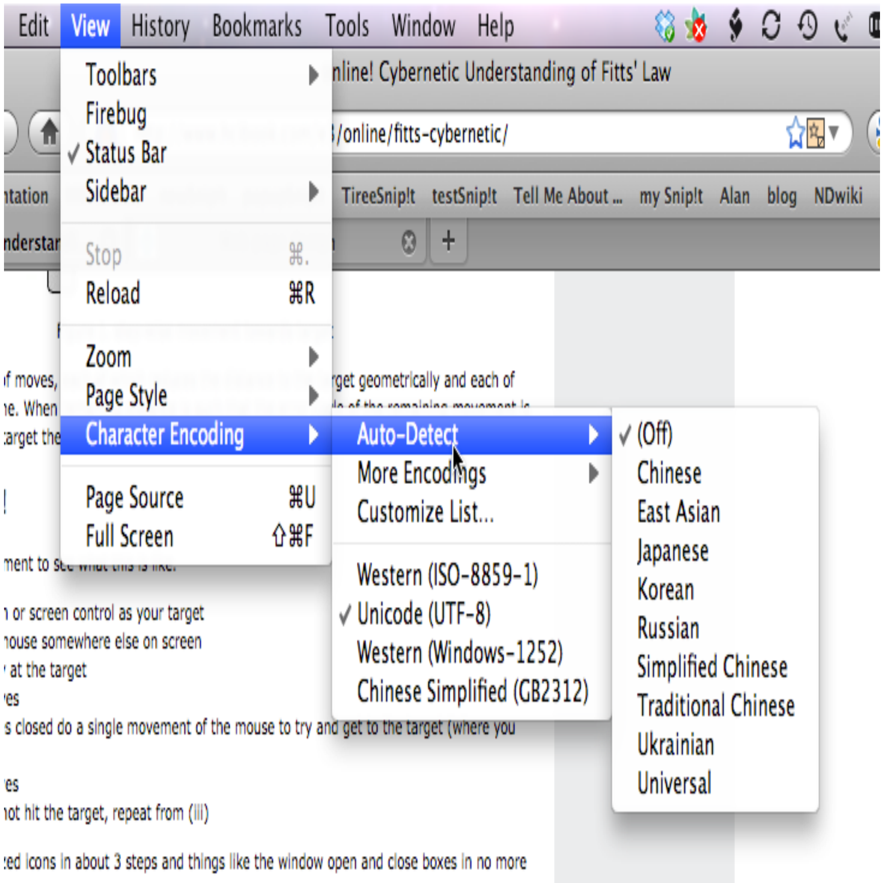
Microsoft Toolbars offer the user the option of displaying a label below each tool. Name at least one reason why labeled tools can be accessed faster.

According to Fitts' Law, the time required to rapidly move to a target area is a function of the ratio between the distance to the target and the width of the target. When Microsoft toolbars offer the user the option of displaying a label below each tool, the label becomes part of the target. This means that the target is now bigger, and according to Fitts' Law, bigger targets can always be accessed faster, all else being equal. This is one reason why labeled tools can be accessed faster in terms of Fitts' Law.

- The label becomes part of the target. The target is therefore bigger. Bigger targets, all else being equal, can always be acccessed faster.

- When labels are not used, the tool icons crowd together.

What is the bottleneck in hierarchical menus and what techniques could make that bottleneck less of a problem?



The bottleneck is the passage between the first-level menu and the second-level menu

- Fitts' law is not just about target size and distance; it's also about the number of targets.
- The more targets, all else being equal, the longer the task will take.
- Hierarchicals automatically add one extra target.
- Making it difficult to enter a second-level menu adds an additional target, the second-level menu itself.

Name at least one advantage circular popup menus have over standard, linear popup menus.

- With the options displayed around you in a circle, you need to only move a pixel or two to enter the "slice of pie" you want. Less travel, good target size. Good design.
- Feeding directional information into your motor memory. As long as the options are few enough, Learnability and Memorability EASY

Nature of the Fitts' Law

- Another important thing about the Fitts' law is that, it is both a descriptive and a predictive model
- Why it is a descriptive model?
  - Because it provides "throughput", which is a descriptive measure of human motor performance

Nature of the Fitts' Law

- Another important thing about the Fitts' law is that, it is both a descriptive and a predictive model
- Why it is a predictive model?
  - Because it provides a prediction equation (an analytical expression) for the time to acquire a target, given the distance and size of the target

Fitts's law says that the time to move to a target depends on how big it is and on how far away it is. As you are creating new UI designs, think about optimizing both these variables by creating well-spaced, big targets and positioning them so that they are close to the user's most probable prior location.

IE 403

Human-Computer Interaction

Week 9-Lec 1

Throughput

Fitts' also proposed a measure called the *index of performance* (IP), now called *throughput* (TP)

- Computed as the difficulty of a task (ID, in bits) divided by the movement time to complete the task (MT, in seconds)

  - Thus, TP = ID/MT bits/s

Exact Equation

- Run empirical tests to determine k_1 and k_2 in $MT = k_1 + k_2 * ID$
- Will get different ones for different input devices and device uses

Implication of Throughput

- The concept of throughput is very important
- It actually refers to a measure of performance for rapid, aimed, error-free target acquisition task (as implied by its original name “index of performance”)
  - Taking the human motor behavior into account

Examples of Test Condition

- Suppose a user is trying to point to an icon on the screen using a mouse
  - The task can be mapped to a rapid, aimed, error-free target acquisition task
  - The mouse is the test condition here
- If the user is trying to point with a touchpad, then touchpad is the test condition

Throughput – Design Implication

- The central idea is - Throughput provides a means to measure user performance for a given test condition
  - We can use this idea in design
- We collect throughput data from a set of users for different task difficulties
  - The mean throughput for all users over all task difficulties represents the average user performance for the test condition

Throughput – Design Implication

- Example – suppose we want to measure the performance of a mouse. We employ 10 participants in an experiment and gave them 6 different target acquisition tasks (where the task difficulties varied). From the data collected, we can measure the mouse performance by taking the mean throughput over all participants and tasks (next slide)

Throughput – Design Implication

D	W	ID (bits)	MT (sec)	TP (bits/s)
8	8	1.00	0.576	1.74
16	8	1.58	0.694	2.28
16	2	3.17	1.104	2.87
32	2	4.09	1.392	2.94
32	1	5.04	1.711	2.95
64	1	6.02	2.295	2.62
			Mean	2.57

Each value indicates mean of 10 participants

The 6 tasks with varying difficulty levels

Throughput = 2.57 bits/s

Speed Vs Accuracy tradeoff

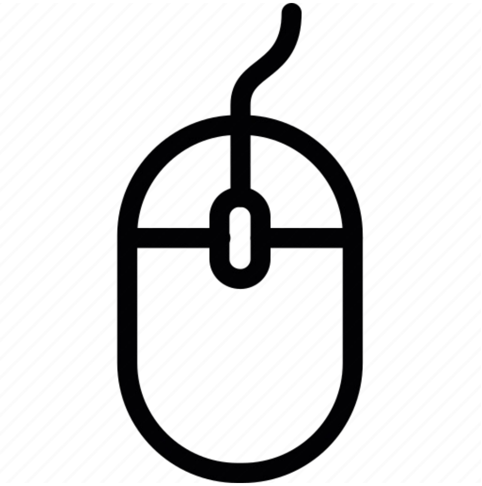
- Suppose, we are trying to select an icon by clicking on it. The icon width is D
- Suppose each click is called a "hit".
  - In a trial involving several hits, we are most likely to observe that not all hits lie within D (some may be just outside)

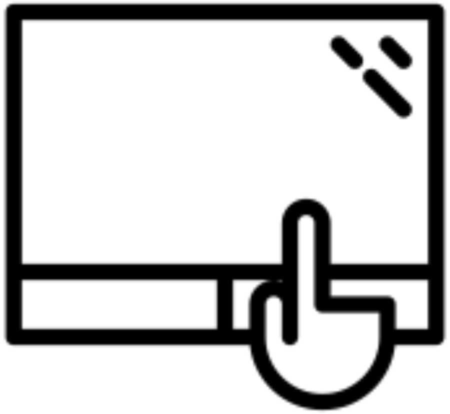
  - If we plot the *hit distributions (i.e., the* coordinates of the hits), we shall see that X% of the hits are outside the target boundary

- This is called the speed-accuracy trade-off
  - When we are trying to make rapid movements, we can not avoid errors

- How TP can help in comparing designs?
- How the Fitts' law can be used as a predictive model?

Suppose you have designed two input devices:

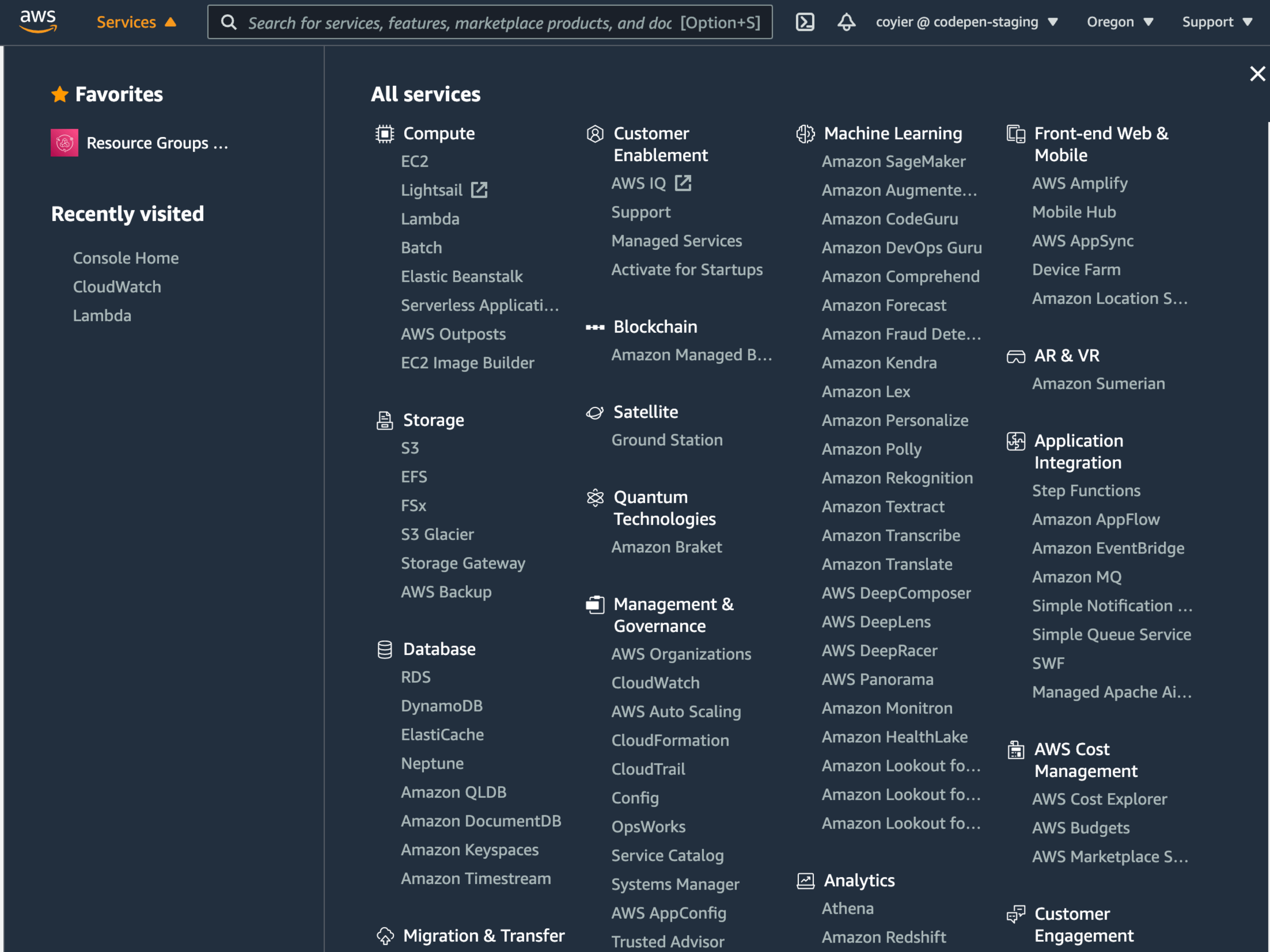




Determine performance for acquiring targets (e.g., for point and select tasks).

Compare TP1 and TP2

- If TP1>TP2, the mouse gives better performance
- The touchpad is better if TP1<TP2


aws
Services
Search for services, features, marketplace products, and doc [Option+S]
coyier @ codepen-staging
Oregon
Support
Favorites
Resource Groups ...
Recently visited
Console Home
CloudWatch
Lambda
All services
Compute
EC2
Lightsail
Lambda
Batch
Elastic Beanstalk
Serverless Applicati...
AWS Outposts
EC2 Image Builder
Storage
S3
EFS
FSx
S3 Glacier
Storage Gateway
AWS Backup
Database
RDS
DynamoDB
ElastiCache
Neptune
Amazon QLDB
Amazon DocumentDB
Amazon Keyspaces
Amazon Timestream
Migration & Transfer
Customer Enablement
AWS IQ
Support
Managed Services
Activate for Startups
Blockchain
Amazon Managed B...
Satellite
Ground Station
Quantum Technologies
Amazon Braket
Management & Governance
AWS Organizations
CloudWatch
AWS Auto Scaling
CloudFormation
CloudTrail
Config
OpsWorks
Service Catalog
Systems Manager
AWS AppConfig
Trusted Advisor
Machine Learning
Amazon SageMaker
Amazon Augmente...
Amazon CodeGuru
Amazon DevOps Guru
Amazon Comprehend
Amazon Forecast
Amazon Fraud Dete...
Amazon Kendra
Amazon Lex
Amazon Personalize
Amazon Polly
Amazon Rekognition
Amazon Textract
Amazon Transcribe
Amazon Translate
AWS DeepComposer
AWS DeepLens
AWS DeepRacer
AWS Panorama
Amazon Monitron
Amazon HealthLake
Amazon Lookout fo...
Amazon Lookout fo...
Amazon Lookout fo...
Analytics
Athena
Amazon Redshift
Front-end Web & Mobile
AWS Amplify
Mobile Hub
AWS AppSync
Device Farm
Amazon Location S...
AR & VR
Amazon Sumerian
Application Integration
Step Functions
Amazon AppFlow
Amazon EventBridge
Amazon MQ
Simple Notification ...
Simple Queue Service
SWF
Managed Apache Ai...
AWS Cost Management
AWS Cost Explorer
AWS Budgets
AWS Marketplace S...
Customer Engagement

Hick Nyman law

- Reaction time" (i.e., the time to react to a stimulus) of a person in the presence of "choices"

When a light comes on, how long does the operator takes to decide which button to press?

In the example,

- The “light on” is the stimulus
- We are interested to know the operator’s “reaction time” in the presence of the stimulus
- The operator has to decide among the 10 buttons (these buttons represent the set of choices)

The Hick-Hyman law can be used to predict the reaction times in such situations

- The law models human reaction time (also called *choice-reaction time) under uncertainty (the presence of choices)*

- The law states that the reaction (decision) time T increases with uncertainty about the judgment or decision to be made

$$T \propto H$$

$$T = kH$$

We can calculate H in terms of the choices in the following way

let, p_i be the probability of making the i-th choice

Then, $H = \sum_i p_i \log_2(1/p_i)$

Therefore,

$$T = k\sum_{i=1} p_i \log_2(1/p_i)$$

When all the probabilities of making choices becomes equal, we have $H = \log_2 N$ (N = no of choices)

- In such cases, $T = k \log_2 N$

Then, what will be the operator's reaction time in our example?

- Here N = 10
- A button can be selected with a probability 1/10 and all probabilities are equal
- Thus, $T = k \log_2 10$

  $= 0.66$ ms (assuming $a = 0$, $b = 0.2$)

Pondering time

- What are the applications of Hick Hyman's Law?

Fitts'-Digraph Model

- The alternative approach makes use of the Fitts'-diagraph (FD) model
- FD model was proposed to *compute* user performance for a VK from layout specification
  - Layout in terms of keys and their positions
  - Performance in text entry rate

Fitts'-Digraph Model

- The FD model has three components
  - **Visual search time (RT)**: time taken by a user to locate a key on the keyboard. The Hick-Hyman law is used to model this time

$$RT = a + b \log_2 N$$

N is the total number of keys, a and b are empirically-determined constants

Fitts'-Digraph Model

- The FD model has three components
  - **Movement time (MT)**: time taken by the user to move his hand/finger to the target key (from its current position). This time is modeled by the Fitts' law

$$MT_{ij} = a' + b' \log_2 \left(\frac{d_{ij}}{w_j} + 1 \right)$$

MT_{ij} is the movement time from the source (i-th) to the target (j-th) key, d_{ij} is the distance between the source and target keys, w_j is the width of the target key and a' and b' are empirically-determined constants

Fitts'-Digraph Model

- The FD model has three components
  - **Digraph probability:** probability of occurrence of character pairs or digraphs, which is determined from a corpus

$$P_{ij} = f_{ij} / \sum_{i=1}^{N} \sum_{j=1}^{N} f_{ij}$$

P_{ij} is the probability of occurrence of the i-th and j-th key whereas f_{ij} is the frequency of the key pair in the corpus

Fitts'-Digraph Model

- Using the movement time formulation between a pair of keys, an average (mean) movement time for the whole layout is computed

$$MT_{MEAN} = \sum_{i=1}^{N} \sum_{j=1}^{N} MT_{ij} \times P_{ij}$$

The mean movement time is used, along with the visual search time, to compute user performance for the layout

Fitts'-Digraph Model

- Performance is measured in terms of characters/second (CPS) or words/minute (WPM)
- Performances for two categories of users, namely novice and expert users, are computed

Fitts'-Digraph Model

- Novice user performance: they are assumed to be unfamiliar with the layout. Hence, such users require time to search for the desired key before selecting the key

$$CPS_{Novice} = \frac{1}{RT + MT_{MEAN}}$$

$$WPM = CPS \times (60 / W_{AVG})$$

W_{AVG} is the average number of characters in a word. For example, English words have 5 characters on average

Fitts'-Digraph Model

- Expert user performance: an expert user is assumed to be thoroughly familiar with the layout. Hence, such users don't require visual search time

$$CPS_{Expert} = \frac{1}{MT_{MEAN}}$$

$$WPM = CPS \times (60 / W_{AVG})$$

W_{AVG} is the average number of characters in a word. For example, English words have 5 characters on average

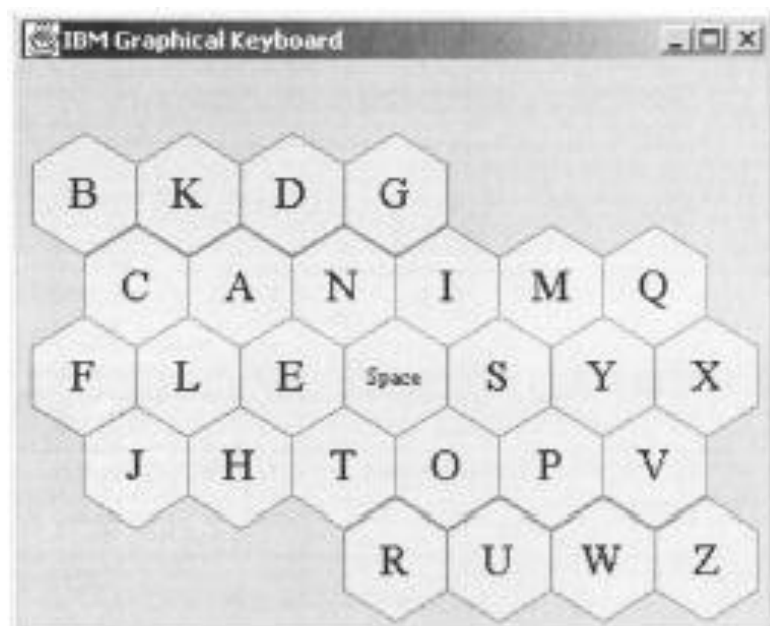
Some VK Layouts with Performance

- The layouts mentioned before were not designed using models
- They were designed primarily based on designer's intuition and empirical studies
- However, the performances shown are computed using the FD model

Some VK Layouts with Performance

- ATOMIK – a layout designed using slightly modified Metropolis algorithm

- Performance of the ATOMIK layout
  - 41.2 WPM (novice)
  - 67.2 WPM (expert)



IE 403

Dialogue Design

Week 9-Lec 2

what is dialogue?

- conversation between two or more parties
  - usually cooperative
- in user interfaces
  - refers to the *structure* of the interaction
  - syntactic level of human–computer ‘conversation’
- levels
  - lexical – shape of icons, actual keys pressed
  - syntactic – order of inputs and outputs
  - semantic – effect on internal application/data

- Input events
- State machines

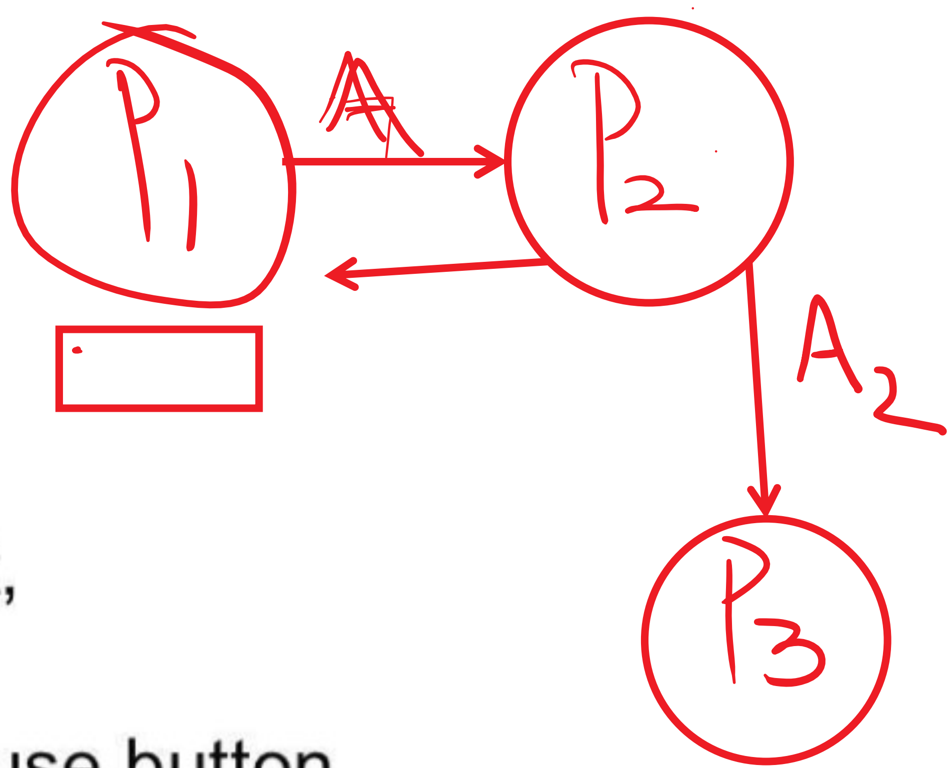
Raw Vs Translated Input Events

- The usual input hardware has state:
  - ~100 keys on the keyboard (down or up)
  - (x,y) mouse cursor position on the screen
  - one, two, or three mouse buttons (down or up)
- A “raw” input event occurs when this state changes
  - key pressed or released
  - mouse moved
  - button pressed or released

- Clicking
- Double-clicking
- Character typed
- Entering or exiting an object’s bounding box

Properties of Input event

- Mouse position (X,Y)
- Mouse button state
- Modifier key state (Ctrl, Shift, Alt,
- Keyboard key, character, or mouse button that changed



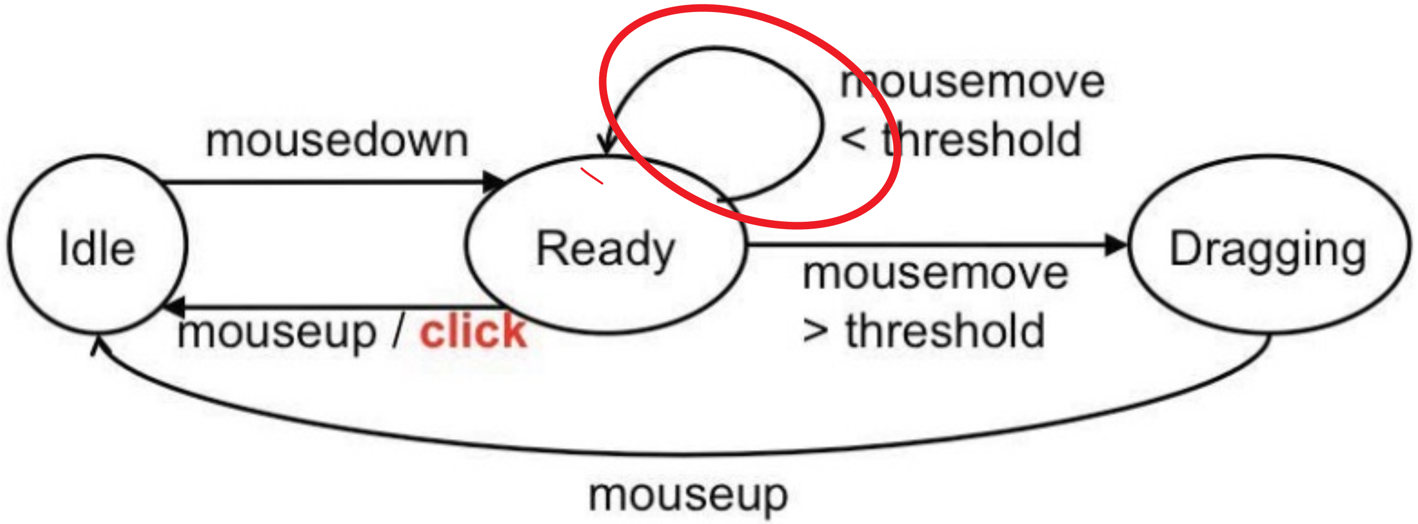
State Transition Diagrams/Networks

Assumes
"dialog essentially refers to a progression from one state of the system to the next in the system state space"

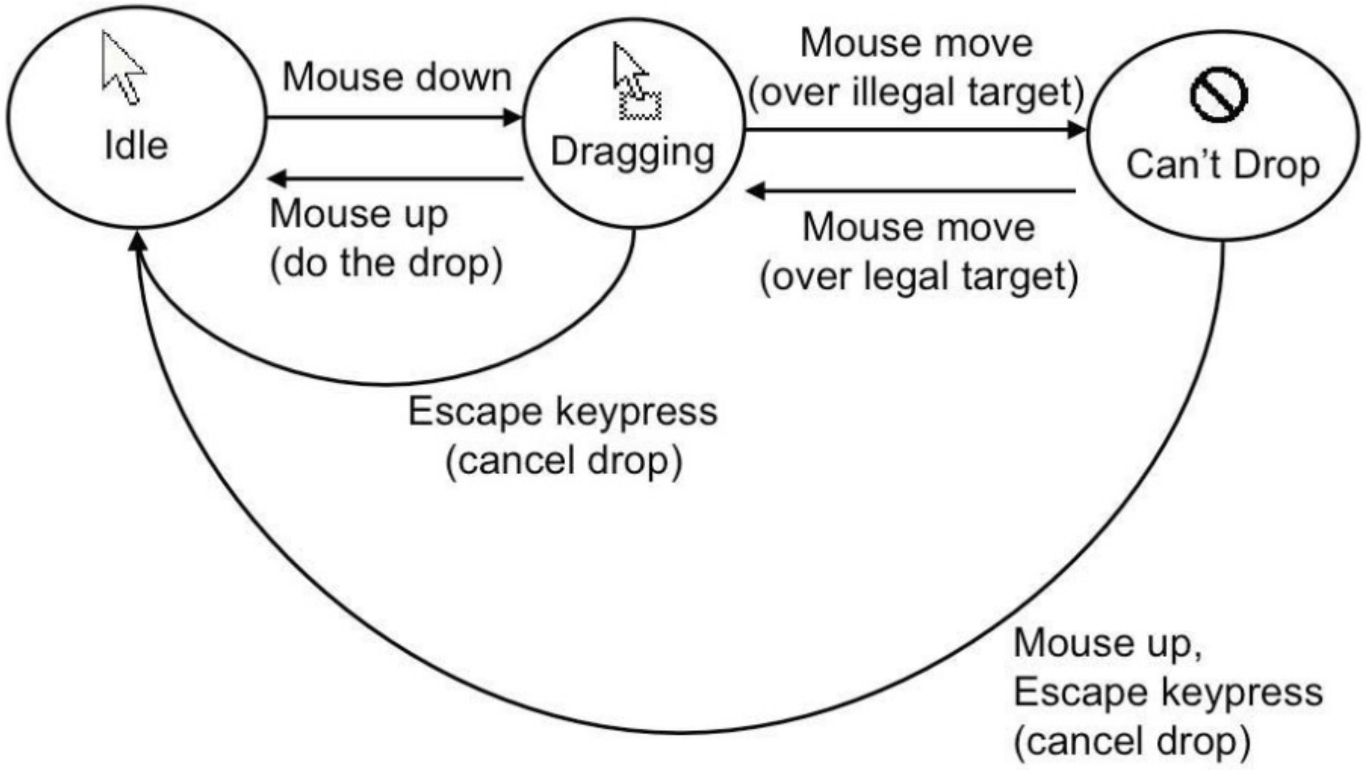
- *Procedural Vs Declarative*

- circles - states
- arcs - actions/events

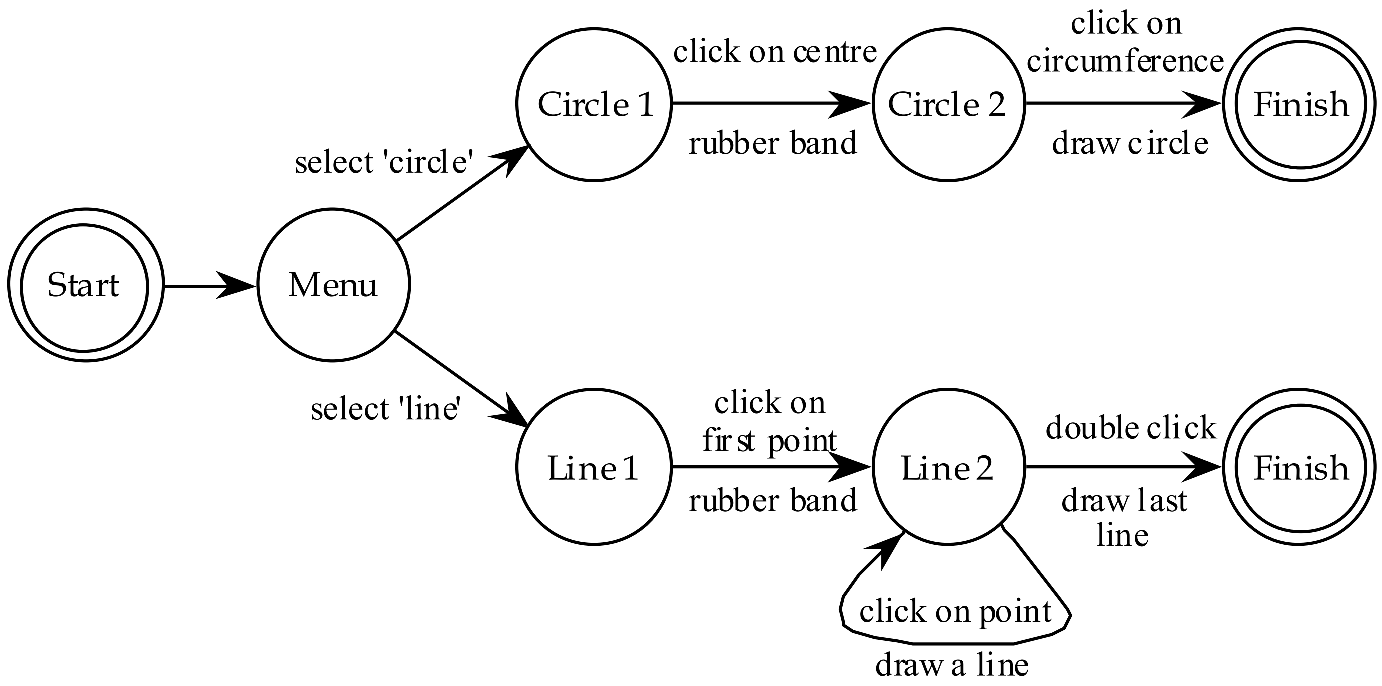
State Machine for generating Click event



STN for Drag and Drop

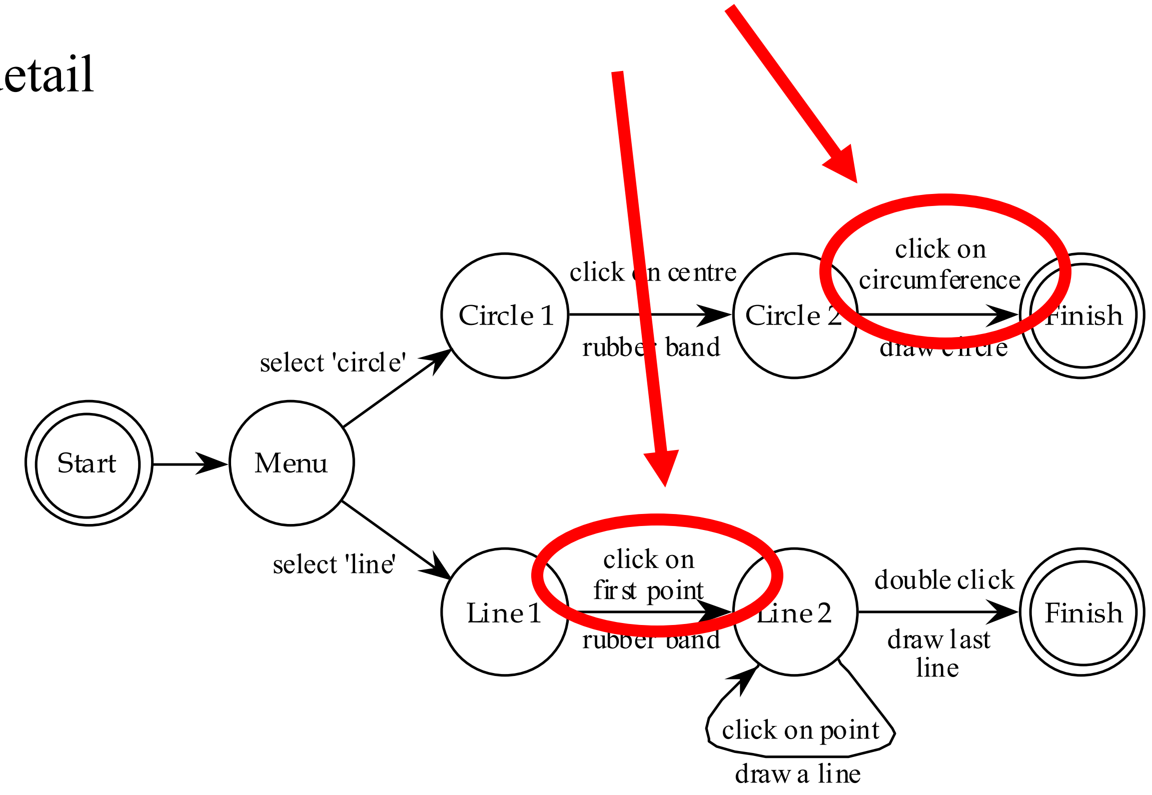


State transition networks (STN)



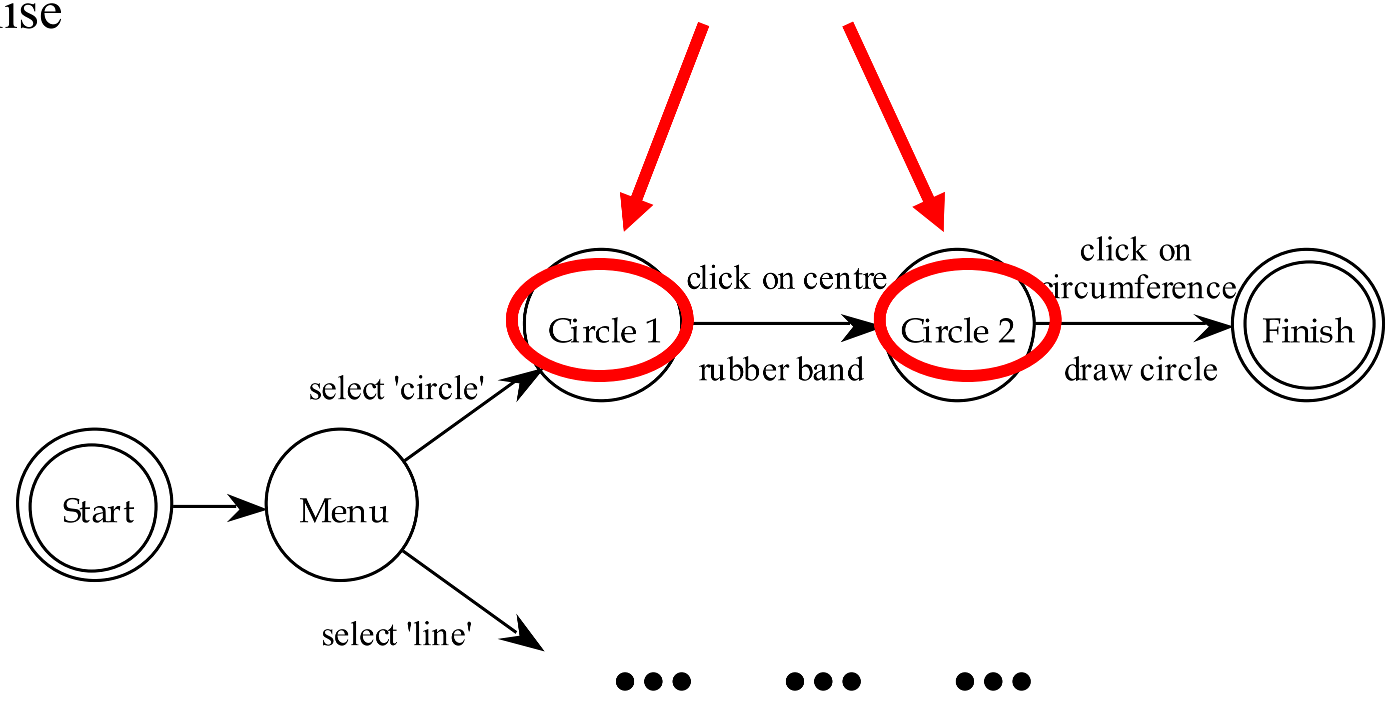
State transition networks - events

- arc labels a bit cramped because:
  - notation is `state heavy‘
  - the events require most detail



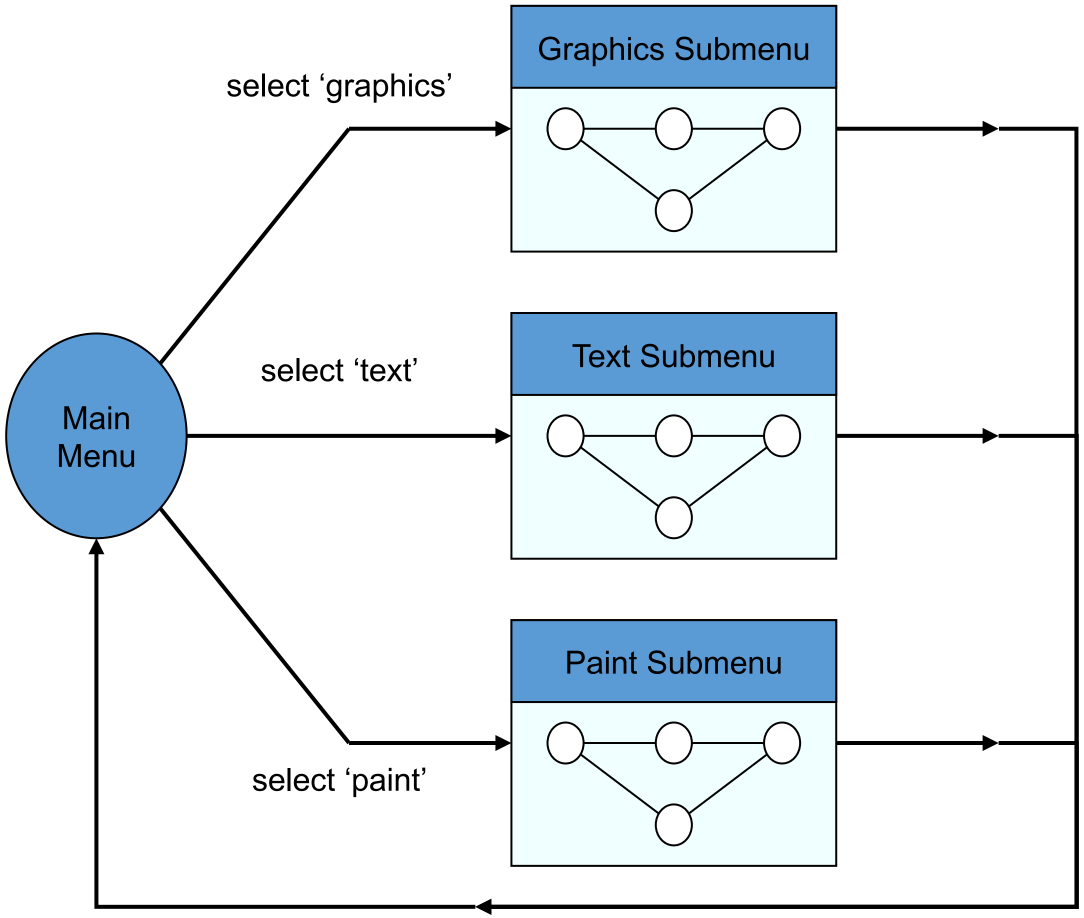
State transition networks - states

- labels in circles a bit uninformative:
  - states are hard to name
  - but easier to visualise



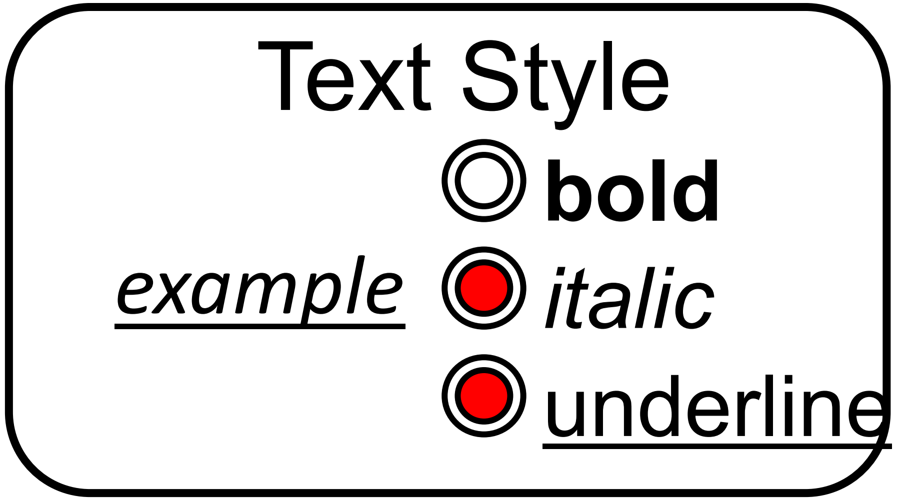
Hierarchical STNs

- managing complex dialogues
- named sub-dialogues



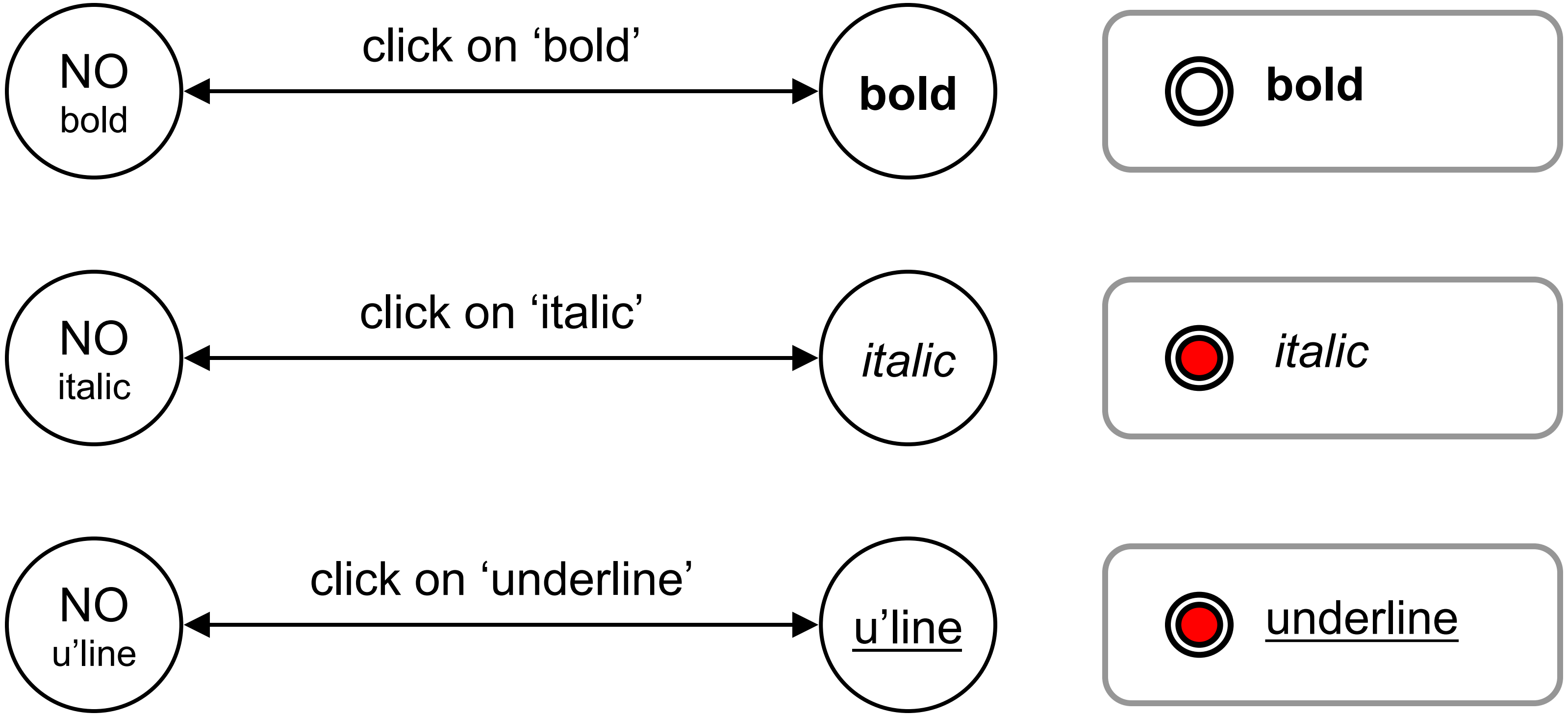
Concurrent dialogues - I

simple dialogue box



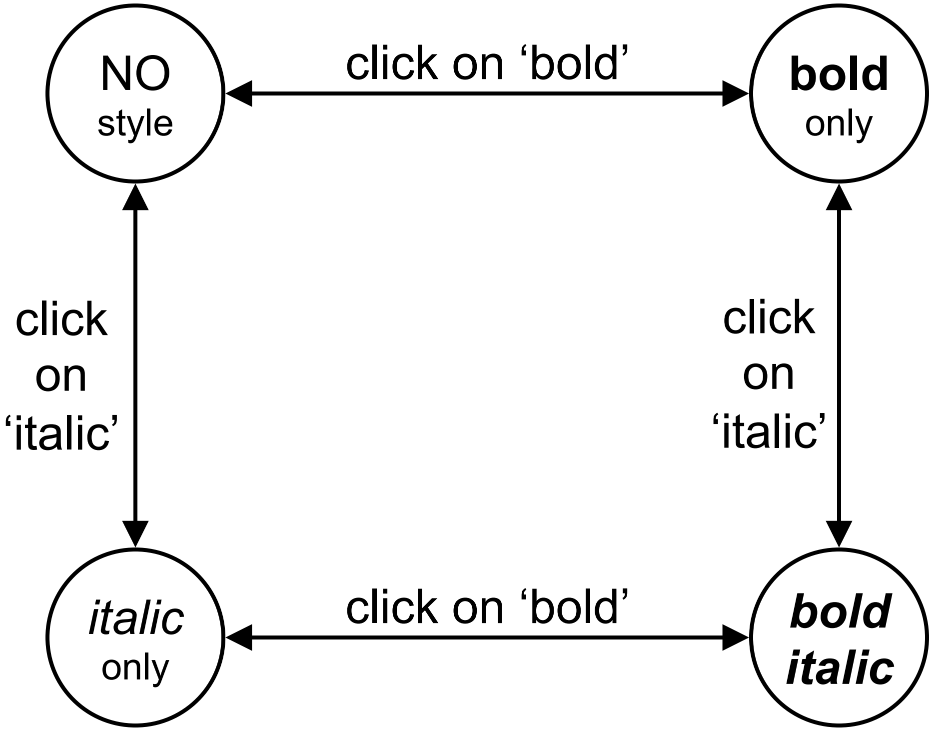
Concurrent dialogues - II

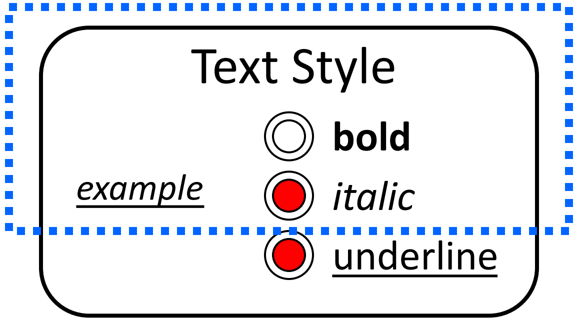
three toggles - individual STNs



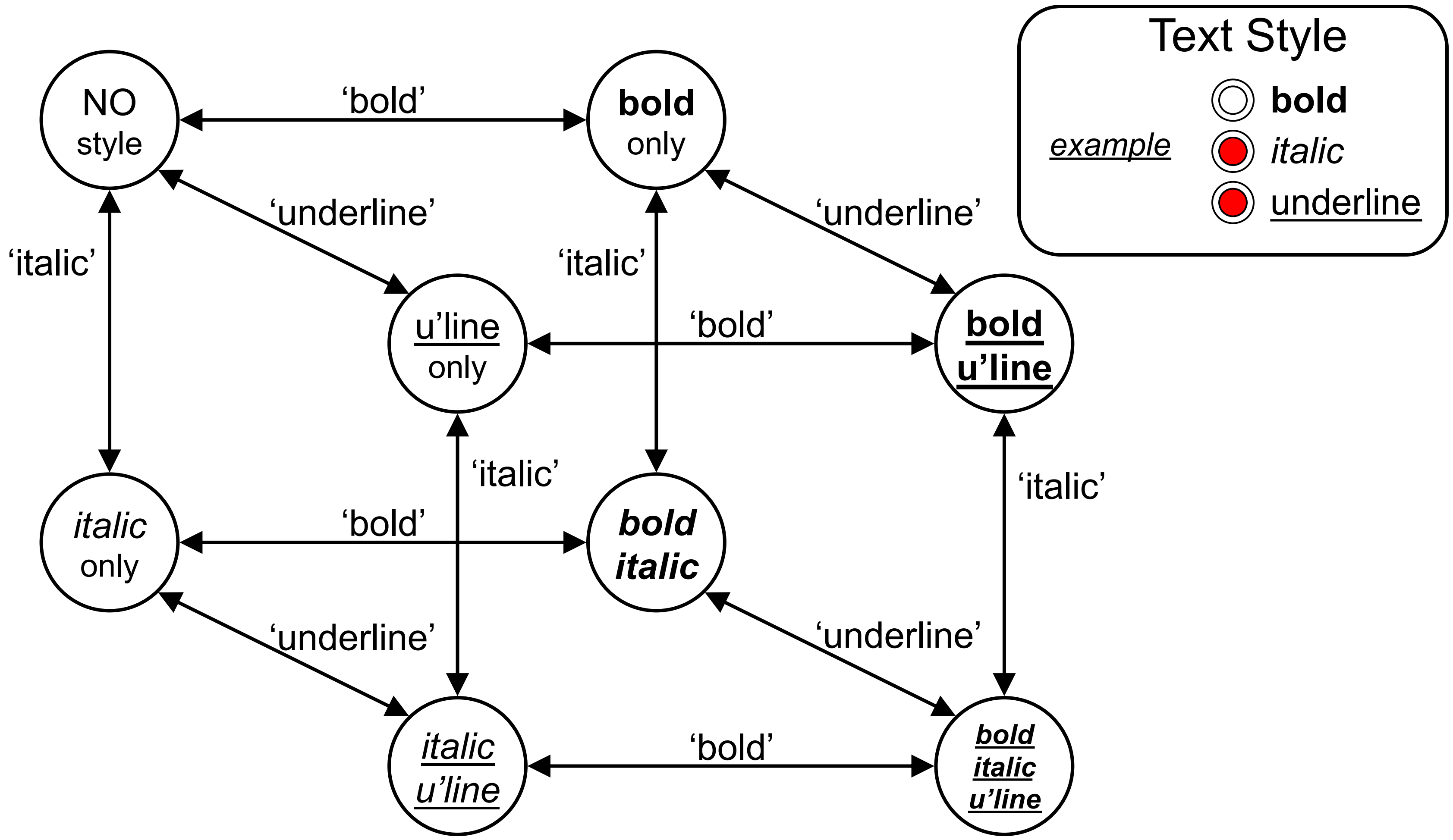
Concurrent dialogues - III

bold and italic combined





Concurrent dialogues - IV
all together - combinatorial explosion



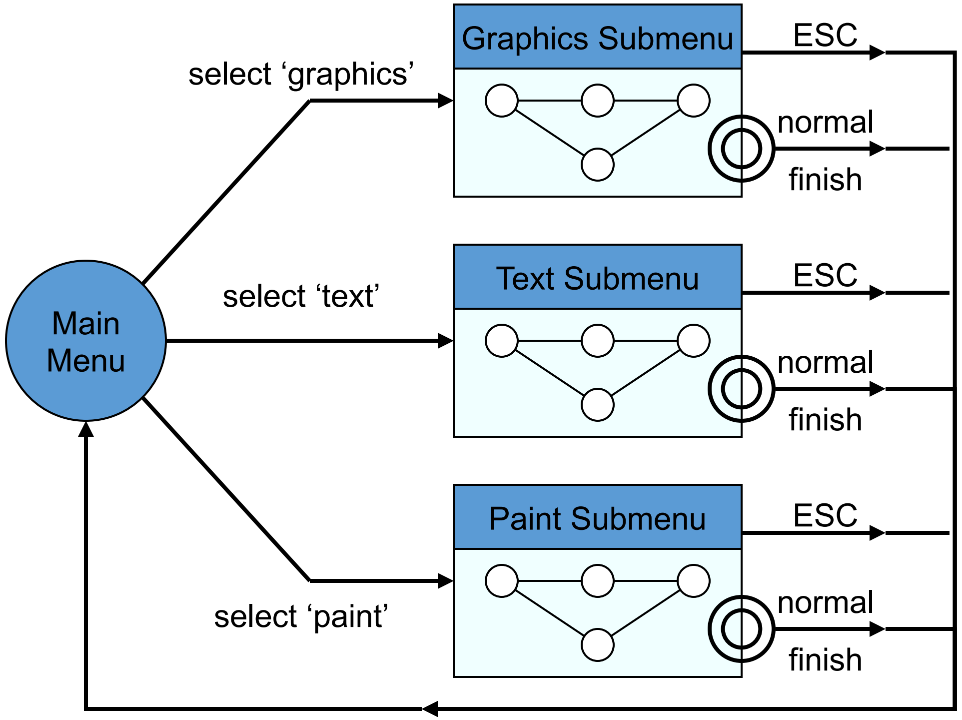
escapes

- ‘back’ in web, escape/cancel keys
  - similar behaviour everywhere
  - end up with spaghetti of identical behaviours
- try to avoid this

e.g. on high level diagram

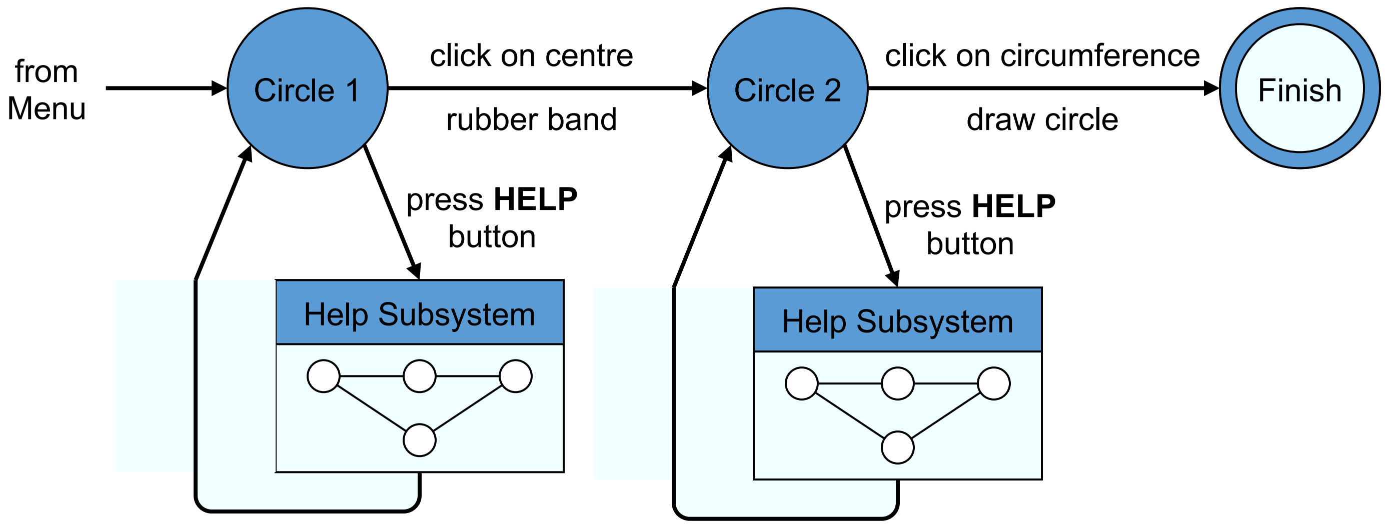
‘normal’ exit for each submenu

plus separate escape arc active ‘everywhere’ in submenu

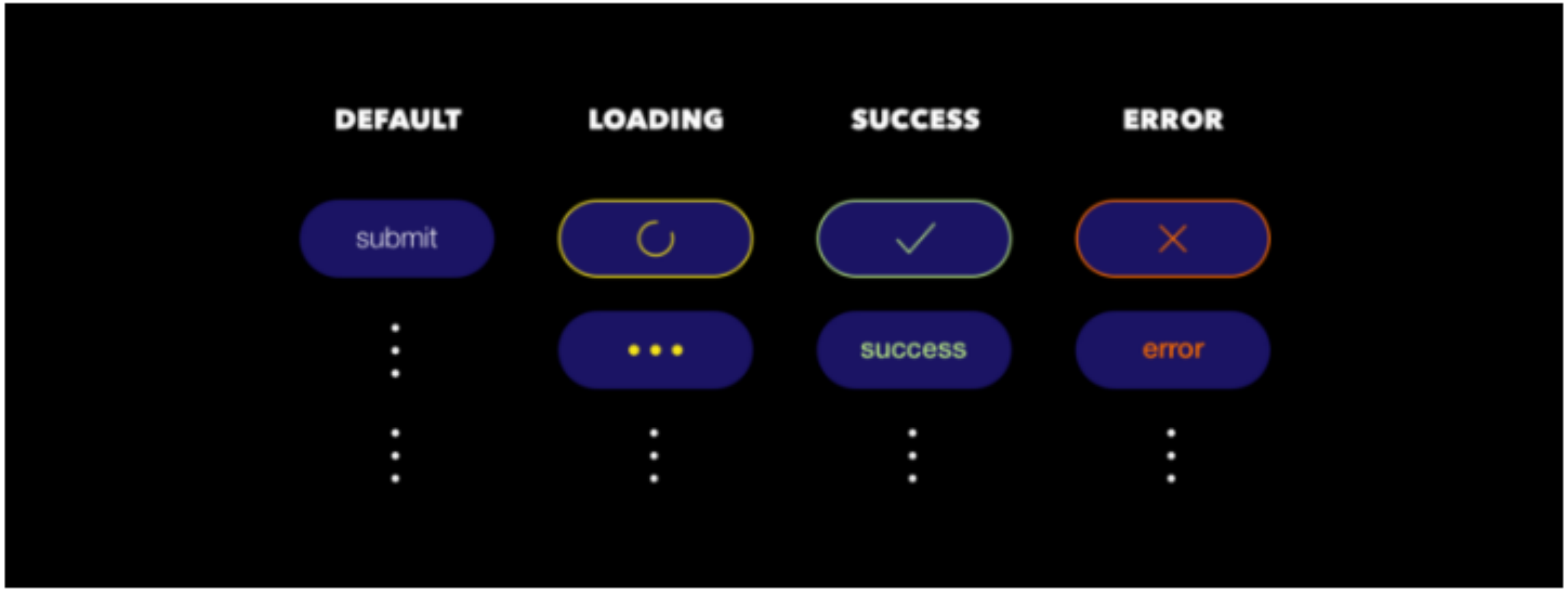


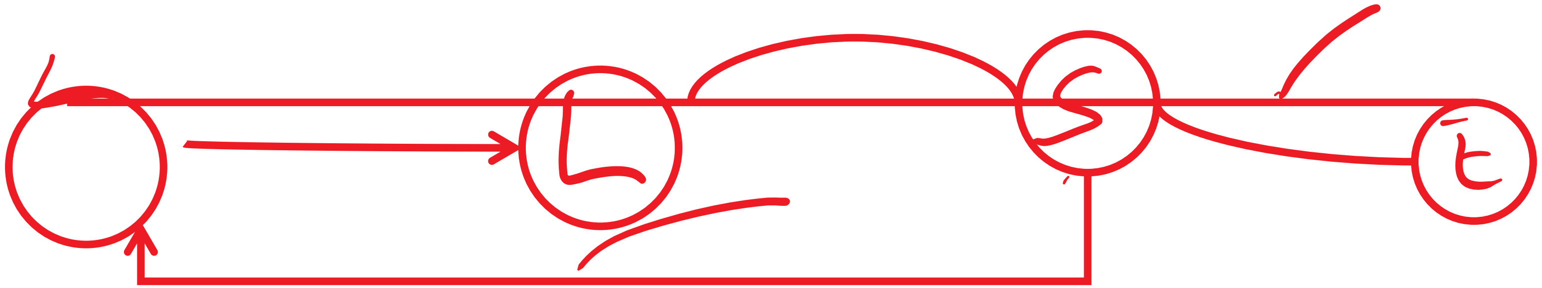
help menus

- similar problems
  - nearly the same everywhere
  - but return to same point in dialogue
  - could specify on STN … but very messy
  - usually best added at a ‘meta’ level

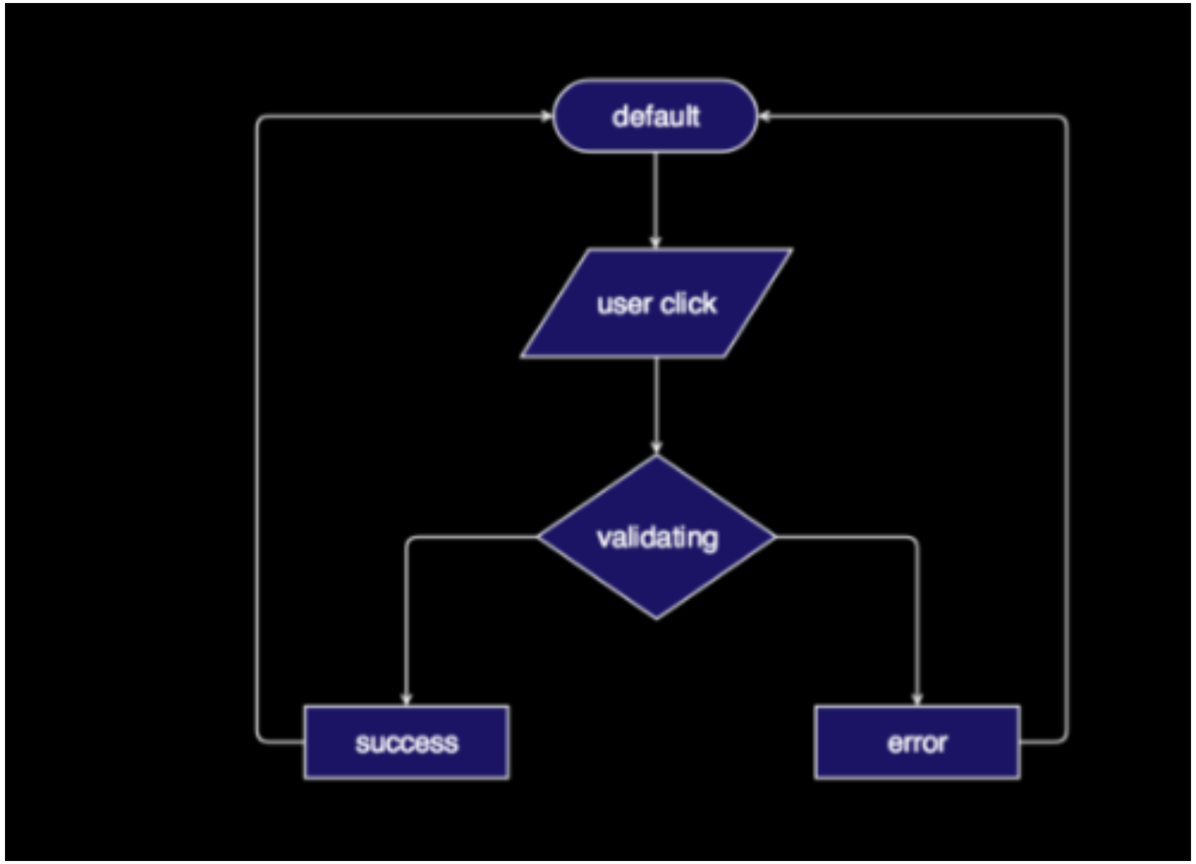


Modeling a Submit button through FSM/STN





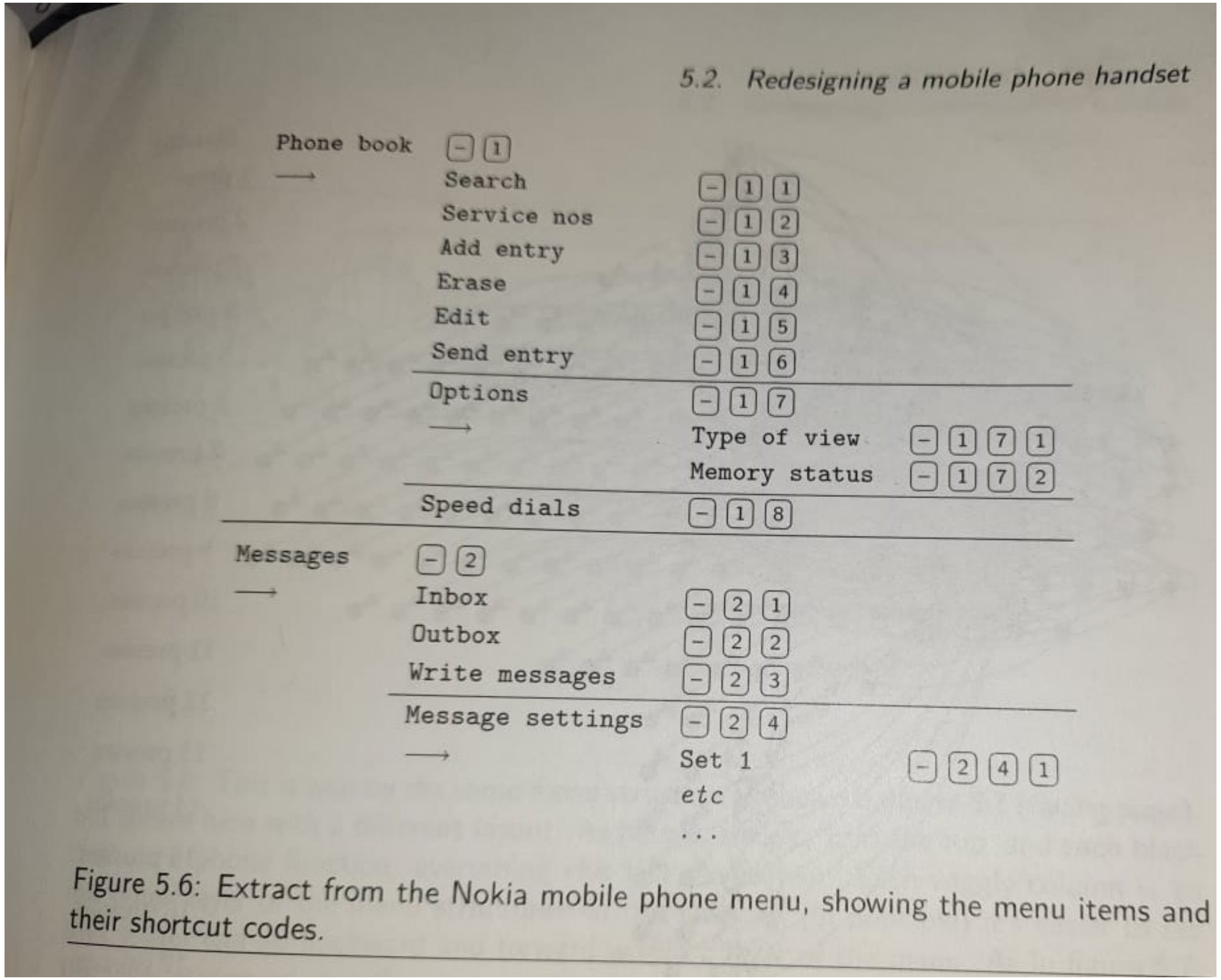
Can a flowchart help with state info?



- Where is state info?
- Where is indication for which state to which?
- Where are the actions that indicate transition of states?

From State	Input	To State
default	click	validating (loading)
validating (loading)	validate_success validate_error	success error
success	10 seconds later	default
error	10 seconds later	default

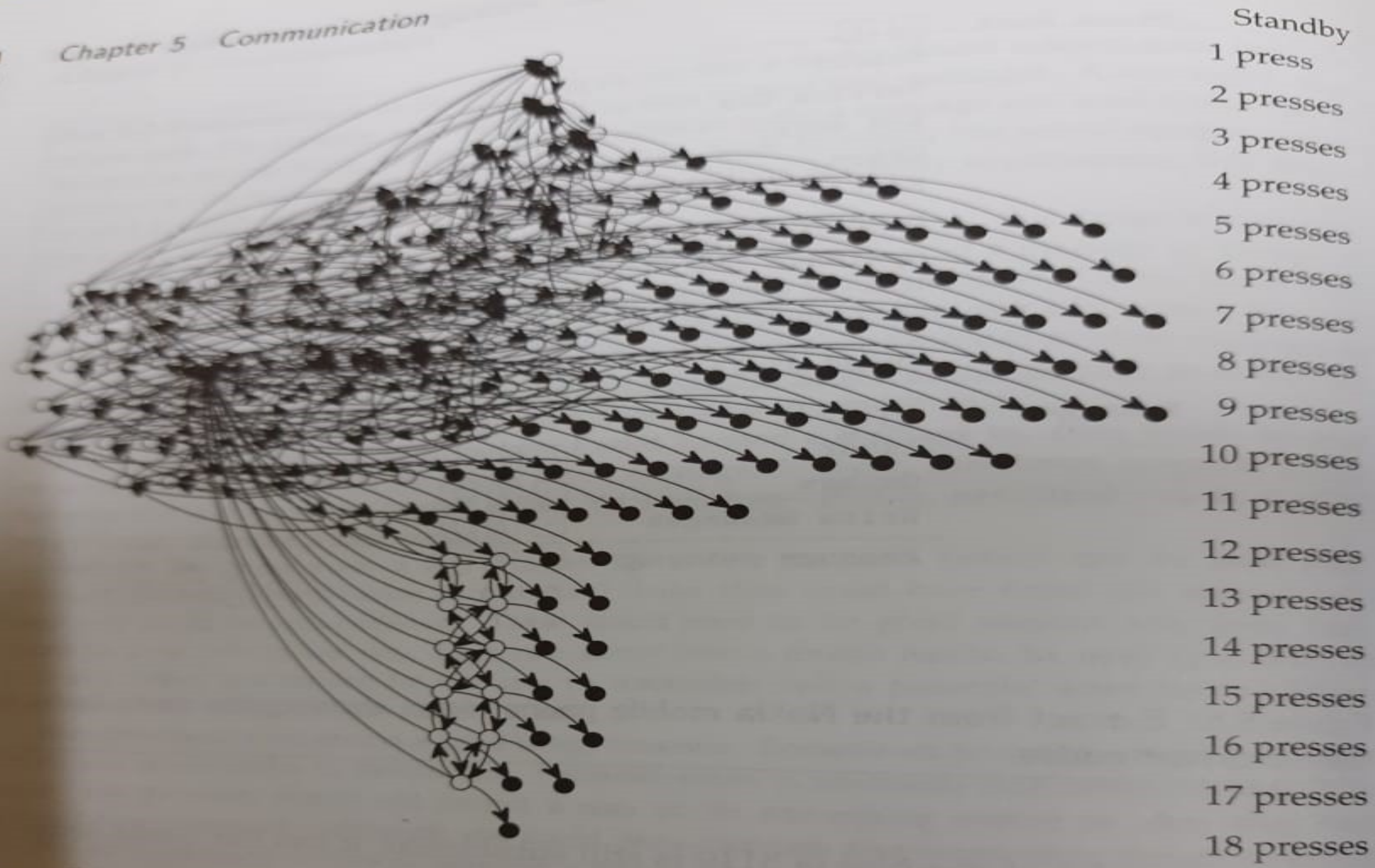
https://www.freecodecamp.org/news/designing-ui-states-and-communicate-with-developers-effectively-by-fsm-fb420ca53215/



Phone book	[–] [1]			
⟶	Search	[–] [1] [1]		
	Service nos	[–] [1] [2]		
	Add entry	[–] [1] [3]		
	Erase	[–] [1] [4]		
	Edit	[–] [1] [5]		
	Send entry	[–] [1] [6]		
	Options	[–] [1] [7]		
	⟶	Type of view	[–] [1] [7] [1]	
		Memory status	[–] [1] [7] [2]	
	Speed dials	[–] [1] [8]		
Messages	[–] [2]			
⟶	Inbox	[–] [2] [1]		
	Outbox	[–] [2] [2]		
	Write messages	[–] [2] [3]		
	Message settings	[–] [2] [4]		
	⟶	Set 1	[–] [2] [4] [1]	
		etc		
		...		

Figure 5.6: Extract from the Nokia mobile phone menu, showing the menu items and their shortcut codes.

- 188 Menu Items
- 84 user selectable functions



igure 5.7: The menu tree for the Nokia 5110 mobile phone. Phone functions are epresented by black dots, and all the white dots are intermediate menus. For clarity, rrows going out of phone functions (e.g., many go back to "Standby") have not een shown. The diagram is a ranked embedding: drawn so that each row is the ame minimum number of presses from "Standby." To use the device, start at the p, and each *correct* key press will move you down a row toward the function you ant—assuming you know the optimal route and you make no errors.

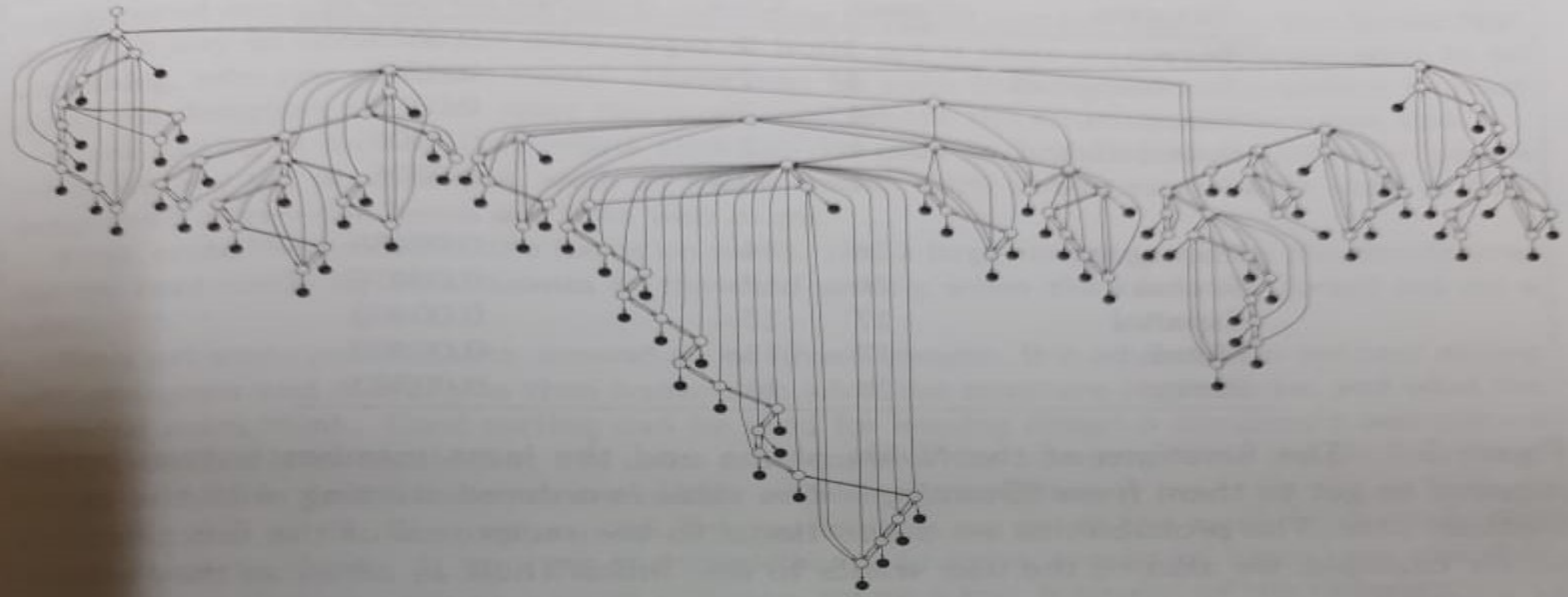


Figure 5.8: This is exactly the same menu structure as shown in figure 5.7 (facing page), but shown here with a different layout. As before, standby is at the top, and each black circle is a phone function; everything else is a menu item. Each wiggly column is an individual layer of the menu structure—in this diagram (in principle) it's easier to see that a user can go backward and forward within a layer of the menu. As in figure 5.7, the back arrows to standby are not shown, as they merely add a lot of clutter. The arrowheads on the lines are too small to see at this scale, but graphs as complex as this would normally be viewed with suitable visualization tools that can pan and zoom.

▷ This diagram was drawn by Dot, the graph-drawing tool discussed in section 9.5.1 (p. 291).

Function	*Presses*	*Rank*	*Assumed probability*
Search	3	1	0.0613
Incoming call	4	2=	0.0306
Inbox	4	2=	0.0306
Speed dials	4	2=	0.0306
Service nos	4	2=	0.0306
...	...	...	...
Português	16	14=	0.00438
Svenska	16	14=	0.00438
Español	17	15=	0.00408
Norsk	17	15=	0.00408
Suomi	18	16	0.00383

Figure 5.9: The functions of the Nokia phone and the least number button pres required to get to them from "Standby." The table is ordered starting with the eas function first. The probabilities are proportional to the reciprocal of the function ra so, for example, we assume the user wants to do "Inbox" half as often as they want do a search.

IE 403

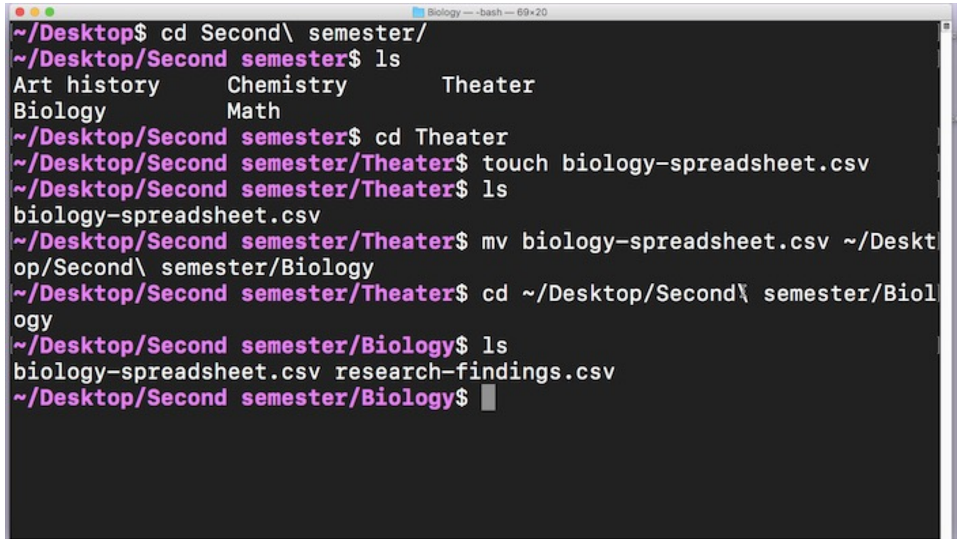
Human-Computer Interaction

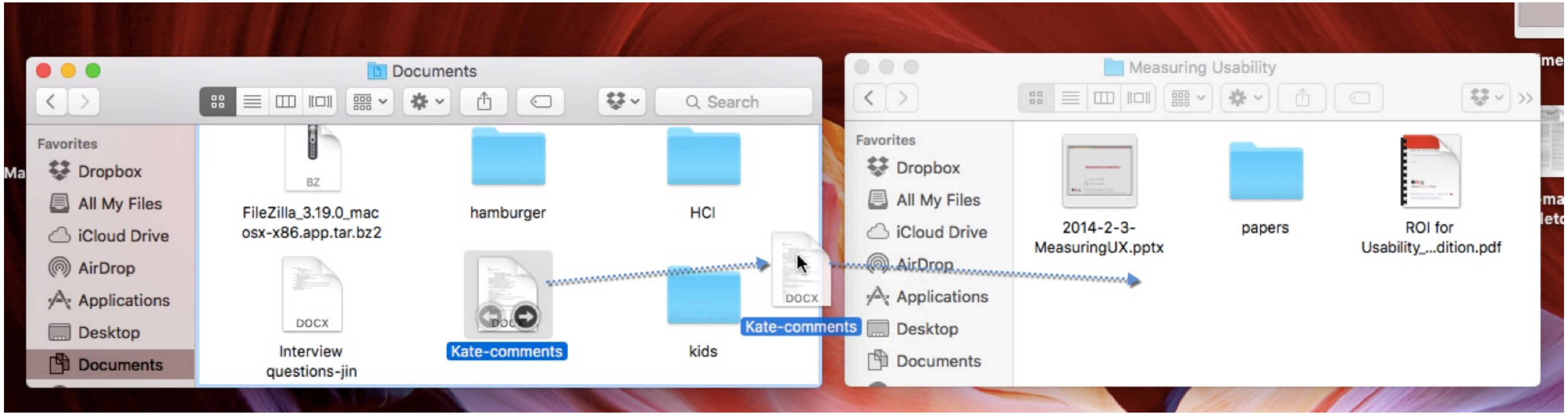
Week 10-Lec 1

Direct Manipulation Interfaces

A *direct manipulation interface allows a user to* directly act on a set of objects in the interface.

Moving File: CLI Vs Dragging





Pros of DMI

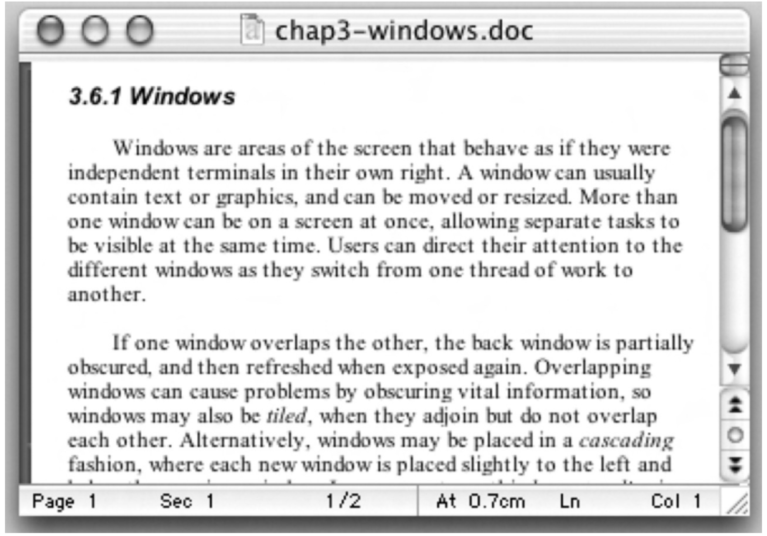
- While interacting with DM interfaces, users feel as if they are
  - *interacting with the domain*
  - rather than with the interface,
- *Focus on the task rather than on the technology.*
- Direct involvement with a world of task objects rather than communication with an intermediary.
- No more Clear distinction between I/P and O/P
  - Document icon as input by user but represents an output state of a file/application

Direct Manipulation

- Windows, Icons, Menus, Pointers, aka WIMP, Forms, (other objects)
- Objects can be acted upon directly by user with pointing device
- Choice of forms and icons important
  - a. match:
    designer's representation - user's understanding
  - b. choice of metaphors
  - c. explicit attention to ways in which instantiated and 'real' objects differ
  - d. cultural bias of representation

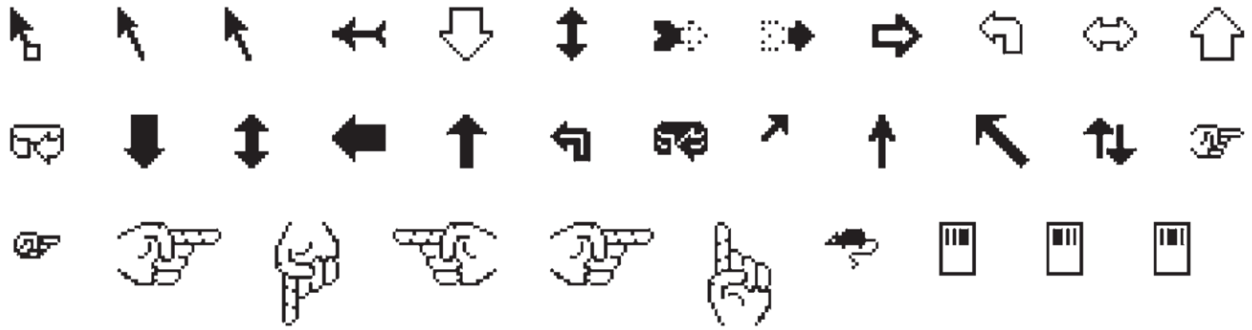
WIMP Interface

Windows

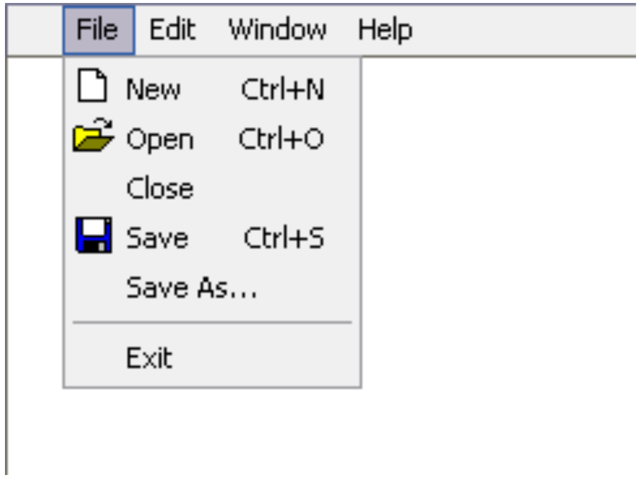


Icons

Pointers



Menus



Pointing Devices

Image courtesy of Louisa Billeter on Flickr.

Mouse

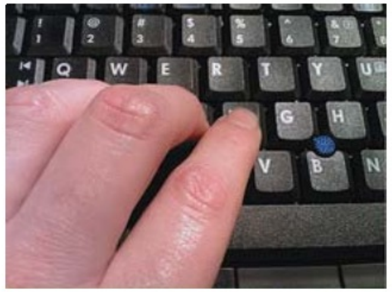


Image courtesy of Amarand Agasi on Flickr.

Joystick

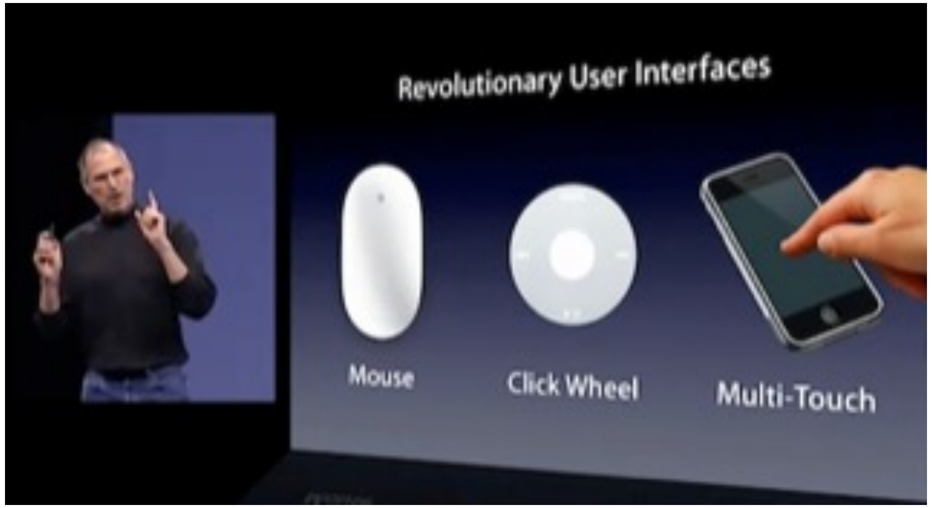
Image courtesy of Frau Bob on Flickr.

Touchpad

Image courtesy of Mike Marttila on Flickr.

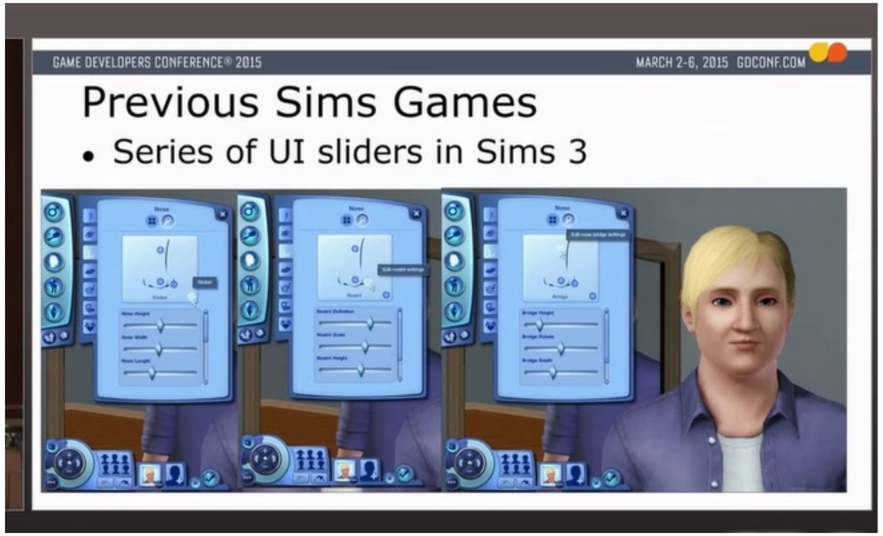
Trackball

- Mouse = Indirect pointing
- Stylus/finger = direct pointing



Video games

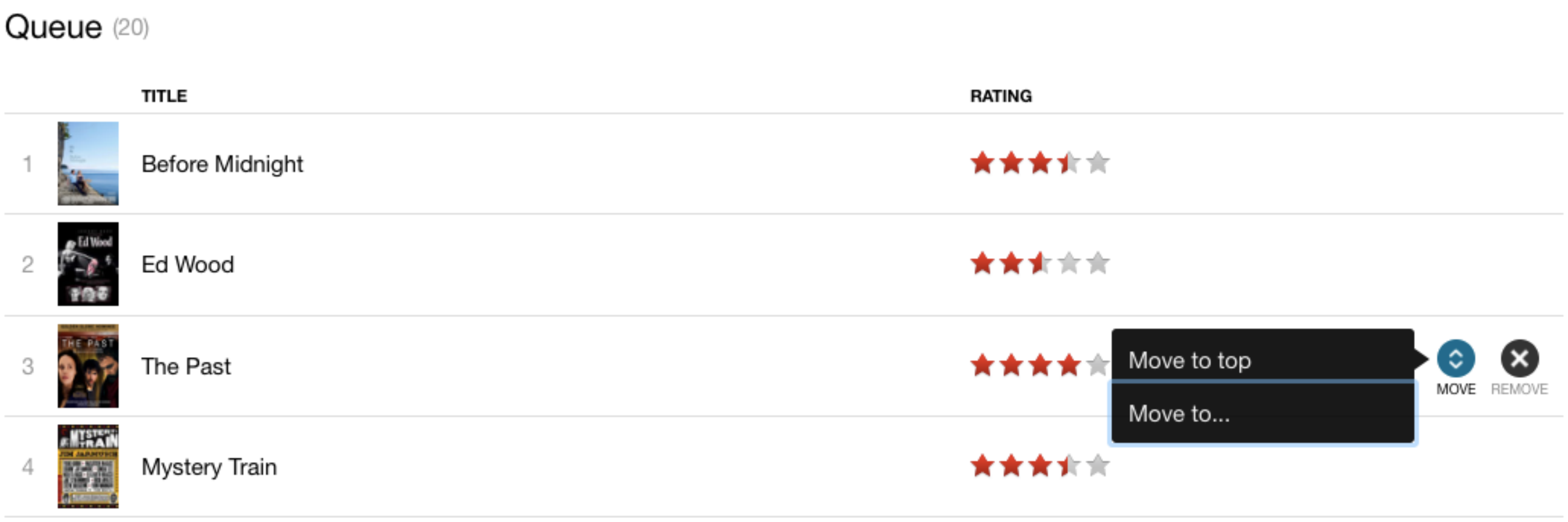
- Nintendo Wii, Sony PlayStation, Microsoft Xbox, Kinect
- Field of action is visual and compelling
- Commands are physical actions whose results are immediately shown on the screen
- No syntax to remember
- Most games continuously display a score
- Direct manipulation in "the Sims"
- Second Life virtual world
- Spore
- Myst well received
- DOOM and Quake controversial



Visual representation of objects and tasks that can be manipulated in arbitrary orders by users: GUI

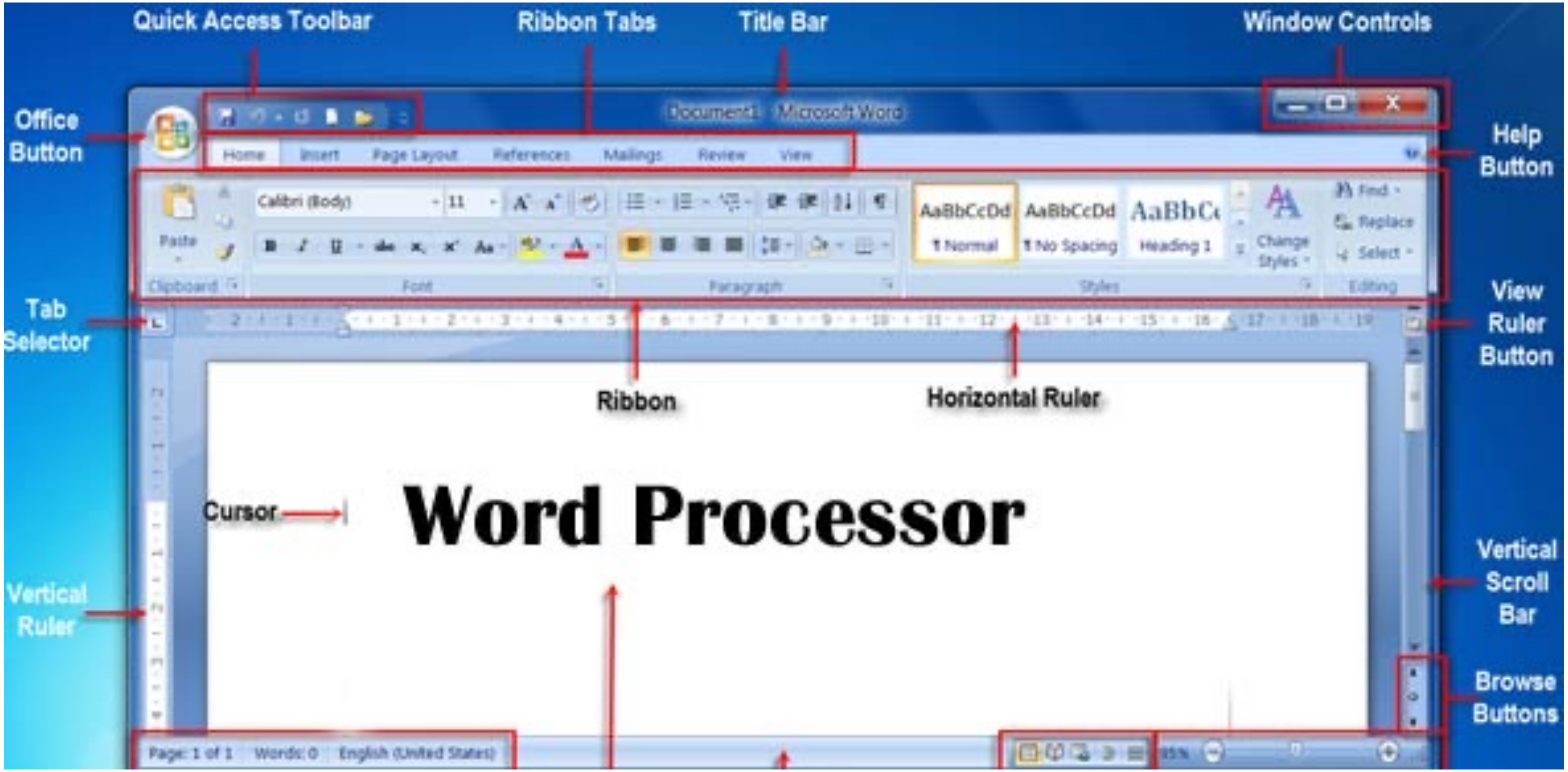
Advantages	*Disadvantages*
• visually presents task concepts • learnability • retention over time • error avoidance • encourages exploration • high subjective satisfaction	• can be difficult to program (especially error handling) • non-sighted users • *requires high-end equipment* • *New developments require to rethink the principles of Direct Interaction*

More Disadvantages: DM not always good



Rearranging movies in a queue: Typing
or dragging nth movie to mth position?

WYSIWYG – Word Processor



Ben Shneiderman’s Principles for DM

1. Visibility of the objects of interest
2. Incremental action at the interface with rapid feedback on all actions
3. Reversibility of all actions, so that users are encouraged to explore without severe penalties
4. Syntactic correctness of all actions, so that every user action is a legal operation
5. Replacement of complex command languages with actions to manipulate directly the visible objects (and, hence, the name direct manipulation).

Shneiderman’s 8 Golden Rules

1. *Strive for consistency*
2. *Cater to Universal Usability (Enable frequent users to use shortcuts)*
3. *Offer informative feedback*
4. *Design dialogs to yield closure*
5. *Offer error prevention and simple error handling*
6. *Permit easy reversal of actions*
7. *Support internal locus of control*
8. *Reduce short-term memory load*

Does Word processor satisfy Shneiderman’s criteria?

- TextBook material

Explanations & Examples

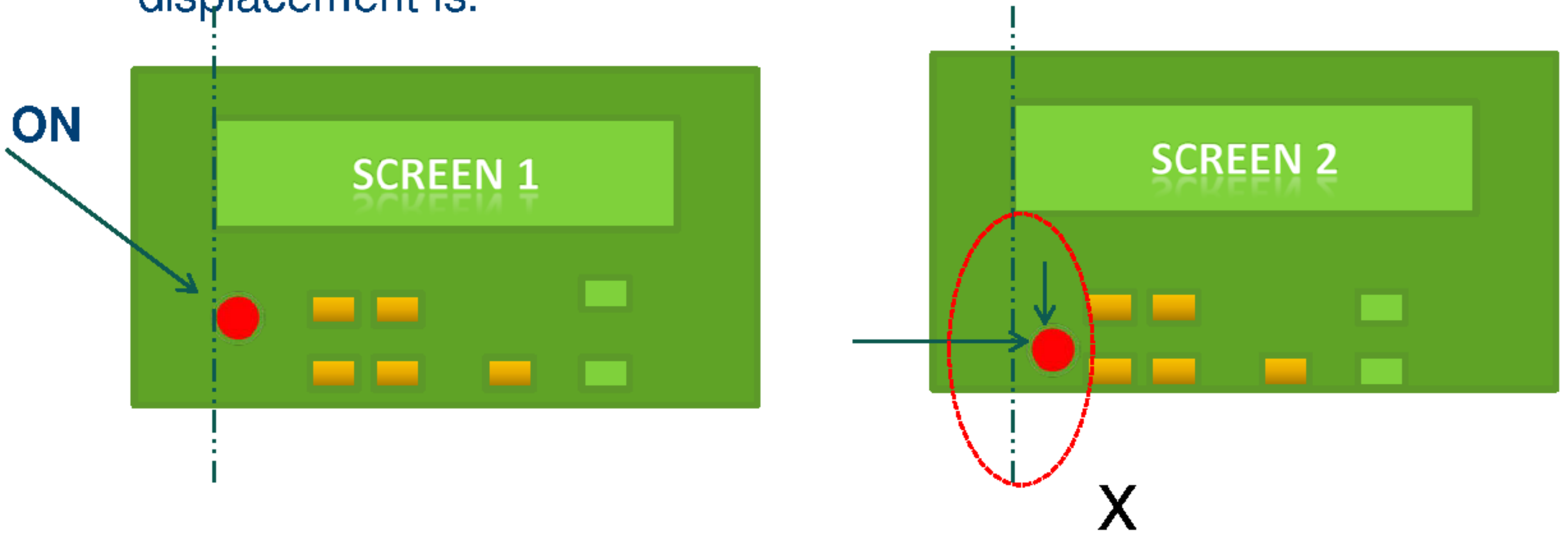
1. Strive for Consistency

- Users need to be able to do the same thing the same way that they have been doing.- every time.
- Interfaces need to exhibit 'consistent' quality across screens/ applications both visually as well as behaviorally.
- Consistency leads to a pattern which is easier to handle cognitively.
- Consistency such as 'similar sequence of actions in similar situations' makes it easy to learn.

Continued........

Consistency can be achieved through graphical elements such as fonts, colour, shape, position being consistently same in all menus & screens, across, categories for a particular software.

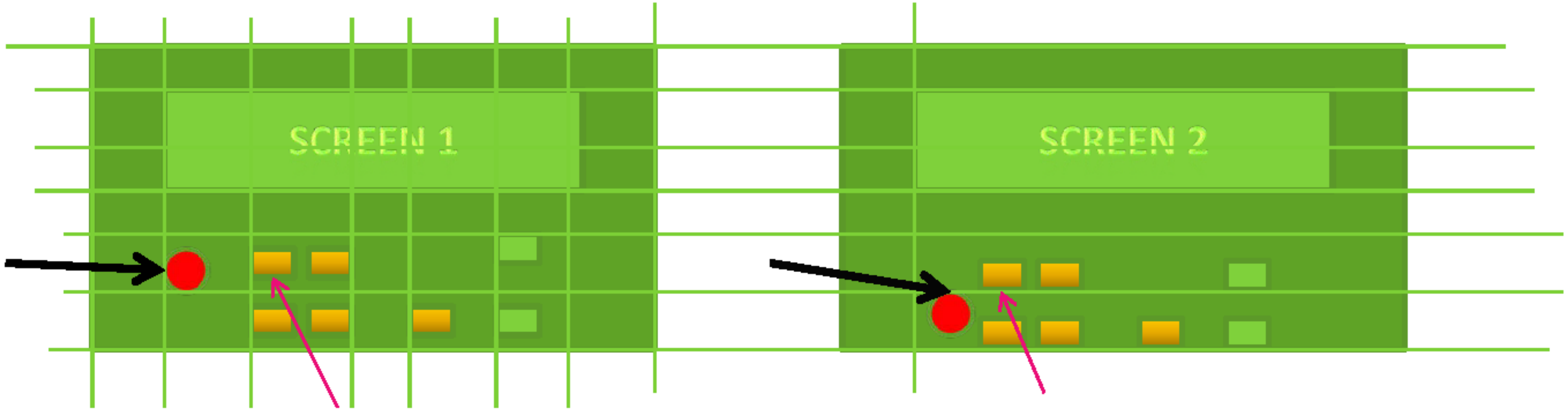
For example: If the **ON** button is on the right in the first screen and moves towards middle in the second screen then positional inconsistency is said to have occurred - however small the displacement is.



GUI designers use a simple technique to maintain consistency of control elements in successive screen.

Consistency….. continued

GUI designers use a background grid to place interactive elements in a consistent and orderly way so as to make them appear both physically as well a visually at the same place across the entire software package.



Inconsistent positioning of GUI elements is evident when observed against a grid. Grids are used as background reference to place the elements consistently

In case certain exceptions in maintaining consistency are required to be made in a subsequent screen, they should be such that they are comprehensible, distinct and limited in number.

2. Cater to wide range & type of Users

1. Strive for Consistency
2. **Cater to Universal Usability**
3. Offer Informative feedback
4. Design Dialogs to yield closure
5. Prevent Errors
6. Permit easy reversal of actions
7. Support internal locus of control
8. Reduce short term memory load

Universal design strives to cater to as wide a range of human users of different characteristics (age, culture, educational level, disability) with a single design.

While this may not be feasible or possible in all contexts, Shniderman's rule none the less needs to be followed so as not to leave out taking into consideration a section of users, other wise competent, who cannot use the interface due to no fault of theirs.

Users are classified as **Novice**, **Intermediate** and **Experts**. Experts tend to use lesser actions at a faster pace. Abbreviations short cuts keys etc are some of the techniques used.

Interfaces need to cater to all levels & classification of users: novice to experts.

3. Offer Informative feedback

1. Strive for Consistency
2. Cater to Universal Usability
3. **Offer Informative feedback**
4. Design Dialogs to yield closure
5. Prevent Errors
6. Permit easy reversal of actions
7. Support internal locus of control
8. Reduce short term memory load

- Interfaces need to not just to be communicative but also need to inform the 'user' in terms of learning & feed back which tells them that they are proceeding in the right direction.
- For every action of the user there needs to be a feedback – only then 'interaction' (in HCI) is said to take place. Specific error messages composed in a appositive tone give affirmative feedback without having to feel punitive.
- Unless the user gets a feed back he/she cannot proceed or becomes unsure of the correctness of the action.

4. Design Dialogs to yield closure

- In an interaction - dialogue needs to have a closure which is recognized by the user as end of an action.
- Sequence of actions need to proceed in a dialogue by engaging the user in a step by step manner.
- Like in a mathematical expression, every enclosing bracket needs a corresponding closing bracket. So also subsequence of actions needs to be grouped with intermittent closing of each sub group followed finally by a closer action of the group.

Ex: A message at the end of a sequence of events gives a feed back & closure of sending a SMS.

Continued…….

5. Prevent Errors

1. Strive for Consistency
2. Cater to Universal Usability
3. Offer Informative feedback
4. Design Dialogs to yield closure
5. **Prevent Errors**
6. Permit easy reversal of actions
7. Support internal locus of control
8. Reduce short term memory load

Interfaces need to minimize errors. Human Computer dialogue can be designed to minimize and prevent errors made by users.

There cloud be many reasons for users errors but the user himself or herself is not one of them ! Users can make errors while interacting with computers as well as while inputting / interpreting information.

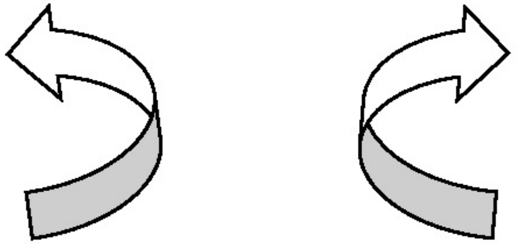
Even if the user makes an error the system needs top be designed to detect it, take corrective or precautionary steps to arrest it. It also needs to offer a way out for recovery from the error.

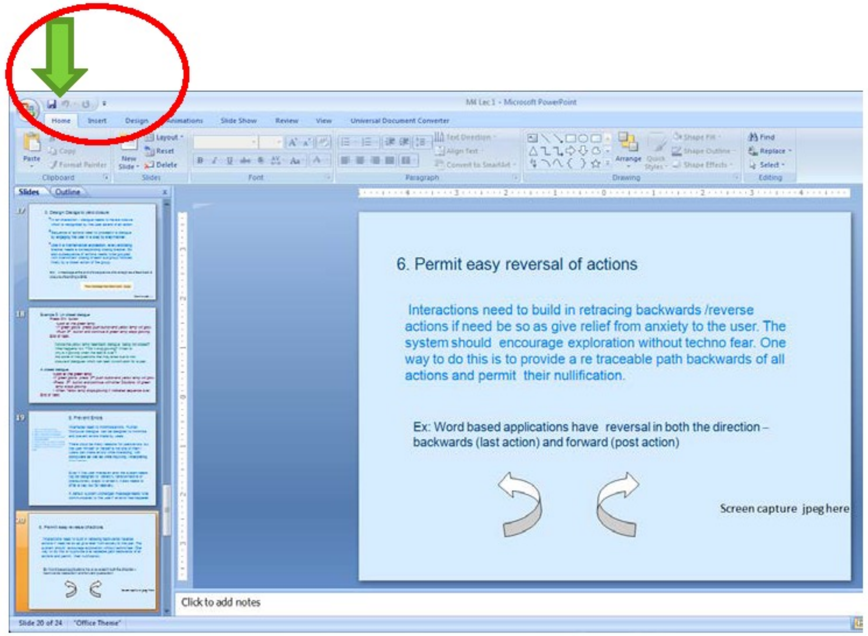
A default system unchanged message needs to be communicated to the user if an error has happened .

6. Permit easy reversal of actions

Interactions need to build in retracing backwards /reverse actions if need be so as give relief from anxiety to the user. The system should encourage exploration without techno fear. One way to do this is to provide a re traceable path backwards of all actions and permit their nullification.

Ex: This PPT application has reversal in both the direction – backwards (last action) and forward (post action)



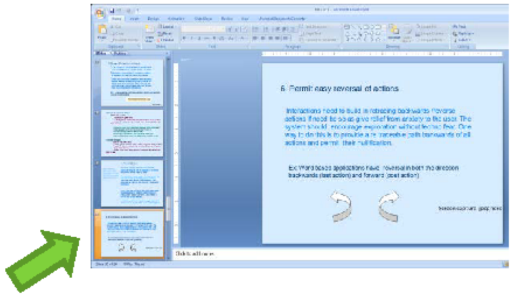


7. Support internal locus of control

Allow user to always feel ‘in control’ of the system and of the situation.

Make the user aware that he/she is in control. User should believe that they are controlling the system and not the other way around. This is achieved by more opportunities for ‘interactions’.

The bearing of where the user presently is helps the user to orient or reorient the interaction. The user should never be allowed to feel lost.



8. Reduce short term memory load

1. Strive for Consistency
2. Cater to Universal Usability
3. Offer Informative feedback
4. Design Dialogs to yield closure
5. Prevent Errors
6. Permit easy reversal of actions
7. Support internal locus of control
8. **Reduce short term memory load**

94 56 781029

Easier to remember if chunked into smaller setc

94 56 7 810 29

Care not load the cognitive short term memory of the user by expecting user to remember several sequences , actions and their consequences at a time. Means loading their short term memory while interacting.

Millers* 7 chunks of information is often prescribed as a solution to limit short term memory. In psychological experiments it has been found that the short term memory can hold 7 +- 2 bits called chunks of information. Long sequential actins requiring more than 7 chunks need to be broken down into smaller chunks.

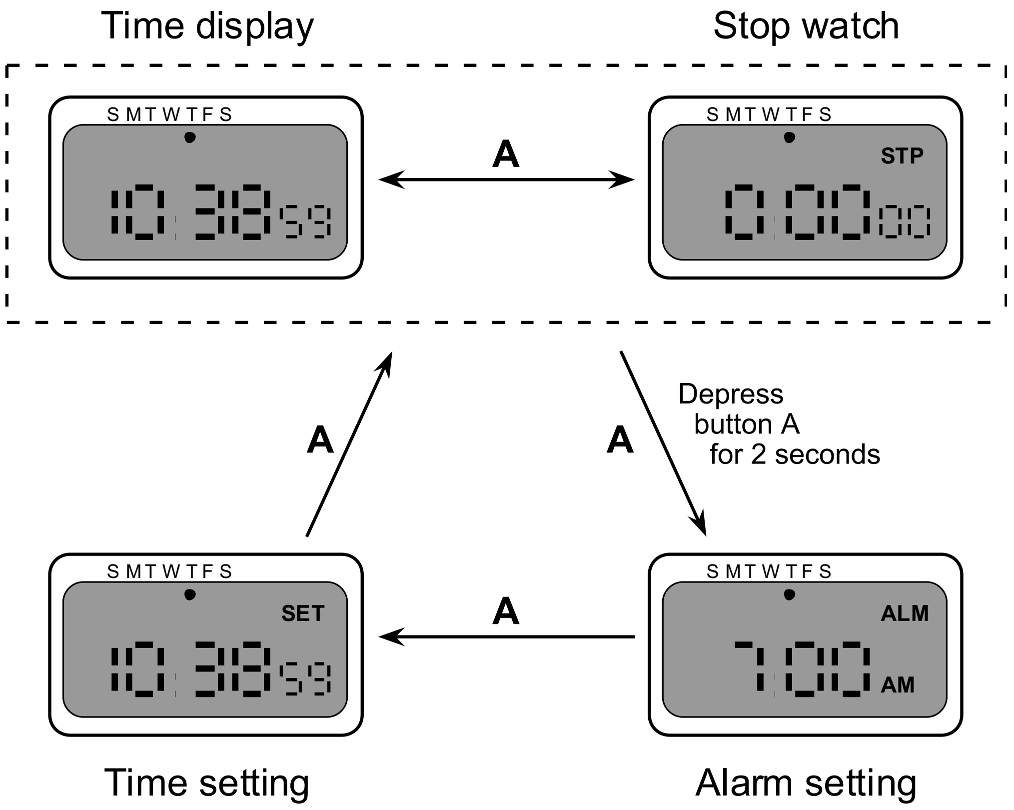
*G.A. Miller; The Magical number seven, plus or minus two: some limits on our capacity to process information. Psychplogical review, 63(2):81-97, 1956.

Conclusion

- GUI – based on DM
- Innovations in DM
  - Augmented Reality
  - Virtual Reality
  - Brain Machine Interfacing

Digital watch – User Instructions

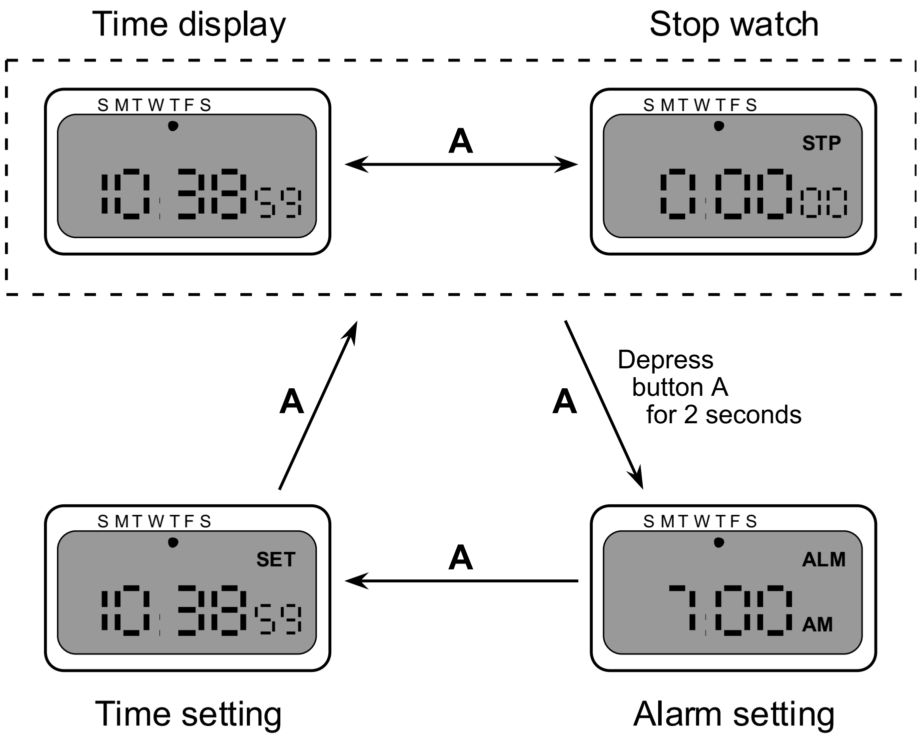
- two main modes
- limited interface
  - 3 buttons
- button A
  changes mode



Digital watch – User Instructions

- dangerous states
  - *guarded*
    … by two second hold

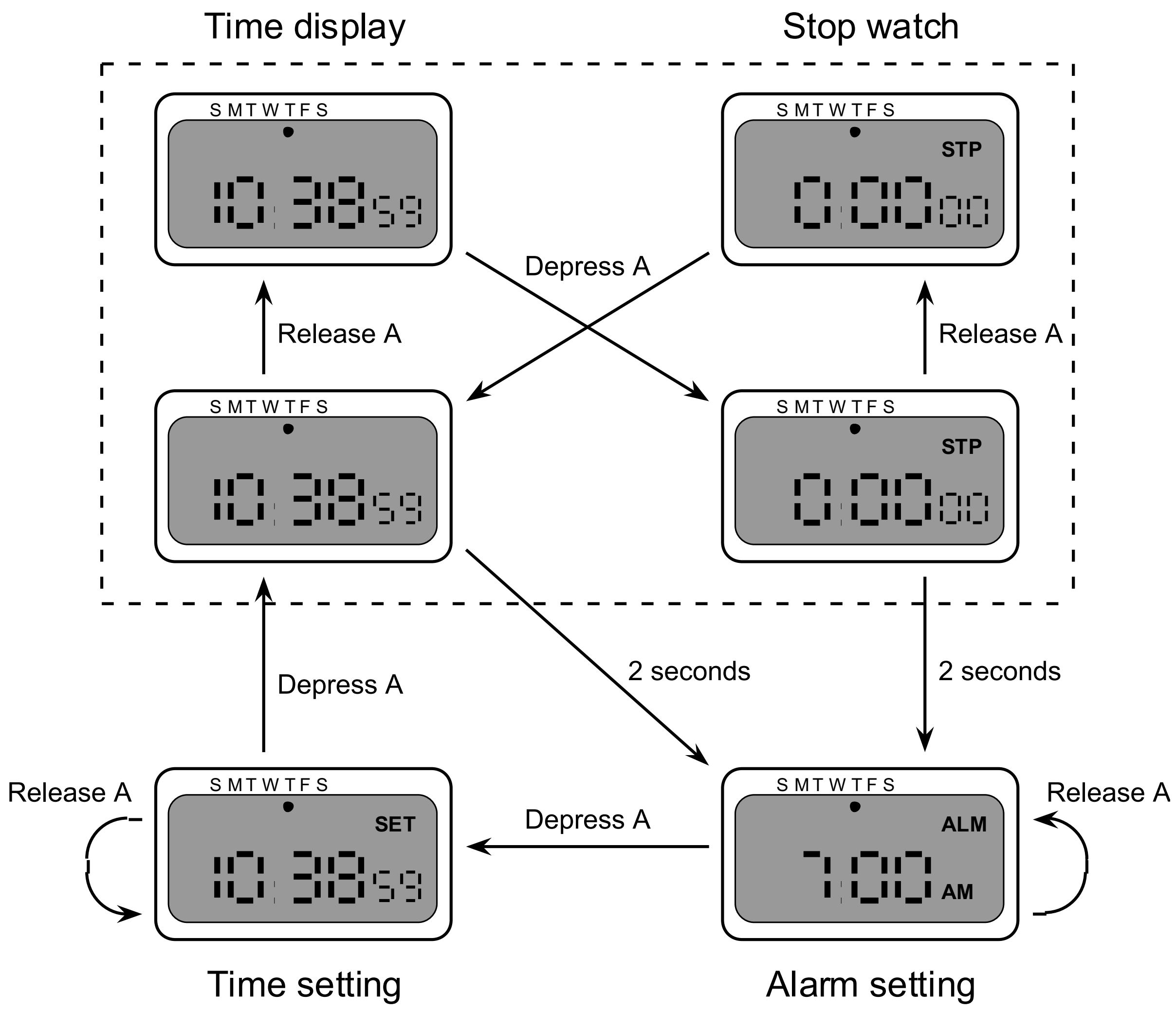
- completeness
  - distinguish depress A and release A
  - what do they do in all modes?



Digital watch – Designers instructions

and ...

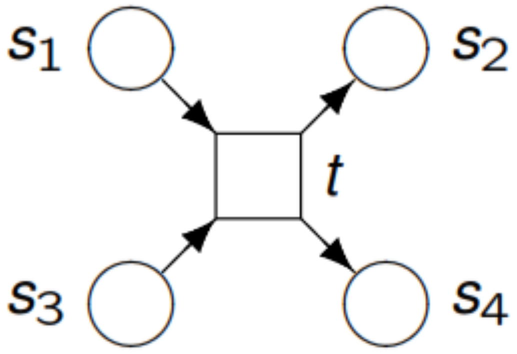
that's just
one button



IE 403

Week 10-Lec 3

Petri nets are a basic model of parallel and distributed systems (named after Carl Adam Petri). The basic idea is to describe state changes in a system with transitions.



Petri nets contain places ○ and transitions □ that may be connected by directed arcs.

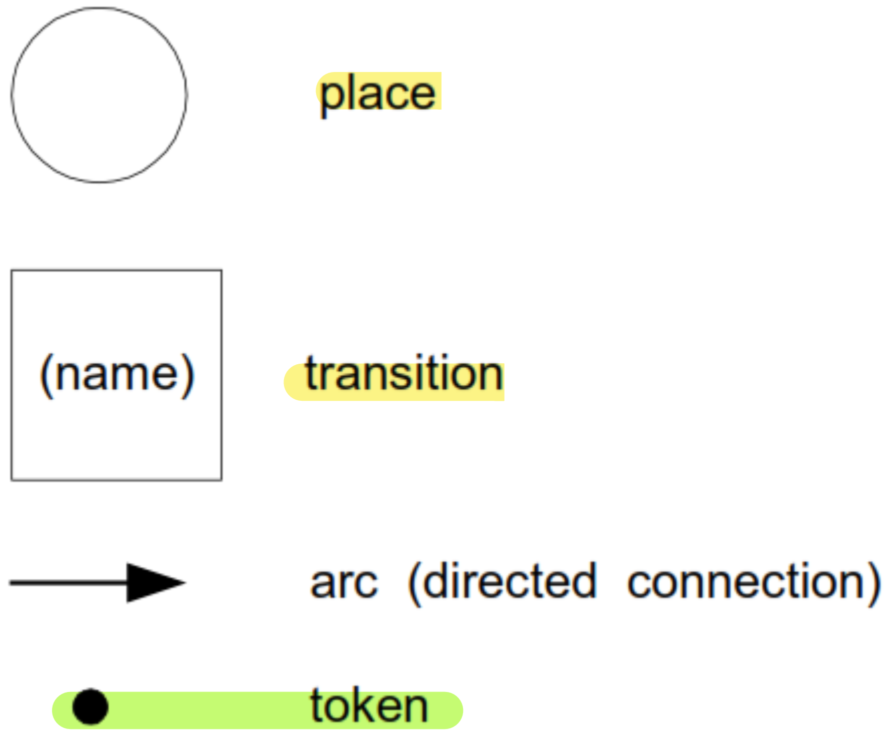
Places symbolise states, conditions, or resources that need to be met/be available before an action can be carried out.

Transitions symbolise actions.

- Place: used to represent passive elements of the reactive system
- Transition: used to represent active elements of the reactive system
- Arc: used to represent causal relations
- Token: elements subject to change

The state (space) of a process/system is modeled by places and tokens and state transitions are modeled by transitions

- A place is represented by a circle
- Transitions are represented by squares/rectangles
- Arc are represented by arrows
- Tokens are represented by small filled circles

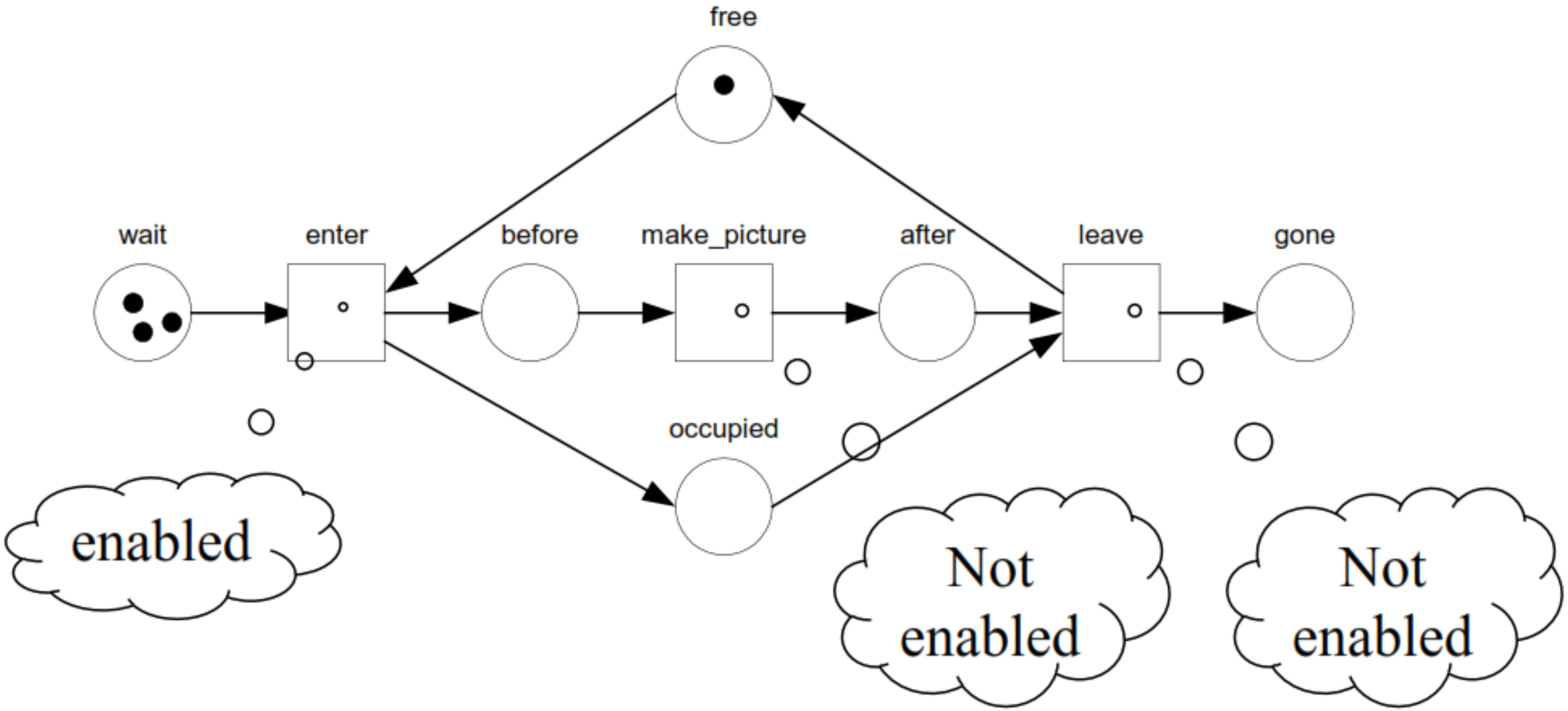


- Tokens can play the following roles
  - A **physical object**, for example a product, a part, a drug, a person
  - An **information object**, for example a message, a signal, a report
  - A **collection of objects**, for example a truck with products, a warehouse with parts, or an address file
  - An **indicator of a state**, for example the indicator of the state in which a process is, or the state of an object
  - An **indicator of a condition**: the presence of a token indicates whether a certain condition is fulfilled

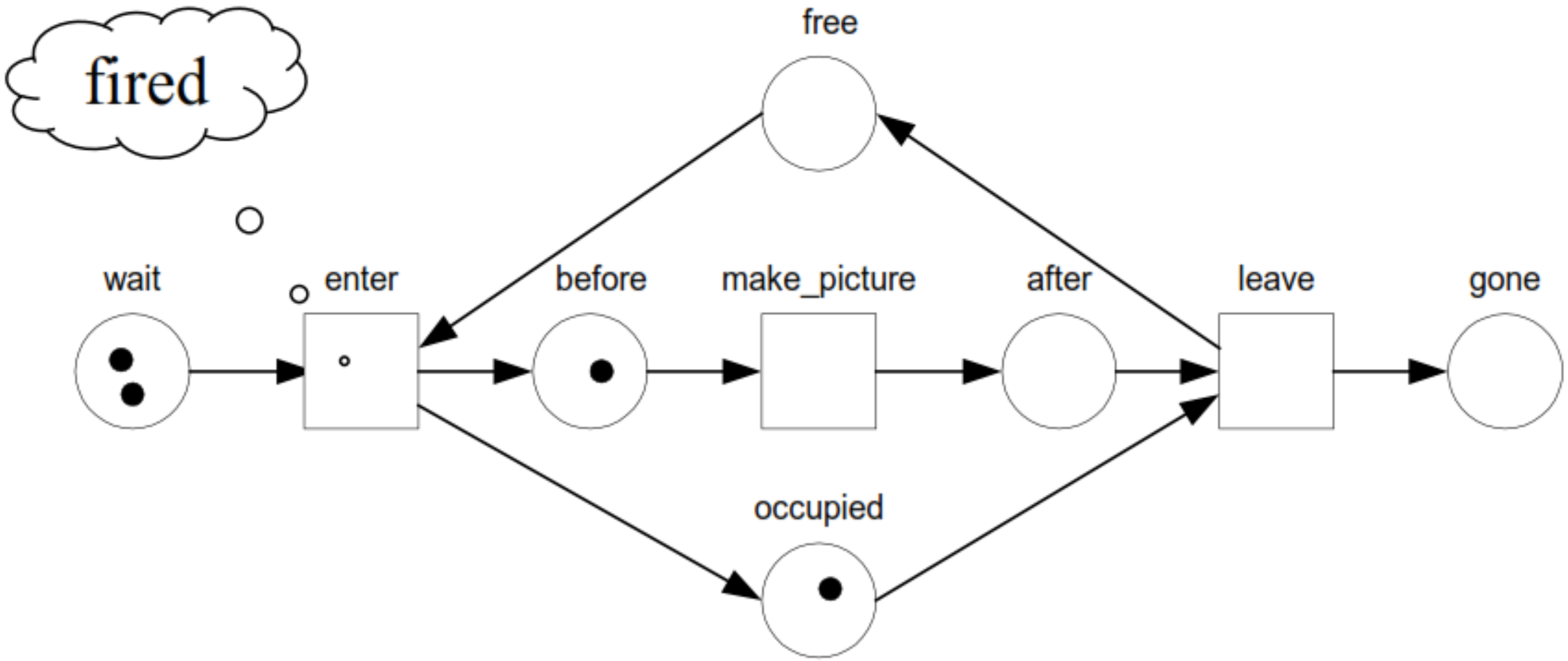
- A place in a PN can represent the following
  - A type of **communication medium**, like a telephone line, a middleman, or a communication network
  - A **buffer**: for example, a depot, a queue or a post bin
  - A **geographical location**, like a place in a warehouse, office or hospital
  - A possible **state or state condition**: for example, the floor where an elevator is, or the condition that a specialist is available

- A transition can be used to represent things such as
  - An **event** (e.g., starting an operation, the switching of a traffic light from red to green)
  - A **transformation of an object**, like adapting a product, updating a database, or updating a document
  - A **transport of an object**: for example, transporting goods, or sending a file

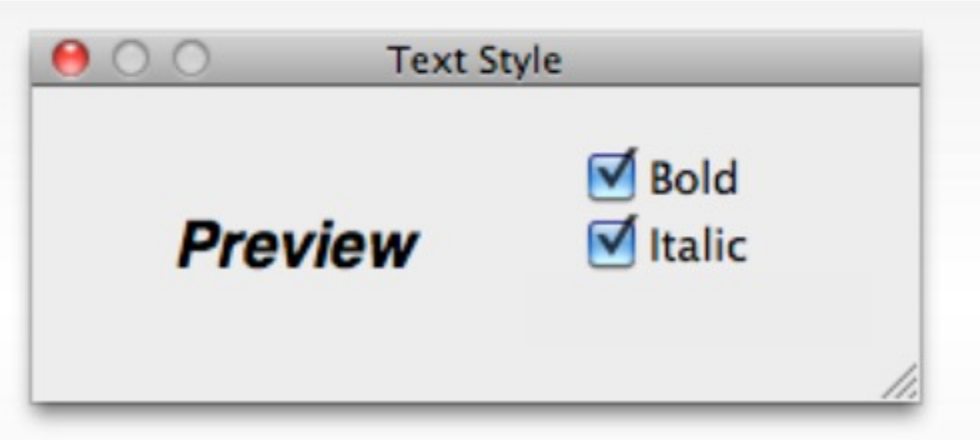
- A transition is **enabled** if each of its input places contains at least one token

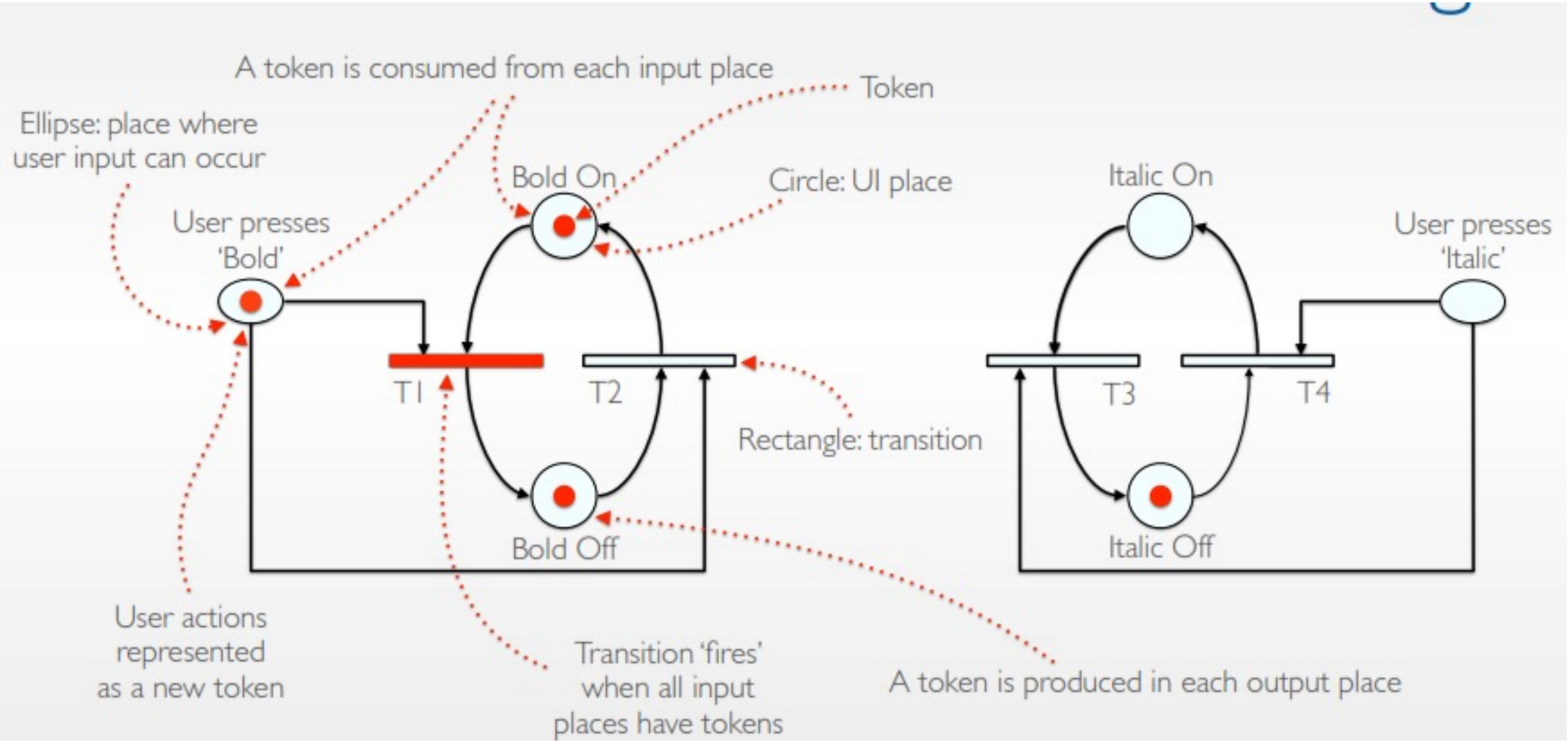


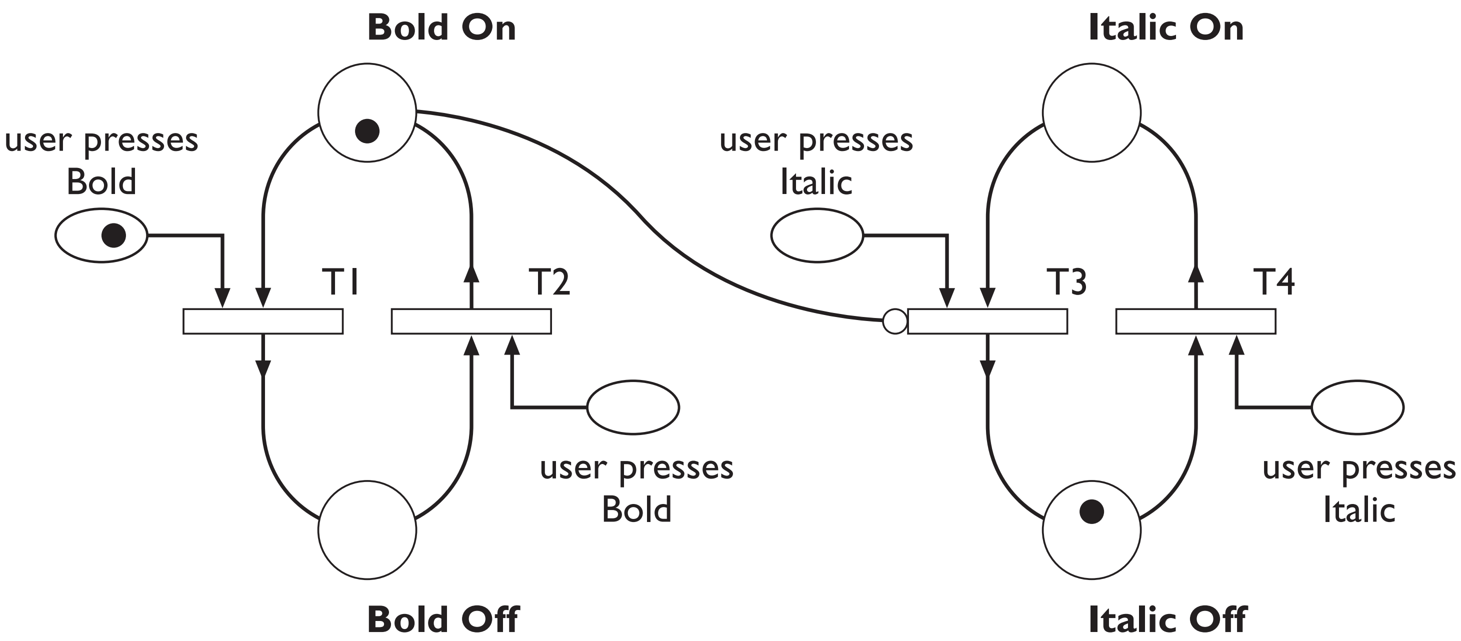
- An **enabled** transition can **fire** (i.e., it occurs)
- When it **fires** it **consumes** a token from each input place and **produces** a token for each output place



Draw the Petri net for our dialog box with concurrent "Bold" and "Italic" options




A token is consumed from each input place
Token
Ellipse: place where user input can occur
Bold On
Circle: UI place
Italic On
User presses 'Bold'
User presses 'Italic'
T1
T2
T3
T4
Rectangle: transition
Bold Off
Italic Off
User actions represented as a new token
Transition 'fires' when all input places have tokens
A token is produced in each output place



This corresponds to a dialog where if the italic toggle is on it cannot be turned off while bold is on.

IE 403/476

Human-Computer Interaction

Week 11-Lec 1

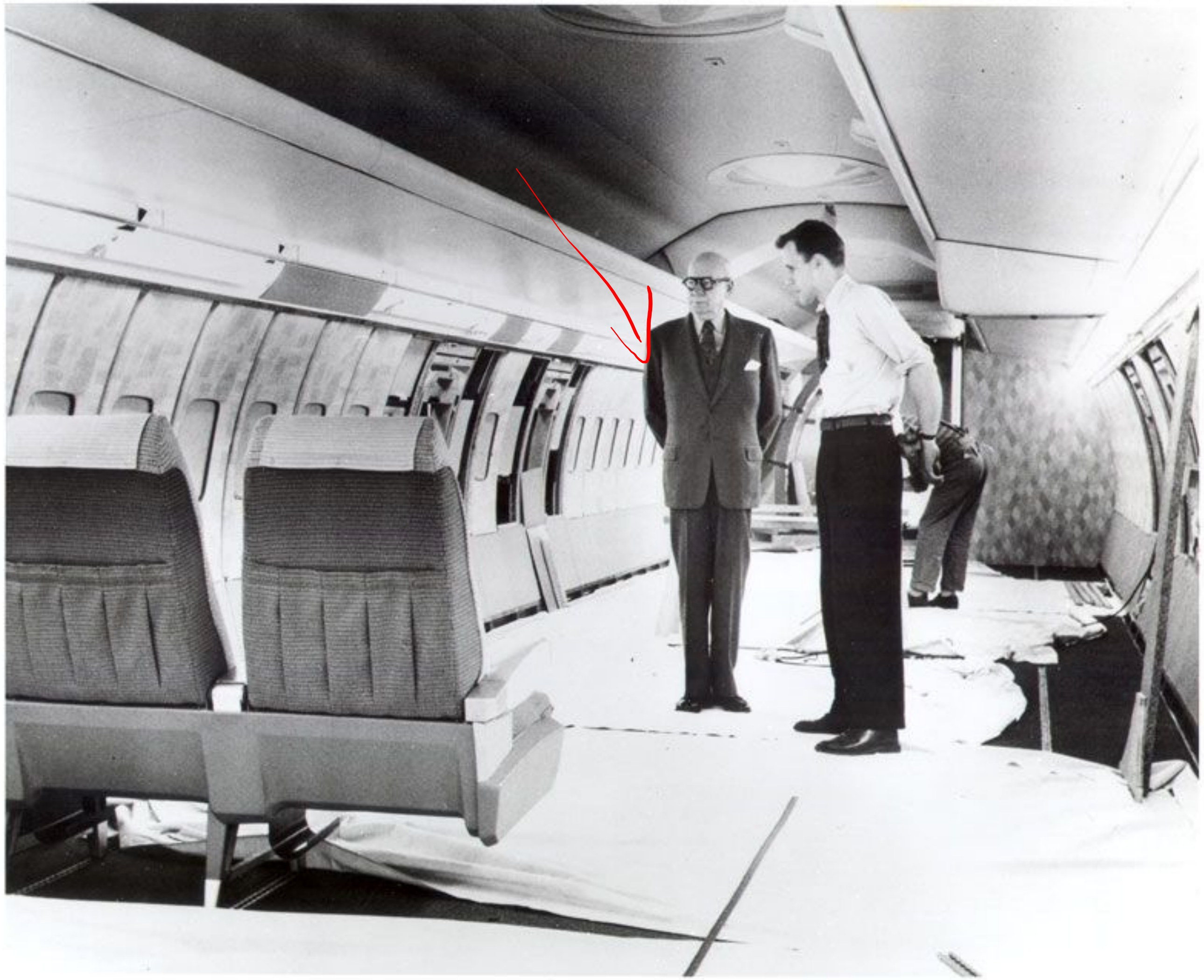
Why prototype?

- Evaluation and feedback are central to interaction design
- Users can see, hold, interact with a prototype more easily than a document or a drawing
- Test out ideas for yourself
- It encourages reflection: important aspect of design
- Prototypes answer questions, and support designers in choosing

between alternatives

Prototypes Can be BIG

- Walter dor win Teague – Industrial Designer
- Design seats for BOEING 707



Prototypes of Mouse

IDEO for Microsoft

Prototype/Mock of iPod

[Prototypes of Expensive cars]

What do Prototypes PROTOTYPE?

FEEL: What might it look like?

IMPLEMENTATION: How would it work?

ROLE: What might the experience be like?

What do prototypes tell us?

- Interest: would people want this?
- Use: does this solve a real problem?
- Usability: is this easy, sensible, and effective?
- Implementation: can this be done?

Focus on the GOALS

Evolve the DESIGN

Prototypes Vs Mock Ups Vs Wireframes

- Mock-ups and wireframes are static,
    - representing a single state of the design
- Prototypes are simulations/models of how
    - the design is supposed to actual work

- Prototypes are usually intended to be shown to the end user

- Wireframes are usually part of a design document to go from design to actual system
    - Usually contain annotations specific to the design team and are not intended for end-user consumption
- Wireframes *can be used as a lo-fidelity prototype to save time*
    - Remove annotations, make it interactive

1. For Q1 release, music search only
2. Related artists determined by user purchasing data mining
3. Album art to be approved by legal

Prototype Fidelity (level of similarity)

- Low fidelity: omits details
- High fidelity: more like finished product

- SANTA CLARA, California -- People thought Jeff Hawkins was crazy when they saw him taking notes, checking appointments, and synchronizing a small block of wood with his PC, pretending all the while that the block was a handheld computer.
- "If I wanted to check the calendar I'd take it out and press the wooden button"

Jeff believed we had to make the product considerably smaller then current PDAs. He carved up a piece of wood in his garage and said this is the size he wanted. He'd walk around with this block in his pocket to feel what it was like. I would print up some screenshots as we were developing UI, and he'd hold it and pretend he was entering things, and people thought he was weird. He'd be in a meeting furiously scribbling on this mockup, and people would say, "Uh, Jeff, that's a piece of wood."

Source: "The Philosophy of the Handheld." Wired Magazine, October 1999.

Fidelity is multidimensional

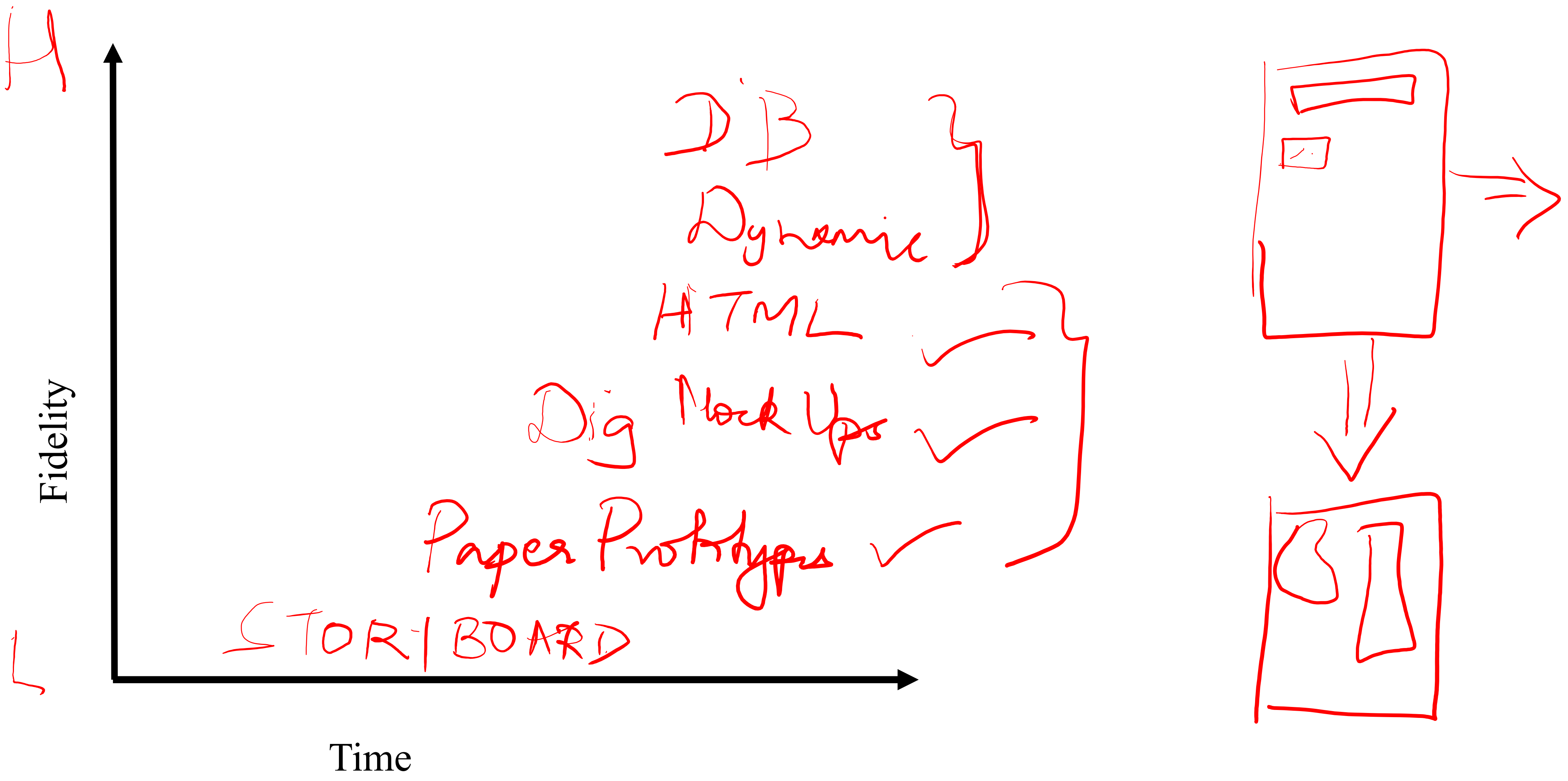
Deep or vertical prototyping

- provide a lot of detail for only a few functions

Broad or horizontal prototyping

- provide a wide range of functions, but with little detail

- Look: appearance, graphic design
  - Sketchy, hand-drawn
- Feel: input method
  - Pointing & writing feels very different from mouse & keyboard


H
DB
Dynamic
HTML
Dig Mock Ups
Paper Prototypes
STORYBOARD
L
Fidelity
Time

Low Fidelity : Not always paper

iPod prototype was a wooded one

From book Jony Ive

FORM FACTOR

Image courtesy of Benjamin Chan on Flickr.

https://www.theverge.com/2021/10/25/22744761/apple-ipod-prototype-original-design

http://www.designinginteractions.com/interviews/JeffHawkins

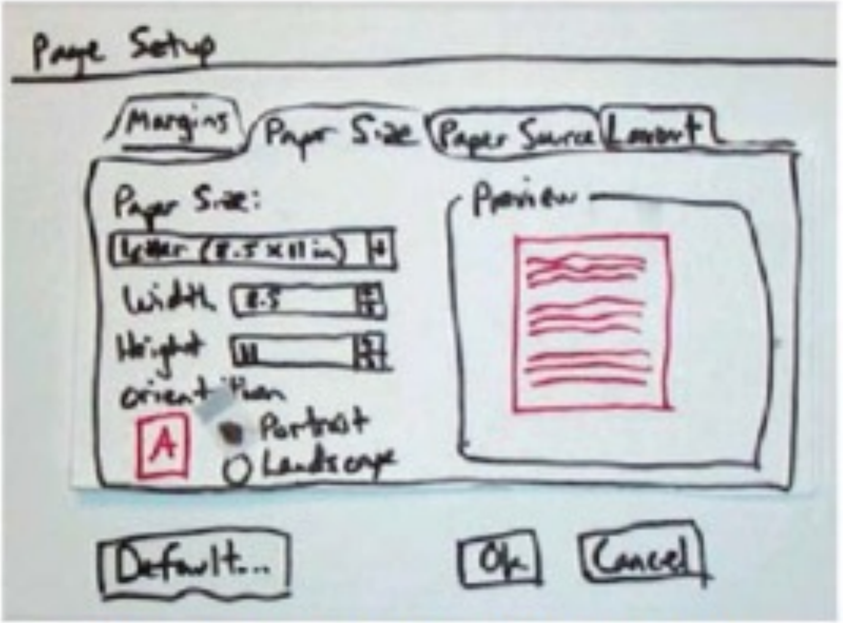
Prototyping Techniques

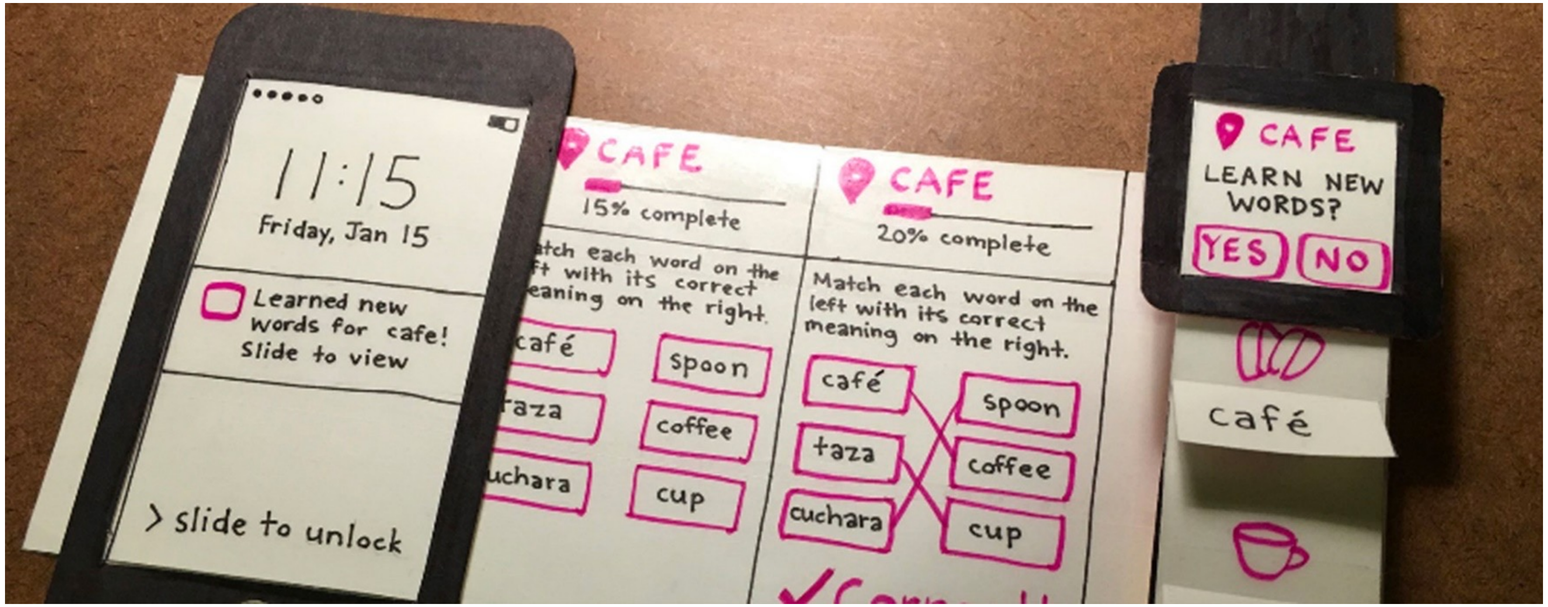
- Storyboard
  - Sequence of painted screenshots
  - Sometimes connected by hyperlinks ("hotspots")
- Form builder
  - Real windows assembled from a palette of widgets (buttons, text fields, labels, etc.)
- Wizard of Oz
  - Computer frontend, human backend

Adobe xd
Figma
Framer

Affinity Diagram

Paper prototypes





Paper prototype

- Interactive paper mockup
  - Sketches of screen appearance
  - Paper pieces show windows, menus, dialog boxes

- Interaction is natural
  - Pointing with a finger = mouse click
  - Writing = typing

Low fidelity in look & feel

High fidelity in depth (person simulates the backend)

- A person simulates the computer's operation
  - Putting down & picking up pieces
  - Writing responses on the "screen"
  - Describing effects that are hard to show on paper

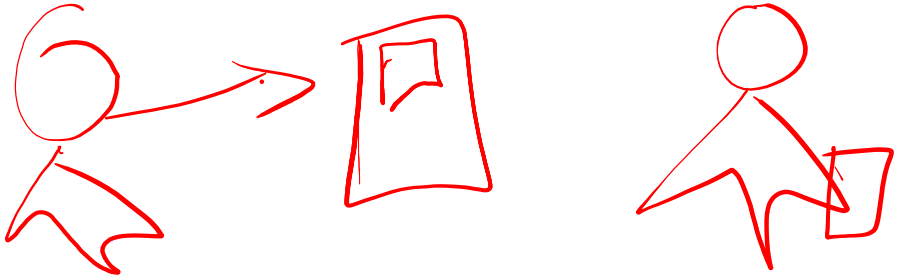
Why paper prototyping?

- Faster to build
  - Sketching is faster than programming

- Easier to change
  - Easy to make changes between user tests, or even *during* a user test
  - No code investment– everything will be thrown away (except the design)

- Focuses attention on big picture
  - Designer doesn't waste time on details
  - Customer makes more creative suggestions,

- Nonprogrammers can help
  - Only kindergarten skills are required



What can we learn from Paper prototype?

- Conceptual model
  - Do users understand it?
- Navigation & task flow
  - Can users find their way around?
  - Are information preconditions met?
- Terminology
  - Do users understand labels?
- Functionality
  - Does it do what's needed? Missing features?
- Screen contents
  - What needs to go on the screen?

[Example of paper prototype: https://www.hashcut.com/v/xjkyCdp](https://www.hashcut.com/v/xjkyCdp)

Design Challenges/Issues and How Paper prototyping helps

- Device constraints
- Time to market
- Multiple stakeholders
- Proving concepts early
- Technology led Not User Centered

What we cannot learn?

- Look: color, font, whitespace, etc
- Feel: efficiency issues
- Response time
- Are small changes noticed?
  - Even the tiniest change to a paper prototype is clearly visible to user
- Exploration vs. deliberation
  - Users are more deliberate with a paper prototype; they don’t explore or thrash as much

Computer prototypes

- Interactive software simulation
- High-fidelity in look & feel
- Low-fidelity in depth
  - Paper prototype had a human simulating the backend; computer prototype doesn't
  - Computer prototype may be **horizontal**: covers most features, but no backend

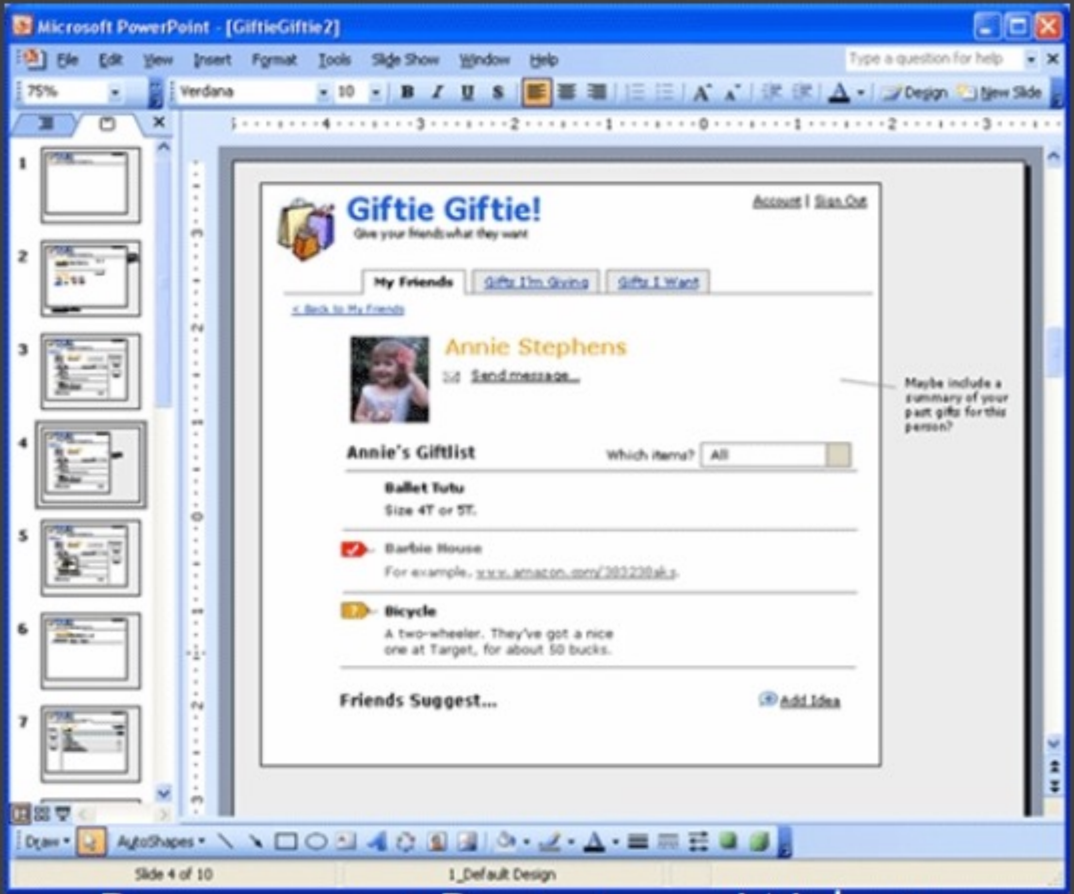
To simulate the functionality of the interface during user testing, a human can act as the "computer" or "backend" by manually changing the paper screens and providing responses to user inputs. This allows designers to test the flow and usability of the interface without having to build a functional prototype.

- Screen layout
  - Is it clear, overwhelming, distracting, complicated?
  - Can users find important elements?
- Colors, fonts, icons, other elements
- Interactive feedback
  - Do users notice & respond to status bar messages, cursor changes, other feedback
- Efficiency issues
  - Controls big enough? Too close together? Scrolling list is too long?

Experience Prototyping

Used inkblots while running to get an idea of how big the buttons needed to be

Powerpoint Prototyping



Powerpoint Prototype Website

What does a prototype look like?

- a series of screen designs (e.g., from photoshop)
- a storyboard, i.e. a cartoon-like series of scenes
- a PowerPoint slide show or HTML pages
- a video simulating the use of a system
- a lump of wood or foam (e.g. to represent a physical design)
- a cardboard mock-up
- a piece of software with limited functionality written in the target language or in another language

Conclusion

Prototypes

ARE QUESTIONS

ASK , EXPLORE AND REFINE

DON’T NEED TO BE COMPLETE

Some resources

- http://grouplab.cpsc.ucalgary.ca/saul/681/1998/prototyping/survey.html
- https://www.youtube.com/watch?v=OlbdIXLunt4

Rapid Prototyping Exercise

- Ordering a PIZZA from a mobile app
- Choices of pizza, Toppings
- Different prices
- Payment types
- In groups of 3, create a paper prototype

15 mins

Part II: The Design Phase

1. Identify User Goals

- Checking the latest wallpapers that could be downloaded
- Downloading the chosen wallpaper
- Changing the wallpaper setting of the phone to the new downloaded wallpaper
- Checking the team news headlines
- Checking an individual news story
- Checking the team’s upcoming fixtures
- Checking the team’s position and scores in the league table

2. Identify features and content

Operator company (client focussed) aims to:

- Enable personalisation
- Encourage customer usage of online content

Content and features

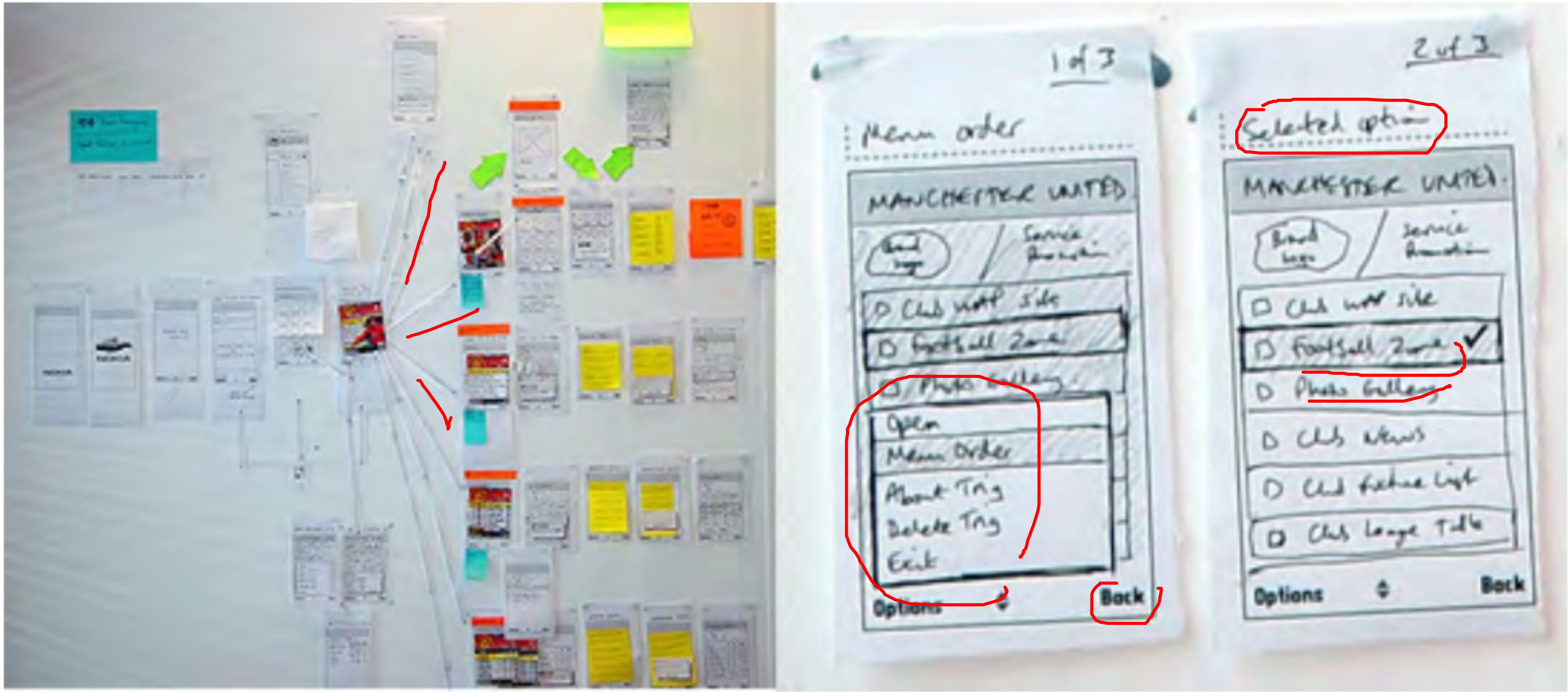
- Manchester United wallpaper images
- Manchester United fixtures information
- Manchester United news stories
- Manchester United league tables updates

User Experience considerations (users focussed)

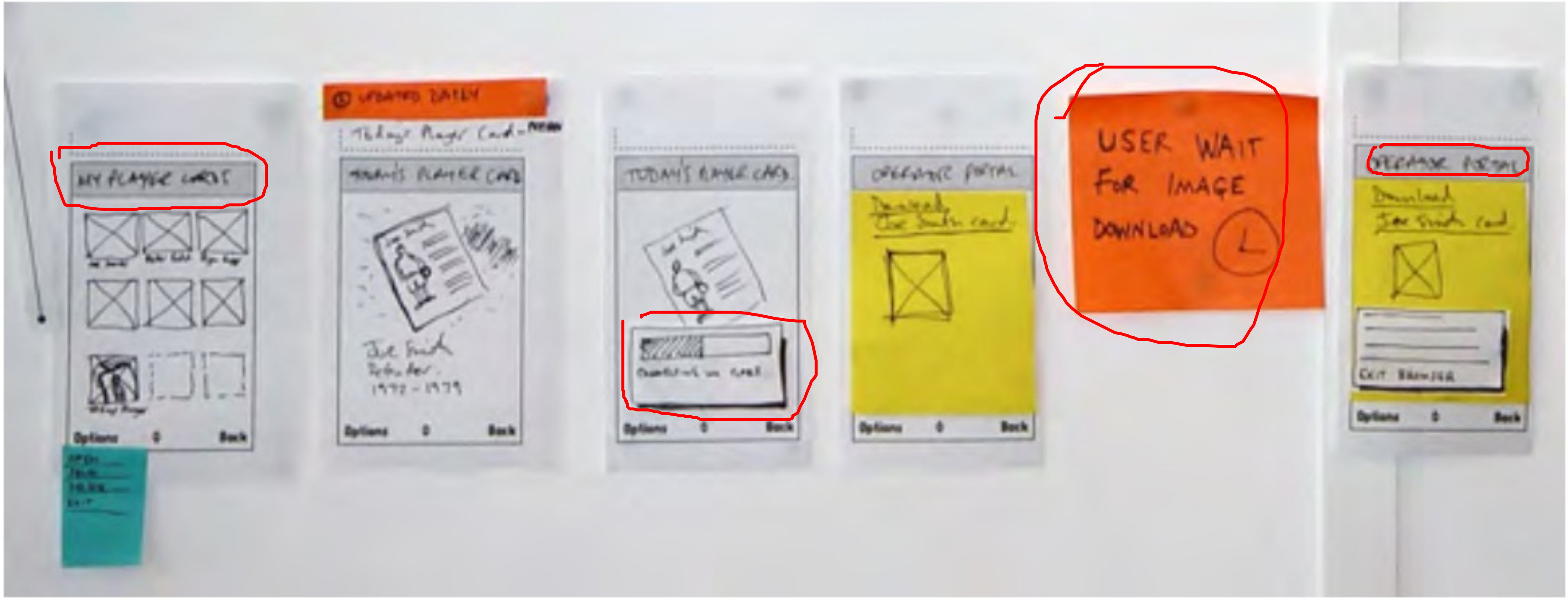
- The download and purchase process
- The installation process for the theme
- The switching between multiple themes
- The navigation of the theme and its contents

Part 3: First design Iteration

3.1 Macro Vs Micro Views



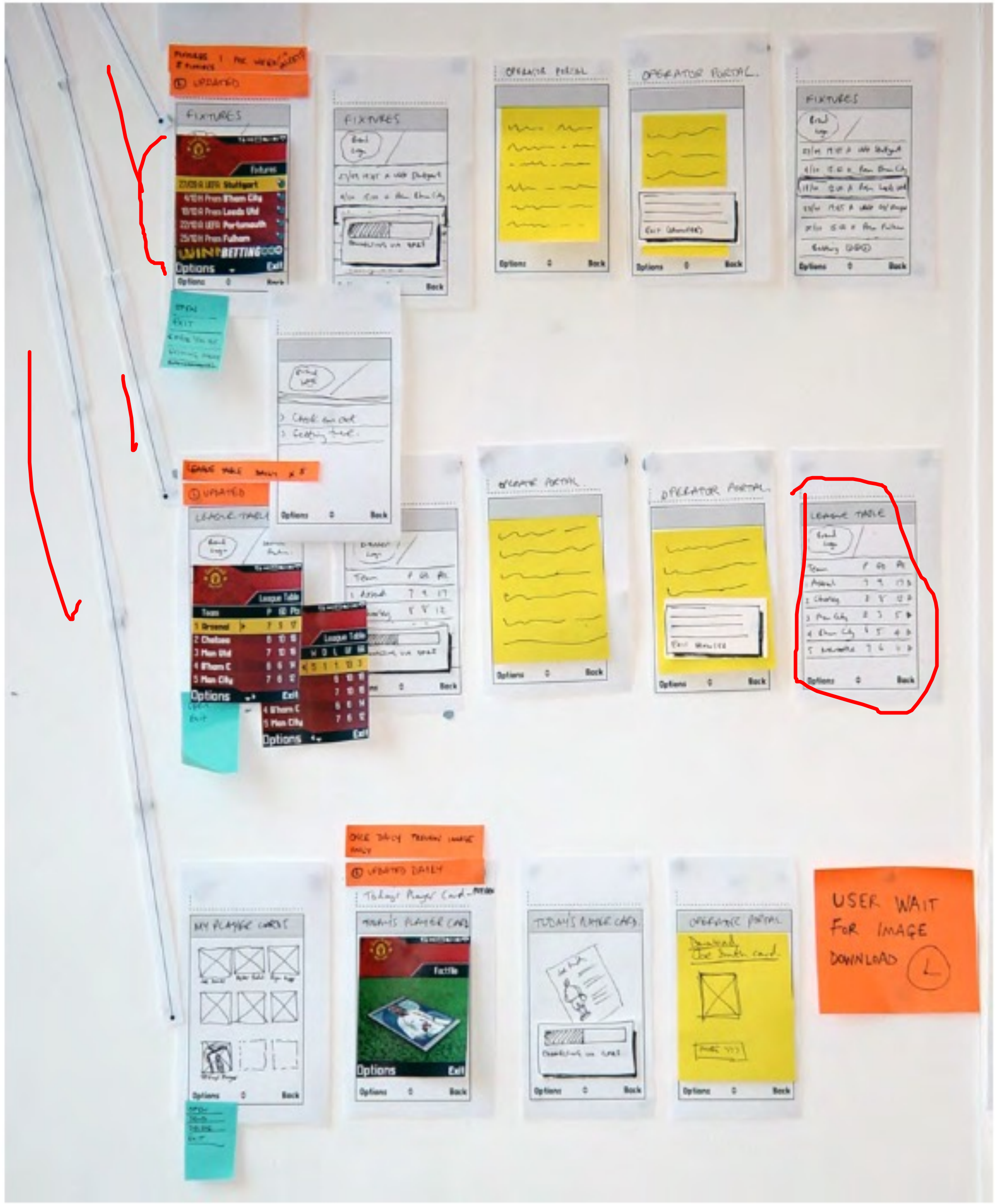
3.2 Screen by Screen flows



3.3 Section and Contents

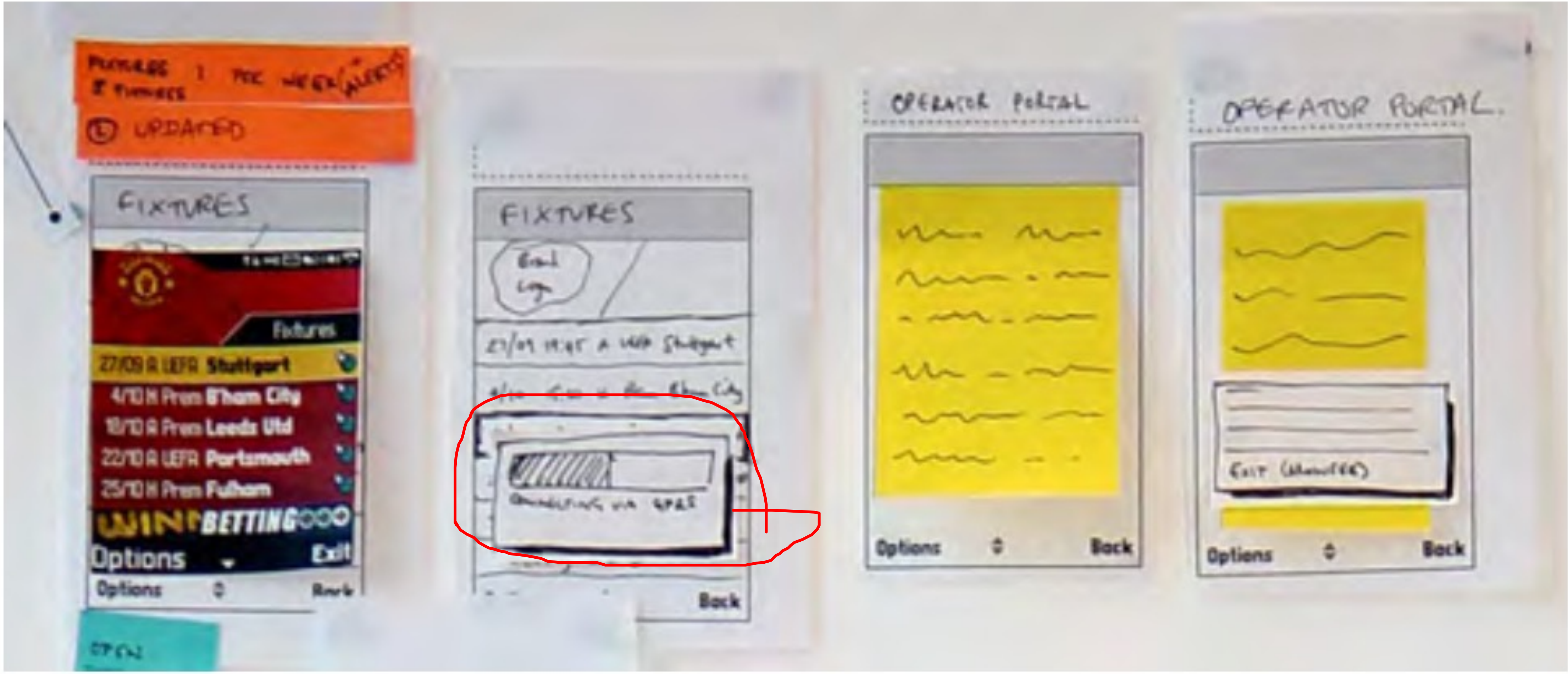
What would users expect when they view the wallpapers for their team?

Do they want to see 20 on the screen at once or just 9?

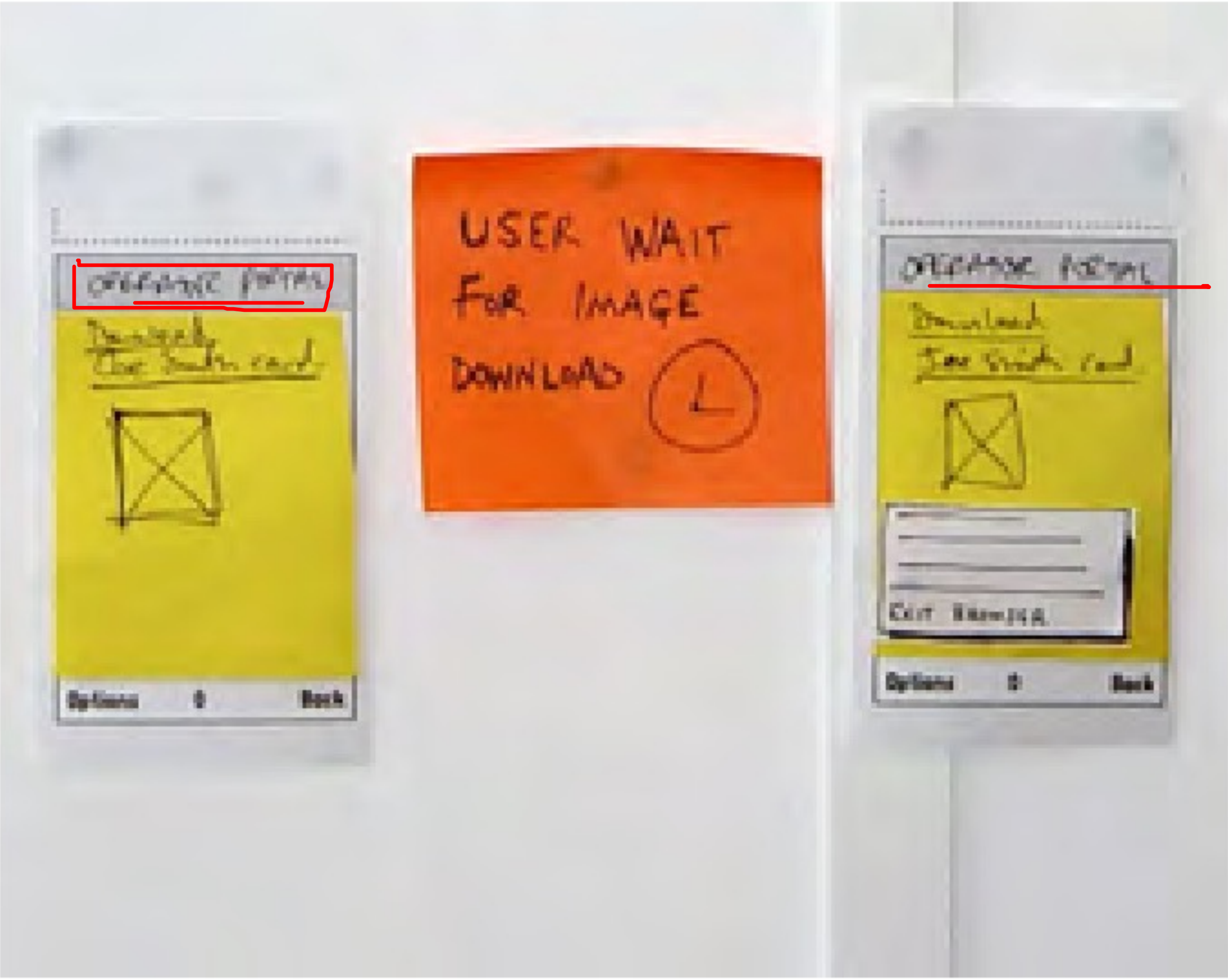


3.4 Design Alternatives

3.5 Mobile Network Issues



3.6 User Wait times & Feedback



3.7 Navigation Consistency

- Labels and words—used in menus and onscreen throughout.
- Soft keys—map to the onscreen word labels via the two hard keys left and right.
- Options Menus—most mobile phone UI's offer the user a set of options, accessed at each screen and relevant to the context of that screen.
- Use of a 'Back' key: how do I get back a screen?

3.8 Visual Design