

EE4213 Human-Computer Interaction

Week 5 Computer and Interaction (Part 2)

Today's Menu

- Output devices
- Interaction Models
- Interface Types
- An introduction to design thinking
- Test paper will be available on CANVAS at noon

Output Devices: Computer Screen

- LCD/LED/Plasma/OLED/QLED

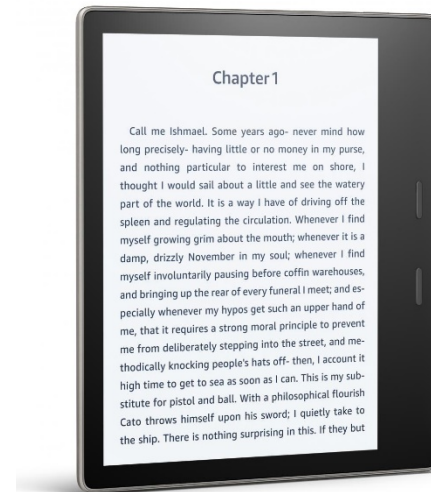


- 3D perception can be possible with or without specific eyewear



Output Devices: Electronic ink (e-ink)

- Display technology designed to mimic the appearance of ordinary ink on paper used in e-readers
- Paper-like readability (300 pixels per inch)
- Support writing
- Used in retail/signages



<http://www.eink.com/>

Output Devices: Speech Output

- Digitized speech - recorded **human** speech directly in digital format
- Synthesized speech - generated using **computers** via speech processing techniques
- Current technology supports numerous spoken language output, e.g., English, Japanese, Mandarin
- Text-to-speech function is readily available

<https://cloud.google.com/text-to-speech/docs>

Output Devices: Speech Output

Advantage:

- Useful when user's eyes are busy or user cannot access the screen

Disadvantages:

- Limited information because speech is transient
- Lack of privacy & security under open environment
- Speech is a single channel mode, i.e., we cannot listen multiple messages simultaneously
- Spoken rate (~120-180 words/min.) is slower than reading (~200-300 words/min.)

Output Devices: Speech Output

Some guidelines for use of speech output

- Consider as output device when user's eyes are busy or user cannot access the screen
- Avoid as output device in open environment, when privacy & security are important, when frequency of use is high, when multiple messages must be sent simultaneously, when human memory is overloaded
- Use output rate of approximately 180 words per min.

Output Devices: Sound

Audio used as feedback, localization and entertainment, e.g.,

- Confirm actions
- Offer warning
- Provide mood context in games
- Simulate various musical instruments
- Provide directional guidance for visually impaired persons using electronic audible signals

Output Devices: Braille Display

- Pins provide output for the blind



- Good for detailed editing (e.g. programming, final editing of papers) for the blind with advanced Braille skills
- Braille printers provide “hard copy” output for blind users.

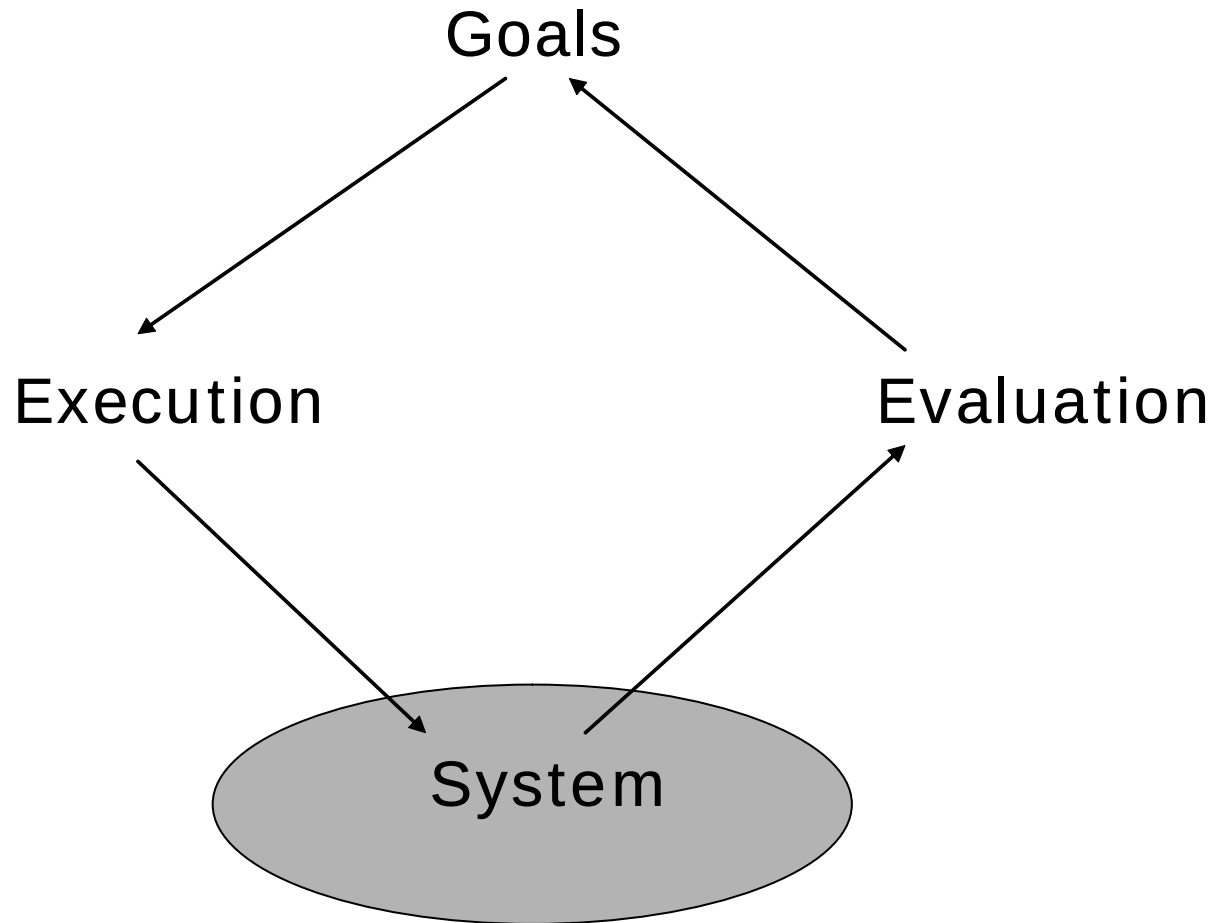
Interaction Model

- **Interaction**: communication between user & system
- Why study interaction models?
- Help us to understand what is going on in the interaction between user & system
- Help us to identify problematic areas within the design
- Address differences, if any, between what user wants & system does

Interaction Model

Norman's interaction framework

2 major phases: **execution** & **evaluation**

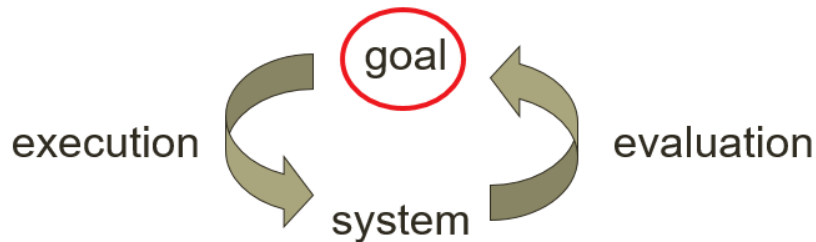


Interaction Model

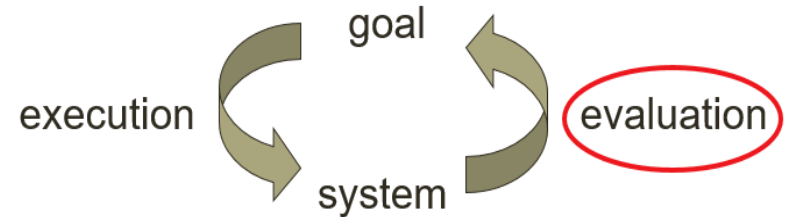
Can be subdivided into further stages:

1. Establish the goal (Desired output from a performed task)
 2. Form intention (Specific action required to meet the goal)
 3. Specify action sequence at interface
 4. Execute action
 5. Perceive system state
 6. Interpret system state
 7. Evaluate the outcome
-
- Concentrate on the user view of the interface
 - Can be considered as execution/evaluation loop

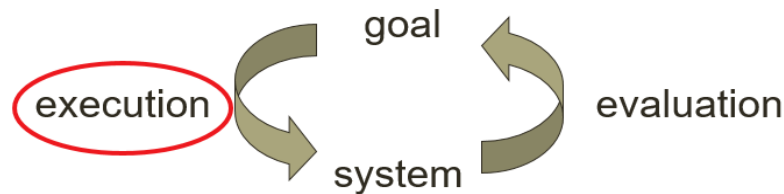
Interaction Model



- user establishes the goal
- formulates intention
- specifies actions at interface
- executes action
- perceives system state
- interprets system state
- evaluates system state with respect to goal



- user establishes the goal
- formulates intention
- specifies actions at interface
- executes action
- perceives system state
- interprets system state
- evaluates system state with respect to goal



- user establishes the goal
- formulates intention
- specifies actions at interface
- executes action
- perceives system state
- interprets system state
- evaluates system state with respect to goal

Interaction Model

You are reading a book, and evening is coming:

- Establish goal: Need more light
- Form intention: Switch on desk lamp
- Specify action sequence: Press lamp switch
- Execute action
- Perceive system state: More light now
- Interpret system state: Lamp is switched on
- Evaluate outcome: Enough light and goal completed

Interaction Model

Establish goal	Move a sentence from one paragraph to another paragraph in a letter
Form intention	Use WORD to carry out the movement of a piece of text from one paragraph to another
Specify action sequence	(1) Highlight the text by moving & clicking a mouse (2) Click the cut button, (3) Move the cursor to new position and (4) Click the paste button
Execute action	Execute the actions: An error scenario - the user highlights the text correct but clicks on the copy button instead of the cut button then completes the action
Perceive result	User sees the text in correct position but notices that it is still in the original position as well
Interpret result	User realizes, he or she used the copy rather than cut
Evaluate outcome	User knows that the goal has not been completed & it is necessary to delete the text in the original position, this lead to a new goal & set of actions

Using Norman's Interaction model

Explain why some systems are harder to use than others

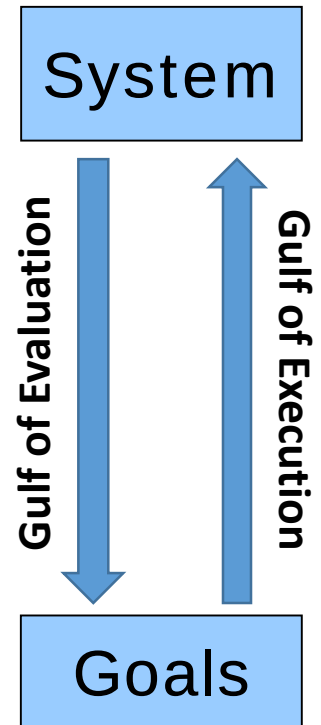
Gulf of Execution

user's formulation of actions
≠ actions allowed by the system

The difference (gap) between the user's perceived execution actions and the required actions is the gulf of execution.

Gulf of Evaluation

user's expectation of changed system state
≠ actual presentation of this state



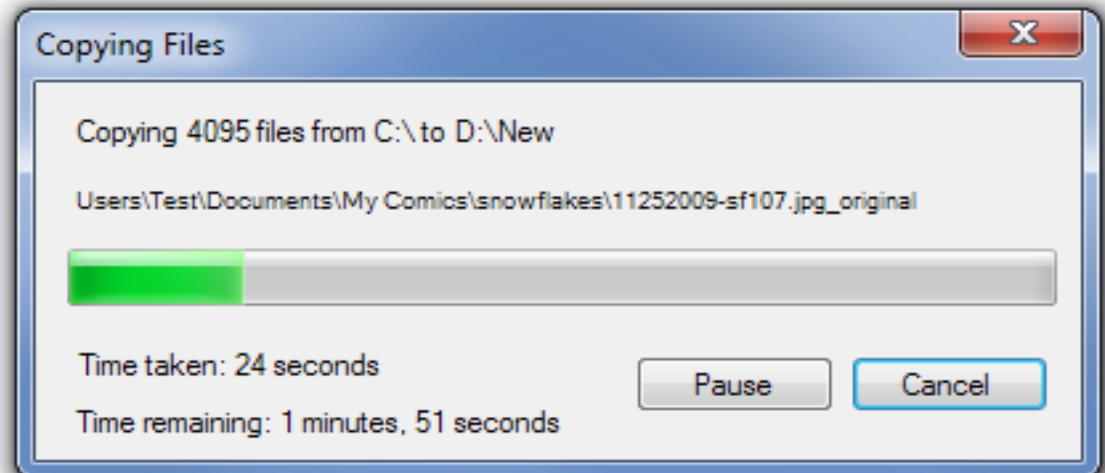
Gulfs of **execution**

- Mismatch between user's intention & allowable/required actions, e.g.,
 - The system does not support the user's goal
 - Sequence of actions is invalid
- Does the interface allow us to carry out the actions required by the intention? E.g.,
 - **Goal** is to save a file
 - **Intention** is to use the file menu
 - **Action** is to click the save option
 - Is there a save option in the file menu?

Gulfs of **evaluation**

- Mismatch between system's presentation & user's expectation, e.g.,
 - Failure to notice current system status
 - Poor feedback from system
- Given a particular interface design, how easily can you:
 - Determine the function of the device?
 - Determine whether the system is in the desired state?
 - Determine what actions are possible?
 - Perform the valid action?

Gulf of Evaluation Example: Copying a large number of files



Human error - slips and mistakes

Slip (careless, due to the change of context)

- understand system and goal
- correct formulation of action
- incorrect action

Mistake (incorrect conceptual model)

- may not even have right goal

How to fix?

slip – better interface design (better tolerance for slip)

mistake – better understanding of system

User failures can occur:

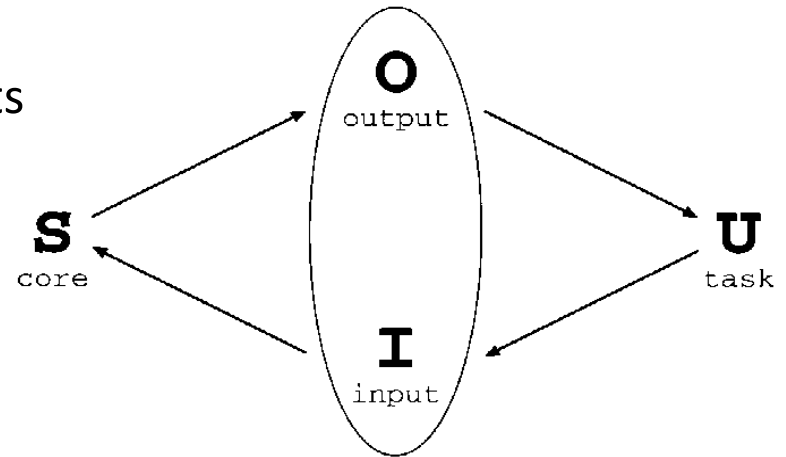
- Users form an inadequate goal
- Might not find the correct interface object because of an incomprehensible label or icon
- May not know how to specify or execute a desired action
- May receive inappropriate or misleading feedback

Gaps can be reduced by:

- Good **conceptual model** – Agree with user's intention
- Good **visibility** – State and action alternatives should be visible
- Good **mapping**
- Clear affordance
- **Feedback** – User should receive continuous feedback

Abowd and Beale framework

- extension of Norman's model
- their interaction framework has 4 parts
 - User (U)
 - Input (I)
 - System (S)
 - Output (O)

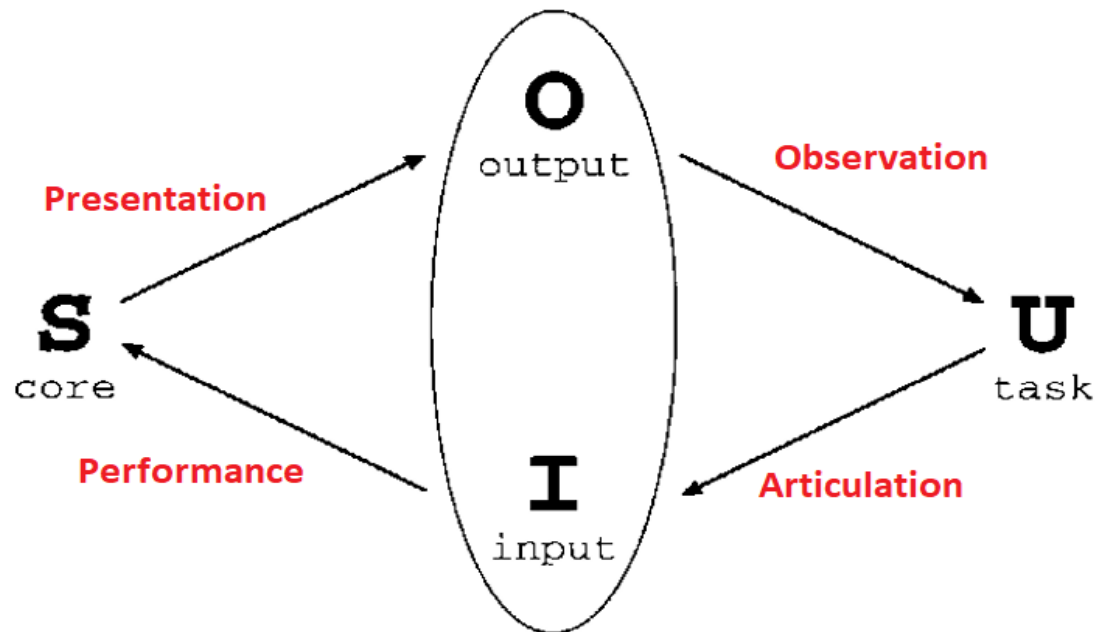


Steps involved

- User intentions translated into actions at the interface
- Translated into alternations of system state
- Which in turn are reflected in the output display
- Which is then interpreted by user

Interaction Model

- Input and output together form the interface
- Each component has its own unique language: problems occur when language translation between two adjacent components is in error



Four Translations between Components

- Articulation – User begins the interactive cycle with the formulation of a goal and a task to achieve that goal. The user articulates the task in input language (translated from task language)
- Performance – Interface translates input language into core language as operations to be performed by system. System is in a new state
- Presentation – System presents the new state (results) in terms of output language
- Observation – User observes output and assess the results of the interaction relative to the original goal. The user translates the output language into personal understanding

Example of translation problem

Each component has its own unique language: problems occur when language translation between two adjacent components is in error

e.g. user is not sure if the DVD recorder is set to record properly. **Maybe:**

- From user to interface, the user has pressed keys on remote control in the wrong order (articulation)
- From interface to system, DVD recorder can record on any channel but remote control cannot select channels (performance)
- From system to interface, DVD recorder display panel does not indicate that the program has been set (presentation)
- From interface to user, user does not interpret the indication properly (observation)

Assessment can be done at each translation

- Articulation
 - Judged in terms of coverage from tasks to input and relative ease with which translation is accomplished
 - Larger the gulf of execution, the more difficulty user will face in translating intentions to actions
 - Conversely, if articulation is easy, gulf of execution will be small
 - Example: GUIs vs. Command-line interfaces for file operations
- Performance
 - Assessing whether the translated input reach as many system states as possible
 - A system design issue: implementation cost
 - Example: immersive VR vs. command-line

Assessment can be done at each translation

- Presentation
 - An interface design issue: interface must be sufficiently expressive to show all relevant system changes
 - Important in minimizing the gulf of evaluation
- Observation
 - Assessing how easy and complete can the system state be observed by the user
 - Problems: incompleteness of the presentation and misinterpretation by user (e.g. HTML editing without seeing browser output, determining the time accurately on an unmarked analog clock)
 - Examples: stove controls, progress meters



Interface Types

- **Command line interface**
 - Can be:
 - Single characters
 - Short abbreviations
 - Whole words
 - Combination of them
 - Typical examples: Unix system & DOS

```
sable.soc.staffs.ac.uk> javac HelloWorldApp
javac: invalid argument: HelloWorldApp
use: javac [-g][-O][-classpath path][-d dir] file.java...
sable.soc.staffs.ac.uk> javac HelloWorldApp.java
sable.soc.staffs.ac.uk> java HelloWorldApp
Hello world!!
sable.soc.staffs.ac.uk>
```

- Suitable for repetitive tasks
- Better for expert users than novices

Interface Types: Command Line

- Oldest and original user interaction style
- Powerful, offering immediate access to system functions. It is also flexible and able to incorporate options or parameters to vary its behaviour
- It requires the user to press a function key or type a command into a designated entry area on a screen
- One problem is that they must be remembered and they test one's power of recall. No clues about what commands are available exist on the screen
- Another problem is that command lines can be cryptic and obscure with complex syntax. They are also very prone to, and intolerant of, typing errors that can lead to user frustration

Interface Types: Menus

- **Set of options** displayed on the screen
- Options visible: rely on recognition rather than recall
- Selected by using mouse, numeric, cursor, function or alphabetic keys
- Menu systems can be purely text based, with options presented as numbered choices
- Menu can have graphical component, with menu appearing in box, & choices are made either by typing initial letter, or moving around with arrow keys or mouse

Interface Types: Form fill-in

- Screen like paper form
- A number of fields in which user is expected to type in data
- Each field has a label (caption) that indicates the type of data to be entered in that field
- **Spreadsheet** is a variant of form filling

Improved:

STUDENT REGISTRATION					
STUDENT					
Name:		*Major:		Year:	
ID:		Track:		Status:	
COURSES					
Number	*Title	Section	Sequence		

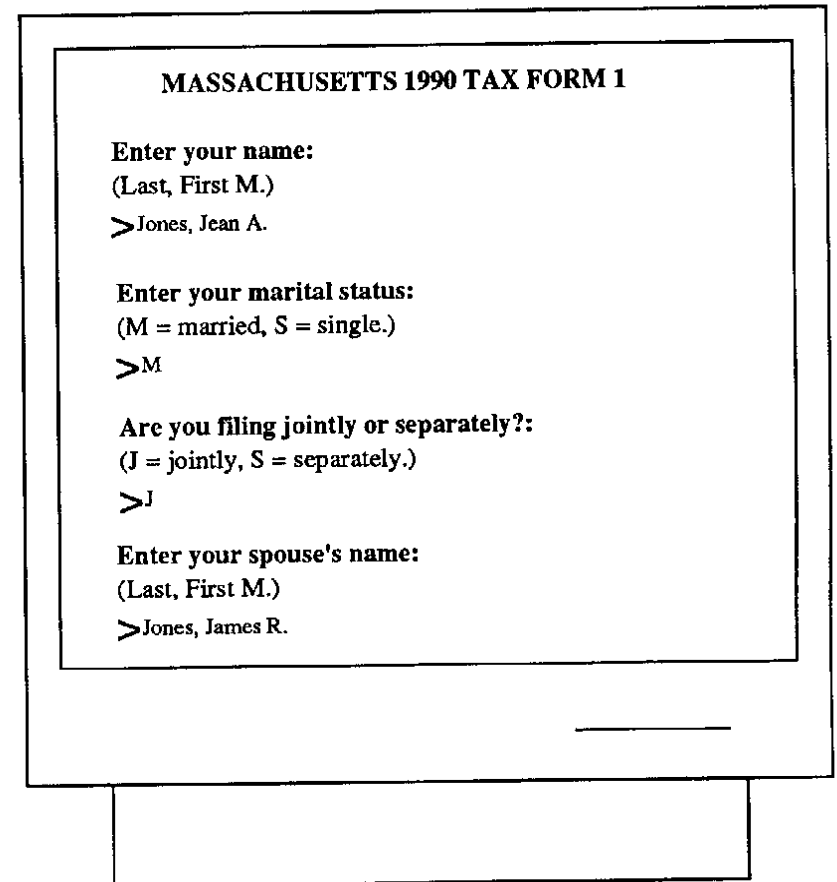
Figure 5-5 Fill-in form caption and field design.

Interface Types: Form fill-in

- Very useful for collecting information
- Today's typical on-screen form contains a series of fields for the user to type information or select option(s) from a list of choices
- In old text-based systems, screen forms were composed entirely of fields into which the user had to type information
- Screen fill-in forms are derived from their antecedents, paper forms, indicating its advantage of familiarity. If it is designed well, a form will aid the user in understanding its purpose and allow fast and easy entry of information
- Conversely, a poorly designed screen form can be inefficient and aggravating to complete

Interface Types: Question & Answer

- Combine some of the features of both menus & fill-in form
- User is led through interaction via a series of questions, yes/no, multiple choice, etc.



The image shows a computer monitor with a white screen. On the screen is a form titled "MASSACHUSETTS 1990 TAX FORM 1". The form contains four sections, each with a prompt, instructions, and a user input line. The input lines show the user's responses: "Jones, Jean A.", "M", "J", and "Jones, James R.". The monitor has a simple black frame and a small rectangular base.

MASSACHUSETTS 1990 TAX FORM 1

Enter your name:
(Last, First M.)
> Jones, Jean A.

Enter your marital status:
(M = married, S = single.)
> M

Are you filing jointly or separately?:
(J = jointly, S = separately.)
> J

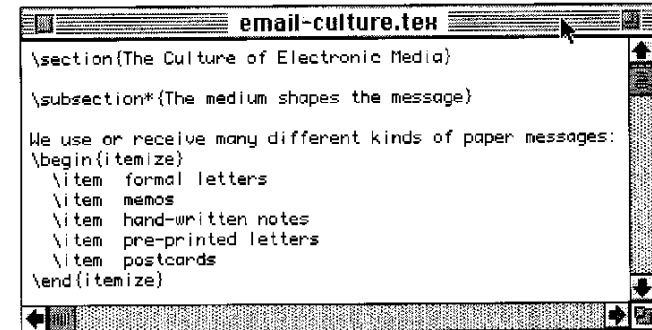
Enter your spouse's name:
(Last, First M.)
> Jones, James R.

Interface Types: WIMP

- Key features of a Graphical User Interface
- Consist of
 - Windows
 - Icons
 - Menus
 - Pointers
- Another form: windows, icons, mice & pull-down menus
- Default style for majority of interactive computer systems today, especially PCs & desktop machines
- Support **direct manipulation** of selecting, moving and resizing objects

Interface Types: WIMP

Windows



- Areas of the screen that behave as if they were independent terminals
- Can contain text or graphics
- Can be moved or resized
- Can overlap & obscure each other, or can be laid out next to one another (tiled)
- Scrollbars allow the user to move the contents of the window up & down or from side to side
- Title bars describe the name of the window

Interface Types: WIMP

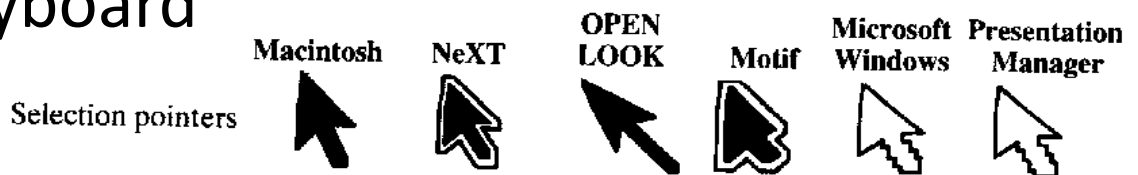


Icons

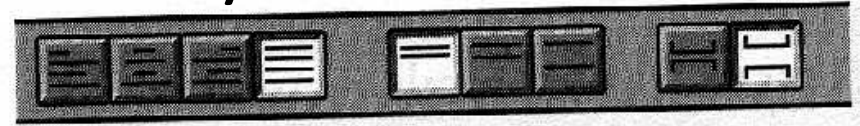
- Small picture or image, used to represent some objects (file, folder, etc.) in the interface
- Windows can be closed down to this small representation (iconised) allowing many windows to be accessible

Pointers

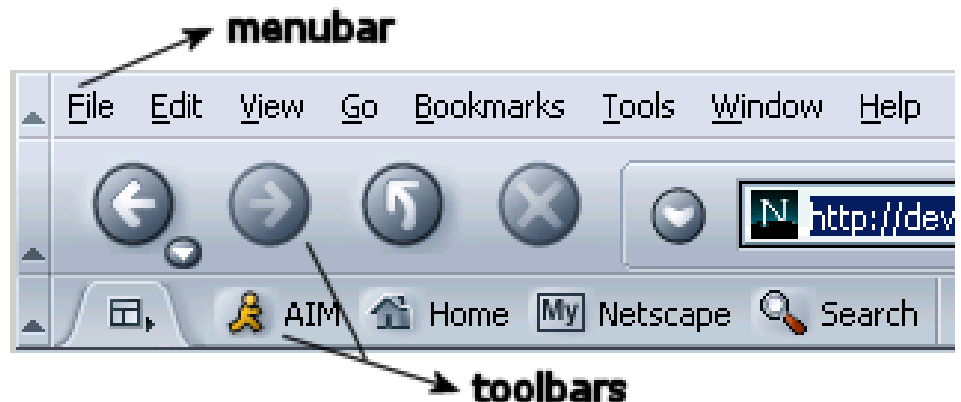
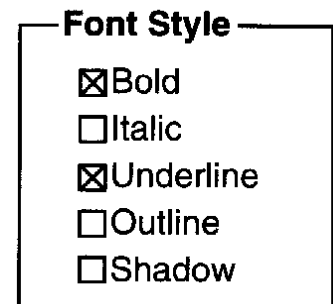
- The pointer is controlled by mouse, joystick, trackball, cursor keys or keyboard



Other WIMP related styles



- **(Radio) button** - set of mutually exclusive choices
- **Check box** - set of non-exclusive choices
- **Toolbar** – collection of small buttons, each with icons



Interface Types

- **Dialogue box** - information windows that pop up to inform of some important event or request certain information
- **Palettes** – mechanism for making the set of possible modes and the active mode visible to user

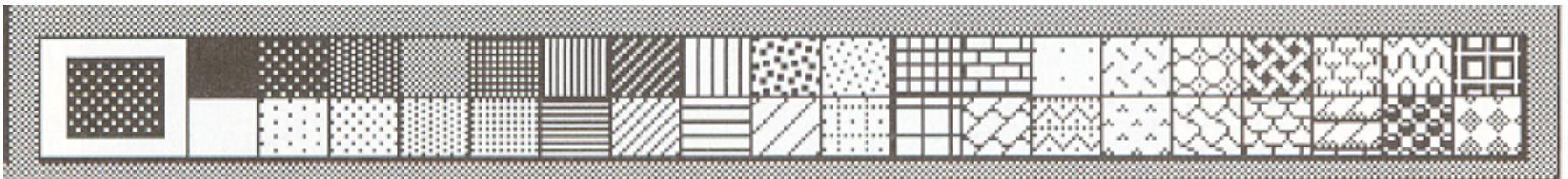
Title:

Subject:

Author:

Version:

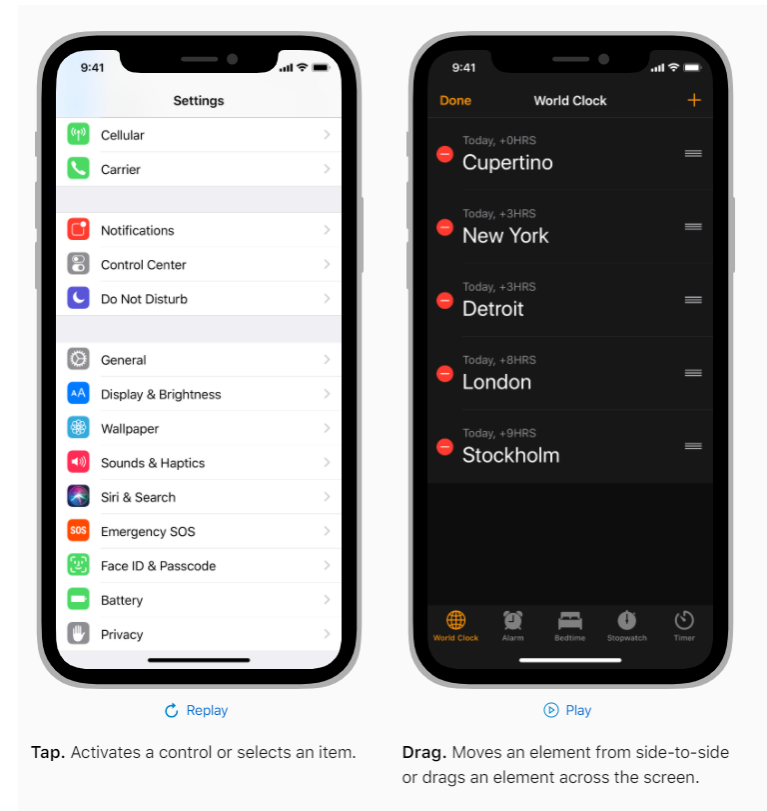
Keywords:



Post-WIMP Interface

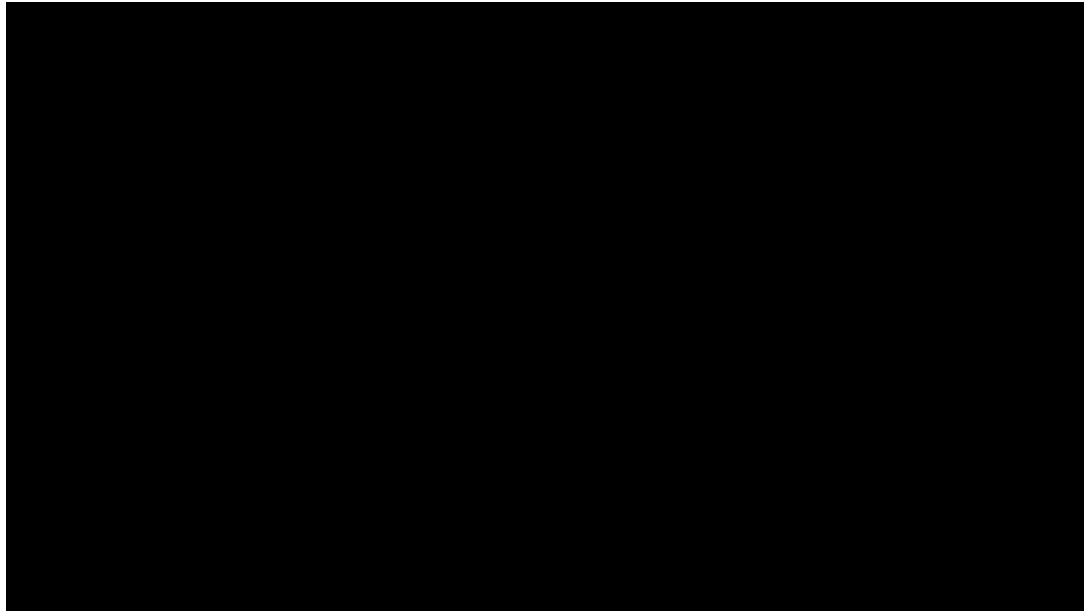
Natural user interface →
Touch/Gesture + Voice

<https://developer.apple.com/design/human-interface-guidelines/ios/user-interaction/gestures/>



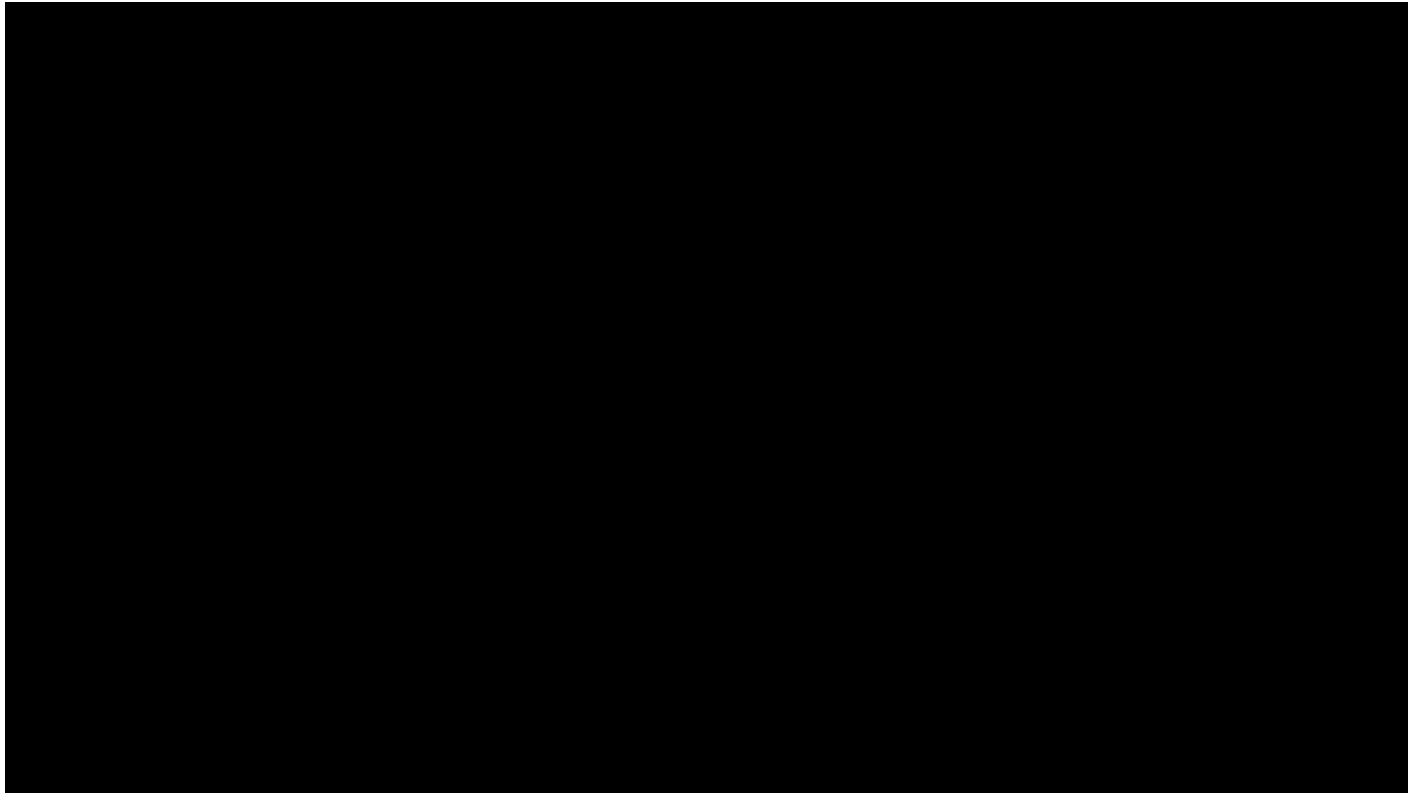
Interface Types: Natural Language

- An attractive option: familiar
- Speech recognition or typed natural language
- Problems:
 - Can be vague & ambiguous



<https://www.youtube.com/watch?v=nwPtcqcqz00>

- Diverse range of users with different accents



<https://youtu.be/J3lYLphzAnw>

- Always prepare a backup for natural language interface

Interface Types – Virtual Reality (VR)



- Refer to the experience of interacting with an artificial environment, which makes it feel virtually real
- Images are displayed stereoscopically to the users through shutter glasses and objects within the vision field can be interacted with via an input device
- Provide immersive experience because users can interact with objects and navigate in 3D space
- Illusion afforded by current technology can make virtual objects appear to be very life-like and behave according to the laws of physics, e.g., landing and take-off terrains developed for flight simulations
- Offer different viewpoints, e.g., first-person perspective in flying/driving simulations and shooter game, third-person perspective in game via controlling the interactions of main character
- Other applications include IMAX cinemas, virtual prototype home for real estate industry, education

港島

九龍

新界東

新界西

Q

VR搜樓

明園 第一期 中層 7室
北角/炮台山 | 3房



VR搜樓 視頻聯樓

售 \$778萬

實 662 呎
售 \$11,752/呎

摩星嶺村 A3座 A3室
西半山 | 4房



VR搜樓 視頻聯樓

售 \$6,600萬

實 2,508 呎
售 \$26,315/呎
建 3,000 呎
售 \$22,000/呎

細城峰 1座 高層 A室
堅尼地城/西貢盤 | 3房



VR搜樓 視頻聯樓

售 \$2,350萬

實 879 呎
售 \$26,734/呎
建 1,186 呎
售 \$19,814/呎

趙苑 中層 A室
西半山 | 4房



VR搜樓 視頻聯樓

售 \$4,600萬

租 \$10萬
實 2,666 呎
售 \$17,254/呎 租 \$38/呎
建 3,729 呎
售 \$12,335/呎 租 \$27/呎

<https://hk.centanet.com/findproperty/list/>

Matterport for everyone

Our customers use Matterport every day to save time, to reduce costs, to streamline processes - to make their lives easier.



Enterprise



Real Estate



iPad & iPhone



App Development



3D Photography



Architecture
Engineering
Construction



Travel &
Hospitality



Insurance



Restoration



Facilities
Management

<https://matterport.com/how-it-works>

Interface Types – Augmented Reality (AR)

- Bridge physical and digital worlds: virtual representations are superimposed on physical devices and objects
- Many AR apps available for a range of contexts, from education to car navigation, e.g., Pokémon Go and virtual fitting
- To reveal digital information, the user opens the AR app on a smartphone or tablet, and the content appears superimposed on what is viewed through the screen



<https://arvr.google.com/ar/>

Interface Types: Haptic

- Provide tactile feedback by applying vibration and forces to the user using actuators that are embedded in clothing or wearables
- Example interfaces include games employing vibration to enrich experience

