SEEM3510 Human-Computer Interaction

Helen Meng

Dept. of Systems Engineering & Engineering Management

Philip Fu

Dept. of Computer Science & Engineering

Spring 2022

Week 3: Guidelines, Principles and Theories

Outline

- Conceptual Models
- Theory
- Principles
- Guidelines

Conceptualizing an Interface

How can we conceptualize an interaction?

- Case: In-car infotainment hub
  - E.g. should we use GUI, speech, gestures, …

Source: Google images

How can we conceptualize an interaction?

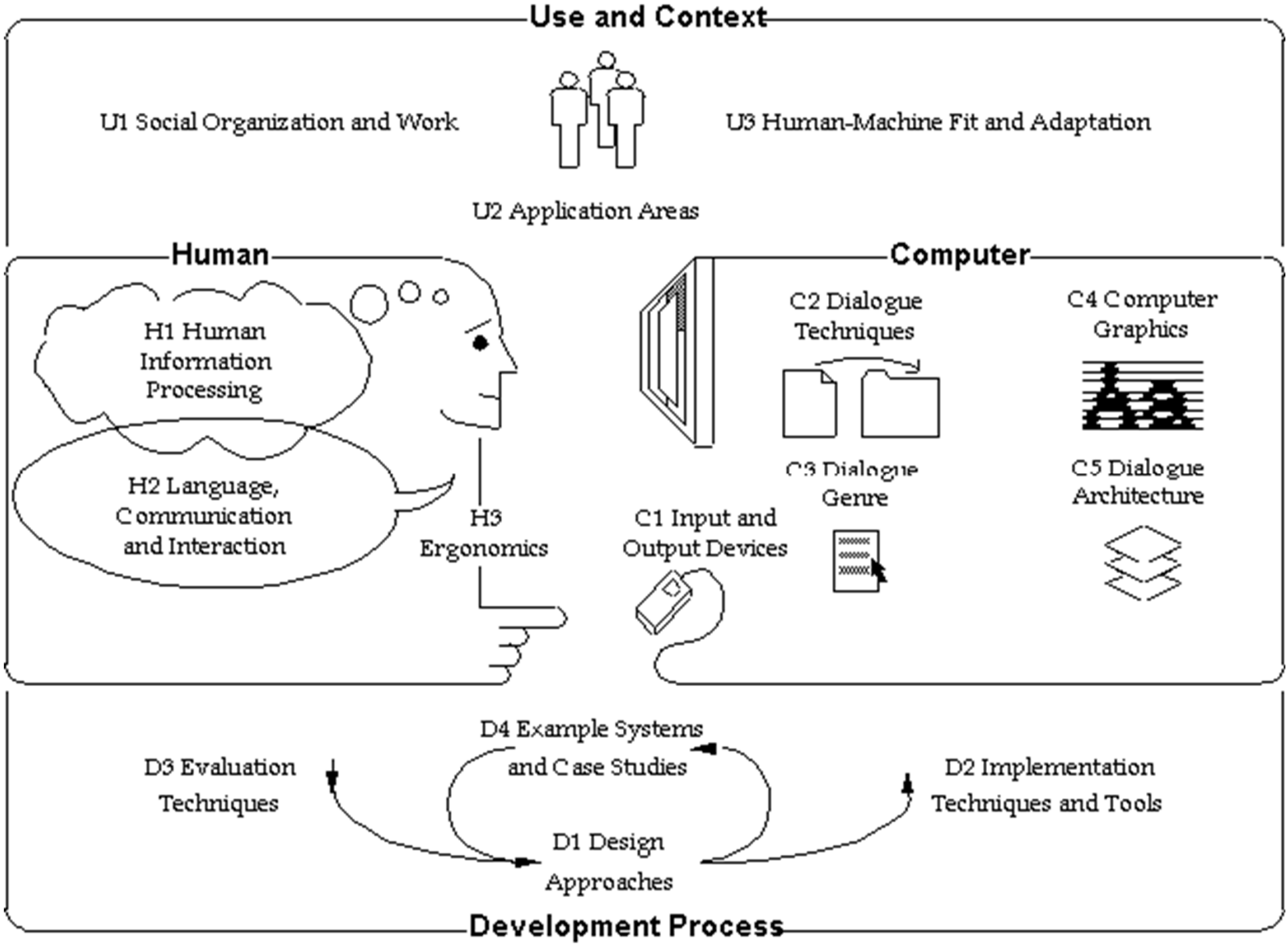
- Case: In-car infotainment hub
  - E.g. should we use GUI, speech, gestures, AR/VR?

Source: Google images

Determining the Problem Space

- Stakeholders
- Needs and preferences
- Competitive advantage(s) in proposed design
- Assumptions and claims
- User activities / actions / environment / usage context
- Resources e.g. budget, timeline manpower

Many different points of view!



Source: ACM SIGCHI

Conceptual Model

A conceptual model is:

- A high-level description of an application

- Enumerates all concepts in application exposed to the user

- Describes how the concepts relate to each other

- Explains how those concepts fit into tasks performed by the user

Reference: [1]

Conceptual Model (Example)

A conceptual model is: Alarm clock

- A high-level description / functionality of an application

- Enumerates all concepts in application exposed to user

- Describes how the concepts relate to each other

- Explains how those concepts fit into tasks performed by the user

Reference: [1]

Conceptual Model (Example)

A conceptual model is: Alarm clock

- A high-level description / functionality of an application

The clock stores the current time of day, displays it constantly, alerts user with alarm

- Enumerates all concepts in application exposed to user

- Describes how the concepts relate to each other

- Explains how those concepts fit into tasks performed by the user

Reference: [1]

Conceptual Model (Example)

A conceptual model is: Alarm clock

- A high-level description / functionality of an application

The clock stores the current time of day, displays it constantly, alerts user with alarm

- Enumerates all concepts in application exposed to user

Time: hour/min/sec, Alarm (specified time)

- Describes how the concepts relate to each other

- Explains how those concepts fit into tasks performed by the user

Reference: [1]

Conceptual Model (Example)

A conceptual model is: Alarm clock

- A high-level description / functionality of an application

The clock stores the current time of day, displays it constantly, alerts user with alarm

- Enumerates all concepts in application exposed to user

Time: hour/min/sec, Alarm (specified time)

- Describes how the concepts relate to each other

Unit relationships, condition of triggering alarm

- Explains how those concepts fit into tasks performed by the user

Reference: [1]

Conceptual Model (Example)

A conceptual model is: Alarm clock

- A high-level description / functionality of an application

The clock stores the current time of day, displays it constantly, alerts user with alarm

- Enumerates all concepts in application exposed to user

Time: hour/min/sec, Alarm (specified time)

- Describes how the concepts relate to each other

Unit relationships, condition of triggering alarm

- Explains how those concepts fit into tasks performed by the user

User sets hr/min/sec, sets alarm at specific time, unset alarm

Reference: [1]

Conceptual Model (Fancier Example)

A conceptual model is: Alarm clock with multiple alarms

- A high-level description / functionality of an application

The clock stores the current time of day, displays it constantly, alerts user with multiple alarms

- Enumerates all concepts in application exposed to user

Time: hour/min/sec, Multiple Alarms (specified times).

- Describes how the concepts relate to each other

Unit relationships, condition of triggering alarm

- Explains how those concepts fit into tasks performed by the user

User sets hr/min/sec, sets alarms at specific times, unset specified alarms

Conceptual Model (Even Fancier Example)

A conceptual model is: Alarm clock with multiple alarms and calendar

- A high-level description / functionality of an application

The clock stores the current time of day, displays it constantly, alerts user with multiple alarms, yr/mnth/day

- Enumerates all concepts in application exposed to user

Time: hour/min/sec, Multiple Alarms (specified times), Date

- Describes how the concepts relate to each other

Unit relationships, condition of triggering alarm, Date

- Explains how those concepts fit into tasks performed by the user

User sets hr/min/sec, sets alarms at specific times, unset specified alarms, set Dates

Conceptual Model (Simpler Example)

A conceptual model is: Smartphone's clock

- A high-level description / functionality of an application

The clock obtains the current time of day, displays it constantly, alerts user with alarm

- Enumerates all concepts in application exposed to user

Time: hour/min/sec, Alarm (specified time)

- Describes how the concepts relate to each other

Unit relationships, condition of triggering alarm

- Explains how those concepts fit into tasks performed by the user

~~User sets hr/min/sec~~, sets alarm at specific time, unset alarm

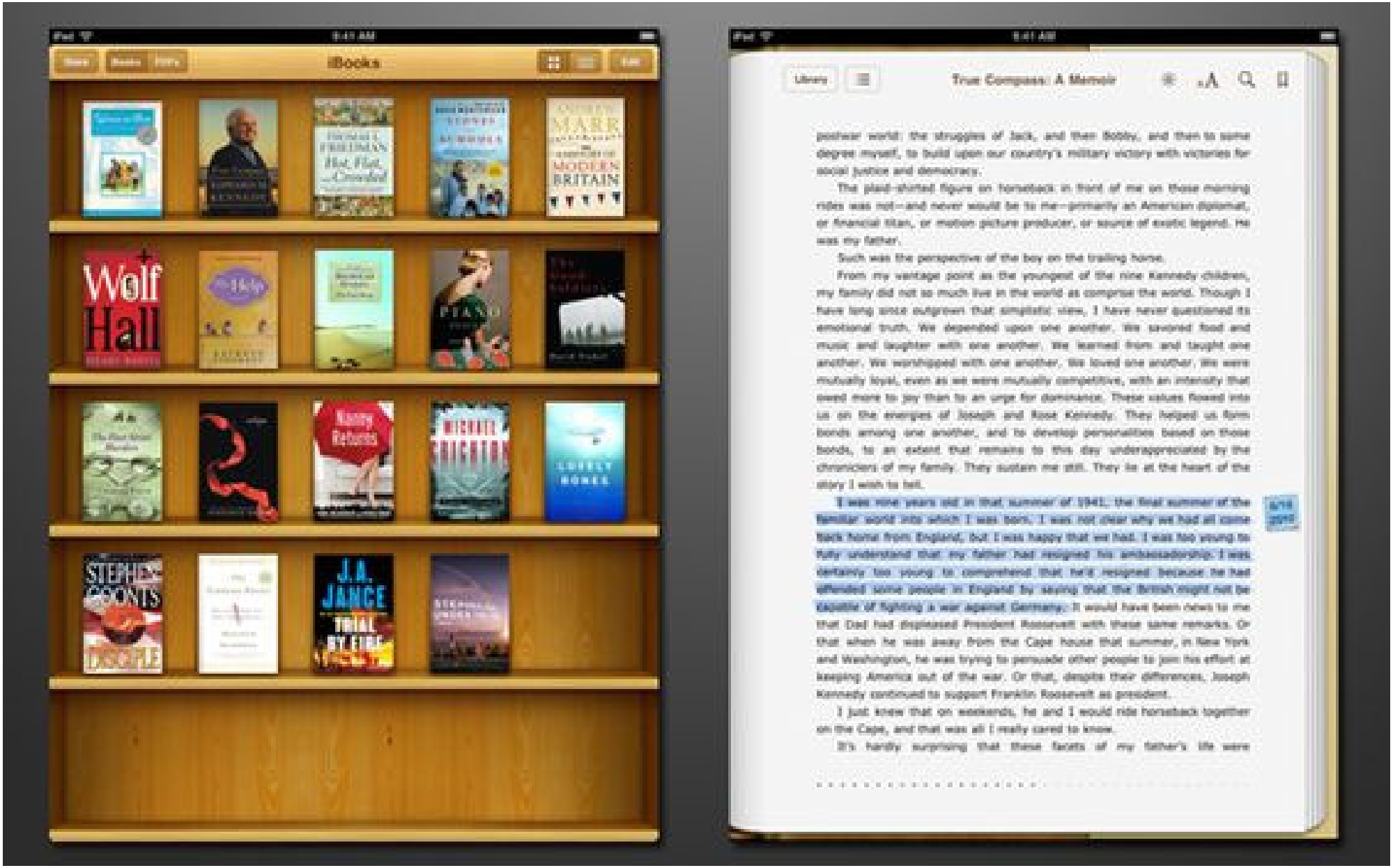
Reference: [1]

Conceptual Models: Desirable Characteristics

- Intuitive and simple
- May provide multiple pathways for certain actions (trade-offs)
- May reference well-established conceptual models, e.g. metaphors

Interface Metaphors

- Intended to provide familiar (physical) components that enable people to readily understand the underlying conceptual model
- Examples:
  - Desktop metaphor
  - Book metaphor
  - Shopping cart metaphor
  - Gallery metaphor
  - Spreadsheet metaphor
  - Calculator metaphor
- Discrepancies between metaphor and interface design exist

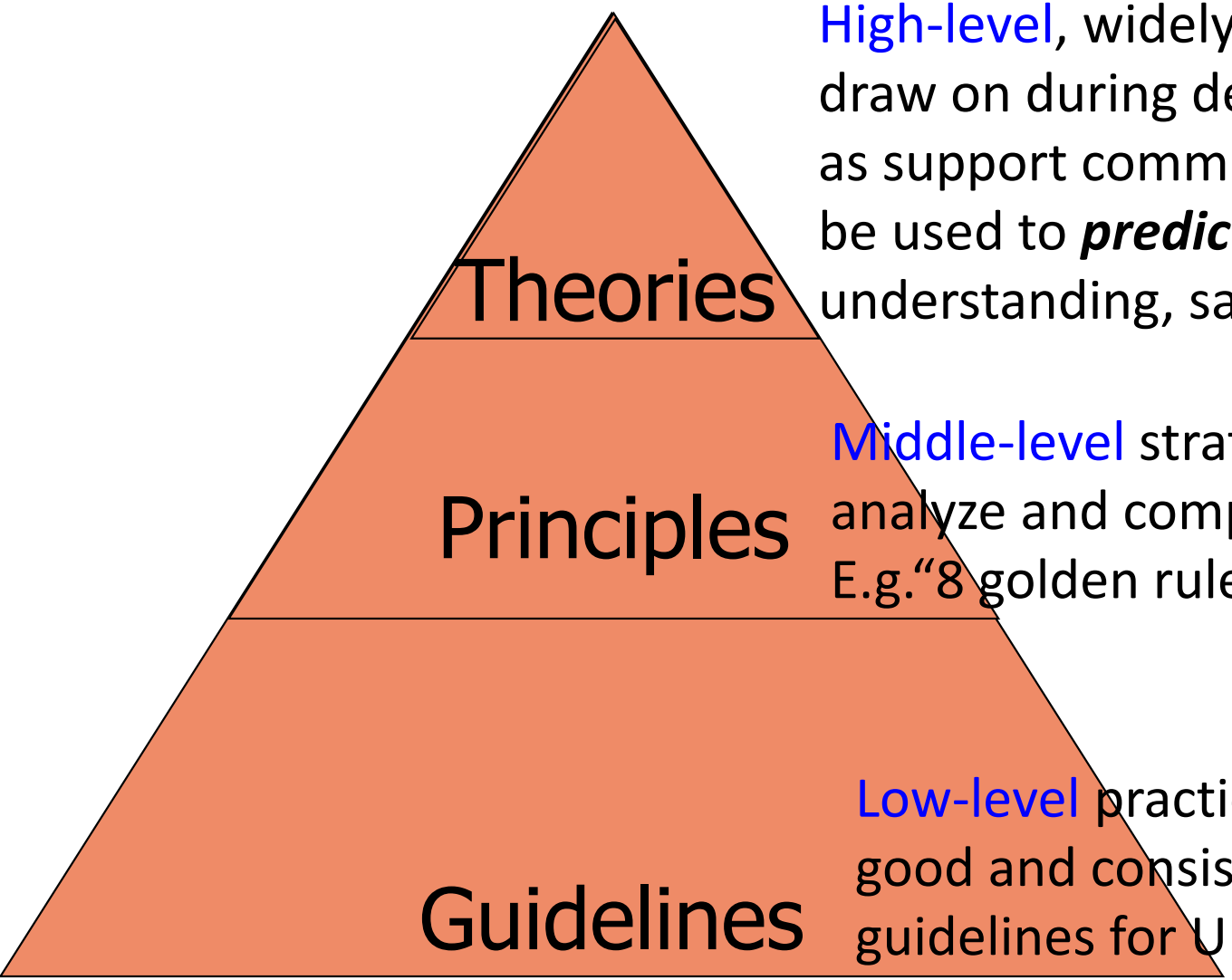

iBooks
Wolf Hall
STEPHEN COONTS
J.A. JANCE
TRIAL BY FIRE
True Compass: A Memoir
postwar world: the struggles of Jack, and then Bobby, and then to some degree myself, to build upon our country's military victory with victories for social justice and democracy.
The plaid-shirted figure on horseback in front of me on those morning rides was not—and never would be to me—primarily an American diplomat, or financial titan, or motion picture producer, or source of exotic legend. He was my father.
Such was the perspective of the boy on the trailing horse.
From my vantage point as the youngest of the nine Kennedy children, my family did not so much live in the world as comprise the world. Though I have long since outgrown that simplistic view, I have never questioned its emotional truth. We depended upon one another. We savored food and music and laughter with one another. We learned from and taught one another. We worshipped with one another. We loved one another. We were mutually loyal, even as we were mutually competitive, with an intensity that owed more to joy than to an urge for dominance. These values flowed into us on the energies of Joseph and Rose Kennedy. They helped us form bonds among one another, and to develop personalities based on those bonds, to an extent that remains to this day underappreciated by the chroniclers of my family. They sustain me still. They lie at the heart of the story I wish to tell.
I was nine years old in that summer of 1941, the final summer of the familiar world into which I was born. I was not clear why we had all come back home from England, but I was happy that we had. I was too young to fully understand that my father had resigned his ambassadorship. I was certainly too young to comprehend that he'd resigned because he had offended some people in England by saying that the British might not be capable of fighting a war against Germany. It would have been news to me that Dad had displeased President Roosevelt with these same remarks. Or that when he was away from the Cape house that summer, in New York and Washington, he was trying to persuade other people to join his effort at keeping America out of the war. Or that, despite their differences, Joseph Kennedy continued to support Franklin Roosevelt as president.
I just knew that on weekends, he and I would ride horseback together on the Cape, and that was all I really cared to know.
It's hardly surprising that these facets of my father's life were

Metaphor category	Sample A	Sample B	Sample C
Iconic visual Metaphor			
Iconic Metaphor ENTAILMENTS	This web mail service is a posting organization So you can send mail You can receive mail You may have posting address And mail box	This Image capturing service Is like camera so you can take the picture. Save your picture in internet and take the images from web	This screen is the visual display that acts as the visual output of internet by which you Interact with web sites
Indexical Metaphor ENTAILMENTS			
Indexical Metaphor ENTAILMENTS	The magnifier first function is magnifying the small particle and in web it is the search tools of the web sites	The cart main function is carrying the goods and in the web it used as selling basket at online digital shops	The main function of scissor is cutting paper and coth and in the web it is used to delete files and folder from its location and restore it in another place

Reference: Missarraf et al., Int. J. Arts, 2018.

Theories

Theories, Principles and Guidelines



High-level, widely applicable frameworks to draw on during design and evaluation, as well as support communication and teaching. Can be used to ***predict*** performance, errors, understanding, satisfaction of user.

Middle-level strategies or rules used to analyze and compare design alternatives. E.g.“8 golden rules of UI design”, etc.

Low-level practices and rules that make for good and consistent design, e.g. Apple’s guidelines for UIs

Theories

- Help designers understand the relationships between concepts and generalize results
- Help designers predict performance for users, tasks, or designs
- Guide practitioners when making design trade-offs for products, e.g. Fitt’s Law

Types of Theories

- For evaluating user interfaces

By Type

a. Descriptive: describes user interfaces and their uses with consistent terminology and taxonomies
b. Explanatory: describes sequences of events with causal relationships
c. Prescriptive: offers guidelines for designers to make decisions
d. Predictive: enables comparisons of design alternatives based on numeric predictions of speed or errors

By Human Capacity (types of skills involved, types of users)

i. Motor: skill in pointing, clicking, dragging or other movements
ii. Perceptual: visual, auditory, tactile and other human sensory inputs
iii. Cognitive: problem solving with short and long-term memory

Some Theories

a) Design-by-level theory (GOMS)
b) Stages of action theory
c) Consistency theories
d) Contextual Theories
e) Dynamic Theories

Stages-of-Action Theory

Seven stages of action theory [Norman 2013]

1. Forming the goal (e.g. cook a nice meal)
2. Forming the intention (e.g. cook chicken with hoisin sauce)
3. Specifying the action (e.g. prepare chicken, prepare sauce…)
4. Executing the action
5. Perceiving the system state (e.g. smell, look, taste of dish)
6. Interpreting the system state (e.g. evaluate chicken dish)
7. Evaluating the outcome (e.g. evaluate chicken dish as a nice meal)

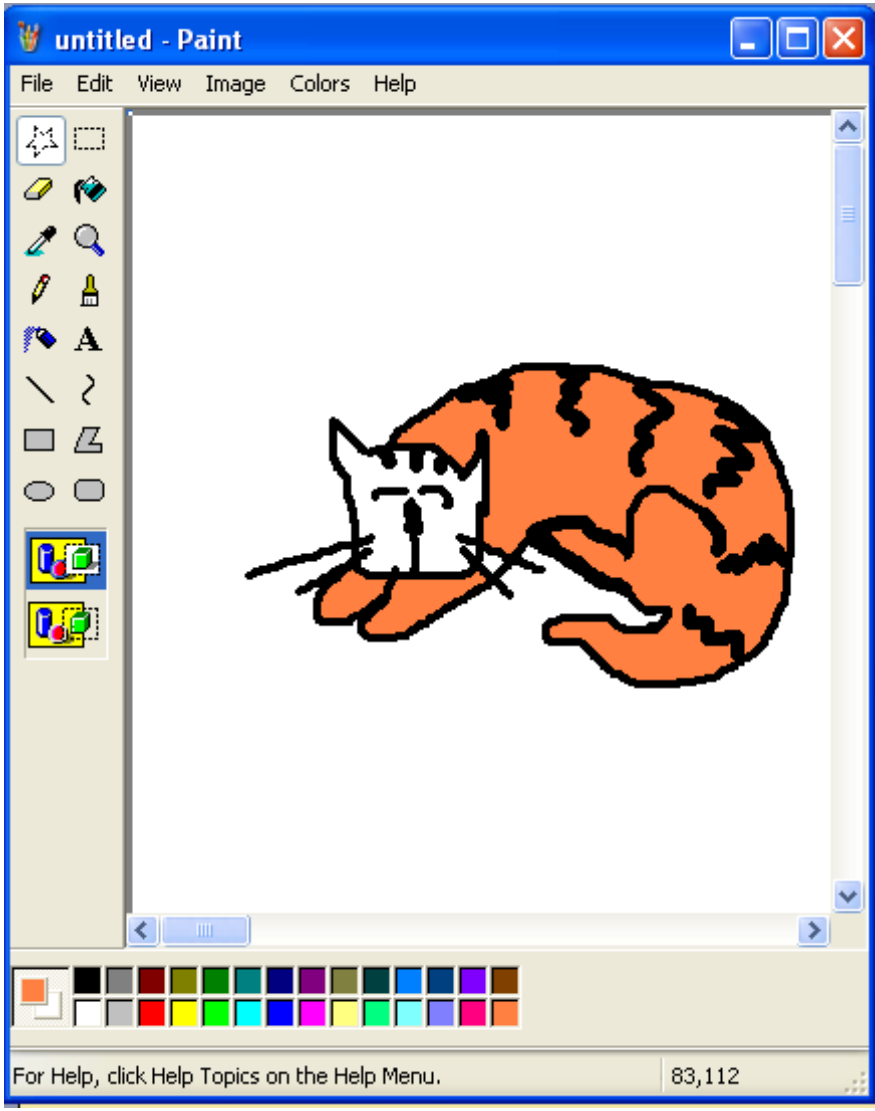
Example

1. Forming the goal
I want to paint the cat's head

2. Forming the intention
I will use the paint bucket
(instead of brush)

3. Specifying the action
To do this, I need to click on the paint-bucket icon, the cat's head region and the color

4. Executing the action
Physically doing the action with mouse and clicks

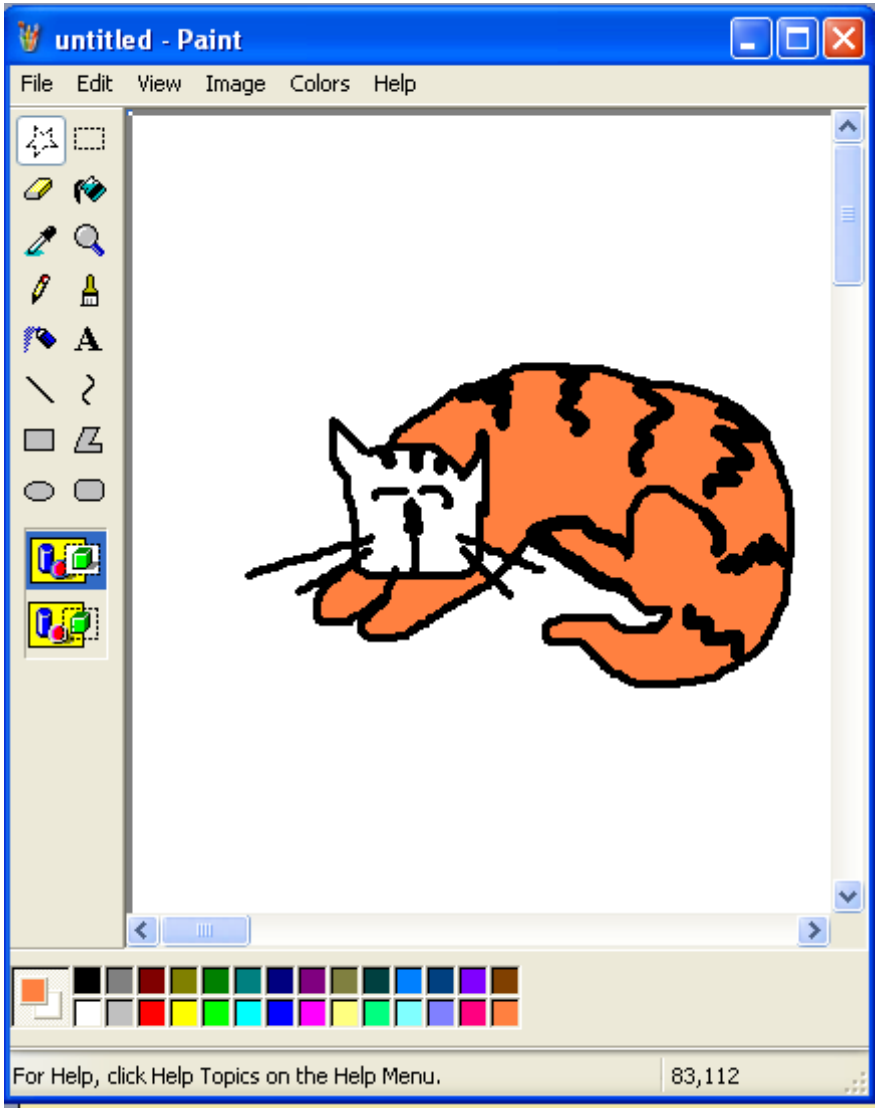


Example (cont-1)

5. Perceiving the system state
Display is the same

6. Interpreting the system state
Cat head is still white

7. Evaluating the outcome
Outcome is not what I want.

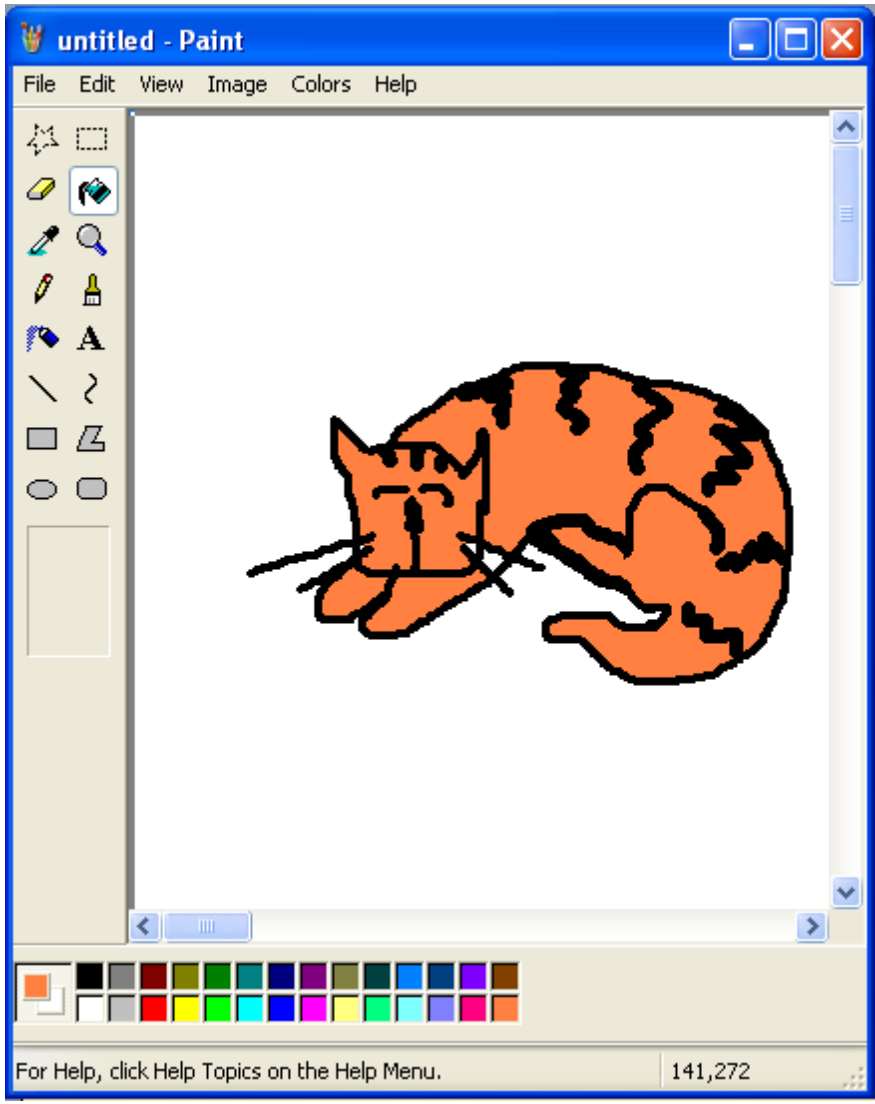


Example (cont-2)

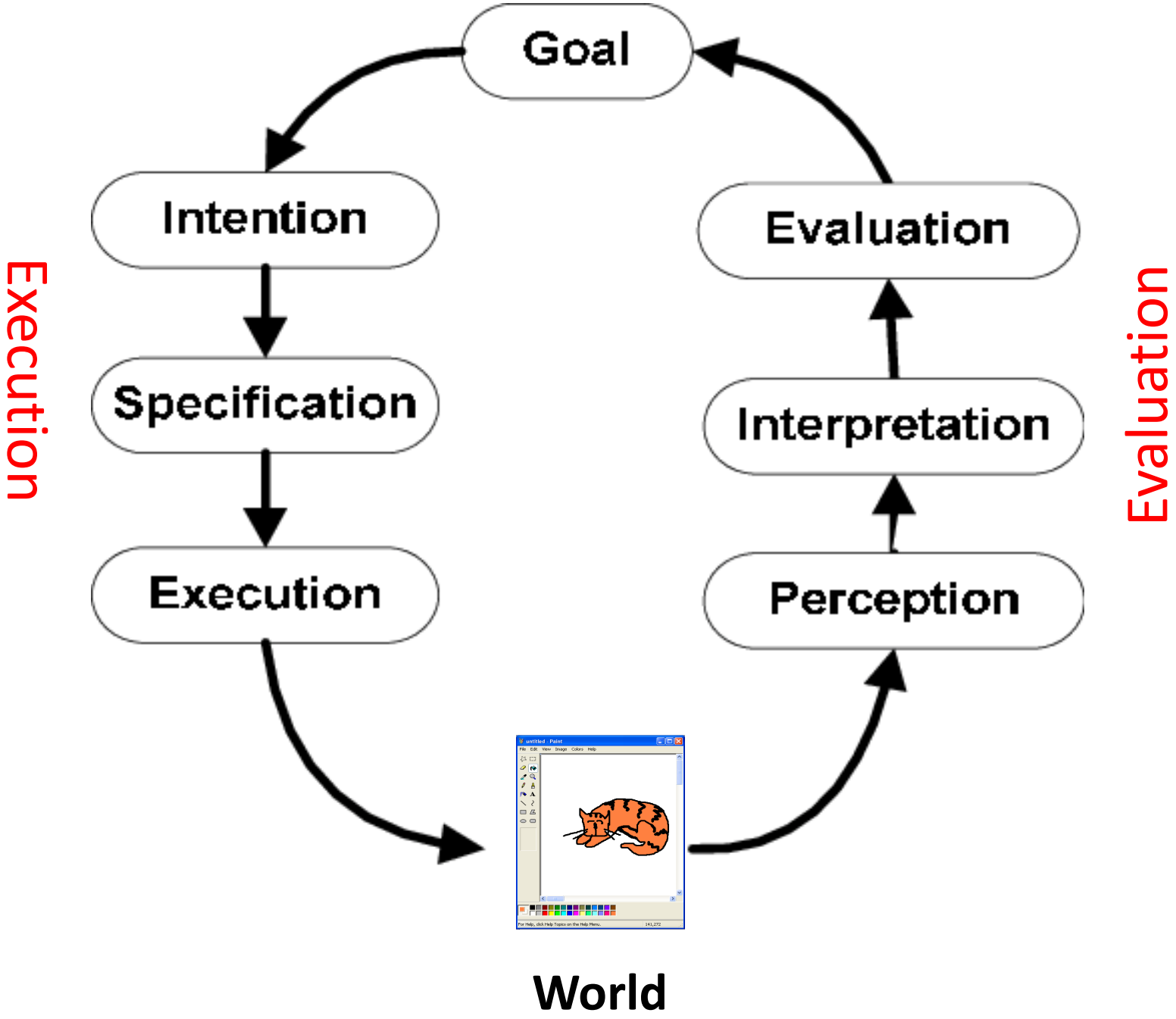
5. Perceiving the system state
Display has changed

6. Interpreting the system state
Cat head is orange

7. Evaluating the outcome
Outcome is good, I'm a happy user.



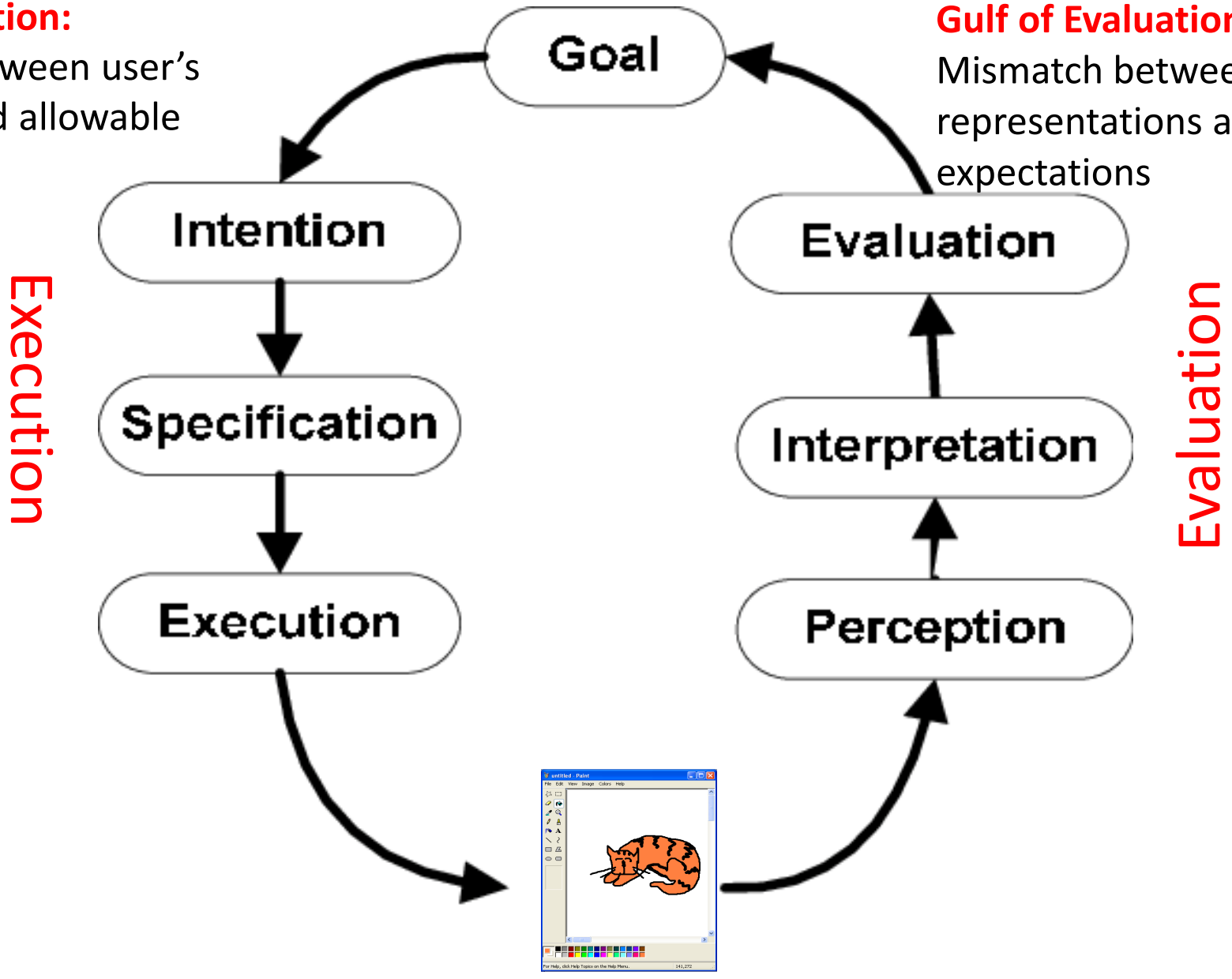
Cycles in Stages-of-Action Theory



Gulfs of Execution and Evaluation

Gulf of Execution:
Mismatch between user's intentions and allowable actions

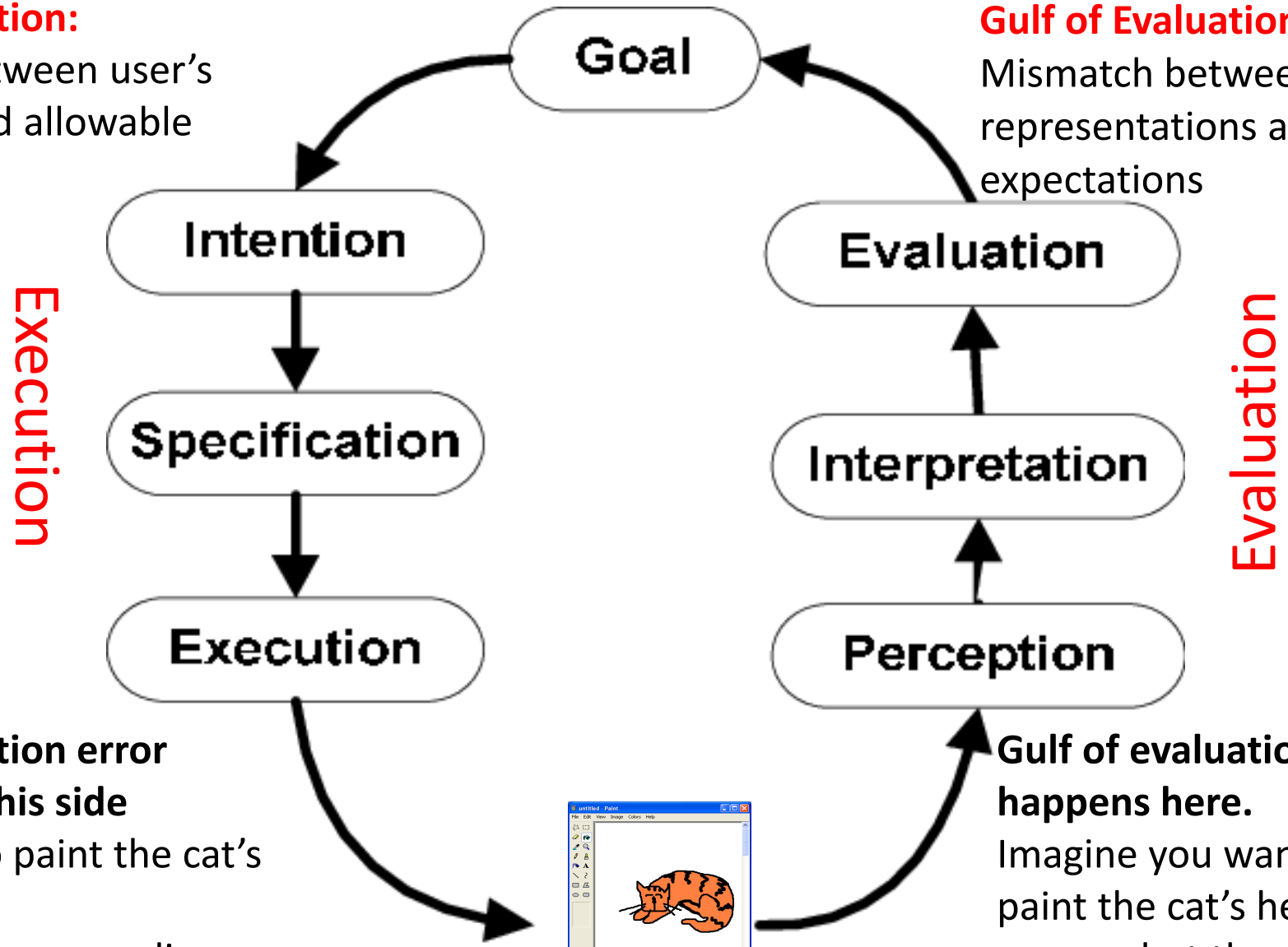
Gulf of Evaluation:
Mismatch between systems' representations and user's expectations



Gulfs and Errors

Gulf of Execution:
Mismatch between user's intentions and allowable actions

Gulf of Evaluation:
Mismatch between systems' representations and user's expectations



Gulf of execution error happens on this side
User wants to paint the cat's head striped
There is no corresponding action to perform this
The user is lost, confused, and makes errors.

Gulf of evaluation error happens here.
Imagine you want to paint the cat's head orange, but the program paints the entire image orange!

Stages-of-Action Theory on Errors

- Four critical points where user failures can occur
  - Users can form an inadequate goal
  - Users may not find correct interface object because of an incomprehensible label or icon
  - Users may not know how to specify or execute a desired action
  - Users may have received inappropriate or misleading feedback

- Four principles of good design (avoid gulf errors)
  - Have a good conceptual model with a consistent system image
  - The state and the action alternatives should be visible
  - Interface should include good mappings that reveal the relationships between stages
  - User should receive continuous feedback

Principles

Principles

Mid-level strategies or rules to analyze and compare design alternatives

1. Determine the user's skill levels
2. Identify the tasks
3. Choose an interaction style
4. 8 golden rules of interface design
5. Prevent errors
6. Increase automation but ensure human control

Principle 1: Determine user's skill levels

- Universal usability: age, gender, physical and cognitive abilities, training, motivation, goals and personality
- Skill levels:
  - Novice (first-time users) – anxiety, restrict options, simplify actions, clear error messages, positive reinforcement, more help
  - Knowledgeable (intermittent users) – organized menus (reduce memory loading), consistency, clear messages, context-dependent help
  - Expert (frequent users) – need efficiency, rapid response, shortcuts and macros, brief and terse feedback

Principle 2: Identify the Tasks

User needs assessment for UI design

- Which tasks are essential?
- Which tasks can be sidelined to preserve simplicity?
- Example – shaping the menu tree
  - Frequent tasks near top, rare tasks deeper down
  - Frequent actions may need special keys
  - Less frequent items may be performed with more steps
  - Infrequent items may be placed in embedded menus to maintain simplicity.

Principle 2: Identify the Tasks (cont)

- Task analysis – task decomposition, task frequencies

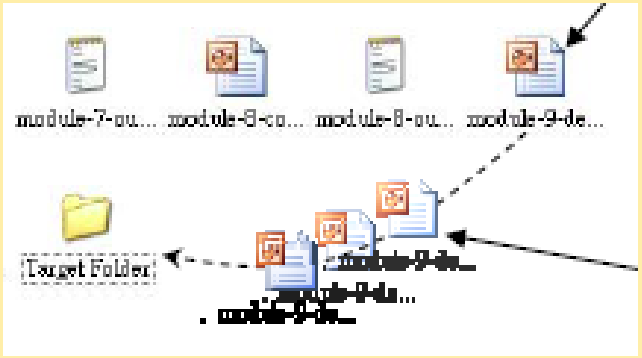
	TASK				
Job Title	Query by Patient	Update Data	Query across Patients	Add Relations	Evaluate System
Nurse	**	**			
Physician	**	*			
Supervisor	*	*	**		
Appointment personnel	****				
Medical-record maintainer	**	**	*	*	
Clinical researcher			***		*
Database programmer		*	**	**	*

FIGURE 3.3

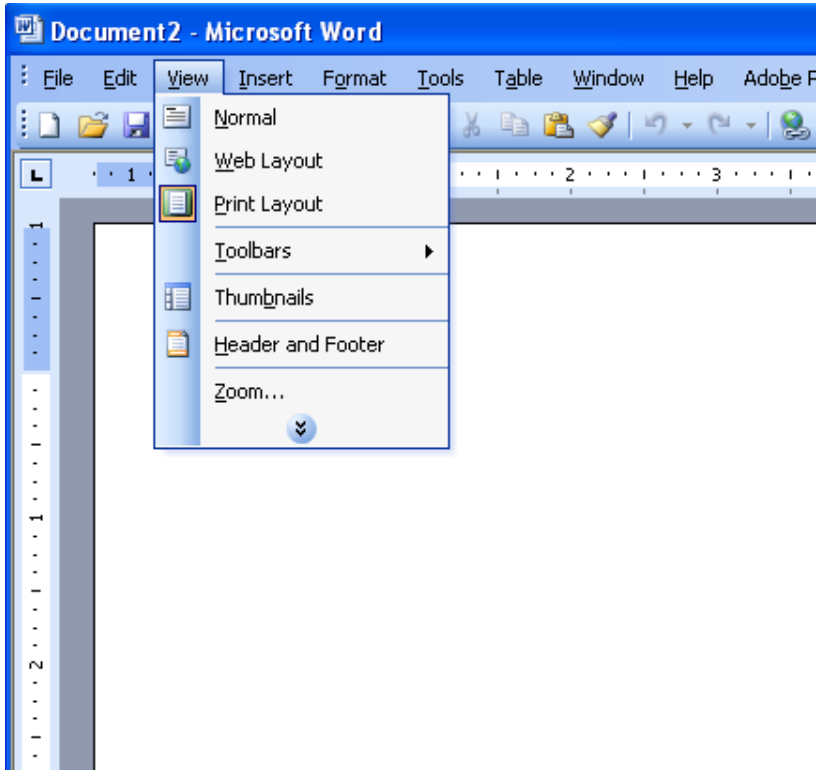
Frequency of Task By Job Title

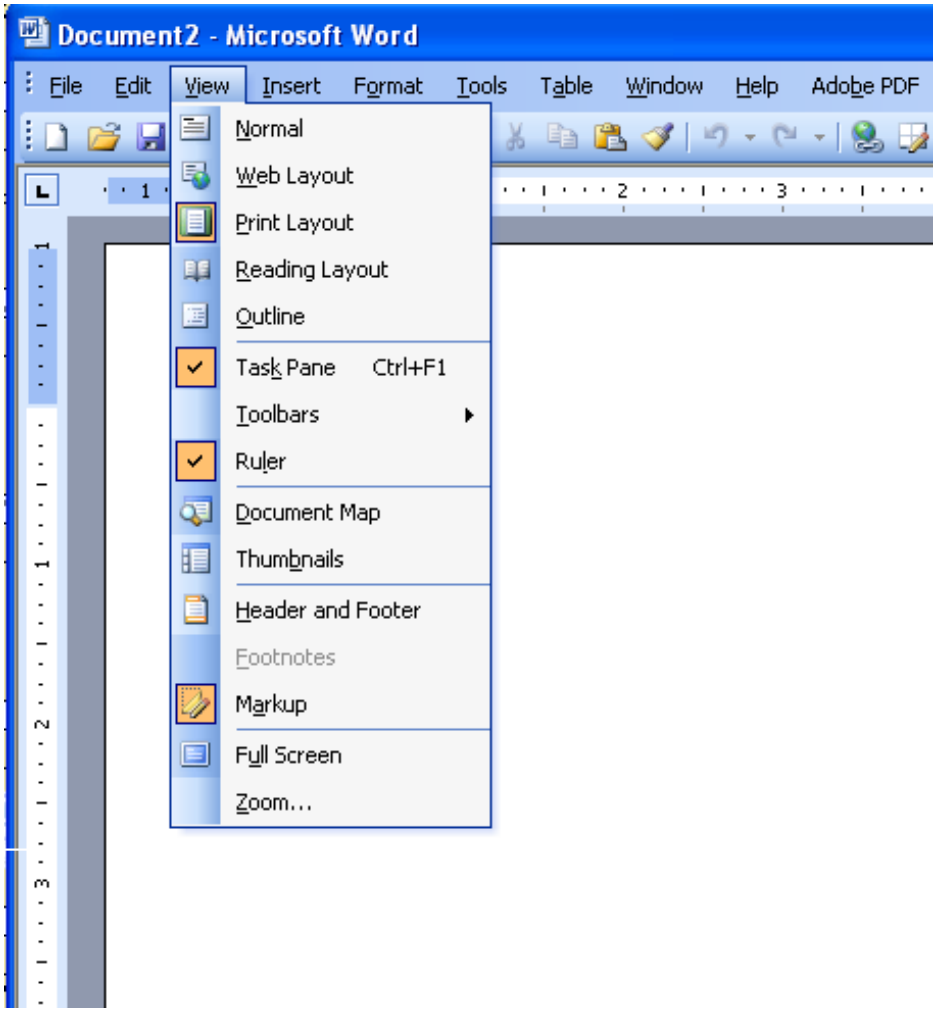
Hypothetical frequency-of-use of data for a medical clinic information system. Answering queries from appointment personnel about individual patients is the highest-frequency task (****), and lower-frequency use is shown with ***, **, or *.

Principle 3: Choose an Interaction Style

Advantages	Disadvantages
Direct manipulation	
Visually presents task concepts	May be hard to program
Allows easy learning	May require graphics display and pointing devices
Allows easy retention	
Allows errors to be avoided	
Encourages exploration	
Affords high subjective satisfaction	
Menu selection	
Shortens learning	Presents danger of many menus
Reduces keystrokes	May slow frequent users
Structures decision making	Consumes screen space
Permits use of dialog-management tools	Requires rapid display rate
Allows easy support of error handling	

Example: Adaptive Menus

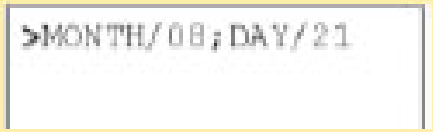




Windows adapt menu choices to usage pattern. This is to prevent "clutter"

Spectrum of directness

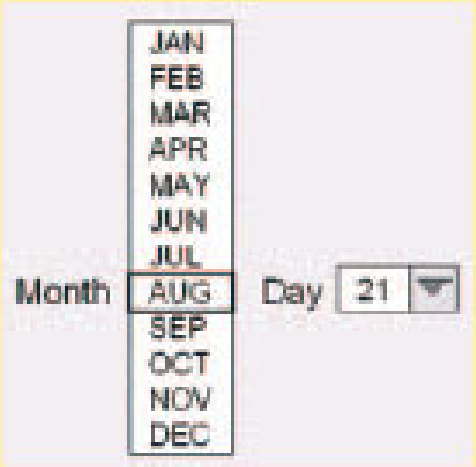
An example of progression towards more direct manipulation: less recall/more recognition, fewer keystrokes/fewer clicks, less capability to make errors, and more visible context.



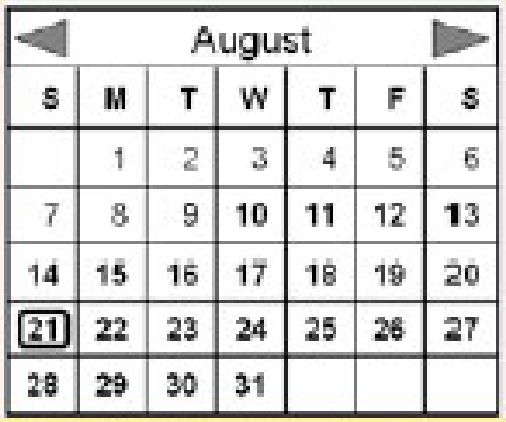
a. Command line

b. Form fill-in to reduce typing

c. Improved form fill-in to clarify and reduce errors



d. Pull-down menus offer meaningful names and eliminate invalid values



e. 2-D menus to provide context, show valid dates, and enable rapid single selection

Principle 3: Choose an Interaction Style

Form fill-in

Simplifies data entry

Requires modest training

Gives convenient assistance

Permits use of form-management tools

Consumes screen space

Command language

Flexible

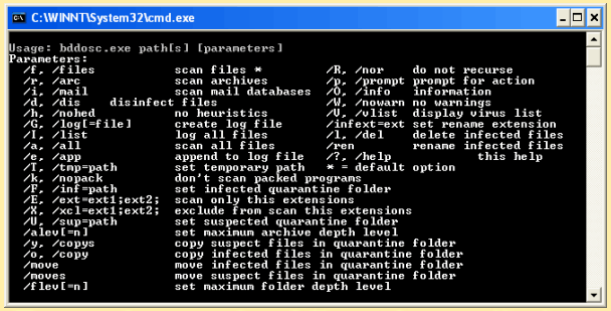
Appeals to "power" users

Supports user initiative

Allows convenient creation of user-defined macros

Poor error handling

Requires substantial training and memorization



Natural language

Relieves burden of learning syntax

Requires clarification dialog

May not show context

May require more keystrokes

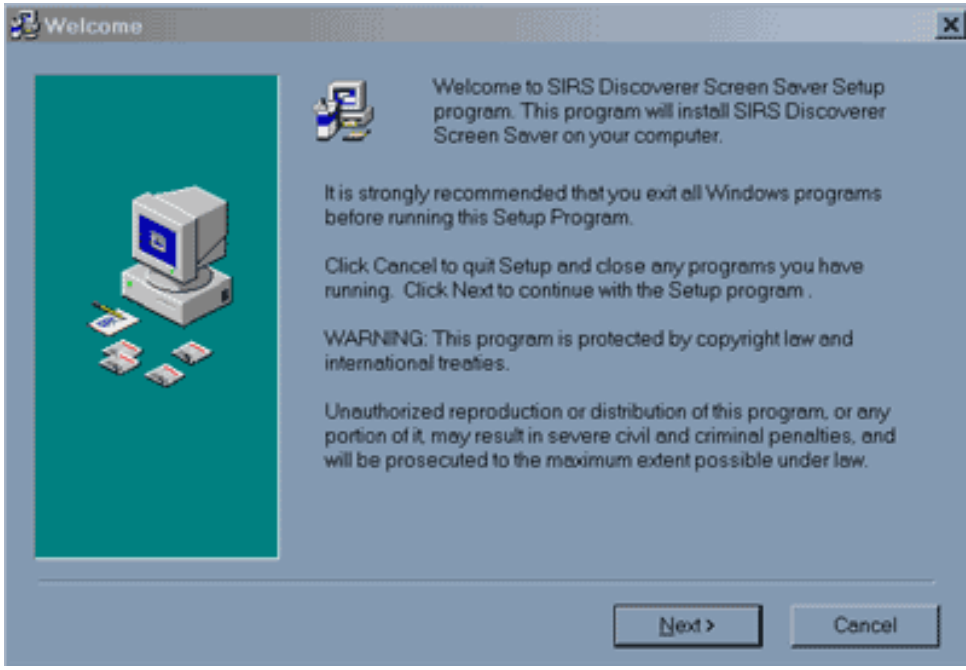
Unpredictable

Principle 4: 8 Golden Rules of Interface Design

A. Strive for consistency
B. Cater to universal usability
C. Offer informative feedback
D. Design dialogs to yield closure
E. Prevent errors
F. Permit easy reversal of actions
G. Keep users in control
H. Reduce short-term memory load

Design Dialogs to Yield Closure

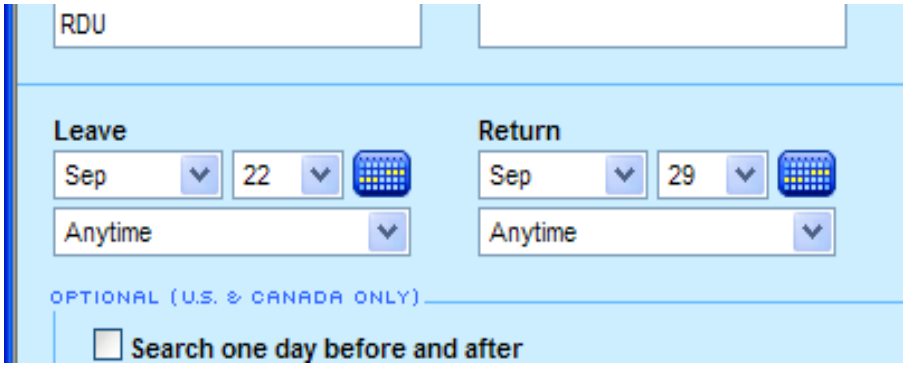
- Action sequences should be grouped with a beginning, middle, and end.
- Feedback provides a sense of accomplishment
- Ex. Purchasing items via internet and installing software has a clearly defined step-by-step process

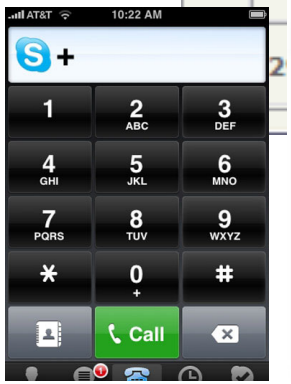


methods:
- step XX / YY
- progressive bar

Principle 5: Prevent Errors

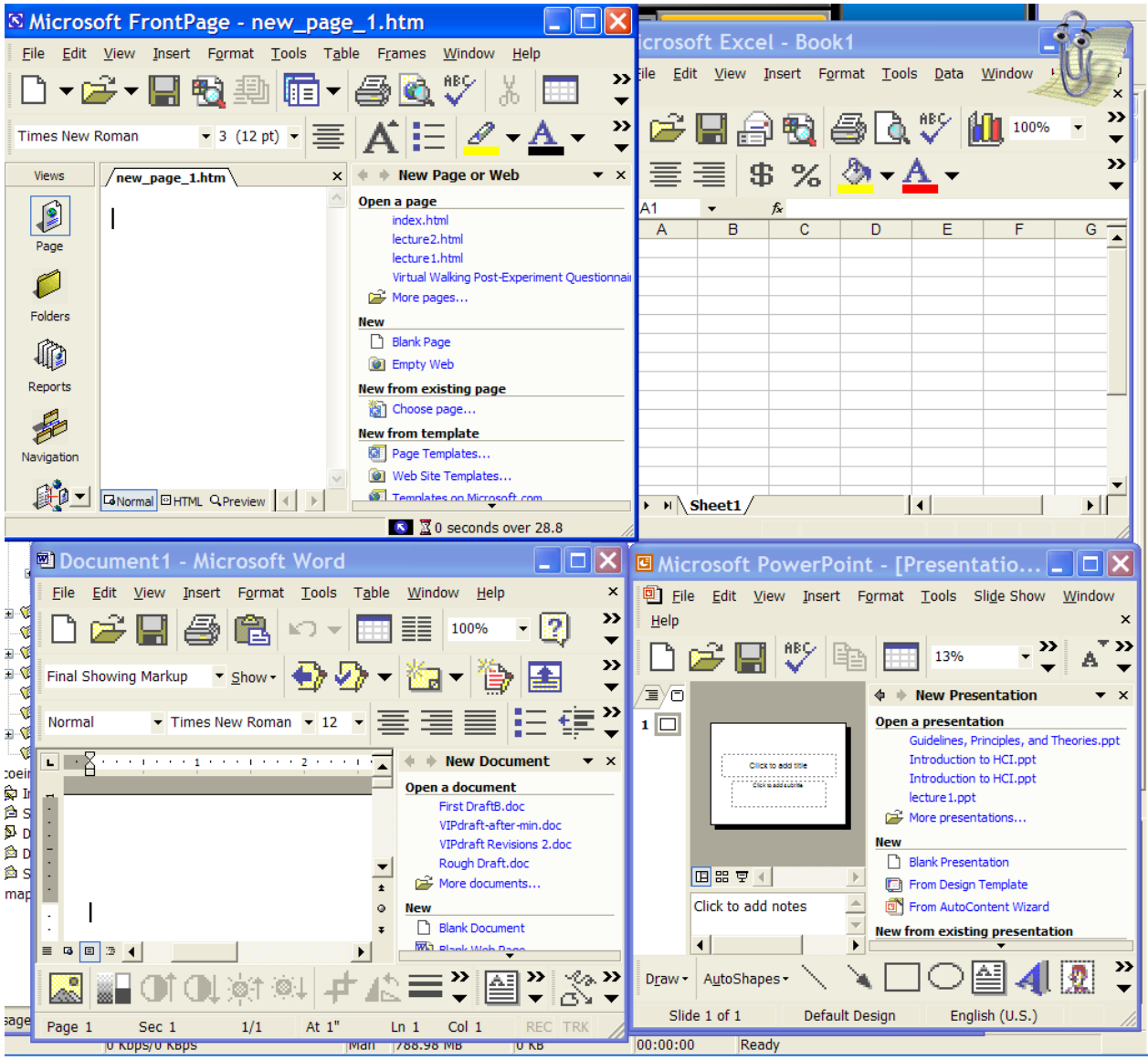
- Limit and avoid errors
  - Gray out menu items that do not apply
  - No text field for numeric input (and avoid typing)
- Maintain readability and consistency of information
- Proactively detect and fix errors
  - Spell checking
  - Auto-complete
  - Study error patterns for detection





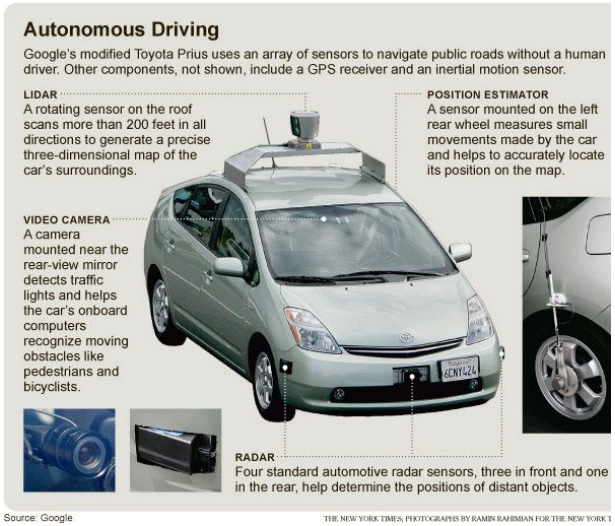
Strive for Consistency

- Consistent sequence of actions for similar situations
- Identical Terminology (prompts, menus, help)
- Consistent visual layout (fonts, color, etc.)



Principle 6: Automation while Preserving User Control

- **Motivation** for using computers: Automation
  - Increases over time as procedures become standardized
  - Improves speed / performance / productivity
  - Reduces error and user workload
- **Why still have humans** if we can automate tasks?
  - Real world: *open system (unpredictable and sys. failures)*
  - Computers: *closed system (predefined set of actions)*
- Humans are there for
  - Unexpected situations
  - Preserve safety
  - Avoid failures
  - Increase product quality
- Example: Air Traffic Control
  - Easy to automate
  - Unpredictable events (weather, emergencies, etc.). Need 'context'
  - Some real world situations are just too complex to model -> Need human judgments



What's the right level of automation in this app?

What happens if some parts of the system go down?

http://en.wikipedia.org/wiki/Google_driverless_car
https://www.youtube.com/watch?v=CgSDWoAhvLU
https://en.wikipedia.org/wiki/List_of_self-driving_car_fatalities

Guidelines

Guidelines

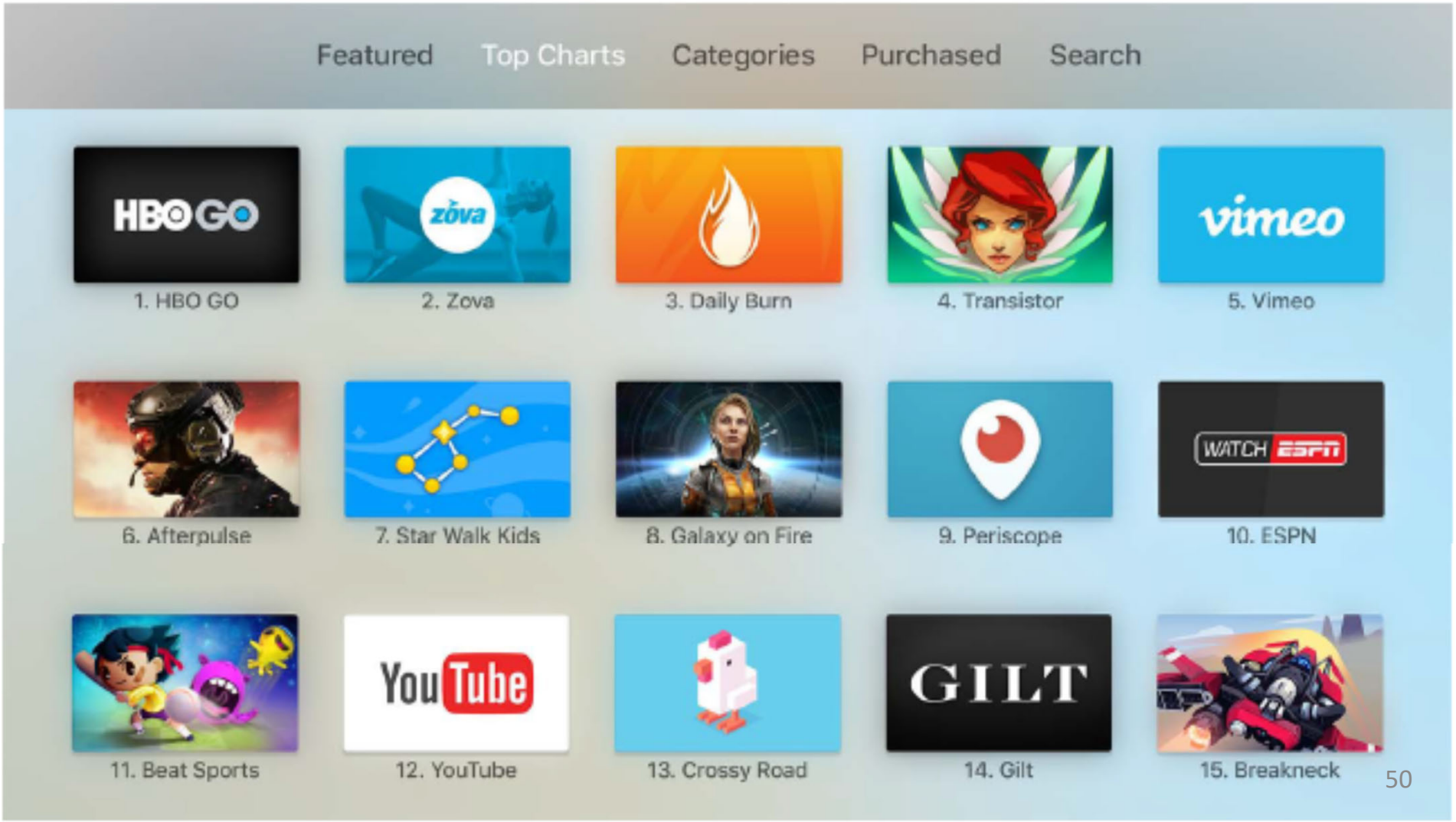
- Guidelines: Low-level focused advice about good practices

E.g. Apple released guidelines of interface design developed for Apple watch
https://developer.apple.com/watchos/human-interface-guidelines/visual-design/color/

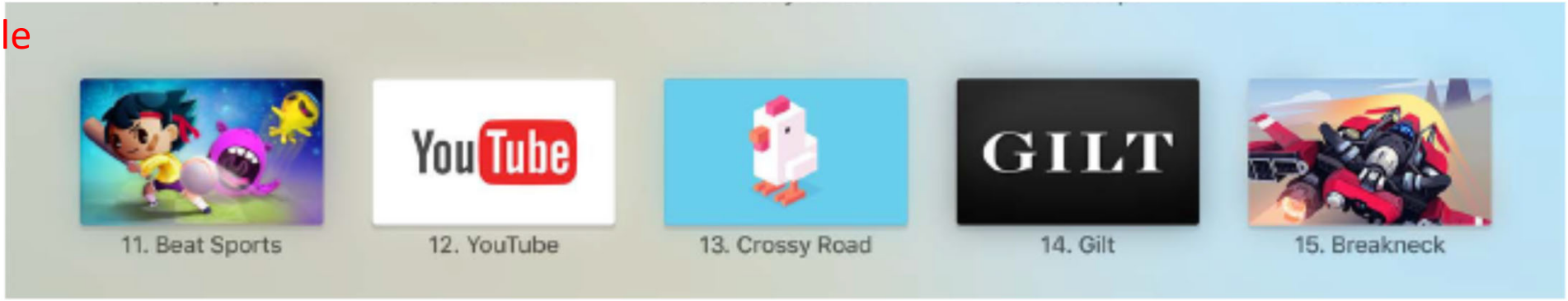
Apple

App Icon

Every app needs a beautiful and memorable icon that attracts attention in the App Store and that stands out on the Apple TV Home screen. Your icon is the first opportunity to communicate, at a glance, your app's purpose.



Apple



Provide a single focus point. Design an icon with a single, centered point that immediately captures attention and clearly identifies your app. People shouldn't have to analyze the icon to figure out what it represents.

Keep the background simple. Don't clutter your icon with too much background noise. Give it a simple background so higher layers stand out and it doesn't overpower other app icons nearby. Remember, you don't need to fill the entire icon with content.

Use words only when they're essential or part of a logo. An app's name appears below its icon when the icon is in focus. Don't include nonessential words that repeat the name or tell people what to do with your app, like "Watch" or "Play." If your design includes any text, emphasize words that relate to the actual content your app offers.

Don't include screenshots. Screenshots are too complex for an app icon and don't generally help communicate your app's purpose. Instead, take the time to design a beautiful and engaging icon that really pops.

Keep icon corners square. The system applies a mask that rounds icon corners automatically.

https://developer.apple.com/tvos/human-interface-guidelines/icons-and-images/

Website for Information Navigation

Designers Website

Website Design and Resource site

Creating A Website That Is Easy To Navigate And Use

Why design a website? If you answered, "Because I can" - wrong answer! The reason to design a website (the right reason anyway) is to set up a place (hence the term site) on the web, where people can access content in a straightforward manner.

Your site must market itself. That is, it must offer something with obvious value. That can include information, products, services, and entertainment, among many things. If it is not clear when someone reaches your home page (such as index.html) what they will find on your site, your site is not well designed.

And the process of delivering the point of your site in a readily accessible way does not just entail having good link buttons (although that can help). You must provide a mission statement or banner - something on your home page that says what your site is about.

Do you know what your site is about yet? If you don't, don't publish it! Sit down and write down your goals.

- What do you wish to accomplish?
- What type of content do you wish to share?
- How will people get to your content?

http://www.designers-website.eu/tips-on-building-a-website.php

Website for Information Navigation

Here are some dos and don'ts:

Do have logical and standard menu links (home, contact, about, etc.)
Do display menu links on each page (not just the home page)
Do display your pages in an organized manner
Do keep your pages simple when possible
Do use smaller compressed files (jpeg, mp3, etc.) (but too small is low quality)
Do make sure every page has a topic heading (helps visitors keep track of navigation)
Do use unique fonts for headings (at least in bold, perhaps a different color)
Do look at major commercial sites for visual and functional inspiration
Do make links descriptive ("Click Here!" for instance, is too generic)

Do not use tiny fonts (size 12 is fairly standard)
Do not use frames (this can cause pages to format incorrectly in browsers)
Do not use many flashy graphics or noises (these may not load, or may annoy)
Do not use unusual fonts for primary content (headings are ok though)
Do not use an image as a text background
Do not use a background color that does not contrast with your font
Do not use light grey text on a white background, for example!
Do not make the same mistakes other sites do (even popular ones)
Do not make pages so wide that people have to scroll to read
Do not use color combinations that strain the eyes of your visitors

http://www.designers-website.eu/tips-on-building-a-website.php

Guidelines for Disabled

- US Rehabilitation Act: promote **web accessibility for the disabled**
- WWW Consortium adopted these guidelines for designing web pages for disabled: http://www.w3.org/TR/WCAG20/ (V2.0 2008)
  - Text alternatives for non-text elements (e.g. images, sounds, videos)
  - Contrast
  - Color coding
  - Title each frame
  - Predictable structure
- Goal of the guidelines: enable screen readers or other technologies for disabled people to access the webpage contents

OGCIO Web Accessibility Guidelines

5. How Persons with Disabilities Use Websites

Most people think about visually impaired persons when it comes to accessibility, however there are many different types of disabilities and hence many different techniques that persons with disabilities can use to access websites.

Disabilities fall into four major categories:

In addition, there are many others who have temporary disabilities, for example, a wounded arm. Such injuries can make accessing websites just as difficult as it is for persons with permanent disabilities.

Examples of disabilities and the ways to overcome the constraints are outlined below.

5.1 Visual Impairment

In this case people either cannot see at all or have difficulty in seeing a computer screen.

It is critical that websites are designed to work with screen readers and screen magnifiers. It is also important that colours are visible to persons with colour blindness.

5.2 Physical Impairment

https://www.ogcio.gov.hk/en/our_work/community/web_mobileapp_accessibility/promulgating_resources/handbook/

Get the User's Attention

- Inverse coloring – inverse coloring
- Blinking Colors – 2-4 blink per second
  - Should blink at 2-4 blinks per second (Hz)
- Audio
  - Soft tones – positive
  - Harsh – emergency
  - Multiple levels are difficult to distinguish, do we like human voices? (interference with user's current communication)
- Choice of fonts, font sizes, colors, markups (e.g. underlining)
- Note:
  - Danger in overusing the above
  - Animation, e.g. show progress

Summary

- Conceptual models help structure design processes
- Design theories, principles and guidelines emerge from practical experience and empirical studies
- Theories can help clarify design implications
- Principles are widely accepted but need updates as technology and applications evolve
- Guidelines accumulate experiences and good practices

References

- Schneiderman et al. (2017) *Designing the User Interface*. Pearson.
- Johnson & Henderson (2012) *Conceptual Models: Core to Good Design.* Morgan and Claypool Publishers
- Preece, Sharp and Rogers (2015) *Interaction Design*. Wiley.