NANYANG TECHNOLOGICAL UNIVERSITY

Interaction and Design Concepts

CZ2004 Human-Computer Interaction

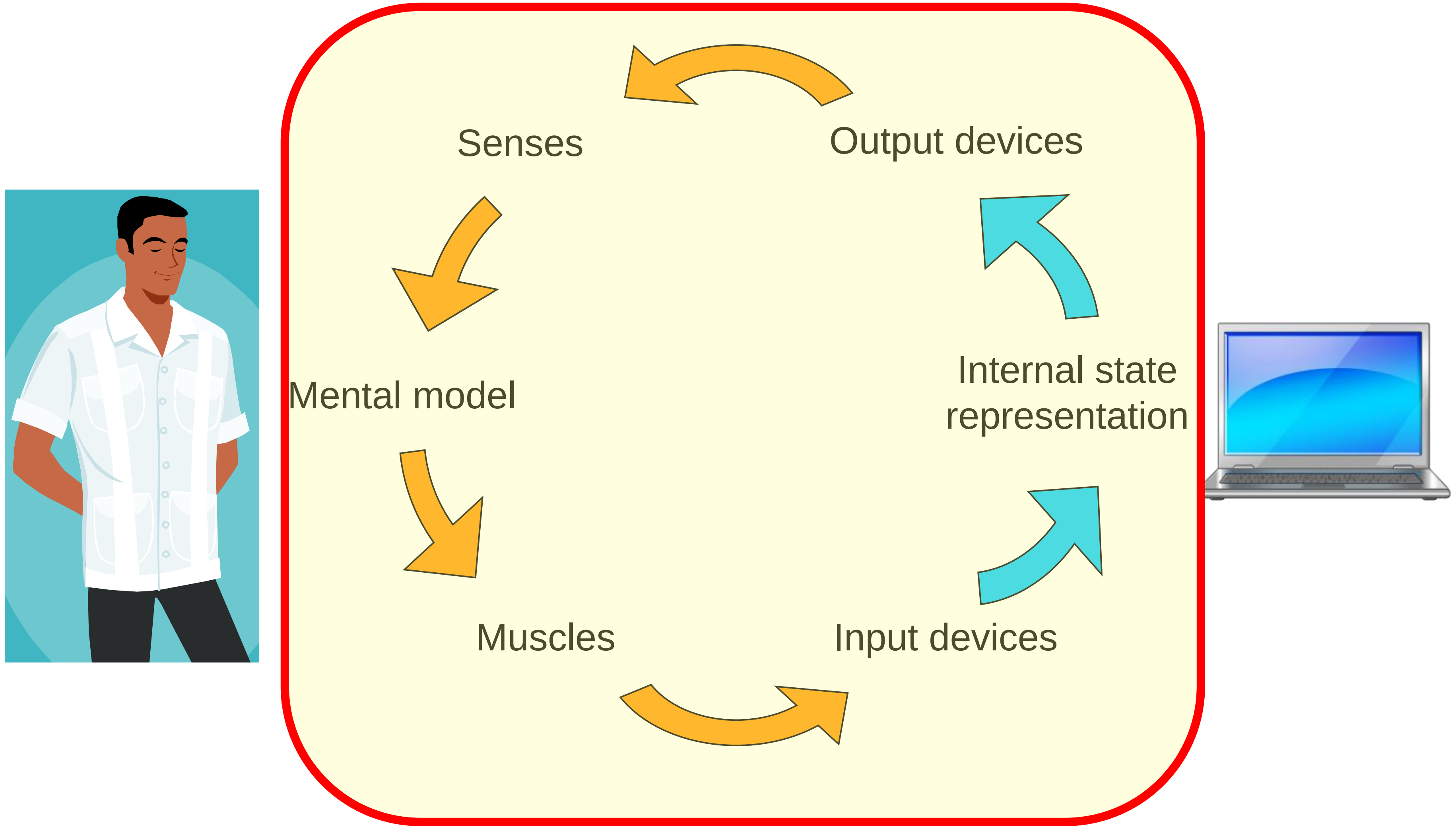
Contents

- Interaction Types
  - Communication, conversation, object manipulation, ego manipulation
  - Peripheral feedback
  - Interactivity and intuitiveness
- Software Behavior
  - Software postures
  - Empathetic and considerate behaviors
  - Anthropomorphism
- Design Concepts
  - Affordances
  - Metaphors
  - Idioms
  - Choice limitation
  - Context awareness
- Design Patterns
  - Templates
  - Widgets and interface builders
  - Design languages
  - Pattern languages

Learning Objectives

- Understand different broad types of interactions, and their interactivity and intuitiveness aspects
  - Able to categorize existing UI's into these types
- Awareness of the different software postures and affective behavior
  - Potential for empathetic and considerate software behavior, and human-like attributes
- Be familiar with various design concepts and terms
  - Such as "affordance", "metaphor", "idiom", etc.
  - Able to analyse existing UI's in terms of these concepts
- Appreciate the use and effectiveness of design patterns
  - Templates, widgets, design and pattern languages
- *Learn the foundations to think systematically when innovating future new UI's, beyond existing ones today*

Interfacing Humans and Computers



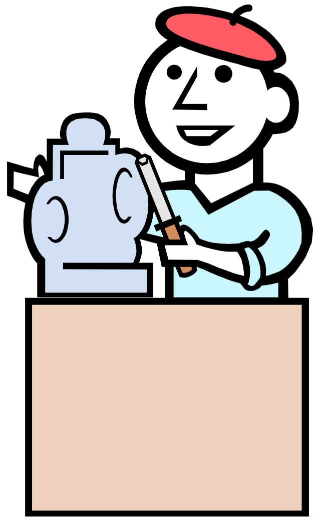
Classification of Interaction Types

Major Interaction Types

- Communication

  The user is giving commands,
  or conversing with the computer

  - Interaction as *instruction*
  - Interaction as *conversation*



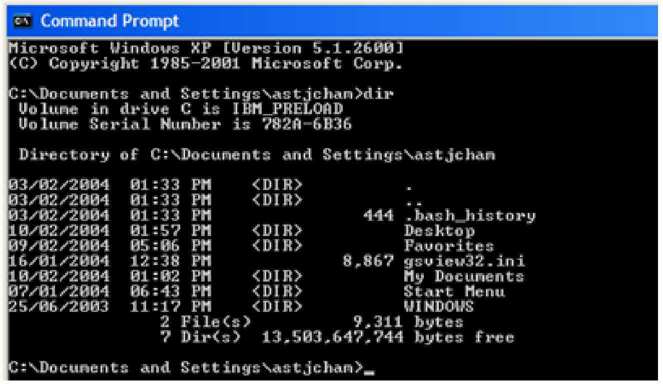
- Manipulation

  The user is manipulating virtual objects,
  or moving virtual self

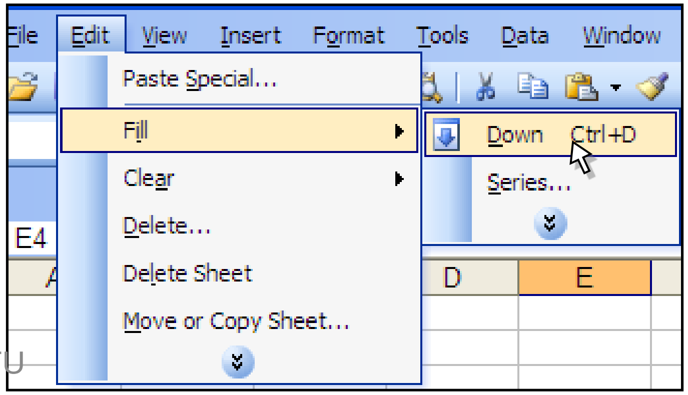
  - Interaction as *object manipulation*
  - Interaction as *ego manipulation*

Interaction as Instruction

- User giving *instructions* to the computer
  - Computer takes action, with terse feedback
- Examples
  - Command-line interfaces
  - Button pushing
  - Speech commands
  - Menu selection
- These differ from object manipulation:
  - Manipulation such as menu navigation does not *physically* relate to intended action
  - Multiple steps to prepare instruction, submitted only at the end
  - Action may not be instantaneous
    - may take a while to process before getting final feedback



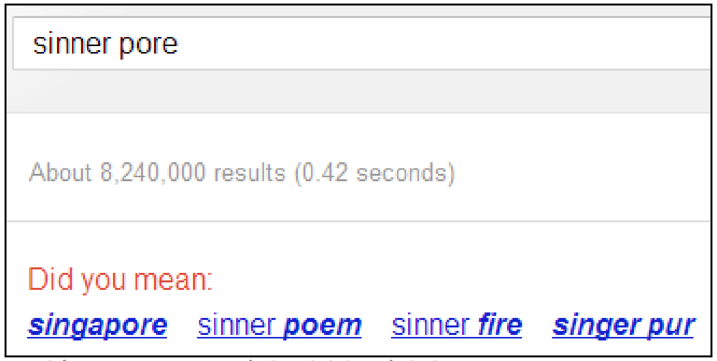
```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\astjcham>dir
 Volume in drive C is IBM_PRELOAD
 Volume Serial Number is 782A-6B36

 Directory of C:\Documents and Settings\astjcham

03/02/2004  01:33 PM    <DIR>          .
03/02/2004  01:33 PM    <DIR>          ..
03/02/2004  01:33 PM               444 .bash_history
10/02/2004  01:57 PM    <DIR>          Desktop
09/02/2004  05:06 PM    <DIR>          Favorites
16/01/2004  12:38 PM             8,867 gsview32.ini
10/02/2004  01:02 PM    <DIR>          My Documents
07/01/2004  06:43 PM    <DIR>          Start Menu
25/06/2003  11:17 PM    <DIR>          WINDOWS
               2 File(s)          9,311 bytes
               7 Dir(s)  13,503,647,744 bytes free

C:\Documents and Settings\astjcham>_
```



Interaction as Conversation

- Involves *turn taking*, like in a dialogue
  - Computer feedback can be substantial
  - Computer may *not* follow instructions directly, instead may
    - request clarification,
    - counter-propose,
    - promise (but not do right away), etc.

- Examples
  - Phone menu systems
  - Interactive fiction (text-based adventure games, e.g. Zork)
  - Google search
  - iPhone Siri

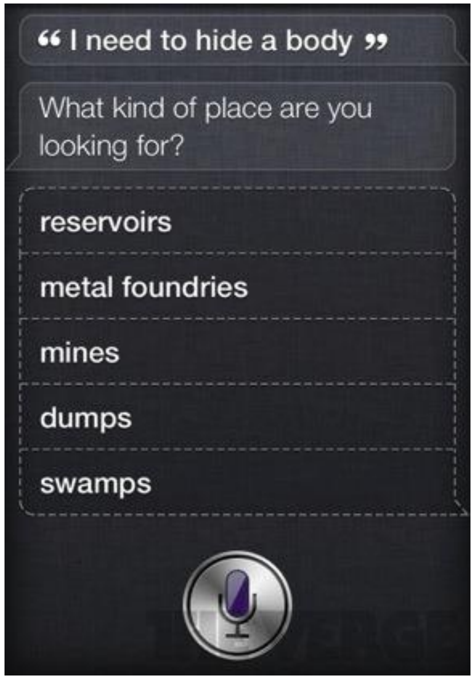


sinner pore

About 8,240,000 results (0.42 seconds)

Did you mean:

singapore sinner poem sinner fire singer pur



“ I need to hide a body ”

What kind of place are you looking for?

reservoirs

metal foundries

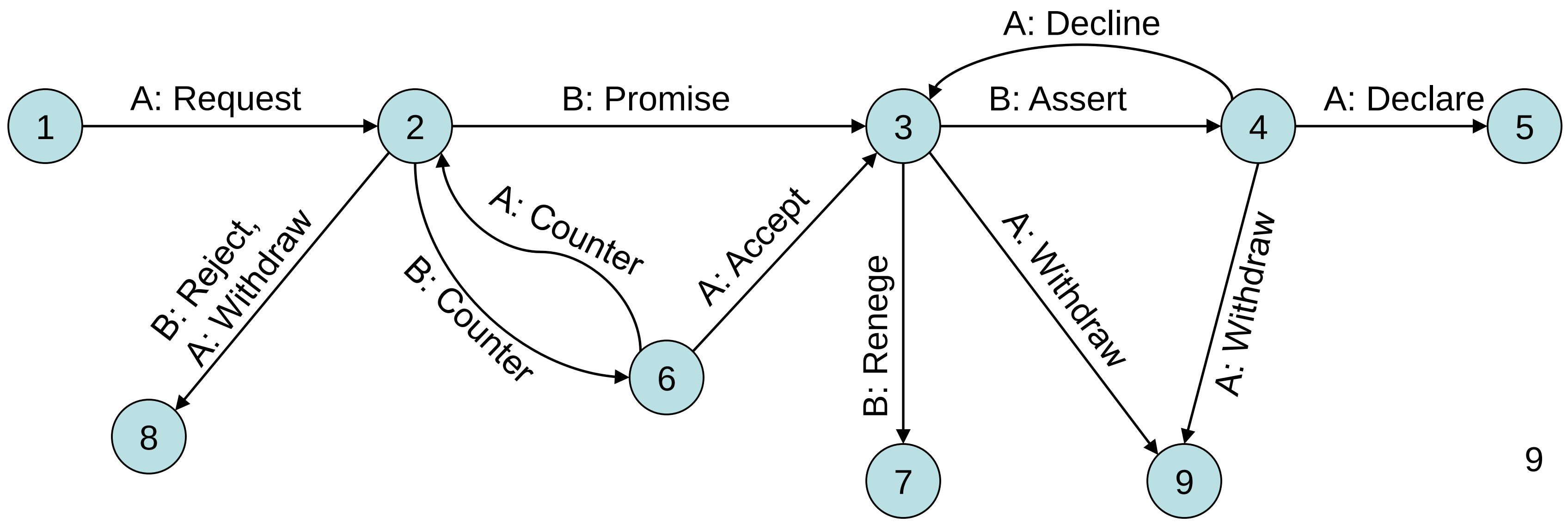
mines

dumps

swamps

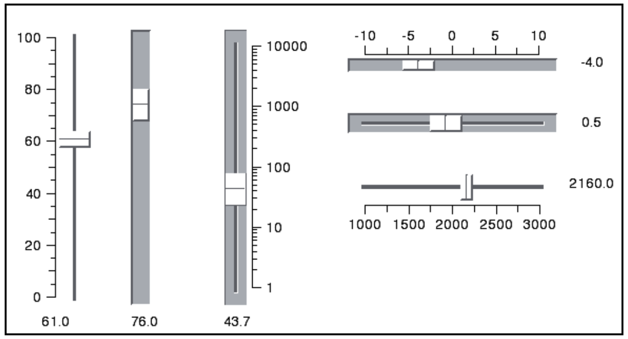
Interaction as Conversation

- Based on Speech Act Theory (Austin 1962, Searle 1975)
  - 5 categories of utterances which are “actions” that:
    - Assert – state an existing truth (“You have homework.”)
    - Direct – request / command (“Can you do the homework?”)
    - Commit – make a promise (“I will do the homework.”)
    - Express – state attitude/emotion (“Sorry I haven’t done it!”)
    - Declare – define a new truth (“You are so grounded!”)
- Terry Winograd’s Conversation for Action Schema (1987)
  - Conversation as a “dance” to complete a task
  - Schema for 2 speakers: A and B



Interaction as Object Manipulation

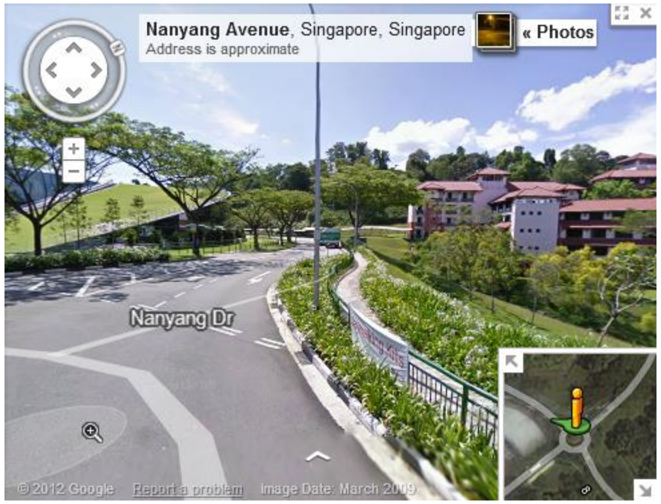
- Users manipulate virtual objects, items or tokens
  - Characterized by continuous, immediate feedback to indicate changing states of the system
  - Typically limited to touch-based manipulation
- Examples
  - Drag and drop icons into folders
  - Two-finger pinch for enlarging images
  - Moving frame slider on a movie player
  - Rotating 3D globe / sliding map in Google Earth



Interaction as Ego Manipulation

- Users move their mental selves around
  - Simple: following hyperlinks
  - Complex: virtually moving 3D avatars

- Examples
  - Exploring Wikipedia via hyperlinks
  - 3D games / virtual worlds, e.g. 2nd Life
  - Flight and car simulators
  - Ground view navigation, e.g. Google Street View



Peripheral Feedback

- Interfaces can provide peripheral feedback
  - Little attention needed
  - Users may not respond, or ignore
- Examples:
  - Modeless Feedback
    - Feedback that does not switch interface mode
    - e.g. status bar information, tooltips
    - Opposite example: focus-grabbing pop-up boxes
  - Augmented Reality (AR)
    - e.g. camera video augmented by labeled info
    - e.g. smart glasses such as Microsoft Hololens (https://youtu.be/SlPs_yxZLSM)
  - Ambient Devices / Calm Computing
    - *Glanceable* output, handled pre-attentively in user's peripheral vision
    - e.g. *Ambient Orb* ~ stock prices, *Power Aware Cord* ~ power being used, *Ambient Umbrella* ~ weather forecast

Top quality gust-buster umbrella canopy design
The magic is in the handle – it glows to indicate when rain or snow is forecast – so you remember to take it with you.

Ambient Umbrella

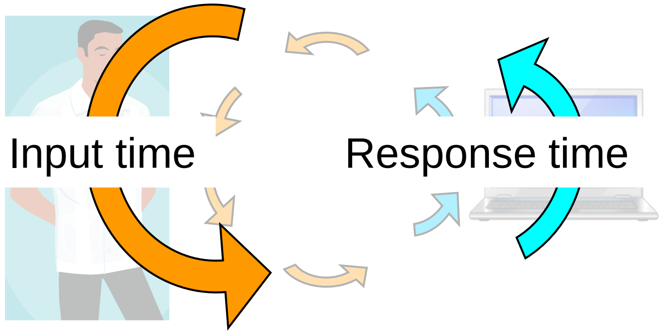
Interaction Attributes

- Various interaction styles can be differentiated by two attributes:

- Interactivity
  - How actively does the interface engage the user?

- Intuitiveness
  - How quickly and easily can a user learn to use the interface or carry out tasks with it?

Interactivity



- Related to rate of human-computer *interaction cycle* and affected by
  - Time it takes for users to input an action
  - Time it takes for the computer to respond
- Communication-based interaction has a slow, turn-taking cycle between users and computers
  - order of seconds (at least)
  - e.g. time taken to type out a command, or navigate a cascade menu, before the computer's turn to act
- Manipulation-based interaction has a very rapid cycle as feedback is continuous
  - fractions of seconds
  - e.g. image is continually resized in a 2-finger pinch interaction, or icon position is continually updated in a drag-and-drop
- Greater interactivity
  - more engaging experience
  - but greater computational cost

Interactivity: Perception of Response Times

- *Communication-based interaction*: users' perception of computer response times are (R.B. Miller, 1968)
  - 0 to 0.1 seconds
    - Considered *instantaneous*
  - 0.1 to 1 second
    - Considered *responsive*
  - 1 to 10 seconds
    - Considered *slow*, wandering attention
  - Greater than 10 seconds
    - Considered *non-interactive*, will switch focus to other tasks (e.g. other windows, coffee)

- *Manipulation-based interaction:* response time also the *latency*
  - Users experience distracting lag from around 170ms and up (www.eurogamer.net/articles/digitalfoundry-lag-factor-article)
  - e.g. Guitar Hero: Aerosmith = 67ms, GTA IV = 133-200ms

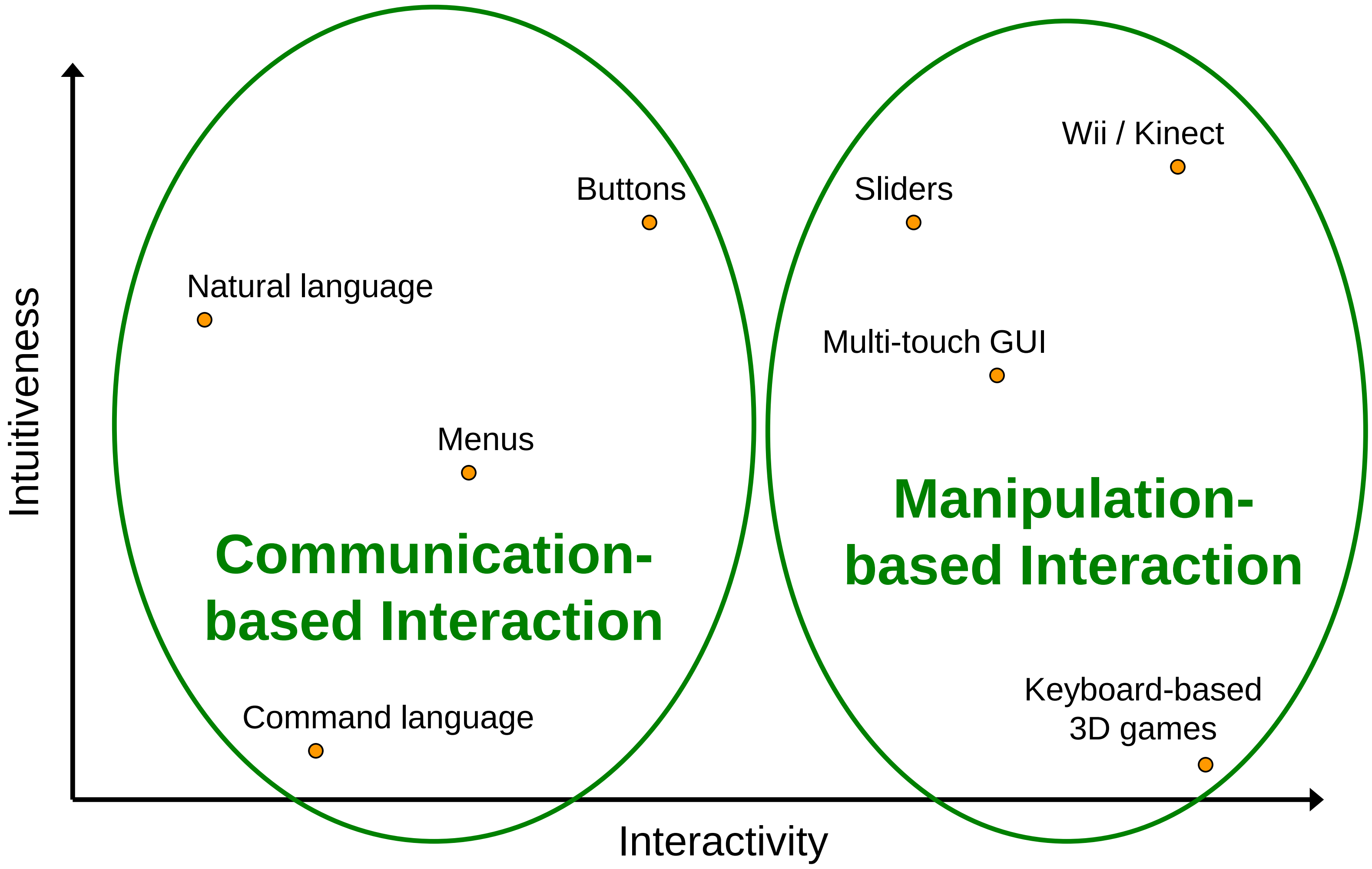
Intuitiveness

- Related to ease of using (or learning to use) an interface
- "Intuitiveness" often about users' *familiarity* with previous software
  - Example: a novice at a first person shooter (FPS) game may find it hard to figure how to move using the keyboard;
  - but another user who has played a different FPS before may find it very easy to pick up
- But should apply for as wide a range of past experiences as possible
  - e.g. tennis with Wiimote is easily picked up by many diverse users, vs gamepad controls
    - More people are familiar with swinging a stick to hit something, than using a gamepad

CC-BY-NC Renee Ya, flickr

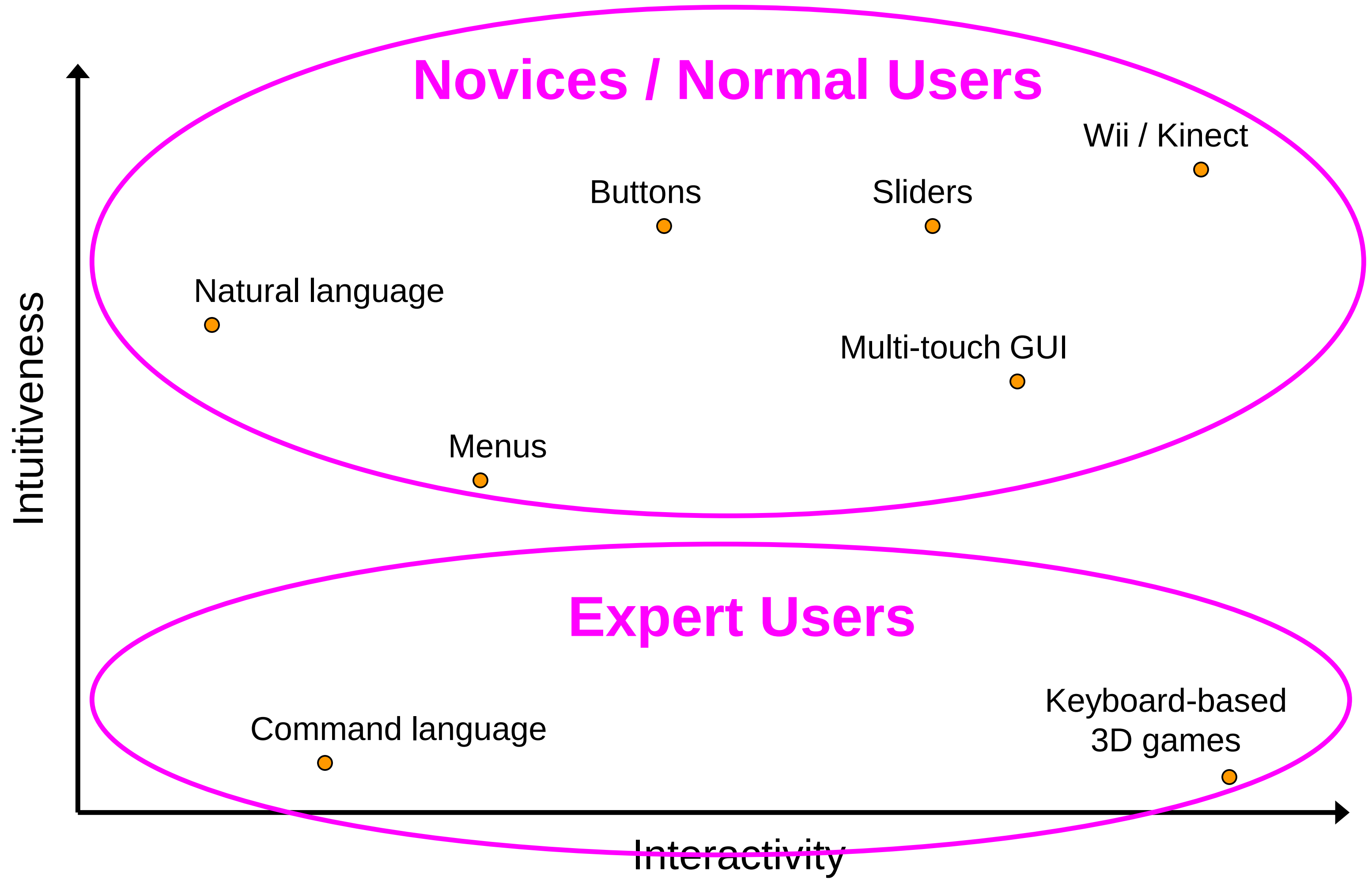
Intuitiveness-Interactivity Chart

- We can plot various interactive forms into a 2D chart comparing intuitiveness and interactivity (D. Frohlich 1993, E. Hutchins et al 1985)



Intuitiveness-Interactivity Chart

- We can plot various interactive forms into a 2D chart comparing intuitiveness and interactivity (D. Frohlich 1993, E. Hutchins et al 1985)

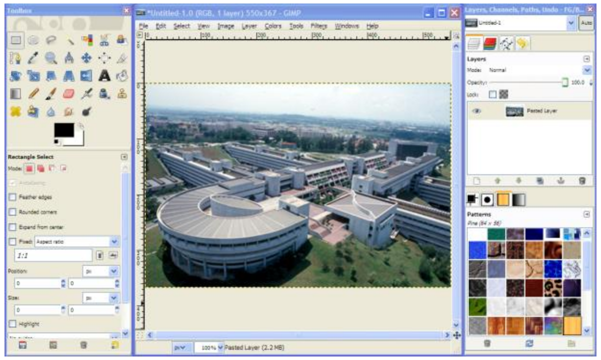


Software Behavior

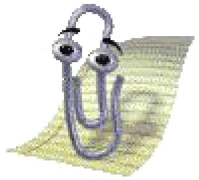
Software Posture

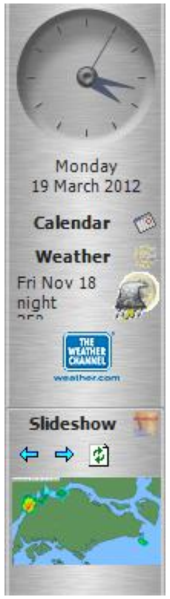
- Software can adopt a range of *postures*

- Sovereign
  - Monopolize user's attention for extended duration
  - e.g. most major applications

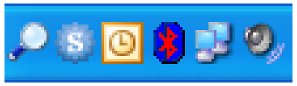


- Transient
  - Briefly capture user's attention from time to time
  - e.g. sidebar gadgets, chat notification, Clippit assistant (old)





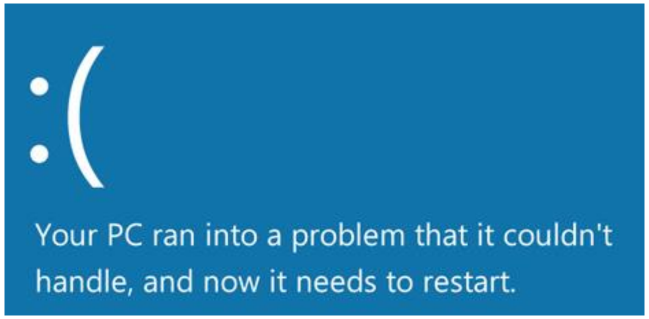
- Daemonic
  - Mostly silent / invisible, very rarely capturing user's attention
  - e.g. network and volume icons in system tray



Empathetic Software Behavior

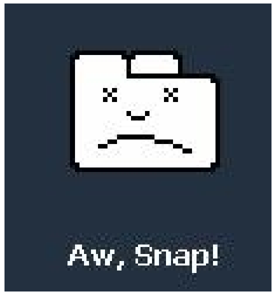
- 2 common negative affects in user interaction
  - Frustration / anger
    - Due to “excise”: unnecessary effort or difficulty in using interface
  - Fear / distress
    - Due to having done a wrong but irreversible action
- Best solution is to have good design to minimize such affects!
- However, if unavoidable, can still try mitigation by empathy…
- Examples
  - Groupon unsubscribe: http://www.youtube.com/watch?v=wMzZvK69QaQ
  - Empathetic error messages

Win 8 BSOD


:(

Your PC ran into a problem that it couldn't handle, and now it needs to restart.

Chrome Error


Aw, Snap!

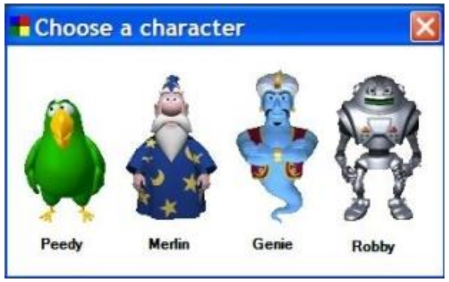
Considerate Software Behavior

- Better to design considerate software behavior in the first place
  - *How would a good host behave with a guest / customer?*
- Proactive
  - *Know user's habits*, e.g. auto-fill, auto-complete
  - *Anticipate needs*, e.g. preload linked pages
- Flexible
  - *Easily reversible*, e.g. don't keep asking user to confirm file deletion, just be able to undelete easily
  - *Don't force the user*, e.g. allow incomplete forms
  - *Adaptive*, e.g. ".com" key (iPhone Safari) will add either ".com" or "com" depending on last symbol
  - *Fail gracefully*, e.g. auto-recover "unsaved" info
- Deferential
  - Avoid informing user of non-critical internal problems (i.e. unnecessary error messages)
  - Avoid asking unnecessary questions

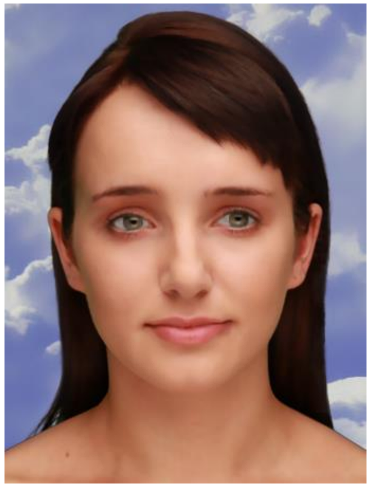
aceattorney.wikia.com/wiki/Bellboy

Anthropomorphism

- Giving human attributes to software
- Suitable for some groups of users and some application types
  - Kids, games, spoken interaction, etc.
- Chatbots
  - Using natural language processing (NLP)
  - e.g. ELIZA, Jabberwacky, Cleverbot, Siri
- Avatars
  - Characters with visual appearance and voice
  - e.g. LivingActor Presenter, MS Agent (old)
- Combined
  - e.g. Cleverbot on www.existor.com, Guile3D Denise
- Consider:
  - *Gender,* e.g. choice of gender for Siri
  - *Personality*, e.g. may be designed using OCEAN traits

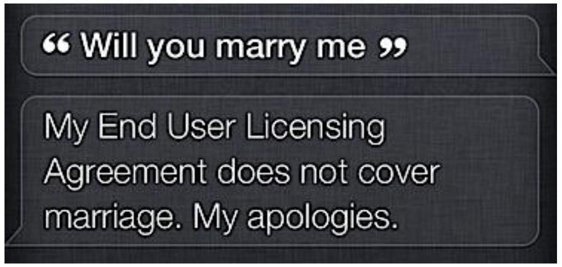


MS Agents



www.existor.com

LivingActor Presenter



“Will you marry me”

My End User Licensing Agreement does not cover marriage. My apologies.

Design Concepts

HCI Design Concepts

- Some widely-used HCI design concepts
  - Affordances
  - Metaphors
  - Idioms
  - Choice limitation
  - Context awareness

- Can be for both design and analysis

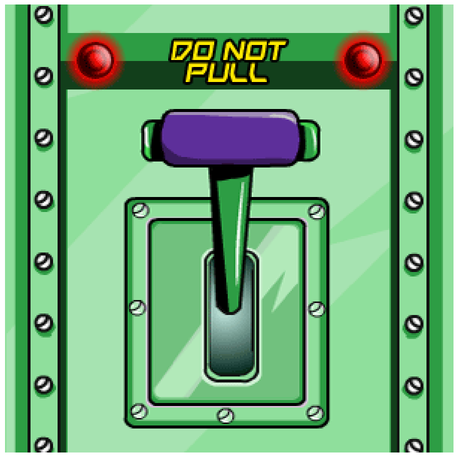
- Different from design principles
  - These concepts don’t *have* to be applied
  - They are just part of the palette of ideas that can be tried out

Affordances

- Affordance: attribute of an item, object or structure, permitting a user to perform some basic action, e.g.
  - button=push, rope=pull, switch=flip, slider=slide, wheel=spin, ladder/stairs=climb, handle=grasp

- Well-designed items have clearly perceived affordance ("good affordance")
  - Users know *instinctively* how to use them,
  - or even *enticed* to use them

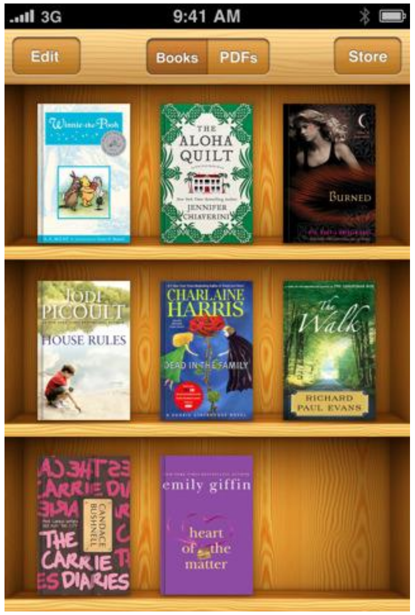
- Affordance does not define effect
  - e.g. affordance of a lever is only pulling, not what happens after



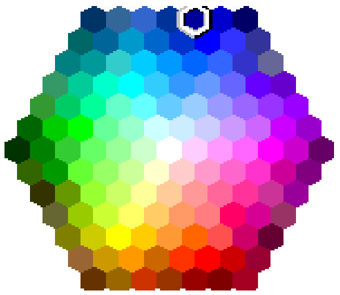
Metaphors

- Computer interactions mapping to real-world interactions that users are more familiar with

- Examples
  - GUI desktop ➔ normal desktop where you pile and arrange paper documents
  - Drag-and-drop of file icons into folder icons ➔ move paper documents into real folders (or bin)
  - Email ➔ Snail mail (hence the term “mailbox”)
  - Order list ➔ shopping cart
  - Speech-based agents ➔ talking to a real person
  - eBook next page ➔ physical turning of page
    - See KAIST system www.youtube.com/watch?v=rVyBwz1-AiE
  - eBook collection ➔ real bookshelf
  - 3D virtual environment ➔ real environment

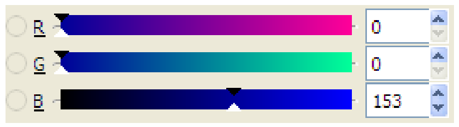


Metaphors



⇔

- Advantages
  - Good metaphors reduce gulfs of execution and evaluation
    - e.g. selecting color via palette, not RGB values

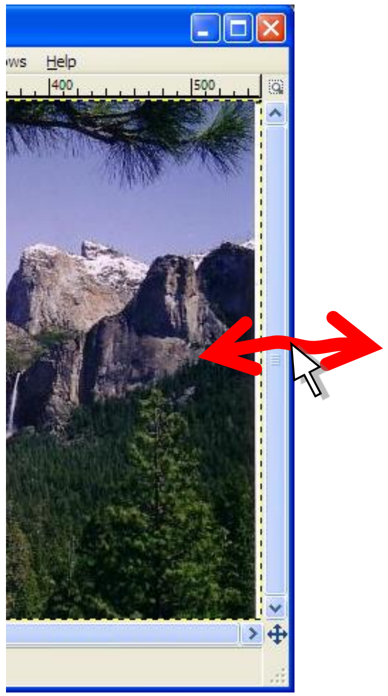


  - Users understand better what can or cannot be done
    - Able to guess available actions not yet tried out

- Disadvantages
  - Bad metaphors can confuse users
    - Worse than no metaphor
    - Even when users know, may still feel uncomfortable
  - Example:
    - Mac OS: eject removable media by dragging disk icon to trashcan icon
    - In real world, eject floppy disk ≠ discard floppy disk!

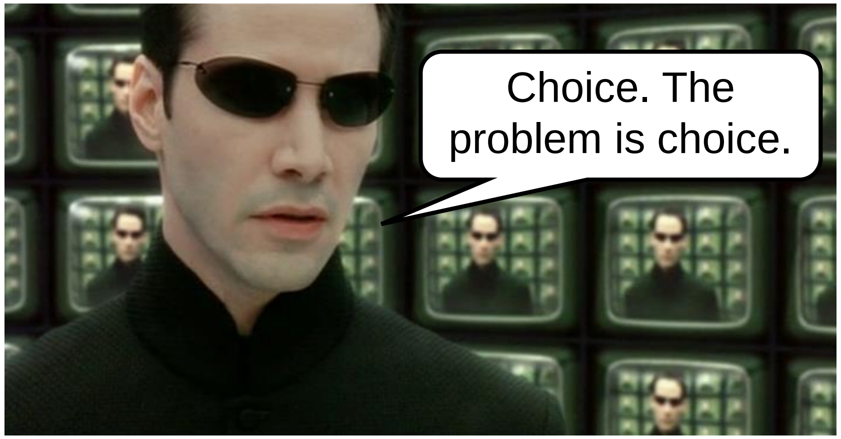
Idioms

- English: commonly-used figurative phrases where
  - implied meaning not the same as literal meaning
  - e.g. “raining cats and dogs”, “a piece of cake”, “pulling your leg”
  - Implied meaning comes from wide cultural usage
- HCI: very widely-used actions, span different platforms / software
  - often not a metaphor (i.e. no physical meaning)
  - feel “natural” and quickly become “familiar”
- Examples of HCI idioms
  - Resize windows by dragging on borders
  - Scroll window views via scroll bars
  - Two-finger pinch to resize map
  - Cascading menus / nested folders
  - Esc / Ctrl-Alt-Del keys to “get back”
  - WASD keys in 3D first-person shooters
- *NOTE: Most GUI elements are idiomatic, not metaphoric*



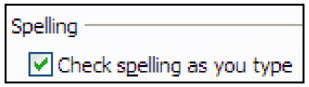
Choice Limitation

- Keyboard and pointer actions are very unconstrained
  - How would a novice user know what to do to solve her task?
- One way: quickly get users to learn idiomatic actions
- Another way: provide noticeable constraints to lead users towards allowable actions
  - Narrower range of actions, e.g. short textboxes or sliders, or
  - List of discrete choices, e.g. menus, checkboxes
- Think of your past experience with exam question types
  - Essays
  - Single sentences
  - Fill-in-the-blanks / Cloze
  - Multiple-choice questions
  - True/False questions



Choice Limitation

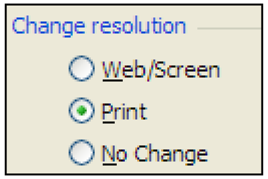
Binary choices, e.g. checkboxes



Spelling

[✓] Check spelling as you type

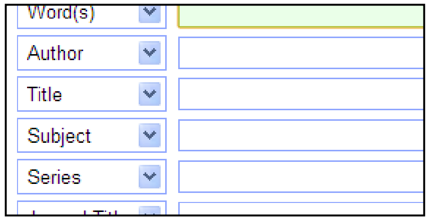
Multiple choices, e.g. radio buttons



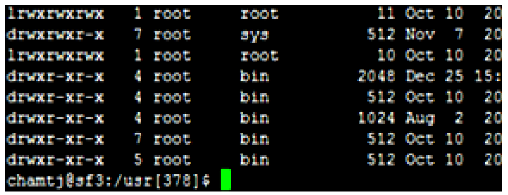
Change resolution

Web/Screen [✓ Print] No Change

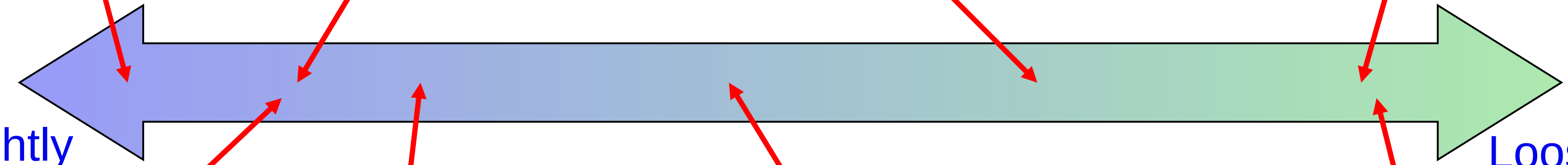
Text boxes



Command line

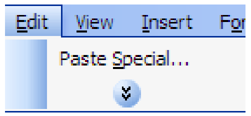


```
lrwxrwxrwx   1 root     root          11 Oct 10  20
drwxrwxr-x   7 root     sys          512 Nov  7  20
lrwxrwxrwx   1 root     root          10 Oct 10  20
drwxr-xr-x   4 root     bin         2048 Dec 25 15:
drwxr-xr-x   4 root     bin          512 Oct 10  20
drwxr-xr-x   4 root     bin         1024 Aug  2  20
drwxr-xr-x   7 root     bin          512 Oct 10  20
drwxr-xr-x   5 root     bin          512 Oct 10  20
chamtj@sf3:/usr[378]$ 
```

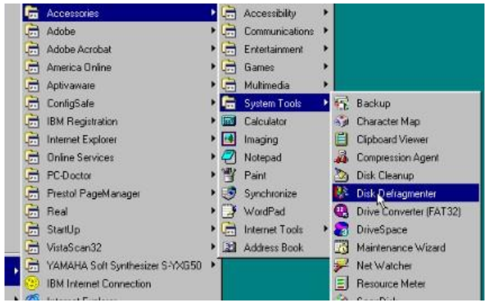


Tightly Limited

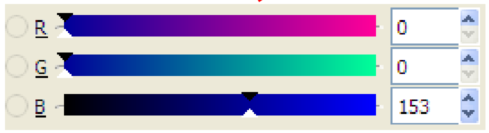
Loose or Unlimited



Smart Partial Menus



Menus

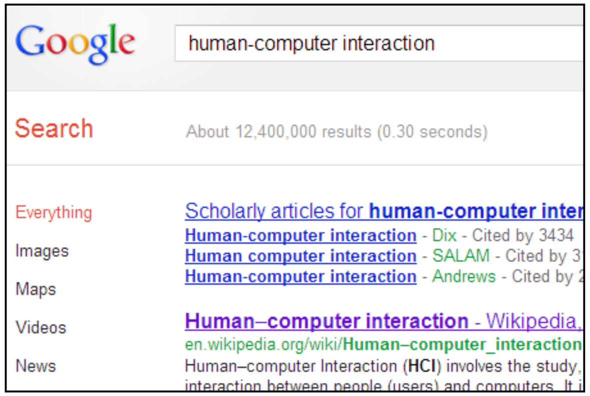


R 0

G 0

B 153

Sliders



Google human-computer interaction

Search About 12,400,000 results (0.30 seconds)

Everything

Images

Maps

Videos

News

Scholarly articles for **human-computer inter**

Human-computer interaction - Dix - Cited by 3434

Human computer interaction - SALAM - Cited by 3

Human-computer interaction - Andrews - Cited by 2

Human–computer interaction - Wikipedia,

en.wikipedia.org/wiki/**Human–computer_interaction**

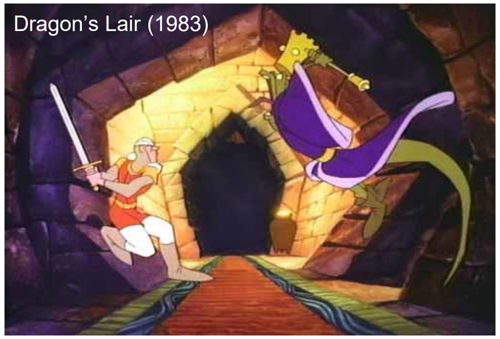
Human–computer Interaction (**HCI**) involves the study,

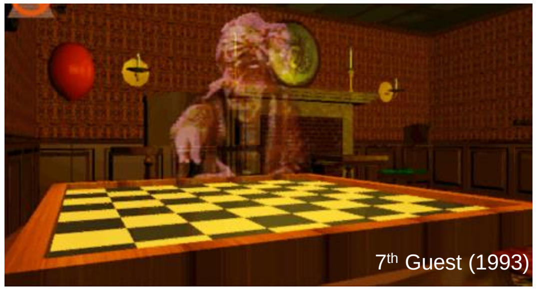
interaction between people (users) and computers. It i

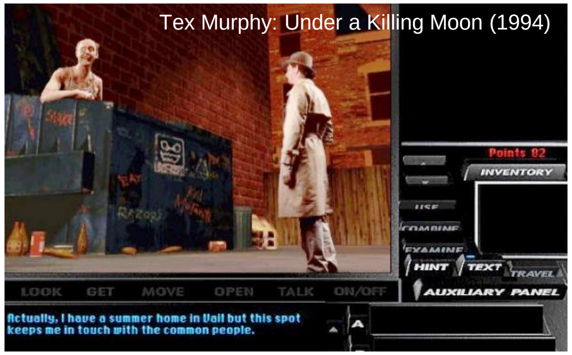
Natural language

Choice Limitation in Games

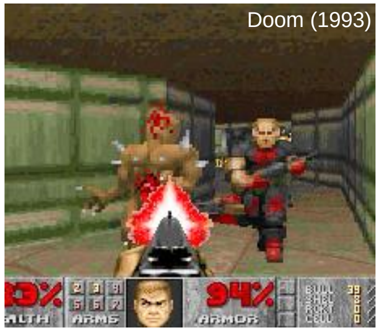
- Evolution of immersive video games
- Original: *Dragon's Lair*, 1983
  - 1st with immersive high-quality graphics
  - Big change: arcade games were then in top-down view, *Space Invaders* still popular!
  - Gameplay: "choices" made during short intervals
    - Each choice: 1 of 4 directions or 1 action button
    - Hand-drawn animations between choices
    - Complete game involves just 200 choices
- Interactive Movie genre, 1990's – early 2000's
  - Point-and-click on many screen hotspots
  - Actual video footage (blue/green screened)
  - e.g. *7th Guest*, *Tex Murphy* series
- RPG / FPS, 1990's onwards
  - Free movement control of 3D player
    - like "infinite" choices…
  - Poor graphics initially (*Doom* = 320x200 resolution)
  - Later: better graphics quality + much greater interactivity

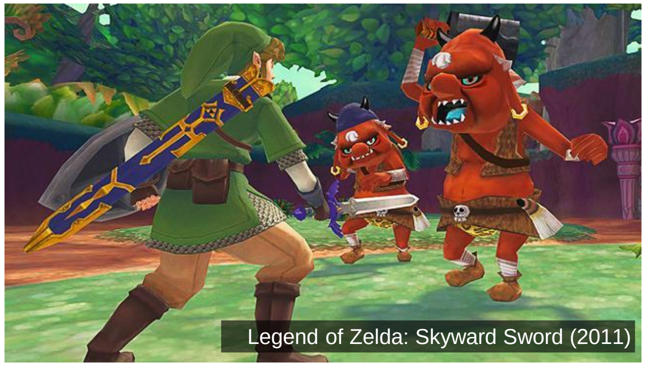

Dragon's Lair (1983)


7th Guest (1993)


Tex Murphy: Under a Killing Moon (1994)

Actually, I have a summer home in Vail but this spot keeps me in touch with the common people.


Doom (1993)


Legend of Zelda: Skyward Sword (2011)

Context Awareness

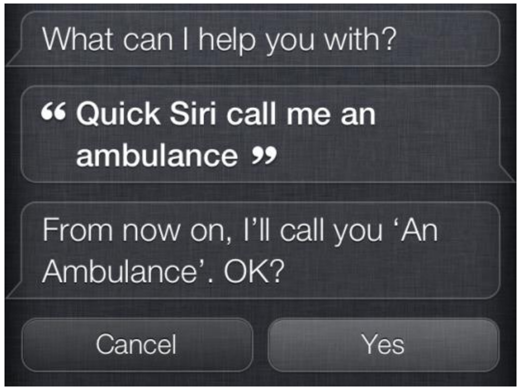
- Choice limitation may not be enough
  - when identical input mean different things
- May be able to use **context** to help distinguish
- Examples

Ambiguous search terms

- "NTU" – Nanyang Technological University or National Taiwan University, etc.?
- Geographical location of user is the context
- **Google / Bing Search** knows user's location ➔ rank results accordingly

3D modeling with 2D GUI

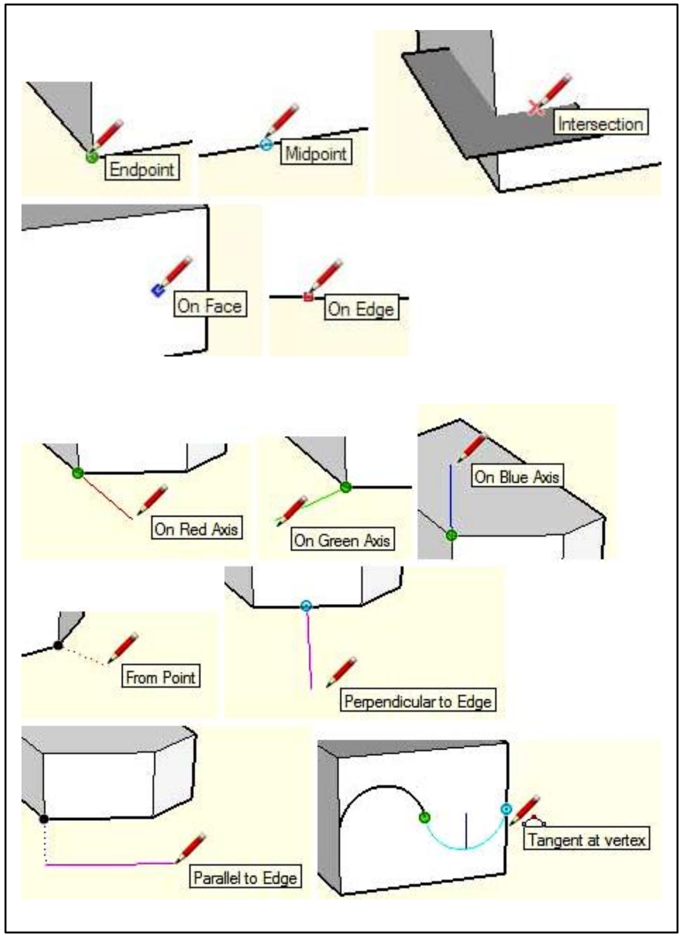
- 3D drawing with 2D I/O is hard: depth unknown
- Existing geometry is the context
- ~~Google~~ **Trimble SketchUp** uses an *inference engine*, tries to figure out whether user is:
  - clicking on edges / mid-points of existing lines
  - drawing 3D line parallel to an axis or other lines, etc.
  - Details: help.sketchup.com/en/article/70143



What can I help you with?

"Quick Siri call me an ambulance"

From now on, I'll call you 'An Ambulance'. OK?



Design Patterns

Design Patterns

- Design patterns: reusable past solutions to design problems
  - Not just visuals, but also the core ideas / techniques
- Why consider?
  1. Difficult to design UI from scratch, even with design principles
     - e.g. “reduce short term memory” principle doesn’t tell the designer how to begin!
  2. There is organic, collective knowledge, of what works and what doesn’t, from past experiences by other designers
  3. Users are already very familiar with some interface idioms

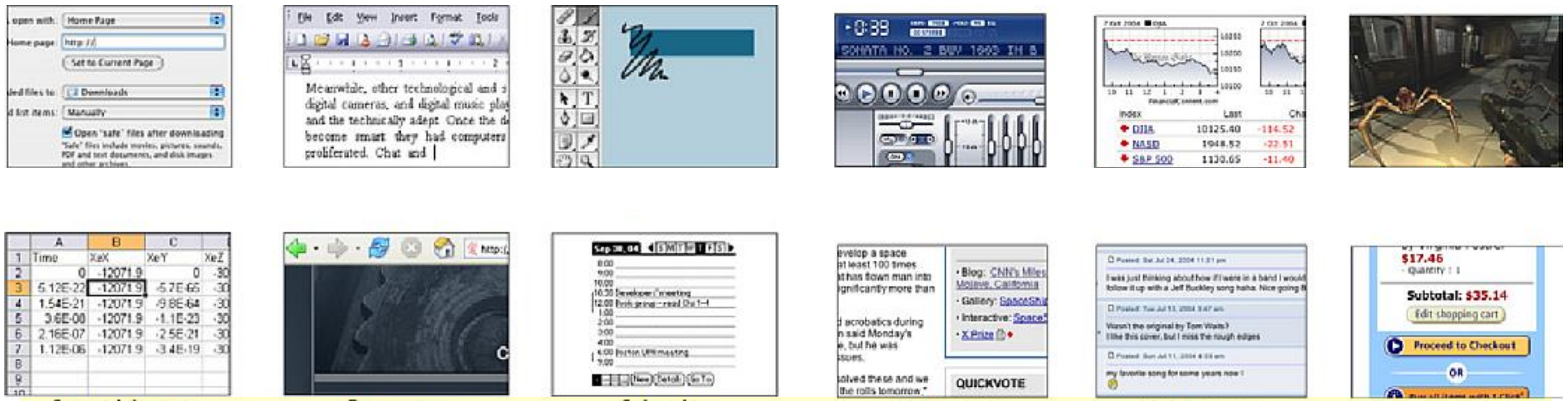


Figure from *Designing Interfaces*, 2nd edition, by Jenifer Tidwell, O’Reilly publishers, 2011

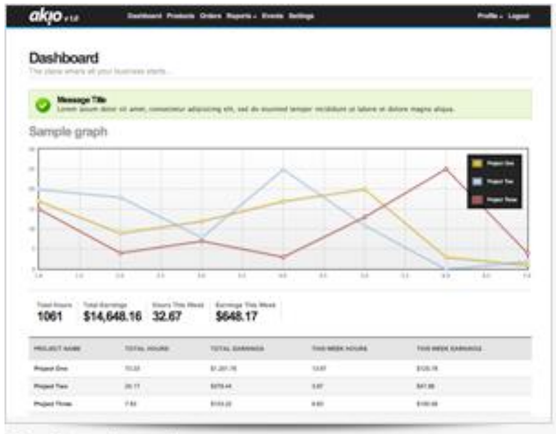
Types of Design Patterns

- Templates
  - Near-complete design prototypes
  - e.g. website templates
- Widgets / controls
  - Basic high-level GUI components
  - e.g. buttons and sliders
- Design languages and guidelines
  - Prescribe expected look and feel of application UI’s
  - e.g. Apple Macintosh Human Interface Guidelines
- Pattern languages
  - Structured way of guiding non-expert designers by decomposing design problems to different levels of detail

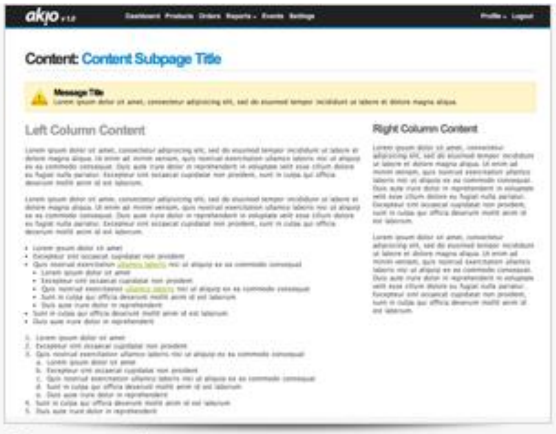
Design Templates

- Near complete design prototypes
  - The most restrictive form of design pattern
  - Easiest to be used directly, hardest to adapt to different situations

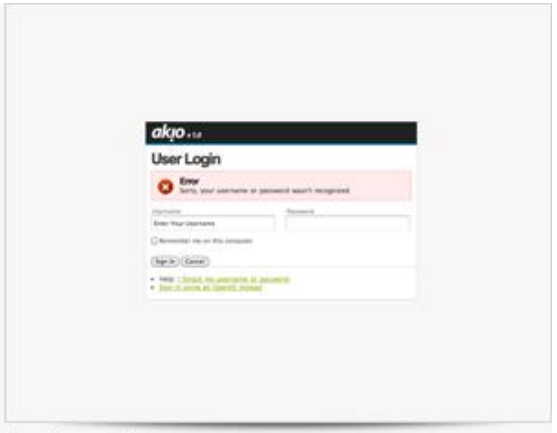
- Examples
  - PowerPoint themes
  - Website templates, e.g.
    - Google Sites templates targeting different uses
    - AKIO site templates for dashboard, content, login, etc.



Dashboard template



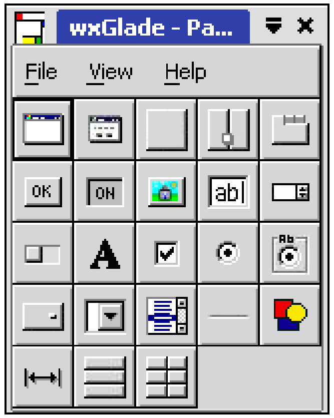
Content template



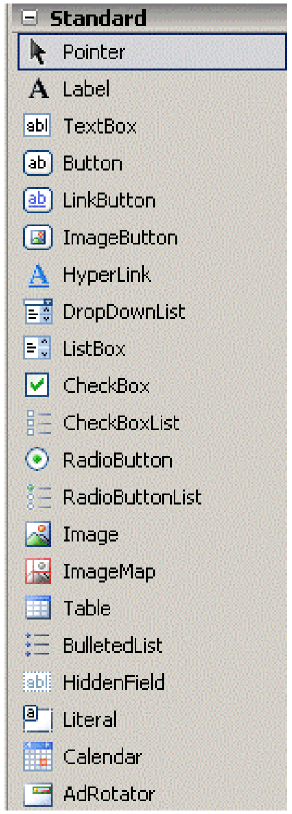
Login template

Widgets and Controls

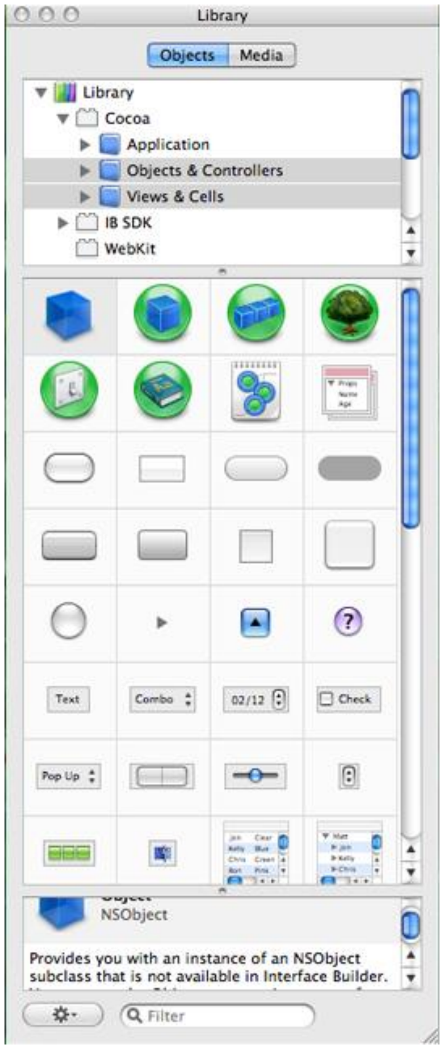
- Well-established idiomatic GUI components
- Widget library
  - Buttons, sliders, textboxes, radio buttons, checkboxes, etc.
- Allow designers to think with a higher-level vocabulary of widgets
  - rather than in terms of pixels and/or low-level input



wxGlade widget palette



MS Visual Studio Designer controls toolbox

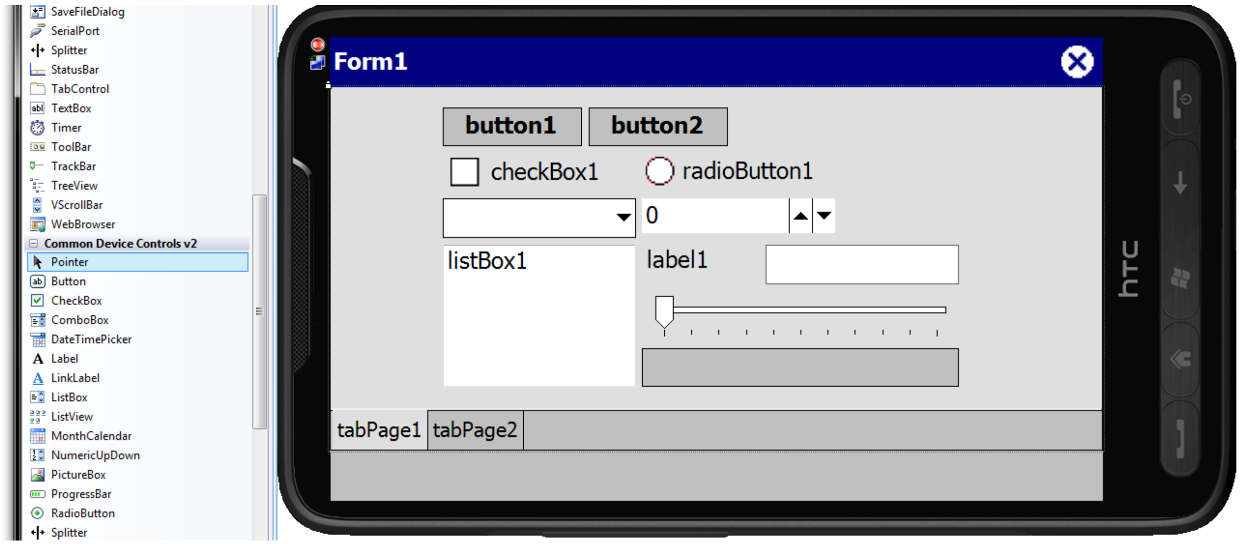


Provides you with an instance of an NSObject subclass that is not available in Interface Builder.

Apple Interface Builder UI objects palette

Graphical User Interface Builders

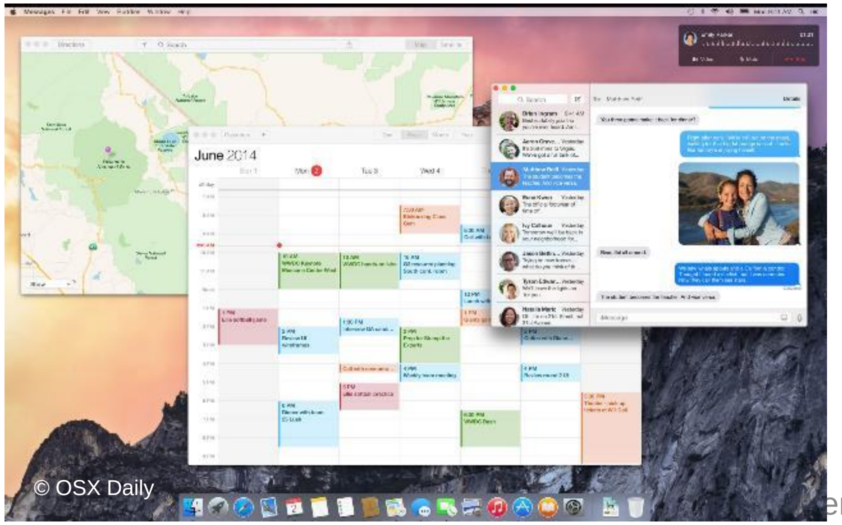
- Software to visually design user interfaces
  - Using widgets from a palette or library
- Able to generate some automatic code
- But doesn't enforce how widgets are combined together into a *good* interface
- Examples
  - Apple *Interface Builder*, GTK+ *Glade*,
    MS Visual Studio *Windows Forms Designer*

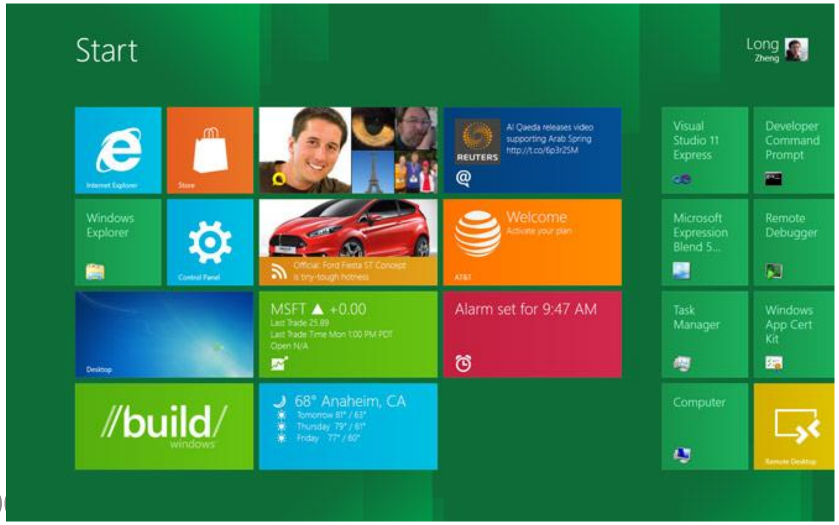


Screenshot extracted from forum.xda-developers.com/showthread.php?p=8045502

Design Languages and Guidelines

- "Language" term used loosely
  - Scheme to define / design a consistent look and feel for UI's
  - Define terminology: e.g. "charms" in Win8, "Dock" in OS X.
- Guidelines determine the design language
  - Prescribe expected look and feel of application UI's
    - e.g. fonts: OS X – (Mavericks) Lucida Grande, (Yosemite) *Helvetica Neue*; Android – *Roboto*, Win Vista onwards – *Segoe UI*
  - Recommend when and how widgets should be used and combined
- Examples
  - Apple *OS X Yosemite Human Interface Guidelines*
    https://developer.apple.com/library/mac/documentation/UserExperience/Conceptual/OSXHIGuidelines/index.html
  - Android *User Interface Guidelines*
    https://developer.android.com/design/style/index.html
  - MS design language for Windows 8 (formerly called "Metro")
    http://msdn.microsoft.com/library/windows/apps/hh465424.aspx





Pattern Language

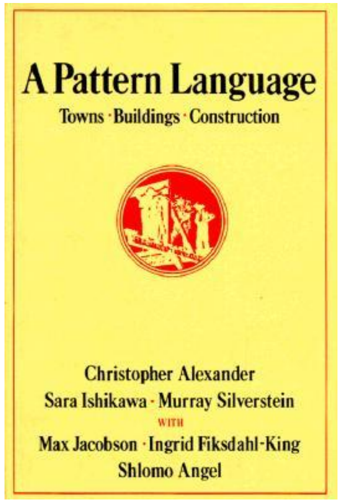
- Integrated set of “patterns” to solve complex design problems
  - High-level patterns “unfold” into lower-level patterns for sub-problems

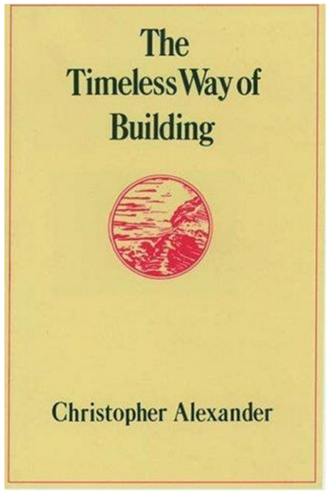
- Each pattern usually presented in sections:
  - **Context**
    - Where is the problem encountered?
  - **Problem**
    - What is the problem?
    - Are there similar related problems?
  - **Solution** (with examples)
    - Set of solution steps and key considerations
    - Dependencies to other patterns to solve sub-problems

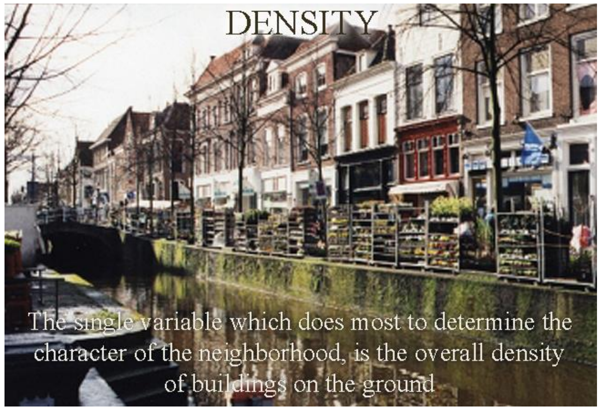
- Main benefit: *non-experts can use this to generate good design!*
  - Just need to “unfold” each pattern into choices of smaller patterns
  - Well-tested ideas about how to choose different patterns
  - *Won’t miss out important steps and considerations*

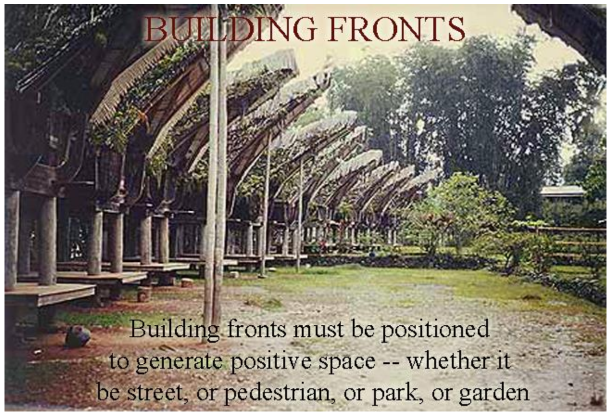
Pattern Language

- Pattern language first proposed by architect Christopher Alexander
  - empowers laypeople to plan and design living space at *any* scale
  - Complete pattern language
    - Deals with "towns" and "cities" …
      … to "chairs" and "ornaments"!
    - *A Pattern Language* and
      The Timeless Way of Building (1977-79)
    - See www.patternlanguage.com, www.livingneighborhoods.org


A Pattern Language
Towns · Buildings · Construction
Christopher Alexander
Sara Ishikawa · Murray Silverstein
WITH
Max Jacobson · Ingrid Fiksdahl-King
Shlomo Angel

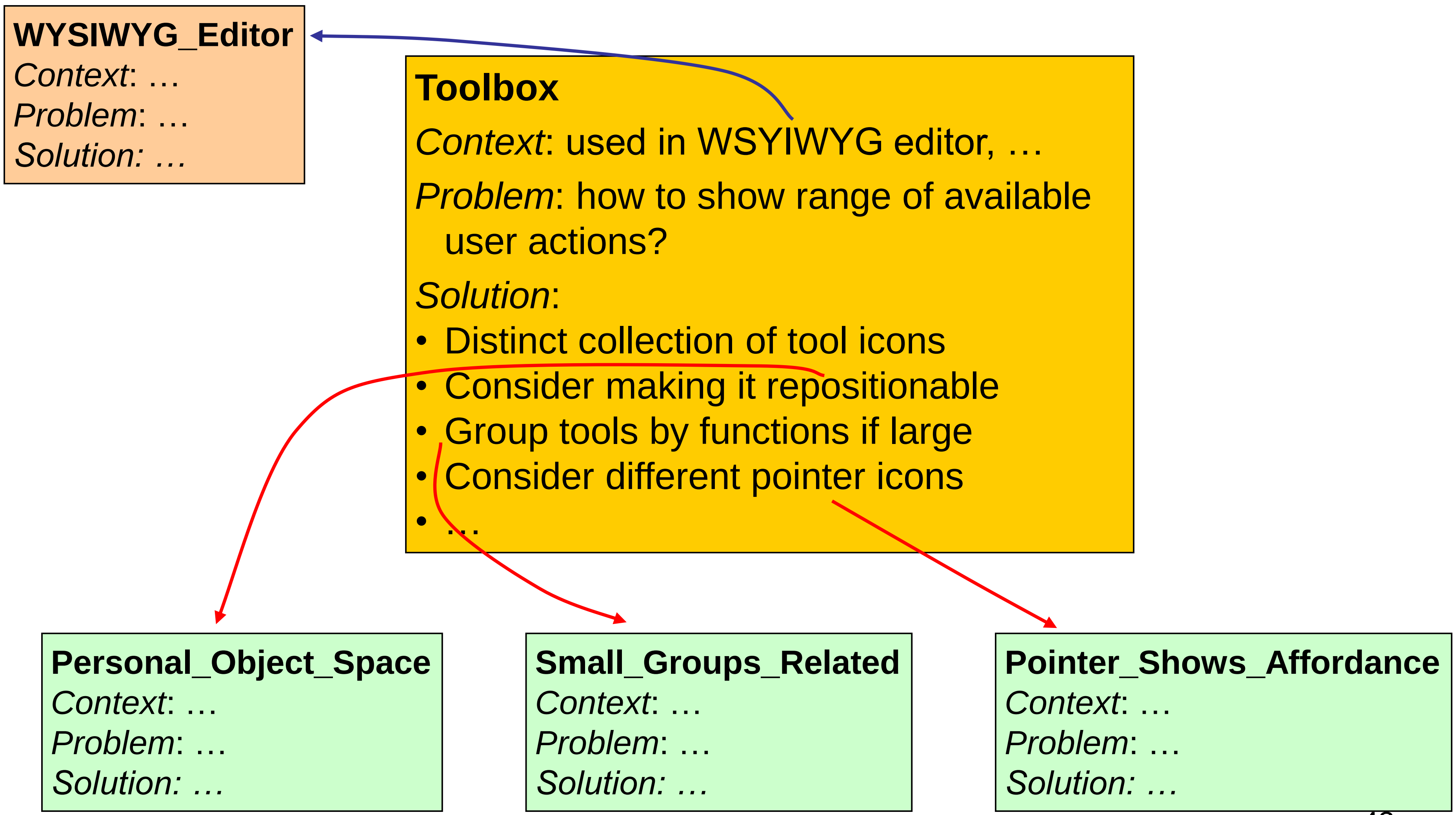



DENSITY
The single variable which does most to determine the character of the neighborhood, is the overall density of buildings on the ground


BUILDING FRONTS
Building fronts must be positioned to generate positive space -- whether it be street, or pedestrian, or park, or garden

Pattern Language

- Example from Jenifer Tidwell's *Common Ground* pattern language (www.mit.edu/~jtidwell/common_ground.html)



WYSIWYG_Editor
Context: …
Problem: …
Solution: …

Toolbox
Context: used in WSYIWYG editor, …
Problem: how to show range of available user actions?
Solution:
- Distinct collection of tool icons
- Consider making it repositionable
- Group tools by functions if large
- Consider different pointer icons
- …

Personal_Object_Space
Context: …
Problem: …
Solution: …

Small_Groups_Related
Context: …
Problem: …
Solution: …

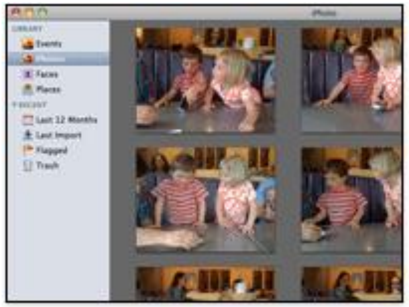
Pointer_Shows_Affordance
Context: …
Problem: …
Solution: …

Pattern Language

- Pattern languages now widely used in software engineering and user interaction design

- Some user interaction pattern languages
  - Jenifer Tidwell’s *Designing Interfaces* book, See designinginterfaces.com
    - based on her earlier patterns at time-tripper.com/uipatterns/ and www.mit.edu/~jtidwell/common_ground.html
  - Martin Van Welie’s interaction pattern library at welie.com
    - Quick case study of “Shopping” pattern:

      www.welie.com/patterns/showPattern.php?patternID=shopping
  - Windows 7 / Vista *User Experience Interaction Guidelines* is written in pattern language form
  - List of different pattern languages www.cs.kent.ac.uk/people/staff/saf/patterns/gallery.html



Picture Manager

News Stream

Wizard