Human Computer Interaction

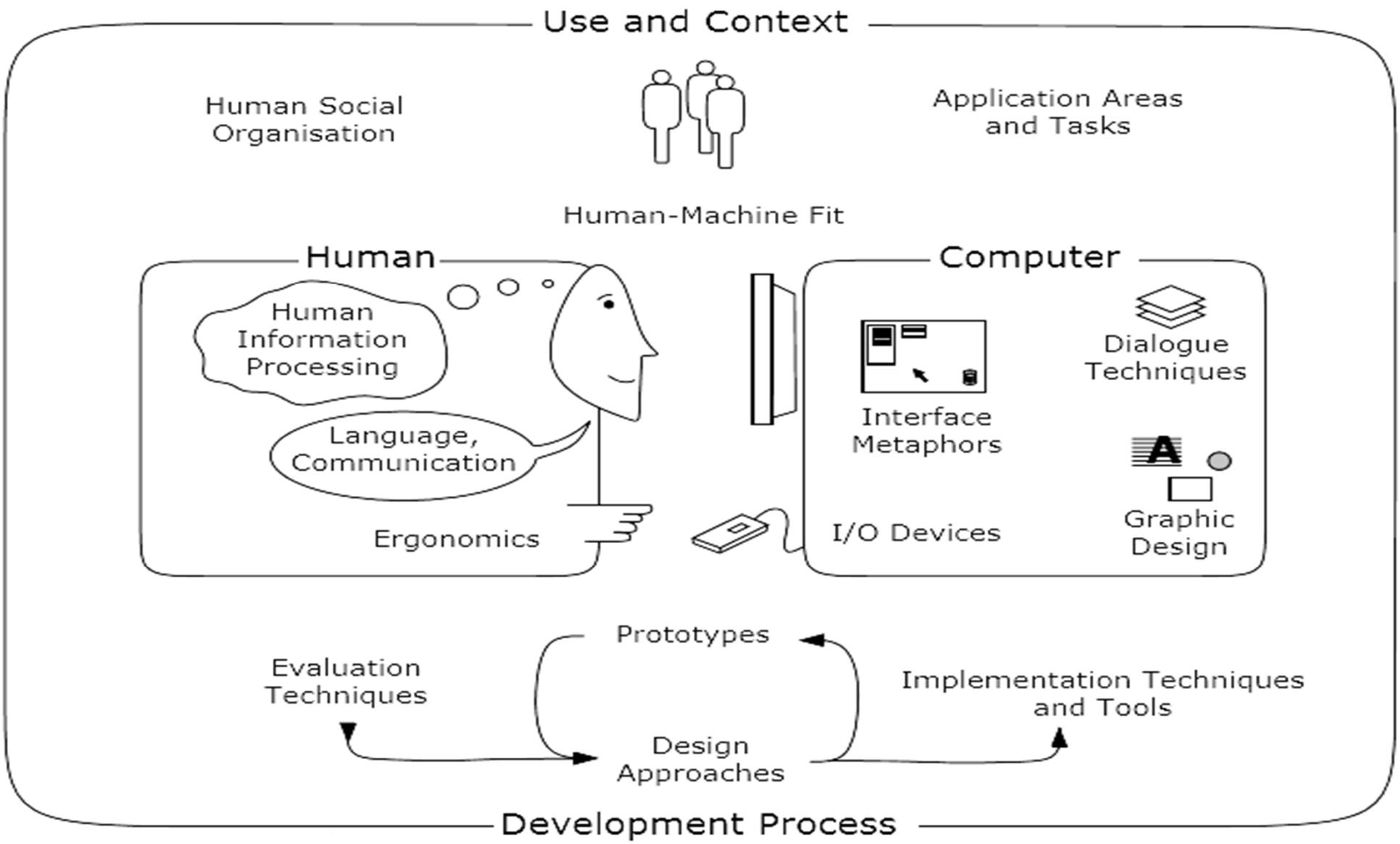
INTERACTION PRINCIPLES FOR DESIGN

By: Nguyễn Công Hoan

Reference

- DonaldNorman, **The Design of Everyday Things, MITPress, 23 Dec 2013**
- **Tutorial Teaching** of Prof. Dr. Keith Andrews, Graz University of Technology

Content



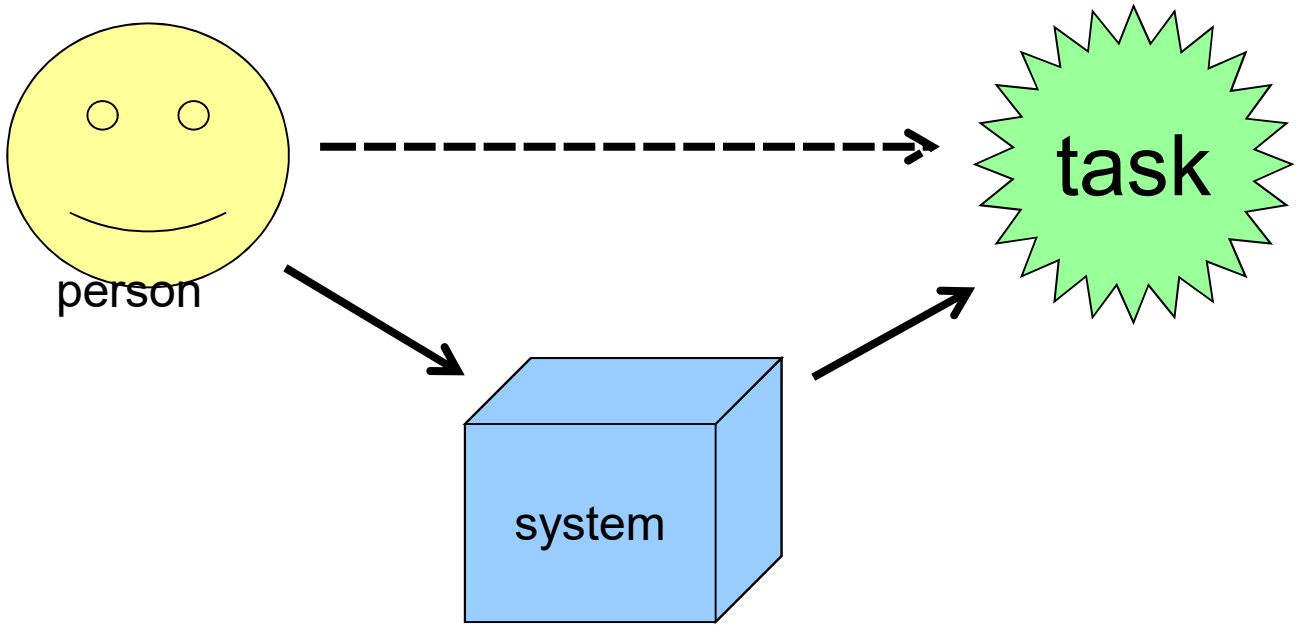
The nature of Human-Computer Interaction. Adapted from the ACM SIGCHI Curricula for Human-Computer Interaction [Hewett et al., 2002]

Agenda

- Interaction design
- Principles for Design
- Apply Norman’s principle

Interaction design

- People are trying to accomplish their tasks in life.
  (system independent)



- Introduce a system,

  User Interface should maximize their ability

Interaction design

- Design interactive products to support people in their everyday and working lives.
- Extend the way people work, communicate & interact.
- Designer vs Engineer

  Designer defines what something ought to be,

  Engineer implements that

Interaction design

→ Focus on design of user interface of software (applications + websites)

- "What is software? *From a technical perspective, a piece of software comprises forms for managing, collecting and transmitting data.* But that is not what a user thinks. *From the user's perspective software is a computer tool for performing tasks quickly, efficiently, accurately and with a minimum amount of cognitive demand.* Aim for the second one, there's a big difference. "

  http://www.ssw.com.au/ssw/standards/Rules/RulesToBetterInterfaces.aspx

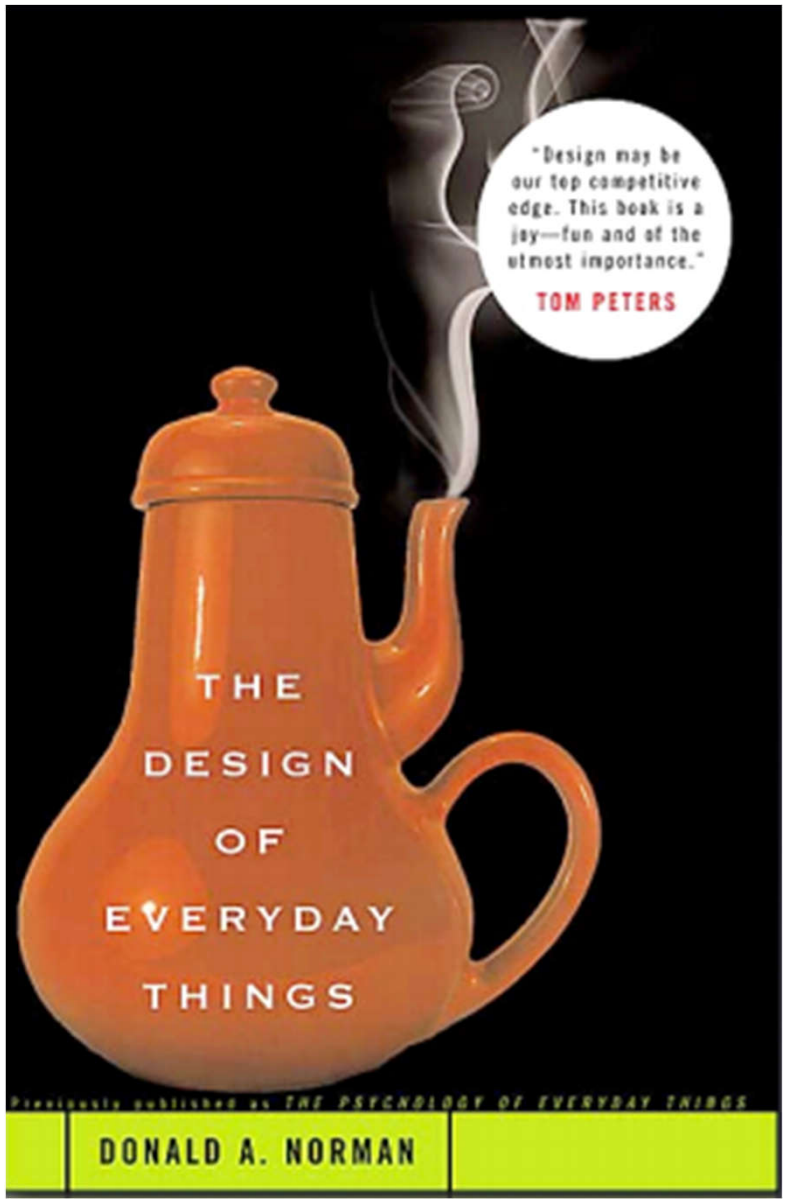
Why Interaction Design?

- Market forces: user expectancy, competitive software…
- Good design earns money
- Bad design results in:
  - Physical and emotional injury
  - Equipment damage
  - Decreased productivity
  - Higher error rates
  - User's anger and frustration

Bad design loses money & even more…

Principles for Design

1. Visibility
2. Affordance
3. Constraint
4. Mapping
5. Feedback



Visibility

- The correct parts must be visible and they must convey the correct message
- By looking at the system, users should know:
- Possible actions
- Results of their actions
- State of the system

Visibility


UNIVERSAL A/C REMOTE
POWER
TEMP
MODE
UNIVERSAL
Mini K-100E
UNIVERSAL A/C REMOTE

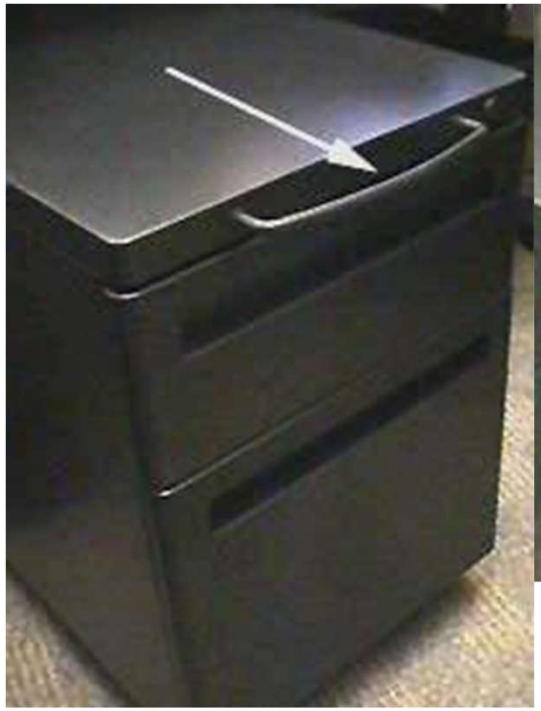

MAGIC
MENU
OK
AV
ONE FOR ALL
URC-7240

Visibility


NOKIA
NOKIA
1 oo
2 abc
def 3
4 ghi
5 jkl
mno 6
7 pqrs
8 tuv
wxyz 9
* +
0
⇧


NOKIA
C
1 oo
2 abc
3 def
4 ghi
5 jkl
6 mno
7 pqrs
8 tuv
9 wxyz
* +
0
⇧

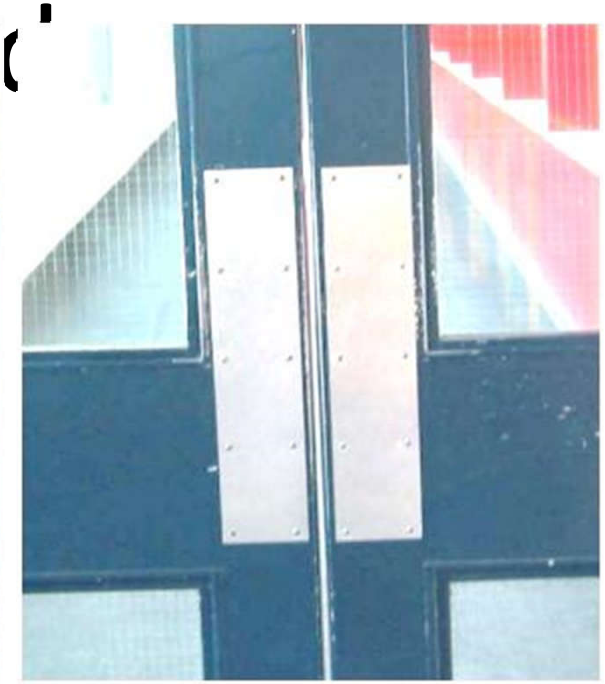
Visibility



Affordance

- The affordances of an object determine, naturally, how it can be used
- Chair affords sitting; Button affords pushing
- Scrollbar affords moving up & down; Textbox affords texting
- By looking, users should know how to use the systems
- Actual affordance vs Perceptual affordance
- Affordances rely on learned conventions.

Pull



Push

?

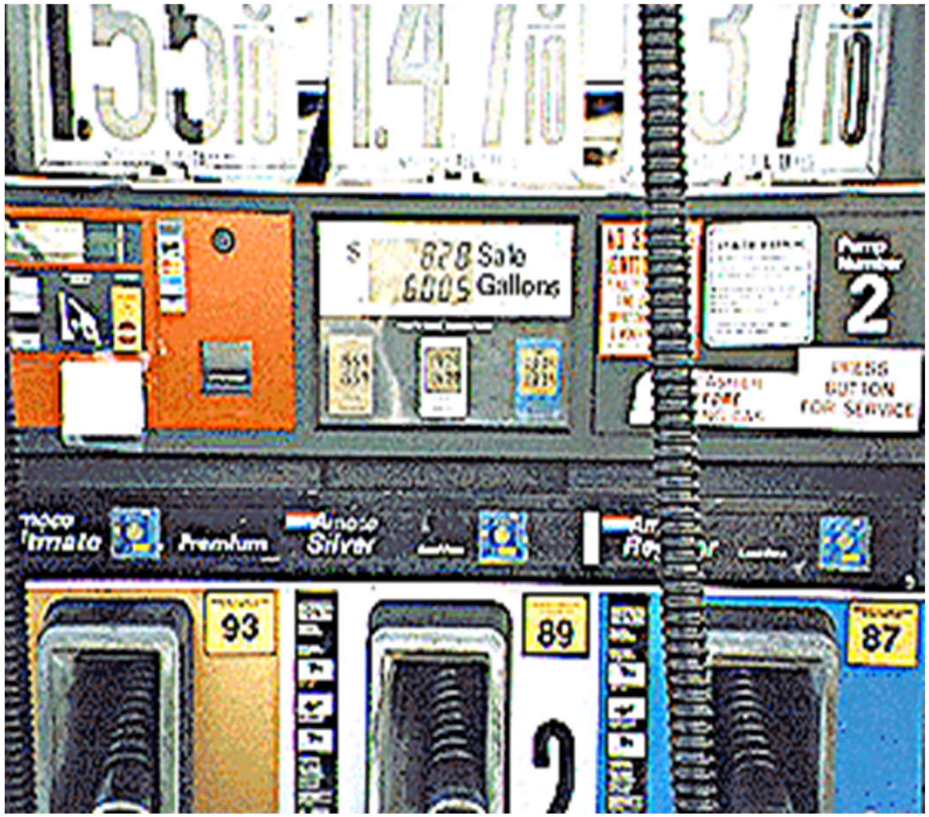
?

?

?

Affordances

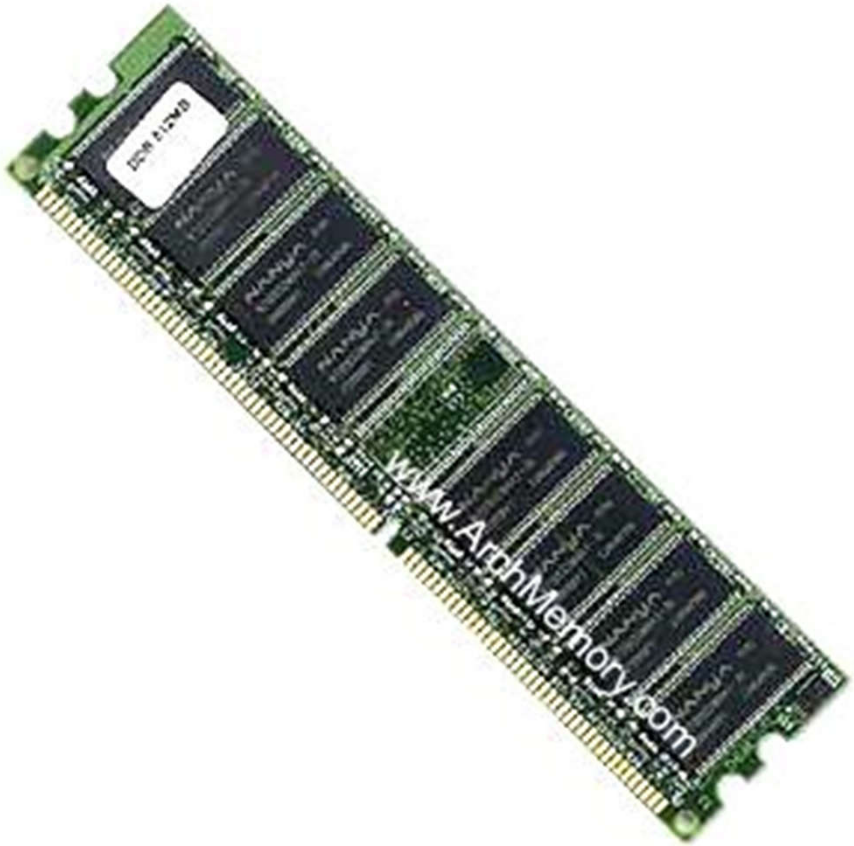
Affordance

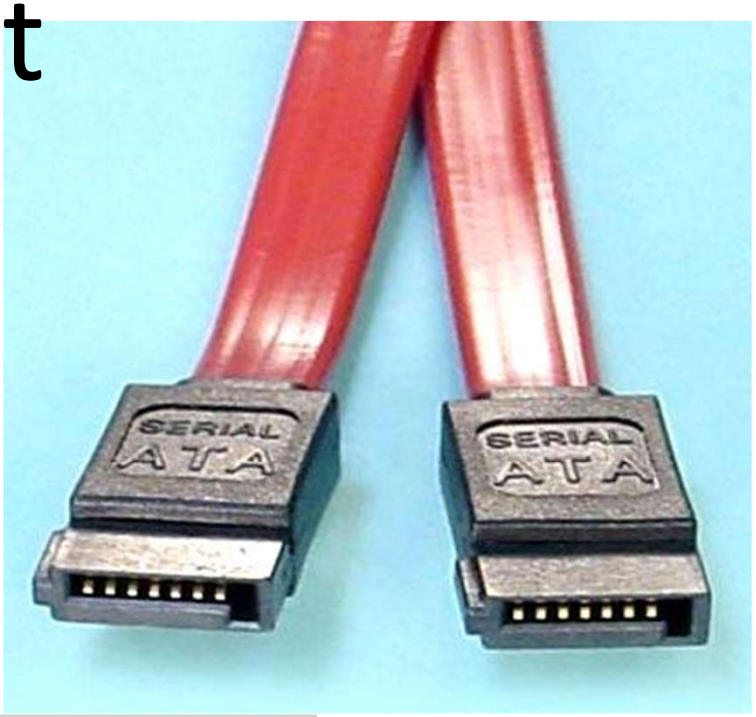


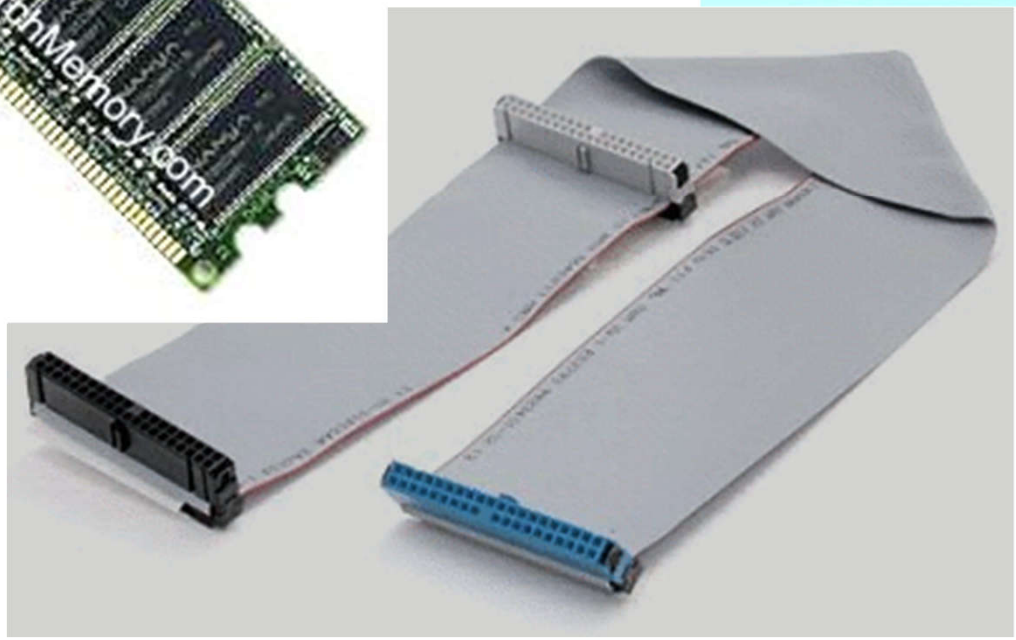
Constraint

- Constraints limit the possible actions of an object.
- Constraints prevent users from making errors.
- Constraints can be
  - Physical
  - Logical
  - Cultural

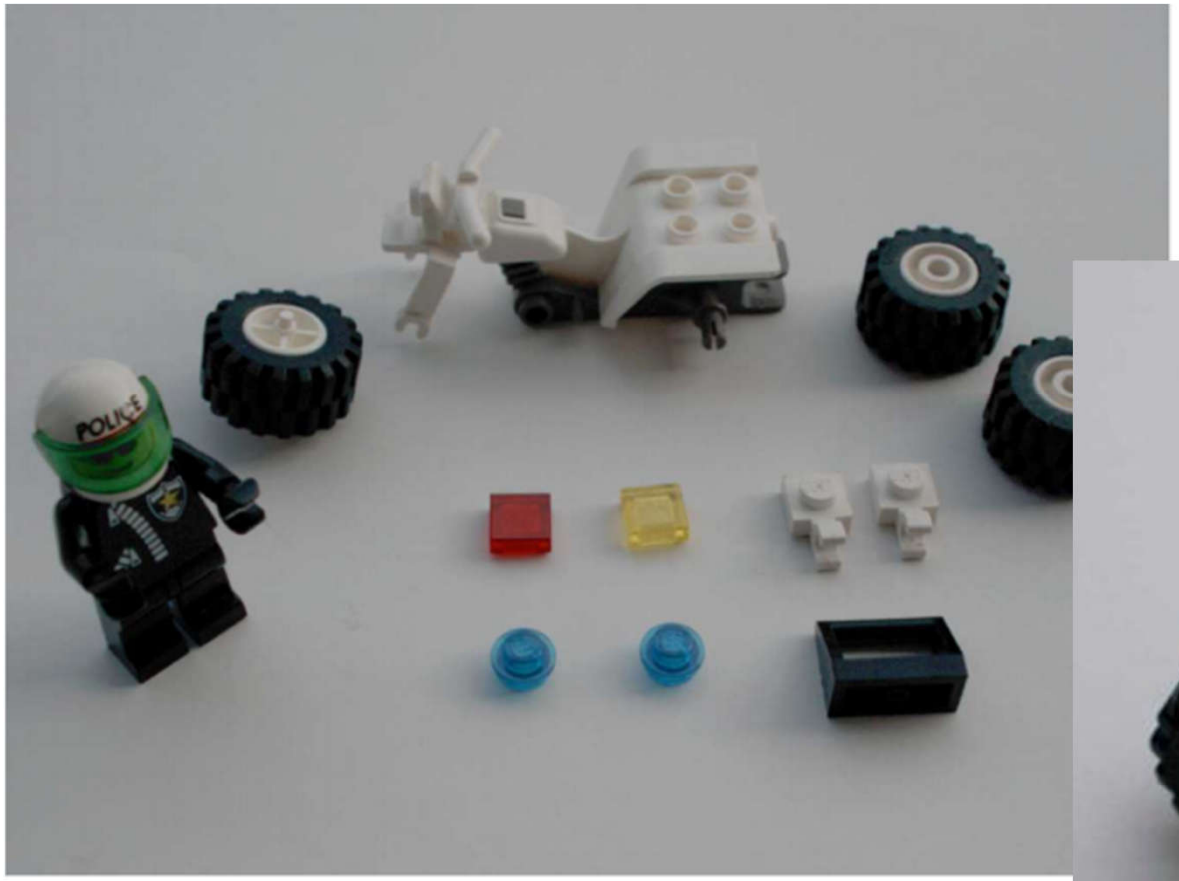
Constraint

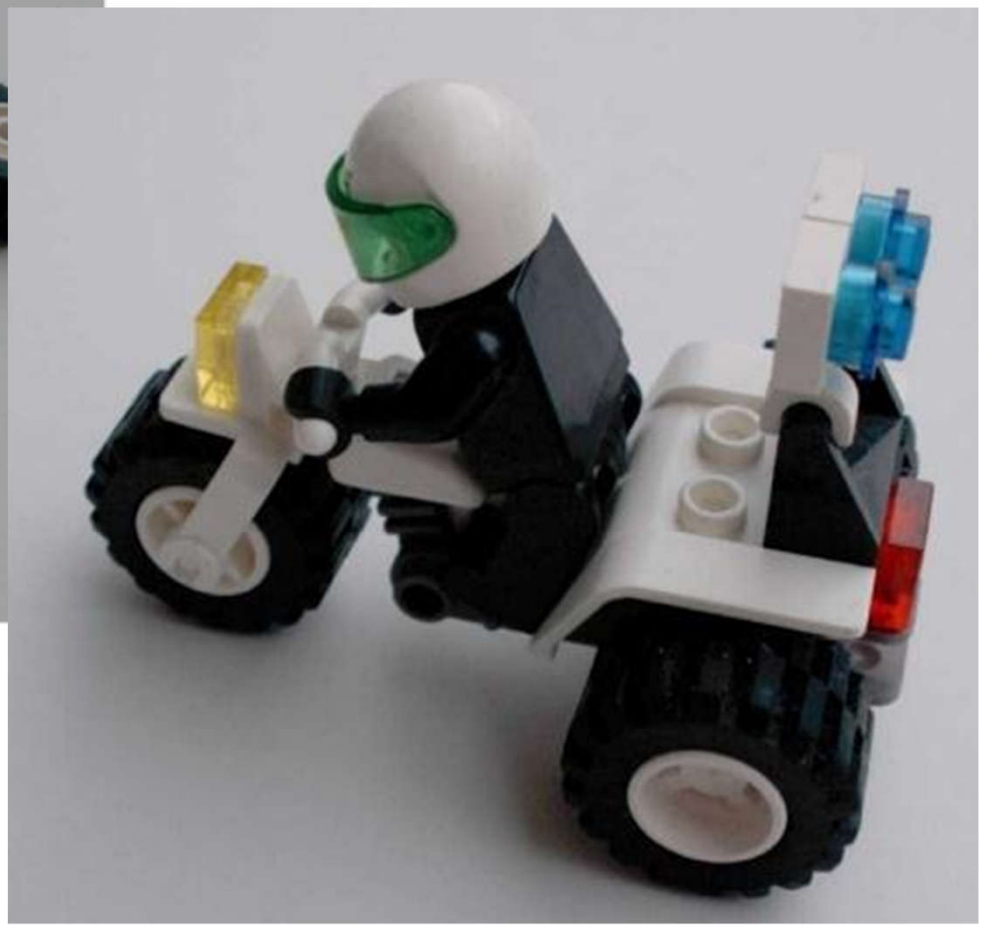






Constraint



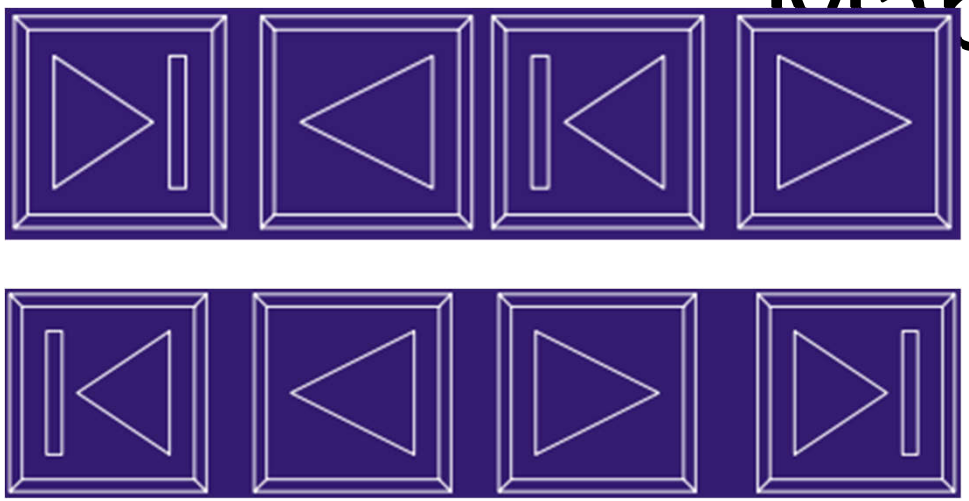


Constraint

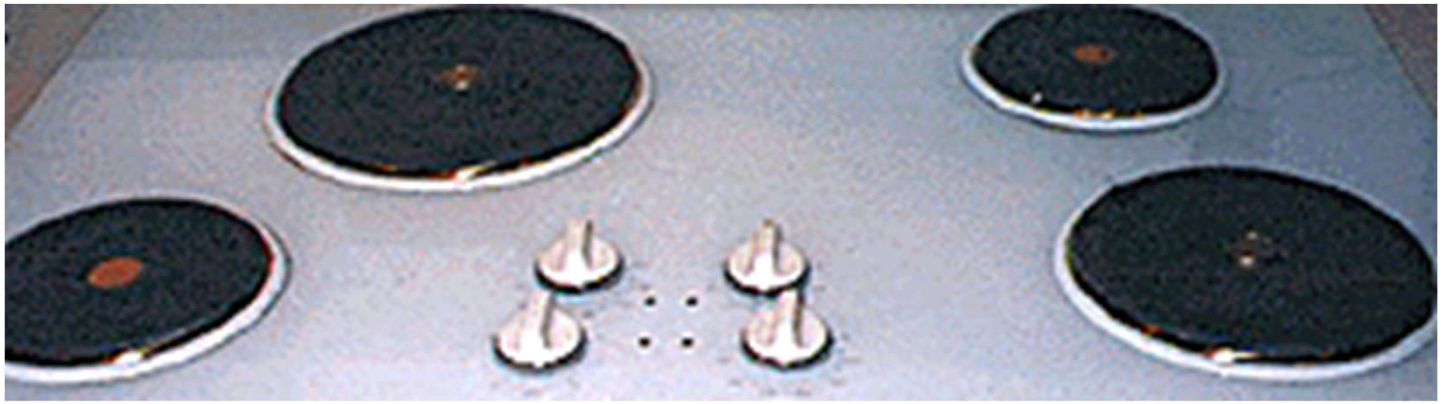
Mapping

- Mappings are the relationships between controls and their effects on a system.
- Natural mappings take advantage of physical analogies and cultural standards.

Mapping



Mapping



Mapping

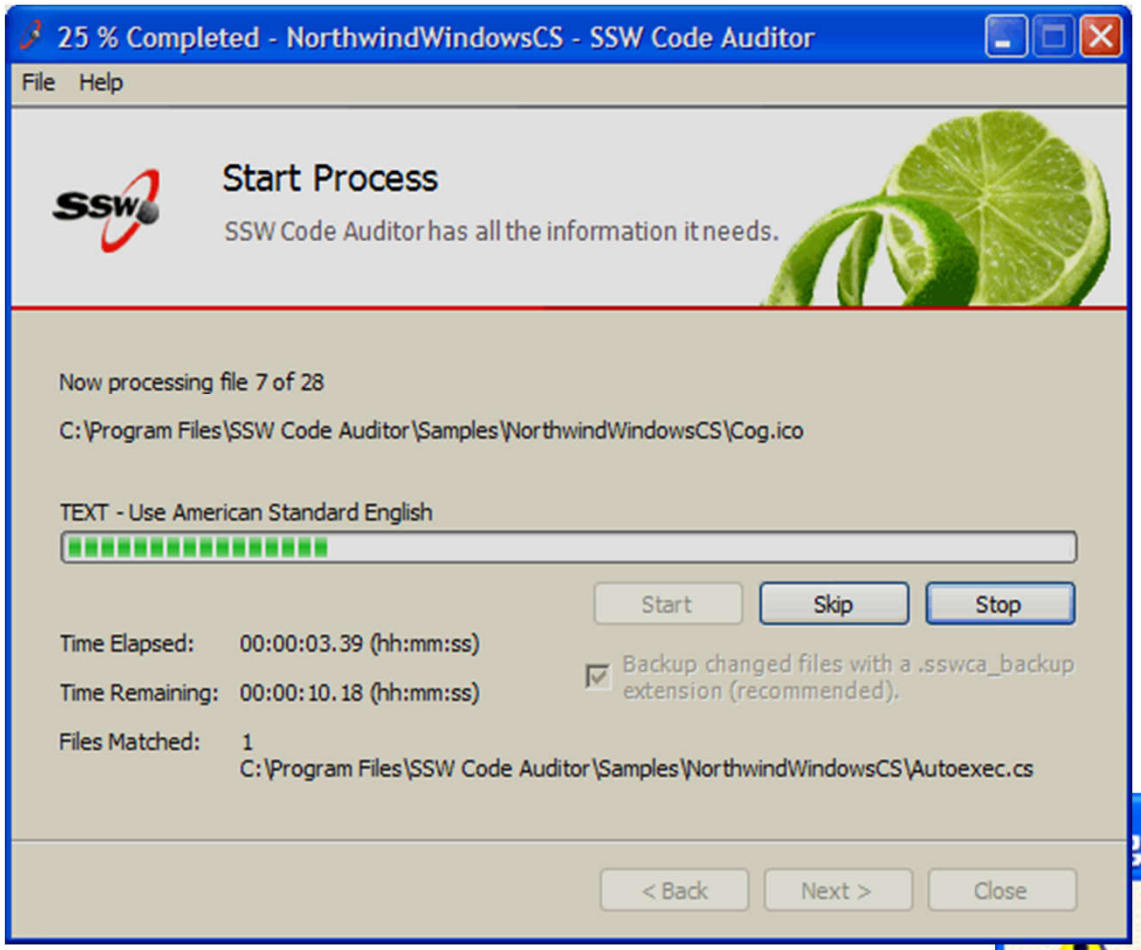
Feedback

- Feedback is sending back to the user information about what action has been done.
- Visibility of the effects of the operation tell whether if something worked correctly.
- Feedback to ensure users know what to do next.

Feedback


UNIVERSAL A/C REMOTE
POWER
TEMP
A.M WIND
M WIND
WIND
MODE
UNIVERSAL
Mod. K-100E
UNIVERSAL A/C REMOTE

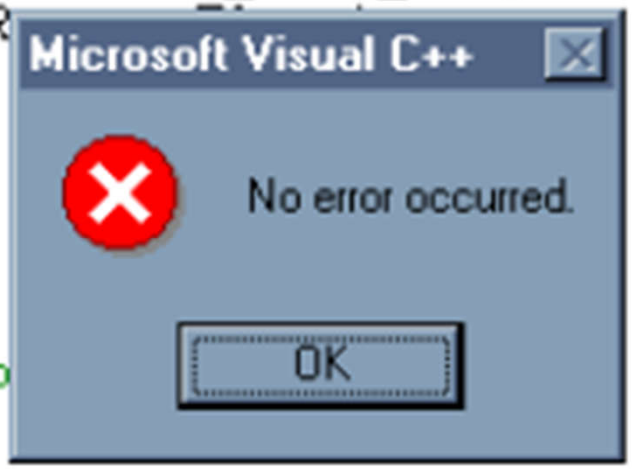
Feedback


25 % Completed - NorthwindWindowsCS - SSW Code Auditor
File Help
Start Process
SSW Code Auditor has all the information it needs.
Now processing file 7 of 28
C:\Program Files\SSW Code Auditor\Samples\NorthwindWindowsCS\Cog.ico
TEXT - Use American Standard English
Start
Skip
Stop
Time Elapsed: 00:00:03.39 (hh:mm:ss)
Time Remaining: 00:00:10.18 (hh:mm:ss)
Backup changed files with a .sswca_backup extension (recommended).
Files Matched: 1
C:\Program Files\SSW Code Auditor\Samples\NorthwindWindowsCS\Autoexec.cs
< Back
Next >
Close


SSW Exchange Reporter
Are you sure you want to exit SSW Exchange Reporter?
Yes
No

Feedback

```
usCallback(TRUE));


ection(strServerName, nPort);

CHttpConnection::HTTP_VERB_GET,
 NULL, dwHttpR
aders);



Ret);

pt the user fo

ED)


g(NULL, ERROR_INTERNET_INCORRECT_PASSWORD,
GENERATE_DATA | FLAGS_ERROR_UI_FLAGS_CHANGE_O

the dialog, bail out
```



Principles for Design

- Visibility
- Affordance
- Constraint
- Mapping
- Feedback

- ➔The relationships of these principles

- D:\Working\GoogleDrive\Working\UIT\Reference\Human–Computer Interaction\California State University Northridge\ **GUIdesign.pdf (page 9 – end)**

How can we apply Norman's principles to Computer Interface Design?

Visual GUIs design

- Communication via Visual Language
- Functional > Aesthetic
- Simplicity > realism
- Geometry: scale, contrast & proportion
- Management: visual organization
- Grids: modular visual design
- Semiotics: icons & symbols as signs
- Style

Visual language

- Vocabulary by visual elements
- Syntax by usage rules for elements
- Literacy by experience of designer & user
- Style skill by w/ vocabulary & syntax

Elegance and Simplicity

- Reduce design elements to a minimum
  - functionality not photo-realism
  - reduce visual search (and cognitive) load
- Simple designs are more: approachable recognizable, remembered usable (immediately and thereafter)

Reduction

- Determine essential qualities(adjectives) -- color, labels, controls
- Is each element needed? Would the design suffer if removed?
- Test element's necessity by removing it. If design is fine omit element.
- Omitted elements can be indirectly accessible via menus -- option
- buttons

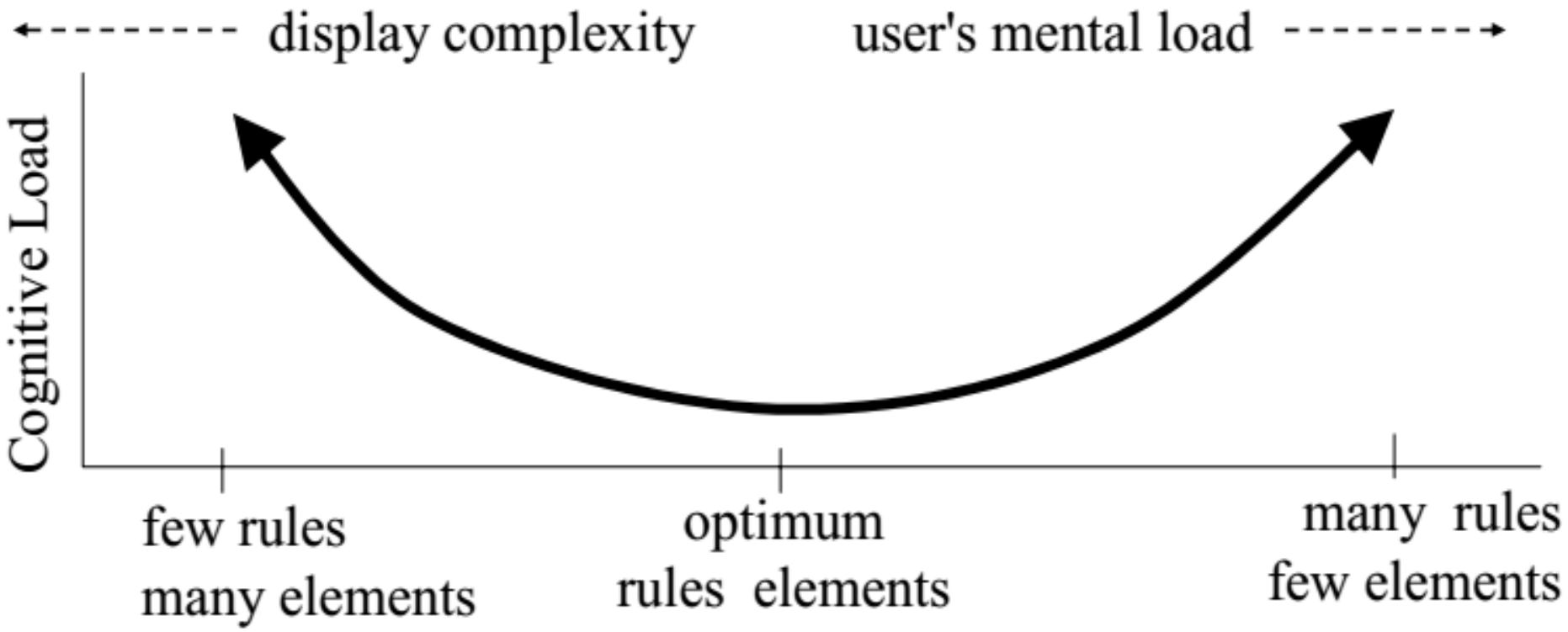
Regulation

- Use regular shapes, simple contours, muted colors
- Make multiple similar forms visual properties identical (size, shape,
- alignment, spacing ...)
- Limit font variation to few sizes in two families
- Do not regularize ***critical elements*** -- make them stand out (novel)

Leverage Design Rules /Elements

- Find multiple elements doing similar functions
- Design a combined element for the functions
- Do not overload the modality of elements.
- Design trade off between simple interface and cognitive load (rules for use)

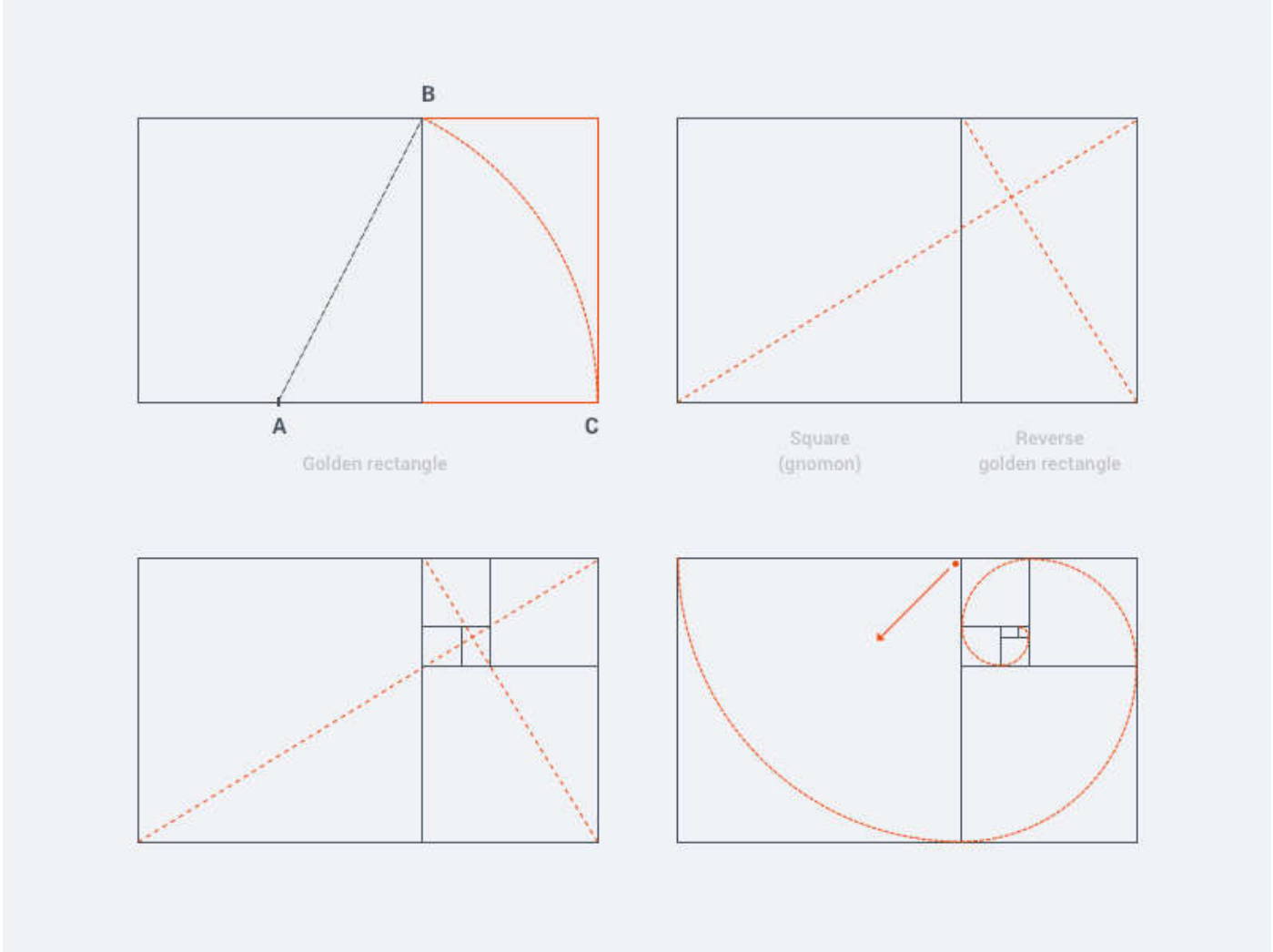
Leverage Design Rules /Elements



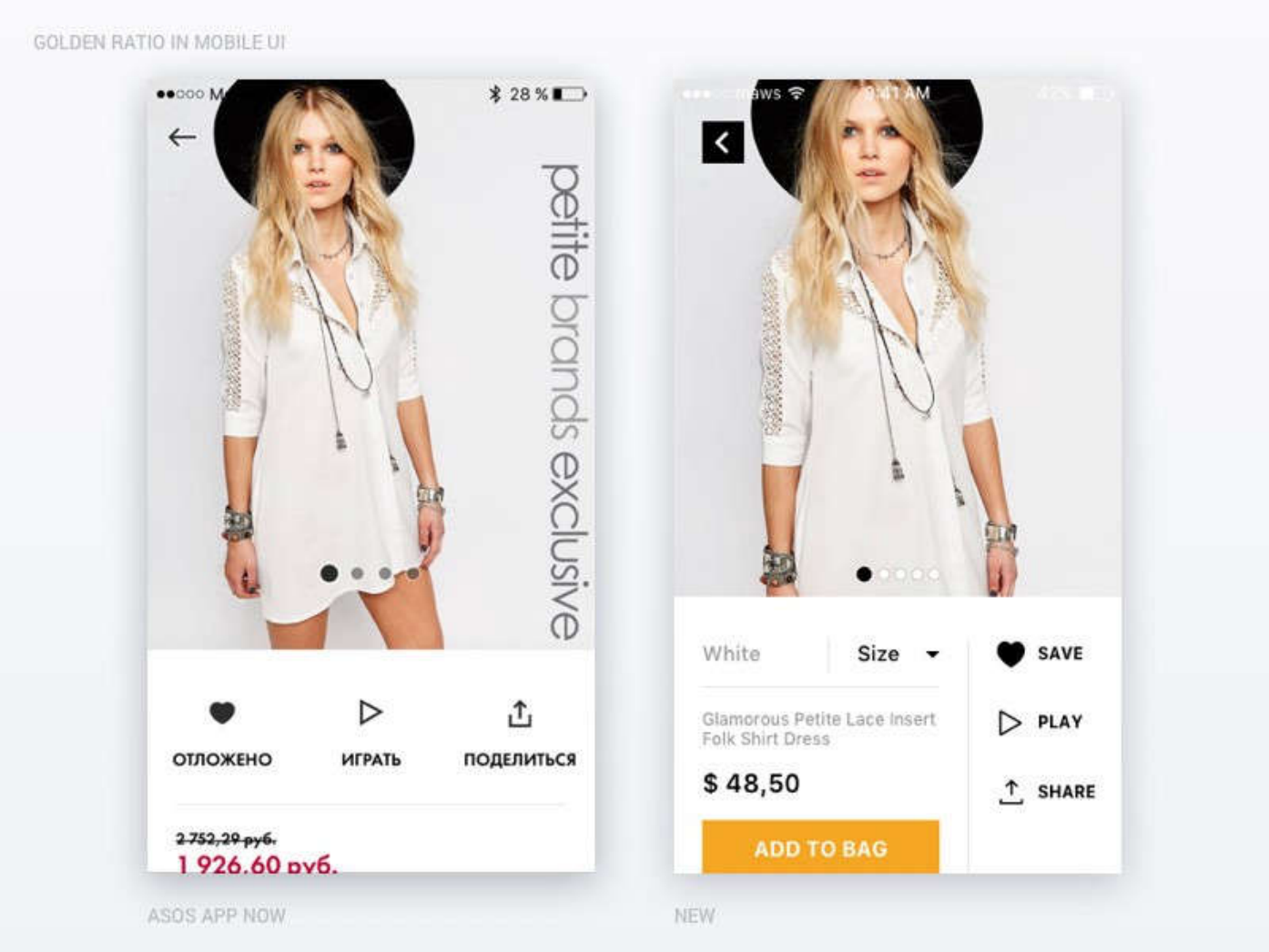
Geometry: Scale, Contrast & Proportion

- Scale is the element's relative size (area)
- Contrast is an element's distinctive dimension.
- How is it differentiated from others?
  - size, value (greyscale), hue (color),
  - position, orientation, texture, shape
- Proportion is the ratio of sizes
  - **Computer displays 1.33 to 1**
  - **Golden (classical) rectangles 1.618 to 1)**
  - **Printed Paper A standard 1,414 to 1**
- Scale and contrast are used to emphasize elements.
- Activity is how a design uses geometry to lead the visual search (view).
- Humans seek patterns, \ design should provide cues to group common, differentiate unique elements, and provide comparison (evaluation) information

Geometry: Golden (classical) rectangles

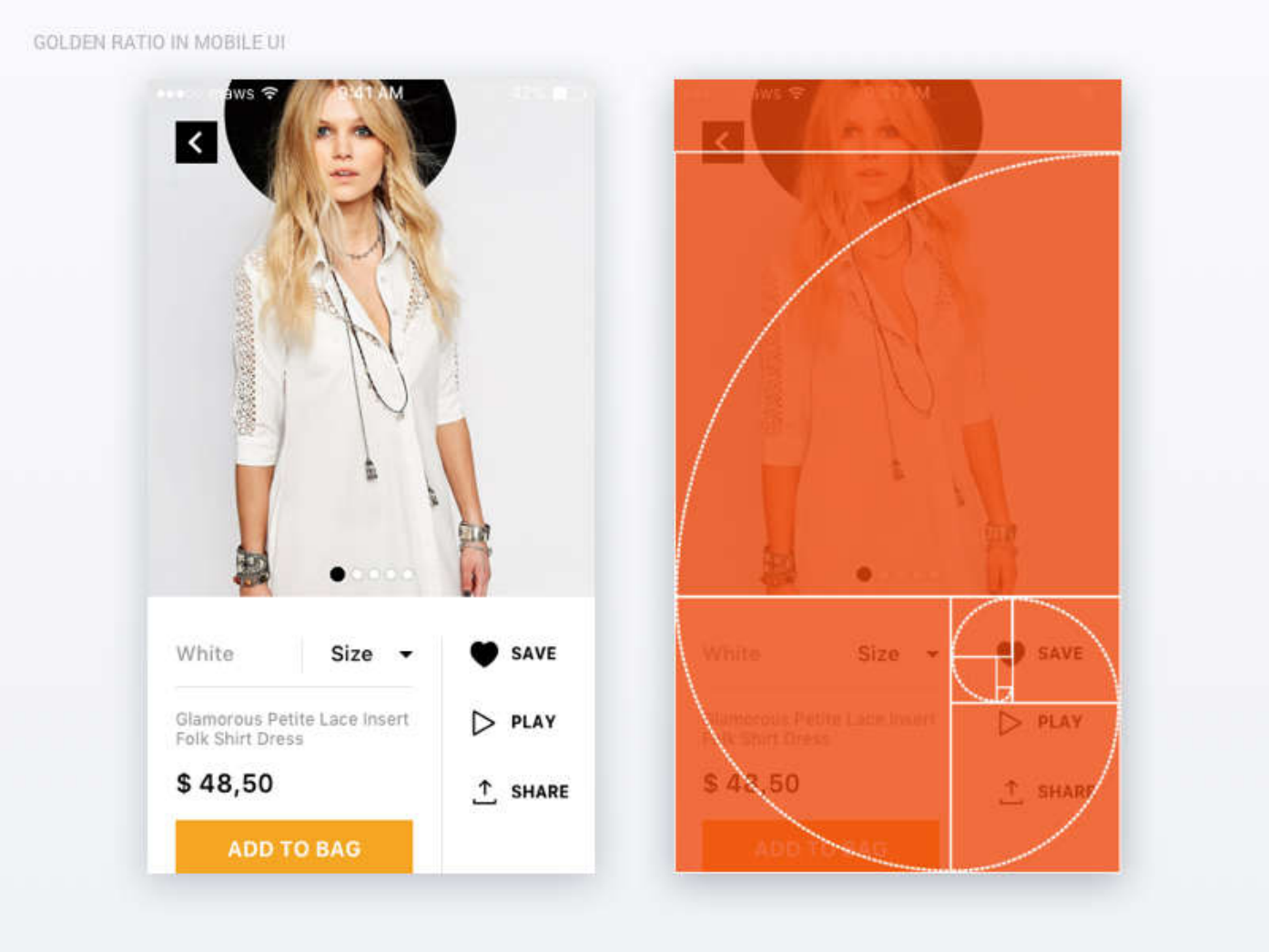


Geometry: Golden (classical) rectangles



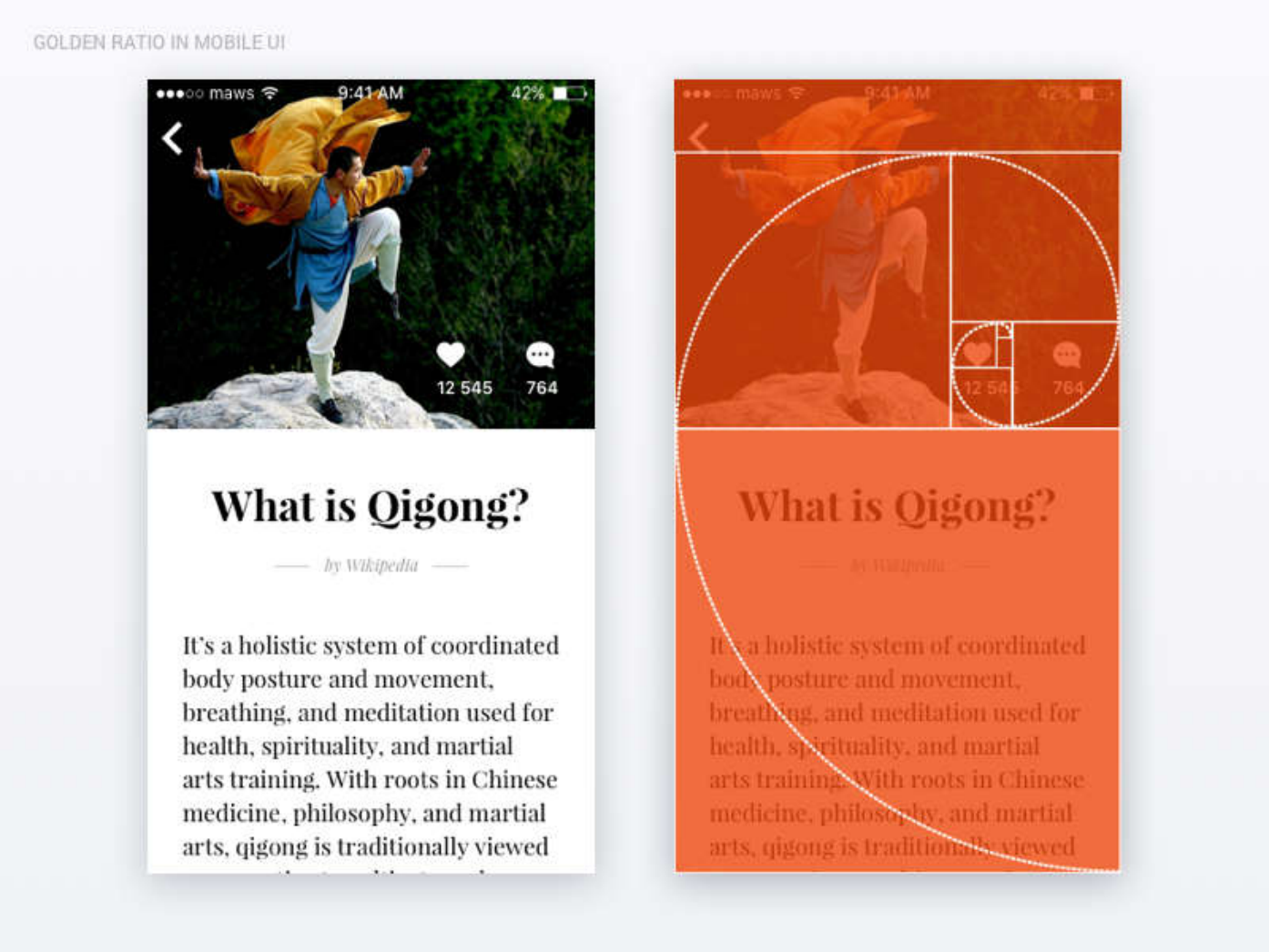
https://blog.prototypr.io/golden-ratio-in-ui-design-8d11e66582c3

Geometry: Golden (classical) rectangles



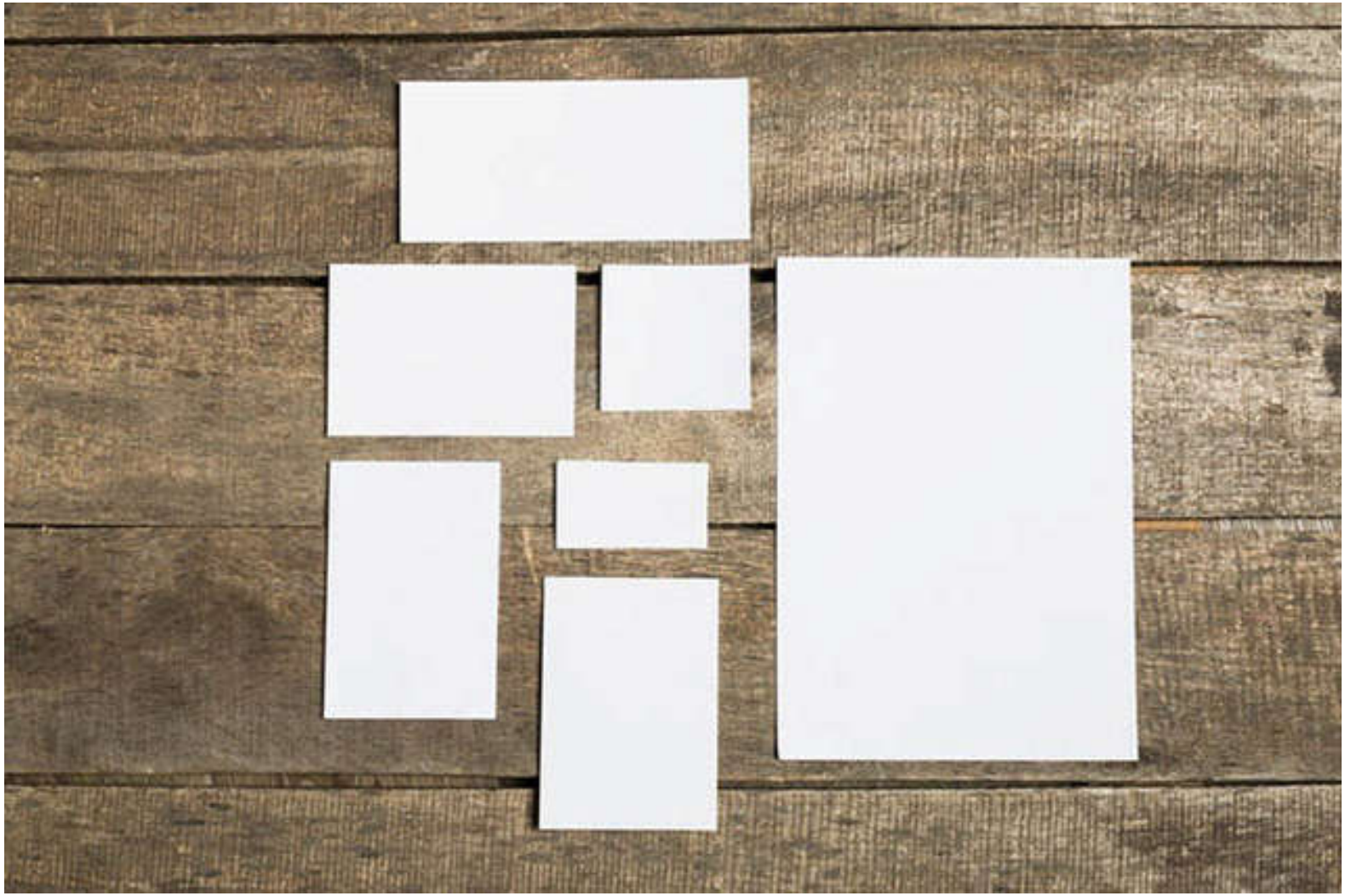
https://blog.prototypr.io/golden-ratio-in-ui-design-8d11e66582c3

Geometry: Golden (classical) rectangles

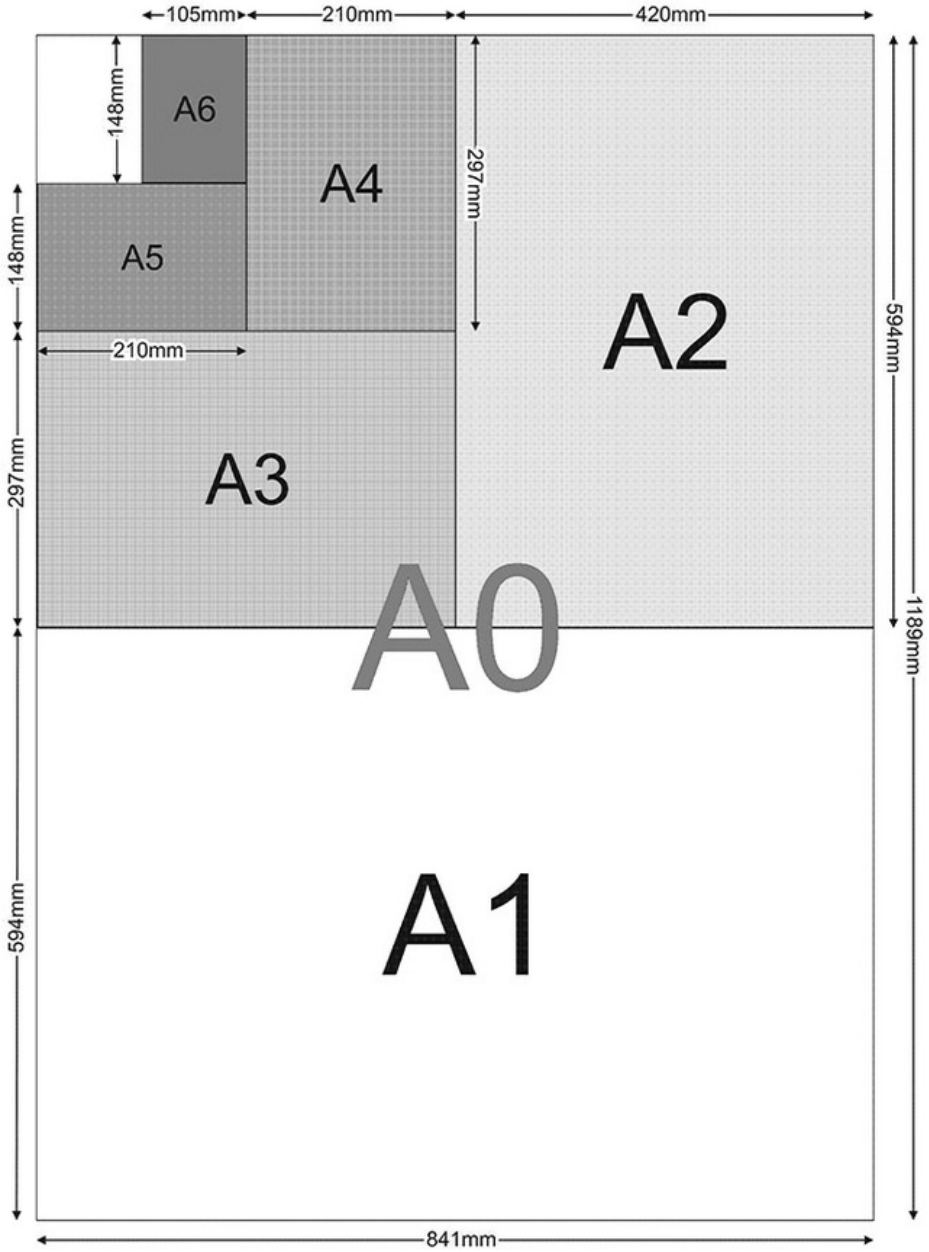


https://blog.prototypr.io/golden-ratio-in-ui-design-8d11e66582c3

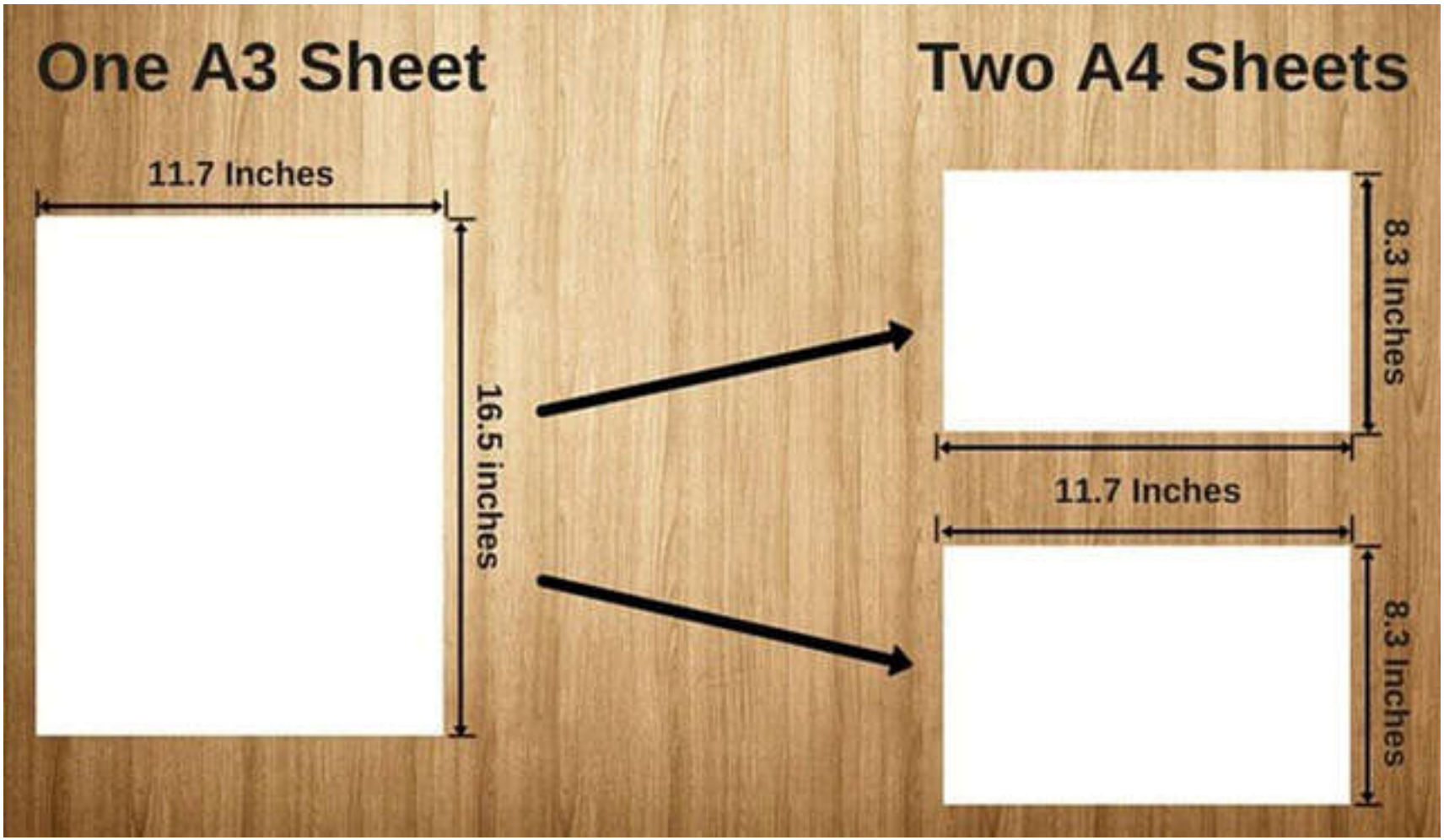
Geometry: **Printed Paper A standard**

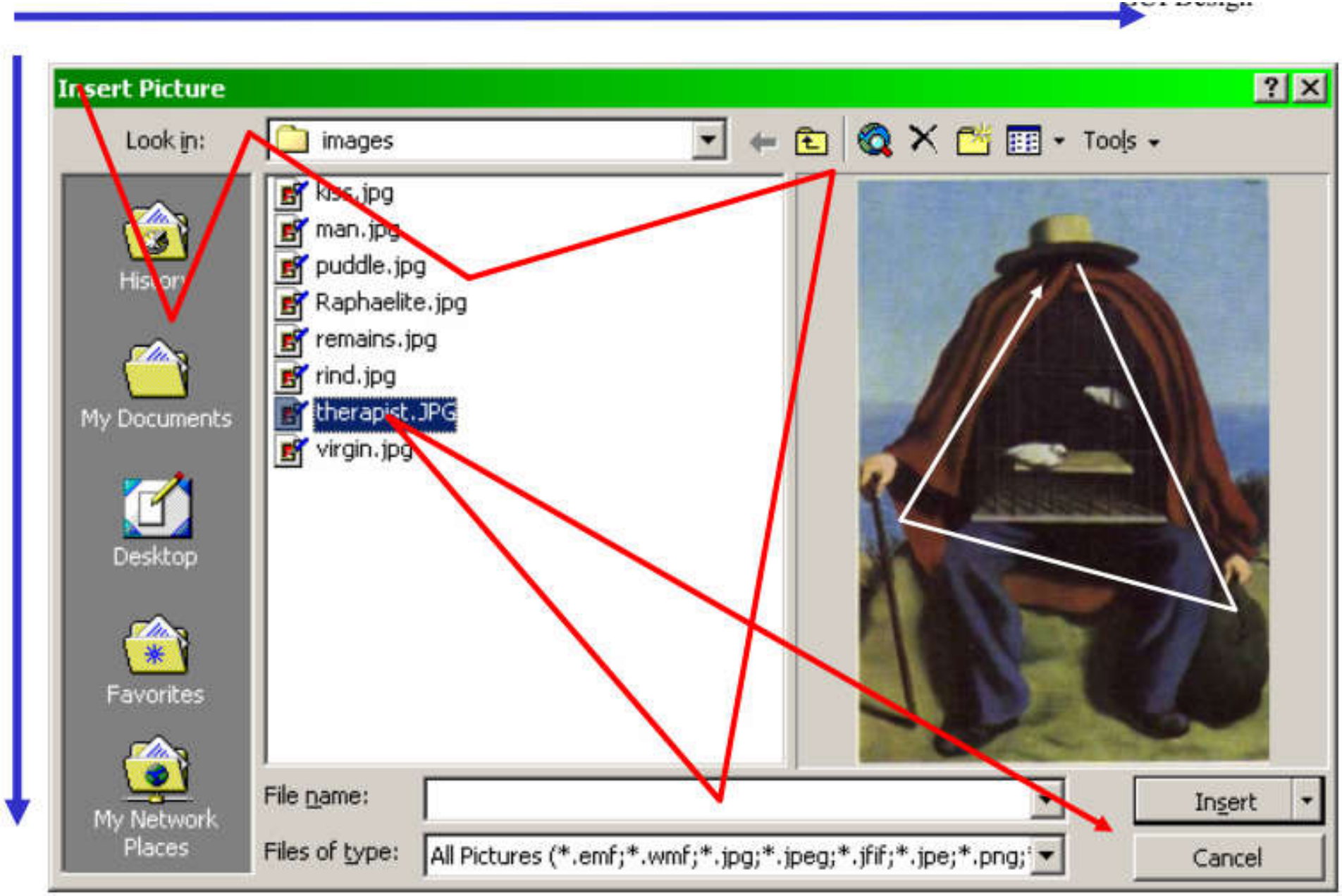


Geometry: **Printed Paper A standard**



Geometry: **Printed Paper A standard**




Insert Picture
Look in:
images
Tools
kss.jpg
man.jpg
puddle.jpg
Raphaelite.jpg
remains.jpg
rind.jpg
therapist.JPG
virgin.jpg
History
My Documents
Desktop
Favorites
My Network Places
File name:
Files of type:
All Pictures (*.emf;*.wmf;*.jpg;*.jpeg;*.jfif;*.jpe;*.png;
Insert
Cancel

Visual Variables

- **Associative**: independent of other variables. Most variables are associative.
  - size and value are dissociative - they affects visibility of other variables. (e.g. line to thin to see color)
- **Selective Perception**: viewing isolates all group members into an image.
  - shape is not selective.
- **Ordered Perception**: viewing can determine ordering (ranking).
  - Ordering reduces need for legends (keys).
  - Position, size and value enable ordering
- **Quantitative Perception**: viewing can determine relative amount of difference not just ranking. Must be easily apparent.
  - size (area) and position are quantitative

Color Usage

- 3 interacting variables of color vision:
  - Hue color
  - Brightness intensity (bright - dull)
  - Saturation %color in field
- Opponent process theory of color vision
  - These colors can't be seen in same patch of light (adjacent).
  - They produce shadows and edges.
  - Avoid use of opponent (opposite) colors.
- Blue is the hardest color to see small changes in hue
- Selection / Applicability
  - Color is very useful to have user selected items stand out in a display.
  - Color can also be used to indicate whether a menu option is valid in the current state or not ("greyed options")

Color Usage

- **Alert / Attention**: Change of color represents change in state or mode
  - (Traffic signals: green, yellow, red).
  - Use few colors that are easy to discriminate
  - Use warning colors sparingly.
  - Consistent system wide analysis of color use.
- **Element Discrimination**: Color provides contrast and improves discrimination. Need high contrast difference. Contrast a function of luminance or hue

Color Usage

- **Category grouping & field definition**: Color can help group display elements and facilitate visual search. Visual search is affected by:
  - number of items
  - color separation of categories
  - legibility of coded symbols
  - relationship between color coding and targets
- As screen density increases color effect increase. Color can define visual fields on display- weather maps

Size & Visual Acuity

- As number of colors increase size of text should also increase.
- Color can't be assumed! Redundantly code display.
- Designer's color perception != user's color perception
  - color & text codes (categorization)
  - color, size & text
  - color, size, text & icon ....
- Color Memory: 5 - 7 color memory for codes. Don't tax Working
- Memory use around 4 colors!

Grouping guidelines

- Group into a small number (7 ± 2)
- Rank the importance of the groups
- Show hierarchical relations with size
- Show non-hierarchical groups with hue
- Maximize differences between groups
  Minimize differences within groups.

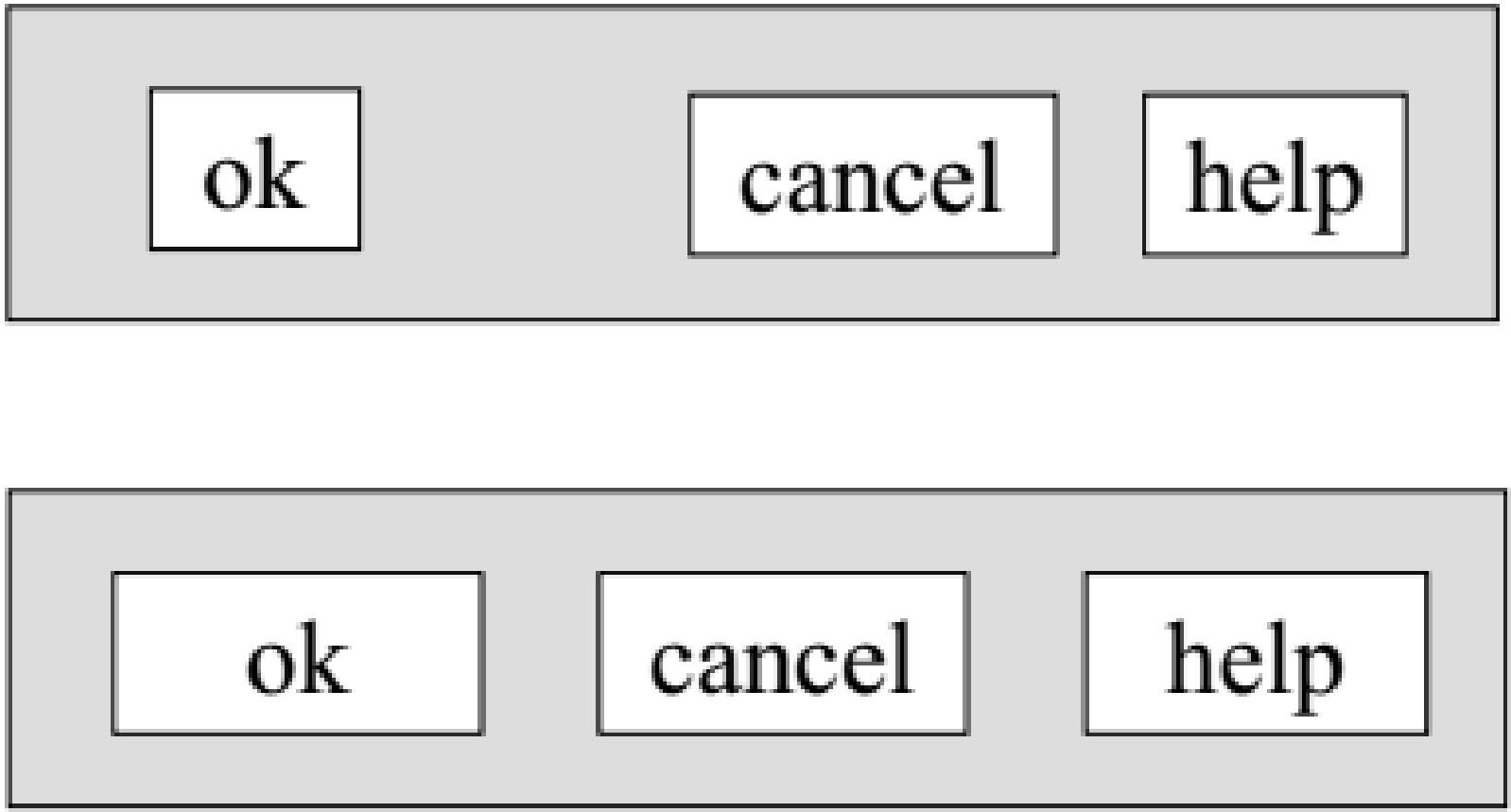
Using Perceptual Distinction

- Determine range of variation (min .. max) for sizes, color dimensions.
- Use logarithmic > linear scaling for discrimination.
- At least double each level!

Balanced use of figure / ground

- Determine and equalize the visual weight of figure / ground.
- Use internal padding to surround the figure and separate it from borders.
- Spend valuable screen real estate for internal spacing!
- Position the figure w/in the ground.
- Centering is usually most appropriate.
- White (Negative) Space is not wasted
- White space is needed for figure / ground integration.
- White space helps spatial separation / organization.
- Increase white space around critical elements

Balanced use of figure / ground



Alignment and Visual Relationships

- Elements should be aligned with boundaries and margins.
- Alter size and proportion when needed to support alignment.
  - Extend elements beyond margin with respect to sharpness of adjacent angles.
  - Greater the acuteness of angle the greater the extension
  - Proportional fonts use optical adjustment.
- Items not aligned to anything on display should be proportional to display

Visual Perception Structure

Gestalt principles:

1. Principle of proximity
2. Principle of similarity
3. Principle of closure
4. Principle of continuation
5. Principle of enclosure
6. Figure-ground principle
7. Principle of symmetry
8. Principle of connection
9. Principle of common fate

Grids

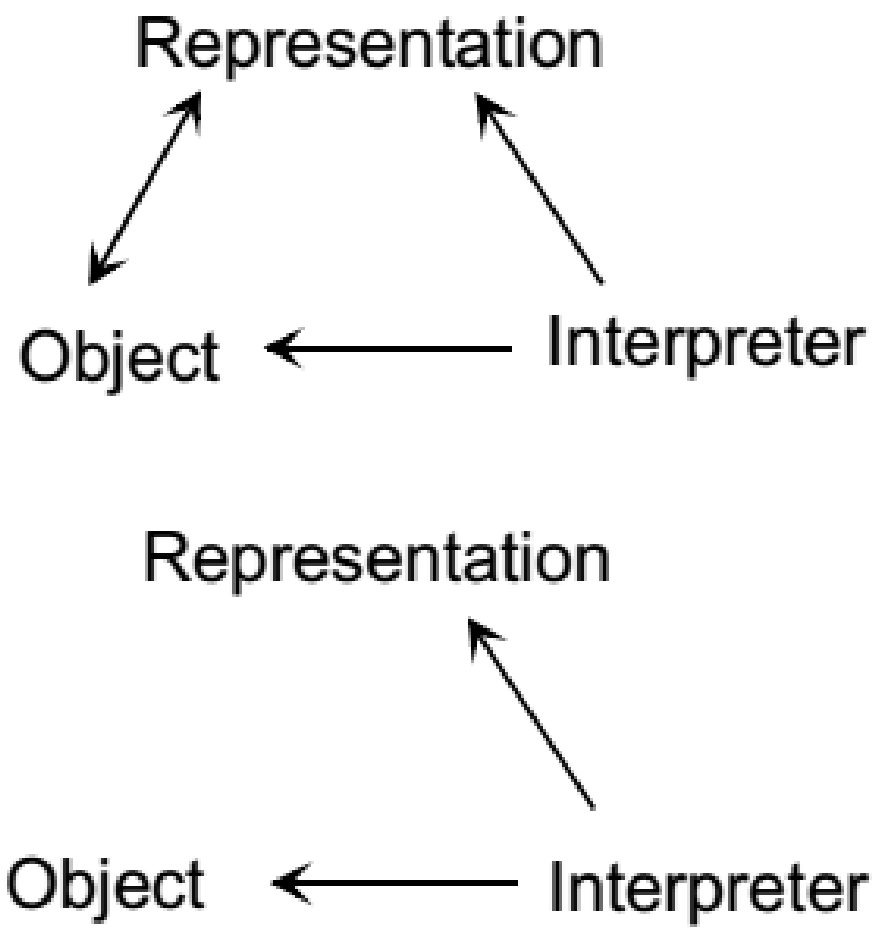
- Modularity in visual structure GUI Design
- Regular visual structures are predictable, flexible, and efficient.
- Grids benefit design and provide "scalability" to a GUI application. As screens and dialogs increase in numbers a grid layout simplifies design and increases use.
- Canonical grid layout enables 6, 4, 3, and 2 division of elements on a display.
- Any remaining visible grid lines should be half intensity in final display

Grids

- GUI Windows are rectangular: grids are the first display layout considered.
- Determine vertical unit that allows any two controls to be adjacent
- When spacing of multi-line controls consider labels as controls (in layout)
- Horizontal unit 3 x as wide as vertical unit
- Use 5-7 column divisions of horizontal units

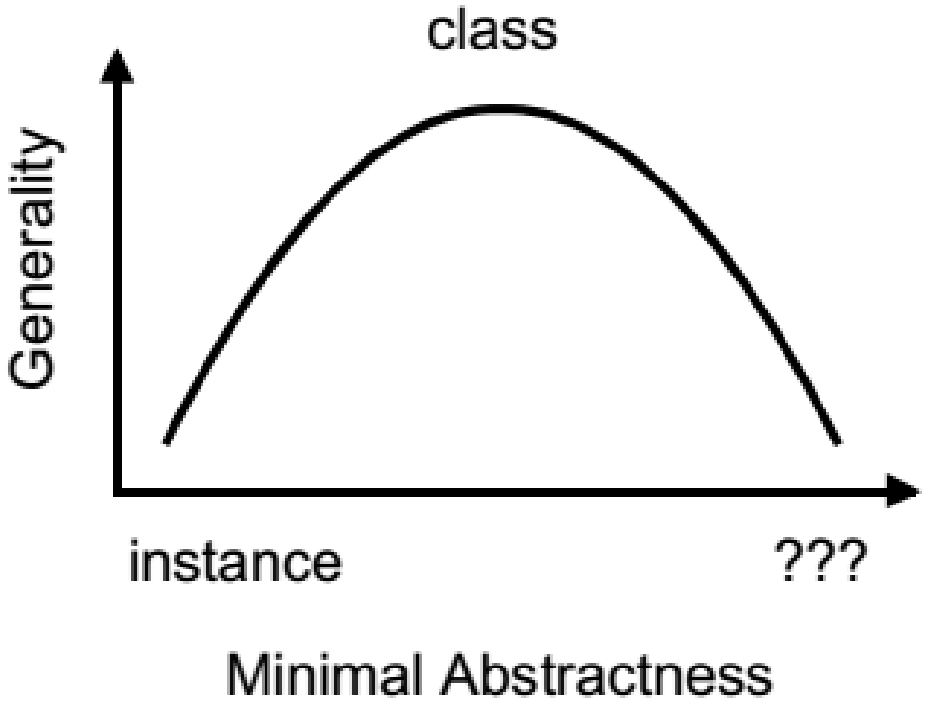
Semiotics -- GUIs as signs

- Iconic sign: representation resembles object
- Symbolic sign: viewer associates representation and object



Semiotics -- GUIs as signs

- Abstractness helps iconic generality
- Realistic icons represent instances
- Stylized (minimalistic) icons represent a class



Icon Selection

- Use iconic representation when communication goal is concrete & familiar.
- Can use symbolic representation for repetitive concepts (learned).
- Use text for abstract or complex (subtle) representations (processes).
- Avoid mixing textual, iconic and symbolic signs w/in image set

Icon Refinement

- Determine the level of abstraction.
- Try simplifying shapes into regular geometric forms.
- Try using negative space to determine contour

Coordinating Icons

- Use a similar perspective & point of view
- Use a similar style of representation -- don't combine icons with symbols.
- Use consistent size, orientation, layout, color, and visual proportion (weight / area) to each image. Grids help internal structure.
- Use the same elements when possible in your image set {lines, rules, textures} -- limit the visual vocabulary

Mastering Style

- Read style guides -- learn the conceptual model from the user's point of view.
- Respect the visual language of the style.
- Learn the usage and methods of user customization -- fonts, color and how they can possible degrade the style

Working across (with) many Styles

- Develop a translation table across the style set
- Extend the widget set to fill out gaps in the translation table.
- Use menu and control mechanisms of the style.
- Focus on high level orienting features -- keep similiar structure when possible

Q&A