02 - Guidelines, Principles, and Theories
Intuitive judgment often leads to cluttered displays, complex and tedious procedures, inconsistent terminologies or sequences of actions, inadequate functionality, insufficient informative feedback, long response times, etc.

As a consequence you will have to deal with frustrated, discouraged or even angry users.

Hence, you need to overcome pure intuitive judgment by relying on some sort of guidance. Such guidance is available in 3 forms:

1. **Guidelines**: Prescribe cures for design problems, caution against danger, provide helpful reminders based on general/common knowledge/experiences; are very specific and practical

2. **Principles**: Help to facilitate a structured design process; are more abstract and widely applicable

3. **Theories**: Describe objects and actions with consistent terminologies, help in analyzing and comparing design alternatives, predict reading, typing or pointing times, etc.; are largely very abstract
Guidelines
Guidelines

Overview

- ≈ Best practice, based on practical experiences or empirical studies
- Written guidelines help to develop a “shared language” and based on it promote consistency among multiple designers and designs/products
- This includes issues such as: terminology, appearance, action sequences, input/output formats, and Dos and Donts of graphic styles
- They build a good starting point but need management processes to facilitate their enforcement
- Examples of guidelines: (A) How to provide ease of interface navigation, (B) how to organize the display, (C) how to draw user’s attention, and (D) how to best facilitate data entry
Guidelines

Example A: Interface Navigation

5 rules (examples) of how to enhance the ease of interface navigation:

- Standardize Task Sequences
- Ensure that embedded links are descriptive
- Use unique and descriptive headings
- Use checkboxes for binary choices
- Use thumbnails to preview larger images
Guidelines

Example B: Display Organization

5 high-level goals of display organization:

- Ensure consistency of data display
- Promote efficient information assimilation by the user
- Put minimal memory load on the user
- Ensure compatibility of data display with data entry
- Allow for flexibility in data display (user controlled)
Guidelines

Example C: Draw User’s Attention

7 issues that can be used to draw user’s attention:

- Intensity: Use high intensity to draw attention
- Marking: e.g. underlining, using arrows, boarders, etc.
- Font Size: Use large fonts to draw attention
- Font Style: Use exceptional font styles to draw attention
- Blinking/Animation: Use blinking to draw attention (careful!)
- Color: Use exceptional colors to draw attention
- Audio: Use harsh sounds for exceptional conditions

Note: Always use these concepts with care (e.g., do not clutter your display with too many different styles or colors, provide an on-off switch for sounds). As a general rule, the target audience determines the extent to which the named features should be applied.
Guidelines

Example D: Facilitate Data Entry

5 high-level objectives should be pursued in order to best facilitate data entry. If applied meaningfully, productivity will increase, time-to-learn will be reduced and the error probability should be lower.

- Ensure consistency of data entry transactions
- Require minimal input by the user (e.g., button vs. typed command)
- Require minimal memory load on user (e.g. command line vs. forms)
- Ensure compatibility of data entry with data display
- Enable flexibility of data entry (user controlled)
Principles


**Principles**

**Overview**

- Are more fundamental, widely applicable and enduring than guidelines (i.e., guidelines often need to be individually specified for every project/organization while principles are project-independent)
- Problem: Clarification is more important in order to ensure consequent interpretation
- 5 tasks/principles that may be performed/followed as part of most UI projects:
  - Determine your target audience (in particular user skill-level)
  - Identify the tasks that users perform
  - Choose an appropriate interaction style
  - Apply the “8 golden rules” of UI design
  - Always try to prevent user errors
Principles

Target Audience - Overview

- Every designer should start with trying to understand the intended user (e.g., population profile, gender, physical and cognitive abilities, education, cultural/language background, knowledge, usage pattern, attitudes, etc.).
- Specifying the intended user is a very important yet very difficult task. One particular challenge is: Many applications cater for several diverse user groups at the same time.
- Determining the user skill level is especially important. Here, you need to distinguish between 2 skill levels:
  A. Skills in regard to interface usage in general
  B. Skills in regard to the particular application/task domain
- Start with classifying your users into one of 3 groups:
  1. Novice or first-time users
  2. Knowledgeable intermittent users
  3. Expert frequent users
Principles

Users’ Skill Levels - Novice or First Time Users

- Know neither the interface concepts nor the task domain (novice users) or are familiar with interface concepts but are new to the task domain (first time users)

- Some specific guidelines for this group are:
  - Provide instructions, dialog boxes, online help
  - Restrict the vocabulary to a small number of familiar and consistently used terms
  - Use only a small number of actions (simple tasks)
  - Provide sufficient informative and feedback about accomplished tasks
  - Provide constructive, very specific error messages
  - Provide user manuals, video demonstrations, and task-oriented online tutorials
Principles

Users’ Skill Levels - Knowledgeable Intermittent Users

- Have stable knowledge about task concepts and broad knowledge about general interface concepts, may have difficulty retaining the structure of menus or the location of features over time
- Some specific guidelines for this group are:
  - Reduce burden on memory
  - Orderly structure menus
  - Use consistent terminology and sequences of actions
  - Use meaningful messages
  - Provide guides to frequent pattern of usage
  - Provide context dependent help
  - Use well-organized reference manuals
Principles

Users’ Skill Levels - Expert Frequent Users

- Are thoroughly familiar with the task concept and general interface concepts
- Main goal of such “power users” is to get work done fast and efficiently
- Some specific guidelines for this group are:
  - Ensure rapid response times
  - Provide brief and non-distracting feedback
  - Provide shortcuts to carry out frequent actions
  - Provide string commands and abbreviations
Principles

Users’ Skill Levels - Summary

- It is crucial to define your user groups for every project; the more diverse groups you have to accommodate the more challenging is the design process.

- So what can you do if there are multiple diverse groups?
  A. Use a “multi layer” design/approach to learning: Structure your UI in a way that limited users may only work with a small set of functions and objects while advanced users can access advanced functionality.
  B. Give the user control over the density of informative feedback (e.g., error or confirmation messages), the number of elements on the display, or the pace of interaction.
Principles

Identify the Tasks

- Answer the question: What tasks are carried out by the user?
- Do this before you start designing the UI and do it in a structured and formal way
- This process often involves interviewing and observing the user, which also helps to understand the task frequencies and sequences
- Be careful regarding the extent of provided functionality (inadequate vs. cluttered)
- Start with high-level tasks, decompose them into smaller steps and finally atomic actions
- Choose the right level of granularity (e.g., depending on task frequencies)
Principles

Interaction Styles - Direct Manipulation

Creates a visual representation of the “world of action.” Keyboard entry of commands is replaced by selecting, dragging and pointing (e.g., Desktop Metaphor).

Pros: Users work with familiar objects in a familiar way, reduced time-to-learn, reduced error rate, high retention over time, encourages exploration

Cons: Not suitable for every scenario, may be too complicated and slow, may be hard to realize (program), requires pointing devices

Skill Level: Novice and intermittent users, possibly experienced frequent users
Principles

Interaction Styles - Menus

Users are provided with a list of items, may select an item appropriate to their task and observe the effect.

Pros: Tasks can be accomplished with little learning, no intense memorization is required, the decision making process is well structured (all choices are presented at once), keystrokes are reduced, good error-handling support through the use of dialog boxes

Cons: Danger of clutter (too many options, menus), may slow down frequent users

Skill Level: Novice and intermittent users, possibly experienced frequent users
Principles

Interaction Styles - Forms

The user enters data into text fields, text areas etc.

Pros: Makes data entry easier than with menus

Cons: The user is required to understand the field labels (must know what to enter where - e.g., the data format), increased error likeliness (hence the user must be capable to respond to error messages, which may require some training)

Skill Level: Knowledgeable intermittent users and expert frequent users
Principles

Interaction Styles - Command Languages

The user types commands in order to accomplish the required task.

Pros: Very flexible, supports user initiative, puts the user into control, enables convenient creation of user-defined macros, enables accomplishing tasks rapidly (if the syntax is memorized)

Cons: High error rates, poor error handling, requires substantial training and memorization

Skill Level: Expert frequent users
Principles
Interaction Styles - Natural Languages

Use natural language in order to accomplish the required task.

Pros: Users do not need to learn a syntax, may be successful in applications where the scope is limited and task sequences are clear from the beginning

Cons: Clarification dialog required (what to do next?), hard to determine the context, may be unpredictable, in applications with broad scope most likely less efficient

Skill Level: Possibly everyone, in particular intermittent users that are not able to learn complex syntax
Principles

The “8 Golden Rules” of UI Design

1. Strive for consistency
2. Cater for universal usability
3. Offer informative feedback
4. Design dialogs to yield closure
5. Prevent errors
6. Permit easy reversal of actions
7. Support internal locus of control
8. Reduce short term memory load
Principles

Error Prevention

General issues that should be considered in order to prevent errors:

- Use functionally organized screens and menus
- Design menu choices and commands to be distinctive
- Make it difficult for users to perform irreversible actions
- Provide feedback about the state of the UI
- Design for consistency of actions
- Consider universal usability

Other more specific issues:

- Prevent incorrect use of the UI (e.g., grey out unavailable functions, provide pull-down menus for specific data entry tasks, etc.)
- Aggregate a sequence of steps into one single action (e.g., specify a lengthy task once, save it and execute it again later)
- Provide reminders that indicate that specific actions are required to complete a task
Theories
Theories

Overview

- Are even more high-level than principles, largely very abstract

- 2 purposes (examples) of theories are:
  - Support collaboration and teaching through the provision of consistent terminologies for objects and actions (descriptive, explanatory)
  - May enhance the quality of the UI by providing predictions for motor-task performance, perceptual activities, execution times, error rates, emotional reactions, etc. (predictive); such characteristics enable designers, for instance, to better serve the users’ needs and compare proposed designs more objectively

- 4 different (classes of) theories (examples):
  - Stages-of-action models
  - Goals, Operators, Methods, and Selection rules (GOMS)
  - Widget-level theories
  - Context-of-use theories
Theories

Stages-of-Action Models (SOAM) - 1/2

- Deal with explaining the stages that users go through when using the UI. Example of a SOAM (by Norman):
  1. Forming the goal
  2. Forming the intention
  3. Specifying the action
  4. Executing the action
  5. Perceiving the system state
  6. Interpreting the system state
  7. Evaluating the outcome

- Based on this model pursue 4 design principles:
  - Make system state and action alternatives visible
  - Provide a good conceptual model with a consistent system image
  - The interface should include good mappings that reveal the relationships between stages
  - Users should receive continuous feedback
Theories

Stages-of-Action Models (SOAM) - 2/2

SOAMs help to describe user exploration of an interface; this may help, for instance, to study user failure.

Potential causes for failure are:

- The user forms an inadequate goal
- The user forms an adequate goal but does not find the right UI object to execute it (e.g., use of inadequate labels)
- The user forms an adequate goal but may not know how to specify or execute a desired action
- The user receives inappropriate feedback

Based on such observations UIs can be improved, which reduces failure probability.
Theories

GOMS: Goal, Operator, Methods, Selection

- Idea: Decompose user actions into small measurable steps
  - User formulates a goal (e.g., publish draft to obtain client feedback) and sub-goals (e.g., upload graphic)
  - User executes a variety of operators in order to change his/her mental state or to affect the task environment (e.g., locate file on hard drive - in the user’s mind)
    - Operator ≈ elementary perceptual, motor, or cognitive act
  - User achieves goals by using methods (e.g., move mouse to click appropriate button on the UI to start the upload procedure)
  - User bases his/her decision as to what method to use in order to accomplish a goal on selection rules (e.g., mouse vs. key)
- Works well for describing steps in the decision making process of users while they carry out interaction tasks
Theories

Widget-Level Theories (WLT)

- ≈ Alternative approach to “hierarchical decomposition”
  Hierarchical decomposition ≈ reduces complexity by breaking down tasks into atomic actions or complex UIs into atomic components
- WLTs deal with higher level components that are self-contained and reusable (≈ Widgets)
- Note: Widgets need to be well-defined and evaluated
- If Widgets become more advanced and widely used the initial investment in building them will pay off
Theories

Context-of-Use Theories

- Many theories are based on controlled lab experiments, isolated environments and only consider isolated phenomena
- Problem: the context (physical and social environment) in which the user operates may also play an important role in influencing usage patterns
- Context \(\approx\) Users’ interactions with other people (e.g., colleagues) and resources, unexpected interruptions, ...
- Theory: Knowledge is not always in the user’s mind but rather distributed in his/her environment (e.g., manuals, Internet, etc.)
- The relevance of Context-of-Use theories increases with growing use of mobile or context-aware devices
Questions?