design rules

Introduction

One of the central problems that must be solved in a user-centered design process is how to provide designers with the ability to determine the usability consequences of their design decisions.

We require ***design rules***, which are rules a designer can follow in order to increase the usability of the eventual software product.

We can classify these rules along two dimensions, based on the rule's authority and generality.

- By **authority**, we mean an indication of whether or not the rule must be followed in design or whether it is only suggested.
- By **generality**, we mean whether the rule can be applied to many design situations or whether it is focused on a more limited application situation.

different types of design rules

- ***Principles*** are abstract design rules, with high generality and low authority.
- derived from knowledge of the psychological, computational and sociological aspects of the problem domains and are largely independent of the technology;
  - they depend to a much greater extent on a deeper understanding of the human element in the interaction.
  - They can therefore be applied widely but are not so useful for specific design advice.

different types of design rules

- ***Standards*** are specific design rules, high in authority and limited in application
- It carry a much higher level of authority, it is more important that the theory underlying them be correct or sound.

- ***Guidelines*** tend to be lower in authority and more general in application.
- It is less abstract and often more technology oriented, but as they are also general, it is important for a designer to know what theoretical evidence there is to support them.
  - A designer will have less of a need to know the underlying theory for applying a standard.

design rules

In this chapter we will discuss:

- Principles of usability
  - general understanding
- Standards and guidelines
  - direction for design
- Design patterns
  - capture and reuse design knowledge

Principles to support usability

- The most abstract design rules are general principles, which can be applied to the design of an interactive system in order to promote its usability.

The principles we present are first divided into **three main categories:**

Learnability

the ease with which new users can begin effective interaction and achieve maximal performance

Flexibility

the multiplicity of ways the user and system exchange information

Robustness

the level of support provided the user in determining successful achievement and assessment of goal-directed behaviour

Principles of learnability

1. Predictability
   - Support for the user to determine the effect of future actions based on past interaction history
   - **operation visibility**
     - refers to how the user is shown the availability of operations that can be performed next
2. Synthesizability

- ability of the user to assess the effect of past operations on the current state.

Principles of learnability (ctd)

3. Familiarity - how prior knowledge applies to new system

- For a new user, the familiarity of an interactive system measures the correlation between the user's existing knowledge and the knowledge required for effective interaction.

- Some psychologists argue that there are intrinsic properties, or ***affordances***, of any visual object that suggest to us **how they can be manipulated**. **The appearance of the object stimulates a familiarity with its behavior.**
- In the design of a graphical user interface, it is implied that a soft button used in a form's interface suggests it should be pushed.
- Effective use of the affordances that exist for interface objects can enhance the familiarity of the interactive system.

Principles of learnability (ctd)

4.Generalizability-extending specific interaction knowledge to new situations

- Generalization can occur within a single application or across a variety of applications.
- For example, in a graphical drawing package that draws a circle as a constrained form of ellipse, we would want the user to generalize that a square can be drawn as a constrained rectangle.
- A good example of generalizability across a variety of applications can be seen in multi-windowing systems that attempt to provide cut/paste/copy operations to all applications in the same way (with varying degrees of success).

Principles of learnability (ctd)

5.Consistency

- likeness in input/output behaviour arising from similar situations or task objectives

-consistency in command naming, or consistency in command/argument invocation

Summary of principles affecting learnability

Principle	Definition	Related principles
Predictability	Support for the user to determine the effect of future action based on past interaction history	Operation visibility
Synthesizability	Support for the user to assess the effect of past operations on the current state	Immediate/eventual honesty
Familiarity	The extent to which a user's knowledge and experience in other real-world or computer-based domains can be applied when interacting with a new system	Guessability, affordance
Generalizability	Support for the user to extend knowledge of specific interaction within and across applications to other similar situations	–
Consistency	Likeness in input–output behavior arising from similar situations or similar task objectives	–

Principles of flexibility

1. Dialogue initiative

- freedom from system imposed constraints on input dialogue
- system vs. user pre-emptiveness

- The system can initiate all dialog, in which case the user simply responds to requests for information. We call this type of dialog system **pre-emptive**.
- For example, a modal dialog box prohibits the user from interacting with the system in any way that does not direct input to the box.

Principles of flexibility

2. Multithreading

- ability of system to support user interaction for more than one task at a time

Concurrent multi-threading allows simultaneous communication of information pertaining to separate tasks.

A very simple example can occur in the windowing system with an audible bell. You are editing a program when a beep indicates that a new electronic mail message has arrived.

Interleaved multi-threading permits a temporal overlap between separate tasks, but instructs that at any given instant the dialog is restricted to a single task.

Ex. Each window can represent a different task, for example text editing in one window, file management in another, a telephone directory in another and electronic mail in yet another.

Principles of flexibility

3.Task migratability

- passing responsibility for task execution between user and system

–Ex. Spell-checking a paper-Equipped with a dictionary, you are perfectly able to check your spelling by reading through the entire paper and correcting mistakes as you spot them.

–In the flight deck of an aircraft, there are so many control tasks that must be performed that a pilot would be overwhelmed if he had to perform them all.

- flight envelope is greatly automated

Principles of flexibility (ctd)

4.Substitutivity

- allowing equivalent values of input and output to be substituted for each other
- representation multiplicity; equal opportunity
- For example, in considering the form of an input expression to determine the margin for a letter, you may want to enter the value in either inches or centimeters.

Principles of flexibility (ctd)

5.Customizability

- modifiability of the user interface by user (adaptability) or system (adaptivity)

- **Adaptability** refers to the user's ability to adjust the form of input and output.
- This customization could be very limited, with the user only allowed to adjust the position of soft buttons on the screen or redefine command names.
- **Adaptivity** is automatic customization of the user interface by the system.
- Decisions for adaptation can be based on user expertise or observed repetition of certain task sequences.

Principles of robustness

1. Observability

- ability of user to evaluate the internal state of the system from its perceivable representation
- Five other principles: browsability; defaults; reachability; persistence; operation visibility

5 Principles of Observability

1. **Browsability** allows the user to explore the current internal state of the system via the limited view provided at the interface.

2. **Defaults** can assist the user by passive recall. It also reduces the number of physical actions necessary to input a value. Thus, providing default values is a kind of error prevention mechanism.

Ex.drop-down menus with values (lessen the error)

3. **Reachability** refers to the possibility of navigation through the observable system states.

Ex.t he user can navigate from any given state to any other state (pinch, zoom in-out, swipe)

5 Principles of Observability

4. **Persistence** deals with the duration of the effect of a communication act and the ability of the user to make use of that effect.

Ex. Visual communication can remain as an object which the user can subsequently manipulate.

-Visual information informs you of the incoming message then that will serve as a reminder that an unread message remains long after its initial receipt

-The effect of vocal communication, if you are informed of a new email message by a beep at your terminal, you may know at that moment and for a short while later that you have received a new message.

5.operation visibility-refers to how the user is shown the availability of operations that can be performed next

Principles of robustness

2. Recoverability

- ability of user to take corrective action once an error has been recognized
- reachability; forward/backward recovery; commensurate effort

Principles of robustness (ctd)

3.Responsiveness

- how the user perceives the rate of communication with the system
- Stability

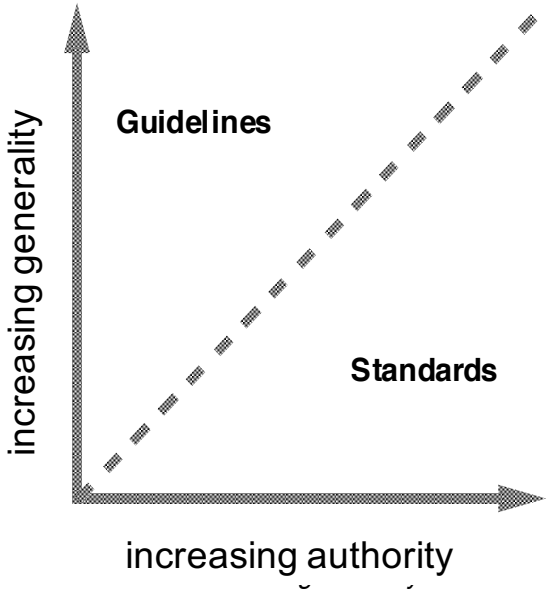
4.Task conformance

- degree to which system services support all of the user's tasks
- task completeness; task adequacy

Using design rules

Design rules

- suggest how to increase usability
- differ in generality and authority



Standards

- set by national or international bodies to ensure compliance by a large community of designers standards require sound underlying theory and slowly changing technology
- Standards can apply specifically to either the hardware or the software used to build the interactive system

Standards

- **ISO 9241** defines usability as effectiveness, efficiency and satisfaction with which users accomplish tasks
- **Effectiveness** The accuracy and completeness with which specified users can achieve specified goals in particular environments.
- **Efficiency** The resources expended in relation to the accuracy and completeness of goals achieved.
- **Satisfaction** The comfort and acceptability of the work system to its users and other people affected by its use.

Guidelines

- more suggestive and general
- many textbooks and reports full of guidelines
- abstract guidelines (principles) applicable during early life cycle activities
- detailed guidelines (style guides) applicable during later life cycle activities
- understanding justification for guidelines aids in resolving conflicts

Golden rules and heuristics

- "Broad brush" design rules
- Useful check list for good design
- Better design using these than using nothing!
- Different collections e.g.
  - Nielsen's 10 Heuristics
  - Shneiderman's 8 Golden Rules
  - Norman's 7 Principles

Shneiderman's Eight Golden Rules of Interface Design

1. *Strive for consistency* in action sequences, layout, terminology, command use and so on.
2. *Enable frequent users to use shortcuts*, such as abbreviations, special key sequences and macros, to perform regular, familiar actions more quickly.
3. *Offer informative feedback* for every user action, at a level appropriate to the magnitude of the action.
4. *Design dialogs to yield closure* so that the user knows when they have completed a task.
5. *Offer error prevention and simple error handling* so that, ideally, users are prevented from making mistakes and, if they do, they are offered clear and informative instructions to enable them to recover.
6. *Permit easy reversal of actions* in order to relieve anxiety and encourage exploration, since the user knows that he can always return to the previous state.
7. *Support internal locus of control* so that the user is in control of the system, which responds to his actions.
8. *Reduce short-term memory load* by keeping displays simple, consolidating multiple page displays and providing time for learning action sequences.

Norman's Seven Principles for Transforming Difficult Tasks into Simple Ones

1. *Use both knowledge in the world and knowledge in the head.*
2. *Simplify the structure of tasks.*
3. *Make things visible: bridge the gulfs of Execution and Evaluation.*
4. *Get the mappings right.*
5. *Exploit the power of constraints, both natural and artificial.*
6. *Design for error.*
7. *When all else fails, standardize.*

Norman's Seven Principles for Transforming Difficult Tasks into Simple Ones

1.Use both knowledge in the world and knowledge in the head.

People work better when the knowledge they need to do a task is available externally – either explicitly or through the constraints imposed by the environment.

But experts also need to be able to internalize regular tasks to increase their efficiency.

So ***systems should provide the necessary knowledge within the environment and their operation should be transparent to support the user in building an appropriate mental model of what is going on.***

2. Simplify the structure of tasks.

-Tasks need to be simple in order to avoid complex problem solving and excessive memory load.

There are a number of ways to simplify the structure of tasks:

1. to provide mental aids to help the user keep track of stages in a more complex task.
2. to use technology to provide the user with more information about the task and better feedback.
3. to automate the task or part of it, as long as this does not detract from the user's experience.
4. to change the nature of the task so that it becomes something more simple.

Norman's Seven Principles for Transforming Difficult Tasks into Simple Ones

3. **Make things visible**: bridge the gulfs of execution and evaluation. The interface should make clear what the system can do and how this is achieved, and should enable the user to see clearly the effect of their actions on the system.

4. **Get the mappings right**. User intentions should map clearly onto system controls. User actions should map clearly onto system events. So it should be clear what does what and by how much. Controls, sliders and dials should reflect the task – so a small movement has a small effect and a large movement a large effect.

Norman's Seven Principles for Transforming Difficult Tasks into Simple Ones

5**. Exploit the power of constraints, both natural and artificial**.

-Constraints are things in the world that make it impossible to do anything but the correct action in the correct way.

-A simple example is a jigsaw puzzle, where the pieces only fit together in one way. Here the physical constraints of the design guide the user to complete the task.

6. **Design for error**. To err is human, so anticipate the errors the user could make and design recovery into the system.

Norman's Seven Principles for Transforming Difficult Tasks into Simple Ones

7. **When all else fails, standardize**.

If there are no natural mappings then arbitrary mappings should be standardized so that users only have to learn them once.

HCI design patterns

- Patterns are an approach to capturing and reusing this knowledge – of abstracting the essential details of successful design so that these can be applied again and again in new situations.
- They capture the essential common properties of good design: they do not tell the designer *how* to do something but **what needs to be done and why**.
- They represent design knowledge at varying levels, ranging from social and organizational issues through conceptual design to detailed widget design.
- The concept of a pattern language is generative and can therefore assist in the development of complete designs.
- They are generally intuitive and readable and can therefore be used for communication between all stakeholders.

Summary

Principles for usability

- repeatable design for usability relies on maximizing benefit of one good design by abstracting out the general properties which can direct purposeful design
- The success of designing for usability requires both creative insight (new paradigms) and purposeful principled practice

Using design rules

- standards and guidelines to direct design activity