Design Rules

Design Rules

Roadmap

- Introduction
- Usability Principles
- Heuristics and Golden Rules
- AOB

Introduction

- Design rules (or usability rules) are rules that a designer can follow in order to increase the usability of the system/product e.g., principles, standards, guidelines.

- NB: Differences [based on level of abstraction/generality and level of authority]:
  Principles
  (e.g. interface should be easy to navigate)
  Abstract and have *high generality* & *low in authority*.
  Widely applicable and *enduring*.

  Guidelines
  (e.g. use colour to highlight links)
  Can guide/advise on how achieve a principle
  Narrowly *focused*.
  Can be too specific, incomplete, & hard to apply BUT they are more general and lower in authority than Standards (e.g. use colour RGB #1010D0 on home links) which are very specific & high in authority.

Introduction

- Principles:
  Example - usability principles by Dix et al (HCI book)

- Standards: They are often set by national (eg British Standards Institution) or international bodies (ISO).
  Example [of standards] - ISO 9241 "Ergonomic Requirements for Office Work with Visual Display Terminals (VDT)s"

- Guidelines:
  Example - Smith and Mosier's "Guidelines for User Interface Software" [MITRE Corporation 1986].

Introduction

- Design rules should be used early in the lifecycle [e.g., during the design; note that they can also be used to evaluate the usability of the system]

- We will:
  - First look at abstract principles for supporting usability
  - Later on, we will look at the most well used and well known sets of heuristics or 'golden rules', which tend to provide a succinct summary of the essential characteristics of good design (Nielsen's heuristics, Shneiderman's golden rules and Norman's principles [the last set, study on your own])

Usability Principles

by Dix et al (HCI book)

1. Learnability: the ease with which new users can begin effective interaction and achieve maximal performance.
2. Flexibility: the multiplicity of ways the user and system exchange information.
3. Robustness: the level of support provided to the user in determining successful achievement and assessment of goal-directed behavior.

Usability Principles

1. Learnability

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

- Predictability, Synthesizability, Familiarity, Generalizability, Consistency.

Usability Principles

Learnability (contd.)

- Predictability: support for the user to determine the effect of *future* action based on *past* interaction history.

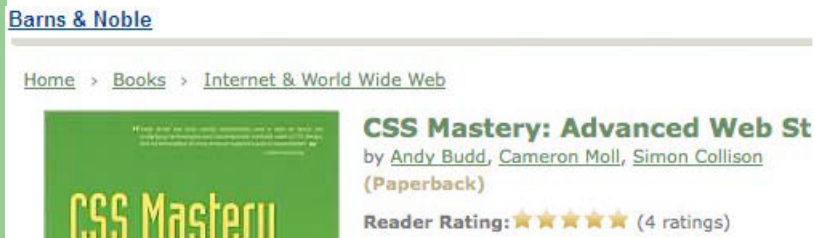
(can I 'tell' what *will* happen based on what I have gone through in the *past*?)

Usability Principles

Learnability (contd.)

- Synthesizability: support for the user to assess the effect of *past* operations on the *current* state.

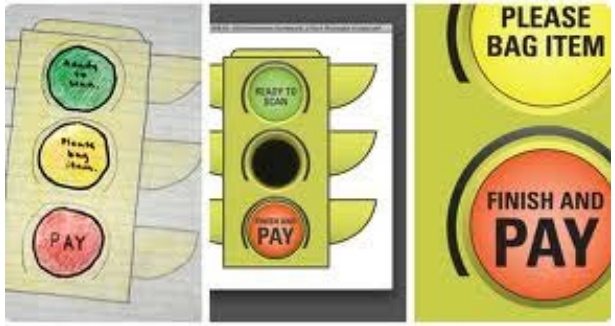
(can I 'tell' why *I am* here based on what I have gone through in the *past*?)

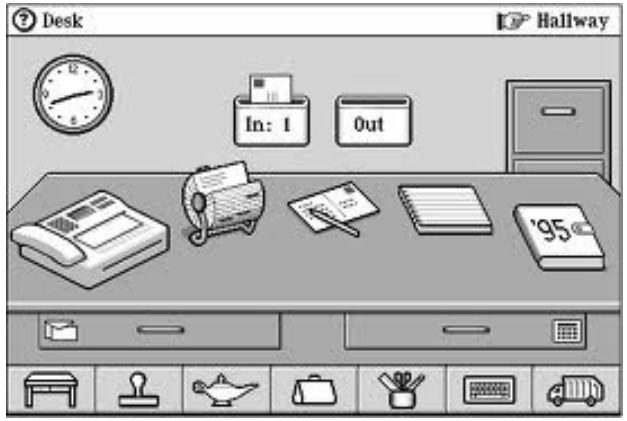


Usability Principles

Learnability (contd.)

- 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.

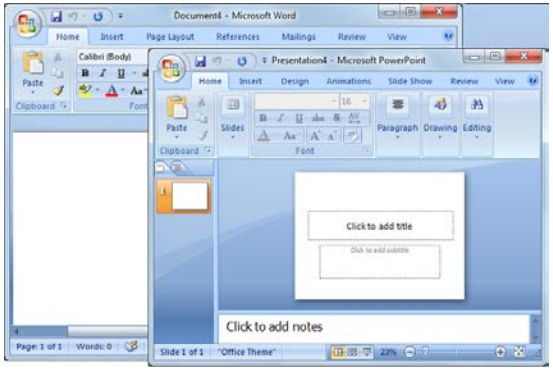




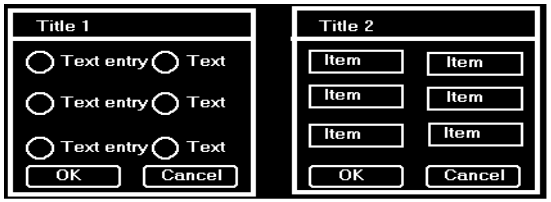
Usability Principles

Learnability (contd.)

- 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. (e.g. size, layout, colour, language, etc)



PS: Familiarity can be considered as 'consistency' wrt past real-world experience. Generalizability as 'consistency' wrt experience with the same system or set of applications on the same platform.

Usability Principles

2. Flexibility

The multiplicity of ways the user and system exchange information.

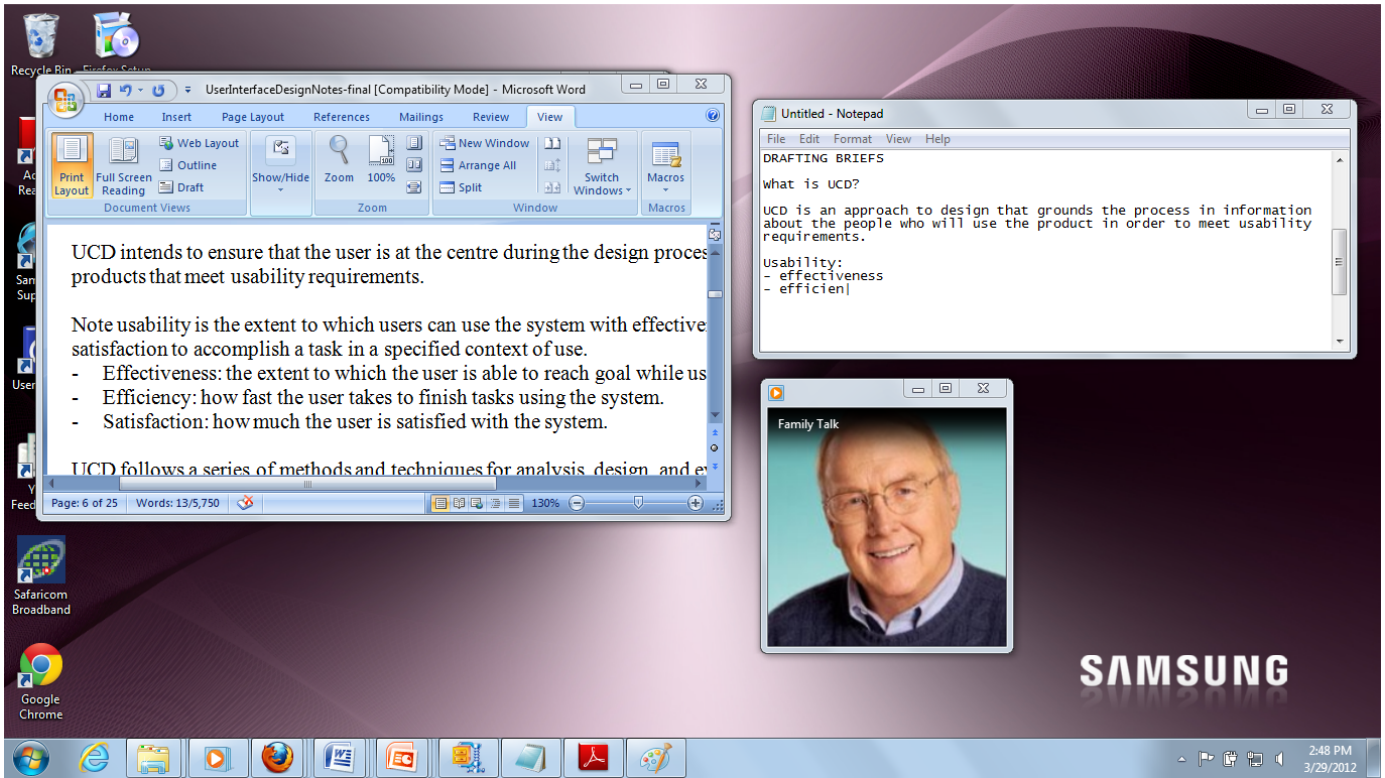
- Dialogue initiative, Multithreading, Task migratability, Subsitutivity, Customizability.

- Dialogue initiative: user freedom from artificial constraints on the input dialog imposed by the system;
  user vs system - who has the initiative in the dialog?

Usability Principles

Flexibility (contd.)

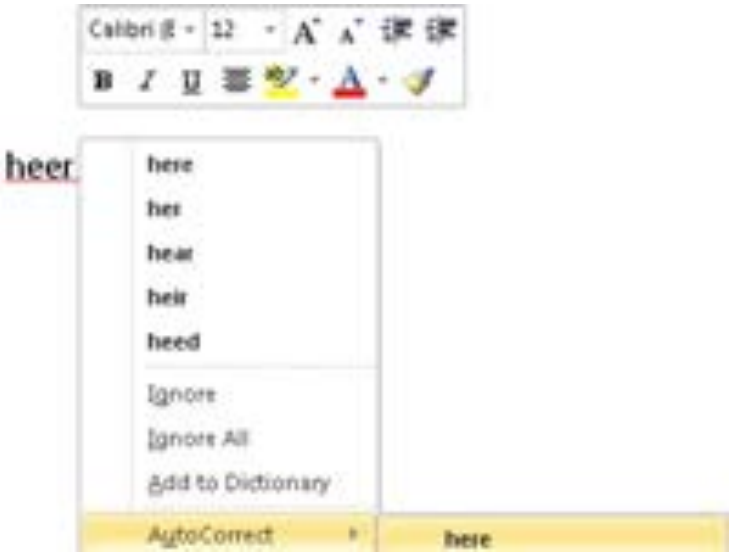
- Multithreading: the ability of the system to support user interaction for more than one task at a time.

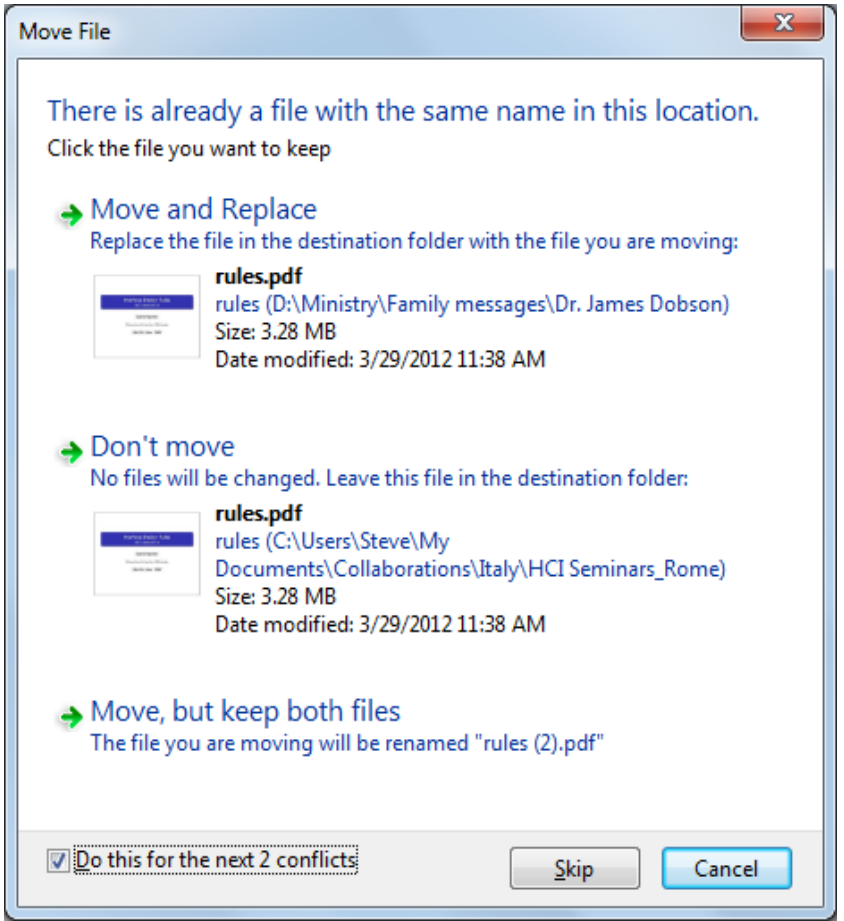


Usability Principles

Flexibility (contd.)

- Task migratability: the ability to transfer control for execution of tasks between the system and the user (consider e.g., spell-checking task).





Usability Principles

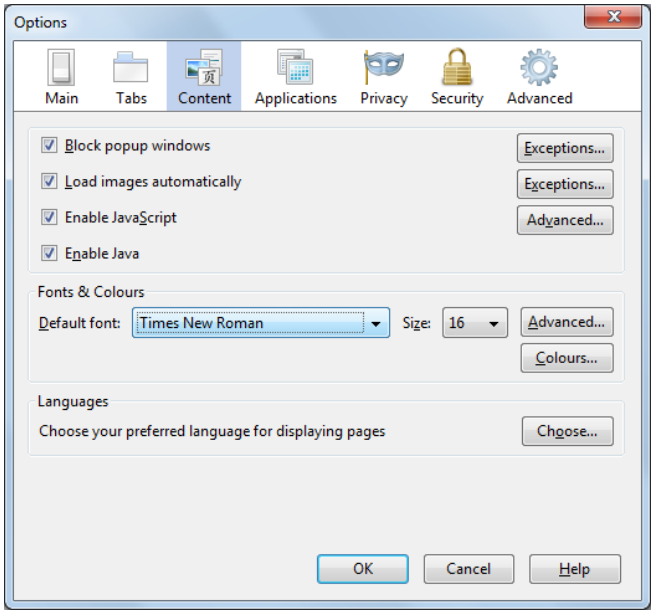
Flexibility (contd.)

- Substitutivity: the extent to which an application allows equivalent input and output values to be substituted for each other (values in input eg fractions/decimals, values in output eg both digital and analog, output/input eg output can be reused as input).

Usability Principles

Flexibility (contd.)

- Customizability: the ability of the user or the system to modify the user interface. (adaptability vs adaptivity) ?-initiated modification.



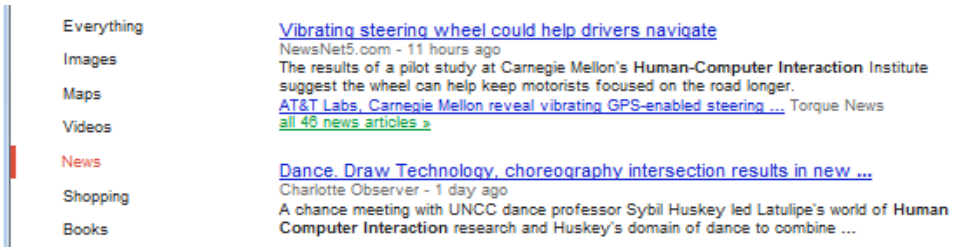
Usability Principles

3. Robustness

The level of support provided to the user in determining successful achievement and assessment of goal-directed behavior.

- Observability, Recoverability, Responsiveness, Task conformance.

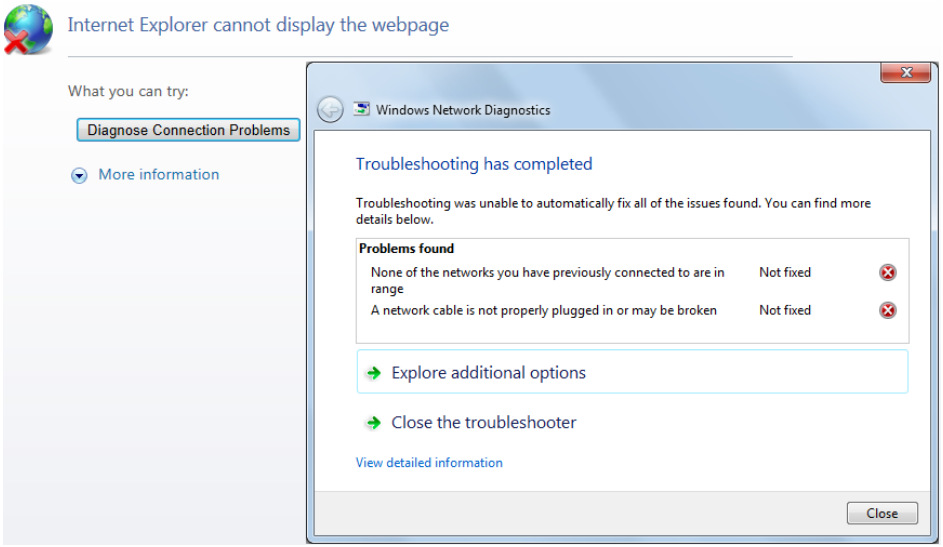
- Observability: the extent to which the user can evaluate the internal state of the system from the representation on the user interface.



Usability Principles

Robustness (contd.)

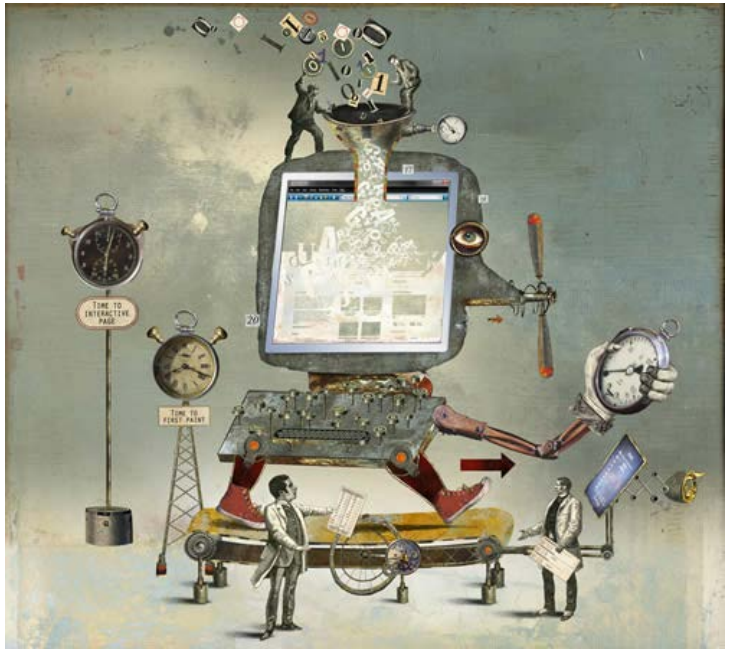
- Recoverability: the extent to which the user can reach the intended goal after recognizing an error in the previous interaction.



Usability Principles

Robustness (contd.)

- Responsiveness: a measure of the rate of communication between the user and the system.



Usability Principles

Robustness (contd.)

- Task conformance: the extent to which the system services support all the tasks the user would wish to perform and in the way the user would wish to perform.

Heuristics and Golden Rules

Jakob Nielsen's 10 Usability Heuristics

1. Visibility of system status: the system should always keep users informed about what is going on, through appropriate feedback within reasonable time.
2. Match between system and the real world: the system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.
3. User control and freedom: users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.
4. Consistency and standards: users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.
5. Error prevention: even better than good error messages is a careful design which prevents a problem from occurring in the first place.
6. Recognition rather than recall: make objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.

Heuristics and Golden Rules

Jakob Nielsen's 10 Usability Heuristics (contd.)

7. Flexibility and efficiency of use: accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.
8. Aesthetic and minimalist design: dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.
9. Help users recognize, diagnose, and recover from errors: error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.
10. Help and documentation: even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

Heuristics and Golden Rules

Ben Shneiderman's 8 Golden Rules

1. Strive for consistency: layout, terminology, command usage, etc.
2. Cater for universal usability: recognize the requirements of diverse users and technology. For instance add features for novices eg explanations, support expert users eg shortcuts.
3. Offer informative feedback: for every user action, offer relevant feedback and information, keep the user appropriately informed, human-computer interaction.
4. Design dialogs to yield closure: help the user know when they have completed a task.
5. Offer error prevention and simple error handling: prevention and (clear and informative guidance to) recovery; error management.
6. Permit easy reversal of actions: to relieve anxiety and encourage exploration, because the user knows s/he can always go back to previous states.
7. Support internal locus of control: make the user feel that s/he is in control of the system, which reponds to his/her instructions/commands.
8. Reduce short-term memory load: make menus and UI elements/items visible, easily available/retrievable, ...

Heuristics and Golden Rules

[Donald] Norman's 7 Principles [study on your own]

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.

AOB

Any Questions?