HCI: Guidelines for Design and Evaluation of Interfaces

Professor Ram Mohana Reddy Guddeti
Information Technology Department
NITK Surathkal, Mangalore, India

Learning Objective

- In the previous lectures, we discussed different GOMS family of models, and underlying laws. Through case studies we observed how these are used in evaluation.
- Here, we discuss Design Guidelines used in HCI so as enable the designers to evaluate existing interfaces or to conceptualise new ones.

The following are the standard guidelines

Shneiderman’s eight golden rules
Norman’s seven principles
Norman’s model of interaction
Heuristic evaluation
Nielsen’s ten heuristics

Ben Shneiderman's Eight Golden Rules for User Interface Design

Background

We use Dix et al's version of the Waterfall model to illustrate where exactly in the design cycle, the Rules - Guidelines & principles being discussed in this module, become important.

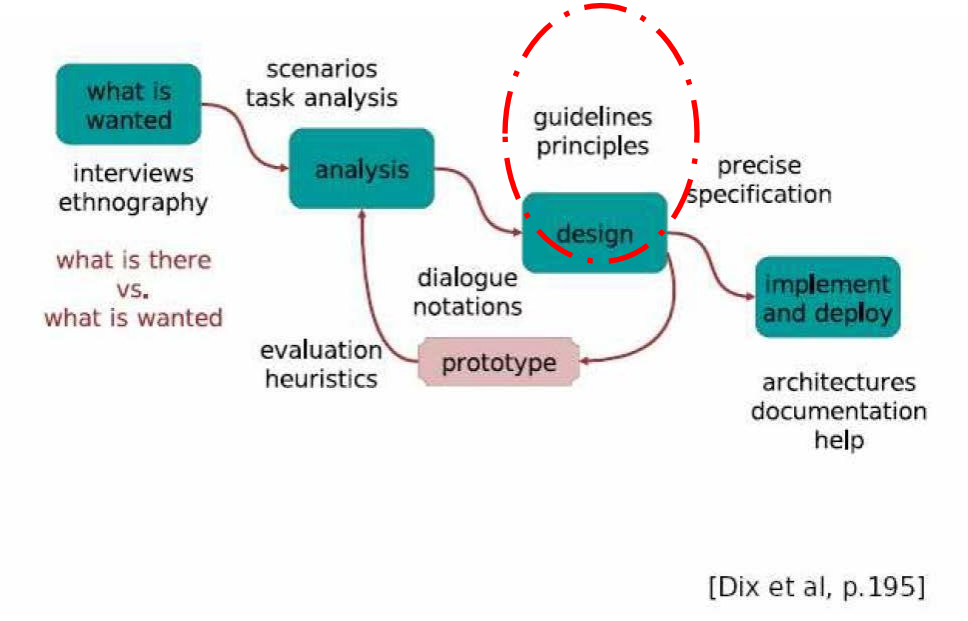
Design is both qualitative as well as quantitative.

The terms 'Guidelines' and 'principles' rather than precise' laws & rules' are used in Design.

Shneiderman formulated eight such guidelines that can be used in Interface designing.

HCI in the design process

- Waterfall model



[Dix et al, p.195]

Introduction

Ben Shniderman* consolidated known tacit knowledge and practice guidelines which are used intuitively by graphic Interface designers - into a set of eight general guidelines for the use of computer science specialist who were being introduced to Visual Graphic designers' work of designing interactive GUI** interfaces. Along with 'looks' the usability of a software depended on functionality.

**GUI:Graphic User Interface

* Bend Shneiderman founded the HCI Lab at the University of Maryland , USA. He is known for Nassi-Shnerderman diagrams used in the field of Software Engineering.

- There is ample empirical evidence published in HCI literature which collaborates and consolidates the applicability of the eight guide lines.
- These are intended more as guidelines rather than ‘rules’ to be strictly adhered to at every step.
- They are useful for designers as well as software engineers involved in design of interfaces.
- Using these eight guidelines it is possible to distinguish a good interface design from a bad one especially from the Human - User interaction point of view.
- These have been put forth in a concise and understandable manner by Ben Shneiderman.

- It needs to be noted that apart from these eight there are many more similar useful pointers available in HCI and Usability literature.
- While merely or blindly applying these eight guidelines is not necessarily going to grantee a good interface ‘design’, they are useful in heuristic evaluation to identify GUIs that fall out of normal ‘pattern’ . The guidelines can be used to rate GUI’s as good or bad.

The Eight ‘rules’ reproduced from published HCI literature are as follows.

1. Strive for Consistency
2. Cater to 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

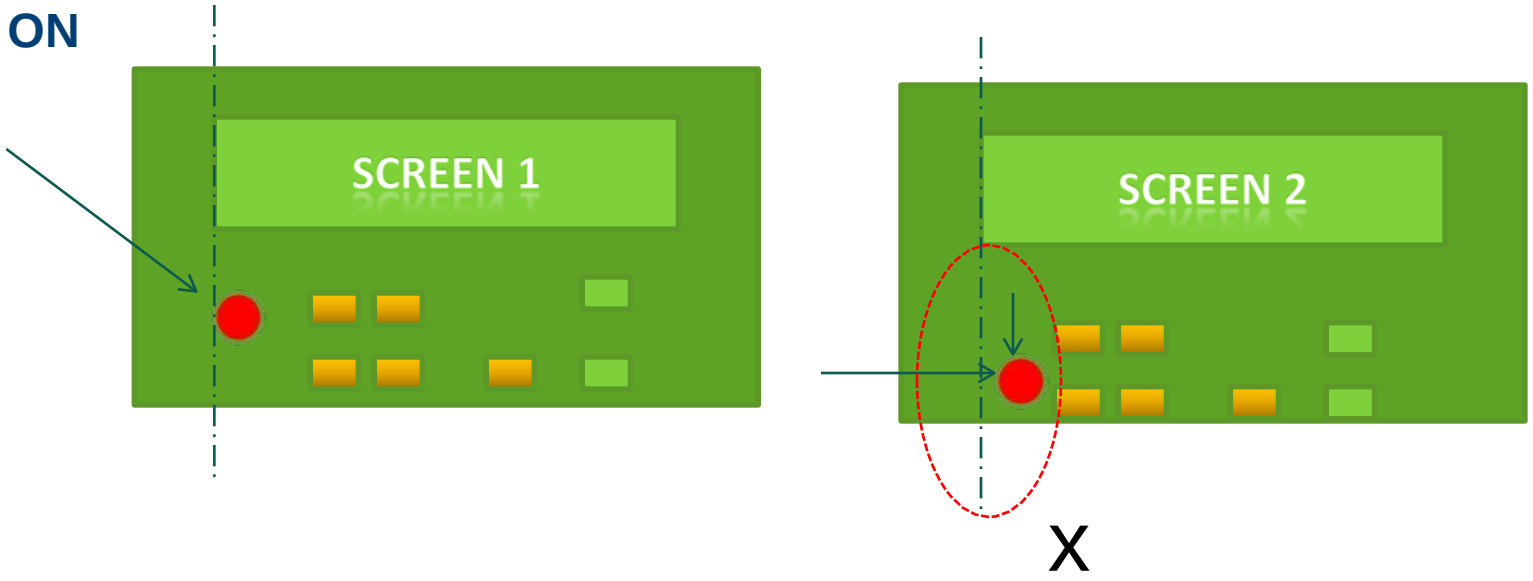
Explanations & Examples

1. Strive for Consistency

- Users need to be able to do the same thing the same way that they have been doing.- every time.
- Interfaces need to exhibit ‘consistent’ quality across screens/ applications both visually as well as behaviorally.
- Consistency leads to a pattern which is easier to handle cognitively.
- Consistency such as ‘similar sequence of actions in similar situations’ makes it easy to learn.

Consistency can be achieved through graphical elements such as fonts, colour, shape, position being consistently same in all menus & screens, across, categories for a particular software.

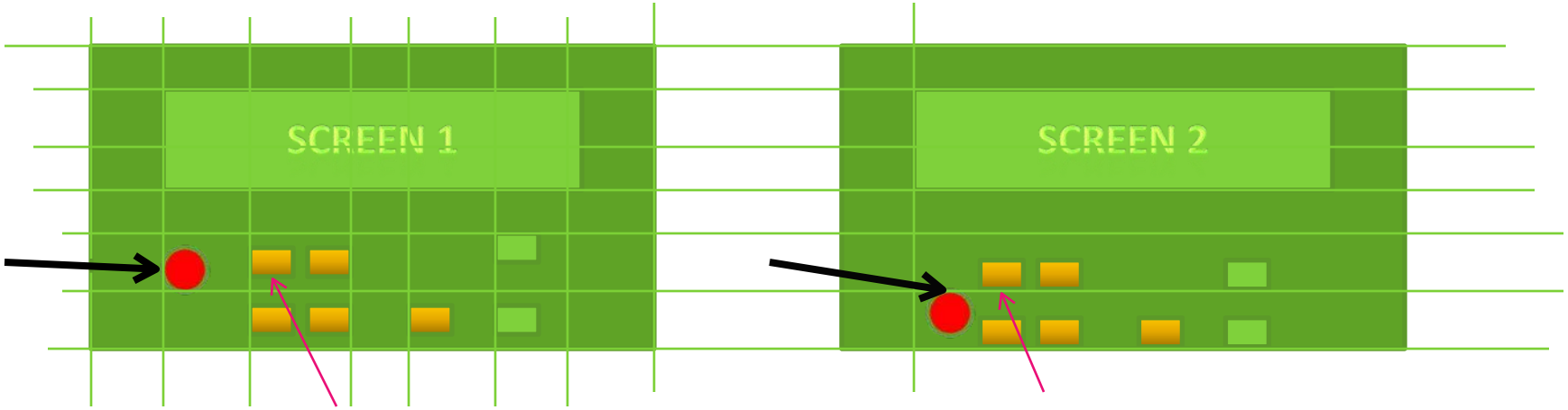
For example: If the **ON** button is on the right in the first screen and moves towards middle in the second screen then positional inconsistency is said to have occurred - however small the displacement is.



GUI designers use a simple technique to maintain consistency of control elements in successive screen.

Consistency (Contd...)

GUI designers use a background grid to place interactive elements in a consistent and orderly way so as to make them appear both physically as well a visually at the same place across the entire software package.



Inconsistent positioning of GUI elements is evident when observed against a grid. Grids are used as background reference to place the elements consistently

In case of certain exception in maintaining consistency are required to be made in a subsequent screen, they should be such that they are comprehensible, distinct and limited in number.

2. Cater to wide range & type of Users

1. Strive for Consistency
2. Cater to 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

Universal design strives to cater to as wide a range of human users of different characteristics (age, culture, educational level, disability) with a single design.

While this may not be feasible/possible in all contexts, Shniderman's rule none the less needs to be followed so as not to leave out taking into consideration a section of users, other wise competent, who cannot use the interface due to no fault of theirs.

Users: **Novice, Intermediate** and **Experts.** Experts tend to use lesser actions at a faster pace. Abbreviations short cuts keys etc are some of the techniques used.

Interfaces need to cater to all levels & classification of users: novice to experts.

3. Offer Informative Feedback

1. Strive for Consistency
2. Cater to 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

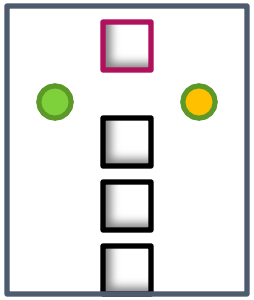
- Interfaces need to not just to be communicative but also need to inform the ‘user‘ in terms of learning & feed back which tells them that they are proceeding in the right direction.
- For every action of the user there needs to be a feedback – only then ‘interaction’ (in HCI) is said to take place. Specific error messages composed in a appositive tone give affirmative feedback without having to feel punitive.
- Unless the user gets a feed back we cannot proceed or becomes unsure of the correctness of the action.

3. Design Dialogs to yield closure

- In an interaction - dialogue needs to have a closure which is recognized by the user as end of an action.
- Sequence of actions need to proceed in a dialogue by engaging the user in a step by step manner.
- Like in a mathematical expression, every enclosing bracket needs a corresponding closing bracket. So also subsequence of actions needs to be grouped with intermittent closing of each sub group followed finally by a closer action of the group.

Ex: A message at the end of a sequence of events gives a feed back & closure of sending a SMS**.**

Example 2: Un-closed dialogue



Press ON button

Look at the green lamp.

If green glows press next push button - yellow lamp will glow

Push 3rd button and continue till green lamp stops glowing.

End of task……

An Example of a closed dialogue:

Press ON button

Look at the green lamp.

If green glows press 2nd push button and yellow lamp will glow.

Press 3rd button and continue with other 3 buttons till green lamp stops glowing.

When Yellow lamp stops glowing it indicates sequence over.

End of task.

Notice the yellow lamp feed back dialogue above being not closed ?
What happens to yellow lamp ? Did it stop glowing? or why it continues glowing when the task is over ?

….. are some of the questions that may arise due to non closure of dialogues which can lead to confusion for a user

1. Strive for Consistency
2. Cater to 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

5. Prevent Errors

Interfaces need to minimize the errors. Human Computer dialogue can be designed to minimize and prevent errors made by users.

There cloud be many reasons for users errors but the user himself or herself is not one of them! Users can make errors while interacting with computers as well as while inputting / interpreting information.

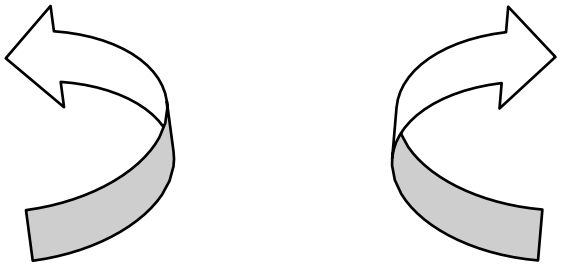
Even if the user makes an error the system needs top be designed to detect it, take corrective or precautionary steps to arrest it. It also needs to offer a way out for recovery from the error.

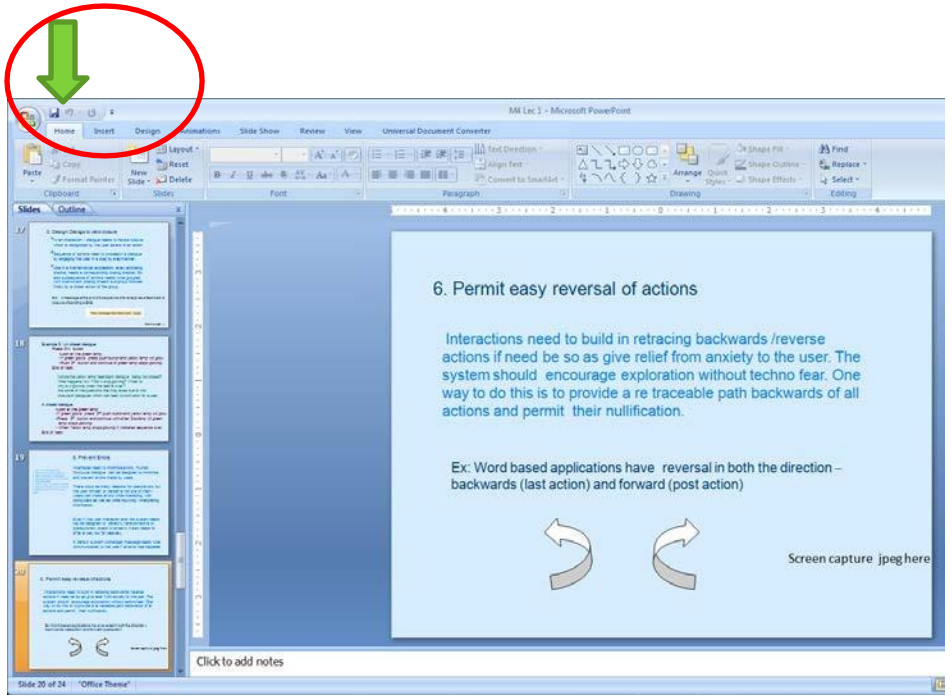
A default system unchanged message needs to be communicated to the user if an error has happened.

6. Permit easy reversal of actions

Interactions need to build in retracing backwards /reverse actions if need be so as give relief from anxiety to the user. The system should encourage exploration without techno fear. One way to do this is to provide a re traceable path backwards of all actions and permit their nullification.

Ex: This PPT application has reversal in both the direction – backwards (last action) and forward (post action)

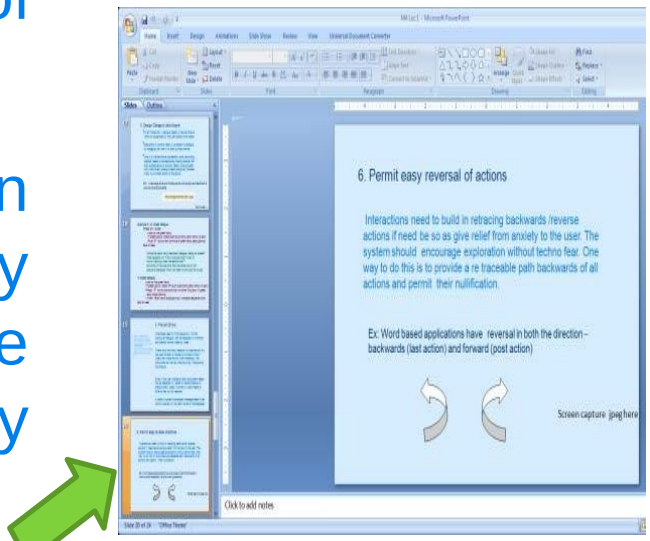




7. Support internal locus of control

Allow user to always feel ‘in control’ of the system and of the situation.

Make the user aware that he/she is in control. User should believe that they are controlling the system and not the other way around. This is achieved by more opportunities for ‘interactions’.



The bearing of where the user presently is helps the user to orient or reorient the interaction. The user should never be allowed to feel lost.

8. Reduce short term memory load

1. Strive for Consistency
2. Cater to 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

Care not load the cognitive short term memory of the user by expecting user to remember several sequences , actions and their consequences at a time. Means loading their short term memory while interacting.

94 56 781029

Easier to remember if chunked into smaller sets

94 56 7 810 29

Miller's* 7 chunks of information is often prescribed as a solution to limit short term memory. In psychological experiments it has been found that the short term memory can hold 7 +- 2 bits called chunks of information. Long sequential actins requiring more than 7 chunks need to be broken down into smaller chunks.

*G.A. Miller; The Magical number seven, plus or minus two: some limits on our capacity to process information. Psychplogical review, 63(2):81-97,1956.

Each of these Shneiderman's rules were examined with the examples

1. **Strive for Consistency**
2. **Cater to 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**

Assignment

Choose any common software interface. Analyze its interfaces by navigating to find out whether it adheres to the eight Shneiderman Rules or not. Use a Novice User as your reference.

Example: Excel Sheet . User:
10th standard student.

Present your findings in terms of number of violations per rule for the chosen software.

References:

1.Shneiderman. B.; Designing the user interface: Strategies for effective Human Computer Interaction; Addsion-Wesley Publishers Treading MA. 2004.)

2.Designing the user interface: Strategies for effective Human Computer Interaction ; Ben Shneiderman and Catherikne Plaisant , Addison-Wesley Publishers Treading MA. 2010.)