M4 L4 Notes

Nielsen’s Ten Heuristics

Objective :

In this Lecture you will be introduced to another set of well known interface design guidelines proposed by Jacob Neilsen .

Their application to specific situations like a web site will be discussed in the background of user centered Designing framework.

The procedure of conducting a Heuristic Analysis will be dealt with in the next lecture number 4.

Introduction

Jakob Nielsen (working along with Molich in 1990) proposed a set of ten guidelines that can be used as Principles of Design for a new Interface. These also can be used as Heuristics for evaluating an Interface.

Since the ten guidelines were more in the spirit of rules of thumbs than specific rules, they are called as ‘ Heuristics’ rather than rules or laws that hold true in every case.

Introduction

Heuristics means “ rules of thumb”. These ten ‘rules of the thumb’ were derived after careful research by Nielsen who after conducting a factor analysis of 249 useability problems, came up with ten simply stated guidelines in 1994.

Nielsen’s heuristics method are empirically based derivations. Widely used by Usability professionals (which includes Interface designers) they are a means of quickly identifying likely design problems in an application’s human interface. Because of its simplicity and low cost it is preferred evaluation technique at the earliest design stages by HCI professional.

Heuristics evaluation is s systematic process of inspection of a user interface for usability problems. It is both a “ before design finalisation’ predictive method as well as an ‘ after design ‘ evaluation and rating method.

The goal of heuristic evaluation is to find the usability problems in design

So that they can be attended to as integral part of an iterative design processes.

Heuristic evaluation method involves having a small set of evaluators (5 to 7) examine the interface and judge its compliance with recognized usability principles such as Nielsen’s ten Usability principles.

The Ten Principles Listed

1. **Visibility of system status**

2. **Match between system and the real world**

3. **User control and freedom**

4. **Consistency and standards**

5. **Error prevention**

6. **Recognition rather than recall**

7. **Flexibility and efficiency of use**

8. **Aesthetic and minimalist design**

9. **Help users recognize, diagnose, and recover from errors**

10. Provision of Help and documentation

Each principle explained. (Copyright © 2005 by Jakob Nielsen. ISSN 1548-5552)

Visibility of system status

The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

Explanation: This means the user needs to be constantly made aware of his/her interaction with the interface while interacting. The control response ratio (input – output time) need to be as small as possible. Any interface needs to communicate that it is in a ready state to be operated upon – at the start of an interaction cycle. (A glowing LED / flashing element indicating that the interface is live.)

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.

Explanation: Technical jargon or using terms like ‘ Initiate’ or ‘Load’ in pace of ‘ Start’ contributes to initial mismatch between the users cognitive process and machines feed back dialogue.

An interface need to allow smooth transition from contextual ‘reality’ world to artificial machine world o9rin other words from ‘ reality’ to ‘digitality’.

Tendency to use programming language and syntax on the display , while understandable to the software programmer, will certainly be a mismatch to a user.

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.

Explanation: Sequential thought process in a user that follows a simple everyday human logic need to be reflected in the dialogue between the machine and the user via the interface.

This leads to the feeling in the user that the machine is obeying the user’s command and therefore the user is in control during the interaction. Being in control implies that one can choose to stop interacting an time rather than be forced or trapped by the interface into inaction.

Consistency and standards

Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

Explanation: Within an interface if multiple words or actions are used to mean the same thing , it only leads to confusion in the user due to perceived lack of consistency. Interaction pattern gets disrupted. When pattern becomes complex, user's cognitive load increases.

Consistency in dialogue as well as in visual elements is achieved by specifying and adhering to a dictionary of words / labels/ symbols/ colors which together form a ' standard' – a prescribed set – compulsorily to be followed.

Error prevention

Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.

Explanation: To err is human. Errors can happen regardless the level of expertise of the user or familiarity of the interface. A good principle of design is to seek out error prone interactions , build in error prevention within the dialogue. Forewarning, restricting , prompting , retracing or recovery routes , etc are means of addressing errors. Errors lead to a situation wherein users feel subdued by a machine. Anticipating for errors and incorporating preventive measures ensures fear free and ego free user thereby giving importance to H in HCI through I.

Recognition rather than recall

Minimize the user's memory load by making 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.

Explanation: Reduction on cognitive load during the interaction ensures that the user is not asked to rely on means and methods that extract human cost. If an interface requires specilised training and use of memory to operate it will be quickly abandoned by the human user.

Analogy, metaphor, symbols, sounds, etc are used as design elements in an interface to ease recall thereby eliminating the need for ‘ thinking while interacting’ and memory loads for the user.

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.*

Explanation: Once a user becomes adept at using an interface , he/she upgrades into a higher level user from a novice. Such users will always seek to complete the task faster . An interface need to allow this. It needs to be flexible and make it possible for the user to adopt quicker dialogues through shortcuts. The user feels efficient as well as proficient.

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.*

Explanation: Visual clutter in he interface only adds to inefficiency however impressive it is visually.   Simplicity is equal to efficiency is equal to elegance is equal to beauty is the aesthetic algorithm in minimalism. Use of least number of elements (minimalism) is more ‘scientific’ rather than ‘artistic’.

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.*
>
> **Explanation:** No body likes to be loudly informed that he/ she has erred. Error messages need to be disused as suggestions / prompts and precise instructions so as to be able to correct the error and recover. The learning component in errors so that the user recognizes the error as it is being made, or recognizes the reason why the error happened in the first place – helps the user learn.

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.*
>
> **Explanation:** This again is to assist the user learn and understand the dialogue between the user and the machine or understand where what went wrong or aid recall during memory lapses due to long useage time gaps. Adequate support system when the user wants and at the point where the user wants it - is a good principle of Interface design.

Conclusions:

These ten heuristics of usability help in refining a potential design into a good design. They ensure that interfaces evolve in the right direction.

These rules of the thumb act a check list to evaluate a design .

References

- Molich, R., and Nielsen, J. (1990). Improving a human-computer dialogue, *Communications of the ACM* **33**, 3 (March), 338-348.
- Nielsen, J., and Molich, R. (1990). Heuristic evaluation of user interfaces, *Proc. ACM CHI'90 Conf.* (Seattle, WA, 1-5 April), 249-256.
- Nielsen, J. (1994a). Enhancing the explanatory power of usability heuristics. *Proc. ACM CHI'94 Conf.* (Boston, MA, April 24-28), 152-158.

Nielsen, J. (1994b). Heuristic evaluation. In Nielsen, J., and Mack, R.L. (Eds.), *Usability Inspection Methods*, John Wiley & Sons, New York, NY.