User Interface Design
Course Objectives

• To understand the need of UI Design

• To appreciate the considerations behind good UI design

• To be aware of general UI design principles

• Do’s and Don’ts List of UI design

Course Objective

In order to understand the issues surrounding user interfaces, it helps to understand what, exactly, a user interface is, and how they've evolved to the point they're at today.
Session Plan

- What is a User Interface?
- History of the evolution of user Interface
- Place Process of UI design in the project life-cycle
- Process of UI design
- Elements of GUI
- Good and bad UI’s
- UI Design - Principles and guidelines
- Heuristic Evaluation
- Case Study

1 day course for 3 hours
References

- “User Interface Design ” found online at http://cfg.cit.cornell.edu/cfg/design/contents.html
- Shneiderman Ben, *Designing the user interface*, Pearson Education Asia, 2000
What is a User Interface?

• **User interface**
  - In short decides the man-machine interaction
  - Man-machine interaction involves how people communicate, interact and make use of the systems running on the machine
  - Is the front end of the system for the users
  - Generally used in terms to computers

• **Users interact with the computers with the help of**
  - video screens
  - keyboards
  - pointing devices

What is User Interface?

The user interface, or human interface determines how people communicate with, interact with, and use machines.

Generally machines refers to computers. For example voice mail system. It communicates the information to the user in the form of audio messages, and the user communicates information back by pressing touch-tone buttons.

Any machine that requires interaction with human beings will have some sort of user interface.

Traditional computers communicated to the users with the help of punch cards. Modern computers interact with users with the help of video screens and typewriter-style keyboards and pointing devices.

The user interface should represent the capabilities of the entire system. The more complex the system, the more important the user interface. The user interface should help the user build an intuitive sense of how the system works. When a system feels natural to use, the user interface is doing a good job. A good user interface helps tailor the system to the user.
Evolution of User Interface

Stage 1
- Communication is through flashing lights & mechanical Switches

Stage 2
- Introduction of the first modern electronic computers

Stage 3
- Invention of video screens
- Introduction of assembly language and higher-order languages

The Evolution of user interface

Stage 1:
The only mode of communication was – mechanical. Computer uses switches for communication which are mechanical in nature and to grab the attention of the users flashing lights were used. Highly trained specialists were able to communicate with a computer. There was an improvement with the invention of punched cards, but still they were inefficient.

Stage 2:
In 1940s with the introduction of the first modern electronic computers, programs were written in machine language. This was an improvement but still only a computer specialist would actually communicate directly with a computer.

Stage 3:
Invention of video screens made the users to communicate directly with a computer. Assembly language and higher-order languages such as Basic, Fortran, and Cobol were introduced during this stage. But they are command-language interfaces Video screens were limited to displaying only the characters that were found on the keyboard. Users were required to memorize commands that were generally tailored more to the computers than the users. Example: Mainframe, Unix environments.
Evolution of User Interface

Stage 4

- Invention of mouse
- Graphical user interface

Stage 4:

With the invention of mouse new interface technologies emerge. Graphical user interface became more popular. Interfaces are separated from the application functions. With the term GUI, we denote here any kind of user interface that makes use of graphics features. The use of graphics to communicate information to the users would be done visually in addition to textually. Users need not memorize and manually enter commands from a virtually unlimited set of options. In this way the interface was focused on the needs of the human beings. This significantly reduced the training that was necessary to use a computer, and for the first time uninitiated users were able to become productive almost immediately.
The process of User Interface Design

- Importance of Quality User Interfaces
- Phases (life cycle) of UI Design
  1. Discovery
  2. Design
  3. Delivery

Quality User Interface
UI Design isn't just arranging stuff on a page or in a dialog box
It may seem a simple and side aspect of the whole application; in fact it is, perhaps, the most important part of the whole system.
User Interface is also called as the front end of the system
And front end always the first impression about the system. Remember you do not have another chance to make the first impression.

To achieve Quality User Interface, All phases of the UI design need to be strictly followed
1. Discovery

- **Read the documents**
  - Marketing Requirements
  - Functional Specification

- **Interviews**
  - Interview different users of the system
  - Understand the differences among the requirement of different users of the UI

- **Brainstorming**
  - Brainstorm among the team members

- **Visual design**
  - Design sample screens

- **Verify Design directions**
  - Test the sample screens

Discovery stage of UI design corresponds to requirement analysis phase of SDLC (Software Development Life Cycle)

In this step, the research needs to be carried out regarding the various users of the system & business requirements analysis related to UI.

In this step, we also need to figure out who the users are, the expertise level & the language of the users

We also need to find out what is their objective being using this UI.
2. Design

- **Concept development**
  - Develop according to the idea which is best amongst the output of the last phase

- **Create a site map**
  - Create the outline by taking help of Flow diagrams

- **Prototyping**
  - Create interactive working prototype

- **Visual design**
  - Develop complete design model using the selected tool for design

- **Usability testing**
  - Test the Visual Design

**Concept development**

In concept development we choose one amongst the various discussed ways of design. Then we develop some initial ideas and select the best features from each one.

**Various tools are available for visual design**

E.g. Composer
Frontpage
PACBASE etc
3. Delivery

- Create Design specification
  
  - In this stage we create the document containing the specification of the design which comes out as an output of the previous stages
  
  - Help the programmers to understand and develop the code according to the specification

This stage corresponds to the release stage of the software life cycle.
In short, in this phase we are delivering the Design which is the output of the last phase.
Elements of User Interface Design

Windows

- Window1

Window2

Events

An "event", with respect to user interfaces, is any function initiated by the user.

Windows

The most widely used element Graphical User Interface (GUI) is a window. Most of the time the window is a container containing other components inside it and having an “x” button which allows you to close the window.

A window can also have inner windows as shown in Window2 on the slide

Events

Selecting something from a pull-down menu, clicking a button or a checkbox, and closing a window are all examples of events
Push Buttons

A push button is usually a rectangular (traditionally rounded rectangle and the latest fashion is accurate rectangle) that appears on a panel with some sort of label. ex: the button with label "open Application"

You can also put an icon on the button. It is usually called as pictorial button.

Thought most of the languages allow an icon on a Button, you need to make sure the type of the picture file that you want to put on the pictorial button is correct. Usually small gif files are accurate to be used on the buttons.

Clicking on a push button will cause some sort of action. Sometimes a panel will have a "default" button, like the sign in button shown on the slide (it appears with an enhanced border). This action associated with this button will be called when the Enter key is pressed.}
Elements of User Interface Design

- Pull-down Menus / Drop-down Menus

Pull-down Menus / Drop-down Menus
Pull-down menus are menus that the user can "pull down" from the menu bar. On some platforms these are called "drop down" menus because the user does not need to hold the mouse button down in order for the menu to remain visible.
Radio Buttons

Radio buttons look like an "o" and will present you with multiple choices. You will be allowed to select one option out of multiple choices.

When you click them, they act like an old-time AM radio button. You select one and they rest are unselected.
Elements of User Interface Design

- Icons

Icons are small pictures that usually represent objects in the physical world. They are used as metaphors for functions or actions. Icons should be "clickable" for initiating an event.

For ex: The Airplane icon takes you to a system which can be used for ticket bookings. The graphics file (airplane picture) which is displayed on the icon usually has .ico extension.
Checkboxes

A checkbox is a small square with some sort of label beside it. Clicking on a checkbox will cause an 'X' ✓ (or some special symbol) to appear in the box. Clicking it again will cause that symbol to disappear.

Multiple checkboxes can be selected at time.
Elements of User Interface Design

- **Scrolling Lists**

Scrolling lists are lists of elements that appear in a box with a scroll bar on the side, allowing the user to scroll through the elements in the list. In this way the list can contain more elements than can be displayed at any one time. Clicking on an element in the list will cause that element to become selected, as shown in the slide.
Elements of User Interface Design

• Single Line / Multi Line Text Field

Text Field
A text field is a space in which the user can enter the text.

Text fields are usually rectangular in shape. It may or may not have a border.

If a text field is meant to be multi line, it can optionally have a scroll bar to allow it to display a subset of the contents but hold all the information at one time.

In many languages single line text box and multi-line text box are provided as different controls.

While as few language provide a single control with a property (single/multi-line) which is to be set by programmer.
Elements of User Interface Design

• **Popup List**

![Popup List Image]

A popup list allows you to store several values in a box. When you click on this box it displays all the values, from which you will be allowed to select one value. This selected item will be displayed in the box. If you have several items for the user to pick from, better use a popup list.

• **Spin Boxes**

![Spin Boxes Image]

Spin box allows you to store values in a box. It has small up and down arrows for increasing and decreasing the value. Users can either click the arrow buttons or type a choice. Having too many increments forces users to click for long periods of time. Use a combo box instead.
Tabbed Pane is mostly used by complex windows/dialog boxes (having lots of Information) where the UI is categorized and displayed on various tabs.

The currently selected tab can be highlighted with different color. Ex: the home tab in the tabbed pane is highlighted.
The home page of a personal web site is shown in the slide

The interface consists of
a) scattered text
b) series of pictures pasted above the text in the illogical order.
c) random animation given to these pictures

The animation makes it impossible for a reader to read anything on the screen for more than few seconds.

The data to be shown on the screen was fairly simple and consistent, but the design of the interface was very poorly done.

The jazzy effects used in incorrect manner have spoiled the design of the web page.
Some good examples of bad UI

The slide shows a lovely dialog box of UI that can earn the title of "the most annoying thing I've seen today or the good example of bad UI"

Here are some reasons that make it bad:
The text displayed on dialog box is not user friendly.
There are spell mistakes in the text.
The Multi-line Textbox is not the accurate component to be used for this message.
The cancel button made red and shifted to the left
Neither button is made the default button
Etc.
Example of Simple, Good UI Design

Yahoo ID: [field] @ yahoo.com
   Examples: "daryman88" or "free2winxye"
Password: [field]
   Must be six characters or more
Re-type Password: [field]

Choosing your ID
You will use this information to access Yahoo! each time. Capitalization matters for your password!

This is example of good design because
a) Sample of the data to be entered is shown
b) The validation rules are put up upfront
c) A caution note is given in a separate box

UI should be designed like designing a building.
Each element should be kept at proper place
   Required functionality should be achieved.
Few more examples of Good UI Design

The popular web site shown in the slide gives the e-view of the newspaper which is quite similar to the hard-copy view.

You are allowed to navigate through page numbers and the complete page is shown on the browser.

Magnification facilities allow you enlarge the news items.
Good Vs Bad UI Design

- Which UI can be called as bad UI?
  - Does not meet the expectations of the target user
  - Does not serve the purpose for which it was made

- Which UI can be called as good UI?
  - Meets the expectation of technically skilled as well as unskilled users expectations
  - Serves the purpose for which it was made

Example of a bad interface?
How to create good interfaces?
Collect specific details such as:
What are the user's goals?
What are the user's skills and experience?
What are the user's needs?

Follow the guidelines of UI design after collecting this information.
Design guidelines for User Interface

• Anticipate users wants and needs
  ➢ It’s always “U” which comes before “I” in UI Design 😊

• Use Objects that are consistent with human perception
  ➢ E.g., Folders, Documents

Anticipate users wants and needs
User need not search for the required information on the screen
Remember Give utmost attention to the comforts of the user while designing the UI.
Design guidelines for User Interface (Contd…)

• **Responsiveness**
  - respond immediately to the user

• **Provide a user with status mechanism to keep user informed and aware. Keep this information within easy view**
  - E.g., Status bar in windows

![Status bar example](image)

• **Choose appealing metaphors that the user can grasp instantly and remember.**
  - E.g., “Briefcase” used in windows maps to the real world

Details of status mechanism
- Acknowledge all button clicks within 50 milliseconds
- For any action taking more than 0.5 seconds display an hourglass
- Use Animated Progress indicators
- Display messages indicating potential wait time
- Make client system beep and give visual indications on culmination of lengthy Processes (>10 sec)
Design guidelines for User Interface (Contd…)

- **Keep it Simple and Flexible**
  - Just enough functionality for the user to accomplish their goal. More can be obtained as shown in the diagram.

  If a user needs more options from this menu then the same would be shown after clicking this symbol.

Simplicity
Keep the interface as simple as possible, A complex interface is difficult to use for a novice user, and often cumbersome to an expert user.

Allow the users to control the interface depending upon the skill level and personal preferences
Layer the interface to reduce the complexity from new users, and provide navigation short cuts for experts users
Example: Short vs long menus in windows
Design guidelines for User Interface (Contd…)

- **Consistency**
  - Use the ability of user to reuse what a he/she has already learned while accomplishing future tasks.
  - E.g. make sure that the same symbols/icons are used at various places in the set of User Interfaces
  - Make objects consistent with its behaviour. Objects which act differently should look differently
  - Also make sure that same color combinations and interface styles are used across the set of screens
  - Style-sheets can be used to achieve this

Style-sheets: It’s always better to separate the style part (color, font etc) from the code that is required to generate the screen elements and the code that is having the data to be displayed
To achieve this the styles can be coded in a separate document called as Style-sheet
The screen elements can be bound to style-sheet by a reference

The obvious advantage of this concept is that if two/more screens refer to the same style sheet, consistency of style is achieved

Let us look at this from another viewpoint.
An Job alert is to be displayed in different formats like SMS for a cell phone and Email message in outlook.
Because the alert data is going to remain constant, the alert screen is created without style information.
E.g.
Style-sheet1 is created to convert the alert data into SMS format
Style-sheet2 is created to convert the alert data into email message format
If the alert screen refers style-sheet 1, then the output would be according to style-sheet1 i.e. SMS format
If the alert screen refers style-sheet 2, then the output would be according to style-sheet2 i.e. Email format

There are two major advantages of this approach
1) Modularity : Style and the Data is separated
2) Reusability : The Same Data can be used to generate different output depending on the different style sheets used for the same
Design guidelines for User Interface (Contd…)

• Robustness

- Allow for common errors and mistakes
- On errors do not make the user recycle or disconnect

Robustness

An sensitive system will quickly discourage users. The interface should prevent errors from occurring, but if they do, it should allow users to quickly recover.

Protect the user information

This feature is sometimes called as forgiveness.

For example, if a user has forgotten to save the changes while exiting, then the same can be requested to the user.
Design guidelines for User Interface (Contd…)

- **Defaults**
  - Set Default button that gets hit when the user presses enter
  - Provide default text in the input controls
    - Should be easy to replace with custom content on default fields

- **Readability**
  - Font sizes should be large enough
  - High contrasts text colours with background

Defaults
Better to place defaults under headings such as “Standards” etc.

Readability
Font Size: If the users are likely to be aged persons, give particular attention to font sizes
Colour Blindness: Most of the colour blindness is associated with recognition of primary colours (Red/Blue/Green)
Hence, always use clear and secondary colours for colour coding schemes
Design guidelines for User Interface (Contd…)

• Efficiency of the User
  • Do not compromise Users productivity to Computers productivity

• Keep the user occupied
  ➢ That’s why there are mirrors in the elevators of tall buildings ☺

Do not compromise Users productivity to Computers productivity:
It is preferable to minimize the number of keystrokes even if it calls for some additional steps which
the application has to perform

Keep the user occupied:
E.g. Offer engaging text messages to the user
While doing this, consider optimization across the Organization and not just for a set of users.
Design guidelines for User Interface (Contd…)

• Use large objects for important functions. E.g.

Fitt’s Law
“The time to acquire a target is a function of the distance to and the size of the target”
Thus
Use large objects for important functions
Use pinning actions for the corners, sides, top and bottom of your display

Questions?
User may need to know information such as - Whether the first time the user has been in the system, where the user is with respect to the site map and where he is heading to, where the user was when he entered this session and left the previous session etc.
Good Documentation

There should be user manuals, readme, installation notes on-line tutoring and of course the Help.
Set of Points to remember for good UI

- Use Simple and Natural Dialog
- Speak the Users’ Language
- Minimize the Users’ Memory Load
- Be Consistent
- Provide Feedback

Use Simple and Natural Dialog
Examples: Messages presented on top imply priority. Color indicates status. Blinking attracts attention - don’t over-do it.

Speak the Users’ Language
Use words and concepts from the users’ world. Don’t use system-oriented terms.

Minimize the Users’ Memory Load
Recognition better than recall. Don’t make the user remember until it is not needed. Instructions for use should be retrievable

Be Consistent
Same action or commands always elicits the same response. Users should be able to learn an action sequence in one part of the system, and apply it again to get similar results in other places.

Provide Feedback
Let users know what effect their actions have on the system
Set of Points to Remember for good UI (Contd…)

- Provide Clearly Marked Exits
- Provide Shortcuts
- Provide Good Error Messages
- Prevent rather than cure
- Provide Help and Documentation

Provide Clearly Marked Exits
If users get into part of the system that doesn’t interest them, they should always be able to get out quickly without damaging anything.

Provide Shortcuts
Shortcuts or accelerators can help experienced users avoid lengthy dialogs and informational messages that they don’t need.

Provide Good Error Messages
Good error messages let the user know what the problem is and how to correct it.

Prevent rather than cure
Whenever you write an error message, you should always ask, can this error be avoided?

Provide Help and Documentation
Such information should be easy to search, be focused on the user’s task, list concrete steps to be carried out, and not be too large.
Some More Tips for UI Design

- **Clustering**
  - Similar controls should be used across screens

- **Visibility**
  - Make the frequently used controls easily accessible

- **Intelligent Consistency**
  - Use similar screens for similar functions. Example: Help button (/F1)

- **Using Color only as a Supplement**
  - Never as them as primary means to differentiate

**Clustering**
Organize the screen into visibly separate blocks of similar controls, preferably with a title for each block. Controls include menus, dialog boxes, on-screen buttons, etc.

**Visibility**
Visibility Reflects Usefulness
So make the frequently used controls obvious, visible, and easy to access. Conversely, hide or shrink controls that are used less often. Example: Dialog boxes, toolboxes.

**Color as a Supplement**
Don’t rely on color to carry information. Use it sparingly to emphasize information provided through other means
Color is easy to misuse. Different colors mean different things to different people, and varies from culture to culture
Good strategy - design in black and white first, make sure interface works, then add color for warning or informational messages so that they stand out.
Cross-Cultural User Interface Design is a subject in itself and has various aspects like Internationalization, Localization, various techniques like locale, resource bundles etc.

Localization is the process of adapting your application for the countries in which it will be used. Localization might involve changing error messages, calendars etc where necessary depending on the country/region for which the software is developed.

The slide shows the example of Google.co.in which is presented in the local language.
Some More Tips for UI Design : Metaphors

• Metaphor is the Concrete model
  – With which the user is familiar.
  – Is most well-known example is the desktop metaphor.
  – Serve as Ways of presenting user objects so that they mimic real life objects in the user’s everyday world.
  – Make software easier to use because users can recognize and generalize

Metaphors are very much part of our everyday lives. Tape deck metaphor is a good example. The use of metaphors can have a significant impact upon end-users. Within an end-user interface metaphors can provide cues for the recognition of symbolism. Also, the appropriate use of metaphors can provide a framework in which users can identify the functional meanings of the symbols.

There are several factors to consider when using a metaphor:

Use the same metaphor throughout the interface. Even better would be to use the same metaphor spread over several applications. When a metaphor is used in several applications implement it in a standard way.

Avoid similar metaphors when using multiple metaphors. The different metaphors that are used should be clear and distinct from each other. Obviously, similar metaphors are likely to cause confusion which might be equally applicable under either metaphor—especially when the functionality of the metaphor is different.

If the natural function of the software itself is easier to comprehend than any real-world analog of it, better do not use a metaphor. Don’t strain a metaphor in adapting it to the program’s real function. Nor should you strain the meaning of a particular program feature in order to adapt it to a metaphor.

Incorporating a metaphor includes certain risks. In particular, whenever physical objects are represented in a computer system, we inherit not only the beneficial functions of those objects but also the detrimental aspects.

Be aware that some metaphors don’t cross cultural boundaries well. For example, Americans would instantly recognize the common U.S. Mailbox (with a rounded top, a flat bottom, and a little red flag on the side), but there are no mailboxes of this style in Europe.
Simple guidelines for metaphors

- Match your major user objects
- Simple is better
  - The simplest metaphors are often the most powerful.
- Your metaphor does not need to be unique
  - A metaphor works best if it is common. If someone else has already used a notebook idea, then it is OK for you to also use it.
  - Look at the users’ real world for metaphor ideas
  - Be flexible

Match your major user objects

Major user objects are the objects in your interface that users have to manipulate in order to get their work done.

Look at your major user objects and the actions users will need to perform to them. Then decide what kind of metaphor would work best for those objects and actions.

Be flexible

Try out your metaphors and get feedback before you make any final design decisions. Be willing to change.
## Examples of Applications and Associated Metaphors

<table>
<thead>
<tr>
<th>Application Area</th>
<th>Metaphor</th>
<th>Similar to</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating environment</td>
<td>Desktop</td>
<td>Office tasks, file management</td>
</tr>
<tr>
<td>Spreadsheet</td>
<td>Ledger paper</td>
<td>Numerical tables</td>
</tr>
<tr>
<td>Object-Oriented environment</td>
<td>Physical world</td>
<td>Real-world behavior</td>
</tr>
<tr>
<td>Hypertext</td>
<td>Index cards</td>
<td>Flexible organization of structured text</td>
</tr>
<tr>
<td>Multimedia environments</td>
<td>Rooms (each associated with a Different medium/task)</td>
<td>Spatial structure of buildings</td>
</tr>
<tr>
<td></td>
<td>Downtown/Malls</td>
<td>Spatial layout of city blocks, shopping center</td>
</tr>
</tbody>
</table>
Web UI Design Issues

The typical design process of a web site should include:

- Establishing the goals of the site and the target user for the site.
- Developing design requirements and technological and artistic needs
- Planning the organization of information of the site and the way that users will navigate through this information
- Creating the pages of the site, (collecting and scanning existing materials that may have been already made for other media.)
- Developing all the graphical elements and artwork for the site as well as the overall style.

Other issues to be considered

Is the site international?
People will access the site from all over the world and will have special needs. These needs have to be understood and accordingly, the site should be designed.

To whom is the site accessible?
The site should be accessible to anyone on the internet.

Will the site ever appear in the Search Engines?
A web site isn't worth much if it can't be searched. The keywords must be developed and design techniques should be used in such way that it is much more likely that the page will appear on the top search engines.
Web UI Design Issues

Goals for Web Sites

- Easy navigation
- Quick Loading
- Deliver required information
- Look nice
- Differentiation from other sites
- Render accurately on all platforms and browsers

Goals for Web Sites

Easy navigation
The contents of the site should be so structured that people will find what is being offered without getting lost or frustrated.

Quick Loading
Special techniques should be employed for minimizing the time it takes for the average person to load the web pages, especially during peak usage times.

Deliver required information
Information in a site should be so designed so that users can quickly find what they are looking for.

Look nice
Page layout, graphic design, illustration and animation skills should be used to create the look wanted without sacrificing performance.

Differentiation from other sites
Tricks should be used to make the site fun and interesting so that the audience would want to keep coming back.

Render accurately on all platforms and browsers
The site should be so designed so as to look right on all browsers (e.g. Macintosh, PC/Windows, Unix, WebTV) and on many different browser configurations.
Heuristic Evaluation

Heuristic evaluation (Nielsen and Molich, 1990; Nielsen 1994) is a usability engineering method for finding the usability problems in a user interface design so that they can be attended to as part of an iterative design process.

Heuristic evaluation involves having a small set of evaluators examine the interface and judge its compliance with recognized usability principles (the "heuristics").

Details of Heuristic evaluation

Heuristic evaluation is difficult for a single individual to do because one person will never be able to find all the usability problems in an interface.

Favorable experience from many different projects has shown that different people find different usability problems. Therefore, it is possible to improve the effectiveness of the method significantly by involving multiple evaluators.
Heuristic Evaluation (Contd..)

Figure in the slide shows an example from a case study of heuristic evaluation where 19 evaluators were used to find 16 usability problems in a voice response system allowing customers access to their bank accounts (Nielsen 1992).

Each of the black squares in the figure indicates the finding of one of the usability problems by one of the evaluators. The figure clearly shows that there is a substantial amount of non-overlap between the sets of usability problems found by different evaluators.

Each row represents one of the 19 evaluators and each column represents one of the 16 usability problems. Each square shows whether the evaluator represented by the row found the usability problem represented by the column: The square is black if this is the case and white if the evaluator did not find the problem. The rows have been sorted in such a way that the most successful evaluators are at the bottom and the least successful are at the top. The columns have been sorted in such a way that the usability problems that are the easiest to find are to the right and the usability problems that are the most difficult to find are to the left.

It is certainly true that some usability problems are so easy to find that they are found by almost everybody, but there are also some problems that are found by very few evaluators.

Furthermore, one cannot just identify the best evaluator and rely solely on that person’s findings.
1) It is not necessarily true that the same person will be the best evaluator every time.
2) Some of the hardest-to-find usability problems (represented by the leftmost columns in Figure 1) are found by evaluators who do not otherwise find many usability problems.
Heuristic evaluation is performed by having each individual evaluator inspect the interface alone. Only after all evaluations have been completed are the evaluators allowed to communicate and have their findings aggregated.

Typically, a heuristic evaluation session for an individual evaluator lasts one or two hours. Longer evaluation sessions might be necessary for larger or very complicated interfaces with a substantial number of dialogue elements, but it would be better to split up the evaluation into several smaller sessions, each concentrating on a part of the interface.

During the evaluation session, the evaluator goes through the interface several times and inspects the various dialogue elements and compares them with a list of recognized usability principles (the heuristics). These heuristics are general rules that seem to describe common properties of usable interfaces. In addition to the checklist of general heuristics to be considered for all dialogue elements, the evaluator obviously is also allowed to consider any additional usability principles or results that come to mind that may be relevant for any specific dialogue element.

In principle, the evaluators decide on their own how they want to proceed with evaluating the interface. A general recommendation would be that they go through the interface at least twice, however. The first pass would be intended to get a feel for the flow of the interaction and the general scope of the system. The second pass then allows the evaluator to focus on specific interface elements while knowing how they fit into the larger whole.

Since the evaluators are not using the system as such (to perform a real task), it is possible to perform heuristic evaluation of user interfaces that exist on paper only and have not yet been implemented (Nielsen 1990). This makes heuristic evaluation suited for use early in the usability engineering lifecycle.

The output from using the heuristic evaluation method is a list of usability problems in the interface with references to those usability principles that were violated by the design in each case in the opinion of the evaluator. It is not sufficient for evaluators to simply say that they do not like something; they should explain why they do not like it with reference to the heuristics or to other usability results. The evaluators should try to be as specific as possible and should list each usability problem separately. For example, if there are three things wrong with a certain dialogue element, all three should be listed with reference to the various usability principles that explain why each particular aspect of the interface element is a usability problem. There are two main reasons to note each problem separately: First, there is a risk of repeating some problematic aspect of a dialogue element, even if it were to be completely replaced with a new design, unless one is aware of all its problems. Second, it may not be possible to fix all usability problems in an interface element or to replace it with a new design, but it could still be possible to fix some of the problems if they are all known.
Questions?
Summary

• What is User Interface
• Need of User Interface Design
• Various Elements of GUI
• Guidelines of User Interface Design
• Heuristic Evaluation
  – a strategy to evaluate the User Interface