NATIONAL OPEN UNIVERSITY OF NIGERIA

CIT 811

User Interface Design and Erogonomics

Module 1

CIT 811 User Interface Design and Ergonomics Module 1

Course Developer/Writer
Dr. A. S. Sodiya, University of Agriculture, Abeokuta

Course Coordinator
A. A. Afolorunso, National Open University of Nigeria

Programme Leader
Prof. MoniOluwa Olaniyi

Credits of cover-photo: Henry Ude, National Open University of Nigeria

National Open University of Nigeria - University Village, Plot 91, Cadastral Zone, Nnamdi Azikiwe Expressway, Jabi, Abuja-Nigeria

oer.nou.edu.ng.nou.edu.ng centralinfo@nou.edu.ng
oer.nou.edu.ng oerunit@noun.edu.ng OER repository
Published in 2013, 2021 by the National Open University of Nigeria

© National Open University of Nigeria 2021

This publication is made available in Open Access under the Attribution-ShareAlike4.0 (CC-BY-SA 4.0) license. By using the content of this publication, the users accept to be bound by the terms of use of the Open Educational Resources repository oer.nou.edu.ng of the National Open University of Nigeria.

The designations employed and the presentation of material throughout this publication do not imply the expression of any opinion whatsoever on the part of National Open University of Nigeria concerning the legal status of any country, territory, city or area or of its authorities, or concerning the delimitation of its frontiers or boundaries. The ideas and opinions expressed in this publication are those of the authors; they are not necessarily those of National Open University of Nigeria and do not commit the organization.

How to re-use and attribute this content

Under this license, any user of this textbook or the textbook contents herein must provide proper attribution as follows: "First produced by the National Open University of Nigeria" and include the NOUN Logo and the cover of the publication. The repository has a version of the course available in ODT-format for re-use.

If you use this course material as a bibliographic reference, then you should cite it as follows: "Course code: Course Title, Module Number, National Open University of Nigeria, [year of publication] at nouonline.net

If you redistribute this textbook in a print format, in whole or part, then you must include the information in this section and give on every physical page the following attribution: Downloaded for free as an Open Educational Resource at nouonline.net If you electronically redistribute part of this textbook, in whole or part, then you must retain in every digital file (including but not limited to EPUB, PDF, ODT and HTML) the following attribution:

Downloaded for free from the National Open University of Nigeria (NOUN) Open Educational Resources repository at oer.nou.edu.ng

Unit 1 Fundamentals of User Interface Design

1.0 Introduction

Having read through the course guide, you will have a general understanding of what this unit is about and how it fits into the course as a whole. This unit describes the general fundamentals of user interface design.

2.0 Objectives

At the end of this unit, you should be able to:
- explain the term user interface design
- identify the significance of user interface
- explain the history behind user interfaces
- describe the modalities and modes of user interface.

3.0 Main Content

3.1 User Interface

The **user interface** (also known as Human Machine Interface (HMI) or Man-Machine Interface (MMI) is the aggregate of means by which people—the *users*—interact with the *system*—a particular machine, device, computer program or other complex tool.

User interface is the point at which a user or a user department or organisation interacts with a computer system. The part of an interactive computer program that sends messages to and receives instructions from a terminal user.

In computer science and human-computer interaction, the *user interface (of a computer program)* refers to the graphical, textual and auditory information the program presents to the user, and the control sequences (such as keystrokes with the computer keyboard, movements of the computer mouse, and selections with the touchscreen) the user employs to control the program.

User Interface Design or **User Interface Engineering**
This is the design of computers, appliances, machines, mobile communication devices, software applications, and websites with the focus on the user's experience and interaction. The goal of user interface design is to make the user's interaction as simple and efficient as possible, in terms of accomplishing user goals—what is often called user-centred design. Good user interface design facilitates finishing the task at hand without drawing unnecessary attention to it. Graphic design may be utilised to apply a theme or style to the interface without compromising its usability. The design process must balance the technical functionality and visual elements (e.g., mental model) to create a system that is not only operational but also usable and adaptable to changing user needs.

Interface design is involved in a wide range of projects from computer systems, to cars, to commercial planes; all of these projects involve much of the same basic human interaction yet also require some unique skills and knowledge. As a result, designers tend to specialise in certain types of projects and have skills centered around their expertise, e.g., software design, user research, web design, or industrial design.

3.2 Significance of User Interface

To work with a system, users have to be able to control the system and assess the state of the system. For example, when driving an automobile, the driver uses the steering wheel to control the direction of the vehicle, and the accelerator pedal, brake pedal and gearstick to control the speed of the vehicle. The driver perceives the position of the vehicle by looking through the windscreen and the exact speed of the vehicle by reading the speedometer. The *user interface of the automobile* is on the whole composed of the instruments the driver can use to accomplish the tasks of driving and maintaining the automobile.

The term *user interface* is often used in the context of computer systems and electronic devices. The user interface of a mechanical system, a vehicle or an industrial installation is sometimes referred to as the **Human-Machine Interface** (**HMI**). HMI is a modification of the original term MMI (Man-Machine Interface). In practice, the abbreviation MMI is still frequently used although some may claim that MMI stands for something different now. Another abbreviation is HCI, which is more commonly used for Human-Computer *Interaction* than Human-Computer *Interface*. Other terms used are Operator Interface Console (OIC) and Operator Interface Terminal (OIT).

3.3 Types of User Interfaces

Currently (as at 2009) the following types of user interface are the most common:

- **Graphical User Interfaces** (GUI) accept input via devices such as computer keyboard and mouse and provide articulated graphical output on the computer monitor. There are at least two different principles widely used in GUI design: Object-Oriented User Interfaces (OOUIs) and application oriented interfaces. Examples are Windows Operating System, LabView, etc.

- **Web-based User Interfaces** or **Web User Interfaces** (WUI) accept input and provide output by generating web pages which are transmitted via the Internet and viewed by the user using a web browser program. Newer implementations utilise Java, AJAX, Adobe Flex, Microsoft, NET, or similar technologies to provide real-time control in a separate program, eliminating the need to refresh a traditional HTML based web browser. Administrative web interfaces for web-servers, servers and networked computers are often called control panels.

User interfaces that are common in various fields outside desktop computing:

- **Command Line Interfaces**, where the user provides the input by typing a command string with the computer keyboard and the system provides output by printing text on the computer monitor. Examples are MS-DOS and UNIX interface. This interface is also used for system administration tasks, etc.

- **Tactile Interfaces** supplement or replace other forms of output with haptic feedback methods. This is used in automated simulators and so on.

- **Touch User Interface** are graphical user interfaces using a touchscreen display as a combined input and output device. It is used in many types of point of sale, industrial processes and machines, self-service machines etc. Examples are touch screen monitors.

Other types of user interfaces are:

- **Attentive User Interfaces** manage the user attention deciding when to interrupt the user, the kind of warnings, and the level of detail of the messages presented to the user.

- **Batch Interfaces** are non-interactive user interfaces, where the user specifies all the details of the *batch job* in advance to batch processing, and receives the output when all the processing is done. The computer does not prompt for further input after the processing has started.

- **Conversational Interface Agents** attempt to personify the computer interface in the form of an animated person, robot, or other character (such as Microsoft's Clippy the paperclip), and present interactions in a conversational form.

- **Crossing-based Interfaces** are graphical user interfaces in which the primary task consists in crossing boundaries instead of pointing.

- **Gesture Interface** are graphical user interfaces which accept input in form of hand gestures, or mouse gestures sketched with a computer mouse or a stylus.

- **Intelligent User Interfaces** are human-machine interfaces that aim at improving the efficiency, effectiveness, and naturalness of human-machine interaction by representing, reasoning, and acting on models of the user, domain, task, discourse, and media (e.g., graphics, natural language, gesture).

- **Motion Tracking Interfaces** monitor the user's body motions and translate them into commands, currently being developed by Apple.

- **Multi-screen Interfaces** employ multiple displays to provide a more flexible interaction. This is often employed in computer game interaction in both the commercial arcades and more recently the handheld markets.

- **Non-command User Interfaces** allow users to infer their needs and intentions, without requiring explicit commands formulation.

- **Object-Oriented User Interface (OOUI)** examples of OOUI are:

Reflexive User Interfaces where the users control and redefine the entire system via the user interface alone, for instance to change its command verbs. Typically this is only possible with very rich graphic user interfaces.

Tangible User Interfaces which place a greater emphasis on touch and physical environment or its element.

Text User Interfaces are user interfaces which output text, but accept other forms of input in addition to or in place of typed command strings.

Voice User Interfaces which accept input and provide output by generating voice prompts. The user input is made by pressing keys or buttons, or responding verbally to the interface.

Natural-Language Interfaces used for search engines and on webpages. User types in a question and waits for a response.

Zero-Input Interfaces get inputs from a set of sensors instead of querying the user with input dialogs.

Zooming User Interfaces are graphical user interfaces in which information objects are represented at different levels of scale and detail, and where the user can change the scale of the viewed area in order to show more detail

Self-Assessment Exercise

Check the GUI features and functions of LabView. Then compare their environments with MS-DOS.

3.4 History of User Interfaces

The history of user interfaces can be divided into the following phases according to the dominant type of user interface:

- Batch Interface, 1945-1968
- Command-line User Interface, 1969 to present
- Graphical User Interface, 1981 to present

3.5 User Interface Modalities and Modes

A **modality** is a path of communication employed by the user interface to carry input and output. Examples of modalities:

Input — allowing the users to manipulate a system. For example the computer keyboard allows the user to enter typed text; digitising tablet allows the user to create free-form drawing.

Output — allowing the system to indicate the effects of the users' manipulation. For example the computer monitor allows the system to display text and graphics (*vision modality*), loudspeaker allows the system to produce sound (*auditory modality*).

The user interface may employ several redundant input and output modalities, allowing the user to choose which ones to use for interaction.

A **mode** is a distinct method of operation within a computer program, in which the same input can produce different perceived results depending on the state of the computer program. Heavy use of modes often reduces the usability of a user interface, as the user must expend effort to remember current mode states, and switch between mode states as necessary.

4.0 Conclusion

In this unit, you have been introduced to the fundamental concepts of user interface. You have also learnt the history and significance of user interface design.

5.0 Summary

In this unit, you have learnt the:

- introduction of user interface which is the aggregate of means by which users interact with a particular machine, device, computer program or any other complex tool

- study of the various types of user interface design which includes graphical user interfaces, web-based user interfaces, command line interfaces e.t.c.

- history of user interfaces which can be divided into batch interface, command-line user interface and graphical user interface

- modality of a user interface which is a path of communication employed by the user interface to carry input and output.

6.0 Self Assessment Exercise

1. Explain the Microsoft's Clippy the paperclip.
2. Write a short note on the Command-line User Interface.

7.0 References/Further Reading

Pinel, J. P. (2008). *Biopsychology* (7th ed.). Boston: Pearson. p. 357. Wikipedia.org

Unit 2 Designing Good User Interfaces

1.0 Introduction

This unit describes the essentials of designing good interface designs and also discusses the various users.

2.0 Objectives

At the end of this unit, you should be able to:

- explain the essentials of a good interface design
- identify the necessary tips needed for designing a good interface
- discuss various users.

3.0 Main Content

3.1 Essentials of Interface Design

There are three pillars to an application's success:
- Features
- Function
- Face

Features refer to what the application will do for the user. Features are the requirements for the software.

Function refers to how well the software operates. Bug-free software will function perfectly.

Face refers to how the application presents itself to the user; the application's "user interface."
Features, function and face can be restated as questions:
- Does the software meet the user's requirements? (Features)
- Does the software operate as intended? (Function)
- Is the software easy to use? (Face)

The goal of user interface design is to put a happy face on the application. Stated in more concrete terms, a successful user interface will require *Zero Training* and will be *friendly not foreboding*.

Zero Training
The goal of Zero Training could be considered as a sub-goal of friendly not foreboding. However, training costs are a major impediment to the usage of software making Zero Training an important goal by itself.

There are two types of training involved in software design: software training and job training. Software training assumes the user knows how to do the job at hand and only needs to learn how to use the software to do the job. Job training teaches the user how to do the job - which can include more than how to use the application to do the job. The goal of Zero Training relates to *zero software training*. Job training can be integrated with software training, but results in a much more ambitious project.

Friendly not Foreboding

Almost everything you do to implement the goal of Zero Training will further the goal of being friendly not foreboding. However, some techniques for reducing training may slow up experienced users. For example, you could pop-up new user messages whenever the user lands in a particular field. Seeing the same message after awhile makes the experienced user dispense with the messages.

Being friendly is an attitude and encompasses more than what is necessary for the Zero Training goal. Applications do have an attitude. For example, consider the following sets of application messages:

"Database does not exist" "I could not find database "CorpInfo". If you are sure this name is correct, CorpInfo could be unavailable due to maintenance or LAN problems. You should contact the help desk to see when CorpInfo will again be available."

"SQL access error 123" "I could not save information to the database. You can try to save again to see if the error clears. If you call the help desk concerning this problem, tell them you have a "SQL access error 123".

"out of hunk"[1] "I have run out of memory (RAM). This typically happens when there is a bug in the program which causes it to lose memory over time. Save your game if possible. To free the memory you will need to reset the computer (turn it off and then on)."

The attitude of the first message is "you did something that caused me a problem" while the attitude of the second message is "I have a problem. Could you give me a hand?"

3.1.1 Designing a Good User Interface

Designing a good user interface is an iterative process. First, you design and implement a user interface using appropriate techniques. Then you evaluate the design. The results of the evaluation feed the next design and implementation. You stop the process when you have met your design goals or you run out of time and/or money.

Note that if you have different user communities (or the same user with different jobs), you may need different user interfaces, customisable user interfaces or both. For example, Microsoft Word provides four user interfaces: normal, outline, page layout and master. In addition, Microsoft Word provides a host of customisation features for the keyboard, menu and toolbars.

While design is important, the real key to creating a good user interface is in your evaluation techniques. Obviously, you should use your own user interface. If you can't use it, how can anyone else? Next, get feedback from your testers.

The best evaluations are done by watching over the shoulder of the user. The key here is watching. If you are telling the user what to do, you will never find out if your interface is easy to use. Let the user figure it out himself or herself. If the user has to ask you what to do or pauses to figure out what to do next, you may need to work on your interface. If the user grimaces, find out why. Learn from the experience. Some of the most innovative designs were shot down when the users could not figure them out.

You will need both new and experienced users for testing your interface. The new users will help you determine if you meet the Zero Training goal. The experienced users will let you know if your methods for meeting the Zero Training goal interfere with getting work done once the user has learned the software.

3.1.2 Tips for Designing a Good User Interface

- **Consistency**: The most important thing that you can possibly do is make sure that your user interface works consistently. If you can double-click on items in one list and have something happen, then you should be able to double-click on items in any other list and have the same sort of thing happen. Put your buttons in consistent places on all of your windows, use the same wording in labels and messages, and use a consistent colour scheme throughout. Consistency in your user interface allows your users to build an accurate mental model of the way that it works, and accurate mental models lead to lower training and support costs.
- **Set standards and stick to them**: The only way that you'll be able to ensure consistency within your application is to set design standards and then stick to them. The best approach is to adopt an industry standard and then fill any missing guidelines that are specific to your needs. Industry standards, such as the ones set by IBM (1993) and Microsoft (1995), will often define 95-99% of what you need. By adopting industry standards, you are not only taking advantage of the work of others, you are also increasing the chances that your application will look and feel like other applications that your users purchase or have built. User interface design standards should be set during the define infrastructure stage.
- **Explain the rules**: Your users need to know how to work with the application that you built for them. When an application works consistently, it means you only have to explain the rules once. This is a lot easier than explaining in detail exactly how to use each and every feature in an application step by step.
- **Support both novices and experts**: Although a library-catalog metaphor might be appropriate for casual users of a library system, library patrons, it probably is not all that effective for expert users, librarians. Librarians are highly trained people who are able to use complex search systems to find information in a library; therefore you should consider building a set of search screens to support their unique needs.
- **Navigation between screens is important**: If it is difficult to get from one screen to another, then your users will quickly become frustrated and give up. When the flow between screens matches the flow of the work that the user is trying to accomplish, then your application will make sense to your users. Because different users work in different ways, your system will need to be flexible enough to support their various approaches. Interface-flow diagrams can be used during the model stage to model the flow between screens.

Navigation within a screen is important: In Western societies, people read left to right and top to bottom. Because people are used to this, so you should design screens that are also organised left to right and top to bottom. All you want is to organise navigation between widgets on your screen in a manner that users will find familiar to them.

Word your messages and labels appropriately:
The text that you display on your screens is a primary source of information for your users. If your text is worded poorly then your interface will be perceived poorly by your users. Using full words and sentences, as opposed to abbreviations and codes makes your text easier to understand. Your messages should be worded positively, imply that the user is in control, and provide insight into how to use the application properly. For example, which message do you find more appealing "You have input the wrong information" or "An account number should be 8 digits in length?" Furthermore, your messages should be worded consistently and displayed in a consistent place on the screen. Although the messages "The person's first name must be input." and "An account number should be input." are separately worded well, together they are inconsistent. In light of the first message, a better wording of the second message would be "The account number must be input" to make the two messages consistent.

- **Understand your widgets**: You should use the right widget (widgets are interface elements that the users interact with) for the right task, helping to increase the consistency in your application and probably making it easier to build the application in the first place. The only way that you can learn how to use widgets properly is to read and understand the user-interface standards and guidelines that your organisation has adopted.

- **Look at other applications with a grain of salt**: Unless you know that another application follows the user-interface standards and guidelines of your organisation, you must not assume that the application is doing things right. Although it is always a good idea to look at the work of others to get ideas, until you know how to distinguish between good and bad user-interface design you have to be careful. Too many developers make the mistake of imitating the user interface of another application that was poorly designed.

- **Use colour appropriately**: Colour should be used sparingly in your applications, and if you do use it you must also use a secondary indicator. The problem is that some of your users may be colour blind – if you are using colour to highlight something on a screen, then you need to do something else to make it stand out if you want people to notice it, such as display a symbol beside it. You also want to use colours in your application consistently so that you have a common look and feel throughout your application. Also, colour generally does not port well between platforms – what looks good on one system often looks poor on another system. We have all been to presentations where the presenter said "it looks good on my machine at home."

- **Follow the contrast rule**: If you are going to use colour in your application, you need to ensure that your screens are still readable. The best way to do this is to follow the contrast rule: Use dark text on light backgrounds and light text on dark backgrounds. It is very easy to read blue text on a white background but very difficult to read blue text on a red background. The problem is that there is no enough contrast between blue and red to make it easy to read, whereas there is a lot of contrast between blue and white.

- **Use fonts appropriately**: Old English fonts might look good on the covers of William Shakespeare's plays, but they are really difficult to read on a screen. Use fonts that are easy to read, such as serif fonts, Times Roman. Furthermore, use your fonts consistently and sparingly. A screen using two or three fonts effectively looks a lot better than a screen that uses five or six. Never forget that you are using a different font every time you change the size, style (bold, italics, underlining,), typeface, or colour.

- **Grey things out, do not remove them**: You often find that at certain times it is not applicable to give your users access to all the functionality of an application. You need to select an object before you can delete it, so to reinforce your mental model the application should do something with the Delete button and/or menu item. Should the button be removed or greyed out? Grey it out, never remove it. By greying things out when they shouldn't be, users can start building an accurate mental model as to how your application works. If you simply remove a widget or menu item instead of greying it out then it is much more difficult for your users to build an accurate mental model, because they only know what is currently available and not what is not available to them. The old adage that out of sight is out of mind is directly applicable here.

- **Use non destructive default buttons**: It is quite common to define a default button on every screen, the button that gets invoked if the user presses the return/enter key. The problem is that sometimes people will accidentally hit the enter/return key when they do not mean to, consequently invoking the default button. Your default button shouldn't be something that is potentially destructive, such as delete or save (perhaps your user really did not want to save the object at that moment).

- **Alignment of fields**: When a screen has more than one editing field, you need to organise the fields in a way that is both visually appealing and efficient. As shown in figure 1.1, the best way to do so is to left-justify edit fields, or in other words make the left-hand side of each edit field line up in a straight line, one over the other. The corresponding labels should be right justified and placed immediately beside the field. This is a clean and efficient way to organise the fields on a screen.

- **Justify data appropriately**: For columns of data it is common practice to right justify integers, decimal align floating point numbers, and left justify strings.

- **Do not create busy screens**: Crowded screens are difficult to understand and hence are difficult to use. Experimental results show that the overall density of the screen should not exceed 40%, whereas local density within groupings shouldn't exceed 62%.

- **Group things on the screen effectively**: Items that are logically connected should be grouped together on the screen to communicate that they are connected, whereas items that have nothing to do with each other should be separated. You can use whitespace between collections of items to group them and/or you can put boxes around them to accomplish the same thing.

- **Open windows in the centre of the action**: When your user double-clicks on an object to display its edit/detail screen then his or her attention is on that spot. Therefore it makes sense to open the window in that spot, not somewhere else.

Pop-up menus should not be the only source of functionality: Your users cannot learn how to use your application if you hide major functionality from them. One of the most frustrating practices of developers is to misuse pop-up, also called context-sensitive, menus. Typically there is a way to use the mouse on your computer to display a hidden pop-up menu that provides access to functionality that is specific to the area of the screen that you are currently working on.

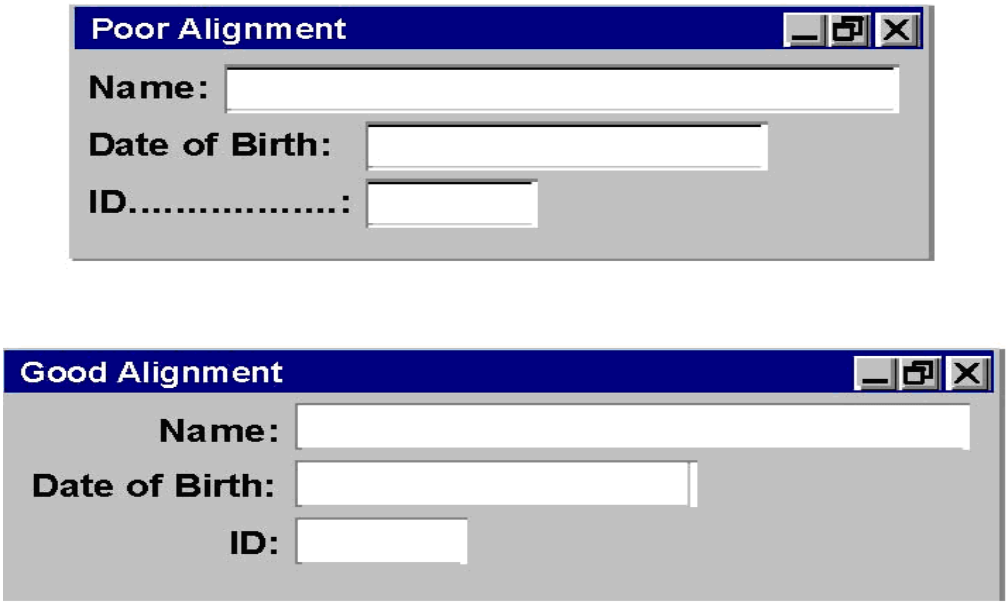

Fig. 1.1: Showing that alignment of fields is critical

3.2 Understanding Users

You must understand the user to be able to put a happy face on your application. You should understand the user's job, how the software fits in with that job and how the user goes about getting the job done. You need to approach the design of software from the user's viewpoint not from an abstract requirements document. Specifically, you should understand what the user will be doing with the application. If you can think like a user, you can create a much better user interface.

Here are some basic principles to remember about users:

- Your software is like a hammer - The user doesn't really care how well crafted it is, the user just wants nails put in the wall. Users just want to do their job (or play their game). They don't care about you or your software. Your software is just an expedient tool to take the user where the user wants to go.

- Given a selection of hammers to buy at the hardware store - The user will select the one which will be most fun to use. Of course, this varies by user - some will want the plastic handle, some the wood, some the green, etc. When evaluating your software, users are often swayed by looks, not function. Thus, steps taken to make the product look good (nice icons, pictures, good colour scheme, fields aligned, etc.) will often favourably enhance evaluations of your software.

- It had better drive nails - The user will not really know if your software is adequate to the job until the user has used the software to do the actual work. From an interface perspective, the software should not look like it can do more than it can.

- Some users will try to use a hammer to drive a screw - If your software is good, some user somewhere will try to use the software for some purpose for which you never intended it to be used. Obviously, you cannot design a user interface to deal with uses you cannot foresee. There is no single rigid model of the right way to use the software, so build in flexibility.

- Users will not read an instruction manual for a hammer- They won't read one for your software either, unless they really have to. Users find reading instruction manuals almost as unpleasurable as dental work.

- A user reading the instruction manual for a hammer is in serious trouble - When you create your help system (and printed manual), remember that the user will only resort to those materials if he or she is in trouble. The user will want a problem solved as fast and as easily as possible.

- Hammers don't complain - You should try to eliminate error messages and any error messages your program needs should have the right attitude.

4.0 Conclusion

In this unit, you have been introduced to the essentials of good interface design. You have also learnt the necessary tips needed for designing a good interface and the need for understanding various users.

5.0 Summary

In this unit, we have learnt:

- the essentials of interface design with emphasis on the features, functions and the face of the software

- that designing a good user interface which has been described as an iterative process involves designing, implementing, evaluating and redesigning until all removable errors have been taken care of

- the tips necessary for designing a good user interface which includes consistency, setting standards and sticking to them, supporting of both novices and experts, e.t.c.

- understanding the user's job, how the software fits in with that job and how the user goes about getting the job done.

6.0 Self Assessment Exercise

1. How do you ensure that the interface support both novices and expert?
2. Write short note on the design of a user interface for a user with hearing disability.

7.0 References/Further Reading

Wkipedia.org

oer.nou.edu.ng://oer.nou.edu.ng.linfo.org/gui.html

Unit 3 Graphical User Interface

1.0 Introduction

This unit describes the general concept of Graphical User Interface (GUI) and also the history and elements of graphical user interface. The concept of three dimensional (3D) graphical user interface is also introduced.

2.0 Objectives

At the end of this unit, you should be able to:

- describe a graphical user interface
- explain the history behind graphical user interface
- list the elements of a graphical user interface
- describe the three-dimensional user interfaces.

3.0 Main Content

3.1 Introduction to Graphical User Interface

Graphical User Interfaces, also known as GUIs, offer a consistent visual language to represent information stored in computers. This makes it easier for people with little computer skills to work with and use computer software. This explains the most common elements of the visual language interfaces.

A **graphical user interface**
Is a type of user interface which allows people to interact with electronic devices such as computers; hand-held devices such as MP3 Players, Portable Media Players or Gaming devices; household appliances and office equipment with images rather than text commands. A *GUI* offers graphical icons, and visual indicators, as opposed to text-based interfaces, typed command labels or text navigation to fully represent the information and actions available to a user. The actions are usually performed through direct manipulation of the graphical elements.

The term *GUI* is historically restricted to the scope of two-dimensional display screens with display resolutions capable of describing generic information, in the tradition of the computer science research at Palo Alto Research Centre (PARC). The term *GUI* earlier might have been applicable to other high-resolution types of interfaces that are non-generic, such as videogames, or not restricted to flat screens, like volumetric displays.

3.2 History of Graphical User Interface

Precursor to GUI
The precursor to GUIs was invented by researchers at the Stanford Research Institute, led by Douglas Engelbart. They developed the use of text-based hyperlinks manipulated with a mouse for the On-Line System. The concept of hyperlinks was further refined and extended to graphics by researchers at Xerox PARC, who went beyond text-based hyperlinks and

used a GUI as the primary interface for the Xerox Alto computer. Most modern general-purpose GUIs are derived from this system. As a result, some people call this class of interface PARC User Interface (PUI) (note that PUI is also an acronym for Perceptual User Interface).

Ivan Sutherland developed a pointer-based system called the Sketchpad in 1963. It used a light-pen to guide the creation and manipulation of objects in engineering drawings.

PARC User Interface
The PARC User Interface consisted of graphical elements such as windows, menus, radio buttons, check boxes and icons. The PARC User Interface employs a pointing device in addition to a keyboard. These aspects can be emphasised by using the alternative acronym WIMP, which stands for **W**indows, **I**cons, **M**enus and **P**ointing device.

Evolution
Following PARC, the first GUI-centric computer operating model was the Xerox 8010 Star Information System in 1981 followed by the Apple Lisa (which presented concept of menu bar as well as window controls) in 1982 and the Atari ST and Commodore Amiga in 1985.

The GUIs that are familiar to most people today are Microsoft Windows, Finder Interface (Mac OS X), and the X Window System interfaces. Apple, IBM and Microsoft used many of Xerox's ideas to develop products, and IBMs Common User Access specifications formed the basis of the user interface found in Microsoft Windows, IBM OS/2 Presentation Manager, and the Unix Motif toolkit and window manager. These ideas evolved to create the interface found in current versions of Microsoft Windows, as well as in Mac OS X and various desktop environments for Unix-like operating systems, such as Linux. Thus most current GUIs have largely common idioms.

Post-WIMP Interfaces
Smaller mobile devices such as PDAs and smart phones typically use the WIMP elements with different unifying metaphors, due to constraints in space and available input devices. Applications for which WIMP is not well suited may use newer interaction techniques, collectively named as post-WIMP user interfaces.

Some touch-screen-based operating systems such as Apple's iPhone OS currently use post-WIMP styles of interaction. The iPhone's use of more than one finger in contact with the screen allows actions such as pinching and rotating, which are not supported by a single pointer and mouse.

A class of GUIs sometimes referred to as post-WIMP include 3D compositing window manager such as Compiz, Desktop Window Manager, and LG3D. Some post-WIMP interfaces may be better suited for applications which model immersive 3D environments, such as Google Earth.

3.3 Elements of Graphical User Interfaces

A GUI uses a combination of technologies and devices to provide a platform which the user can interact with in order to achieve the tasks of gathering and producing information. Series of elements conforming to visual languages have evolved to represent information stored in computers. This makes it easier for people with little computer skills to work with

and use computer software. The most common combination of such elements in GUIs is the WIMP paradigm, especially in personal computers.
User interfaces use visual conventions to represent the generic information shown. Some conventions are used to build the structure of the static elements on which the user can interact, and define the appearance of the interface.
The key elements of GUI are divided into two categories viz Structural and Interactive elements.

3.3.1 Structural Elements

User interfaces use visual conventions to represent the generic information shown. Some conventions are used to build the structure of the static elements on which the user can interact, and define the appearance of the interface.

Window
A window is an area on the screen that displays information, with its contents being displayed independently from the rest of the screen. An example of a window is what appears on the screen when the "My Documents" icon is clicked in the Windows Operating System. It is easy for a user to manipulate a window: it can be opened and closed by clicking on an icon or application, and it can be moved to any area by dragging it (that is, by clicking in a certain area of the window – usually the title bar along the top – and keeping the pointing device's button pressed, then moving the pointing device). A window can be placed in front or behind another window, its size can be adjusted, and scrollbars can be used to navigate the sections within it. Multiple windows can also be opened at one time, in which case each window can display a different application or file – this is very useful when working in a multitasking environment. The system's memory is the only limitation to the number of windows that can be opened at once. There are also many types of specialised windows.

- A **Container Window:** a window that is opened while invoking the icon of a mass storage device, directory or folder and which is presenting an ordered list of other icons that could be again some other directories, or data files or may be even executable programs. All modern container windows could present their content on screen either by acting as browser windows or text windows. Their behaviour can automatically change according to the choices of the single users and their preferred approach to the graphical user interface.

- A **browser window:** allows the user to move forward and backward through a sequence of documents or web pages. Web browsers are examples of these types of windows.

- **Text terminal windows**: are designed for embedding interaction with text user interfaces within the overall graphical interface. MS-DOS and UNIX consoles are examples of these types of windows.

- A **child window:** opens automatically or as a result of a user activity in a parent window (A parent window can be any type of window). Pop-up windows on the Internet can be child windows.

- A **message window**, or **dialog box**: is a type of child window. These are usually small and basic windows that are opened by a program to display information to the user and/or get information from the user. They usually have a button that must be pushed before the program can be resumed.

Menus

Menus allow the user to execute commands by selecting from a list of choices. Options are selected with a mouse or other pointing device within a GUI. A keyboard may also be used. Menus are convenient because they show what commands are available within the software. This limits the amount of documentation the user reads to understand the software.

- A **menu bar** is displayed horizontally across the top of the screen and/or along the top of some or all windows. A pull-down menu is commonly associated with this menu type. When a user clicks on a menu option, the pull-down menu will appear.
- A **menu** has a visible title within the menu bar. Its contents are only revealed when the user selects it with a pointer. The user is then able to select the items within the pull-down menu. When the user clicks elsewhere, the contents of the menu will disappear.
- A **context menu** is invisible until the user performs a specific mouse action, like pressing the right mouse button. When the software-specific mouse action occurs the menu will appear under the cursor.
- **Menu extras** are individual items within or at the side of a menu.

Icons

An icon is a small picture that represents objects such as a file, program, web page, or command. Icons are used as a quick way to execute commands, open documents, and run programs. They are also very useful when searching for an object in a browser list, because in many operating systems all documents using the same extension will have the same icon.

Controls (or Widgets)

The interface element that a computer user interacts with is known as a **control** or **widget**.

Window

A paper-like rectangle that represents a "window" into a document, form, or design area.

Pointer (or mouse cursor)

The spot where the mouse "cursor" is currently referencing.

Text box

A box in which texts or numbers are entered.

Button

An equivalent to a push-button as found on mechanical or electronic instruments.

Hyperlink

Text with some kind of indicators (usually underlining and/or colour) that shows that clicking it will take one to another screen or page.

Drop-down list

A list of items from which to select: The list normally only displays items when a special button or indicator is clicked.

Check box
A box which indicates an "on" or "off" state via a check-mark or an "×".

Radio button
A button, similar to a check-box, except that only one item in a group can be selected. Its name comes from the mechanical push-button group on a car radio receiver. Selecting a new item from the group's buttons also deselects the previously selected button.

Data grid
A spreadsheet-like grid that allows numbers or text to be entered in rows and columns.

Tabs
A tab is typically a rectangular small box which usually contains a text label or graphical icon associated with a view pane. When activated the view pane, or window, displays widgets associated with that tab; groups of tabs allow the user to switch quickly between different widgets. This is used in the web browsers Firefox, Internet Explorer, Konqueror, Opera, and Safari. With these browsers, you can have multiple web pages open at once in one window, and quickly navigate between them by clicking on the tabs associated with the pages. Tabs are usually placed in groups at the top of a window, but may also be grouped on the side or bottom of a window.

3.3.2 Interaction Elements

Some common idioms for interaction have evolved in the visual language used in GUIs. Interaction elements are interface objects that represent the state of an ongoing operation or transformation, either as visual remainders of the user intent (such as the pointer), or as affordances showing places where the user may interact.

Cursor
A cursor is an indicator used to show the position on a computer monitor or other display devices that will respond to inputs from a text input or pointing devices.

Pointer
One of the most common components of a GUI on the personal computer is a pointer: a graphical image on a screen that indicates the location of a pointing device, and can be used to select and move objects or commands on the screen. A pointer commonly appears as an angled arrow, but it can vary within different programs or operating systems. Example of this can be found within text-processing applications, which use an I-beam pointer that is shaped like a capital I, or in web browsers, which often indicate that the pointer is over a hyperlink by turning the pointer in the shape of a gloved hand with outstretched index finger.

The use of a pointer is employed when the input method, or pointing device, is a device that can move fluidly across a screen and select or highlight objects on the screen. Pointer trails can be used to enhance its visibility during movement. In GUIs where the input method relies on hard keys, such as the five-way key on many mobile phones, there is no pointer employed, and instead the GUI relies on a clear focus state.

Selection
A selection is a list of items on which user operations will take place. The user typically adds items to the list manually, although the computer may create a selection automatically.

Adjustment handle
A handle is an indicator of a starting point for a drag and drop operation. Usually the pointer shape changes when placed on the handle, showing an icon that represents the supported drag operation.

Self-Assessment Exercise 1

Identify and study these elements within Window operating system

3.4 Three-dimensional User Interfaces

For typical computer displays, *three-dimensional* are a misnomer—their displays are two-dimensional. Three-dimensional images are projected on them in two dimensions. Since this technique has been in use for many years, the recent use of the term three-dimensional must be considered a declaration by equipment marketers that the speed of three dimensions to two dimensions projection is adequate to use in standard GUIs.

Motivation
Three-dimensional GUIs are quite common in science fiction literature and movies, such as in *Jurassic Park*, which features Silicon Graphics' three-dimensional file manager, "File system navigator", an actual file manager that never got much widespread use as the user interface for a Unix computer. In fiction, three-dimensional user interfaces are often immersible environments like William Gibson's Cyberspace or Neal Stephenson's Metaverse. Three-dimensional graphics are currently mostly used in computer games, art and computer-aided design (CAD). There have been several attempts at making three-dimensional desktop environments like Sun's Project Looking Glass or SphereXP from Sphere Inc. A three-dimensional computing environment could possibly be used for collaborative work. For example, scientists could study three-dimensional models of molecules in a virtual reality environment, or engineers could work on assembling a three-dimensional model of an airplane. This is a goal of the Croquet project and Project Looking Glass.

Technologies
The use of three-dimensional graphics has become increasingly common in mainstream operating systems, from creating attractive interfaces—eye candy— to functional purposes only possible using three dimensions. For example, user switching is represented by rotating a cube whose faces are each user's workspace, and window management is represented in the form or via a Rolodex-style flipping mechanism in Windows Vista (see Windows Flip 3D). In both cases, the operating system transforms windows on-the-fly while continuing to update the content of those windows.

Interfaces for the X Window System have also implemented advanced three-dimensional user interfaces through compositing window managers such as Beryl, Compiz and KWin using the AIGLX or XGL architectures, allowing for the usage of OpenGL to animate the user's interactions with the desktop.

Another branch in the three-dimensional desktop environment is the three-dimensional GUIs that take the desktop metaphor a step further, like the BumpTop, where a user can manipulate documents and windows as if they were "real world" documents, with realistic movement and physics.

The Zooming User Interface (ZUI) is a related technology that promises to deliver the representation benefits of 3D environments without their usability drawbacks of orientation problems and hidden objects. It is a logical advancement on the GUI, blending some three-dimensional movement with two-dimensional or "2.5D" vector objects.

4.0 Conclusion

In this unit, you have been introduced to graphical user interface (GUI). The history of graphical user interface was also discussed. The elements of graphical user interface were also explained. You were also introduced to three-dimensional user interfaces.

5.0 Summary

In this unit, you have learnt the:

- introduction to graphical user interface which is a type of user interface that allows people to interact with electronic devices such as computers, hand-held devices, household appliances and office equipment with images rather than text commands
- history of graphical user interface, precursor to GUI, PARC user interface and the evolution of other graphical user interfaces
- Elements of graphical user interface which are divided into two categories that includes structural and interactive elements.

Self-Assessment Exercise 2

1. What do you understand by graphical user interface?
2. Explain the structural and iterative elements of graphical user interface.

6.0 Self-Assessment Exercise

Explain the PARC graphical user interface

7.0 References/Further Reading

http://oer.nou.edu.ng.linfo.org/gui.html. Retrieved on 12 November 2008.
oer.nou.edu.ng.wikipedia.com

Unit 4 Human- Computer Interaction

1.0 Introduction

In this unit, you will be introduced to Human-Computer Interaction (HCI) and its differences with other related fields. The goals and future development of human- computer interaction and the general concept of human-computer interface will be introduced.

2.0 Objectives

At the end of this unit, you should be able to:

- explain the term human- computer interaction
- identify the various goals of human -computer interaction
- differentiate human -computer interaction from other related fields
- describe the future development of HCI and explain the human -computer interface.

3.0 Main Content

1 Introduction to Human -Computer Interaction
Human–Computer Interaction (**HCI**) is the study of interaction between people (users) and computers. It involves the study intersection of computer science, behavioural sciences, design and several other fields of study. Interaction between users and computers occurs at the user interface (or simply *interface*), which includes both software and hardware. The association for computing machinery defines ***human-computer interaction*** as "a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them”

Since human-computer interaction studies a human and a machine together, it draws its supporting knowledge from the machine and human sides. On the machine side, techniques in computer graphics, operating systems, programming languages, and development environments are relevant. On the human side, communication theory, graphic and industrial design disciplines, statistics, linguistics, social sciences, cognitive psychology, and human performance are relevant. Engineering and design methods are also relevant. Due to the multidisciplinary nature of HCI, people with different backgrounds contribute to its success. HCI is sometimes referred to as **Man–Machine Interaction (MMI)** or **Computer–Human Interaction (CHI)**.

3.1.1 Human–Computer Interface

The human–computer interface can be described as the point of communication between the human user and the computer. The flow of information between the human and computer is defined as the loop of interaction. The loop of interaction has several aspects to it including:

Task Environment: The conditions and goals set upon the user.
Machine Environment: The environment that the computer is connected to, i.e. a laptop in a college student's dorm room.

Areas of the Interface: Non-overlapping areas involve processes of the human and computer not pertaining to their interaction. Meanwhile, the overlapping areas only concern themselves with the processes pertaining to their interaction.

Input Flow: Begins in the task environment as the user has some task that requires using their computer.

Output: The flow of information that originates in the machine environment.

Feedback: Loops through the interface that evaluate, moderate, and confirm processes as they pass from the human through the interface to the computer and back.

3.2 Goals of HCI

The basic goal of HCI is to improve the interactions between users and computers by making computers more usable and responsive to the user's needs. Specifically, HCI is concerned with:

- methodologies and processes for designing interfaces (i.e., given a task and a class of users, design the best possible interface within given constraints, optimising for a desired property such as learning ability or efficiency of use)
- methods for implementing interfaces (e.g. software toolkits and libraries; efficient algorithms)
- techniques for evaluating and comparing interfaces
- developing new interfaces and interaction techniques
- developing descriptive and predictive models and theories of interaction.

A long term goal of HCI is to design systems that minimise the barrier between the human's cognitive model of what they want to accomplish and the computer's understanding of the user's task.
Researchers in HCI are interested in developing new design methodologies, experimenting with new hardware devices, prototyping new software systems, exploring new paradigms for interaction, and developing models and theories of interaction.

3.3 Differences with Related Fields

HCI focuses on user interface design mainly for users of computer system and promotes effective interaction between computers and users (human). User Interface Design is concerned with the users of devices such as computers, appliances, machines, mobile communication devices, software applications, and websites. In HCI, efficient user interface is critical.

HCI differs from human factors in that the focus is more on users working specifically with computers, rather than other kinds of machines or designed artifacts. There is also a focus in HCI on how to implement the computer software and hardware mechanisms to support human-computer interaction. Thus, human factors are a broader term; HCI could be described as the human factors of computers, although some experts try to differentiate these areas.

According to some experts, HCI also differs from ergonomics (which will be fully introduced in the next unit) in that there is less focus on repetitive work-oriented tasks and procedures, and much less emphasis on physical stress and the physical form or industrial design of the user interface, such as keyboards and mice. However, this does not take full account of ergonomics, which recently has gained a much broader focus (equivalent to human factors). Cognitive ergonomics, for example, is a part of ergonomics, of which *software ergonomics* (an older term, essentially the same as HCI) is a part.

Three areas of study have substantial overlap with HCI even as the focus of inquiry shifts. In the study of Personal Information Management (PIM), human interactions with the computer are placed in a larger informational context - people may work with many forms of information, some computer-based, many not (e.g., whiteboards, notebooks, sticky notes, refrigerator magnets) in order to understand and effect desired changes in their world. In Computer Supported Cooperative Work (CSCW), emphasis is placed on the use of computing systems in support of the collaborative work of a group of people. The principles of Human Interaction Management (HIM) extend the scope of CSCW to an organisational level which can be implemented without the use of computer systems.

3.4 Future Development of HCI

As the means by which humans interact with computers continues to evolve rapidly, human–computer interaction is affected by the forces shaping the nature of future computing. These forces include:

- decreasing hardware costs leading to larger memories and faster systems
- miniaturisation of hardware leading to portability
- reduction in power requirements leading to portability
- new display technologies leading to the packaging of computational devices in new forms
- specialised hardware leading to new functions
- increased development of network communication and distributed computing
- increasingly widespread use of computers, especially by people who are outside the computing profession
- increasing innovation in input techniques (i.e., voice, gesture, pen), combined with lowering cost, leading to rapid computerisation by people previously left out of the "computer revolution"
- wider social concerns leading to improved access to computers by currently disadvantaged groups.

The future for HCI is expected to include the following features:

Ubiquitous communication: Computers will communicate through high speed local networks, nationally over wide-area networks, and portably via infrared, ultrasonic, cellular, and other technologies. Data and computational services will be portably accessible from many if not most locations to which a user travels.

High functionality systems: Systems will have large numbers of functions associated with them. There will be so many systems that most users, technical or non-technical, will not have time to learn them in the traditional way (e.g., through thick manuals).

Mass availability of computer graphics: Computer graphics capabilities such as image processing, graphics transformations, rendering, and interactive animation will become widespread as inexpensive chips become available for inclusion in general workstations.
Mixed media: Systems that will handle images, voice, sounds, video, text, formatted data. These will be exchangeable over communication links among users. The separate worlds of

consumer electronics (e.g., stereo sets, VCRs, televisions) and computers will partially merge. Computer and print worlds will continue to cross assimilate each other.

High-bandwidth interaction: The rate at which humans and machines interact will increase substantially due to the changes in speed, computer graphics, new media, and new input/output devices. This will lead to some qualitatively different interfaces, such as virtual reality or computational video.

Large and thin displays: New display technologies will finally mature enabling very large displays and also displays that are thin, light weight, and have low power consumption. This will have large effects on portability and will enable the development of paper-like, pen-based computer interaction systems very different in feel from desktop workstations of the present.

Embedded computation: Computation will pass beyond desktop computers into every object for which uses can be found. The environment will be alive with little computations from computerised cooking appliances. to lighting and plumbing fixtures to window blinds to automobile braking systems to greeting cards. To some extent, this development is already taking place. The difference in the future is the addition of networked communications that will allow many of these embedded computations to coordinate with each other and with the user. Human interfaces to these embedded devices will in many cases be very different from those appropriate to workstations.

Augmented reality: A common staple of science fiction, augmented reality refers to the notion of layering relevant information into our vision of the world. Existing projects show real-time statistics to users performing difficult tasks, such as manufacturing. Future work might include augmenting our social interactions by providing additional information about those we converse with.

Group interfaces: Interfaces to allow groups of people to coordinate will be common (e.g., for meetings, for engineering projects, for authoring joint documents). These will have major impacts on the nature of organisations and on the division of labour. Models of the group design process will be embedded in systems and will cause increased rationalisation of the design.

User tailor ability: Ordinary users will routinely tailor applications to their own use and will use this power to invent new applications based on their understanding of their own domains. Users, with their deeper knowledge of their own knowledge domains, will increasingly be important sources of new applications at the expense of generic systems programmers (with systems expertise but low domain expertise).

Information utilities: Public information utilities (such as home banking and shopping) and specialised industry services (e.g., weather for pilots) will continue to proliferate. The rate of proliferation will accelerate with the introduction of high-bandwidth interaction and the improvement in quality of interfaces.

4.0 Conclusion

In this unit, you have been introduced to the concepts of human -computer interaction. You have also been introduced to the various goals of HCI and also the difference between HCI and other related fields. The future of HCI was also discussed. The concept of the human-computer interface was also introduced.

5.0 Summary

In this unit, you have learnt the following:

- introduction to human–computer interaction (HCI) which is the study of interaction between people (users) and computers. HCI is also sometimes referred to as man–machine interaction (MMI) or computer–human interaction (CHI)
- the human–computer interface which can be described as the point of communication between the human user and the computer
- the goal of HCI which is basically to improve the interactions between users and computers by making computers more usable and receptive to the user's needs
- the difference between HCI and other related fields like graphical user interface, ergonomics e.t.c.
- the future development in HCI like Ubiquitous communication, High functionality systems, Mass availability of computer graphics e.t.c.

Self- Assessment Exercise

1. What do you understand by Human -Computer Interaction?
2. Describe the Human -Computer Interface

6.0 Self-Assessment Exercise

Discuss briefly the future development in HCI.

7.0 References/Further Reading

More discussion of the differences between these terms can be found in the ACM SIGCHI Curricula for Human-Computer Interaction

Brown, C. M. (1998). *Human-Computer Interface Design Guidelines*. Intellect Books. pp.2–3. Green, P. (2008). Iterative Design. Lecture presented in Industrial and Operations

Engineering 436 Human Factors in Computer Systems, University of Michigan, Ann Arbor, MI, February 4, 2008.

Wickens, C. D., John, D. L, Yili, L., & Sallie, E. G. (2004). *An Introduction to Human Factors Engineering*. (2nd ed.). Upper Saddle River, NJ: Pearson Prentice Hall. pp.185–193.

Unit 5 Ergonomics

1.0 Introduction

This unit describes in great detail the various aspects of ergonomics alongside the history behind it. The efficiency and benefits of ergonomics will also be discussed. Different fields of ergonomics will also be highlighted.

2.0 Objectives

At the end of this unit, you should be able to:

- define the term ergonomics
- identify the various aspects of ergonomics
- explain the history of ergonomics
- describe efficiency and ergonomics
- identify the various benefits of ergonomics
- list the various fields of ergonomics.

3.0 Main Content

3.1 Introduction to Ergonomics

Ergonomics is derived from two Greek words: ergon, meaning work, and nomoi, meaning natural laws, to create a word that means the science of work and a person's relationship to that work.

The International Ergonomics Association has adopted this technical definition: ergonomics (or human factors) is the scientific discipline concerned with the understanding of interactions among humans and other elements of a system, and the profession that applies theory, principles, data and methods to design in order to optimise human well-being and overall system performance.

Ergonomics is the science of making things comfy. It also makes things efficient. However for simplicity, ergonomics makes things comfortable and efficient.

At its simplest definition, ergonomics literally means the science of work. So ergonomists, i.e. the practitioners of ergonomics, study work, how work is done and how to work better. It is the attempt to make work better that ergonomics becomes so useful.

However, what you, or the user, is most concerned with is, "How can I use the product or service, will it meet my needs, and will I like using it?" Ergonomics helps define how it is used, how it meets you needs, and most importantly if you like it. It makes things comfy and efficient.

Ergonomics is concerned with the 'fit' between people and their work. It takes account of the worker's capabilities and limitations in seeking to ensure that tasks, equipment, information and the environment suit each worker.

To assess the fit between a person and his/her work, ergonomists consider the job being done and the demands on the worker; the equipment used (its size, shape, and how appropriate it is for the task), and the information used (how it is presented, accessed, and changed). Ergonomics draws on many disciplines in its study of humans and their environments, including anthropometry, biomechanics, mechanical engineering, industrial engineering, industrial design, kinesiology, physiology and psychology.

3.2 Five Aspects of Ergonomics

There are five aspects of ergonomics: safety, comfort, ease of use, productivity/performance, and aesthetics. From these aspects of ergonomics, examples are given of how products or systems could benefit from redesign based on ergonomic principles.

- Safety – This has to do with the ability to use a device or work with a device without short or long term damage to parts of the body. For example in Medicine bottles: The print on them could be larger so that a sick person who may have bad vision (due to sinuses, etc.) can easily see the dosages and label. Ergonomics could design the print style, colour and size for optimal viewing.
-
- Comfort – Comfort in the human-machine interface is usually noticed first. Physical comfort in how an item feels is pleasing to the user. If you do not like to touch it you won't. If you do not touch it you will not operate it. If you do not operate it, then it is useless. For example, in Alarm clock display: Some displays are harshly bright, drawing one's eye to the light when surroundings are dark. Ergonomic principles could re-design this based on contrast principles.
-
- Ease of use – This has to do with the ability to use a device with no stress. For example in Street Signs: In a strange area, many times it is difficult to spot street signs. This could be addressed with the principles of visual detection in ergonomics.
-
- Productivity/performance – For example in High-definition Television (HD TV): The sound on HD TV is much lower than regular TV. So when you switch from HD to regular, the volume increases dramatically. Ergonomics recognises that this difference in decibel level creates a difference in loudness and hurts human ears and this could be solved by evening out the decibel levels.
- Aesthetics - the look and feel of the object, the user experience.

3.3 History of Ergonomics

The foundations of the science of ergonomics appear to have been laid within the context of the culture of Ancient Greece. A good deal of evidence indicates that Hellenic civilisation in the 5th century BC used ergonomic principles in the design of their tools, jobs, and workplaces.

The term ergonomics is derived from the Greek words *ergon* [work] and *nomos* [natural laws] and first entered the modern lexicon when Wojciech Jastrzębowski used the word in his 1857 article *Rys ergonomji czyli nauki o pracy, opartej na prawdach poczerpniętych z Nauki Przyrody* (The Outline of Ergonomics, i.e. Science of Work, Based on the Truths Taken from the Natural Science).

Later, in the 19th century, Frederick Winslow Taylor pioneered the "Scientific Management" method, which proposed a way to find the optimum method for carrying out a given task. Taylor found that he could, for example, triple the amount of coal that workers were shoveling by incrementally reducing the size and weight of coal shovels until the fastest shoveling rate was reached. Frank and Lillian Gilbreth expanded Taylor's methods in the early 1900s to develop "Time and Motion Studies". They aimed at improving efficiency by eliminating unnecessary steps and actions. By applying this approach, the Gilbreths reduced the number of motions in bricklaying from 18 to 4.5, allowing bricklayers to increase their productivity from 120 to 350 bricks per hour.

World War II marked the development of new and complex machines and weaponry, and these made new demands on operators' cognition. The decision-making, attention, situational awareness and hand-eye coordination of the machine's operator became keys to the success or failure of a task. It was observed that fully functional aircraft, flown by the best-trained pilots, still crashed. In 1943, Alphonse Chapanis, a lieutenant in the U.S. Army, showed that this so-called "pilot error" could be greatly reduced when more logical and differentiable controls replaced confusing designs in airplane cockpits. The experience of the World War II promoted the study. Man-Machine interaction became a major focus when weapons meant to attack the enemies were malfunctioning – attacking friends instead of foes.

In the decades since the war, ergonomics has continued to flourish and diversify. The Space Age created new human factors issues such as weightlessness and extreme g-forces. How far could environments in space be tolerated, and what effects would they have on the mind and body? The dawn of the Information Age has resulted in the new ergonomics field of human-computer interaction (HCI). Likewise, the growing demand for and competition among consumer goods and electronics has resulted in more companies including human factors in product design.

At home, work, or play new problems and questions must be resolved constantly. People come in all different shapes and sizes, and with different capabilities and limitations in strength, speed, judgment, and skills. All of these factors need to be considered in the design function. To solve design problems, physiology and psychology must be included with an engineering approach.

3.4 Ergonomics in Workplace

Fig. 5.1: **Description of Workplace Environment**
(Source: oer.nou.edu.ng.wikipedia.org)

Fundamentals for the Flexible Workplace variability and compatibility with desk components, that flex from individual work activities to team settings. Workstations provide supportive ergonomics for task-intensive environments.

Outside of the discipline itself, the term 'ergonomics' is generally used to refer to physical ergonomics as it relates to the workplace (example ergonomic chairs and keyboards). Ergonomics in the workplace has to do largely with the safety of employees, both in the long and short-term. Ergonomics can help reduce costs by improving safety. This would decrease the money paid out in workers' compensation.

Workplaces may either take the reactive or proactive approach when applying ergonomics practices. Reactive ergonomics is when something needs to be fixed, and corrective action is taken. Proactive ergonomics is the process of seeking areas that could be improved and fixing the issues before they become a large problem. Problems may be fixed through equipment design, task design, or environmental design. Equipment design changes the actual, physical devices used by people. Task design changes what people do with the equipment. Environmental design changes the environment in which people work, but not the physical equipment they use.

3.5 Efficiency and Ergonomics

Efficiency is quite simply making something easier to do. Several forms of efficiency are:

- Reducing the strength required makes a process more physically efficient.
- Reducing the number of steps in a task makes it quicker (i.e. efficient) to complete.
- Reducing the number of parts makes repairs more efficient.
- Reducing the amount of training needed, i.e. making it more intuitive, gives you a larger number of people who are qualified to perform the task. Imagine how in-efficient trash disposal would be if your teenage child wasn't capable of taking out the garbage. Have you tried an ergonomic trash bag?

Efficiency can be found almost everywhere. If something is easier to do you are more likely to do it. If you do it more, then it is more useful. Again, utility is the only true measure of the quality of a design. And if you willingly do something more often you have a greater chance of liking it. If you like doing it you will be more comfortable doing it. So the next time you hear the term ergonomics you will know what it means to you. And I hope that is a comforting thought.

Ergonomics can help you in many ways. Among other things, it can benefit your life, health, productivity and accuracy. One of the best benefits of ergonomics is saving time. We never seem to have enough of it as it is, so why not try to get a little more out of your day? Ergonomics is about making things more efficient. By increasing the efficiency of a tool or a task, you tend to shorten the length of time it takes to accomplish your goal.

3.6 Benefits of Ergonomics

The three main benefits of ergonomics are:

Slim Down the Task

Have you ever wondered why some things are so convoluted, cumbersome and chaotic? And they take forever to complete. And most of what you do does not aid the outcome. For example, think back to the last time you got hired for a job, bought a house or car, or did something else that required a ton of paperwork. How many different forms did you write the same information on? That was not very ergonomic.

You can almost always make a task a little leaner. But first you have to understand the task. Ergonomics requires that tasks are properly slimed down and steps involved in a task are well written out. This is why task analysis is normally done in ergonomics (Task analysis is fully discussed in module 2, unit 2).

Once you have all the steps written out, you need to take a good look at them and identify areas that you can "ergonomise":

- Repetition – Look for steps that are repeated and see if they are all necessary.
- Order – See if you can re-order the steps to optimise your effort.
- Synergy – Can you combine things or somehow get more bang for your buck?
- Value Added – Look at every step and make sure it adds value to the outcome. If it doesn't, cut it.
- Necessity – Make sure the quantity of the step is needed. Do you really need to brush your teeth with 57 strokes, or will 32 do?

Simplify the Task

You can also save time by simplifying the task. This is not just about reducing the number of steps, but making those steps easier to perform. The less training and/or skill that is required for a task, the quicker the pace at which it tends to get finished.

This is a great ergonomic tip, especially when the task requires more than one person. If you are trying to get your kids to pick up their toys before they go to bed, you can save a lot of time by making it easier. That is what the toy chest is for. Instead of having different places for different things, they can just throw everything in one place.

Increase Body Mechanism

Ergonomic can increase your body mechanics. A good ergonomic tool acts as extension of your body enhancing capabilities. Some tools make you more effective and faster at completing a task. (Try cutting a log without an axe and see how long it takes you.)

3.7 Fields of Ergonomics

Engineering Psychology

Engineering psychology is an interdisciplinary part of Ergonomics and studies the relationships of people to machines, with the intent of improving such relationships. This may involve redesigning equipment, changing the way people use machines, or changing the location in which the work takes place. Often, the work of an engineering psychologist is described as making the relationship more "user-friendly."

Engineering Psychology is an applied field of psychology concerned with psychological factors in the design and use of equipment. Human factors are broader than engineering psychology, which is focused specifically on designing systems that accommodate the information-processing capabilities of the brain.

Macroergonomics
Macroergonomics is an approach to ergonomics that emphasises a broad system view of design, examining organisational environments, culture, history, and work goals. It deals with the physical design of tools and the environment. It is the study of the society/technology interface and their consequences for relationships, processes, and institutions. It also deals with the optimisation of the designs of organisational and work systems through the consideration of personnel, technological, and environmental variables and their interactions. The goal of macroergonomics is a completely efficient work system at both the macro- and micro-ergonomic level which results in improved productivity, and employee satisfaction, health, safety, and commitment. It analyses the whole system, finds how each element should be placed in the system, and considers all aspects for a fully efficient system. A misplaced element in the system can lead to total failure.

Seating Ergonomics
The best way to reduce pressure in the back is to be in a standing position. However, there are times when you need to sit. When sitting, the main part of the body weight is transferred to the seat. Some weight is also transferred to the floor, backrest, and armrests. Where the weight is transferred is the key to a good seat design. When the proper areas are not supported, sitting in a seat all day can put unwanted pressure on the back causing pain.

The lumbar (bottom five vertebrate in the spine) needs to be supported to decrease disc pressure. Providing both a seat back that inclines backwards and has a lumbar support is critical to prevent excessive low back pressures. The combination which minimises pressure on the lower back is having a backrest inclination of 120 degrees and a lumbar support of 5 cm. The 120 degrees inclination means the angle between the seat and the backrest should be 120 degrees. The lumbar support of 5 cm means the chair backrest supports the lumbar by sticking out 5 cm in the lower back area.

Another key to reducing lumbar disc pressure is the use of armrests. They help by putting the force of your body not entirely on the seat and backrest, but putting some of this pressure on the armrests. Armrest needs to be adjustable in height to assure that shoulders are not overstressed.

4.0 Conclusion

In this unit, you have been introduced to the fundamental concepts of Ergonomics. You have also learnt the different aspect of ergonomics and also the history of ergonomics. Ergonomics in workplace was also discussed alongside achieving efficiency in ergonomics. The various benefits of ergonomics were also discussed. The various fields of ergonomics were also briefly explained.

5.0 Summary

In this unit you, have learnt the following:

- introduction to ergonomics which is derived from two Greek words: ergon, meaning work, and nomoi, meaning natural laws, to create a word that means the science of work and a person's relationship to that work
- highlighting of the various aspect of ergonomics like safety, comfort, ease of use e.t.c.

- the history of ergonomics whose foundations appears to have been laid within the context of the culture of Ancient Greece
- the discussion of ergonomics in workplace and achieving efficiency in ergonomics
- explanation of the various benefits of ergonomics which was discussed in greater detail
- the discussion of various fields of ergonomics like engineering psychology, Macroergonomics, Seating ergonomics.

Self- Assessment Exercise

1. What do you understand by ergonomics?
2. Highlight the various benefits of ergonomics, giving examples.

6.0 Self-Assessment Exercise

Discuss briefly any two fields of ergonomics.

7.0 References/Further Reading

Berkeley Lab. *Integrated Safety Management: Ergonomics*. Website. Retrieved 9 July 2008.

Berkeley Lab. *Today at Berkeley Lab: Ergonomic Tips for Computer Users*. Retrieved 8 January 2009.

Brookhuis, K., Hedge, A., Hendrick, H., Salas, E., & Stanton, N. (2005). *Handbook of Human Factors and Ergonomics Models*. Florida: CRC Press.

Wickens & Hollands (2006). *Engineering Psychology and Human Performance*. oer.nou.edu.ng.wikipedia.com