

USER INTERFACE DESIGN: DOES IT REALLY MATTER?

Orhan Dikener, Enderhan Karakoç, Adem Demirsoy

Faculty of Communication, Selcuk University, Konya, Turkey



Abstract

The intention of this paper is to provide an overview on the subject of Human - Computer Interaction as a new research field for graphic design and visual communication. We tried to answer some questions like; Why does interface design matter to computer users? How does interface design affect computer users and the work they need to accomplish? Which interface design components are important to know about when it comes to judging different information resources? How do field practitioners and research findings describe what makes for good interface design versus bad design? On the most basic level, design matters because it plays a large role in determining whether we can get our work done. A well-designed tool is one that is easy to interpret and satisfying to use. In fact, many software developers say that the best designs are ones that users never give a second thought about. They describe this quality as invisibility and it is the hallmark of effortless user interaction and good design. In contrast, a poorly designed tool is far from invisible, taking far too much time to use and delivering few results for our work in return. Whether an interface design is a good one or a poor one is a complex and involved issue. But one thing is certain for users, issues of design quality begin with a resource's interface.

Key words: Information Design, User Interface (GUI), Human Computer Interaction (HCI)

1. INTRODUCTION

Human-computer interaction (HCI) is the study of how people interact with computing technology. The intellectual foundations of HCI derive from a variety of fields: computer science, cognitive psychology, social psychology, perceptual psychology, linguistics, artificial intelligence, and anthropology (Myers et al., 1996).

Succinctly defined, HCI is the science of designing usable, discoverable, and satisfying software interfaces (Dillon, 1983). According to cognitive psychology, the human mind is a specific type of an information processing unit. From the traditional cognitive point of view, the HCI system is composed of two information processing units, the human being and the computer, so that the output of one unit enters the other's input, and vice versa. In other words, human-computer interaction can be described as an information processing loop (Kaptelinin, 1992).

As it mentioned before, Human-computer interaction (HCI) is the study of how humans interact with computer systems. Many disciplines contribute to HCI, including computer science, psychology, ergonomics, engineering, and graphic design. HCI is a broad term that covers all aspects of the way in which people interact with computers.

In their daily lives, people are coming into contact with an increasing number of computer-based technologies. Some of these computer systems, such as personal computers, we use directly. We come into contact with other systems less directly — for example, we have all seen cashiers use laser scanners and digital cash registers when we shop. And, as we are all too aware, some systems are easier to use than others. When users interact with a computer system, they do so via a user interface (UI).

The basic user interface is usually understood to include things like menus, windows, the keyboard, the mouse, the “beeps” and other sounds the computer makes, and in general, all the information channels that allow the user and the computer to communicate.

Of course, using a modern computer system also involves reading manuals, calling help lines, attending training classes, asking questions of colleagues, and trying to puzzle out how software operates. A successful computer system or software package supports those activities, so that support is part of the user interface too.

But in a sense, all of these parts are the “peanut butter” we mentioned previously. No matter how well they are crafted, the interface will be a failure if the underlying system doesn’t do what the user needs, in a way that the user finds appropriate. In other words, the system has to match the users’ tasks (Lewis and Rieman, 1994).

Utilizing computers had always begged the question of interfacing. The methods by which human has been interacting with computers has travelled a long way. The journey still continues and new designs of technologies and systems appear more and more every day and the research in this area has been growing very fast in the last few decades.

The growth in Human-Computer Interaction (HCI) field has not only been in quality of interaction, it has also experienced different branching in its history. Instead of designing regular interfaces, the different research branches have had different focus on the concepts of multimodality rather than unimodality, intelligent adaptive interfaces rather than command/action based ones, and finally active rather than passive interfaces (Karray, et. al., 2008).

2. WHY THE USER INTERFACE MATTERS?

Whether we are aware of it or not, computers pervade our life. Computer applications are used either by us, or for us, in some way almost every day. The user interface (or just interface) is that part of the computer system with which a user interacts in order to undertake his or her tasks and achieve his or her goals. The user interface and the ways of interacting with computer-based systems are different for each system. For example, digital watches generally have buttons that users press to set the time or use the stopwatch facility. Microwave ovens might have dials to turn or a digital display and a touchpad of buttons to set the cooking time. PCs have a screen, a keyboard, and a mouse (or sometimes a trackball or a joystick) that enable interaction to take place. So each user interface is different. Depending on the design of the interface, each of these systems will either be usable — that is, easy to learn and easy to use — or problematic for users. Earlier we described a computer system as the combination of hardware and software components that receive input from, and communicate output to, a user to support his or her performance of a task. Although the user interface is simply the part of the computer system that enables interaction and serves as a bridge between users and the system, to users the interface often is the system (Constantine and Lockwood, 1999). The user’s view of a computer system is often limited to and based solely on his or her experience of the user interface

3. THE IMPORTANCE OF GOOD USER INTERFACE DESIGN

A good user interface design encourages an easy, natural, and engaging interaction between a user and a system, and it allows users to carry out their required tasks. With a good user interface, the user can forget that he or she is using a computer and get on with what he or she wants to do. Just as knowledge of the transmission mechanism of a car is of little concern to most motorists, knowledge of the internal workings of a computer system should be of little consequence to its users. Although we have used the

adjectives “good,” “poor,” and “bad” to describe user interfaces, it is worth noting that each of these terms is subjective: they have different meanings for different people and their use to rate various aspects of a user interface will vary. You may have used the terms “good” or “bad” to describe, for example, the colors used in an interface, the pictures on the icons, or how attractive or eye-catching the interface was. These attributes describe the overall look or aesthetics of the UI.

4. NECESSITY OF USER-CENTERED DESIGN

User-centered design (UCD) is an approach to user interface design and development that involves users throughout the design and development process. User centered design not only focuses on understanding the users of a computer system under development but also requires an understanding of the tasks that users will perform with the system and of the environment (organizational, social, and physical) in which they will use the system. Taking a user-centered design approach should optimize a computer system's usability.

The three areas of user interface design usability principles are:

1. Place users in control of the interface
2. Reduce users' memory load
3. Make the user interface consistent.

Attributes of usability could be:

Learnability; How long does it take a new user to become productive with the system?

Speed of Operation; How well does the system response match the user's work practice?

Robustness; How tolerant is the system of user error?

Recoverability; How good is the system at recovering from user error?

Adaptability; How closely is the system tied to a single model of work?

Definition of usability could be; *Human- Centered Design Processes for Interactive Systems*; provides guidance on and lists the main principles and essential activities for human (user)-centered design, for achieving usability in systems. Briefly, the four main principles of human-centered design are (ISO, 1997) :

1. The active involvement of users
2. An appropriate allocation of function between user and system
3. The iteration of design solutions
4. Multidisciplinary design teams

Adopting the approach prescribed by ISO 13407 ensures that the users' perspectives form part of the HCI design and development process, which will positively influence the usability of the final product.

There are three pillars to an application's success:

- Features
- Function
- Face

Features refers 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)

User-centered design approach should provide the specifications listed below for usability (Ambler, 2000):

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 color 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 not only take advantage of the work of others you also increase the chance 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 should you design screens that are also organized left to right

and top to bottom. You want to organize 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 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 organization 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 organization, 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 user-interface design 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 color appropriately; Color 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 color blind – if you are using color to highlight something on a screen then you need to do something else to make it stand out if you want these people to notice it, such as display a symbol beside it. You also want to use colors in your application consistently so that you have a common look and feel throughout your application. Also, color generally does not port well between platform – 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 color 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 not 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 hard to read on a screen. Use fonts that are easy to read, such as serif fonts like 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 color.

Gray 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 grayed out? Gray it out, never remove it. By graying things out when they shouldn’t be used people can start building an accurate mental model as to how your application works. If you simply remove a widget or menu item instead of graying 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 to them, and not what is not available. 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 want to organize the fields in a way that is both visually appealing and efficient. As shown in Figure 1 I have always found that 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 organize 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 center 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 in.

CONCLUSION

In conclusion, designers and developers of tools and visualizations need to have a strong understanding of human-computer interaction and cognitive psychology. So long as it is easier to learn to write code than to learn to communicate with your life partner it will be easier to adopt and maintain a machine centered focus rather than a human-centered focus. Both knowledge of the user's cognitive performance limitations and knowledge of how to design representations which will contribute to the user's task relevant goals are essential to the successful creation of new tools or visualizations which will be both useful and usable in dealing with visualizations of abstract entities and relationships. An understanding of basic HCI and human cognition will help designers and developers to remain focused on the problems that should be solved rather than on the problems which can be solved.

Additionally, user interfaces, including information visualization, for successful products and services enable users around the world to access complex data and functions. Solutions to global user-interface design consist of partially universal and partially local solutions to the design of metaphors, mental models, navigation, appearance, and interaction. By managing the user's experience of familiar structures and processes, the user-interface designer can achieve compelling forms that enable the user

Journal of International Scientific Publications: Media and Mass Communication, Volume 2

ISSN 1314-8028, Published at: <http://www.scientific-publications.net>

interface to be more usable and acceptable to a wider range of users. The user will be more productive and satisfied with the product in many different locations globally.

Graphic artists and book designers are skilled in the art of presenting information using the right medium and then designing the presentation of that information appropriately. These skills should be represented on the user interface design team. Human Computer Interaction in the field of input and output techniques has developed a lot of new techniques over the last few years. With the recently released full multi-touch tablets and notebooks the way how people interact with the computer is coming to a new dimension. As humans are used to handle things with their hands the technology of multi-touch displays or touchpad's brought much more convenience for use in daily life. But for sure the usage of human speech recognition will also play an important part in the future of human computer interaction.

REFERENCES

Constantine, L. L., & Lockwood, L. A. D. (1999) Software for use: A practical guide to the models and methods of usage-centered design, New York, NY: ACM Press.

Dillon, R.F. (1983) Human factors in user-computer interaction: An introduction. Behavior Research Methods & Instrumentation, 15, 195-199.

Fakhreddine Karray, Milad Alemzadeh, Jamil Abou Saleh and Mo Nours Arab (2008) Human-Computer Interaction: Overview on State of the Art, International Journal on Smart Sensing and Intelligent Systems, vol. 1, no. 1, March 2008.

Kaptelinin, V. (1992). Human computer interaction in context: The activity theory perspective. In J. Gornostaev, ed., Proceedings of EWHCI'92 Conference. Moscow: ICSTI.

Shneiderman, Ben. (1992) Designing the User Interface: Strategies for Effective Human-Computer Interaction. Massachusetts: Addison-Wesley.

Scott W. Ambler (2000) User Interface Design: Tips and Techniques, Cambridge University Press.