Date:02/07/2020

HUMAN COMPUTER INTERACTION

TOPIC: -

Assignment # 1

SUBMITTED BY: -

M. SAQIB REHAN

ROLL NO.: -

503

BSCS 8TH SEMESTER (2016-2020)

DEPARTMENT OF COMPUTER SCIENCE

ISLAMIA UNIVERSITY BWP

(SUB CAMPUS RYK)

Q # 1: What are interaction paradigms? Discuss different types of paradigm.

We believe that we now build interactive systems that are more usable than those built-in the past. We also believe that there is considerable room for improvement in designing more usable systems in the future. The great advances in computer technology have increased the power of machines and enhanced the bandwidth of communication between human and computer. The impact of the technology alone, however, is not sufficient to enhance its usability. As our machines have become more powerful, they key to increased usability has come from the creative and considered application of the technology to accommodate and augment the power of the human.

For many years the prevailing paradigm in interaction design was to develop application for the desktop - intended to be used by single user sitting Infront of a CPU, monitor, keyboard and mouse. A dominant part of this approach waste design software applications that would run using a GUI or WIMP interface

Different types of paradigm

- Time Sharing
- Video Display Units
- Programming Toolkits
- Personal computing
- Window systems and the WIMP interface
- The metaphor
- Direct Manipulation
- The World Wide Web

Time Sharing

In the 1940s and 1950s, the significant advances in computing consisted of new hardware technologies. Mechanical relays were replaced by vacuum electron tubes. Tubes were replaced by transistors, and transistors by integrated chips, all of which meant that the amount of sheer computing power was increasing by orders of magnitude.

By the 1960s it was becoming apparent that the explosion of growth in computing power would be wasted if there were not an equivalent explosion of ideas about how to channel that power.

Video display units

As early as the mid-1950s researchers were experimenting with the possibility of presenting and manipulating information from a computer in the form of images on a video display unit (VDU). These display screens could provide a more suitable medium than a paper printout for presenting vast quantities of strategic information for rapid assimilation. It was not until 1962, however, when a young graduate student at the Massachusetts Institute of Technology (MIT), Ivan Sutherland, astonished the established computer science community with the Sketchpad program, that the capabilities of visual images were realized

Programming toolkits

Douglas Engelbart's ambition since the early 1950s was to use computer technology as a means of complementing human problem-solving activity. Engelbart's idea as a graduate student at the University of California at Berkeley was to use the computer to teach humans. This dream of nave human users actually learning from a computer was a stark contrast to the prevailing attitude of his contemporaries that computers were purposely complex technology that only the intellectually privileged were capable of manipulating.

Personal computing

Programming toolkits provide a means for those with substantial computing skills to increase their productivity greatly. But Engelbart's vision was not exclusive to the computer literate. The decade of the 1970s saw the emergence of computing power aimed at the masses, computer literate or not. One of the first demonstrations that the powerful tools of the hacker could be made accessible to the computer novice was a graphics programming language for children called LOGO. The inventor, Semen Paper, wanted to develop a language that was easy for children to use.

As technology progresses, it is now becoming more difficult to distinguish between what constitutes personal computer, or workstation, and what constitutes a mainframe.

Window systems and the WIMP interface

With the advent and immense commercial success of personal computing, the emphasis for increasing the usability of computing technology focused on addressing the single user who engaged in a dialog with the computer in order to complete some work. Humans are able to think about more than one thing at a time, and in accomplishing some piece of work, they frequently interrupt their current train of thought to pursue some other related piece of work. A personal computer system which forces the user to progress in order through all of the tasks needed to achieve some objective, from beginning to end without any diversions, does not correspond to that standard working pattern.

The metaphor

Metaphor is used quite successfully to teach new concepts in terms of ones, which are already understood. It is no surprise that this general teaching mechanism has been successful in introducing computer novices to relatively foreign interaction techniques. Metaphor is used to describe the functionality of many interaction widgets, such as windows, menus, buttons and palettes. Tremendous commercial successes in computing have arisen directly from a judicious choice of metaphor. The Xerox Alto and Star were the first workstations based on the metaphor of the office desktop. The majority of the management tasks on a standard workstation have to do with the file manipulation. Linking the set of tasks associated with file manipulation to the filing tasks in a typical office environment makes the actual computerized tasks easier to understand at first. The success of the desktop metaphor is unquestionable. Another good example in the personal computing domain is the widespread use of the spreadsheet for accounting and financial modeling

Direct Manipulation

In the early 1980s as the price of fast and high-quality graphics hardware was steadily decreasing, designers were beginning to see that their products were gaining popularity as their visual content increased. As long as the user-system command line prompt computing was going to stay within the minority population of the hackers who reveled in the challenge of complexity. In a standard command line interface, the only way to get any feedback on the results of previous interaction is to know that you only have to ask for it and to know how to ask for it. Rapid visual and audio feedback on a high-resolution display screen or through a high-quality sound system makes it possible to provide evaluative information for every executed user action.

He highlights the following features of a direct manipulation interface.

- visibility of the objects of interest incremental action at the interface with rapid feedback on all actions
- reversibility of all actions, so that users are encouraged to explore without
- severe penalties syntactic correctness of all actions, so that every user action is a legal
- operation replacement of complex command language with actions to manipulate
- directly the visible objects.

The World Wide Web

Probably the most significant recent development interactive computing is the Worldwide Web, often referred to as just the web, or WWW. The web is built on top of the Internet, and offers an easy to use, predominantly graphical interface to information, hiding the underlying complexities of transmission protocols, addresses and remote access to data

References:

https://www.zeepedia.com/read.php?interaction_paradigms_the_wimp_interfaces_interaction_paradigms_human_computer_interaction&b=11&c=15

Q # 2 : What do you know about interaction design? Briefly explain all its phases.

A simple definition is:

> "achieving goals within constraints"

Goals

- What is the purpose of the design we are intending to produce?
- Who is it for?
- Why do they want it?

Constraints

- What materials must we use?
- What standards must we adopt?
- How much can it cost?
- How much time do we have to develop it?
- Are there health and safety issues?
- In the case of the personal movie player: does it have to withstand rain?
- Must we use existing video standards to download movies?
- Do we need to build in copyright protection?

Trade-off

Choosing which goals or constraints can be relaxed so that others can be met.

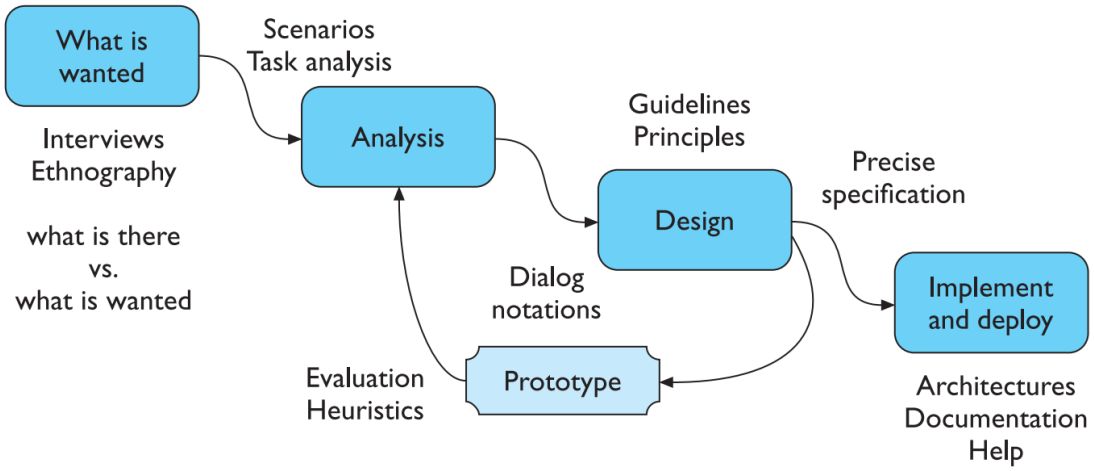
The golden rule of design

For Human–Computer Interaction the obvious materials are the human and the computer. That is, we must:

- understand computers
  - limitations, capacities, tools, platforms
- understand people
  - psychological, social aspects, human error.

THE PROCESS OF DESIGN

A system has been designed and built, and only when it proves unusable do, they think to ask how to do it right! In other companies' usability is seen as equivalent to testing – checking whether people can use it and fixing problems, rather than making sure they can from the beginning. In the best companies, however, usability is designed in from the start.



Requirements – what is wanted

The first stage is establishing what exactly is needed. As a precursor to this it is usually necessary to find out what is currently happening. For example, how do people currently watch movies? What sort of personal appliances do they currently use?

There are a number of techniques used for this in HCI:

interviewing people, videotaping them, looking at the documents and objects that they work with, observing them directly.

Analysis

The results of observation and interview need to be ordered in some way to bring out key issues and communicate with later stages of design. Rich stories of interaction, which can be used in conjunction with a method like task analysis or on their own to record and make vivid actual interaction. These techniques can be used both to represent the situation as it is and also the desired situation.

Design

Well, this is all about design, but there is a central stage when you move from what you want, to how to do it. There are numerous rules, guidelines and design principles that can be used to help with

this. We need to record our design choices in some way and there are various notations and methods to do this, including those used to record the existing situation.

Iteration and prototyping

Humans are complex and we cannot expect to get designs right first time. We therefore need to evaluate a design to see how well it is working and where there can be improvements. Some forms of evaluation can be done using the design on paper, but it is hard to get real feedback without trying it out.

Implementation and deployment

Finally, when we are happy with our design, we need to create it and deploy it. This will involve writing code, perhaps making hardware, writing documentation and