CHPATER 1

INTRODUCTION

Human Computer Interaction focuses on the interactions between human and computer systems, including the user interface and the underlying processes which produce the interactions. The contributing disciplines include computer science, cognitive science, human factors, software engineering, management science, psychology, sociology, and anthropology. Early research and development in human-computer interaction focused on issues directly related to the user interface. Some typical issues were the properties of various input and output devices, interface learn ability for new users versus efficiency and extensibility for experienced users, and the appropriate combination of interaction components such as command languages, menus, and graphical user interfaces (GUI). Recently, the field of human-computer interaction has changed and become more devoted to the processes and context for the user interface. Functionality of a system is defined by the set of actions or services that it provides to its users. However, the value of functionality is visible only when it becomes possible to be efficiently utilized by the user [67]. Usability of a system with a certain functionality is the range and degree by which the system can be used efficiently and adequately to accomplish certain goals for certain users. The actual effectiveness of a system is achieved when there is a proper balance between the functionality and usability of a system [52].

1.1 Human-Computer Interaction: Definition, Terminology

The rapid growth of computing has made effective human-computer interaction essential. HCI (human-computer interaction) is the study of how people interact with computers and to what extent computers are or are not developed for successful interaction with human beings. 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 [66].

Gustav Evertsson describes Human Computer Interaction is about designing computer systems so the user can carry out their activities productively and safely. It is not how easy something is to use, it is about how usable it is. Or, a broader definition of HCI is;

“Human Computer Interaction is a discipline concerned with the design, evaluation and implementation of interactive computer systems for human use and with the study of major phenomena surrounding them” [36][13].

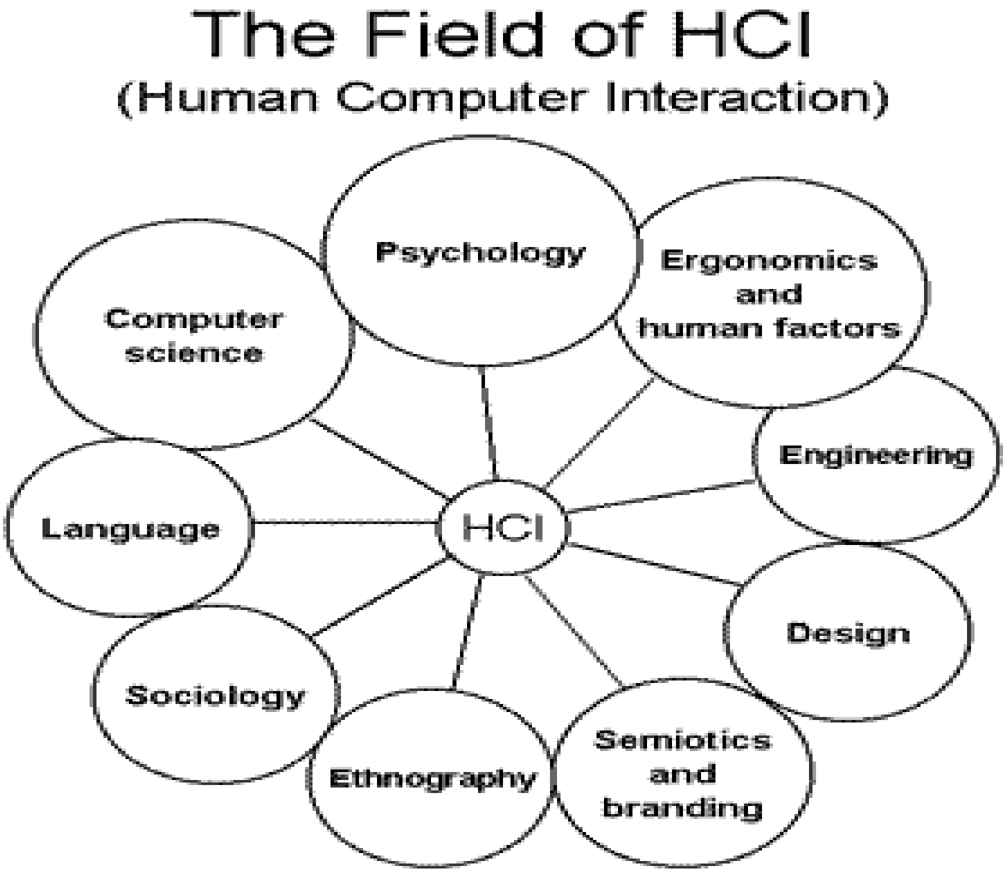
" It is a wide variety of different kind of people and not just technical specialists as in the past, so it is important to design HCI that supports the needs, knowledge and skills of the intended users" [36][13].

As its name implies, HCI consists of three parts: the user, the computer itself, and the ways they work together.

User: By "user", we may mean an individual user or a group of users working together. An appreciation of the way people's sensory systems (sight, hearing, touch) relay information is vital. Also, different users form different conceptions or mental models about their interactions and have different ways of learning and keeping knowledge. In addition, cultural and national differences play an important part.
Computer: When we talk about the computer, we're referring to any technology ranging from desktop computers, to large scale computer systems. For example, if we were discussing the design of a Website, then the Website itself would be referred to as "the computer". Devices such as mobile phones or VCRs can also be considered to be "computers".
Interaction: There are obvious differences between humans and machines. In spite of these, HCI attempts to ensure that they both get on with each other and interact successfully. In order to achieve a usable system, you need to apply what you know about humans and computers, and consult with likely users throughout the design process. In real systems, the schedule and the budget are important, and it is vital to find a balance between what would be ideal for the users and what is feasible in reality [13]. Human-Computer Interaction studies how people design, implement and use computer interfaces [31]. HCI has become an umbrella

term for a number of disciplines including theories of education, psychology, collaboration as well as efficiency and ergonomics [32].

Figure 1.1: some of the disciplines involved in the field of Human-Computer Interaction [16]



Recent developments in the area of HCI have shown an interest in adaptive interfaces, speech recognition, gestures and the role of time [33] [34] [35] [1].

Having these concepts in mind and considering that the terms computer, machine and system are often used interchangeably. In this context, HCI is a design that should produce a fit between the user, the machine and the required services in order to achieve a certain performance both in quality and optimality of the services [61]. Determining what makes a certain HCI design good is mostly subjective and context dependant. For example, an aircraft part designing tool should provide high precisions in view and design of the parts while a graphics editing software may not need such a precision. The available technology could also affect how different types of HCI are designed for the same purpose. One example is using commands, menus, graphical user interfaces (GUI), or virtual reality to access functionalities of any given computer [66].

1.2 The Goals of HCI

The goals of HCI are to produce usable and safe systems, as well as functional systems. Usability is concerned with making systems easy to learn and easy to use [36]. In order to produce computer systems with good usability developers must attempt to:

- Understand the factors that determine how people use technology
- Develop tools and techniques to enable building suitable systems
- Achieve efficient, effective and safe interaction
- Put user first

Underlying the whole theme of HCI is the belief that people using a computer system should come first. Their needs, capabilities and preferences for conducting various tasks should direct developers in the way that they design systems. People need not change themselves in order to fit in within the system. Instead, the system should be designed to match their requirements.

1.3 HCI Technologies

Gustav Evertsson describes HCI design is about designing the computer system for the people and not the people for the computers. There are a lot of important factors that have to be considered by designers. Example of factors is:

- Physiology such as the human behavior and mental processes.
- Organizational such as the influence of one individual in a group with the other member as attitude and behavior.
- Ergonomics such as how people interact with different artifacts[36]

HCI design should consider many aspects of human behaviors and needs to be useful. The complexity of the degree of the involvement of a human in interaction with a machine is sometimes invisible compared to the simplicity of the interaction method itself. The existing interfaces differ in the degree of complexity both because of degree of functionality/usability and the financial and economical aspect of the machine in the market. For instance, an electrical kettle need not to be sophisticated in interface since its only functionality is to heat the water and it would not be cost-effective to have an interface more than a thermostatic on and off switch. On the other hand, a simple website that may be limited in functionality should be complex enough in usability to attract and keep customers [60][66]. Therefore, in design of HCI, the degree of activity that involves a user with a machine should be thoroughly thought. The user activity has three different levels: physical, cognitive, and affective. The physical aspect determines the mechanics of interaction between human and computer while the cognitive aspect deals with ways that users can understand the system and interact with it. The affective aspect is a more recent issue and it tries not only to make the interaction a pleasurable experience for the user but also to affect the user in a way that make user continue to use the machine by changing attitudes and emotions toward the user [60][66].

The existing physical technologies for HCI basically can be categorized by the relative human sense that the device is designed for. These devices are basically relying on three human senses: vision, audition, and touch [60][66]. Input devices that rely on vision are the most used kind and are commonly either switch-based or pointing devices [62][63]. The switch-based devices are any kind of interfaces that use buttons and switches like a keyboard [63][66]. The pointing devices examples are mice, joysticks, touch screen panels, graphic tablets, trackballs, and pen-based input [64][66]. Joysticks are the ones that have both switches and pointing abilities. The output devices can be any kind of visual display or printing device [52][66].

The recent methods and technologies in HCI are now trying to combine former methods of interaction together and with other advancing technologies such as networking and animation. These new advances can be categorized in three sections: wearable devices [53], wireless devices [54], and virtual devices [55]. The technology is improving so fast that even the borders between these new technologies are fading away and they are getting mixed together. Few examples of these devices are: GPS navigation systems [56], military super-soldier enhancing devices (e.g.

thermal vision [57], tracking other soldier movements using GPS, and environmental scanning), radio frequency identification (RFID) products, personal digital assistants (PDA), and virtual tour for real estate business [58]. Some of these new devices upgraded and integrated previous methods of interaction. As an illustration, there is the solution to keyboarding that has been offered by Compaq's iPAQ which is called Canesta keyboard. This is a virtual keyboard that is made by projecting a QWERTY like pattern on a solid surface using a red light. Then device tries to track user's finger movement while typing on the surface with a motion sensor and send the keystrokes back to the device [59][66].

1.4 Future Directions in HCI

During the 1990s, the concerns of HCI started to shift towards communication between people enabled by computers. This mirrored the growth in communication networks linking computers together. If, earlier, the concern of HCI was to determine how to let users interact, efficiently and effectively with a computer, now researchers have started asking how users might interact with each other via a computer. Researchers with a background in more socially-oriented sciences, like Anthropology and Sociology, began to engage with HCI. These disciplines not only emphasized the effects of computing on groups of people (or users) but also how computers were interpreted and appropriated by those same groups of users. These disciplines also brought a concern for the social, emotional, as well as technical ways in which the relationship with technology develops. Eventually the approaches of these disciplines were amalgamated so that concerns that had been central before, such as those related to cognitive processing and so forth, were supplemented (and in some ways replaced) by more complex social modeling views and techniques[2].

We make few recommendations for how to bring about a new way of undertaking HCI research and design and to make it more relevant to today's world.

1.4.1 Explore New Ways of Understanding Users

This will require the articulation of diverse methodologies. Over the last decade we have seen, for example, techniques rooted in design-based practices (such as cultural probes) come to prominence. These have complemented existing techniques of understanding that have emerged from scientific and engineering traditions – Human Factors and Cognitive Science, for instance. Other ways of extending and complementing existing techniques will be required beyond design; these may include views from more diverse disciplines and cultural traditions. The use of conceptual analysis as the first stage of a new HCI is a case in point. This technique derives from analytic philosophy, and entails clarifying the systems of meaning and value any particular set of activities involve.

1.4.2 Explore New Ways of Designing and Making

The design and building of prototypes of new devices will need to be undertaken in ways that are directed at particular kinds of user value. These will have to complement and extend existing ways of designing and building which emphasize usability and closeness of fit between prototyped and engineered solutions. In the future, more lightweight, rapid prototyping and design iteration processes will be required, and ones that will allow complex eco-system experiences to be investigated as well as simpler, human-machine relationships. New prototyping tools and technologies will be especially important, allowing the rapid and easy assembly of novel hardware and software to test alongside and within everyday artifacts and living spaces [2].

1.5 Interactive Software

Interactive software are reactive computer systems that require or encourage interaction with the human operator during the course of operation. The interaction between humans and machines is performed through a user interface that is a collection of devices that enable the user to manipulate and to monitor the system status [3]. Computers will be everywhere. Interactive devices will be everywhere. At the end of the twentieth century we saw a progress and this progress will change the way in which people interact with technology. But, still there are many interactive systems that do not provide proper support for the job they are meant for. There are so many usability problems and these usability problems found, suggest that the design process of interactive systems should be more focused to the work that needs to be supported, and to the users who need to operate these systems.

1.6 Human Factors

A human factor is a physical or cognitive property of an individual or social behavior which is specific to humans and influences functioning of technological systems as well as human-environment equilibriums. Human factors focus on how people interact with tasks, machine and the environment with the consideration that humans have limitations and capabilities.

1.6.1 Human Factors Evaluation Methods

Ethnographic analysis: Using methods derived from ethnography, this process focuses on observing the uses of technology in a practical environment. It is a qualitative and observational method that focuses on "real-world" experience and pressures, and the usage of technology or environments in the workplace [27].

Focus Groups: Focus groups are another form of qualitative research in which one individual will facilitate discussion and elicit opinions about the technology or process under investigation. This can be on a one to one interview basis, or in a group session. This method can be used to gain a large quantity of deep qualitative data [28], though due to the small sample size; it can be subject to a higher degree of individual bias [27]. It can be used at any point in the design process, as it is largely dependent on the exact questions to be pursued, and the structure of the group. But it can be extremely costly.

Iterative design: This method is also known as prototyping, the iterative design process seeks to involve users at several stages of design, in order to correct problems as they emerge. As prototypes emerge from the design process, these are subjected to other forms of analysis and the results are then taken and incorporated into the new design. Trends amongst users are analyzed, and products redesigned. This can become a costly process, and needs to be done as soon as possible in the design process before designs become too concrete[27].

Meta-analysis**:** A supplementary technique used to examine a wide body of already existing data or literature in order to derive trends or form hypotheses in order to aid design decisions. As part of a literature survey, a meta-analysis can be performed in order to discern a collective trend from individual variables [27].

Subjects-in-tandem: Two subjects are asked to work concurrently on a series of tasks while vocalizing their analytical observations. This is observed by the researcher, and can be used to discover usability difficulties. This process is usually recorded.

Surveys and Questionnaires: A commonly used technique outside of Human Factors as well, surveys and questionnaire have an advantage in that they can be administered to a large group of people for relatively low cost, enabling the researcher to gain a large amount of data. The validity of the data obtained is, however, always in question, as the questions must be written and interpreted correctly, and are, by definition, subjective. Those who actually respond are in effect self-selecting as well, widening the gap between the sample and the population further [27].

Task analysis: A process with roots in activity theory, task analysis is a way of systematically describing human interaction with a system or process to understand how to match the demands of the system or process to human capabilities. The complexity of this process is generally proportional to the complexity of the task being analyzed, and so can vary in cost and time involvement. It is a qualitative and observational process [27].

Think aloud protocol*:* Also known as "concurrent verbal protocol", this is the process of asking a user to execute a series of tasks or use technology, while continuously verbalizing their thoughts so that a researcher can gain insights as to the users' analytical process. It can be useful for finding design flaws that do not affect task performance, but may have a negative cognitive affect on the user. It is also useful for utilizing experts in order to better understand procedural knowledge of the task in question. This method is less expensive than focus groups, but tends to be more specific and subjective [28].

User analysis*:* This process is based around designing for the attributes of the intended user or operator, establishing the characteristics that define them, creating a persona for the user. A user analysis will attempt to predict the most common users, and the characteristics that they would be assumed to have in common. This can be problematic if the design concept does not match the actual user, or if the identified are too vague to make clear design decisions from. This process is, however, usually quite inexpensive, and commonly used [27].

Wizard of Oz**:** This is a comparatively uncommon technique but has seen some use in mobile devices. Based upon the Wizard of Oz experiment, this technique involves an operator who remotely controls the operation of a device in order to imitate the response of an actual computer program. It has the advantage of producing a highly changeable set of reactions, but can be quite costly and difficult to undertake [19].

While user interface designs are key outputs of any information technology project, the user interface design process itself can also contribute other, ancillary benefits to the development process, quite apart from the final designs [12].

1.6.2 Human Factors in Interactive Software

Shneiderman summarizes, the ultimate success or failure of a computerized system is highly dependent upon the man-machine interaction. Even though interactive software was developed to improve productivity of the user, the user's productivity is often reduced by "cumbersome data entry procedures, obscure error messages, intolerant error handling, inconsistent procedures, and confusing sequences of cluttered screens"[4].

The most important feature of modern interactive computer systems is the level of support they provide for the end user. A lot of work has been done in the field of system designing but human performance capabilities, skill limitation, and response tendencies were not adequately considered in the designs of the new systems. This results in the user frequently becoming puzzled or irritated when trying to interact with the system.

The following human factors activities should be considered when designing and implementing interactive software:

Study Phase Activities: User Involvement: Early user involvement in a system can be very beneficial. End users may have little knowledge of interactive software development, but they can be very good resources for describing system needs and problems associated with their job. In order to determine these needs, one member of the design team should act as liaison with the end user. It has been suggested that the interface design responsibility be limited to one person in order to ensure continuity. It is imperative that the interface liaison establish a good rapport with both the end user and other members of the design team. Early user involvement is also an effective method of overcoming resistance to change. "If the initiators involve the potential resistors in some aspect of the design and implementation of the change, they can often forestall resistance. With a participative change effort, the initiators listen to the people the change involves and use their advice."

Problem Definition: The design team can next proceed to define the problems by combining their own observations with the information obtained. Because the design team's problem definition may vary from that of management or the end user, both groups should be consulted to determine if they concur. This variance is compounded with interactive software in that the future user of the system may not fully understand the nature of real time systems. If either management or the end users do not agree with the problem definition, representatives from each group must meet to resolve any conflict. Once the problem definition is agreed upon, alternative solutions to the problem can then be formulated. Once formulated, the design team should again consult management and the end user to determine the best solution to the problem.

Design Phase Activities: Interface Design : Novice and experienced users differ in both competence and their demands on an interactive system. While frequent users emphasize usability, novices or infrequent users emphasize ease of learning. Software designers must consider both factors when designing the user interface.

Test Requirements: During this phase procedures are to be established for testing each screen format, logical branch and system message for both logic and spelling errors. Since most interactive software packages consist of more than one module or program, with several calls between modules, all logic errors may not be discovered during simple module testing. Integration test procedures need to be established for this purpose.

User Consultation: Design Phase activities may require many months and user requirements may change during that period. Major changes in personnel, technology, or business strategies can promote a change in design requirements. Before proceeding to the Development Phase, management and the user should determine whether or not any changes should be implemented during this release, or deferred for implementation at a later date. Almost all interactive designs require some trade-offs. Hardware limitations, personnel, time, and money are the major reasons alternative designs or procedures are necessary. Since the end users are ultimately affected, they should be included in major trade-off evaluations.

Development/ Operation Phase Activities-Documentation:

Documentation is considered a necessary evil among interactive software designers. Although they recognize the necessity of good documentation, they prefer not to do it. They tend to focus more on technical aspects of the design, rather than how to use the interactive software. Good documentation is especially critical with an interactive system, because user productivity can be highly affected when time is spent looking for information which either does not exist, or is poorly documented. For these reasons, technical writers should be consulted to write user interface manuals, as they are more skilled in explaining how a system operates. Operation manuals for interactive software have traditionally consisted of an alphabetical listing of commands and their definitions. Manuals of this type provide easy reference for the experienced user, but novice users find them very difficult to understand. One alternative is to present the method of performing the most frequently used procedures, then listing and defining the computer commands necessary to perform the procedure. It is also useful to provide a listing of the commands in alphabetical order. Although this method is redundant, it is much more easily understood by novice users.

Module and Integration Testing: Good design and thorough testing require time and money, but are worthwhile investments. A thoroughly tested, well-designed system is more easily implemented and requires less maintenance. As Shneiderman states, "Faster task performance, lower error rates, and higher user satisfaction should be paramount goals of the designer". Each screen format, logical branch, and system message is to be thoroughly tested. Any modules containing errors should be corrected and retested. Module integration testing should be performed when all modules have passed individual tests. Some changes may be necessary to protect the system from the user. Users of interactive systems do not always utilize the system in the same manner in which it was designed. This is especially true when they are confused or frustrated. They start issuing a variety of commands, or pressing keys at random to get out of their present dilemma. This could be catastrophic. Although it is impossible to predict all user actions, the test team should attempt to place the system under rigorous testing. If a catastrophic situation should occur, code changes would be necessary to remedy the situation.

System Testing Once integration testing procedures are complete, the system should be tested with the anticipated user population as the subjects. By this time the design team is familiar with the software, and not likely to use the directions and documentation. The novice user is highly dependent upon good directions and documentation and is therefore more likely to detect confusing or erroneous information. If the design team is able to unobtrusively observe the users during system test, they may witness situations which could lead to user frustration and/or error. They should interview the test subjects to determine other possible problems. At this time, minor changes to the software can be made. Major design changes which enhance the software, but not correct errors, should be scheduled for implementation at a later date.

User Training: Holt and Stevenson state that when human errors occur, interactive system designers tend to blame the human interface or data display designs. They cite one example of a company which had an influx of human errors when a new system was implemented. The system designers' solution to the problem was to replace the data entry clerks with more highly skilled personnel and substitute CRT terminals for input forms. Before implementing this solution, a special task force was assigned to work with the users. They discovered that the data entry clerks were not properly trained in the use of the system, and that the input forms were cumbersome. System performance soon rose above the acceptable level when input forms were redesigned and users were trained. In addition to the initial training session, ongoing training to accommodate new or transferred employees needs to be considered. Computer assisted instruction (CAI) is an effective method of training multiple users. The major factor to be considered with CAI training is that it is critical that the instruction modules be updated each time significant changes are made to the system. User experimentation and browsing should be encouraged. Prior to total implementation of the system, a pilot system should be implemented to allow this. Pilot systems help the end user overcome any fears of the system. They are able to become acquainted with the system without the fear of accidently destroying it.

***Code and Documentation*:** Changes Regardless of how careful one has been, there will always be unforeseen code and documentation changes. Some changes may be due to actual errors, while others may be due to user confusion or frustration [4].

1.7 User Interface

Nowadays, computers play a very important role, that is to say, as a communication tool between people. This introduces the interface between human and machines as a key player, therefore the importance of these interfaces. The user interface (UI) of a computer program is the part that handles the output to the display and the input from the person using the program. Designing user interfaces for interactive systems is a task that depends, among other things, on the designer's creativity. Traditional UI design methods and techniques help in the design tasks but do not usually provide new ideas by themselves [6][20][21].The user interface has been recognized as one of the most important elements of a software project. It had been estimated that 48% of work on a project goes into the design and implementation of the user interface [14]. Recently, as confirmation, [15] noted the importance of Human Computer Interaction suggesting at least 50% of programming code is devoted to the user interface [5]. The abstract characteristics of user interfaces: learn ability, ease of use, error tolerance, efficiency, user satisfaction and others [9] – which make interactive systems usable and these can be achieved through a careful design and construction process [6].

Interface evaluation is the process of assessing the usability of an interface and checking that it meets user requirement. Therefore, it should be the normal verification and validation process for software systems. Ideally, an evaluation should be conducted against a usability specification based on usability attributes, for these usability attributes can be devised. For example, in a learn ability specification, you might state that an operator who is familiar with the work supported should be able to use 80% of the system functionality after a three-four hour training session[51]

User interface design can be approached from different perspectives. There is the view from sociology: how do computers influence the behavior of their users? Do computers make tasks more enjoyable, easier, and less stressful? Psychology can look at the way people's skills, learning ability and character are cooperating with or working against certain styles of interaction. Ergonomics studies the relation between various measures of people's abilities to perform a task and measurable properties of their tools, both hardware and software. Last, but not least, there is computer science, which concentrates on the hardware and software that makes various styles of interaction possible. Computer science studies efficiency both with respect to the machine and with respect to the user; it creates new methods and develops the coordination between software, documentation and training. Designers of user interfaces, especially if they come from computer science, are usually not aware of the differences among people. For lack of better information, they usually take themselves as examples of typical users. This is fine if they are designing for themselves or fellow-designers, but their view of a system is likely to be very different from the view the user will have. To simplify in a gross way: if those users had the same character as the designer, they would have become designers themselves.

1.7.1 The User's Perspective of User Interface

When a program exhibits complex behavior, the users of the program are likely to build a model that is anthropomorphic. They project into the system motives and desires they themselves would have if they were in the system's place. Alternatively, they transfer their model of some other well-known person or system to the new system. These are powerful, but often unconscious phenomena. People have a tendency to see causes and effects in events that occur together or soon after one another, even when the co-occurrence is purely accidental. This may lead to false beliefs that may even extend to things outside the computer, like a belief that a certain program is sensitive to the force with which you press the keys on the keyboard, or even that the program only works if you lift your elbows of the table when it is running. Usually, this points to a lack of trust in the program or an inadequate grasp of its limitations. The user may fail to see a pattern at all and fall back on describing a system's behavior as random. Most systems other than games aren't random. The apparent randomness is caused by gaps in the provided information, by things not discernible to humans (like nanoseconds) or by limitations in human memory: the information that explains a certain behavior may have been given too long ago, there may have been too much intervening information, or it may have been shown in a position or manner that seemed unrelated to the present situation [10].

1.7.2 Efficiency of a User Interface

If computer systems are to improve and make end users' tasks easier, system designers and instructional designers should address processes that will encompass and meet the needs of the average user. Designers should also try to account for the average users' behavior patterns and attempt to incorporate these patterns into the design of user-friendly systems [46]. There are two types of efficiency at play in a user interface. The first is the ease with which it allows a user to perform a task, as perceived by the user. Time plays only a minor role. This type of efficiency can be enhanced by providing sensible defaults, choices adapted to the user's expectations and shortcuts. Efficiency in a more technical sense is the time required to perform a certain action: how fast can a menu be drawn, how fast a typed sentence can be parsed.

Clearly, efficiency is something to strive for. But there is the strange effect that computers can sometimes be too fast. At least that is sometimes claimed with the following example: slow compilers force programmers to think more about their code, resulting in fewer edit-compile cycles than programmers who rely on the compiler to find errors, eventually leading to faster development.

Computer hardware has become very powerful and continues to grow in speed and capabilities. There is a definite lack of interfaces that will allow more than just the experts to make use of that power. It is the task of the interface designer to create effective interfaces for different kinds of users. He has few hard and fast rules to go by, but there are some rules of thumb. One of the most important technologies is direct manipulation, graphics is another, and the combination promises to be even more powerful [11].

1.7.3 Type of User Interfaces

Paradigms, metaphors, mental models and personas are driving forces behind the user interface and design employed in a particular system. There are three commonly recognised user interfaces in use today. The Graphical User Interface, which is possibly the most familiar to most users; the Voice User Interface, one that is rapidly being deployed in many aspects of business; and the Multi-Modal Interface, a relatively new area of research that combines several methods of user input into a system.

1.7.3.1 Graphical User Interface

Graphical user interfaces make computing easier by separating the logical threads of computing from the presentation of those threads to the user, through visual content on the display device. This is commonly done through a window system that is controlled by an operating system's window manager. The WIMP (Windows, Icons, Menus, and Pointers) interface is the most common implementation of graphical user interfaces today. The appeal of graphical user interfaces lies in the rapid feedback provided by the direct manipulation [23] that a GUI offers [23]. Direct manipulation interfaces provide the following features [23]:

The robustness of the direct manipulation interface for the desktop metaphor is demonstrated by the documents and folders being visible to the user as icons that represent the underlying files and directories [22]. With a drag-and-drop style command, it is impossible to make a syntactically incorrect operation. For example, if a user wants to move a file to a different folder, the move command itself is guaranteed to be syntactically correct; and even though the user may make a mistake in placing the file in the wrong place, it is relatively easy to detect and recover from those errors. While the document is being dragged, continual visual feedback is provided to the user, creating the illusion that the user is actually working in the desktop world [22].

1.7.3.2 Voice User Interface

Voice User Interfaces (VUIs) use speech technology to provide people with access to information and to allow them to perform transactions.

VUI development was driven by customer dissatisfaction with touchtone telephony interactions, the need for cheaper and more effective systems to meet customer needs, and the advancement of speech technology to the stage where it was robust and reliable enough to deliver effective interaction. With the technology finally at the stage where it can be effectively and reliably used, the greatest challenge remains in the design of the user interface [24].A Voice User Interface is what a person interacts with when using a spoken language application. Auditory interfaces interact with the user purely through sound. Speech is input by the user, and speech or nonverbal audio is output by the system [24].

People learn spoken language implicitly at a very young age, rather than explicitly through an education system. Therefore, speakers generally do not explicitly think about the technical means of getting a message across, they simply need to focus on the meaning of the message they wish to convey. Consequently, VUI designers must focus on understanding the conventions, assumptions and expectations of conversation instead of its underlying constructs [24].

VUIs are comprised of three main elements [24]:

- Prompts, also known as system messages, are the recorded or synthesised speech played to the user during the interaction.
- Grammars are the possible responses users can make in relation to each prompt. The system cannot understand anything outside of this range of possibilities.

- Dialog logic determines the actions the system can take in user's response to a prompt.

Additionally, auditory interfaces provide opportunity to use non-verbal audio such as background music and earcons to create an auditory environment for the user, and thus creating a unique "sound and feel" for a business or application [24].

Auditory interfaces provide new ways of supplying information, enabling transactions, and facilitating communication. They have benefits for both the company and the end user. Speech systems normally save companies significant amounts of money by lowering call abandonment rates and increasing automation rates, as well as reducing toll call charges through shorter call durations. They can enable companies to extend their reach to customers who don't have web access and those who desire mobility whilst retaining the same level of service. Additionally, auditory interfaces can solve problems or offer services that were not available or possible in the past, such as automated personal agents that use speech technology to act as a personal assistant [24]. For the end user, voice systems can be more efficient and intuitive by drawing on the user's innate language skills and simplifying input sequences. Telephones are ubiquitous, which makes speech systems mobile. This also means that users can employ them anywhere and without occupying the user's hands and eyes, which makes speech systems easily accessible. A well-designed speech system can thus free up the user's time by more efficiently meeting their needs [24].

Aside from speech recognition systems, other speech technologies include Text-to-Speech Synthesis and Speaker Verification. Speaker Verification involves collecting a small amount of a person's voice to create a voice template, which is used to enrol a person into a system and then compare future conversation. The system can be used, for example, to replace personal identification numbers [24].

Text-to-Speech technology, on the other hand, synthesises text into speech. The technology has improved significantly in recent times, and although it does not yet duplicate the quality of recorded human speech, it is still a good option for creating messages from text that cannot be predicted, such as translating web pages for blind users [24].

Understanding the nature of speech systems as well as their benefits and limitations, is necessary in order to design effective Voice User Interfaces. Unique design challenges presented by VUIs are due to the transitory and non-persistent nature of their messages [24]. Messages are invisible – once the user hears the message, it is gone, with no screen to display the information for further analysis. The user must usually decide then and there how they will respond to a prompt. The cognitive limitations of the user necessarily restrict the quantity of information that can be presented by the auditory interface at any one time. Thus voice interface designs should not unnecessarily challenge the user's short term memory, and they should provide a mechanism for the user to adjust the pacing of the interaction to better suit their own needs [24].

1.7.3.3 Multi-modal User Interface

Multi-modal interfaces attempt to address the problems associated with purely auditory and purely visual interfaces by providing a more immersive environment for human-computer interaction. A multi-modal interactive system is one that relies on the use of multiple human communication channels to manipulate the computer. These communication channels translate to a computer's input and output devices. A genuine multi-modal system relies on simultaneous use of multiple communication channels for both input and output, which more closely resembles the way in which humans' process information [22][23].

In the field of psychology, Gestalt Theory is used to describe a relationship where the whole is something other than the sum of its parts. This theory has recently been used to describe a new paradigm for human-computer interaction, where the interface reacts to and perceives the desires of the user via the user's emotions and gestures [25]. This paradigm is called the gestalt User Interface (gUI) and paves the way for a truly personalised user experience.

Marriott & Beard [2004] write that "a gUI is an interface where input to the application can be a combined stimulus of text, button clicks, and analysed facial, vocal, and emotional gestures". The user's emotions, captured through input devices such as video and audio, are translated into input for a gUI, and the program's response is rendered using a markup language.

The value of an interface that can interpret a user's emotions has applications in fields ranging from business management, safety, and productivity, to entertainment and education. For example, if a program could recognise that the user was getting frustrated, it could modify its behaviour to compensate. When evaluating the text-and-GUI-based Mentor System application used by students from Curtin University to assist with their assignments, Marriott [2003] found that "personality conflict" occurred between the system and some users. For example, one user became intensely annoyed at the beeping sound the program made, while another user found the program to be discourteous. Marriott suggests that incorporating a dynamically adjusting module into the user interface could eliminate or reduce some of these problems [25][26].

Marriott and Beard [2004] suggest that an anthropomorphic metaphor should be used to specify complete interaction with an Embodied Character Agents (ECA) that incorporates emotion and gesture analysis. The Mentor System, even though it does not incorporate an ECA, also uses an anthropomorphic metaphor of a virtual lecturer.

One of the more well-known commercial examples of the ECA is the virtual newscaster at www.ananova.com. Whilst the implementation technology for Ananova has yet to meet the empathy requirements of a gestalt user interface, such as realistic facial animation, the speech synthesis software is quite good and it nevertheless provides a good example of what could be achieved when the technology is realized. Another product that makes use of multi-modal interfaces is Avatar-Conference, an alternative to video-conferencing that uses avatars to represent the conference participants in a virtual environment. Although it does not incorporate dynamically adjusting modules, it nevertheless provides more than one mode of user interaction [25].

1.7.3.4 Other User Interface

Many other paradigms for human-computer interaction exist. Perhaps one of the best known paradigms is the World Wide Web. The web itself did not provide any technological breakthroughs, because all the required functionality, such as transmission protocols, hypertext and distributed file systems, already existed. The breakthrough came with the advent of the browser and HTML, which enabled easy access to information on the internet, first through academia and then through business and leisure circles [23].

Ubiquitous, or pervasive, computing is another emerging paradigm for computing. The aim of pervasive computing is to create a computing infrastructure that permeates our physical environment such that we no longer notice the computer [22][23]. Technologies that facilitate pervasive computing include wireless networks, voice and vision systems, nanotechnology and molecular electronics [22]. User interfaces for modern applications must support a rich set of interactive features [18].

1.7.4 User Interface Principles

User Interface Principles comprises mainly of User Familiarity, Consistency, Minimal Surprise, Recoverability, User Guidance and

User Diversity help guide the design of user interfaces. When making user interface design decisions, the human factors are the critical factor for the design principles.

The following principles are fundamental to the design and implementation of effective interfaces, whether for traditional GUI environment for the web [48].

1.7.4.1 The principle of User Profiling

One of the main objectives of “User Profiling” is to create the user interfaces so as to make the work of the user easier.The interface is the key in providing the user with the ease of making use of even the most complex applications in an efficient and simplified manner. User models works on the theory of creating “profile” of all the possible users. As a result, a detailed description of the user’s specification such as user’s goal, user’s skill, user’s experience and user’s needs, etc can be formulated in an organized manner [49].

1.7.4.2 The principle of Metaphor

The principle of Metaphor states that an interface should implemented behavior from system that the users are familiars with. It increases the understandability of even most complex application by introducing the element of being familiar with the environment user has been working as for this entire for have.

Furthermore, metaphor can be further extended to provide functionality that is not only acceptable but also usable in the whole interface. This property of “extendibility”, differentiate between a strong metaphors from a weak one.

1.7.4.3 The principle of Exposure

The principle of Exposure says that the user should be aware of all the functions & functionality that is available to him via the interface. The interface is to provide an environment that should be able to concentrate and the sensory details of the environment rather than the perfection of abstraction. The interface should be designed in such a way that it is “Sensible” to the general population, rather than being only “Intuitive”.

1.7.4.4 The principle of Coherence

There’s been some argument over whether interfaces should strive to be “Intuitive”, or whether an “Intuitive” interface is even possible. However, it is certainly arguable that an interface should be coherent—in other words logical, consistent and easily followed.

Internal Consistency means that the program’s behaviors make “sense” with respect to other parts of the program. For example, if one attribute of an object (e.g. color) is modifiable using a pop-up menu, then it is to be expected that other attributes of the object would also be editable in a similar fashion. One should strive towards the principle of “least surprise”. External Consistency means that the program is consistent with the environment in which it runs. This includes consistency with both the operating system and the typical suite of applications that run within that application.

1.7.4.5 The principle of State Visualization

Each change in the behavior of the program should be accompanied by a corresponding change in the appearance of the interface. One of the big criticisms of “modes“in interfaces is when a program that many of the class is “bad example” programs have modes that are visually indistinguishable from one another. Similarly, when a program changes its appearance, it should be in response to a behavior change; a program that changes its appearance for apparent reason will quickly teach the user to depend on appearances for clues as to the program’s state.

1.7.4.6 The principle of Shortcut

The principle of Shortcut says that in order to achieve a task or to get a work done, the accessing methods should not only be abstract but concrete too. Once a user has become experienced with an application, she/he will start to build a mental model of that application. She/he will be able to predict with high accuracy what the results of any particular user gesture will be in any given context. At this point, the program’s attempts to make things “easy” by breaking up complex actions into simple steps may seem cumbersome.

1.7.4.7 The principle of Focus

The principle of Focus states that some aspects of the User Interface enjoy more attention of the user than others. The user finds same attributes and aspects of the user interface more attractive as compared to others. The human mind is a highly non-linear and has a perfect coordination with the eyes. Thus our eyes are more observant towards animated areas rather than the static are in an application.

1.7.4.8 The principle of Help

There are five basic types of Help: Goal-Oriented, Descriptive, Procedural, Interpretive and Navigational [47]. A help browser or tool tips can be useful in responding to questions related to procedural help, but these can sometimes be more efficiently addresses using “cue card’, interactive “guides”, or “wizards” which guide the user through the process step –by-step. The interpretive type of help has not been well addressed in current applications, although well written error messages can help. The navigational type can be answered by proper overall interface design, or by creating an application “roadmap”. None of the solutions listed in this paragraph are final or ideal; they are simply the ones in common use by many applications today.

1.7.4.9 The principle of Safety

The principle of Safety states that the interface should develop confidence amongst the user by providing them a feeling of safety. The User Interface should ensure that the novice user should not feel him at risk while accessing the program: i.e. he should not feel unsafe while navigating, accessing or a doing a task. A certain level of comfort should be provided to the user, in almost all the situation.

In a new environment, any novice will feel unconfident; in fact a novice user will be unconfident not only regarding his technical but also his intelligent skills. The novice users should be well assured that they will be protected from their own lack of skill. At the same time, the expected user should be able to turn off the safety nets, so that it may not act as a bottleneck in their work. For these reason "Safety Level" is an important application configuration options.

1.7.4.10 The Principle of Context

The principle of Context states that the user's activity should be limited to a well defined context unless significant reason is there to support his freedom to access more.

Each user action takes place within a given context the current document, the current selection, the current dialog box. A set of operations that is valid in one context may not be valid in another. Even within a single document, there may be multiple levels for example, in a structured drawing application, selecting a Text Object(which can be moved or resized) is generally considered a different state from selecting an individual character within that Text Object. It's usually a good idea to avoid mixing these levels. It is important to make a mental makeup of the relationship between contexts.

1.7.4.11 The principle of User Testing

The principle of User Testing states that the inevitable defects in the design of the user interface should be spotted. Generally, it is the fundamental defects in a user interface that the designer of the interface can spot. There are however various kinds of defects that are not easy to spot. The testing of user interface is actually the process of testing of the interface using actual end users. It is an extraordinary effective technique through which various design defects can be discovered [47][49].

1.8 User Interface Software Tools

There is no question that research in the area of user interface software tools has had an enormous impact on current practice of software development. Virtually all applications today are built using window managers, toolkits and interface builders that have their roots in the research of the 70’s, 80’s and 90’s[37]. All the forms of computer-human interaction will need to be supported by appropriate tools. The interfaces of the future will use multiple modalities for input and output (speech and other sounds, gestures, handwriting, animation, and video), multiple screen sizes (from tiny to huge), and have an "intelligent" component ("wizards" or "agents" to adapt the interface to the different wishes and needs of the various users). The tools used to construct these interfaces will have to be substantially different from those of today. Whereas most of today's tools well support widgets such as menus and dialog boxes, these will be a tiny fraction of the interfaces of the future [38].

There are software libraries and tools that support creating interfaces by writing code. The User Interface Management System provides the functionality of a GUI builder as well as the capability to define and execute the functionality of the displays from within the tool. A UIMS can also generate code to build a version of the interface that will execute independently of the tool.

1.8.1 Importance of User Interface Tools

Software development and user interface researches have a long and intertwined history. User interface community has developed and used various software tools, such as window managers and toolkits, interactive graphical tools, component systems, event languages, scripting languages, hypertext, object-oriented programming, but also some promising but less successful products, such as, user interface management systems and formal language-based tools. However, these existing user interface software tools solve just a part of the human-computer interaction problem. In addition, they use specialized notation and conventions, what limits their practical usage and complicates integration of the user interface with the rest of the software system. Furthermore, many of these existing tools will not be able to support development of user interfaces in the future [39].

1.8.2 EVALUATING USER INTERFACE TOOLS

There are many dimensions along which you might evaluate user interface tools. The importance given to these different factors will depend on the type of application to be created, and the needs of the designers.

- ***Depth:*** How much of the user interface does the tool cover? For example, Interface Builders help with dialog boxes, but do not help with creating interactive graphics. Does the tool help with the evaluation of the interfaces?
- ***Breadth:*** How many different user interface styles are supported, or is the resulting user interface limited to just one style, such as a

sequence of cards? If this is a higher-level tool, does it cover all the widgets in the underlying toolkit? Can new interaction techniques and widgets be added if necessary?

- ***Portability:*** Will the resulting user interface run on multiple platforms, such as X, Macintosh, and Windows?

- ***Ease of Use of Tools:*** How difficult are the tools to use? For toolkits and most language-based higher-level tools, highly trained professional programmers are needed. For some graphical tools, even inexperienced end-users can generate user interfaces. Also, since the designers are themselves users of the tools, the conventional user interface principles can be used to evaluate the quality of the tools' own user interfaces.

- ***Efficiency for Designers:*** How fast can designers create user interfaces with the tool? This is often related to the quality of the user interface of the tool.

- ***Quality of Resulting Interfaces:*** Does the tool generate high-quality user interfaces? Does the tool help the designer evaluate and improve the quality? Many tools allow the designer to produce any interface desired, so they provide no specific help in improving the quality of the user interfaces.

- ***Performance of Resulting Interface:*** How fast does the resulting user interface operate? Some tools interpret the specifications at run-time, or provide many layers of software, which may make the resulting user interface too slow on some target machines. Another consideration is the space overhead since some tools require large libraries to be in memory at run-time.

- ***Price:*** Most personal computers and workstations today come with a free toolkit. Commercial higher-level tools can range from $200 to $50,000, depending on their capabilities.

- ***Robustness and Support:*** In one study, users of many of the commercial tools complained about bugs even in the officially released version [50], so checking for robustness is important. Since many of the tools are quite hard to use, the level of training and support provided by the vendor might be important.

Naturally, there are tradeoffs among these criteria. Generally, tools that have the most power (depth and breadth) are more difficult to use. The tools that are easiest to use might be most efficient for the designer, but not if they cannot create the desired interfaces [37].

1.8.3 Current Scenario of User Interface Tools

Over the years, the human-computer interaction community has produced a large number of software tools and environments for the design and development of user interfaces [40][41]. But still user interface designers have very few options in terms of software tools when designing their system. Designers always complaint that they have very limited options. Many tools are created with the distinct purpose of performing a single design task, such as sketching a user interface [37][41], or analyzing a task model [41][42]. In their intended tasks, these tools often perform satisfactorily.

1.8.4 The Future of User Interface Software Tools

Results of several studies shown that designers must know the user. Designers should have data about the user. The data may include age, experience and capabilities etc. because these human factors affect the usability of the system.

As Ying Wang et al describe, any designed system should be user-friendly. However, in many cases, it is not easy to satisfy this requirement because it is difficult to quantify the human's requirement or the human's performance [43].To test system with real user designers can use prototype of the system. By testing the interaction with the user, the designer can easily find the problems and then quickly modify the design and test again. Sometimes, iterative design at the cost of time and money is necessary. In order to reduce the iterations for time and cost concern, designers can use data about the target user and designed the system according to their needs. Currently, developing this next generation of interfaces is difficult, time-consuming, and requires a high level of technical expertise [44].

1.9 Human Factors Based User Interface Design

Before designers start designing of user interface they need information about the target users. These factors can be very important in other development activities, primarily in analysis and design, as many early design decisions are based on these specifications. Considering human factors early in the development process goes a long way toward improving quality of the system. Specifying details about users and contexts is important as designers should become familiar with users' psychological characteristics(for example ,cognitive abilities, motivation), level of experience, domain and task characteristics, cultural background, as well as their physical attributes (for example, age, vision and hearing) [39].

Specific applications can sometimes be used with little difficulty by cognitively impaired users. Part of the reason is the designers have very less knowledge about human factors. Source of knowledge for designers are manuals, technical reports or guidelines etc. But mostly designer want computer based design aids which should be integrated in their design tools. Several user interface tools are available for obvious physical disabilities. To reach the goal of designing human factor based user interfaces, the software designer needs excellent tools**.** Especially for graphical and hypermedia user interfaces. These tools should allow the designer to concentrate on the human factors along with design process and on the quality of the design results [45]. There is a definite need of such tools which help them to design human factor based user interface.

REFERENCE

[1] Wild, P.J. and Johnson, P.,(2004), “**Deepening Consideration of Temporal Factors in Task Knowledge Structures”** Position paper presented at, Workshop on 'Temporal Aspects of Work for HCI'. CHI'2004,Bath,pp.1,4

[2] Harper Richard, Rodden Tom, Rogers Yvonne, Sellen Abigail(2008), “**Being Human Human-Computer Interaction In The Year 2020”**, Microsoft Research Ltd, England,pp.80,85

[3] Loer Karsten(2003), **“Model-based Automated Analysis for Dependable Interactive Systems”,** pp.21,

[4] Yestingsmeier Jan (1984), **“Human Factors Considerations In Development Of Interactive Software”**,SIGHI bulletin,Denton TX, pp1-3

[5] Peslak Alan(2005), **“A Framework and Implementation of User Interface and Human-Computer Interaction Instruction”**, Journal of Information Technology Education,USA,vol. 4,pp.2,

[6] Vinicio Leonel, Díaz Morales, Arturo Jorge, Perezgil Rivera(2010), **“Structured User Interface Design as a Lateral Thinking Tool for User Interface Design”**, IEEE Computer Society, Guatemala,pp.1

[7] A. Dix, J. Finlay, G. Abowd, R. Beale(1998), “**Human-Computer Interaction”**, Second Edition, Pearson Education Limited, England

[8] E. Butow(2007), **"User Interface Design for Mere Mortals"**, Pearson Education, Inc., Indiana

[9] W. Quesenbery, **"What Does Usability Mean: Looking Beyond 'Ease of Use'"**. Proceedings of the 48th Annual Conference, Society for Technical Communication, 2001

[10] Bos Gijsbert (1993), **" Rapid User Interface Development With The Script Language GIST"**,University of Groningen, Netherlands, pp.27

[11] Hix,D. and Hartson H.R.(1993), "**Developing User Interfaces"**, Wiley , New York.

[12] Morphett Matthew, Morris Shane (2006), **"User Interface Design as a Facilitator of IT Project Communication – A Case Study"**, OZCHI 2006 Proceedings, Australia,pp.2

[13] Preece Jenny (1994), **"Human Computer Interaction"**, Addison Wesley pp.6,26,

[14] Myers, B. & Rosson M. (1992), **"Survey on user interface programming"**, Proceedings SIGCHI'92: Human Factors in Computing Systems, 195-202.

[15] Douglas, S., Tremaine, M., Leventhal, L., Wills, C., & Manaris, B. (2002), **"Incorporating human-computer interaction into the undergraduate computer science curriculum"**, Proceedings of the 33th SIGCSE technical symposium on Computer science education, 211-212.

[16] **http://www.optimum-web.co.uk/images/HCI_diagram.gif** Retrieved September 10, 2004

[17] H.Willemse and G.Lind_er(1988), **"Software ergonomie. Schoonhoven"**

[18] J¨arvi Jaakko, Marcus Mat, Parent Sean, Freeman John(2009), **"Algorithms for User Interfaces"**, ACM GPCE'09,USA,pp.1

[19] http://en.wikipedia.org/wiki/Human_factors

[20] J. Löwgren, E. Stolterman(1999), **"Methods & tools: design methodology and design practice"**, ACM Interactions Vol. 6, pp. 13-20

[21] K. Nakakoji, Y. Yamamoto, M. Ohira, **"A Framework that Supports Collective Creativity in Design using Visual Images" ,** Proceedings of Creativity and Cognition, Loughborough, UK, 1999

[22] Nanni Patrizia (2004),**"Human Computer Interaction: Principles of Interface Design"**

[23] Dix, A., Finlay, J., Abowd, G. D. & Beale, R.(2004), **"Human-Computer Interaction"**, 3 edn, Addison-Wesley Pearson Education, London,pp.171

[24] Cohen, Michael. H, Giangola, James. P. , Balogh, Jennifer(2004), **"Voice User Interface Design"**, Addison-Wesley, Boston, pp.4-8,13

[25] Marriott, A. & Beard, S. (2004), **"gUI: Specifying Complete User Interaction"**, in H. Prendinger & M. Ishizuka, (eds.), Life-Like

Characters. Tools, Affective Functions and Applications, Springer-Verlag, Berlin, pp. 111-134

[26] Marriott, A. (2003), "**Mentor System Customisation",** in AAMAS 2003: Embodied Conversational Characters as Individuals Workshop, Melbourne

[27] Carrol, J.M. (1997), **" Human-Computer Interaction: Psychology as a Science of Design"**, Annual Reviews, Inc., USA,pp.48, 61-83

[28] http://www.Nedarc.org

[29] Wickens, C.D.; Lee J.D.; Liu Y.; Gorden Becker S.E. (1997), **"An Introduction to Human Factors Engineering"**, 2nd Edition. Prentice Hall. ISBN 0321012291.

[30] Kuusela, H., Paul, P. (2000), **"A comparison of concurrent and retrospective verbal protocol analysis"** , The American Journal of Psychology, 113, 387-404

[31] Hollan James D.(2003), "**Human-Computer Interaction",**The MIT Encyclopedia of the Cognitive Sciences,Robert Wilson and Frank Keil (Eds), MIT Press, San Diego,pp.1

[32] Hinze-Hoare V.(2007),**"Review and Analysis of Human Computer Interaction (HCI) Principles "**, Southampton University, pp.2

[33] Savidis, A., Stephanidis, C., (2006), **"Inclusive development: Software engineering requirements for universally accessible**

interactions, Interacting with Computers", 18 Elsevier, www.elsevier.com/locate/intcom, pp.8,37

[34] Wald, M. (2005), **"Enhancing Accessibility through Automatic Speech Recognition"**, Proceedings of Accessible Design in the Digital World.

[35] Karam. M., and Schraefel, M. C., (2005), **"A Taxonomy of Gestures in Human Computer Interaction"**, ACM Transactions on Computer-Human Interactions 2005, Technical report, Electronics and Computer Science, University of Southampton,

[36] Evertsson Gustav(2001), **"Human Computer Interaction"**, pp.6-8

[37] Myers Brad, Hudson Scott E., Pausch Randy(1999), **"Past, Present and Future of User Interface Software Tools"**, ACM Transactions on Computer-Human Interaction , Carnegie Mellon University , pp.1,32-33

[38] Myers Brad , Hollan Jim , Cruz Isabel (1996),**"Strategic Directions in Human Computer Interaction"**, ACM Computing Surveys,pp.12

[39] Obrenović Željko , Starčević Dušan(2006),**"Adapting the Unified Software Development Process for User Interface Development"**, ComSIS,pp.34-35

[40] Norman, D. (1999), "**The Invisible Computer**", Basic Books.

[41] Lee Carrie A. (2007),**"HCI & Aesthetics: The Future of User Interface Design"** , www.carrielee.net

[42] Andrews Keith(2011),**"Human-Computer Interaction, IICM Graz University of Technology"**, Graz,

[43] Wang Ying, Zhang Wei, Bennis Fouad, Chablat Damien(2006) , **"An Integrated Simulation System for Human Factors Study"**, The Institute of Industrial Engineering Annual Conference and Exhibition, Orlando, FL, USA,pp.1

[44] Olsen Dan R. Jr., Klemmer Scott R. (2005), **"The Future of User Interface Design Tools"**, ACM, CHI 2005,USA,pp.2

[45] Reiterer Harald (1993), **"The Development of Design Aid Tools for a Human Factor Based User Interface Design"**, in: Conference Proceedings. 1993, IEEE International Conference on Systems, Man and Cybernetics.Volume 4, IEEE,pp.1

[46] Turner Sherri G.(1998), **"A Case Study Using Scenario-Based Design Tools and Techniques in the Formative Evaluation Stage of Instructional Design: Prototype Evaluation and Redesign of a Web-Enhanced Course Interface"** , ProQuest Dissertations and Theses, Virginia

[47] Laurel Brenda (1991), **"The Art of Human Computer Interface Design"**,Addison-Wesley,ISBN 0-201-51797-3

[48] Myhew,D.J.(1992), **"Principle and Guidelines in Software User Interface Design"**,Englewood Cliffs,NJ:Prentic-Hall.

[49] Tognazzini Bruce (1991), **"Tog on Interface"**, ,Addison-Wesley,ISBN 0-201-60842-1

[50] Myers, B. A. And Rosson, M. B.(1992), **" Survey on user interface programming. In Human Factors in Computing Systems"**, Proceedings SIGCHI'92 (Monterey, Ca., May). ACM, New York, 195–202.

[51] Vorderer P.(2000), **"Interactive entertainment and beyond, Media Entertainment: The phychology of its appeal"** ,pp.21-36

[52] Nielsen J., Kaufman Morgan (1994), **"Usability Engineering"**, San Francisco

[53] Barfield Woodrow , Caudell Thomas(2001), **"Fundamentals of Wearable Computers and Augmented Reality"**, Lawrence Erlbaum Associates, Mahwah

[54] Yacoub Michel Daoud(2002), **"Wireless Technology: Protocols, Standards, and Techniques"**, CRC Press, London

[55] McMenemy K. ,Ferguson S.(2007), **"A Hitchhiker's Guide to Virtual Reality"**, A K Peters, Wellesley

[56] Global Positioning System, "Home page", **http://www.gps.gov/,** visited on 10/10/2007

[57] Burnay S.G., Williams T.L. and Jones C.H(1988), **"Applications of Thermal Imaging"**, A. Hilger, Bristol

[58] Chai J. Y., Hong P. and Zhou M. X.(2004), **"A probabilistic approach to reference resolution in multimodal user interfaces",**Proceedings of the 9th International Conference on Intelligent User Interfaces, Funchal, Madeira, Portugal, pp 70-77

[59] Bretz E.A.(2002), **“When work is fun and games”**, IEEE Spectrum, 39(12), pp 50-50

[60] Te'eni Dov , Carey , Jane M. , Zhang Ping (2007), “**Human computer interaction: developing effective organizational information systems”**, University of British Columbia A Balanced Look at HCI in Business

[61] Te’eni DOV (2006), **“Designs that fit: an overview of fit conceptualization in HCI”**, in P. Zhang and D. Galletta (eds), Human-Computer Interaction and Management Information Systems: Foundations, M.E. Sharpe, Armonk . pp.1

[62] Greenstein J.S.(1997), **“Pointing devices”,** in M.G. Helander, T.K. Landauer and P. Prabhu (eds), Handbook of Human-Computer Interaction, Elsevier Science, Amsterdam.

[63] Myers B.A.(1998), **“A brief history of human-computer interaction technology”**, ACM interactions, 5(2), pp 44-54 .

[64] Shneiderman B.(1998), **“ Designing the User Interface: Strategies for Effective Human-Computer Interaction (3rd edition)”**, Addison Wesley Longman, Reading .

[65] Murata A.(1991), **“An experimental evaluation of mouse, joystick, joycard, lightpen, trackball and touchscreen for Pointing - Basic Study on Human Interface Design”**, Proceedings of the Fourth International Conference on Human-Computer Interaction 1991, pp 123-127 .

[66] Karray Fakhreddine , Alemzadeh Milad, Saleh Jamil A., Arab Mo N., (2008), **“Human-Computer Interaction: Overview on State of the Art”**, International Journal on Smart Sensing and Intelligent Systems, Vol. 1, NO. 1 pp.1-3

[67] Shneiderman B., Plaisant C.(2004), **“Designing the User Interface: Strategies for Effective Human-Computer Interaction (4th edition)”**, Pearson/Addison-Wesley, Boston