IDEA GROUP PUBLISHING
701 E. Chocolate Avenue, Hershey PA 17033-1117, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com
#ITB4336

Human Factors in Interface Design: An Analytical Survey and Perspective

Qiyang Chen and Vinai Sharma
Department of Information and Decision Sciences, Montclair State University, Upper Montclair, NJ 07043
chenq@mail.montclair.edu

ABSTRACT

This paper discusses the issues of human factors that affect interface design. It addresses the challenges that system analysts may face. It presents the strategies of incorporating human factor engineering into the process of system analysis and design. The user performance and their mental models are also discussed.

INTRODUCTION

Human factors are defined as knowledge of human abilities and limitations to the design of systems, organizations, jobs, machines, tools, and consumer products for safe, efficient and comfortable use [7]. In the United States, the *human factors engineering* was initially emphasized by the US military with concentration on human engineering and engineering psychology. Since then a great deal of research efforts have been focusing on the roles of users within a complex system [9]. Behavioral studies of programming, which emerged in the late 1970s, were among the earliest in the field of human-computer interaction (HCI) [12].

With its origin in experimental psychology and systems engineering, the study of human factors is defined as the study of human beings and their interaction with products, environments, and equipment in performing tasks and activities. The functions of human factors are to augment the performance of systems. The difference between studies in human factors of HCI and in other parallel sciences such as anthropology, cognitive science, psychology, sociology, and medicine is in the use of the knowledge of humans and their behavior in interaction design. The subject of human factors has become an exciting combination of basic and applied research for designing HCI.

Human factors are design oriented [6]. The design process includes comparing and designing systems, tasks, and environments to provide intellectual interaction to adapt the limitations of human beings. Alternatively, the user of the system can be trained or educated to work with a system. The latter approach faces unlimited challenges based on individual variations among human beings. Therefore, the systems are to be designed such that they are insightful and easy to use, calling for no special training or education.

This paper will discuss the issues of human factors that affect interface design. It addresses the challenges that system analysts may face. It emphasizes that human factor engineering must be incorporated into the process of system analysis and design. The user performance and their mental models are also discussed.

EMBEDING HUMAN FACTORS IN SYSTEM ANALYSIS AND DESIGN

Modern interactive systems are event driven. While system analysts may focus on the complex and open-ended nature of software design problems, they may overlook the following design categories that involve a great deal of human factors.

- Work environment (physical demands, skill demands, risk demands, time demands).
- Psychosocial environment (social and cultural style).
- Ergonomic environment (hardware design, anthropometrics and biomechanics)

Ergonomic considerations in the interface design include physical factors, biological factors, psychological factors, work factors, and organizational factors [11].

System analysis and design, is a formally structured, time-driven, interactive process with limitations of costs, resources, and organizational and environmental requirements, involving stages with distinct activities that vary as a function of system requirements and is categorized into planning, designing, testing, and evaluating. Created with the specified goals and objectives of transforming inputs into outputs, systems design starts and proceeds with branching and divergence from a higher to lower levels of activities and errands. Human factors engineering must be incorporated into this process.

Despite the historic differences between human factors and ergonomics in the type of knowledge used and in the goals for design, the two approaches are converging [12]. The common objectives of incorporating human factors in system analysis are

- Enhancing the efficiency of interaction and user productivity. The vital difference between people and machines is that people make mistakes and if we look positively at how people make mistakes – 'error analysis', we can increase the ease of use and reliability of performance. This can increase productivity and effectiveness of the system [10].
- Enhancing certain desirable human values at work by enhancing safety, increasing comfort level of using technology, decreasing stress.

A human-machine interaction takes place within an environment, which relates to physical environment, social, and/or organizational environment. The overall performance of a system depends on the result of requirement analysis, which is fundamental to human factors engineering. The *systems approach* connotes the whole system rather than its individually isolated parts or elements. This requires an analysis of the each specific system function in context of human computer interaction. Whereas the more popular *reductionism* approach, on the other hand, concentrates on a particular system component or element separately, and considers the technical components without paying much attention or regard to the human side of the system [5].

There are three levels to distinguish among tasks of human factors engineering performed in system design: (1) the design process – how the system is designed, (2) the design philosophy – the conceptual framework of the design, (3) the design architecture – the specification of the structure the system and the human-machine interface [8]. The interface design cannot be characterized as a stable process in terms of technology, environment, and work procedures and, therefore, new design approaches are essential. Users of the system are facing more challenging dynamic working conditions and have to adapt to these changes immediately. "No where are human factors as important as they are in user interface design. Just ask the typical systems analyst who

This paper appears in the book, *Managing Information Technology in a Global Economy*, the proceedings of the Inforrmation Resour es Management Association International Conference. Edited by Mehdi Khosrow-Pour.

spends half the day answering phone calls from system users who are having difficulty using the computer system.” [11]

The system users have been broadly classified as either dedicated or casual [13]. A dedicated user spends considerable time in various programs and is likely to become more comfortable and known to the system, whereas, the casual user makes use only of a particular program and that too occasionally and would never become friendly to the system. The latter would tend to have many problems and questions about the system and might become critical of the system refusing to accept it. Therefore, in an attempt to make the interface more user-friendly and easier, several general rules and guidelines should be followed [9].

Today’s computer systems are studded with multimedia interfaces that thrill a user at first place. Yet, the system may become irritating as soon as the user starts using it because he cannot find the way to get what he needs. An interface is *humane* if it is responsive to human needs and considerate of human frailties. A careful design and detailed specifications can enhance the interaction. This is particular important to the decision support systems, intelligent systems, and expert systems that are created to help decision-making at a higher level. Interfaces and their impacts on creativity and learning should be a major thrust to these systems [13]. However, no matter how much is done to improve interfaces, it becomes difficult to find a solution that solves a particular problem without creating new ones. And even if a separate solution for every problem were created, an interface of such complexity would be unusable [11].

In simpler words, the solutions to interface design are nothing more than compromises and finding a perfect solution would almost impossible. It’s like satisfying each and every user individually at a time. Not only that, how about the varying physical capabilities and intellectual levels of humans; the variability is so much so that the two individuals might respond very differently to the same medication or, even worse, the same human responds differently to the same medication at different times. With such variations and dynamism among humans, one cannot be sure of any system. Going further into this problem, The multitude of requirements for a successful interface design come from various sources, making it a multidisciplinary process that now has an added dimension of problems that are political or cultural in nature [13].

Obviously, then, a multidisciplinary team for the interface design is needed. But, then, the question arises, who and how, would resolve the conflicting interests and issues of all these specialists because every specialist has his/her own rigid thinking, priorities and perspectives, methods, and criteria for success. All solutions are shaped by a crowd of problems, which are faced by the designers and are imperceptible to the outsiders, and therefore designers’ efforts, even if worthwhile, are not appreciated.

The reasons why interface design is difficult include: (1) it is difficult to create optimal or good solutions, (2) there are so many competing interests and desires involved in interface process that any solution would not be more than a compromise and which compromise would be the best, and (3) the sources of requirements are many and therefore tend to be highly political.

HUMAN-CENTERED INTERACTION DESIGN

Human-centered development puts users’ needs first, technology second. It focuses on human activities. It makes technology invisible, embedded within activity-specific information appliances [5]. The skills built in the multidisciplinary design team should fully reflect *human-centered* concept. The people in the team should include field analysts, behavioral designers, model builders and rapid prototypers, user testers, graphical and industrial designers, and technical writers [7]. We need a framework for a system design that represents all aspects of work systems in a coordinated and compatible fashion. All well-designed interfaces, classified as command line, natural language, menu, form filling, or direct manipulation systems, must be in accordance with the user’s task needs, capabilities, and learning abilities. The general principles for designing a good interface include naturalness, consistency, non-redundancy, supportiveness, and flexibility. Some desired features for human-centered design process are discussed in [7].

Human factors, according to Clarke [8], can be centered in terms of three levels, namely *psychosocial*, *mental functions*, and *sensory-motor*. Stressed in this model is the need for an appropriate communication between the computer and the human elements that could be used for the support of design. At the first level, the human element of the dialogue represents the goal that the user is pursuing. Within the computer, virtual objects (things that a user can manipulate or know about) exist to support the fulfillment of the user goals. At the second level, the computer programs help the user accomplish goals. At the third level, effector and affector agents coordinate with the system and its physical components to produce the desired goal. Human diversity in the above context plays pivotal role and has always been intriguing and challenging system designers. The remarkable diversity of human abilities, backgrounds, motivations, personality, and work styles challenges interactive-system designers. Understanding the physical, intellectual, and personality differences among users is vital.

Referred to as “too-little-too late” phenomenon [6, 8], the ‘human factors’ issues are disregarded, and even if considered, are limited only to the evaluation stage, thereby, limiting and undermining the effectiveness and usefulness of human factors. Various reasons why human factors engineers are not considered as “equal partners” in a design team include:

- Design engineers recognize and discern human beings as very stretchy and flexible and think that humans can settle into the system requirements easily and therefore, if given more importance, could lead to technical compromise.
- Design engineers are not used to converting ‘human factors’ inputs into the systems language, it becomes a problem for them. In other words, the usefulness and usability of human factors inputs are limited in technical form and make not much sense to the design engineers. That calls for a better education of the two sides about their counterparts. The inputs are either so specific that they apply to a particular design situation and not to the design process in question or they are vague and overly general. In contrast, guidelines for designing intelligent interfaces need to be expressed at the cognitive task level, independent of a particular technology [11]. Thus one important task for human factors engineers is to ensure that design inputs are in a form that is useable and useful to designers.
- The major problem faced by human factor engineers is convincing the management and design engineers of the benefits over the cost, in terms of money or time, of integrating human factor knowledge and expertise into the systems design process. And even if they are able to do so, it is so late in the process that going back becomes feasibly impossible leading to very meager contributions. Hence, it becomes necessary in order for ‘human factors’ to be effective that a complete and continuous involvement of human factors engineers be encouraged and practiced throughout the design process.

A major breakthrough to make certain that design is human centered would be selling the prospective contributions of human factors to design engineers and managers, either by the use of case studies that highlight the advantages of human factors integration in to the design or by forwarding a cost-benefit analysis, which might be difficult due to the fact that segregating human factors contribution and then comparing it to other variables quantifiably is not easily feasible [2]. But there are methods of doing that kind of analysis, not objective though, and decision makers still have to make judgments, often intuitive.

Having established that the systems analysis and design is a problem-solving process, which involves the formulation of the problem, the generation of solutions to the problem, analysis of these alternatives, and selection of the most effective alternative, and also having established that human factors engineers need to be involved in all phases of the process, the question arises 'what would they bring in?' And, the answer is 'the application of behavioral principles, data, and methods to the design process.' "These activities include specifying inputs for job, equipment and interface design, human performance criteria, operator selection and training, and inputs regarding testing and evaluation."[4]. The major questions, then, are when, how, and how much of this integration would be optimum for the system to perform; how much of the resources be directed toward this *systems approach*, and who should or can justify these resources.

But the fact remains that the approaches to integrate human factors and expertise with software development are still tentative and sprouting [2]. The declining cost of computer systems, an ever growing resistance to the poor interfaces, and an intensifying need for product differentiation in the market underwrite that human factors will inevitably become pervasive all over systems [1]. Although, in the past, allocating resources to enhance user interface has always been considered lavishness [3]. Human factors are now starting to be widely recognized and accepted as a distinct discipline requiring integration with the process of systems analysis and design.

Approaches that have been advocated in order to manage human factors in search for the maximization of the influence of *systems approach* on the user interface development include [13]:

- Hiring human factors engineers or psychologists directly into development teams
- Concentrating human factors engineers in a support organization
- Making use of external consultants with user interface knowledge
- Placing a development group under the leadership of a human factors professional and
- Forming an educational center in which software engineers learn about human factors approaches.

A mix of perspectives of would be ideal and practical. That is to educate the system developers about human factors and educate human factor specialists about design and programming. This process would enlighten both groups with the importance of the two hand-in-hand fields as well as open them up for more justified evaluations and more flexible approaches toward this blazing field of systems development. At the same time, since educating might take longer than expected to bring things up, a wise and open-minded use of *objective outsiders* - the consultants, who can see through the things without any bias, would fill the gap easily, nicely, and presently. All other approaches are too extreme, and also too optimistic, in that an inevitable difference of opinions and interests would cross, no matter how powerful the human factors practitioners are.

USER PERFORMANCE AND THE MENTAL MODEL

Studies relevant to human factors also include evaluation of user performance and user mental models in interface design. The ideal HCI design should include the system knowledge about the users' cognitive characteristics, referred as user mental models. The user's mental model of an information system has a critical impact on the user's ability to use systems effectively. This concept has gained widespread acceptance in the field of HCI [1]. This acceptance is underlined by the "*mental model hypothesis.*" It suggests that faulty or incomplete representations (misconceptions) lead to errors or ineffectiveness, and that the types of errors users make can be understood once a model of their perceptions or knowledge about the system is derived [2]. However, verifying this concept empirically and applying it in practice to system design has led to different results [3,4]. Some studies have found that performance modeling benefits user learning of an appropriate model of a system or language [5]. Others have found little benefit or inconsistent results in certain situations [8].

Two fundamental issues that contribute to this state of affairs are addressed [8]. The first is that the target body of knowledge represented by a software system has rarely been articulated. Without the system model, what the user should know and therefore what the mental model should contain are unknown. Defining an adequate model of a system is very difficult due to the disagreement about what kind of knowledge it should encompass, and our limited understanding of the relationship between different kinds of knowledge and user performance. This issue has been addressed also by Stanley and Norcio [5]. The second fundamental problem is the difficulty of capturing the user's mental model, particularly in a way that can be dynamically and systematically compared with a system model.

Many techniques for deriving a performance model or mental model have been employed and discussed [5]. They are critical to any assessment of whether the user's mental model is an important determinant of performance. In addition, pragmatic cost-benefit characteristics must be taken into account. This determines if the information gained is worth the effort to acquire it.

Production system modeling, protocol analysis, and scaling techniques [11] can all be used in the study of mental models. Usually, the type of model to be derived depends on the types of knowledge being represented and the application environment to which the model will be utilized. Production system modeling or analytical approaches such as GOMS [1] are useful in evaluating the efficiency and consistency of design methods, but may be less useful in understanding the genesis of user errors, or in evaluating the difficulty for learners. In addition, it primarily focuses on decomposing tasks rather than the individual's cognitive difference [3]. Protocol techniques involve real-time "think-aloud," post task video confirmation, interviews, and inferring knowledge fragments and misconceptions from a session of user keystrokes [2]. These approaches are particularly appropriate for associating users' errors to misconceptions. However, they are hard to summarize and compare systematically. The scaling approach may have both advantages of the above two methods. But, the explanation of its numeric results is so critical that it is questionable whether it can yield valuable and reliable information to the design process.

In addition the psychological issues are often oversimplified in the study of performance models. Thomas [10] presents several typical oversimplifications in psychology that seem particularly prevalent in computer modeling:

- Ignoring motivation;
- Oversimplifying or ignoring individual differences;
- Ignoring social context;
- Oversimplifying or ignoring the human ability to learn and adapt.

A suggested performance model presents a framework for analyzing human cognitive behavior [12]. It classifies human behavior into three levels:

- *Skill-level*. This type of behavior is the result of long and intensive training. It permits rapid stimulus-response type operation.
- *Rule-level.* This level of behavior is done by a specific plan or procedure that does not require "intermediate" thinking.
- *Knowledge-level.* This is a true level of intelligent behavior. The expertise plays an important role. The synthesis, analysis, and inference are the primary activities in the performance of a knowledge-level task.

CONCLUSION

Although all technology-based systems are created with intent to serve some human purpose, this objective is difficult to achieve, thanks to the leadership of engineers who design such systems and who overemphasize technology in the design, paying scarce attention to the human components of the system. This paper presents the strategies for incorporating human factors into system analysis and design. Obviously, there are varieties of issues that need to be addressed in order to reach to the status as some researchers advocated, a *proactive computing* environment [6], which would take humans out of the 'loop' and move from *human-centered* to *human-supervised* (or even unsupervised) systems with a reduced human involvement, not only for users but also for programmers.

Furthermore, to provide a *universal usability*, the research agenda for HCI, according to Shneiderman [14], should take *technology variety*, *user diversity*, and *gaps in user knowledge* into account. This implies that an interface system must be able to adapt the computing needs of manifold users. Accommodating such a large user-base is a great challenge to HCI designers.

REFERENCES

1. Card, S. K., Moran, T. P. and Newell, A., *The Psychology of Human-Computer Interaction*, Lawrence Erlbaum Associates, 1983.
2. Diaper, D. (ed.) *Knowledge Elicitation: Principles, Techniques and Applications*, Ellis Hoewood Ltd. Publisher, England, 1989.
3. Chen, Q. and Norcio, A. F., "A neural network approach to User modeling," *Proceedings of 1991 IEEE International Conference on Systems, Man, and Cybernetics*, pp.1429-1434 (1991).
4. Macaulay, Linda. *Human-Computer Interaction for Software Designers, 1995.* ITP. UK.
5. Staggers, N. and Norcio, A. F., "Mental Model: Concepts for Human-Computer Interface Research," *International Journal of Man-Machine Studies*, 1993.
6. Raskin, Jeff. *The Humane Interface, New Directions for Designing Interactive Systems, 2000.* Addison Wesley Longman, Inc.U.S.A.
7. Salvendy, Gavriel. *Handbook of Human Factors and Ergonomics, 2nd Edn., 1999.* John Wiley & Sons, Inc., U.S.A.
8. Smith, Andy. Human Computer Factors: A Study, 1997. McGraw-Hill, UK.
9. Murray, D., "Modeling for Adaptivity," *Mental Models and Human-Computer Interaction 2*, M. J. Tuaber and D. Ackermann (eds.) , Elsevier Sciemnce pub , North-Holland, 1991, pp. 82-97.
10. Thomas, J. C. "Human Factors and Artificial Intelligence," *Advances in Human-Computer Interaction*, Vol. 2 Hartson, H. R. and Hix, D. (ed.), pp. 1-44, Ablex Pub. Co., 1988
11. Sage, A., "Knowledge, Skill and Information Requirement for System Design," in *System Design: Behavioral Perspectives on Designers, Tools, and Organizations,* W. Rouse and K, R. Boff (eds.), North-Holland, NY, 1987, pp. 285-303
12. Rosson, Mary Beth. "Human Factors in Programming and Software Development." ACM Computing Surveys, Vol. 28, No. 1, March 1996.
13. Summersgill, R., and Browne, D.P. "Human Factors: Its Place in System Development Methods." *Proceedings of the fifth international workshop on software specification and design,* 1989, pp. 227-234.
14. Shneiderman, Ben. "Universal Usability." *Communications of the ACM,* May 2000, Vol. 43, No. 5. pp.85-91.