Introduction & Overview to Human-Computer Interaction

Keith A. Butler
Boeing Computer Services
keith@atc.boeing.com

Robert J.K. Jacob
Tufts University
jacob@tufts.cs.edu

Bonnie E. John
Carnegie Mellon University
bej@cs.cmu.edu

ABSTRACT
The objective of this special introductory seminar is to provide newcomers to Human-Computer Interaction (HCI) with an introduction and overview of the field. The material will begin with a brief history of the field, followed by presentation and discussion of how good application development methods pull on the interdisciplinary technologies of HCI. The topics will include the psychology of human-computer interaction, psychologically-based design methods and tools, user interface media and tools, and introduction to user interface architecture.

KEYWORDS: human-computer interaction, usability engineering, human performance engineering, cognitive modeling, analysis methods, interaction styles, interaction hardware, user interface software, user interface management systems.

INTRODUCTION

The rapid growth of computing has made effective human-computer interaction essential. It is important for the growing number of computer users whose professional schedules will not allow the elaborate training and experience that was once necessary to take advantage of computing. Increased attention to usability is also driven by competitive pressures for greater productivity, the need to reduce frustration, and to reduce overhead costs such as user training. As computing affects more aspects of our lives the need for usable systems becomes even more important.

DESIGNING FOR HCI

Design in HCI is more complex than in many other fields of engineering. It is inherently interdisciplinary, drawing on and influencing diverse areas such as computer graphics, software engineering, human factors and psychology. Furthermore, the developer's task of making a complex system appear simple and sensible to the user is in itself a very difficult, complex task

The principles for applying human factors to machine interfaces became the topic of intense applied research during the 1940's, when equipment complexity began to exceed the limits of human ability for safe operation. However, the complexity of computing and of software development projects pose additional demands. An engineering paradigm that is common to many other fields can be generalized to a technical approach for engineering usability in computing systems [1] and is now in widespread use [2]. The paradigm follows an iterative cycle through analysis, design, implementation, and evaluation. Usability engineering structures human factors activity to work within software engineering projects.

Development of usable systems draws on technologies from user interface media, software architecture, process and data modeling, standards, and tools for modeling, building and testing user interfaces. Each can be a topic of research or application. These technologies will be covered in the following sections on the psychology of HCI and the computer science of HCI.

THE PSYCHOLOGY OF HCI

Information about human performance enters into the design of user interfaces from three sources: empirical data about human performance, theories of performance, and methods of observing and analyzing HCI systems. This portion of the overview will introduce each of these sources, the benefits of using them, and potential pitfalls to be avoided.

The empirical data of human performance can be classified as being either basic (collected in the service of basic psychology, usually in laboratory administered, knowledge-lean tasks) and applied (collected in the service of human factors design problems, often on realistic tasks). Further, these data can be classified as being about individual performance or group performance. A source book containing summaries of much useful data, and references to many more sources, called The Engineering Data Compendium [3], will3be referred to extensively for examples of such data and how it can be used in design.

A seminal source of both a useful theory of human performance and methods of analyzing HCI tasks is The Psychology of Human-Computer Interaction [4]. This session will introduce the GOMS method of analysis detailed in that work and several other cognitive- and system-modeling techniques that have arisen in the

CHI Companion 95, Denver, Colorado, USA
© 1995 ACM 0-89791-755-3/95/0005...$3.50

decade its publication. These techniques can be used to predict the usability and learnability of systems before they are built. We will discuss how to use these techniques in concert with rapid-prototyping and empirical testing for the benefit of an HCI design. We will present a directory of ongoing research in both psychological theory and analysis methods within the CHI community so that participants will know where to look for theories and methods applicable to their design problems.

THE COMPUTER SCIENCE OF HCI

As progress in HCI is making user interfaces easier to learn and use, they are becoming more difficult to build. Simple command line interfaces were difficult to use but easy to program. Modern direct manipulation and virtual environment interfaces are easier to understand and use, but harder to program, largely because they have more possible execution paths. The area of Computer Science in HCI studies and develops the abstractions, techniques, languages, and tools to address this problem. This section of the overview discusses research and practice in the field of user interface software [5].

An important concept in user interface software is to separate the design of an interactive system into distinct levels, i.e., the conceptual, semantic, syntactic, and lexical levels, and to develop a design for each level [6]. Another significant concept is the user interface management system (UIMS), which provides a separate software component that conducts all interactions with the user, distinct from the application program that performs the underlying task [7]. It is analogous to a database management system in that it separates a function used by many applications and moves it to a shared subsystem. This approach separates the problem of programming the user interface from each individual application and permits some of the effort of designing tools for human-computer interaction to be amortized over many applications and shared by them.

Since user testing is an important part of good interface design, techniques for rapidly prototyping and modifying user interfaces are needed. For this purpose, one needs methods for specifying user interfaces that are precise, so that the interface designer can describe and study a variety of possible user interfaces before building one, and that allow rapid, perhaps automatic production of prototypes for user testing.

In addition, we will consider interaction styles, from conventional command language and menu-based styles to newer direct manipulation and virtual environment styles and interaction techniques for using devices to perform tasks in an interface.

In a graphical direct manipulation style of user interface, a set of objects is presented on a screen, and the user has a repertoire of manipulations that can be performed on any of them. This means that the user has no command language to remember beyond the standard set of manipulations, few cognitive changes of mode, and a reminder of the available objects and their states shown continuously on the display. Examples are spreadsheets, the Xerox Star desktop and its descendants such as the Apple Macintosh, and, of course, many video games.

Recent work has carried the user's illusion of manipulating real objects still further. By coupling a the motion of the user's head to changes in the images presented on a head-mounted display, the illusion of being surrounded by a world of computer-generated images or a virtual environment is created. Hand-mounted sensors allow the user to interact with these images as if they were real objects located in space surrounding him or her.

REFERENCES

1. Butler, K. A. (1985) Connecting Theory and Practice: A Case Study of Achieving Usability Goals. In: *Proceedings of CHI'85 Human Factors in Computing Systems* (April 14-18, 1985,San Francisco, CA) ACM, pp. 85-88.

2. Wilkund, M. E. (1994) *Usability in Practice: How Companies Develop User-Friendly Products*, Cambridge, MA: Academic Press.

3. Boff, K. R. and Lincoln, J. E. (1988). *Engineering Data Compendium: Human Perception and Performance vols 1-3.* Harry G. Armstrong Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, Ohio.

4. Card, S.K., Moran, T.P., and Newell, A., (1983) *The Psychology of Human-Computer Interaction*, Hillsdale, NJ: Erlbaum.

5. Foley, J.D., van Dam, A., Feiner, S.K., and. Hughes, J.F. (1990) *Computer Graphics: Principles and Practice,* Reading, MA: Addison-Wesley.

6. Myers, B. A. (1989) "User-interface Tools: Introduction and Survey," *IEEE Software,* vol. 6(1) pp. 15-23.

7. Olsen, D.R. (1992) *User Interface Management Systems: Models and Algorithms,* San Mateo, CA: Morgan Kaufmann.