CSC 406 ABUAD (3)

Dynabook

Alan Kay designed the first object-oriented programming language in the 1970s. Called Smalltalk, the programs were the basis for what is now known as windows technology— the ability to open more than one program at a time on a personal computer. However, when he first developed the idea, personal computers were only a concept. In fact, the idea of personal computers and laptops also belongs to Kay. He envisioned the Dynabook—a notebook-sized computer, with a keyboard on the bottom and a high resolution screen at the top.

Star

- **Start this discussion from the handwritten note (2)**

The Xerox Star was born out of PARC's (Palo Alto Research Center) creative ferment, designing an integrated system that would bring PARC's new hardware and software ideas into a commercially viable product for use in office environments. The Star drew on the ideas that had been developed, and went further in integrating them and in designing for a class of users who were far less technically knowledgeable than the engineers who had been both the creators and the prime users of many PARC systems (one of PARC's favorite mottoes was "Build what you use, use what you build.") The Star designers were challenged to make the personal computer usable for a community that did not have previous computer experience.

The Star pioneered the now-familiar constellation of icons, moveable scrollable windows, and inter-mixed text and graphic images. The widely used graphic user interfaces (GUIs) of today are all variants of this original design.

The visible mechanisms on the Star display were backed up with a set of design principles that grew out of a user-oriented design methodology and by a great deal of empirical testing. Several principles were central to the Star design :

Direct manipulation

The core concept that distinguished Star (and other Alto programs) from the conventional computer interfaces of their time was the use of a bitmapped screen to present the user with direct visual representations of objects. In the Star's desktop metaphor, documents, printers, folders, collections of folders (file drawers and cabinets), in and out boxes, and other familiar office objects were depicted on the screen. To print a document, for example, the user could point (using the mouse) to the icon for the document and the icon for the printer, while using a key on the keyboard to indicate a Copy operation.

WYSIWYG (what you see is what you get)

In previously available programs for producing sophisticated graphical output—such as drawings or page layout with multiple fonts—the user created and edited a representation that looked like a programming language, and then compiled the resulting program into a visible form. Alto programs pioneered a new style that Star unified, in which the user works directly with the desired form, through direct manipulation. The user makes changes by operating on a direct representation of what will appear on the printed page. The Star user could intermix text, tables, graphs, drawings, and mathematical formulas. In fact, most of the popular microcomputer applications of today have not yet reached the degree of integration that Star offered more than a decade ago.

Consistency of commands

Because a single development group developed all Star applications in a unified way, it was possible to adhere to a coherent and consistent design language. The Star keyboard embodied a set of generic commands, which were used in a consistent way across all applications: Move, Copy, Delete, Open, Show Properties, and Same (copy properties). Evoking one of these

commands produced the same behavior whether the object is beingmoved or copied, for example, was a word of text, a drawing element, or a folder of documents. Through the use of property sheets the user could manipulate the aspects that were specific to each element, such as the font of a text character, or the brush width of a painted line. The Open command was the basis for applying a technique of progressive disclosure—showing the user only the relevant information for a task at hand, and then providing a way to reveal more possibilities, as they were needed.
In addition to these three key concepts, many specific design features made the Star unique, including its attention to the communicative aspects of graphic design, its integration of an end-user scripting language (CUSP), and its underlying mechanisms for internationalization—from the very beginning, Star versions were developed in several languages, including non-European languages with large character sets, non–left-to-right orthography, and so on. Some of the aspects that led to the Star's design quality may have also hampered its commercial success—in particular Xerox's dependence on development groups within a single company to produce all the applications software.

Lisa by Apple
The GUI (Graphical User Interface) that started it all. If you are sitting in front of a computer with a mouse and pull down menus you owe it to this machine. Windows proponents will tell you that Xerox PARC developed GUIs and Apple stole it from them. Xerox had the core idea but the video of the early PARC work showed it was advanced but it was not nearly what the Lisa (and later the Mac) became.

While giving credit to the workers at Xerox it should also be mentioned that much of the groundwork was done in the 1960s and early 1970s. One influential researcher was Licklider (1960), who visualized a symbiotic relationship between humans and computers. He envisaged computers that would be able to do more than simply handle information: the partnership of computer and human brain would greatly enhance thinking processes and lead to more creative achievements. Another influential development was the pioneering work of Sutherland (1963), who developed the Sketchpad system at MIT. The Sketchpad system introduced a number of powerful new ideas, including the ability to display, manipulate and copy pictures represented on the screen and the use of new input devices such as the light pen. Alongside developments in interactive graphic interface, interactive text processing systems were also evolving at a rapid rate. Following in the footsteps of line and display editors was the development of systems that allowed users to create and edit documents that were represented fully on the screen. The underlying philosophy of these systems is captured by the term WYSIWYG, which stands for ‘what you see is what you get’ (pronounced ‘whizzee-wig’). In other words, the documents were displayed on the screen exactly as they would look in printed form. This was in stark contrast to earlier document editors, where commands were embedded in the text and it was impossible to see what document would look like without printing it.

Discipline of Human Computer Interaction

Human-computer Interaction is the kind of discipline, which is neither the study of human, nor the study of technology, but rather the bridging between those two. So you always have to have one eye open to the question: what can the technology do? How can you build it? What are the possibilities? And one eye open to the question: what are people doing and how would this fit in? What they would do with it? If you lose sight of either of those, you fail to design well. And of course they require different ways of thinking. So I think the challenge is to keep knowledge of both the technology and the people playing off against each other in order to develop new things.

If you build something you need to consider not just 'I'm building something because I need to build it', but 'what effect is it going to have on the way people work and the way people live?'

Quality

Quality is conformance to specifications". So, according to this definition quality is the measure of degree to which the design specifications are followed during manufacturing. The greater the degree of conformance, the higher the level of quality is. Philip Crosby describes, "Quality is conformance to requirements." Here software requirements are the foundation from which quality is measured. Lack of conformance to requirements is lack of quality. With respect to HCI, quality is something beyond meeting the specifications, requirements or customer expectations. For example, consider a scenario, as you know, there is always a quality assurance department in any software house which checks the final products with reference to their specification or requirements. The products that do not fulfil their specifications or requirements they are considered bugged. The question is "what will be the matter if the specifications or requirements, which are being used to measure quality, are not complete?" That's why, Quality is beyond the conformance to specifications or requirements or even the customer expectations. Quality cannot be measured just by the requirements or specifications described by the customer rather you should approach to that end user who will use this product. The expectations or needs of the end user can be the measure of quality. So, we can say, as much as the product will be useable for end user as much higher will be its quality.

To understand the relationship of quality and usability in a software reference, look at the definition of software quality. "The extent to which a software product exhibits these characteristics"

- Functionality
- Reliability
- Usability
- Efficiency
- Maintainability
- Portability

Interdisciplinary nature of HCI

The main factors that should be taken in account in HCI design are described below. Primarily, these relate directly to users, such as comfort and health, or are concerned with users' work, the work environment or the technology being used. What makes the analysis even more complex is that many factors inevitably interact with each other. For example, if changes are made to improve productivity, this may have undesirable effects on users' motivations and levels of satisfaction because issues relating to job design and work organization are ignored.

Productivity Factors

Increase output, increase quality, decrease costs, decrease errors, decrease labour requirements, and decrease production time, Increase creative and innovative ideas leading to new products

System Functionality

Hardware, software, application

Task Factors

Easy, complex, novel, Task allocation, repetitive, Monitoring, skills, multi-media

User Interface

Input devices, output displays, dialogue structures, User of colour, icons, commands, graphics, natural language

Comfort Level

Seating Equipment layout
Cognitive processes and capabilities

The User

Motivation, Enjoyment, Satisfaction, Personality

Experience level

Health and Safety
Stress, headaches, Muscular-skeleton, disorders
Environmental Factors
Noise, heating, ventilation, lighting
Organizational Factors
Training, job design, politics, roles Work organization

HCI understands the Complex Relationship between Human and Computers, which are two distinct 'Species'. Successful Integration is dependent upon a better understanding of both Species. Hence HCI borrows and establishes its roots in Disciplines concerned with both.

Human
• Cognitive Psychology
• Social Organizational Psychology
• Ergonomics and Human Factors
• Linguistics
• Philosophy
• Sociology
• Anthropology
Machine
• Computer Science
• Artificial Intelligence
Other
• Engineering
• Design

Cognitive Psychology
Psychology is concerned primarily with understanding human behaviour and the mental processes that underlie it. To account for human behaviour, cognitive psychology has adopted the notion of information processing. Everything we see, feel, touch, taste, smell and do is couched in terms of information processing. The objective cognitive psychology has been to characterize these processes in terms of their capabilities and limitations.

Social and Organizational psychology
Social psychology is concerned with studying the nature and causes of human behaviour in a social context. Vaske and Grantham identify the four core concerns of social psychology as:
• The influence of one individual on another person's attitudes and behaviour
• The impact of a group on its members' attitude and behaviour
• The impact of a member on a group's activities and structure
• The relationship between the structure and activities of different groups.
The role of social and organizational psychology is to inform designers about social and organizational structures and about how the introduction of computers will influence working practices.

Ergonomics or human factor
Ergonomics, or human factor, developed from the interests of a number of different disciplines primarily during World War II. Its purpose is to define and design tools and various artefacts for different work, leisure and domestic environments to suit the capabilities and capacities of users. The role of ergonomist is to translate information from the above sciences into the context of

design, whether for a car seat or a computer system. The objective is to maximize an operator's safety, efficiency and reliability of performance, to make a task easier, and to increase feelings of comfort and satisfaction.

Linguistics

Linguistics is the scientific study of language (Lyons, 1970). From the point of view of HCI there are several issues that may be better understood by applying knowledge and theories from linguistics. For example, in the early days of command languages there was some debate about whether or not the object to which a command applied should come before or after the command itself. When deleting a file called 'xyz', for example, should you type delete 'xyz' or 'xyz' delete.

Philosophy, Sociology and Anthropology

A major concern of these disciplines until relatively recently has been to consider the implication of the introduction of IT to society. More recently, attempts are being made to apply methods developed in the social sciences to the design and evaluation of systems. The reason for applying social science methods of analysis to HCI, it is argued, are that a more accurate description of the interaction between users, their work, the technology that they use and the environment in which they are situated can be obtained. One application of social science methods has been to characterize computer supported cooperative writing (CSCW), which is concerned with sharing software and hardware among groups of people working together. The is to design tools and ways of working which optimize the shared technology so that maximum benefit can be obtained by all those who use or are affected by it.

Artificial Intelligence

Artificial Intelligence (AI) is concerned with the design of intelligent computer programs which simulate different aspects of intelligent human behaviour. The relationship of AI to HCI is mainly concerned with user's needs when interacting with an intelligent interface. These include, for example, the use of natural language and speech as a way of communicating with a system and the need for system to explain and justify its advice.

Computer Science

One of the main contributions of computer science to HCI is to provide knowledge about the capabilities of technology and ideas about how this potential can be harnessed. In addition, computer scientists have been concerned about developing various kinds of techniques to support software design, development and maintenance. In particular, there has been a strong interest in automating design and development when feasible.

Engineering and design

Engineering is an applied science, which relies heavily on model building and empirical testing. Design contributes creative skills and knowledge to this process. In many respects the greatest influence of engineering on HCI and subsequently on interface and system development is through software engineering. Design too is a well-established discipline in its own right, which has potential benefits when applied to HCI problems. An obvious example is graphic design.