Human Computer Interaction

**Usability of interfaces. Types of interfaces: command line interface, text interface, graphic interface.**

**Physical and mental characteristics of the user. Development stages of the user interface.**

**Types of testing of interfaces (testing of users).**

**Perspectives of development of interfaces.**

1. Понятие пользовательского интерфейса. Эволюция интерфейсов. Революция взаимодействия человека с компьютером.
2. Виды интерфейсов взаимодействия.
3. Проектирование интерфейсов. Юзабилити интерфейсов. Этапы разработки пользовательского интерфейса.

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.

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 more 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„ (Human Computer Interaction, Gustav Evertsson 2001)

”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„ (Human Computer Interaction, Gustav Evertsson 2001)

**Types of interfaces**

The first computers used command input (console) (CLI) to control. To use the programs required special training and a certain level of knowledge. Later, in 1973, the first computer using a graphical interface appeared in the Xerox laboratory, and even later, the idea was embodied in more successful commercial Apple and Windows systems. Since then, the concept of “interface” has been associated precisely with the graphical user interface (GUI).

The advent of computers with a graphical interface has revolutionized, and made it possible to use it, having a minimum level of training and knowledge. Moreover, the experience of interaction has become universal. Whatever program or site I open, in less than a minute, I can learn at least 80% of the ways I can interact with it. All this, thanks to the experience gained earlier.

NUI

A distinctive feature of NUI is the interpretation of experience gained in the real world. For example, from birth, we understand how to move objects in space, in the same way we can move objects in a virtual environment. NUI is an intuitive interface.

Many applications with a touch interface are much closer to the ideal NUI.

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. Human-Computer Interaction studies how people design, implement and use computer interfaces. HCI has become an umbrella 4 term for a number of disciplines including theories of education, psychology, collaboration as well as efficiency and ergonomics.

Figure 1.1: some of the disciplines involved in the field of Human Computer Interaction



Recent developments in the area of HCI have shown an interest in adaptive interfaces, speech recognition, gestures and the role of time. 5 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. 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.

**Types of interfaces:**

Typography and text

For many years, computer displays resembled paper documents. This does not mean that they were simplistic or unreasonably constrained. On the contrary, most aspects of modern industrial society have been successfully achieved using the representational conventions of paper, so those conventions seem to be powerful ones. Information on paper can be structured using tabulated columns, alignment, indentation and emphasis, borders and shading. All of those were incorporated into computer text displays. Interaction conventions, however, were restricted to operations of the typewriter rather than the pencil. Each character typed would appear at a specific location. Locations could be constrained, like filling boxes on a paper form. And shortcut command keys could be defined using onscreen labels or paper overlays. It is not text itself, but keyboard interaction with text that is limited and frustrating compared to what we can do with paper (Sellen & Harper 2002). But despite the constraints on keyboard interaction, most information on computer screens is still represented as text. Conventions of typography and graphic design help us to interpret that text as if it were on a page, and human readers benefit from many centuries of refinement in text document design. Text itself, including many writing systems as well as specialised notations such as algebra, is a visual representation that has its own research and educational literature. Documents that contain a mix of bordered or coloured regions containing pictures, text and diagrammatic elements can be interpreted according to the conventions of magazine design, poster advertising, form design, textbooks and encyclopaedias. Designers of screen representations should take care to properly apply the specialist knowledge of those graphic and typographic professions. Position on the page, use of typographic grids, and genre-specific illustrative conventions should all be taken into account.

Summary: most screen-based information is interpreted according to textual and typographic conventions, in which graphical elements are arranged within a visual grid, occasionally divided or contained with ruled and coloured borders.

**Maps and graphs**

The computer has, however, also acquired a specialised visual vocabulary and conventions. Before the text-based ‘glass teletype’ became ubiquitous, cathode ray tube displays were already used to display oscilloscope waves and radar echoes. Both could be easily interpreted because of their correspondence to existing paper conventions. An oscilloscope uses a horizontal time axis to trace variation of a quantity over time, as pioneered by William Playfair in his 1786 charts of the British economy. A radar screen shows direction and distance of objects from a central reference point, just as the Hereford Mappa Mundi of 1300 organised places according to their approximate direction and distance from Jerusalem. Many visual displays on computers continue to use these ancient but powerful inventions – the map and the graph. In particular, the first truly large software project, the SAGE air defense system, set out to present data in the form of an augmented radar screen – an abstract map, on which symbols and text could be overlaid. The first graphics computer, the Lincoln Laboratory Whirlwind, was created to show maps, not text.

Summary: basic diagrammatic conventions rely on quantitative correspondence between a direction on the surface and a continuous quantity such as time or distance. These should follow established conventions of maps and graphs.

**Mental models – what the user infers about the system**

Don Norman, one of the first generation of cognitive scientists investigating HCI, also wrote the first popular book on the topic – The Design of Everyday Things1 . What most people remember about this book is the example of door handles that are so badly designed they need labels telling you to pull them. But his key message was to draw attention to the gulf of evaluation and the gulf of execution– how does the user know what the system is doing, and how do they know what they need to do, in order to achieve their goals?

Computer systems are so complex, that nobody really knows what is happening inside (except, possibly, the designer). In the face of incomplete information, the gulf of evaluation is unavoidable. The user has to make inferences (or guess) what is happening inside. The user’s conclusions form a mental model of the system. One way of thinking about the design problem is that the designer must give sufficient clues to the user to support that inference process, and help the user form an accurate (or at least adequate) mental model. The idea of a visual metaphor is that the screen display simulates some more familiar real world object, and that the user’s mental model will then be understood by analogy to the real world. The metaphor/analogy approach can potentially help with the gulf of execution too. If the system behaved exactly like the real world objects depicted, then users would know exactly what to do with them. In practice, computer systems never behave exactly like real world objects, and the differences can make the system even more confusing. (Why do you have windows in your desktop? Why do I have to put my USB drive in the rubbish before unplugging it?) Furthermore, designers inadvertently create metaphors that correspond very well to their own understanding of the internal behaviour of the system, but users should not be expected to know as much as designers. User studies can help to identify what users actually know, what they need to know, and how they interpret prototype displays.

**Mental models research**

Mental models research attempts to describe the structure of the mental representations that people use for everyday reasoning and problem solving. Common mental models of everyday situations are often quite different from scientific descriptions of the same phenomena. They may be adequate for basic problem solving, but break down in unusual situations. For example, many people imagine electricity as being like a fluid flowing through the circuit. When electrical wiring was first installed in houses, it appeared very similar to gas or water reticulation, including valves to turn the flow on and off, and hoses to direct the flow into an appliance. Many people extended this analogy and believed that the electricity would leak out of the light sockets if they were left without a lightbulb. This mental model did not cause any serious problems - people simply made sure that there were lightbulbs in the sockets, and they had no trouble operating electrical devices on the basis of their model.

The psychological nature of unofficial but useful mental models was described in the 1970s, and these ideas have been widely applied to computer systems. Young's study of calculator users in 1981 found that users generally had some cover story which explained to their satisfaction what happened inside the device. Payne carried out a more recent study of ATM users, demonstrating that even though they have never been given explicit instruction about the operation of the ATM network, they do have a definite mental model of data flow through the network, as well as clear beliefs about information such as the location of their account details.

The basic claim of mental models theory is that if you know the users' beliefs about the system they are using, you can predict their behaviour. The users' mental models allow them to make inferences about the results of their actions by a process of mental simulation. The user imagines the effect of his or her actions before committing to a physical action on the device. This mental simulation process is used to predict the effect of an action in accordance with a mental model, and it supports planning of future actions through inference on the mental model. Where the model is incomplete, and the user encounters a situation that cannot be explained by the mental model, this inference will usually rely on analogy to other devices that the user already knows.

**Think aloud studies**

A great deal of cognitive psychology research, including some basic research on mental models, has been based on think-aloud studies, in which subjects are asked to carry out some task while talking as continuously as possible. The data are collected in the form of a verbal protocol, normally transcribed from a tape recording so that subtle points are not missed. Use of this technique requires some are. It can be difficult to get subjects to think aloud, and some methods of doing so can bias the experimental data. A detailed discussion of this kind of study is provided by Ericsson & Simon (1985).

**Performance models of users**

Early HCI research was largely concerned with the performance of the user, measured in engineering terms as a system component (‘cognitive psychology’ is closely associated with ‘artificial intelligence’, investigating human performance by simulating it with machines). One of the most famous findings in cognitive psychology research, and the one most often known to user interface developers, is an observation by George Miller in 1956. Miller generalised from a number of studies finding that people can recall somewhere between 5 and 9 things at one time - usually referred to as “seven plus or minus two”. Surprisingly, this number always seems to be about the same, regardless of what the “things” are. It applies to individual digits and letters, meaning that it would be very difficult to remember 25 letters. However if the letters are arranged into five 5-letter words (apple, grape …), we have no trouble remembering them. We can even remember 5 simple sentences reasonably easily. Miller called these units of short-term memory chunks. It is rather more difficult to define a chunk than to make the observation - but it clearly has something to do with how we can interpret the information. This is often relevant in user interfaces - a user may be able to remember a sequence of seven meaningful operations, but will be unable to remember them if they seem to be arbitrary combinations of smaller elements.

Short term memory is also very different from long term memory - everything we know. Learning is the process of encoding information from short term memory into long term memory, where it appears to be stored by association with the other things we already know. Current models of long-term memory are largely based on connectionist theories - we recall things as a result of activation from related nodes in a network. According to this model, we can improve learning and retrieval by providing rich associations - many related connections. This is exploited in user interfaces that mimic either real world situations or other familiar applications.

A further subtlety of human memory is that the information stored is not always verbal. Short term memory experiments involving recall of lists failed to investigate the way that we remember visual scenes. Visual working memory is in fact independent of verbal short term memory, and this can be exploited in mnemonic techniques which associate images with items to be remembered – display icons combined with associated labels provide this kind of dual coding.

**Intelligent interfaces – what the system infers about the user**

A further inference problem is that, in addition to the user not knowing what is happening inside the system, the system doesn’t ‘know’ what is happening inside the user. Advanced systems can be designed to record and observe user interactions, and on the basis of that data, make inferences about what the user intends to do next, and present short-cuts, usability cues or other aids. These kinds of ‘intelligent user interface’ are becoming more common, but they can also introduce severe usability problems. A notorious early example was the Microsoft Word ‘Clippy’, which analysed features of the document, and offered to help with automatic formatting (“You appear to be writing a letter …”). Although some users found it useful, a far larger number found the tone patronizing and the automated actions inaccurate. Google ‘Death to Clippy’ to see the extent to which smart user interface technology can get it wrong.

Many intelligent user interfaces emerge from the machine learning community, and especially Bayesian inference techniques. Bayesian techniques are more appropriate to user interfaces than other techniques for a range of reasons:

* They don’t rely on large training sets (as is the case with neural net approaches), so they can adapt more quickly to individual users
* Bayesian consideration of prior probabilities corresponds better to commonsense human reasoning under uncertainty.
* Bayes formula provides a consistent way to combine data from user interactions with historical data and heuristic rules.

An inference framework provides a valuable analytic perspective on many current trends in user interaction. For example, the behaviour of Google, or of recommender systems such as Amazon or Facebook friend finder, use inference techniques to apply statistical data and guess what the user really wants. It remains the case that when the system makes inaccurate inferences, the results will be annoying, confusing, or even damaging. This means that some advanced research areas, such as Programming by Example (where automated scripts or macros are created by inference, after observing repeated actions) provide a major challenge for HCI. These are active areas of research in Cambridge at present, and a few advanced prototypes are available for experimental use, such as the Koala project at IBM's Almaden Research Center (Allen Cypher, one of the Koala team, has worked in this area for many years – his ‘Eager’ prototype at Apple Research was an early success).

**Interface Design Usability of interfaces. UI Development stages.**

Design

* Functional requirements: definition of the development goal and initial requirements.
* User analysis: identifying user needs, developing scenarios, assessing the compliance of scenarios with user expectations.
* Conceptual design: modeling the process for which the application is being developed.
* Logical design: the definition of information flows in the application.
* Physical design: choosing the platform on which the project will be implemented and development tools.

Implementation

* Prototyping: development of paper and / or interactive layouts of screen forms.
* Design: creating an application, taking into account the possibility of changing its design.

Testing

* Usability testing: testing the application by various users, including and users with disabilities (Accessibility testing).

Development stages

The full interface development cycle includes the following steps:

* Research
* User scripts
* Interface structure
* Interface prototyping
* Style Definition
* Design concept
* The design of all screens
* Interface animation
* Preparation of materials for developers