What is HCI / Interaction Design?

With the exception of some embedded software and operating system code, the success of a software product is determined by the humans who use the product. A ***usercentred design process***, as taught in earlier years of the tripos and experienced in many group design projects, provides a professional resource to creating software with functionality that users need. However, the availability of technical functionality does not guarantee that software will be practically usable. Software that is usable for its purpose is sometimes described by programmers as "intuitive" (easy to learn, easy to remember, easy to apply to new problems) or "powerful" (efficient, effective). These terms are vague and unscientific, but they point in the right direction. This course presents scientific approaches to making software that is "intuitive" and "powerful".

HCI helps us to understand why some software products are good and other software is bad. But sadly it is not a guaranteed formula for creating a successful product. In this sense it is like architecture or product design. Architects and product designers need a thorough technical grasp of the materials they work with, but the success of their work depends on the creative application of this technical knowledge. This creativity is a craft skill that is normally learned by working with a master designer in a studio, or from case studies of successful designs. A computer science course does not provide sufficient time for this kind of training in creative design, but it can provide the essential elements: an understanding of the user's needs, and an understanding of potential solutions.

In professional work, the most important attributes for HCI experts are to be both creative and practical, placing design at the centre of the field.

Visual representation

How can you design computer displays that are as meaningful as possible to human viewers? Answering this question requires understanding of visual representation – the principles by which markings on a surface are made and interpreted.

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 on screen 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

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.

Schematic drawings

Ivan Sutherland introduced several more sophisticated alternatives (1963). The use of a light pen allowed

users to draw arbitrary lines, rather than relying on control keys to select predefined options. An obvious application, in the engineering context of MIT, was to make engineering drawings such as a girder bridge. Lines on the screen are scaled versions of the actual girders, and text information can be overlaid to give details of force calculations. Plans of this kind, as a visual representation, are closely related to maps. However, where he plane of a map corresponds to a continuous surface, engineering drawings need not be continuous. Each set of connected components must share the same scale, but white space indicates an interpretive break, so that independent representations can potentially share the same divided surface – a convention introduced in Diderot's encyclopedia of 1772, which showed pictures of multiple objects on a page, but cut them loose from any shared pictorial context.

Summary: engineering drawing conventions allow schematic views of connected components to be shown in relative scale, and with text annotations labelling the parts.

White space in the representation plane can be used to help the reader distinguish elements from each other rather than directly representing physical space.

Pictures

Sutherland also suggested the potential value that computer screens might offer as artistic tools. His Sketchpad system was used to create a simple animated cartoon of a winking

girl. This is the first computer visual representation that might suffer from the ‘resemblance fallacy’, i.e. that drawings are able to depict real object or scenes because the visual perception of the flat image simulates the visual perception of the real scene. Sutherland’s cartoon could only be called an approximate simulation, but many flat images (photographs, photorealistic ray-traced renderings, ‘old master’ oil paintings) have been described as though perceiving the representation is equivalent to perceiving a real object.

In reality, new perspective rendering conventions are invented and esteemed for their accuracy by critical consensus, and only more slowly adopted by untrained readers. The consensus on preferred perspective shifts across cultures and historical periods, as is obvious from comparison of prehistoric, classical, medieval and renaissance artworks. It would be naïve to assume that the conventions of today are the final and perfect product of technical evolution. As with text, we become so accustomed to interpreting these representations that we are blind to the artifice. When even psychological object recognition experiments employ line drawings as though they were objects, it can be hard to insist on the true nature of the representation. But professional artists are fully aware of the conventions they use – the way that a photograph is framed changes its meaning, and a skilled pencil drawing is completely unlike visual edge-detection thresholds. A good pictorial representation need not simulate visual experience any more than a good painting of a unicorn need resemble an actual unicorn.

Summary: pictorial representations, including line drawings, paintings, perspective renderings and photographs rely on shared interpretive conventions for their meaning. It is naïve to treat screen representations as though they were simulations of experience in the physical world.

Node-and-link diagrams

The first impulse of a computer scientist, when given a pencil, seems to be to draw boxes and connect them with lines. These node and link diagrams can be analysed in terms of the graph structures that are fundamental to the study of algorithms (but unrelated to the visual

representations known as graphs or charts). A predecessor of these connectivity diagrams can be found in electrical circuit schematics, where the exact location of components, and the lengths of the wires, can be arranged anywhere, because they are irrelevant to the circuit function. Another early program created for the TX-2, this time by Ivan Sutherland's brother Bert, allowed users to create circuit diagrams of this kind. The distinctive feature of a node-and-link connectivity diagram is that, since the position of each node is irrelevant to the operation of the circuit, it can be used to carry other information. Marian Petre's research into the work of electronics engineers (1995) catalogued the ways in which they positioned components in ways that were meaningful to human readers, but not to the computer – like the blank space between Diderot's objects a form of 'secondary notation' – use of the plane to assist the reader in ways not related to the technical content.

Circuit connectivity diagrams have been most widely popularised through the London Underground diagram, an invention of electrical engineer Henry Beck. The diagram has been clarified by exploiting the fact that most underground travellers are only interested in order and connectivity, not location, of the stations on the line. However, popular resistance to reading 'diagrams' means that this one is more often described as the London Undergound 'map', despite Beck's complaints.

Summary: node and link diagrams are still widely perceived as being too technical for broad acceptance. Nevertheless, they can present information about ordering and relationships clearly, especially if consideration is given to the value of allowing human users to specify positions.

Icons and symbols

Maps frequently use symbols to indicate specific kinds of landmark. Sometimes these are recognisably pictorial (the standard symbols for tree and church), but others are fairly arbitrary conventions (the symbol for a railway station). As the resolution of computer displays increased in the 1970s, a greater variety of symbols could be differentiated, by making them more detailed, as in the MIT SDMS system that mapped a naval battle

scenario with symbols for different kinds of ship. However, the dividing line between pictures and symbols is ambiguous. Children's drawings of houses often use conventional symbols (door, four windows, triangle roof and chimney) whether or not their own house has two storeys, or a fireplace. Letters of the Latin alphabet are shapes with completely arbitrary relationship to their phonetic meaning, but the Korean phonetic alphabet is easier to learn because the forms mimic the shape of the mouth when pronouncing those sounds.

The field of semiotics offers sophisticated ways of analysing the basis on which marks correspond to meanings. In most cases, the best approach for an interaction designer is simply to adopt familiar conventions. When these do not exist, the design task is more challenging.

It is unclear which of the designers working on the Xerox Star coined the term 'icon' for the small pictures symbolising different kinds of system object. David Canfield Smith winningly described them as being like religious icons, which he said were pictures standing for (abstract) spiritual concepts. But 'icon' is also used as a technical term in semiotics. Unfortunately, few of the Xerox team had a sophisticated understanding of semiotics. It was fine art PhD Susan Kare's design work on the Apple Macintosh that established a visual vocabulary which has informed the genre ever since. Some general advice principles are offered by authors such as Horton (1994), but the successful design of icons is still sporadic. Many software publishers simply opt for a memorable brand logo, while others seriously misjudge the kinds of correspondence that are appropriate (my favourite blooper was a software engineering tool in which a pile of coins was used to access the 'change' command).

It has been suggested that icons, being pictorial, are easier to understand than text, and that pre-literate children, or speakers of different languages, might thereby be able to use computers without being able to read. In practice, most icons simply add decoration to text labels, and those that are intended to be self-explanatory must be supported with textual tooltips. The early Macintosh icons, despite their elegance, were surprisingly open to misinterpretation. One PhD graduate of my acquaintance believed that the Macintosh folder symbol was a briefcase (the folder tag looked like a handle), which allowed her to

carry her files from place to place when placed inside it. Although mistaken, this belief never caused her any trouble – any correspondence can work, so long as it is applied consistently.

Summary: the design of simple and memorable visual symbols is a sophisticated graphic design skill. Following established conventions is the easiest option, but new symbols must be designed with an awareness of what sort of correspondence is intended - pictorial, symbolic, metonymic (e.g. a key to represent locking), bizarrely mnemonic, but probably not monolingual puns.

Visual metaphor

The ambitious graphic designs of the Xerox Star/Alto and Apple Lisa/Macintosh were the first mass-market visual interfaces. They were marketed to office professionals, making the 'cover story' that they resembled an office desktop a convenient explanatory device. Of course, as was frequently noted at the time, these interfaces behaved nothing like a real desktop. The mnemonic symbol for file deletion (a wastebasket) was ridiculous if interpreted as an object placed on a desk. And nobody could explain why the desk had windows in it (the name was derived from the 'clipping window' of the graphics architecture used to implement them – it was at some later point that they began to be explained as resembling sheets of paper on a desk). There were immediate complaints from luminaries such as Alan Kay and Ted Nelson that strict analogical correspondence to physical objects would become obstructive rather than instructive. Nevertheless, for many years the marketing story behind the desktop metaphor was taken seriously, despite the fact that all attempts to improve the Macintosh design with more elaborate visual analogies, as in General Magic and Microsoft Bob, subsequently failed.

The 'desktop' can be far more profitably analysed (and extended) by understanding the representational conventions that it uses. The size and position of icons and windows on the desktop has no meaning, they are not connected, and there is no visual perspective, so it is neither a map, graph nor picture. The real value is the extent to which it allows secondary notation, with the user creating her own meaning by arranging items as she wishes.

Window borders separate areas of the screen into different pictorial, text or symbolic contexts as in the typographic page design of a textbook or magazine. Icons use a large variety of conventions to indicate symbolic correspondence to software operations and/or company brands, but they are only occasionally or incidentally organised into more complex semiotic structures.

Summary: theories of visual representation, rather than theories of visual metaphor, are the best approach to explaining the conventional Macintosh/Windows ‘desktop’. There is huge room for improvement.

Unified theories of visual representation

The analysis in this lecture has addressed the most important principles of visual representation for screen design, introduced with examples from the early history of graphical user interfaces. In most cases, these principles have been developed and elaborated within whole fields of study and professional skill – typography, cartography, engineering and architectural draughting, art criticism and semiotics. Improving on the current conventions requires serious skill and understanding. Nevertheless, interaction designers should be able, when necessary, to invent new visual representations.

One approach is to take a holistic perspective on visual language, information design, notations, or diagrams. Specialist research communities in these fields address many relevant factors from low-level visual perception to critique of visual culture. Across all of them, it can be necessary to ignore (or not be distracted by) technical and marketing claims, and to remember that all visual representations simply comprise marks on a surface that are intended to correspond to things understood by the reader. The two dimensions of the surface can be made to correspond to physical space (in a map), to dimensions of an object, to a pictorial perspective, or to continuous abstract scales (time or quantity). The surface can also be partitioned into regions that should be interpreted differently. Within any region, elements can be aligned, grouped, connected or contained in order to express their relationships. In each case, the correspondence between that arrangement, and the intended interpretation, must be understood by convention or

explained. Finally, any individual element might be assigned meaning according to many different semiotic principles of correspondence.

The following table summarises holistic views, as introduced above, drawing principally on the work of Bertin, Richards, MacEachren, Blackwell & Engelhardt and Engelhardt.

	Graphic Resources	Correspondence	Design Uses
Marks	Shape Orientation Size Texture Saturation Colour Line	Literal (visual imitation of physical features) Mapping (quantity, relative scale) Conventional (arbitrary)	Mark position, identify category (shape, texture colour) Indicate direction (orientation, line) Express magnitude (saturation, size, length) Simple symbols and colour codes
Symbols	Geometric elements Letter forms Logos and icons Picture elements Connective elements	Topological (linking) Depictive (pictorial conventions) Figurative (metonym, visual puns) Connotative (professional and cultural association) Acquired (specialist literacies)	Texts and symbolic calculi Diagram elements Branding Visual rhetoric Definition of regions
Regions	Alignment grids Borders and frames Area fills White space Gestalt integration	Containment Separation Framing (composition, photography) Layering	Identifying shared membership Segregating or nesting multiple surface conventions in panels Accommodating labels, captions or legends
Surfaces	The plane Material object on which marks are imposed (paper, stone) Mounting, orientation and display context Display medium	Literal (map) Euclidean (scale and angle) Metrical (quantitative axes) Juxtaposed or ordered (regions, catalogues) Image-schematic Embodied/situated	Typographic layouts Graphs and charts Relational diagrams Visual interfaces Secondary notations Signs and displays

As an example of how one might analyse (or working backwards, design) a complex visual representation, consider the case of musical scores. These consist of marks on a paper surface, bound into a multi-page book, that is placed on a stand at arms length in front of a performer. Each page is vertically divided into a number of regions, visually separated by white space and grid alignment cues. The regions are ordered, with that at the top of the page coming first. Each region contains two quantitative axes, with the horizontal axis representing time duration, and the vertical axis pitch. The vertical axis is segmented by lines to categorise pitch class. Symbols placed at a given x-y location

indicate a specific pitched sound to be initiated at a specific time. A conventional symbol set indicates the duration of the sound. None of the elements use any variation in colour, saturation or texture. A wide variety of text labels and annotation symbols are used to elaborate these basic elements. Music can be, and is, also expressed using many other visual representations (see e.g. Duignan 2010 for a survey of representations used in digital music processing).