

USER INTERFACE – STANDARDS AND RESEARCH TOOLS

Grzegorz Sowa

IT Institute, Academy of Management, Lodz, Poland
gsowa@swspiz.pl

Abstract

The article outlines a wide range of issues related to human computer interface. The key concept here is the interface, however it may be defined. After presenting in short interface history, the interface research tools are discussed, especially eye movement tracing equipment. Principles of interface design (screen layout, error prevention, interaction design) are discussed later in more detail. A short presentation of specific issues – mobile phones and advertising banners follows. Interface concepts in development finish the article.

Key words: user interface, tracing eye movement, interface design principles, adaptive interfaces, screen design

1 Introduction – the Concept of an Interface

The concept of an interface is best understood in the widest possible terms. Daniell [4] presents a good set of examples. Thus an interface can be: a watch, a door, a tin opener, a sink, a toothbrush, a speedometer, a CD player etc.

Today's computer interface: screens, windows, keyboards, mice, etc., could be something entirely different tomorrow. Focusing on computer user interfaces, it should be born in mind that an interface is in fact a means for people to communicate with technical devices. These aspects can provide inspiration, and are also the endpoint for solutions created for computers.

2 The Beginnings of Computer User Interfaces

2.1 Equipment

Without going back as far as „ancient Rome”, in this case the electric typewriter and line printer, it can be agreed that a computer interface worthy of note was developed when the following three innovations appeared on the graphic screen:

- direct manipulation of graphic objects,
- the mouse,
- windows.

The first experiments with manipulation were carried out in 1963 at MIT by Sutherland [14], and this technology matured at the beginning of the 1980's with the arrival on the market of the Apple Lisa and soon afterwards the Macintosh computers [12]. The key elements here were the ability to grab objects, move them and change their size. In 1968, also at MIT, icons were invented (the name itself was popularized later), and a dynamic menu with selected items. The idea of WYSIWYG (What You See Is What You Get) was developed at Xerox PARC in the 1970's.

The mouse was created at Stanford University in 1965, the aim was to find a cheap alternative to the light pen, which was already popular. This also matured in Apple computers. The first windows were shown as index cards („tiles”), the concept of overlapping windows only appeared in 1969 (Kay, [10]) and was brought out in the Smalltalk system. And once again the idea matured technically as part of Apple.

This short reminder illustrates that the real beginnings of the interface as we understand it today was the Apple computer, which brought together in one system ideas that had been evolving over the previous 20 years.

2.2 Applications

The functions of modern graphic interfaces were quickly applied to useful applications, whose popularity accounted for them taking over the market so rapidly. Three basic areas can be listed here:

- word processors,
- graphics programs,
- spreadsheets.

The first word processor to include the basic functions which we take for granted today (scrolling, wrap, search, replace, move, copy, delete), was developed in Stanford in 1962, and the first popular product on the market was TVEdit in 1965. The possibilities of the graphic screen and the selecting tools were quickly exploited in programs which we now call graphics programs. The basic concepts were presented by Sutherland [14] in 1963 in his Sketchpad program. At the same time the first CAD applications appeared.

The spreadsheet, an ingenious discovery from today's point of view, was developed in the form of the VisiCalc program for the Apple II computer in 1978 by Frankstone and Brickline, then students at MIT Harvard [12].

Alongside these basic types of application, ideas and products which added to their capabilities were developed. The surprisingly old idea of hypertext

(Bush, 1945, [3]) found a practical application in the 1960's, when the area of computer games and joysticks was intensively developed.

3 Interface Research Tools

3.1 Tracing Eye Movement

One of the most obvious techniques for evaluating the interface is tracing the user's eye movements. This has two main uses: analyzing the ways the interface is used and controlling it (retrospective and real time eye tracking). From a research point of view these concepts are related. The basic problem in this area is creating a psychological theory which related data on movements of the cornea with cognitive processes.

There have been studies of eye movements since the 19th century, but meaningful results have only been achieved since the beginning of the computer age. For a long time, however, information about eye movements could only be processed post factum, from recordings, and not in real time.

A user looking at the screen directs his eyes in turn to one point (fixation) – then the information is processed, and quickly moves them around the screen (saccade). The pattern of eye movements on the screen is called the scan path [1].

The typical results of this kind of research can be presented in the form of a map, reminiscent of physical levels. In this way objective data can be received about which elements of an internet page attract visual attention. By dividing the page into specific rectangles, we come up with areas of interest, AOI.

Nevertheless the dynamic character of the modern interface: various windows open adjacent at one another, moving text, expandable menus, animations on screen, icons which can be moved by the user and so on, complicate the evaluation of eye fixations considerably.

Russel [13] notes that this is only the beginning of such difficulties. Once we establish where the user's eyes are directed, we then need to decide what he is thinking about at that time – cognition. The top-down approach arises from a particular theory. It is supposed, for example, that a longer look at the control panel is caused by difficulties in remembering what it is used for. The bottom-up approach, on the other hand, has established through experiments that objects at the bottom of the screen are looked at for a longer time. In other words: one could investigate the idea that since a part of the screen attracts a disproportionate amount of users' attention, then which of its elements are they in fact looking at?

An interesting concept is increasing the resolution of the area of the screen which the user is looking at a given time [17]. This could be even more useful

in a virtual 3-dimensional environment, where the computing power of processors will be inadequate for a long time to come, and limits in the area of processing could be extremely significant.

3.2 Controlling with Eye Movements

The visual interface is of course mainly the stream of information passed from the computer to the user's brain. Passing this information in the other direction could also be considered, however; modern devices that track eye movements are easy enough to use and reliable enough to consider this kind of application.

The idea of controlling the cursor with eye movements instead of the mouse is an obvious one, but it has been found that eye movements are not as precise as hand movements, and the idea has not so far been successful [13]. The fundamental problem is the fact that the user looking at something does not mean that he or she wants to do something with it – it is not necessarily worth activating that function. Unlike the mouse, the eyes are always active, but a reaction is not always required. Additionally, it is not clear what the equivalent of the mouse buttons would be. Blinking would not be a good solution – we do this often and unconsciously in order to moisten the eyeball.

Moreover, the feeling of controlling something all the time is irritating. This problem has been named the Midas touch – whatever we look at, something happens. It would be good to distinguish a normal state of looking around from a control mode, but for now it is not clear how to accomplish this comfortably. A comparable mistake was made by the creators of the most recent (2007) Microsoft Office program. An excessive amount of keyboard shortcuts means that accidentally pressing various combinations of keys, which is hard to avoid, for example when writing Polish letters, results in a – frequently unwelcome – system reaction.

A sensible approach is proposed by Jacob and Karn [8] (p.18): *“A user does not say or mean much by a movement of the eyes — far less than by a keyboard command or mouse click. The computer ought to respond with correspondingly small, subtle responses. Rearranging the screen or opening a new window would typically be too strong a response. More appropriate actions might be highlighting an object for future action, showing amplifying information on a second screen, or merely downloading extra information in case it is requested.”*

The first uses of eye movement control were military, as is often the case – monitoring the cornea movements of pilots in order to control the display of data on the windscreen and weapon turrets. The applications for the disabled are obvious. Despite its slow speed, it is in fact sometimes the only possibility.

3.3 Keeping up with the User

When optimizing computing, we normally try to achieve the desired effect in the shortest possible time. Jacob, Deligiannis, Morrison [9] presented an interesting idea: we can reverse this ideas and try to achieve the greatest amount (for example the highest resolution) in a given time (for example during the approach of the next portion of information to be processed). This can be especially useful in virtual reality environments which demand a lot of processing activity: the demand for fast reactions is often more important than the quality of the picture.

4 Design Methods

Every large software company has its own methods of designing interfaces. One of the more popular approaches is formalizing the design stages, with an emphasis on the organizational surroundings. Thus IBM highlights the following elements in its design templates [18].

Analysis of user needs:

- Who is going to use this product?
- What should the specifications of the product be?
- What kind of surroundings is this product going to function in?
- What are the priorities – which of its functions will be used most often?

Evaluation of competition and mutual dependence:

- How are these tasks carried out today?
- What do users like and dislike in applications that are currently being used?

User-centered design:

- definition of business aims
- understanding the user
- evaluation of competition
- making use of the user's entire experience in the design

The completion of each stage of the design process is emphasized: advertising, ordering, purchase, packaging, maintenance, installation, administration, documentation, upgrade and support. Objectivity has a significant impact on the design process.

Wolpert [21] has suggested a sensible systematization here, distinguishing three aspects of the user interface:

1. The physical aspect: desk, chair, lighting, keyboard, mouse, screen, textbooks, documents, formulae.

2. The application aspect: Windows, Menu, dialogue boxes, buttons, colors, sound.
3. The practical aspect: Clients, products, orders, catalogues, addition, update, printing.

It is important to understand that objects in the user graphic interface do not necessarily have equivalents in object-oriented systems. User objects such as folders, documents, or the trash folder feature in the user environment but do not necessarily appear in the application code objects. The presentation layer is one thing and the data layer another – experience from database design is useful here too. Apart from this, many early user interfaces were created in non object-oriented environments.

5 Principles of Interface Design

5.1 Simplicity

One of the first and most obvious principles of design is simplicity. A closer examination demonstrates the main aspects involved by Waloszek [19]:

- simplified, reduced – the reduction aspect,
- transparent, learnable – the understandability aspect,
- effective, efficient – the ease of use aspect.

The principle of avoiding redundancy does not unfortunately have many followers in the Windows system. The rule is a large variety of methods to achieve the same aim.

5.2 Error Prevention

It is easier to prevent errors than to correct them. A few basic techniques:

- making sure the information entered is complete,
- monitoring numerical, date and currency fields,
- ensuring that only buttons which have a role at a particular stage are in use,
- repeated checking and explanation of irreversible actions e.g. deleting.

The reversibility of actions carried out is significant here. Research has proved [15] that people working in an „high risk” environment make the same amount of mistakes as those operating under appropriate security restrictions. They work much more slowly and carefully, however, in order to make less mistakes. The result of the lack of support for the popular undo command is the need to provide numerous dialogue boxes which ask: *„are you definitely 100% certain that you want to continue”*, which slows down the user’s work.

Incidentally, the ability to undo an action is far more difficult to provide from a technical point of view (just like the complexity of the equivalent transaction service mechanisms in databases).

5.3 Effectiveness

One of the oldest principles in interface design is the so-called Fitts' Law, a maxim which in fact dates back to the pre-computer era – 1954 [5]. The original wording is rather complex, but it can be summed up as: the time taken to indicate a component on the screen is a logarithmic function of its width and distance on the screen from the current cursor position.

The simplest conclusion to be drawn from the rule above is basic: large objects should be used for important functions – large buttons are „quicker” to use. But there are also more interesting observations: according to Fitts' Law, menus in Macintosh operating systems should drop down five times faster than the Windows menu (!). A single row of icons in the task bar, attached to the edge of the interface, is many times faster than a double row of icons separated from the edge of the display a unclickable single pixel separator [15].

One should aim to create effective work habits in users. The previously cited Tognazzini [15] gives a beautiful example: *“The Macintosh, for years, showed an icon of a trashcan of imminent danger of explosion if a single document was placed there in. Users quickly formed the habit of emptying the trashcan as soon as the first document hit. This not only turned a single-step operation into a two-step operation (drag to the trash, then empty the trash), it negated the entire power of the trashcan, namely, undo.”*

The effectiveness of a system increases the memory of what it was used for. Information about users can be stored in cookies on the user's computer. If work is being done online, and this kind of information is also stored on the server, users can resume work exactly where they left off, regardless of the physical location they were last working in.

The simple Fitts' law does not describe what happens in the menu when a user moves the cursor there. At this stage, he or she decides on the action to be taken. It has been found (among others by Hornof and Kieras [7]) that looking through the menu is a fairly complex combination of sequential, simultaneous and random searching, and there are also considerable differences between particular people. Additionally, the same person will use various strategies according to the situation.

Certain types of scroll-down/expanded menu - dropdown and popup – are problematic. The second kind, or so-called context menus, are expanded with the right mouse button, and are popular in the Windows system. The problem is that it is not clear when this kind of menu is available. The same problem is found with invisible structures: objects such as the right text border in

MS Word, which has many properties and parameters – as long as the user knows that they exist.

5.4 Interaction Design

One fundamental action is choosing particular elements or set from a collection of them on the screen.

The basic methods are:

- selecting pieces of text: words, paragraphs, etc,
- choosing parts of the tree structure, usually by scrolling down with the (+) sign, or scrolling up with the (-) sign,
- choosing elements freely situated in a window.

Additionally, standard class or procedures libraries set out two typical selection methods. Radio buttons (the equivalent of the XOR operator) can be used to choose one particular option from many, whereas so-called checkboxes are used for choosing any combination of objects.

Lists are an important element of the screen; here it is important to distinguish whether the list from which we are to choose one element fits entirely into a window (a single selection list) or whether it has to be scrolled down (a drop down list box), generally by clicking on an arrow.

Elements on the screen can be containers – that is they contain other elements hidden in a lower layer, or not. Waloszek [19] gives a clear example:

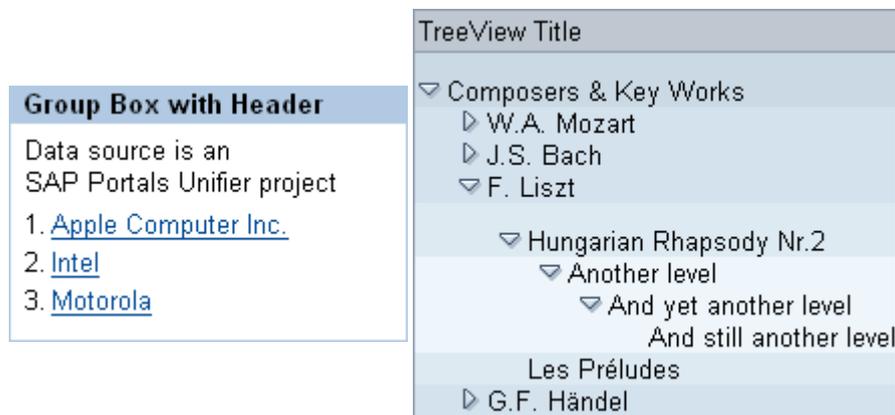


Figure 1. Left – an window with containers, Right – without

5.5 Comfort at Work

One of the fundamental principles of a good interface is maintaining contact with the user, among other things by reducing the sense of delay. The most important pieces of advice are:

- confirm clicking on all buttons visually or with a sound,
- provide an animated hourglass, to reassure the use that the system is working (has not crashed),
- display information on the estimated time that longer processes will take,
- provide information about processes in progress,
- inform the user through sounds about completed processes, so that they can carry out other tasks outside the application and come back to it at a convenient time,
- when functions are unavailable, this should be signaled with a beep rather than an error message,
- prevent multiple pressing of buttons. Since web page interfaces have a delayed reaction, users often click many times in order to force a reaction, which causes delays [15].

6 Screen Design

6.1 Distribution of Elements

Three key concepts feature here: the sequence of the elements, nesting and their distribution on screen. The classic principles of screen design are:

- logical sequence of fields: from left to right and from top to bottom,
- distinguishing active buttons from inactive buttons,
- if there are several buttons on the screen, this means that they can be pressed in any order.

Simplicity demands that unnecessary functions are hidden. There is generally no information at all on the screen about, for example, the right text margin. There are advantages to this – the screen is not overcrowded – but the obvious disadvantage is that information about these functions (e.g. the possibility of moving the margin) is available only to experienced users. Another method to cope with complexity are different views of the same problem: table, drafting and map for example.

6.2 Progressive Disclosure and Adaptive Interfaces

The principle of progressive disclosure was formulated in the design norms of Apple Computers [2] and was applied in the design of dialogue windows. It is assumed here that we make available to the user only the information, options or fields that are essential for him in a given context; the advanced options are hidden. This of course necessitates a thorough knowledge of typical use needs and errors are likely here.

One of the oldest progressive disclosure methods (rarely thought of in this context) is scrolling. A popular technique in contemporary interfaces is replacing full-size pictures with thumbnails. Homquist's [6] "*focus and context idea*" is particularly interesting. An analogy to Tong and Fisher's research [17], mentioned previously, is obvious here.

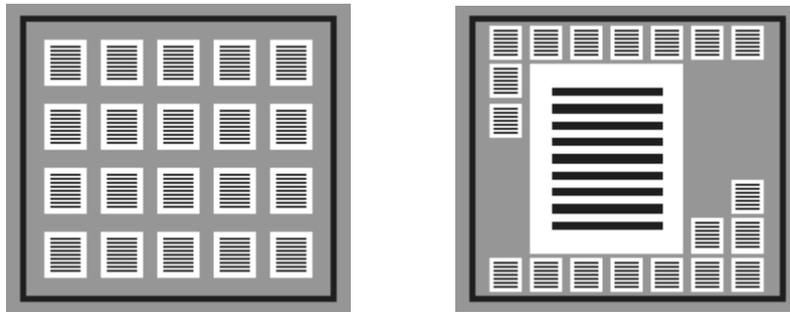


Figure 2. The principles of flip zooming – unzoomed view (Left) and zoomed view (Right) [6]

The most popular version of adaptive interfaces is currently the expandable menu in contemporary Microsoft applications. At first only selected, key options appear, and the full menu appears only after moving the arrow down; the interface remembers the user's preferences and in subsequent uses, the options most frequently chosen by him or her appear first.

The aim of both techniques, adaptive systems and progressive disclosure, is the same: minimizing demands on the user's memory and speeding up and maximizing its effectiveness. With progressive disclosure, the designer must decide which element is important at which time, and can easily make a mistake. It might seem that adaptive systems are considerably better: it is the user who decides, often unconsciously, how the system should behave, and the system adapts to the user's style of work. The problem in practice is that these systems are unstable and are constantly changing, which is not convenient.

6.3 Navigation: Where are We, How did We get Here and Where can We go from Here?

Human beings are capable of concentrating on one action at a time, and complex operations must be carried out one after another. On each screen, then, the way back, forwards and the current position should be indicated. A few obvious conventions have been developed here. The access path shows us our place in the catalogue hierarchy. Arrows indicate the previous and next screen.

The structure of contemporary applications is often so complex that it is not possible to define a simple series of screens which have to be reached in order to carry out a function. Special navigation techniques have to be defined. In integrated systems these tools are, for example, business process reference models.

7 Specific Issues – Mobile Phones and Advertising Banners

7.1 Interaction on a Small Screen

It is beneficial to imagine the difficulty of this issue in many practical applications. When researching the typical ways of using a personal computer with a popular interface of the WIMP (windows, icons, menus, pointer) type, it could be expected that the user would limit his or her head movements to an area of a few square decimeters – these are the conditions that research instruments usually work in. Problems begin, however, if we want to investigate interactions with a mobile phone, pocket computer or a periphery tool.

7.2 Banner Blindness

In a famous experiment Panero and Lane [11] demonstrated that Fitts' Law cannot be applied directly to the evaluation of the correct design of internet pages. It appears that users very often overlook the most visible information banners, even if they really are relevant to them. From among the several attempts to explain this phenomenon, the closest to the truth appears to be that the most glaringly highlighted pieces of information are immediately treated as unnecessary spam and automatically ignored. In any case, this is typical when users are looking for some particular information on a page. In other words, if we are excessive in highlighting the importance of a piece of information on a page, we may achieve the opposite effect to what we desire – it will instead be ignored. It is hard to talk about disclosure here – there is an old saying that the best way to hide information is to paste it on a door at eye level.

A practical design tip advises us to make this kind of information a slightly differentiated (e.g. through color) feature of the normal menu. It is worth noting that the same situation exists with television advertisements, during which viewers often change channel or turn off the sound.

8 Conclusion – Concepts in Development

From among the many predictions about the development of the interface, the most likely seems to be the concept of the ubiquitous computer, formulated as far back as 1991 by Wiser [20]. Objects in our surroundings such as notebooks, mobile phones, televisions, even air conditioning and lighting – the list is endless – all have interfaces. Their functions overlap and the fundamental issue has begun to be reducing the demands made on the user (the context-aware interface). This „computer” is supposed to predict what we want to do and adapt the interface to the situation the user is in. This can be seen in office tools that are currently used, but these are only modest beginnings. It can be predicted that the interface will pull away from these tools and become an independent go-between layer – a new class of operating system.

References

1. Aaltonen A., Hyrskykari A., Rähkä K., 1998, *101 Spots, or how do users read menus?* In: *Proceedings of CHI 98 Human Factors in Computing Systems*, ACM Press, pp. 132–139.
www.cs.uta.fi/~ah/CV-Hyrskykari.pdf
2. *Apple Human Interface Guidelines*,
http://developer.apple.com/documentation/userexperience/Conceptual/AppleHIGuidelines/XHIGIntro/chapter_1_section_1.html
3. Bush V., 1945, *As We May Think*, *The Atlantic Monthly*, 176 (July), pp. 101-108.
Reprinted and discussed In: *ACM Interactions*, 1996, 3, 2, pp. 35-67.
www.theatlantic.com/doc/194507/bush
4. Daniell M.J., *Bad Human Factors Designs*,
<http://www.baddesigns.com/index.shtml>
5. Fitts P.M., 1954. *The information capacity of the human motor system in controlling the amplitude of movement*, *Journal of Experimental Psychology*, 47, 6 (June), pp. 381-391.
Reprinted in *Journal of Experimental Psychology: General*, 1992, 121, 3, pp. 262–269,
http://eidetic.ai.ru.nl/egon/education/IntroCE/literature/classics/Fitts54 Information_capacity_motor_system.pdf

6. Holmquist L.E., 1997, *Focus+context visualization with flip zooming and the zoom browser*, Conference on Human Factors in Computing Systems, Atlanta. portal.acm.org/citation.cfm?id=1120383
7. Hornof A.J., Kieras D.E., 1997, *Cognitive modeling reveals menu search is both random and systematic*, Proceedings of CHI'97, Atlanta, March 1997, ACM Press, pp. 107-114.
8. Jakob R.J. K., Karn K.S., *Eye Tracking in Human-Computer Interaction and Usability Research*. www.alexpoole.info/academic/Poole&Ball%20EyeTracking.pdf
9. Jacob R.J.K., Deligiannis L.S. Morrison R., 1999, *A Software Model and Specification Language for Non-Wimp User Interfaces*, ACM Trans. On Computer-Human Interaction, 6, 1, pp. 1-46. <http://realeyes.ee/research/EyetrackingPromises.pdf>
10. Kay A., 1969, *The Reactive Engine*, PhD Thesis, Electrical Engineering and Computer Science University of Utah. www.usabilityviews.com/uv004819.html
11. Panero J., Lane D.M., 1998, *Banner Blindness: Web Searchers Often Miss "Obvious" Links*, Internetworking Dec. 1998. http://www.internettg.org/newsletter/dec98/banner_blindness.html
12. Myers B.A., 1998, *A Brief History of Human Computer Interaction Technology*, ACM Interactions, 5, 2, pp. 44-54. www.cs.cmu.edu/~amulet/papers/uihistory.tr.html
13. Russell M., 2005, *Using Eye-Tracking Data to Understand First Impressions of a Website*, Usability News, 1, 7. <http://www.surl.org/usabilitynews/71/pdf/Usability%20News%2071%20-%20Russell.pdf>
14. Sutherland I.E., 1963, *SketchPad: A Man-Machine Graphical Communication System*, In: AFIPS Spring Joint Computer Conference, 23, pp. 329-346, www.cl.cam.ac.uk/techreports/UCAM-CL-TR-574.pdf
15. Tognazzini B., 2009, *First Principles of Interaction design*, <http://www.asktog.com/basics/firstPrinciples.html>
16. Tolliver B., 1965, *TVEdit*, Stanford Time Sharing Memo Report, March, 1965 www.cs.cmu.edu/~amulet/papers/uihistory.tr.html
17. Tong H.M., Fisher R. A., 1984, *Progress Report on an Eye-Slaved Area of interest Visual Display*, Report No.: AFHRL-TR-84-36, Air Force Human Resources Laboratory, Brooks Air Force Base, Texas. Proceedings of IMAGE III Conference.
18. *User Centered Design*, <https://www-01.ibm.com/software/ucd/designpatterns.html>
19. Waloszek N., Gerd Z., *SAP AG, Product Design Center, SAP Design Guild, Design Tidbits*, 10/09/2000, 11/11/2003, <http://www.sap.com/>

20. Wiser M., 1991, *The Computer for the Twenty-First Century*, Scientific American, Sept. 1991, pp. 94-110.
www.citeulike.org/user/karin/article/771482
21. Wolpert A., 2009, *GUI Traditional and OO Approach*, Jan. 2009,
<http://faculty.roosevelt.edu/wolpert/cst270/13.pdf>