INTER!FACE

>> paper version

> Toon Van de Putte
> Hertstraat 24
> 2590 Berlaar
> toon@katdesign.be
> (015) 24 73 81

> Hogeschool Sint Lukas Brussel
> Grafische en Reclamevormgeving
> 1999-2000

> promoter: Steven Devleminck

!Note

My graduation project is primarily an online, dynamic project. Therefore, its main focus and value is in the online version [9]. This means the paper version of my thesis is not the most important one, it is just a simplified version of the online one.
I have discussed this with my promoter and department head, since this is not a traditional method of creating a graduation project. They both agreed, since the Internet is such an important aspect of my project. In fact, it would have been a shame not to use it to the extent I have.
It is impossible for this paper version to be equal to the online information. A website offers a whole range of possibilities for interaction and multimedia which are not available in print. Since my graduation project is mainly focused on interactivity and new media, it's impossible to fully explain and demonstrate the scope of this in a static paper document. It is therefore very important that you also visit the website that accompanies my graduation project [9], to get a full view of what it's all about.

!Contents

1!Motivation....4
2!Introduction....8
3!Materials and Methods....11
4!Research....14
4.1!I/O/D 4 browser....15
4.2!Total Annihilation RTS game (1997)....22
4.3!K10K, 'the designer's lunchbox'....26
4.4!VW Golf 3 (dashboard and controls)....28
4.5!Interfaces in society....32
4.6!Interview with Agnieszka Gasparska....37
5!Conclusion....44
6!References....46
7!Appendices....48
7.1!Interview with David Linderman....49
7.2!Illustrations....53

1!Motivation

Problem/context

Since the advent of electricity, designers of machines have had to actively design with the user in mind. And practice shows that is not always the case. Many average users today have simply reconciled to the fact that machines are just too complex for them. They are, of course, wrong. Indeed, machines are very complex today, but they still perform simple functions that everybody can grasp and understand. It is logical that not everyone can operate a particle accelerator, because its function is not universally understood. A VCR, on the other hand, should be usable by everybody. It simply records TV programmes for later viewing, a very simple function, which is only made more complex by the machine itself.

Facts

Since the first Industrial Revolution, the average complexity of our machines has increased dramatically. In the same time interval, more and more people have started operating these ever more complex machines. This has, over time, created a need for ease-of-use, for user-friendliness. In the beginning of the Industrial Age, there was little room for keeping the user in mind, it was considered spectacular that machines worked at all, nobody cared if it was easy to use. That doesn't necessarily mean they were hard to use, most pre-industrial and early Industrial Age tools and machines were fairly simple in concept and their working was very transparent. Take a bicycle, for example, nobody needs a manual for a bicycle. In the early Industrial Age the machine and its user interface were often one, there was simply nothing to design.

The complexity of machines gradually rose, and was boosted by the invention of electricity. Electricity introduced a ‘mysterious power’ that could make things move, buzz, light up, etc. Electricity is quite an abstract concept, which can be hard to grasp. The first major gap between user and machine was created by electricity. The relationship between user action and machine reaction was broken, intuitivity was gone. This is where the need for user interface design arose.

The home and personal computer were the next step in the alienation between man and machine. The inner workings of a computer are extremely

abstract and are impossible to visualise. Through miniaturisation, its parts have become very delicate and complex, which makes it undesirable that the user manipulates them directly. Here, there was a real need for a user interface. Without a user interface, a computer can not be operated at all. In fact, the first home computer, The 1975 Altair 8800, had flip switches for input and a row of LEDs for output. It didn't really do anything at all. This is an extreme example, but most of the early home computers were only usable by people who were willing to spend a lot of time learning how to operate these new machines.

That all changed when Apple released the first Macintosh. The first commercially available computer with a graphical user interface (GUI). Its goal was to bring the power of computers to everyone, not just programmers. Although the first Macintosh almost bankrupted Apple, the graphical user interface would become a success that boosted sales of personal computers. The GUI made using a computer intuitive and easy to understand. You didn't need to memorise a 500-page manual full of commands to use it. It meant that anyone could use a computer. And all that credit is due to the interface design. The machine itself is, up until today, basically the same. The success of the GUI proves that user interface design has become hugely important for the commercial success of any product, not just personal computers.

Because of miniaturisation and the ever more complex nature of modern technology, the functioning of most new machines is very abstract and not immediately apparent to the average user. This means there is a need for more and better user interfaces, devices that translate the user’s commands to the machine and the machine's output to the user. In fact, modern user interfaces are machines themselves, designed as translators. Rarely do users directly manipulate the machine today. Their input is instead fed into an interface and then translated to the machine. Most machines have some sort of (LCD) display that translates the machine output to something the user can understand. The impact of the GUI is clearly visible in the displays found on many electronic devices we use today. They all have icons, buttons, scrollbars, etc.

Personal

User interface design is not regarded as a very important design field, especially in website design and software design in general. It is often regarded simply as an extension to more traditional graphic design. Something decorative and of secondary importance. Of course, nothing is farther from the truth.

User interface design for software will be a very important factor in the next century, when it will have a great impact on how people use just about any machine. Even now, interface design is becoming ever more important and large corporations are starting to see the competitive edge that good interface design can give them. This is good news for all interface designers around the globe, but it is much more than just that. It signifies a shift in the way we design and build our machines.

The fact that user interfaces are so important today and will be even more important in the future, while at the same time their design is often neglected and not given due attention, has made me choose this subject. A lot has been written about user interface design, but I found most of it too conceptual and sometimes even dreamy. There are surprisingly few schoolbook-type publications on user interface design. Maybe because it is a design field that is fairly young and doesn't really have any established experts who'd rather write books than design interfaces themselves.

I hope this thesis will provide computer interface designers with some new perspectives on their work, or at least will give them new inspiration. It has been written with that idea in mind. This thesis is no how-to guide to interfaces, it is a study of user interface design today, with an emphasis on web site design. I do have the ambition, however, to make it usable for user interface designers out there in the real world. Usable as a guide, a reference work that contains the basic parameters.

2!Introduction

Problem/context

In order to write about interfaces, one has to know what an interface is, exactly. A basic requirement, sure, but hardly a simple one. As it seems, there is no clear, generally accepted, definition of what an interface is. There are literally hundreds of definitions circulating, all vaguely pointing in the same direction, but without a really unified core.

First of all, the term 'interface' is extremely broad and open to interpretation. 'Interface' is basically a homonym, with a lot of different meanings in a lot of different contexts. It is used in every scientific field where human-machine interaction is a subject. Its origins lie even further, as the following excerpt from The New Shorter Oxford English Dictionary [3] illustrates:

1 *A surface at which two portions of matter or space meet (ca.1870-1899)*
2 *A means or place of interaction between two systems, organisations, etc.; a meeting point or common ground between two parties, disciplines, etc. (ca.1870-1899)*

This is the basic definition that covers the essential meaning of 'interface'. It reaches beyond human-computer interaction, it covers basically any interaction or contact between two parties. It could just as easily be applied to the interaction between driver and car as to readers and books or viewers and TV's. It has no false limitations, but instead gives the term 'interface' a tangible meaning.

This definition also indicates that interfaces are always between two different environments. There is no such thing as the interface *of* a computer program, it is always *between* the software and the user. This definition tells interface designers once and for all that *it takes two to interface*.

The Oxford Dictionary also uses the phrase 'common boundary', which, in turn, is very important. An interface encompasses some sort of common ground, something that is mutually understood. In computer interfaces, an example can easily be found in the evolution from command-line interface to

the desktop metaphor we know today. In the case of the command-line, the common ground was language. The user could more or less understand the words and phrases the computer displayed, and the computer could interpret the text input from the user. In today's desktop metaphors, the common ground has broadened a lot. In addition to language, we can now communicate with computers through more visual means, like icons and windows. The computer knows which icon is associated with which file type, how many windows are open and how they are stacked, and so do we, the user.

Steven Johnson, in his book ‘Interface Culture’ defines ‘interface’ as ‘software that shapes the interaction between user and computer, … a kind of translator’ [1]. Note that he uses the terms ‘software’ and ‘computer’, limiting the possible applications for his definition. This definition implies that interfaces are always pieces of software running on a computer, used for facilitating interaction with the computer. Throughout his book, Stephen Johnson maintains the emphasis on computer interfaces. He focuses mainly on computer interfaces in a historical perspective and provides a possible scenario for the future.

In Nicholas Negroponte’s ‘Being Digital’ [2], no definition is even mentioned, although the book contains an entire chapter titled ‘Interface’. One might call his subtitle for this chapter a definition, ‘Where people and bits meet’, but it would be a very vague one.

3!Materials and Methods

Problem/context

I will approach the subject of user interface design via two main routes. The first is literature; books and articles, websites,... on user interface design and design for the web in particular. The second route is analysis of the current state of events in the field of computer user interface design.

Together with my promoter, I have assembled a list of works written by authorities in the field. I regard these as basic information, providing the essential background to performing a study in user interface design.

In addition to that, I try to keep as up-to-date as possible on the subject, by reading magazines, checking the web, subscribing to online mailing lists, and generally by staying in touch with people in the field.

The second route is analysis of the subject. Because this thesis is part of my graduation project, there is an interaction between practice and theory where I can test my theoretical findings to the practice of my interface design. The website [9] I created about my project is an online logbook of this process. In addition to this form of self-study, to broaden the scope of this text, I have also conducted a number of case studies. These are not limited to the web, or even computer user interfaces, but also include a study of the user interface of the Volkswagen Golf (Rabbit in the US) III car. I am a firm believer in interaction between different specialisations and professions, and am convinced that cross-breeding between all different types of user interface design can result in true, productive creativity and originality.

Additionally, I have also conducted a survey among designers in the multimedia field and I have interviewed a few of them more extensively.

The problem with the survey was that not enough people I contacted reacted, so it isn't representative at all. In addition to that, some of those that did respond, provided useless results, with some questions unanswered or not filled out correctly. The total number of usable results was about twelve, so I made the decision not to use the (limited) results of this survey.

The interviews suffered from similar problems. First, I interviewed David Linderman (Fork Unstable Media [11]), which proved to be a waste of time. Most of my questions were answered with a simple 'yes' or 'no', while they

were not yes-or-no questions, other answers were completely off-the-wall and didn't have any relevance whatsoever. No usable answers, so I didn't use it (it is in the appendices at 7.1 though, if you're interested).
Because this first interview failed miserably, I interviewed Funny Garbage's [12] Agnieszka Gasparska. She was very very busy, and got sick on top of that, so her answers only came in hours before the deadline. I managed to move the deadline, to incorporate her interview, which I think is a very valuable addition to this text (a lot more valuable than Mr. Linderman's reply, at any rate).

Facts

With each case-study, I explain why I chose that particular design and what it contributes to my thesis. I also explain what it is, how it functions, who made it and give some broader context to put it all in perspective. Finally, I summarise shortly the findings of the case study in question.
In these case-studies, I explore the different aspects of user interface design, trying to give as complete an overview as possible of what is achievable and what is being created today.
In the conclusion of my thesis, I summarise the conclusions of my case studies and literature study and formulate a general conclusion of the entire research process. This conclusion is given a real-world identity in the interface to the text of my thesis. This interface should ideally be a perfect representation of my research results. Because of technical and time limits, it will probably be just an approximation of this, but I hope it will be valuable as a research experiment.

Personal

By using case-studies, my text will have a real-world resonance and won't become too philosophical. It also makes it much more readable and ‘alive’, because it is based less on dry theoretical literature and more on practical examples.

4!Research

4.1!I/O/D 4 browser [7] - see also 7.2

Problem/context

The I/O/D browser is a browser unlike any other. On a technical level, it does what you'd expect any browser to do: view web documents. But the way in which it does that is radically different from anything Netscape or Microsoft has ever produced. I/O/D confronts you with the fact that it's the browsers that shape the web, not the sites themselves. The code of the pages on a site are just an indicator of what a browser will render on screen. Newer technologies such as Macromedia Shockwave and Shockwave Flash are an attempt at creating uniformity in the virtual world, but since HTML is still the basis of most information on the web, I will ignore them in this case study (incidentally, I/O/D ignores them too).

Facts

Web documents are generally written in HTML, which is a markup language. This means that the client decides how it will translate HTML tags to the screen. It has always been the Holy Grail of webdesigners to have a site that looks identical on all browsers on all platforms, since this is not at all a given fact. I/O/D does away with all that, it's very individual about how it renders what it finds on the web. In fact, I/O/D is largely useless as a browser in the traditional sense of the word. It is an experiment, a visually stunning technological experiment. You might even call it work of art, a comment on the WWW.

Visually, I/O/D is quite a leap from any other usual piece of software, were it only because it starts up with nothing but a completely black window. This window invites the user to do something, to ‘fill the void’, so to speak. By dragging the mouse in the window, the user can create white rectangles, which can then be given a function. There are six functions in total. The manual gives the following explanation:

The six functions are:

Crawler
Map
Dismantle
Stash
HTML Stream
Extract

Crawler
function:
The Crawler is the part of the Web Stalker that actually links to the World Wide Web. It is used to start off a web session. The Crawler window shows the current status of the Web Stalker.

how to use it:
Open a window. Turn the window into a Crawler by opening the pop-up menu and clicking the right mouse-button. From the menu, choose the function 'Crawler'. In the window a bar will appear split into three equal sections.
Open the pop-up menu again. It will list three choices: 'Open URL', 'Stop' and 'Close'.
To start a new web session choose 'Open URL'. When the dialogue box appears, type in the address of the site that you wish to access. Once it has been typed in correctly, press 'Return' or click the cursor on the button in the window. The Stalker will then open your Internet connection and access the site. You can also use this function to go to a different site whilst you are already online.
If you have difficulty in opening up the Internet connection from the Web Stalker, try opening a net connection first and then opening the Crawler.

When you want to finish a Web session, choose 'Stop' from the pop-up menu.
To close the Crawler window choose 'Close' from the pop-up menu.
The dot moving across the split bar shows what stage the Crawler is at. The first section of the bar shows the progress of the Net connection. Once connection is made and a URL is found, the dot jumps to the next section of the bar. The second section displays the progress of the Web Stalker as it reads through the found HTML document, looking for links to other URLS. The third section of the bar monitors the Web Stalker as it logs all the links that it has found so far.

Map
function:
The Map maps the links between HTML documents. It shows references to individual HTML documents as circles and the links between them as lines. Sites with more links to them have brighter circles.

how to use it:
Once a Web session has been started you can begin to Map it.
The Map starts at the first URL opened by the Crawler and then moves through all the links from that site, then through the links from those sites, and so on.
To see the URL of an HTML document, and hence its likely contents, click on the circle. The URL will be displayed at the top left of the Map window.
So that you can keep track of the previous HTML document you clicked, another circle will appear inside it.
To read the text of a specific HTML document, drag the circle into an Extract window, (see below).
If you right-click in the Map window to get the pop-up menu, you can

choose to 'Save' the map of HTML documents. A saved Map can be read off-line. The name of your saved Map file, will be appended with '.iod'. This '.iod' format enables the Web Stalker to distinguish these Map files from other file types. The saved file will contain all the information regarding URLs and links that are needed to re-draw the Map at another time.
You can open a saved Map file into the Map window at any time by choosing the 'Open' option from the pop-up menu.
If you go back online with a saved Map, there is a slim chance that some of the sites will have changed. If the URL cannot be located by the Web Stalker, you will get a message such as 'File not found' back from the server, or 'Connection cannot be established'. These will appear in the Crawler window.

Dismantle
function:
The map shows HTML documents and links between them. The Dismantle window is used to work on specific URLs within HTML documents. URLs at this level will be specific resources such as images, e-mail addresses, sound files, downloadable documents, etc.

how to use it:
Open a window. Open the pop-up menu by right-clicking the mouse. From the menu, choose the function 'Dismantle'. Clicking and dragging a circle from the Map window into the Dismantle window will display all URLs referenced within the HTML document you have chosen. These resources are also indicated by circles. Clicking on any of these circles reveals the URL of the resource as text in the top left hand side of the Dismantle window.
You can save these references to resources on the Net by clicking and dragging them into a Stash window described below.

Stash

function:

A flexible way to save the URLs. The Stash provides a document format that can be used to make enhanced records of web use. It can even be read by old-style web 'Browsers'. Stash files can be passed around as a separate document between different users so that you can share information on web resources.

how to use it:

Open a window. Open the pop-up menu by right-clicking the mouse. From the menu, choose the function 'Stash'.

When you want to add a URL to a Stash, drag it from the Map and drop it into the Stash window. It will then link to any other URLs in that Stash.

To save a Stash, choose the pop-up menu from the window by right-clicking the mouse and choosing, 'Save'. A dialogue box will appear asking in which folder you wish the Stash to be saved.

To open a previously saved Stash, open the Web Stalker and make a 'Stash' window. Choose 'Open' from the pop-up menu and choose the saved Stash you want to use. Stash files will be appended by '.stx'

You can save the Stash as an HTML file if you choose 'Save as HTML' from the pop-up menu. This will allow you to read or publish the Stash as a Web 'page' of links which can be read by Web 'Browsers'. To open a Stash in a web 'Browser', open the 'Browser' first and then use the 'Open File' or 'Open Page' option under its File menu.

You can have several Stash windows open at once. (This is useful to make separate resource documents for different subjects during one web session for instance).

HTML Stream

function:

Shows all of the HyperText Mark-up Language, (the computer language that describes the content and links of the World Wide Web) as it is read by the Web Stalker.

how to use it:

Open a window. Open the pop-up menu by right-clicking the mouse. From the menu, choose the function 'HTML Stream'. You only need one HTML Stream window open at once.

You can close the window by choosing this option from the pop-up menu.

Extract

function:

Shows all the text from a URL. Use your Map or Stash windows to get to these URLs.

how to use it:

Open a window. Open the pop-up menu by right-clicking the mouse. From the menu, choose the function 'Extract'. You may have more than one Extract window open at once.

To Extract the text from a URL click onto a circle illustrating that URL in the Map window. Drag it from the Map into the Extract window. It will then open up as a text file.

The Extract window has a scroll bar which can be used to move up and down the text in the normal manner. Text in the window can also be saved. To do so, open the pop-up menu again by right-clicking the mouse in the window.

You can close the window by choosing this option from the pop-up menu.

Now, it is essential that each window can be given any of these functions. The user first creates the window and then assigns it with a function. This is a radically new concept in software interface design, it is in many aspects even an inverted version of how a software programme is usually organised.

Personal

One of the most interesting functions is the ‘Map’ function. This function displays a sort of ‘sitemap’ in the window. It starts from the first page it finds on a url the user has provided, and then starts tracing the links. It builds a web-like map as it finds more and more linked pages, ultimately resulting in a star-like cluster of nodes, each node representing a single document. The user can then drag one of these nodes into another window to perform a function on it (such as the ‘Extract’ function). So, this map becomes some sort of starting point in the I/O/D browser. The map functions as an ‘invitation’ to explore the site it represents. It challenges the user to find out what that lonely little dot that sits there by itself is. It gives a sense of power, of being able to oversee it all. The surprising thing is that I/O/D doesn't use any amazing new technology for this, it's just a Director movie that uses standard HTML pages found everywhere on the web. It interprets the code almost to the extreme while at the other hand, it doesn't interpret at all, it just represents the links between documents. It goes to show that browsers are nothing but the glasses through which we see the web. In fact, the web is just a collection of ASCII text files, with some binary files scattered here and there. The web contains nothing but raw information, which is translated to the user via software. The I/O/D demonstrates this in a very clear way. It reveals that, as Erik Davis puts it in ‘Flame Wars’ [6]: “..., complexity space is complexity space – any information system, when dense and rigorous enough, takes on a kind of self-organised coherence that resonates with other systems of complexity”.

4.2!Total Annihilation RTS game (1997) [8] - see also 7.2

Problem/context

A computer game is different from any other computer interface in that it should make things hard, not easy. Its objective is, in a way, exactly opposite to that of most interfaces. That might seem simple enough, but it's often a very thin line. As Donald Norman [4] puts it:

> *As you might imagine, making things difficult is a tricky business. If a game isn't difficult enough, experienced players lose interest. On the other hand, if it is too difficult, the initial enjoyment gives way to frustration. In fact, several psychological factors hang in a delicate balance: challenge, enjoyment, frustration, and curiosity.*

This is why computer games are an interesting subject: they stretch the whole concept of interface design and experiment at reshaping it. Maybe this is why many radically new interface elements are often first discarded as toys or gadgets. (a lot of die-hard computer users thought of the first Macintosh's GUI as a toy, usable only for complete beginners [1])

Facts

Total Annihilation fits in the Real Time Strategy (RTS) genre, which was born together with Dune 2, in the early nineties. An RTS game is a combination of arcade-type action and tactical and strategical insight. The object of most RTS games boils down to destroying your opponent, no surprises there. But in RTS, the player has to do more than just point a gun and shoot. One has to collect resources in the playing area, and use them to build vehicles, structures, train soldiers, etc. It is the objective of most RTS games to, essentially, build a better army than your opponent, and preferably being faster at it.

Most RTS games have a more or less linear mission structure, where each mission takes place in a different playing area (map), and victory leads you to the next mission. The opponent in an RTS game can be either the computer's

AI or a human opponent (usually via the Internet or a LAN). Not all RTS games are limited to two players, especially since the advent of multiplayer games over the Internet, as many as 10 people can join in one game.
The view a player has of the playing field is generally top-down, or at a slight angle. This main view only covers a small area of the total playing field, the player cans scroll sideways or up and down to view the rest of the playing area. Usually there is also a small ‘minimap’, which is a small stylised overview of the total playing field. This basic layout has remained virtually unchanged since Dune 2, with only minor adjustments.
Total Annihilation is, in many aspects, a typical RTS game. One could call it an ‘updated’ version of Dune 2, which looks quite dated today. Total Annihilation was, and still is, a very popular game. It does exactly what an RTS game should do: provide the player with a lot of different options, so the outcome of a battle is often very unpredictable. On the other hand, it does provide ease-of-use and a lot of tools for managing a large army and base, so the amount of mindless pointing and clicking is reduced to a minimum.
The game is constructed out of a series of missions. The player chooses a side –either Arm or Core- and then proceeds to the first mission for that side (the missions for each side are different to an extent, although there are similarities). The player gets a full mission briefing first, with a short introduction to the general storyline of the game and an explanation of that mission's objectives (most of the time, the objective is simply to wipe out the other side, but sometimes the player has to build a certain structure or capture a location in order to complete the mission). Then the actual mission begins.
The main screen is taken up mostly by a top view of the battlefield. At the top of the screen, two status bars can be seen indicating the levels of metal and energy the player has. These two resources are needed to build structures and vehicles. In the top left corner of the screen is the ‘minimap’, a small representation of the total battlefield. A small yellow rectangle in the minimap indicates the portion of the playing field that is visible in the main window. Below the minimap are the ‘build’ and ‘order’ menus. Depending on the type

of unit selected either one of these or both of these menus are available. Some units can build and move (construction vehicles), while others can not build (tanks, artillery,…) or can't move (construction buildings). Generally, all buttons are visible, but the ones that have no function with the given unit are greyed out. Further indicators in the main screen show the energy/metal consumption for the selected unit and its level of damage.

Personal

The way in which the controls are organised in Total Annihilation shows the designers knew very well what they were doing and had an understanding of what game interface design is all about. First of all, the entire interface is geared towards the object of the game: winning the mission. The player is not frustrated by clumsy controls and ‘hidden’ information. Everything is made as easy and as accessible as possible, the real challenge is where it should be: on the battlefield.

The game also offers tremendous depth and an almost infinite array of possible scenarios. Where most RTS games focus on traditional land units such as tanks, jeeps, etc., Total Annihilation gives the player access to a large amount of airborne units and sea units. Later versions of the game even include the possibility of building an entire base on and under water. The unit structure of Total Annihilation is also completely customisable. The game designers have included the possibility of adding new units and modifying existing ones. This has spawned many websites offering new and improved units for Total Annihilation, almost like real arms dealers. It also allows for shifts in gameplay and changes in how players will fight certain battles. Just like in real warfare, a single new type of weapon can change the rules of war. And the remarkable thing is that all of this is uncontrollable by the developers of Total Annihilation, it is al in the hands of the player community. Just like Quake II and III, this game has been taken over by the Internet community.

The open structure of this game has made it a typical example of how the Internet transforms the way we develop and use software in some sort of Darwinian environment. Not everything is engineered anymore, developers only create the first ‘seed’, after which a whole community of users hacks into it and continually changes it. It's very remarkable how customisable and flexible games like Total Annihilation and Quake II and III keep getting played years after they were released, while other, more rigid variants, die out after

the initial fascination is gone.

4.3!K10K, ‘the designer's lunchbox’) [10] - see also 7.2

Facts

K10K (Kaliber 10.000) is a web-based magazine, featuring graphic designers from all over the globe. K10K publishes weekly ‘issues’, which are basically small websites or web-based design projects by individual designers. The issues vary greatly in style and size, but they all have to qualify by the high standards set by the two owners of the site. The site has three main sponsors: Apple, Information Highway (who do the backend) and Netburauet Araneum (their host).

The site is managed by a core team of two people, who call themselves ‘token’ and ‘msschmidt’. Apart from them, there is an entire team of people working on the backend, and of course the designers featured in the weekly issues.

The Kaliber 10.000 site has more to offer than the weekly issues, however, and has the look and feel of a graphic design portal site. It has an ongoing [on]display exposition, where basically anyone can submit their desktop images, an online store where you can order their merchandise, and an e-mail newsletter you can subscribe to. The homepage also has a number of smaller items, such as the ‘kaliberkam’ and special features.

The site has a powerful backend and most content is generated dynamically via an SQL database. This allows it to be very huge indeed, with browsable and searchable archives of just about everything. There is also a lot of javascript to be found on the K10K site, with custom scrollbars and rollovers. Flash or Shockwave are not used in the K10K site itself, but do crop up regularly in the weekly issues.

Personal

The site looks very clean and crisp, and seems to have a strong affinity with the ‘pixel’ look, reminiscent of old-school arcade games. Small visual jokes and animated gifs like the 'kaliber kam' ensure the design looks fresh and alive. This style offers a look that's very different from more traditional portal sites, which suits the objective of its creators. The site is not designed from a

purely utilitarian perspective, which makes it a fun experience.

It is immediately apparent when one views the site's homepage, that it is almost entirely dynamically generated from a database. The page does not look rigid, but is divided in several 'modules', that are stacked up and next to each other. The page is divided into several columns, which are in turn divided in 'windows' or clearly separated rows. This allows for easy updating and altering of content, and at the same time gives the site a dynamic look.

The Kaliber 10.000 site is a great example of a web site that excels both technically and designwise. It demonstrates that one does not necessarily exclude the other. It's also a look into the future of magazine publishing, with its weekly issues of featured designers. These issues are the core of the site, everything else is structured around them. But the site is not limited to the issues alone, it also has an online store, a desktop exhibition, a browsable archive, and other extras. This is how the future of publishing looks: sites that offer a lot of content, based around a central theme or product. Not just a few hyperlinks, but genuine additional content that creates a true multimedia experience.

4.4!Volkswagen Golf 3 (dashboard and controls) - see also 7.2

Problem/context

I selected this particular model, because it is one of the most common cars on European roads. It has a very broad user base, which makes it and interesting subject in terms of usability and interactivity.

I will not discuss the entire car in this case study, but instead will focus on the design of dashboard and controls. The dashboard design of a car might seem like a leap from computer interface design, but I think a lot of parallels can be found.

The basic idea is the same: control and guide the interaction between man and machine. Visual information and feedback is equally important in both fields, as is guiding the user's intuitive impulses and logical reasoning.

The main difference is that a car is near 100% hardware, where a computer program is near 100% software. Examining this difference and how it lets designers find different solutions for common problems is a key goal of this case study.

Another key difference is the audience for which both types of interface are intended. Most cars are designed to be driven by basically anybody, within broad parameters. They should be simple enough for most people to comprehend, after some basic training. Software programs have a different status. Only in the last decade or so have home/personal computers reached a true mass audience. Initially, they were designed for a specialised audience, specially trained to handle and operate them. A lot of current software programs still carry this legacy and are not as transparent and straightforward as they should be. Software designers have only had about ten years to build up experience in designing for a mass audience and are still learning.

The relationship between the similar context and goals and the radically different situations of the VW Golf 3 dashboard design and software interface makes this car an interesting subject, since it can give new insights and be a

powerful source of inspiration.

Facts

Since its conception in the 1970's, the VW Golf was one of the most popular cars available. It introduced the hatchback concept and provided affordable and reliable transportation. It also saved the Volkswagen company.

There have so far been four generations of this car, more or less updates of the same basic concept. The one i discuss here is the third generation model, which dates back more or less ten years and is still a common sight.

The basic VW Golf 3 is doesn't have a lot of bells and whistles. The dashboard design is fairly plain and common, designed to be practical more than beautiful.

The locations of most controls are similar to those of other cars, with only slight differences. This ensures that users who are new to this particular car, but have driven before, can benefit from their experience.

This does not mean that the only guideline for positioning the controls is that set by other cars. Visual clues and mapping, intuitivity, logic, etc., are also very important in determining which controls go where and what size or shape they should be.

One example might be the steering wheel. It seems like the most straightforward and simple solution to the problem of controlling the car's movement, but it's a lot more complex than that. History shows that at the dawn of the automobile, designers and engineers struggled and experimented a lot with different devices for the steering control. Levers, handles, sticks, and many more exotic designs came and went. Finally, the steering wheel, as a totally new concept emerged, and survived until today. It's easy to use, doesn't impair the user's movement and technically simple. It's more or less the perfect steering device. And that's why no one gives it a second thought, it's so familiar and easy that anyone can use it. Most controls share a common background of decades of experimentation and trial-and-error before eventually reaching their final status.

This historical context, which software interfaces mostly lack, allows for a meaningful analysis of how the controls are positioned and how the whole interface is conceptualised. They have stood the test of time and have proven that their design works for the majority of users, which is very valuable information.

Most controls on the VW Golf 3 dashboard are grouped in a logical manner. Ventilation, lights, visual feedback, ... all have their specific areas on the dashboard where they can be easily and intuitively accessed. This grouping is inevitable in a design as complex as the dashboard design of a car. The human brain can only store seven to eight 'digits' of information in the short-term memory, which means interfaces with more than this number of controls can disorient the user. The solution to this problem is grouping controls into a logical organisation. This creates a 'shortcut' for the user, who now only has to store the outline of the interface in the short-term memory, and can still locate the right controls or readouts.

The positioning of these groups is determined by the frequency of their use. Controls that are used most often, are closest to the steering wheel, and those that are used less often are further away. The dials and LCD-display that provide information about speed, mileage, rpm, etc., and the warning lights, are located above the steering wheel so the driver has them in his field of vision most of the time.

Not only the positioning is important, also the shape and size of the controls. There are basically four main different types of controls: turnable dials, scrollers, push-buttons and levers.

The turnable dials are used for controlling the headlights and ventilation, where different 'modes' can be selected. They have a round shape and a handle, both hints at how they can be operated.

The push-buttons are more or less rectangular and flat, which indicates they should be pushed. They are used for functions that only have two modes or for toggle-functions, such as electrical windows or the air-conditioning.

A notable detail is that the turnable dials have icons positioned above or

around them, where the push-buttons have the icons in the pushable area of the button.

The scrollers are used for gradually controlling the seat heating, the level of the headlights and the lighting level of the display panels. They bulge slightly out of the dashboard, and have a small rounded frame around them, so it becomes clear how to operate them.

The levers are all positioned around the steering wheel and control the windscreen wipers, headlights, indicator lights and cruise-control. These are the most frequently used functions, which explains why they are so close to the steering wheel. The shape of the levers allows the driver to control these functions without releasing the steering wheel, which could be dangerous. Furthermore, it would be extremely cumbersome for a function like the indicator lights to be located out of the driver's immediate reach. An additional advantage of the levers is that they can be operated in a large variety of ways, which allows many functions to be combined on one single lever.

There are also, of course, the pedals, gear shift and handbrake. These are all more particular controls that have limited relevance to the point I am trying to make, so I will not discuss them.

Personal

The VW Golf 3's interface is not a bad design at all. It has some flaws here and there, such as a slight inconsistency in how the lighting controls are organised, but overall it is very straightforward and easy to use.

The downside is, of course, that this interface is extremely dull and plain. It is practical to the extreme, testimony to the German origins of the car. However, I don't think practicality and aesthetics necessarily contradict each other. The bluntness and lack of subtlety in this car's interface have nothing to do with practicality, they just indicate that not a lot of attention was paid to making it all look good as well. A more imaginative and developed interface design could even enhance the usability of the car, in making it more accessible. The Volkswagen Golf 3 is clearly a driving machine, not an artwork on wheels.

4.5!Interfaces in society

Problem/context

In the last quarter of a century, interface design has become an ever more important part of society. As our machines and tools become more complex all the time, there is a growing need for more potent user interfaces, which can translate the inner workings of the machine to the user. Because machines are becoming more complex as technology advances, their user interfaces need to bridge ever greater gaps between machine and user. Interfaces are becoming easier, while the machines they operate become harder.

The development of user interface design has been greatly influenced by the development of the personal computer. In fact, a lot of machines we use today are in some way variations on the computer theme (pocket calculators, cash dispensers, vending machines, stereos, cellular phones, …). The input-processing-output model has proven to be a useful model for a whole variety of appliances. A basic computer has no generic function, it just computes. It gets a function because someone writes a programme that uses the computer's calculating power and interprets it. This means that computers are very flexible machines, you could use them for virtually anything, if you provide them with the right in- and output organs for the job. It is in fact the software that does the real work, not the computer. It's the software that allows for user input, does something useful with it and generates output. This importance of software means that interface design is one of the most important factors in how people use machines today and certainly in the future.

Software programmes rely on quality interface design more than anything else. A hammer is a hammer, its shape is determined by its function, but a software programme is, in a way, shapeless. It is completely virtual, it has no material incarnation, which means that an interface designer can determine to a great extent how a programme works. Colours, shapes, size, functions, sounds, buttons, … can all be custom-designed. In fact, they have to be

custom-designed.

The user interface determines how people will operate a machine. It provides the user with certain information and holds back other information, so it becomes clear how the system can be manipulated. This means that a perfectly good machine with a bad user interface will be perceived as a cumbersome piece of equipment, people will think it is a bad machine. For the user, the interface *is* the machine, especially with computer programmes. This means interface designers have a great responsibility towards manufacturers and consumers. Not only can a badly designed interface affect sales or overload the customer support service, it can even be dangerous. Badly designed control panels for nuclear power plants or aeroplanes can cause serious accidents, which are afterwards usually blamed on ‘human error’ [4]. So interface design is not purely aesthetical, it's not just about picking the right colour or sound for a button.

Technology has taken quite a few leaps since WWII, up to the point where a lot of technology we use every day is, in fact, ‘invisible’, we have no visual clues as to how it works. Microwave ovens are a typical example, but also cellular phones, personal computers or compact discs. In fact, the transition from vinyl records to CDs is a good example of how miniaturisation and hi-tech can make technology more invisible and less straightforward. Have a look at the comparison table on the next page.

Vinyl record	**CD**
The disc is visible when it's playing	Disc is generally hidden within the player device
The user puts the needle on the groove	The disc is inserted in the player and the machine takes it from there
The progress of the needle is clearly visible and it's easy to see if a record is just beginning or has nearly ended	The user can't see where the player is extracting music from the CD. The only indication is the track nr. and, with most players, the remaining tracks which are displayed on an LCD screen
The groove on a vinyl record is visible	The holes in the CD's surface are invisible, the only thing the user sees is a rainbow-like reflection
A record player does not have an LCD screen that displays information on the status of the player	The LCD screen of a CD player is the only source of information the user has. It can display time remaining, time played, number of tracks, current track,…

This is a perfect example of how the gap between people and the technology they use is becoming greater with the advancing complexity of electronic equipment today. This is, of course, inevitable. As technology gets more complicated, people will lose track. The only way to bring it back in touch is to use older and/or simpler technology. But in most cases, that is not an option. And anyway, do people have to understand exactly how a machine they use operates?

This is not the problem, the real problem is giving people a good 'mind view' of how a given system operates, so that they can predict the outcomes of actions and the system becomes transparent. This user's model [4] provides

the user with a logical scheme of how the system operates and allows him to use it more intuitively (as opposed to just learning how to operate it).
Now, this user's model can be completely different from the way the system really handles user input, all it has to do is make sense. This is especially true in the computer world, where everything is virtual. All an interface designer does, essentially, is provide the user with a conceptual model of how a programme works. A perfect example is the Graphical User Interface (GUI), as used in Windows and MacOS. It is nothing more than a visualisation of how a computer works. Beneath this top layer of windows and icons, computers work with ones and zeroes, tiny pulses of electricity, nothing more. The GUI is a carefully constructed model, designed to give the user more clues as to what he is doing. It is not reality, it's just a model.

Personal

Although the GUI has opened up the world of computers to the mainstream community, the philosophy behind it also has a darker side. If the user's model of a system does not have to correspond with reality for it to work, then that means a whole spectrum of illusions can be created and cultivated. The freedom that the virtuality of the digital world provides, can be perverted into a prison, it can be used as a control mechanism.
The success of a completely virtual character as Lara Croft, the heroine of the computer game series ‘Tomb Raider’, indicates that it is possible to create a pseudo-reality, completely engineered and designed, and to get away with it. The movie ‘The Matrix’ provides an extreme view of how this might change the world we live in. It is probably not going to become a reality, but ‘The Matrix’ gives an indication of what could be possible.
In my opinion, the virtual and the real will start to coexist and get entangled in the near future. We already have Lara Croft, virtual Japanese singers, on-line ‘personal assistants’ and robotic dogs, the next step is that these creations will become as real as reality and computer-generated people will be ‘more human than a human’ [5].
Even today, an increasing number of people spend a substantial amount of their time online, plugged into the Internet. The virtual world of computers and

the invisible ‘space’ of the Internet, has become a part of their reality, of their world. As more and more powerful computers start generating a new layer of reality, the older ones will become buried increasingly deeper. We will have modified nature itself, and created an escapist, ‘censored’ version of reality. And it will be normal.

4.6!Interview with Agnieszka Gasparska (Funny Garbage [12])

PERSONAL

What direction where you going in college, and how did you decide to become a (graphic) designer?

When I came out of high school I wanted to study Science or Math or something academic like that, but I got a full scholarship from the Cooper Union for Art and since it was an amazing opportunity and the school was renowned for its Engineering and Architecture programs as well, I felt that I wouldn't limit any of my possibilities. On the contrary. So because I had never planned on seriously pursuing anything purely artistic before, I went through Cooper trying everything that interested me to take full advantage and see what would happen. I took painting, photography, sculpture, typography, motion graphics, interactive design etc. After studying in France in my third year and doing mostly photography, drawing and traveling, I came back to school and got really

fascinated by interactive design and using my photographs and things I had collected to create design work.

I really got into programming my own stuff in Flash and Director and it became like a perfect union between math and logic and visual art. Because I was also doing a lot of experimental typography and professional graphic design it just seemed like a natural thing to pursue at the time and see what type of opportunities I could have in the field.

Have you used computers since you were a kid, or have you only started using them later on in life?

No, I never used computers as a kid - I grew up in Poland and we simply never had any around. My family bought a PC when I was in my last year of high school but I never used it. I started using Mac's occasionally for writing papers when I started college, but it wasn't until my first typography class in my second year that I became a religious user.

TECHNICAL

Do you consider yourself an interface designer, web designer, graphic designer, ... ?

I think all of the above, but most importantly a graphic designer because you can't be a successful interactive or web designer if you are not a good graphic designer first. Unless you're just into the technical side of things. Otherwise the first two really can't thrive without the third.

What do you design mostly? CD-roms, websites, kiosks, ...?

Working at Funny Garbage right now, I've worked on a variety of different projects including interactive kiosks for several new museum spaces in the United States such as the Museum of Natural History's new planetarium or the Experience Music Project in Seattle. I've done a few CD-Rom's as well, and some websites ranging from something like the Scooby Doo home site for CartoonNetwork to a corporate site for Knoll. I also do a significant amount of work on my own outside of Funny Garbage, including identity work, art direction and design with my boyfriend of a quarterly print magazine called Sweden & America, as well as some experimental web projects.

Do you also do print design?

Yes, see above.

In designing websites, what tools/programs do you use? And on which platform(s) do you work?

Illustrator, Photoshop, Flash etc. All Mac based programs

Do you know any programming or scripting?

Yes, some. I've done some interactive work in Director and know some Lingo and a tiny bit of Javascriipt.

Do you think it is useful or necessary to be able to do extensive programming and scripting, as a designer?

Not necessarily but I think it is definitely very important to be able to understand and foresee how a design is going to be implemented technically. Knowing some of the technical stuff involved allows you to incorporate that understanding into the initial designs right from the beginning because you can really think about what factors affect the making of it. It's like trying to make a bronze sculpture without having a clue about how bronze behaves and how to work with it - only how it looks. You just have to imagine it and hand it off to someone who knows what they're doing and pray that they do the right thing with it. I prefer the other way - if I can't do it myself, at least if I have some idea I can work together with the people that can.

GENERAL INTERFACE DESIGN

Do you think there is a relationship between the design of a site like AltaVista and the interface design of a car?

However, the design for a site like AltaVista serves a very different purpose than that of a car - AltaVista is an information source, a listing of stuff , a directory with no physical aspect to it, whereas a car is really a mainly physical experience. It is a means of moving around a passage. So while maybe the structure of information between an AltaVista site and a car

shares a similar logic for hierarchy of information, the intrinsic difference between the two experiences makes the interfaces very different.

Would you like to design the interface for, say, a VCR, a cellphone, or a car?

Sure. I think designing interfaces and user experiences for things that are physical objects rather than computer spaces is very interesting.

Is there a contradiction between aesthetical appeal and the usability of a design?

I don't think there's necessarily a contradiction but they are two different things that combined can create a great interactive experience.

Do you think interactivity has changed the way people look at the world?

It has definitely changed the way people look at technology. But I think in some cases though, it may have made people look less at the world and more at computers - or at the world through computers I guess.

What are the criteria you use to judge the design of an interface?

Pretty much clarity is the main criteria. Beauty is subjective and personal but I judge by that also.

If I get lost and confused - that's bad.

What guidelines do you use when designing an interface? Is there a set procedure, or is it different every time?

I think it's different every time because every project is different in what it's trying to communicate and do. But there is s certain sensibility that is present in every project equally. An awareness of clarity and simplicity and logic most of all.

Have you got an alternative for the desktop metaphor in current GUIs? Do you think the desktop is a metaphor that works?

I'm not quite sure what you mean, but I think if I do understand a bit then I would have to say that the desktop metaphor is good for computer interfaces

like MacOS or something but not for everything I do.

Do 3D metaphors have a future in interface design? Have you ever designed an interface that used a 3D metaphor?

I don't mind meaningful 3D metaphors, and I've used them in certain project like the guitar kiosks for the Seattle music museum, but I don't like the cliches of 3D visuals used in projects in any which way.

Do you think the success of computer games like Quake, Tomb Raider, Dune 2(000), etc. is a good thing for the mainstream acceptance of new media, or does it give people the wrong idea?

I don't necessarily think it gives people the wrong idea ass long as they realize that video games are not the only thing that new media is.

What do you think will be the most significant evolution in interface design in the (near) future?

I don't know. Hopefully a move away form the "everything at once" approach.

THE WEB

What is the most significant evolution in web site design in recent years (months?), in your opinion?

See below, the "Do you ever get frustrated..." question. Given that, I think all of the sites and web projects that have been created in the last few years that vehemently challenge and break up that standard of "big name" web design are the ones that I admire the most. Movement and more complicated programming also allows for more fluidity on the web which is great.

Do you want your sites to look the same on all browsers on all platforms?

They don't necessarily have to look the same because you really can't always

control it, but if I can help it, I'd like for things to not look awful - that's pretty much the goal.

Do you use Flash a lot?

On certain web projects - like CartoonNetwork for example has a lot of flash. But I'm really against sticking Flash into everything just because it's cool - there are a lot of sites out there that have a ton of stuff in Flash and it does nothing for the site - it gets annoying.

Do you ever use Dynamic HTML?

Yes , we used it heavily in building the Knoll site, for example.

Do you ever get frustrated by the vast amount of bad sites out there?

Yes and no. No because there's always going to be bad stuff out there no matter what medium you take, be it tv, film, theater - anything creative is bound to have its bad and its good. But yes, because in the case of the web, a lot of the sites I find to be bad and annoying are the ones that are setting a sort of standard for the rest of the people - the insane information overload portal sites, that make everyone else out there believe that you need to have everything on the page at one time. People making and wanting websites are afraid of empty space. It's amazing. Finally there is a medium where the space is literally free (as opposed to print where each extra page of a printed piece is a jump in cost for paper and ink), and we feel that we can't deal with it - we have to fill up every corner of it right away.

What, would you say, is the most common mistake made by web designers today?

I guess it depends according to what criteria is something a mistake. But like I said in the above question, following a set standard, a bunch of web cliches is what I find most frustrating out there, but the clients are hard to fight on those issues because they believe in the standard.

Does television, as it exists now, have a future, or will WebTV or some derivative of it, rule the world?

I don't know, it's hard to tell. There are some people that are totally into it and others not at all. I'm not sure.

Funny Garbage

Funny Garbage designs large corporate projects, aswell as small, more experimental interfaces. Is it hard to switch between both 'modes'?

Not really - it depends. If anything, it's harder to go from a super creative job to a more strict, rigid one, I guess, but I think it's good exercise for a designer to have a possibility to do all types of work, even if you prefer one over the other.

Do you ever have to convince clients to go with a certain design?

Yes, and sometimes I fail. The most frustrating thing is when you cannot understand why they are making a certain decision because you can just see a mile away that its going to be awful but they won't be convinced, and you can't just say to them "You have to do it my way or I won't do it!". That's hard. But I guess that's just a part of working for clients - you have to pick your battles sometimes.

Funny Garbage is a very large company, for a design studio. Have you ever felt that this size can hinder creativity and flexibility?

Well, when you get to be a certain size there is definitely a need for more structure and organization than if you are small, otherwise things would turn into chaos. But as far as hindering creativity and flexibility that really depends on the type of jobs you take rather than how many people there are. I mean, it's definitely less loose and open than when it was 10 or 15 people, but energy is still there and the jobs that come in are great so I guess it's just a

bit of a compromise.

At Funny Garbage, do you always work in teams or are there projects you do all by yourself?

It really depends on the size of the project, bust most of the time we work in teams because even if the design is done by one person alone, the technical implementation is almost always done by someone else.

5!Conclusion

Problem/context

The advent of home and personal computers and, more recently, the Internet, has radically changed the lives of user interface designers. Where user interfaces were traditionally designed by engineers, since they were completely mechanical, this is now shifting towards graphic designers and others with similar skills. The fact that a personal computer has a monitor display as its most important feedback device, is an important factor in this. It is a flat surface which has to be organised and structured, and that's something graphic designers have been doing for quite some time.

That being said, the computer display is also a very new medium. It allows for animation and user interaction in a revolutionarily new way. It is dynamic in all respects and extremely flexible. And I believe this is where graphic designers experience the most problems. Most have never been trained in designing for extensive user interaction, and don't have a lot of experience with interactive designs. They experiment, develop new strategies and sometimes re-invent the wheel. Many forget that a lot can be learned from the decades of experience built up by more traditional, mechanical interface design.

Personal

The Web is one of the most exciting new media, but most graphic designers have a hate-love relationship with it. In print design, everything is rigid and very precise, the web is slightly more erratic. Different browsers, platforms, resolutions, modem speeds and colour depths ensure that one design can be displayed in a lot of different ways. It is my opinion that designers should keep this flexibility in mind, and design for it, instead of holding on to the unrealistic ideal of a rigid design that always looks the same.

6!References

[1] Johnson, Steven. Interface Culture. San Francisco: HarperEdge (HarperCollins), 1997

[2] Negroponte, Nicholas. Being Digital. Northbrook: Coronet, 1996

[3] Lesley Brown (editor). New Shorter Oxford English Dictionary. Great Britain: Oxford University Press, 1993

[4] Norman, Donald A. The Design of Everyday Things. New York: Basic Books, 1990

[5] Dick, Philip K. Do Androids Dream of Electric Sheep? Great Britain: Rapp & Whiting Ltd., 1969

[6] Johnson, Steven (editor). Flame Wars. Durham and London: Duke University Press, 1994

[7] I/O/D 4 (Webstalker) browser: http://bak.spc.org/iod/Disorder.html

[8] Total Annihilation: http://www.cavedog.com

[9] INTER!FACE site: http://www.katdesign.be/interface

[10] k10k site: http://www.k10k.net

[11] Fork Unstable media: http://www.fork.de

[12] Funny Garbage: http://www.funnygarbage.com

7!Appendices

7.1!Interview David Linderman (Fork [11])

What direction where you going in college?

>> art student. graphics were more available and better form of expression.

Did you ever think you were going to found a design firm in Germany?

>> no clue.

Why Germany?

>> why not? this place is funky in a twisted way. adventure.

On your website, you state that Fork specializes in new media and modern brand placement. Do you think there will be a time when the new media of today will be the only media, when print will be no more?

>> uh no. good design has always been about authorship as well as organization. i dont define myself in terms of the surface i work on.

Do you ever get frustrated by the vast amount of bad sites out there?

>> no. too many good ones.

Fork designs are often innovative or at least avoid the conservative approach. Do (or did) you ever have to convince clients to go 'the Fork way'? What are your tactics?

>> we do it our way. but we^re pretty upfront and intelligent about it. We are very very stubborn about details. =)

Humour is a very important aspect of Fork. Are you trying to be universal by being funny?

>> we are having fun.. or fucking with stuff we think needs to be challenged.

What can interface design learn from print design? What is the most underestimated achievement of print design?

>> interface and most print design are like totally different fields of study. associates only through graphic design, the use of type, image and visualized motion.
if designers from school are trying to categorize themselves as °print or °interactive, i think they are fairly short-sighted as to the whole spectrum of designed surfaces like tv, film, comics, sound, games, magazine, newspaper and identity work. i dont think i named them all.

Is graphic design art?

>> i think that in the same way an artist acquires a personality or propogates an idea through their work, so do designers. -important for both is that it is done passionately or eloquently. it its just a job then it is not art.
i think that most people dont understand what art is in the first place. art creates an itch. or expresses something that we can consume. Art necessarily implies consumption and is therefore related to commerce.

What are your 'great examples' in (design) history?

>> wwII

Is interface design graphic design or product design?

>> god. who cares?

Would you like to design the interface for, say, a VCR or a toaster? Why (not)?

>> hell no. if i did it would be art. =)

Have you ever read 'The Design of Everyday Things' by Donald A. Norman?

>> i think so. it must have been very boring. i don°t really care about shaver design or toasters.

How much has digital interface design have to do with product design?

>> how much do you think it has to do with it?

What guidelines do you use when designing an interface?

>> none. check parameters, identify source. design flow.

Is there a set procedure, or is it different every time?

>> obviously.

What is the most valuable book on interface design you have ever read?

>> the diamond age -a victorian primer for girls- by neil stephensen

Do you think the text interface is heading for a comeback?

>> did it ever go out of style?

Have you got an alternative for the desktop metaphor in current GUIs?

>> of course.

Are 3D metaphors useful in interface design, or are they just a useless waste of resources?

>> you can do whatever you want to. do you think amy franchisi from future farmers is a useless waste of resources?

Are all those different languages in Europe unnecessary barriers or a unique breeding ground for creativity?

>> language is beautiful. its even better when you understand it.

Do you think computer games like Quake, Tomb Raider, Dune 2(000), etc. are culturally valuable, or are they just the digital equivalent of blockbuster movies?

>> i think blockbuster movies is culturally valuable for about 14 billion people around the world.

How about on-line gaming?

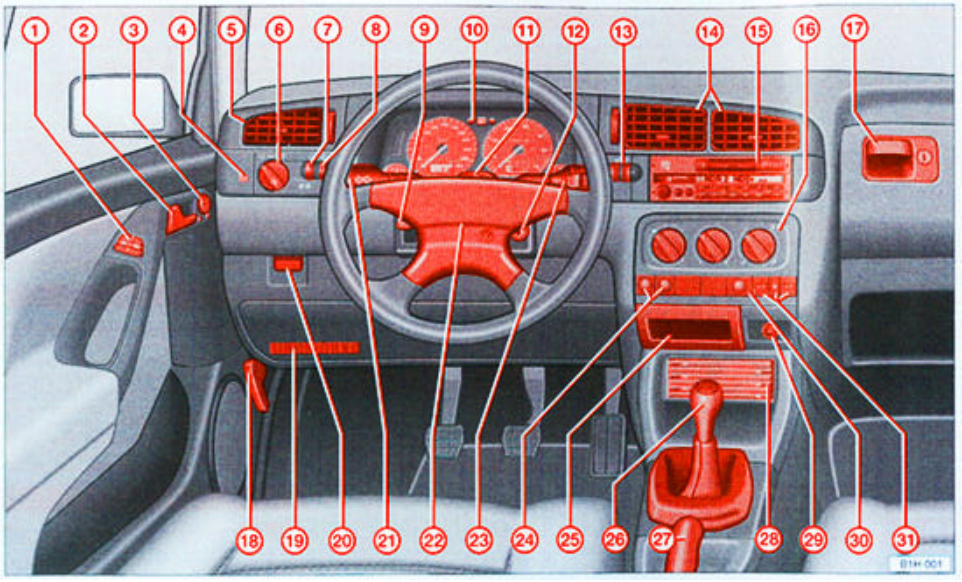
>> great.

Does television have a future?

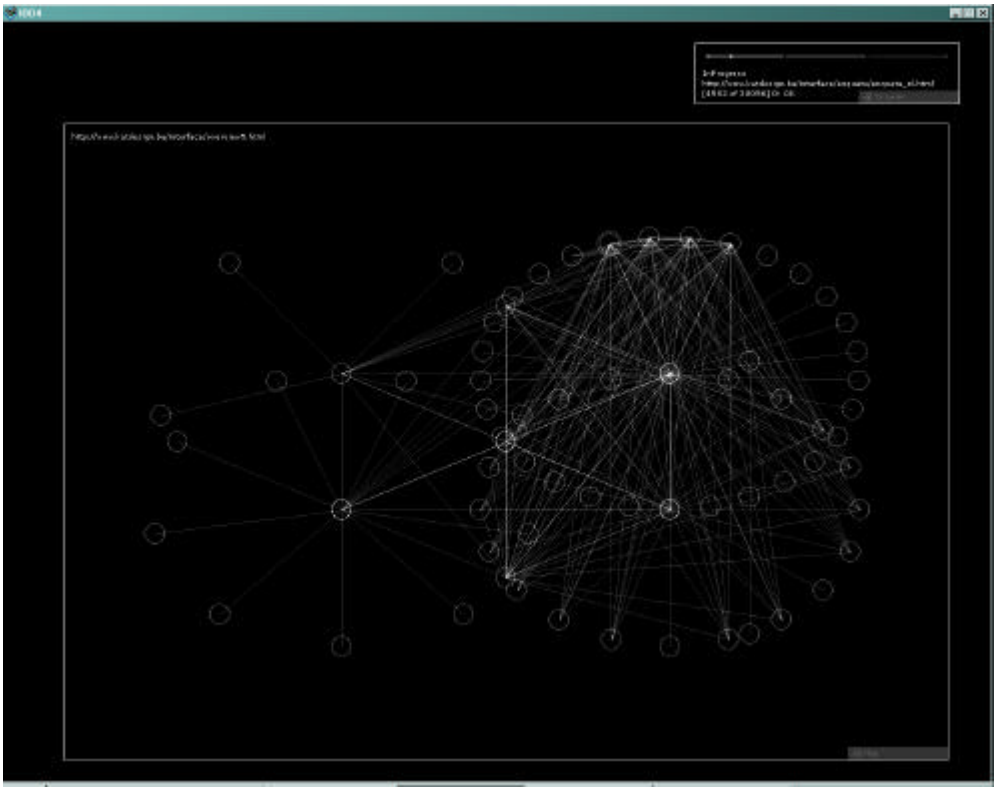
>> of course. on the net.

7.2!Illustrations

VW Golf 3



IOD browser



Total Annihilation

K10K