

Toys R Us

Introduction

`What you see is what you get` became a term in art critical writing in relation to American Minimal Art of the mid '60s to early '70s. This was particularly true in talking about the paintings of Frank Stella ("I'd like to keep the paint as good as it is in the can.") or the sculpture of Donald Judd (he described his work as being "like counting, 1, 2, 3, 4,...). The term referred to a certain denial of subjectivity and a rejection of the tenets and terms of the history of European Art. An idea of the facticity and inexpressiveness of the material world is involved, as well as an affirmation of that materiality. There is some overlap with certain American Pop Artists, notably Andy Warhol and Claes Oldenburg (British and European Pop Art tended to look towards America as a world of unattainable material abundance). In computing, the term 'what you see is what you get' refers to there being a correspondence between that which appears and is being worked on on a screen and the printed result. It also refers to something on the screen having the same intuitive look and feel as a piece of paper - that is something which can be edited, rewritten and altered without overtly cumbersome procedures or commands. The user interface, the point of contact between the person and the computer is, ideally, comprehensible and familiar. The overlap with the use of the same term in art criticism is in the idea of something being in the real world.

This essay looks at the origins of WYSIWYG, both as an outcome of the history of the development of the personal computer and in relation to the visionary ideas of Douglas Englebart. It also asks the question as to whether the apparent transparency is, in fact, real.

The Steam Engine & The Hobbyist.

In order for something to become useful one has to be able to use it. This is perhaps less obvious than it first appears, particularly in relation to the personal computer which has developed through a process that is perhaps more like evolution than design. As in any evolutionary process, it has been characterized by employing existing resources in new contexts.

My father had a model steam engine which he'd built for himself whilst an apprentice. It had a small boiler, heated by a methyated spirit burner, which let

steam into a pressure chamber. This in turn drove a piston connected to a small flywheel. The steam engine was a hobbyists toy, interesting (and very temperamental) but not in itself useful and was the result of the miniaturization of an existing design. The large mainframe computers of the fifties, in universities, government offices and large corporations, worked with vast amounts of data, were rather specific to their individual tasks and extremely specialist in their operation. The first small computers can perhaps be thought of as akin to my fathers model steam engine - miniaturized, hobbyist versions of a pre-existing model that demonstrate a means of operation but don't do very much. There is, however, a difference that can be explored by examining the concepts and different roles of hardware and software. In order for the steam engine to work, my father had to fuel the burner, fill the boiler, adjust the valves and provide initial momentum to the flywheel. He provided the elements of the operating system. Without his initial input and his adjustments during running, the thing would not work. In fact I think that a great deal of the pleasure to be got was in the sense of delicate and skilful handling of a recalcitrant system. This may well be true of all hobbyist activities (all male dominated ones at any rate), whether it's steam engines, old cars, WW II airplanes, crystal radio receivers or whatever. And the same is probably true of the first hobbyist attempts to build a model computer.

At the heart of any computational device are a series of switches that are either off or on. It doesn't matter whether these switches control the flow of water down sluices, beads on strings or electrons. The off/on opposition allows the binary representation of number and, by encoding, of letters or functions. Thus binary representation contains information and allows the construction of languages and of software written in these languages. The operating system, written in such a language, enables the computer to function. Without such an operating system the computer is like the steam engine without my father. Of course, bigger steam engines are more powerful, bigger computers are not necessarily so, it depends on the size of the switches. Representing a data stream by water sluices is large, cumbersome, slow and pretty limited; doing the same with thermionic valves is less so but still needs an enormous machine in order to achieve any significant computational power. The transistor, a switching device invented by Bill Shockley in the 1940's, has been remorselessly miniaturized. In 1959 Bob Noyce invented the integrated or printed circuit whereby electronic components are etched onto a prepared silicon surface by a photographic process. This led to ever larger numbers of transistors being deposited on silicon chips. Gordon Moore, co-founder in 1959 with Noyce of Intel, looked at the continuing

decrease in size of components and proposed Moores law. This states that the number of transistors that can be put onto the same size piece of silicon will double every eighteen months. That is, information processing capacity will double every eighteen months. This is an extraordinary growth in capacity and has held true through successive waves of technical and manufacturing innovation. This reduction in scale allows for the increasing complication of electronic sub-assembly. Robert Cringely, in his book *Accidental Empires*, is correct to point out that Silicon Valley is called Silicon Valley, not Computer Valley, and that it is the increasing sophistication of silicon chips that has permitted the development of the Personal Computer. It is this process of adaptation and increased complexity in components, building on immediately preceding models, that one thinks of as evolutionary. In 1971, Ted Hoff, also working at Intel, produced a microprocessor, a single chip containing most of the logic elements of a computer. This was too expensive and sophisticated for use in most industrial or mainframe applications, but there was a convergence between hobbyist or enthusiast interest in computing and manufacturing tendency. For members of computer clubs and amateur enthusiasts, this was the beginning of a process leading up the introduction in 1975 of the Altair 8800, the first self-assembly small computer. There were no obvious business or academic applications and, initially, no software to run. For an enthusiast however it was probably sufficient that, like my fathers steam engine, it worked (perhaps even more so if it was difficult or awkward). There are, however, two very important distinctions to be made. Miniaturizing a steam engine renders it effectively dysfunctional - the power output is insufficient to drive a can opener - there are no real world applications. For electronics, thanks to miniaturization of the components, power is not lost - a small computer can do something. This is where the second distinction becomes relevant - my father's steam engine has no internal software - he supplied it and was the operating system, with a little help from James Watt. Of necessity a computer needs software, an operating system. It has to be programmable and ipso facto, can also have applications available via programs. Software (either operating system or applications) can be copied and distributed. My father could learn how to run the steam engine and he could have taught us, but he was not distributable. These are key and linked points:

- (a) A small computer needs an operating system and can have applications.
- (b) Software can be distributed.

The January 1975 edition of Popular Electronics magazine featured the Altair 8800 on its cover. Reputedly, when Paul Allen showed the magazine to Bill Gates he realized both that the Altair would support a high level programming

language and, secondly, that this would be their future. This realization was a combination of (a) and (b). The first part was visionary, the second a business opportunity but it would be hard to separate the one from the other.

I have laboured the point because the absolute conviction that `this thing can do something` and the finding out what that `something` is seems to me to be absolutely integral to the history of the personal computer. As Cringely points out *'Mainframes weren't that impressive. There were no fancy high resolution graphics in the mainframe world - nothing that looked even as good as a television set. Right from the beginning, it was possible to draw pictures on an Apple II that were impossible to do on an IBM mainframe.'* (Accidental Empires p.76).

The Argument From Design.

Classically, evolution happens by accident, continual slight random mutations in the genome are occasionally positively reinforced (we don't know about the ones that are negatively reinforced, there are no survivors). The accumulation of these random mutations leads to adaptation and speciation although it is cardinally important that nothing comes from nothing. There is always only the existing genome available. For creationists this is not good enough, there is incredulity that the richness of life could come from random processes. The strongest argument they advance is the argument from design. There are two elements to this, one is that some organs are so complex that it is impossible to imagine them having a primitive state. The second is that the processes occurring at a molecular and cellular level are far too complex to have occurred randomly and that certain of those chemical processes are pre-prepared. That is to say the organism makes some components in advance of another process but which only make sense in relation to that anticipated process. Therefore, so the argument runs, although adaptation clearly takes place, there has to be some kind of designer. The most sophisticated version of this argument presents him/her/it as clever but a little bit bumbling, certainly not omnipotent and having no connection to ethical or moral issues. In the previous section, my point of view was largely evolutionist. In this next section it is largely creationist.

Doug Englebart was somewhat different from the other major figures in the development of the personal computer industry. Clearly at some fundamental level Bill Gates or Steve Wozniak had faith that the introduction of personal computers had the potential to change peoples lives - for example by changing

the way people worked or played. For a multitude of reasons they and many others wanted to be involved in what was developing. Doug Englebart was an academic, based at Stanford University's Menlo Park campus, and was somewhat of an idealist. He had a faith in science and a belief in the ability of technology to ameliorate the human condition. He thought that computers (in general) have a capacity to make people better and that this would be achieved through people being enabled to become more intelligent. These ideas significantly pre-date the development of the microprocessor or the beginnings of the personal computer industry. His first paper outlining the conceptual and research basis for his ideas, '*Augmenting Human Intellect: A Conceptual Framework*' is dated October 1962 and was prepared for the Air Force Office of Scientific Research. In places it reads more like science fiction than a conventional research paper, which is perhaps inevitable given that he is imagining the future. (The interplay between science fiction and developments in information science and technology is not insignificant - Microsoft first appears as a word in the work of the science fiction writer William Gibson, as does cyberspace). In the summary abstract to this extensive paper Englebart states '*One of the tools that showed the greatest immediate promise [as an augmentation device] is the computer which can be harnessed for direct on-line assistance, integrated with new concepts and methods.*' There is a faint echo of the film '*Forbidden Planet*' here, where one of the astronauts short circuits his brain by attaching himself to a brain amplification device left behind by the vanished superior civilization on Mars (and unleashing the id monster at the same time). Or, more believably sinister, of '*The Manchurian Candidate*' where a captured American airman, brain washed in Korea, becomes a walking time bomb. Or even of the rogue psychologist Wilhelm Reich's 'Organon boxes, reputed to amplify human sexual satisfaction (the film '*WR; Mysteries of the Organism*' was a cult hit at the end of the 60s). Englebart's 1968 paper, co-authored with William English and produced from the Stanford Research Institute at Menlo Park, is titled '*A Research Center for Augmenting Human Intellect.*' The paper describes the construction and use of an interactive computer network, accessed via individual work stations (the NLS system). The system described is linked to a mainframe time sharing computer (65K, 24 bit core), a 4.5 MB swapping drum and a 96 MB file storage disk. Twelve work stations were served from this centre. Each work station was equipped with a display monitor, an alphanumeric keyboard, a mouse and a five key handset. This is the first description of a computer mouse. The mouse controls an on screen pointer. The screen was bit-mapped, that is divided into a number of discrete sections (pixels) on a grid. Moving the mouse produced a

corresponding movement of the onscreen pointer, clicking on the mouse button activates that section of the screen - positions the cursor or sends a command. Any part of a document could be altered, cut, copied or otherwise edited by scrolling up or down, pointing and clicking. Output from this system was via the screen or an online printer. The central computer held files, both personal and shared. Documents or editorial aids could be retrieved either by entering commands or by using the mouse to activate links. Much of the basic structure is familiar, although the user has to have a clearer grasp of complex command languages than would be tolerated by a PC user. The paper also describes the supplementing of this system by a live feed video link which is displayed on the screen simultaneously with text. This is pretty interesting, like entering text over a web-cam feed. The striking feature of Englebart's paper is that he thinks across the interface. He is concerned with the organisation, access and display of the data contained within the memory of the computer AND he is equally concerned with ways in which people think. Thus his concern is to make the information system both useful and user friendly. The introduction to the paper begins by stating:

'2a In the Augmented Human Intellect (AHI) Research Center at Stanford Research Institute a group of researchers is developing an experimental laboratory around an interactive, multi-console computer-display system, and is working to learn the principles by which interactive computer aids can augment their intellectual capability.'

2b The research objective is to develop principles and techniques for designing an "augmentation system".

2b1 This includes concern not only for the technology of providing interactive computer service, but also for changes both in ways of conceptualizing, visualizing and organizing working material, and in procedures and methods for working individually and cooperatively.'

On the back cover of *'Accidental Empires'* the first of the three things that Robert Cringely identifies in the history of personal computing is *'(1) It all happened more or less by accident.'* Most of his account supports this view, in that contingent circumstances created conditions in which individuals saw opportunities, and that the very fact of making something which did not exist before is itself a discovery process (thus somewhat accidental). This is an argument for evolution. Doug Englebart provides one of the counterexamples, by looking at the interface he produces something which begins to look like the concept of a personal computer. This is design, not evolution.

The Word On The Street.

In 1970 Rank Xerox, then the world's leading manufacturer of office copying machines, began to fear that a future of paperless offices was a possibility. Palo Alto Research Center (PARC) was established to think about this for them. Staffed by a relatively small, but brilliant, group of researchers (about fifty), it was extraordinarily innovative. The model they were looking at was office equipment based, used by lightly trained office workers, not an academic research centre staffed by trained scientists. Thus though the NLS system, the interface that Englebart and his team had been developing at Stamford, was clearly a part of the equation, the computing environment itself was not.

The Xerox Alto, developed at PARC, was an individual workstation. Each Alto came with built in networking, black on white bit-mapped screen, a mouse and hard disk data storage. The networking capability was built around Ethernet. At PARC, along with developing the workstation, they also came up with the first laser printer. The printers were computers in themselves and very fast. As Cringely points out, with both the workstation and the printer operating at high speed, the two way transmission of data from one to the other became a major cause of delay - or worse data loss through collision. This was further compounded when several computers and printers were networked together. Ethernet is a solution to this problem, only permitting transmission of data down the line when the line is clear and waiting a random period if not. The increased efficiency meant that the transmission time for a page of text was reduced from minutes to seconds.

These are the elements of a useful personal computer, although it was not a Xerox that became, effectively, the first personal computer, but an Apple, the Apple II developed and built by Steve Jobs and Steve Wozniak. [The quoted awe-struck comment by Steve Jobs on visiting Xerox-PARC, 'Why don't you guys market this?' should perhaps have had an addendum 'Because if you don't, I will.']. Released in 1977 the Apple II came with colour graphics, pre-assembled and ready to use with a mouse, bit-mapped screen and floppy disk drive. Although expensive, it was affordable and usable - you could buy software that would run on the machine.

The programme that worked and that drove the personal computer out from the hobbyist and game-playing ghettos into the mainstream, was Visi-Calc. Developed by Dan Bricklin in 1977, Visi-Calc is a spreadsheet programme. In any business or accounting procedure a spreadsheet is an analytic tool for looking at trading performance or for seeing the effect of changing variables in a production

process or any other quantifiable operation. As a manual task it is long winded, the calculation of results as a result of a 'what if?' question both slow and tedious. With a virtual spreadsheet extending far beyond the screen of an office-based computer but in which the data are linked, the operation becomes beautifully simple, powerful and immensely productive. Used with the Apple IIs graphic interface - a WYSIWYG system - you 'got it'. Cringely makes the point that it was individuals buying computers and bringing them into the office that changed things, not policy decisions. The word on the street brought the product to the marketplace and funded a revolution. Something not dissimilar happened in 1707 when Abraham Derby and a young iron-founder called John Thomas hit on a new way to make cast iron pots. Previously cooking pots were expensive, hand made items, cherished as family heirlooms. By making them cheaper, and selling through markets and country fairs, Derby could afford to give up his job and set up his blast furnace at Coalbrookdale. It was individuals recognising something that made things easier and for which they were willing to pay that funded the Industrial Revolution.

The Matrix.

Two developmental processes, one evolutionary and one designed, came together in the mid 1970s to produce something which hadn't existed before. The spread and use of personal computers, desk-top publishing and all the rest of it have fundamentally changed the world. New ideas have been formed, new languages developed, new world views have been articulated. Different communities and forms of association have appeared linked across the net, united by shared interests - some of which are anti-social. Writing from deep within the publishing community, Conrad Taylor, a typographer and graphic designer asks '*What has "WYSIWYG" done to us?*' It's an interesting question. Mostly he is for it - life for the graphic designer has become easier, processes have speeded up. He does, however, also have concerns. One concern is the de-skilling of a community. The other and more telling concern is with the consequences of the severance of design from content. If we can all edit, design and publish, what role do graphic designers or typographers have other than as stylists (there is a Japanese programme that allows you to evolve your own font). But the understanding of a text can be enhanced by its typography and design - whether it is a poem or an instruction manual. And in some circumstances illegibility could cost you your life. Taylor's anxiety is about the absence of discussion amongst the writers of graphic software of the meta-issues involved. He realise

that 'What You See Is All You Get', and that there are things lost or ignored which should be addressed. Englebart's belief was that the computer augmented intelligence by making one conscious of the hierarchies and categories of information processing. Tyler is suggesting that the computer could also make you ignorant. William Burroughs, writing in the 60's, ('*Naked Lunch*' and '*The Soft Machine*') articulates a paranoia about the ways in which the language we use colonizes our brains like a virus, preventing us from becoming conscious of our reality. For Burroughs, only through disruption by using the device of the cut-up to fragment text can reality emerge. [The movie '*The Matrix*' has a similar thesis.]

The headline in The Guardians' UK online supplement of May 24 2001 reads 'Microsoft's plan for world domination'. Taylor's questioning of WYSIWYG is timely.

Note: This essay was written 15 years ago, in June 2001, before the advent of social networks and smartphones. Whilst evidently I was thinking that something was up, I cannot claim to have anticipated the dreadful truth. Two and a half months after I had written this piece a group of determined individuals, some of whom had learnt to fly on computer simulators, would seize control of four passenger aircraft and fly them into buildings. Toys R Us indeed. This is the first publication.

RICHARD DEACON, June 14 2016.

References

- <http://www.bootstrap.org/> - Home page for Douglas Engelbart's Bootstrap Institute, all his research papers can be found there as html documents.
- <http://www.ideography.co.uk/library/seybold/WYSIWYG.html> - For Conrad Taylor's thoughtful paper on WYSIWYG.
- <http://www.intel.com> - for the history and current preoccupations of the leading chip manufacturer.
- <http://www.sciam.com/interviews/moore/092297moore1.html> - an interview with Gordon Moore.
- <http://www.ruku.com/noyce.html> - a biography of Bob Noyce
- <http://www.stanford.edu/group/mmdd/SiliconGenesis/TedHoff/Hoff.html> - an interview with Ted Hoff.
- <http://www.microsoft.com/billgates/default.asp> - a biography and the collected articles of Bill Gates

<http://www.woz.org/> - Steve Wozniaks home page.

<http://news.cnet.com/news/0-1005-200-321228.html?st.ne.fd.mdh> - A collection of Steve Jobs comments and articles, with particular relation to Microsoft

<http://www.bricklin.com/> - Dan Bricklins home page.

Books:

Gregory Batcock - Minimal Art - A Critical Anthology - Questions to Stella and Judd.

Robert Cringely - Accidental Empires.

Out of the Cauldron - New Scientist # 2164; 28 December 1998