THE MACINTOSH COMPUTER: ARCHETYPAL CAPITALIST MACHINE?

William Bowles

+ **Looking Backward**

This essay was written in 1987 before the domination of Microsoft, at a time when I thought that the people running Apple computer had their act together. How wrong can you be! In other respects however, all of the ideas and concepts advanced in the essay have come to pass and in a most fundamental way. One further observation that I think is worth making about the process that led to the emergence of the Macintosh and indeed the PC itself is the role of serendipity, for it is true to say that none of it was planned, least of all by those in the corporate world - the last place to find innovation of virtually any kind. Yet the innovative period of any revolution in production is strictly proscribed. As soon as the results of innovation are incorporated into the mainstream of economic life, investment and return on investment become the determining factor. Today, the evolution of the Web epitomises this process, for what started as a tool for the academic world with which to share information, rapidly became an incubator for innovation. Once adopted by the corporate world its ability to evolve into the kind of 'general tool' of communication already incubating inside the Macintosh was - and I'm sorry to say still is being - severely hampered.

The Macintosh

The Macintosh computer represents a fundamentally new approach to the way machines interact with people. The philosophy inherent in the Macintosh is, for the vast bulk of working people, an augur of the direction that contemporary capitalism would like to see production relations take. At the same time the Macintosh also expresses many of the contradictions of capitalist relations, and so for this and other reasons the philosophy inherent in the Macintosh makes it an ideal example for study if we want to understand how science and technology are being used to wrest more and more surplus value from labour, whilst at the same time reducing the amount of control workers have over their own lives.

But what makes the Macintosh so different from other kinds of personal computer? Aside from the raw power of its main processor (equivalent to a room-sized machine of say 15 years ago), the operating system represents a radical departure from the essentially 'science'-based systems of such machines as the Apple II or large mainframes. By this, I mean that in order to interact with earlier forms of the computer, some considerable knowledge of the computer itself is necessary in the form of a computer 'language' that the user must first master before being able to make use of the machine's computing power.

In this sense, virtually all computers prior to machines such as the Macintosh represent the formative stages of the development of computer technology as it is expressed under capitalism.

The Rise of the Machine

There are many useful analogies available to us from the first era of machine development during the Industrial Revolution that can help give us a better understanding of what the Macintosh represents. During that period the development of industrial tools followed a somewhat similar path insofar as the kinds of techniques embedded in the machines were discrete reflections of specific human skills. In other words, the first machines were not 'general tools' in the sense, for example, that the modern numerically-controlled machine tool is.

What do we mean by a 'general tool'? The process of transferring 'skills' from human to machine is essentially done in stages. The first stage involves a craftsperson building a prototype machine which consists of the craftperson's brain/hand skill being broken down into its component parts and each process being assigned to a specific element of the machine. A good example of this process would be the metal turning lathe. The lathe itself consists of several elements. The first is motive power (this emulates muscle power), the second is measurement (which embodies generational experience of the physical environment), the third is the process of transforming the raw material into a useful product (this represents the actual tool using capability of the craftsperson).

In the development of the lathe the process of synthesising these elements may take place as follows: Power in the form of rotational motion must first be transferred to the chuck (the chuck represents a discrete, artificial form of hand for holding the work in place); a method of transferring measurements to the material to be worked on, and finally the cutting tool itself and its interaction with the material to be transformed. In the initial development of the lathe the cutting tool had to be brought in contact with the raw material manually and the measurements for how much metal to remove were also done manually (with calipers and later micrometers). But as more and more expertise became embedded in the lathe, the operator could simply set vernier dials on the machine and eventually even the act of cutting was automated via a screw-driven feed connected both to the cutting tool and the rotating chuck (it is here that we begin to see the emergence of feedback systems of some complexity, eg the lathe has to 'know' when to stop cutting). The act of developing the lathe then is a two part process: first, the job is broken down into its discrete parts which are then 're-united' via the interaction of various forms of generalised feedback (as in the above example).

The end-product of this process is the emergence of what I refer to as a 'general tool'; that is, a tool whose basic principles embody not only the specific skills of the craftsperson, but more importantly, the 'skills' are embedded in the lathe in such a way as to 'mask' not only the craft origins of the process in terms of the skills needed by the operator to use the machine, but more importantly, the tasks are standardised via specific elements incorporated into the operating system of the lathe. This is done by 'pre-setting' the lathe as much as possible for a single task or series of tasks. In this way the operator need only know, firstly, how to load the lathe with the raw material, then how to turn it on, and finally, how to start and stop the sequence of operations that results in the end-product, the finished article.

The account above is an accurate, if abbreviated, description of the nature of the technical transformation brought about by the advent of the industrial system. That machine tools are now many orders of magnitude beyond the originals in complexity and versatility does not alter the fundamental concepts that they all utilise. Indeed, until the advent of the computer, basic machine tool design has not fundamentally changed since the 19th century, and even with the addition of computer control, such tools still use the same basic principles.

Microchip Meets Machine Tool

The process started by the industrial revolution has reached a pinnacle in the form of the computer, for the computer is essentially the 'end-product' of industrialism in the sense that it acts as a unifier of discrete, industrial processes in the same way as the lathe did for craft processes. At this point we could ask a hypothetical question about the nature of the 'end-product' of computer development: what form would a computer take, if it too, were to go through the same process of rationalisation as the machine tool has? By this I mean is there an equivalent computer version of the 'general tool' for people with a 'generalised' education? The implications of such developments are, in my opinion, as revolutionary as the development of the machine tool.

The slogan 'the computer for the rest of us' is extremely misleading (and probably has a lot to do with why Apple dropped it), but buried in the idea is a kernel of truth, for indeed if - and it's a big if - some kind of standard for using computers were to be adopted by all computer makers, then the promise in the slogan could have read: 'The Macintosh, universal tool, the computer for all of us'. There are heavy ironies on many levels, which are not only interesting to pursue simply as ideas, but also relevant to the direction society is, or could be taking.

The market economy as it is now constituted presents many obstacles to the adoption of a general tool, such as I have described above. A corporation like IBM of course has the clout to try and force its standard on everyone else, but in fact it is more likely to be the State in the form of the IRS (Internal Revenue Service) or some other large bureaucracy that decides what the standard should be (simply by virtue of sheer numbers bought and the need for a common protocol of communication).

In a sense the Macintosh operating system is a form of 'State Socialism' in that its effective operation depends on absolute adherence to what are euphemistically known as 'the Macintosh guidelines'. The user interacts with the operating system via a command structure that is the same regardless of the application. Now while I have no fundamental quarrel with this approach, for it to work effectively everything must be 'in' there - that is to say, every possible contingency must be planned for.

This, after all, is what the Toolbox is all about (note the description of the sub-routines or 'mini-programs' as Tools). It parallels very closely the kinds of standards developed in machine tools, for as with the industrial tool, the operating system effectively 'masks' the 'real' operation of the computer by interposing itself between the user and the Central Processing Unit. The operating system then is itself the 'general tool' that I referred to earlier. This approach has other drawbacks for it means that all applications written for the machine must conform to the rules or 'guidelines'. Apple even suggest that the application be sent to them for 'clearance'. What this means is that they check it to see that it doesn't 'collide' with some other application that may be co-resident with it, or parts of the operating system itself (which in turn may have already been 'harmonised' for some other application already loaded into the machine). This further suggests that one fundamental error was made in the design of the operating system, namely that it is not a true multi-tasking machine, for a multi-tasking machine is inherently designed to accommodate different applications concurrently.

Even this modest scratching of the surface of the Macintosh reveals the incredible complexity of such a general tool, not so much because it has so many disparate functions but because they can be combined in a never-ending array of permutations. It also illustrates just how much expertise and labour is actually embedded in the machine's operating system.

A Further Look at the 'User Interface'

If one looks at the commands and functions built into the Macintosh, we see that the majority of them emulate basic communications functions like drawing, positioning and pointing at objects (the so-called Quick-Draw and associated routines), as well as font manipulation. In the background of course, the operating system is active continuously, monitoring the keyboard, disk drive and so on.

The Macintosh then is a multi-purpose graphics-based computer that has a built in set of 'tools' for manipulating the Central Processing Unit (as well as the auxiliary processors) which interacts with the user through a set of choices represented by words or images. The icons are simply generalised signs for objects or functions (the use of language independent images for universal communication is well known to us through for example, international traffic signs). For example, the 'undo typing' command in Macwrite doesn't know what typing it's undoing, it just does it. The command itself is a 'generic' term, which in turn acts on certain 'assumptions' made about the command.

But with all the talk of icons on the Macintosh, it is the Word that has become the real icon, in that by generalising English words, the operating system has been 'colloquialised', or opened up to the speech of everyday interaction. In other words, 'any fool can use it'. And it is a fact that the Macintosh really is easy to use (as well as being extremely frustrating at times); anybody can master the basic system in a very short time.

The Dictatorship of the Machine

One of the chief objections to such machines as the Macintosh is the fact that it is essentially a 'black box'. By this, I mean that the inner workings of the machine are 'sealed off' from external access by the user interface. The 'shell' erected around the operating system (the menus and commands), although extremely comprehensive and easy to use, deny access beyond a certain 'depth'. By contrast, machines such as the Apple II allow penetration by any user to the basic binary system of operation that the central processor uses. Not only that, a computer such as the Apple II is physically open to anyone, with direct access to the main processor, enabling anyone with sufficient knowledge to 'tinker' with the workings of the machine itself.

By contrast many people have raised serious objections to the 'black box' approach used by machines such as the Macintosh, arguing that by making the machine into a closed system it not only reduces the range of choices open to the user, but perhaps more importantly it encourages a particular attitude towards machines in general by mystifying the processes involved, which in turn leads to a state of unquestioning acceptance of the supremacy of technology. This is of course a process that began with the industrial revolution.

A comparison between products of the first industrial revolution and the revolution we are in the middle of, illustrates the difference. The first products of the machine age were essentially simplified versions of the craft original (simplified because the machines themselves still reflected on the one hand their craft origins, and on the other because they were still relatively crude machines their powers of 'resolution' were limited). What this meant was that the products of the early machines were still accessible to the craft worker, they could be repaired or modified by hand, but perhaps more important than that, the processes embedded in the products were comprehensible to the worker.

Inevitably as the techniques used in production got more and more complex and the sophistication of the machines grew, so too the products became more and more inaccessible to the ordinary individual. In this sense then, the Macintosh reflects the general trend of industrial production to further alienate the worker from the processes she is involved in.

There are obviously a variety of forces at work that result in this development which reflect on the one hand, the nature of productive relations (increasing complexity), and on the other, the drive to increase profits (which in turn has an important effect on such things as complexity, repair versus replacement). It obviously benefits the manufacturer to replace rather than repair a product (the tag, 'no user serviceable parts inside' is by now well known to us). The issue is however more complex and reflects a much larger problem, that of the relationship between consumer and producer, which in turn is predicated on the level of education.

Elsewhere in this essay I mentioned 'general education' as a reflection of the generalising effect of industrial production on the labour process. The specialisation necessary for modern science-based production methods is predicated on the existence of a strata of the workforce who possess unique knowledge of the processes involved. This technocratic 'caste' is indispensable to modern productive forces, but even this highly trained segment of the workforce is under threat from developments in the field of so-called 'expert systems and Artificial Intelligence'.

Hoisted by Its Own Petard!

Driven by the necessity of maximising profits, yet hounded by the inherent contradictions of ever more efficient production processes, capitalism has sought to resolve the conflict by on the one hand, eliminating human labour as much as possible from the process of production, and on the other, by binding it as closely as possible to the organisation and nature of production. This has been achieved through a series of scientific, technical and political revolutions.

But ever more efficient production eventually lowers profits, this is the irony of industrial capitalism. Once you have maximised the efficiency of production there is no place else to go! If wages have been held to their lowest, and you are using the most efficient machines (more efficient than any of your competitors) you will eventually find that production exceeds consumption.

Each round of technical advances has heightened the contradiction, by making production cheaper and cheaper, which means that in order to make a profit, you have to squeeze more out of the consumer, who is also a producer (or at least some of them!). The time lag between the introduction of a new technology and its eventual absorbtion by society grows ever shorter. Hence technical change is forced on us with greater frequency. Eventually however, it must 'bottom out', as there is a finite limit to the amount of production the world economy can absorb, at least as it is presently set up.

Revolutions in Production

Each revolution of production under capitalism has been based on the introduction of a new, key invention or process. Depending on where in history you want to start from (I like to 'start' from the Renaissance, or about 500 years ago), the 'progress' of the development of machine technology can be traced by the advent of each new technology and its effect on society. In the 19th century, first the canal, then the railway. In the 20th century, first the internal combustion engine, followed by the airplane, and finally computers, have in turn formed the basis for a revolution in production.

For example, the chronograph can be seen as a key invention, which in turn stimulated and/or created the right conditions for other, connected kinds of inventions and processes. But accurate timekeeping was the result of the necessity to bind together an empire, for without it accurate navigation and hence mapping was impossible. Greenwich Mean Time is one obvious 'general tool' to emerge as a result of that event, or 'general time'; a fixed standard whereby no matter where you were in your empire, you knew how quickly you could move your resources from one location to another. The ‘spinoff’ from the chronograph was amongst other things, an increase in the accuracy of measuring tools. This was prompted by the need for precisely made cogs and other moving parts. This in turn meant that the tools needed to turn out such devices had to be more accurate, which in turn prompted more accurate devices for making tools...

Homogenisation of Knowledge

As with the invention of ‘general’ time, which was the culmination of a long historical process, each wave of innovation has eventually arrived at the point whereby general principles and standards have been extracted. Standardised units of measurement (the decimal system, electric voltages, screw thread dimensions, etc.) are the end product of many millennia of observation and practice.

Taylorism, for example, does for the actual integration of the production process, what standard units of measurement does for the machine tool itself. The invention of the telephone initiated the process of the standardisation of communications protocols. Ultimately then, it would follow that the introduction of computers into production and distribution would eventually arrive at the same destination, that of standardisation and the extraction of general principles of use. General principles would be laid down about, for example, the way computers relate to production processes. We already see such things in the field of electronic communications, but the process is of course, fragmented and uneven in its development and application.

The General Tool

What they all hold in common though, is that each process is eventually so thoroughly assimilated by society, as to become a part of the ‘general knowledge’ of society (much in the same way as everyone knowing how to drive a car).

It is interesting to note that Apple has, so far successfully, squashed all attempts to imitate its user interface (the so-called desktop, pulldown menus, etc.), threatening to sue any company that comes close to imitating the 'look and feel' of the Macintosh environment. It is tempting to speculate about what kind of long term view Apple have of the development of the computer/human environment (for good or bad). It would appear that Apple have recognised the necessity for a 'universal' means of accessing the computer. If, in one form or another, a set, standardised way of accessing computers can be established that enables the 'de-skilled' and 'unskilled' to access computers and the dead labour they contain, the complex problem of maintaining society can be handled without resort to educating everyone to the level of the university.

But for this to happen, for a critical period of time, one system must dominate! This is obviously what Apple are banking on happening. AT&T has done it in telecommunications. IBM has already done it in the 'business' environment, but that is the land of the Nabobs, we are talking about the domain of Burger King! As office automation accelerates, and virtually all forms of commercial interactions are 'standardised', the problem of utilising a deliberately under-educated workforce to handle extremely complex tasks, becomes a 'manageable' one. The standardised interface of the Macintosh lends itself well to dissolving the difference between 'factory' and 'office' work. The old, artificial hierarchy of blue and white collar work is on its way out, to be replaced by the generic, general service worker, who has enough skills to work a slick automated terminal like the Macintosh, and dispose of the output in some way (i.e. post it, stuff it or shred it), but a person who has no control over the work being done! For proof of this, we need look no further than the cash register of a typical fast food chain. The only numbers you see are the final bill! All the cashier need do is punch a button marked 'cheeseburger', or 'coke'; the built in processor handles all the addition and taxes.

The Macintosh that I sit writing this on, runs in a similar way to the fast food cash register in that, in order for example to change the font that I am using, all I need do is move the cursor to the 'button' marked font, and select one! The old way would mean knowing a set of commands that would load a different font into memory, and then only when the document was printed. They might be, 'ESC E-56, ESC-CTRL L', and further, they would have to be inserted in the text at precisely the right point and then turned off at the appropriate point by yet another set of commands.

User Friendly?

If the technical/professional elite are to maintain the system, they must make it as simple as possible to operate. By embedding the maximum number of possible states inside the code of the machine, it is possible to account for most of the situations likely to be encountered. In effect, all you need is the ability to read and follow instructions. As we saw above, no knowledge of a complex command language is necessary to make the computer do different things, the computer itself already contains all the necessary linked sets of instructions. If it goes wrong, or you do something wrong, monitors will spot it and a supervisor will be dispatched, no big deal. Each cog in the complex machine holds no indispensable power or leverage. Notice how the 'toolbox' that the Macintosh contains, parallels the synthesis of general sets of knowledge that may be accessed and comprehended by all! What we are seeing then, is an exact duplication of the first industrial revolution where craft skills were stolen and locked into the industrial machine, then perfected to the point whereby general principles could be extracted and applied to ever more sophisticated machines, each in turn, requiring less and less skill (and labour) to operate!

The languages that computers use reflect this process, for the first languages were specialised tools of mathematics and logic (again reflecting the 'craft' origins of computers), but arcane and abstruse, understood only by the select 'few'. Further, the very nature of the specialised origins of computers has led to a mystification of the processes, leading to the common misapprehension that computers are complicated, 'devilish' devices that only 'hackers' and 'eggheads' can comprehend. The Macintosh breaks with that tradition, at least in one sense. It is also, paradoxically, a logical extension of the same process! But it is the general nature of the principles embodied in computers, that makes them 'all things to all people'. It is this apparently contradictory nature of the computer that makes it so difficult to deal with. The computer is inherently a two-edged sword, unlike the factory, yet very much a part of it. Uncannily 'human', it is nevertheless seen as the ultimate in 'inhumanity'.

Conclusion

The Macintosh is very much a creature of two worlds. On the one hand, it represents the highest level of collective labour currently possible. By this, I mean that only the most integrated form of collective work could have produced such a device, utilising virtually every discipline available to us. The 'toolbox' routines represent the distillation of literally thousands of years of collective experience.

On the other hand, the computer is also an archetypal device, like the assembly line, except that it is diffused throughout the fabric of society. It is the precursor of the 'general-general' tool, a tool which will either enslave us or take an active part in our liberation. For the end product actually is the synthesis of the living, collective labour process that created it; this is one of the reasons why the conflicts raised by its existence are so intense. This is also precisely the reason why it makes such an interesting object of investigation. The key originators of the Macintosh interface, Alan Kay and R. Buckminster Fuller had a very clear picture of what they wanted it do, and how it should do it. Called the Dynabook, it was to be a paperback sized version of the Macintosh, battery powered with a complete 'toolbox' contained within it, all designed to be the literal extensions of the literate people who would use it. They saw the Dynabook as a universal tool, enabling people to communicate with each other using the collective skills embedded in the ROM chips. Add to this, the access afforded to databases of collective knowledge and you have not so much a technology but a philosophy of technology. You might call such a vision 'idealist', on the other hand the alternatives are far worse. For capital sees such tools as a means of extracting more and more surplus value from our labour. The very people who make the corporate decisions about the direction society should take, are also the same people who would delegate the role of starting nuclear war to computers! They care very little about the impact of computers and automation on life. They would entomb ALL living labour in machines had they the power!

In very many ways, the computer, especially in its Macintosh form, also represents the very antithesis of capitalism - for in spite of the fact that it represents the forefront of capitalist innovation, it also represents the very highest level of socialised labour currently possible. Not only that, but in order to extract the maximum advantage from such technology, private ownership actually gets in the way, unless that is, there is to be one computer company, one telecommunications company, and one manufacturing company! All this tells us is that computers and automation are an inevitable end product of monopoly capitalism, which would remove all competition from our so-called 'free enterprise' system, of which Apple Corp. is so much a part. The universal tool of which the Macintosh is the precursor, has the potential to open up knowledge and hence control to all people; that is why I can regard such a tool as an extension of the intellect, and someone else might be enslaved by its simplistic, collective 'mind'. More's the pity that for most of us, such potentially liberating tools will be used against us, making them objects of fear, and in the process imbuing them with almost mystical abilities as they apparently mimic aspects of human behaviour. But like any window, the Macintosh window can be a view from a prison cell or open on to a new world waiting to be explored.

↺

Brooklyn, NY (October 1987)

Looking Forward

Firstly, some further comments on some of the observations I made about the Macintosh back in 1987. I incorrectly assumed that it would be the Mac operating system that would become the standard for the new 'general-general tool' for the 'Information Age' - the Macintosh computer. The reasons for this are essentially three-fold: Firstly, Apple decided to make the operating system proprietary, thus restricting its spread to the key people who could have made it ubiquitous, the programmer and applications developer. Second, as I said in the original essay, not making the operating system multi-tasking limited its application in the business and scientific world, an essential prerequisite for its general adoption. Third, who could have predicted that Microsoft would essentially 'steal' the Graphical User Interface from Apple and given Microsoft's association with what was then the biggest computer manufacturer in the world, IBM, dominate the market place (Steve Jobs' three major - and almost fatal for Apple - business errors). This said, everything else I touched on has remained true, including the adoption of a standard for computer/human interaction - the 'pull-down menu' with its 'Desktop' metaphor, 'File', 'Edit', 'View', 'Tools' and so forth, more commonly called the GUI. I also touched upon the emergence of a communications standard that governments and business needed to adopt and the emergence of XML (Extensible Markup Language) conforms to this view. Most importantly, XML is, like its parent, HTML, a product of the public domain. In fact, every key innovation aside from the GUI has been created and remains in the public domain.

Eventually Apple introduced OSX, its own version of a multi-tasking OS, but perhaps it might have made more sense to have adapted a version of Linux, the free UNIX OS if it really wanted to assure itself a future. Once more, the contradictions of private ownership of intellectual property in an environment that *demands* open, public domain systems is exposed. Imagine a world where the metric system was proprietary or where the Internet protocols TCP/IP were proprietary? There are two mutually exclusive processes at work here: on the one hand, the interconnected world of the computer (now an indispensable part of virtually everything we do) demands open, non-proprietary standards and on the other, private ownership militates against sharing the profits.

Yet without a mutual, information-sharing environment, there would have been no personal computer in the first place ('All Information is Free!' was the motto of the early pioneers of the PC). This is the paradox of the computer revolution, that illustrates the contradiction of the emergence of the computer within the capitalist domain. For everything that makes the computer possible also creates the conditions for an entirely new kind of economy, that is based upon an interdependent culture of sharing.

As I wrote in the original essay, the key elements of the Mac are its 'Toolbox' routines, for this is the real genius of the Mac design. Indeed, elsewhere, I've drawn a comparison between the 'Toolbox' and DNA, insofar as the ROM chips in which all the thousands of routines are stored also contains much of the now 'redundant' code from earlier versions of the computer's operating system. Thus the Mac's operating system was (and is) possibly the first step on the long road toward the ultimate emulation of nature, conscious (possibly even self-conscious) 'life'.

The Macintosh 'Toolbox' is literally the amalgamation of thousands of generations of experience and understanding of the natural world in the form of equations, as well as the fusion of thousands of workers' creative efforts, in what was the first expression of a collectivised (socialised) labour process of a new kind. If the first industrial revolution saw the embedding (or theft) of experience plus the hand-eye coordination of the craftsperson in the machine tool, then the Mac represented the next (possibly final) stage with the capturing of living, intellectual effort in a machine, the stuff that underpins everything we do, our brains. Thus the uniting of the processes that underpin the principles embodied in the machine tool and in the computer was a logical next step. This is why I call it a 'general-general tool'. But is the brain to be commodified too? After all, the other revolution in our age, genetics, has already commodified reproduction.

The 'Toolbox' then, contains a vast array of equations covering all the basic elements for manipulating ideas in a variety of forms: visual, language, time, dimensions, mathematics and so forth. Once packaged and incorporated into a machine, it enables any literate person to access most of the basic knowledge acquired up until this point in time by the human species. Admittedly not all of it by any means, hence the need for additional programmes that actually utilise the Toolbox's routines. However, the programme 'Mathematica' contains virtually every equation used in mathematics, engineering, physics and so on, which when loaded into the Mac, gets very close to the creation of a portable 'container' of all the key knowledge that enables us to do work of all kinds. It should be noted that we are still in the very early stages of the process of embedding intellectual labour in machines and just as with the early expressions of the first industrial revolution, the process is still relatively crude. Hence as I pointed out in the original essay, its limitations are, and will remain, as with all analogues of human intellectual effort, finite.

What prompted me to write the original essay? When I wrote the essay back in 1987, I had since about 1979 started to explore the implications of 'computers and capitalism' and for a period of about two years spent time trying to figure out firstly, what exactly were the questions that needed to be asked? A gut feeling informed me that something very fundamental was happening that had implications for all of us but defining it proved difficult. A chance meeting guided me to the New York Marxist School in downtown Manhattan on West 19th Street where I was able to facilitate a weekly workshop called appropriately, 'Computers and Capitalism' where a small group of us explored the themes thrown up by the emergence of the computer, some of which ended up in the essay reproduced here. One of the descriptions that emerged I named 'Information Capital' (later to become known as Intellectual Capital) and it was clear even then that the computer had very fundamental implications not only for the future shape and direction of capitalism but for the very nature of work and what is now called the Cultural Production industry. Additionally and importantly, it has implications for the very nature and form of education, implications that have yet to be taken onboard by an Establishment still locked firmly in the 19th century world of the factory.

But oddly, or perhaps not, it was the Bulletin Board System (or BBS) and FidoNet[1] that really triggered my interest in the potential for the PC to transform the world, and for eight years I ran a BBS called *New York On-Line* on my first Macintosh in 1984. NYOL ran out of my loft in Brooklyn, networking news and information to the planet over the Fidonet network, the precursor to the Web. Now I'm not a 'techie' or even much of a programmer (I don't have the endless patience needed to debug code) but I know how it all works and how all the pieces fit together, essential if one is to comprehend the enormity of the IT (or production) revolution. For enormous as the impact of the first industrial revolution was, the advent of the Macintosh revealed the *ubiquity* of its operating system and the power inherent in its 'Toolbox' routines.

A little earlier I had discovered the online world of Compuserve[2] and what was then called CB Radio (what we now call Chat Rooms) and I made my first creative foray in the world of the computer when I wrote a play for the CB environment appropriately called *online.pla* about a future world where all social interactions took place online overseen by an online police force. Hence my initial forays into the world of the computer were not technical but creative and political.

Perhaps it was my training at art school and my involvement with kinetic art that enabled me to recognise the fundamentally different nature of the Mac as a 'general-general' tool and also its beauty as a seamless fusion of (ergonomic) design and function, so good in fact, that it has not been improved on (merely degraded as in the Microsoft copy). Let me put it this way, the computer brought together four passions of mine: political economy, the arts/culture, history and technology.

A little later, I was to read David Noble's groundbreaking *Forces of Production* (1986) about the rise of the numerically controlled machine tool.[3] A little earlier

I had already come across Stafford Beer's prescient work with the short-lived Allende government in Chile in 1971 and the creation of a real-time cybernetic model of the economy[4] - one small expression of which is Supply Chains.[5] The idea of 'cybernetic socialism' took root in my mind even though I didn't yet possess a full comprehension of what it would take. Call it a premonition if you like. But even before then whilst still at art school I'd been able to work with R. Buckminster Fuller on geodesic domes and had come across Fuller and Alan Kay's precursor to the Macintosh, the Dynabook that I referred to in the original essay.[6] And in a classic example of Fuller's 'Synergetics', the sum proved to be greater than its parts.

It should be remembered that back then, the idea that knowledge in its software form as a 'commodity' raised the hackles of the traditional Left - for how could a commodity in its traditional Marxist form not be consumed? Indeed, the software programme could lose and gain value in a way that *appeared* to contradict many of the basic tenets of Marxist notions of value. A key phrase comes to mind that I believe sums up the apparent contradiction with Marx's definition of value:
'If the whole class of wage-workers were to be abolished owing to machinery, how dreadful that would be for capital which, without wage labour, ceases to be capital.' (1891)

Of course the total abolition of wage labour is simply not possible but within the developed economies if a *sufficient percentage* of key production is supplanted by machines, then the issue of the value and role of capital is seriously undermined. An example of this process is to be found in the telecommunications sector where the automation of the telephone network resulted in the classic capitalist dilemma of the falling rate of profit due in no small part to the massive reduction of human labour in the production process. Indeed, the cost of making a telephone call (in the US) fell as close to zero as can be calculated and as a result, the source of surplus value vapourised as price competition meant that there was a limit on what could be charged for a voice call and of course, surplus value (profit) cannot be extracted from machines alone.

The answer? AT&T realised that unless it took possession of the *content* that the network distributed, it faced a rocky future. As the distribution of *data* overtook that of *conversations*, it became obvious that it needed to own the data and thus it embarked on a process of acquiring the content so that it possessed not only the means of distribution but also the product (this process was also accelerated by mergers and acquisitions). The problem of course is that information as a *commodity* cannot be consumed. Moreover, its value is dependent on several factors not normally associated with traditional products. When it is incorporated into an actual physical product, for example as a computer program in a washing machine or as a movie or a piece of music or a software program the information itself cannot be 'consumed' and hence replaced, thus the increasing emphasis on the ownership of intellectual property. Of course continual 'innovation' of software and hardware drives the IT sector but as with the first industrial revolution, companies have no alternative but to buy the latest innovations - to stay ahead of the competition or go bust.[7]

The rise of intellectual capital brought to the fore the issue of *use value* and increasingly the role of *time* as a value. So, for example, the concept of 'freezing' and 're-freezing' the same content but in a different form, thus adding value to the same production over and over again has taken centre stage. I first came across this when developing the first online product for Times Media Limited in South Africa back in 1994, an electronic version of an existing print product. Essentially the product had already been paid for and profit extracted through advertising revenues on the print version, hence once the relatively small investment in developing pre-press to Web tools had been paid for, it was, as the CEO commented, 'a license to print money'.

The other fundamental process that I believe underpinned the Macintosh revolution was the realisation that in order to bring together all the elements necessary to make the 'general-general tool' a reality, it was necessary to step outside the boundaries of the Victorian concept of specialisation - if you like a 'return' to the idea of a Renaissance Man (Buckminster Fuller is another example of this and originally trained as a naval architect, designing sea-going 'Spaceship Earths'). And the team that developed the Macintosh epitomised this approach (it included fourteen typeface designers!). Underlying this was another, even more fundamental, and in my opinion, revolutionary idea that I had actually come across as a teenager in the work of Professor J.D. Bernal in his epic five-volume work *Science in History* (1971; sadly now out of print). Bernal, who changed professions several times during his life, advocated an entirely different approach to education and even though computers didn't even exist when he wrote, his ideas emerged (in theory if not in reality) in the concept of the 'Knowledge-based Economy'. Bernal advocated the idea of a 'general' education based upon five fundamental areas of knowledge that formed the basis for all specialisations: 1. History; 2. Culture; 3. Language; 4. Biology; 5. Mathematics.

He felt that if an individual possessed a basic comprehension of these five areas of knowledge then they could pursue virtually any discipline and cited his own life as a living example. To fully appreciate this idea, consider the computer as a 'mediator' of work, an approach embodied in the idea of the Knowledge Worker. The Mac as a general tool epitomises this, as it enables someone to utilise its built-in knowledge base and apply it to their specific area of work. No longer do you need to be a programmer or an 'expert' with knowledge of IT, what is critical is an understanding of how the computer facilitates and mediates the specific skills and experience of an individual. However, in order to make this a reality an entirely new approach to education is needed, for the universal adoption of the computer puts *thinking* rather than *learning* at the centre of the educational process. If the idea of the Knowledge Worker is to be made a reality then it requires that we recognise that Intellectual Capital is not the proprietary ownership of ideas but the creative mind unleashed.

↺

London (November 2004).

NOTES:

1. Fidonet originated like virtually all the innovations in the world of the computer with a group of programmers in 1984 as a free and open method of sharing information (the same year I started my BBS) and is still in use to this day. A Google search yields 606,000 pages on Fidonet - <http://www.fidonet.org/> for more information. Like the Web, each connection or Node in the Fidonet network has a unique identifier (mine was 278/607). This is from Randy Bush's brief history of Fidonet and is indicative of the philosophy of the founders:
'Tom Jennings intended FidoNet to be a cooperative anarchy to provide minimal-cost public access to electronic mail. Two very basic features of FidoNet encourage this. Every node is self-sufficient, needing no support from other nodes to operate. But more significant is that the nodelist contains the modem telephone number of all nodes, allowing any node to communicate with any other node without the aid or consent of technical or political groups at any level. This is in strong contrast to the uucp network, BITNET, and the Internet. [my emphasis]'
Randy Bush, 'FidoNet: Technology, Use, Tools, and History' <http://www.fidonet.org/inet92_Randy_Bush.txt>

2. Compuserve grew out of the national computer network that serviced HR Block's tax offices and was another example of serendipity at work as some bright spark in Block's HQ in Columbus, Ohio realised that they had an office block full of mainframe computers that did nothing after office hours and so was borne the Compuserve network (later to be absorbed by AOL) that utilised the processing power and national network of leased lines to 're-freeze' the 'product'; in this case, content supplied for free by subscribers.

3. Noble's book documents the history of the numerically controlled machine tool and specifically, how the early implementers tried to exclude the engineer and toolmaker from the programming side of the process by physically locking the computers, thus preserving control at the management level. The first expressions of numerically controlled machine tools or NCMT were the defence contractors based on Long Island, NY, where the complexity of supersonic wing design was beyond even the most skilled engineer's abilities. There is more serendipity at work here as it presaged Buckminster Fuller's observation about the need for the sub-visible tolerances needed to build the components of his geodesic domes.

4. Cf. 'Designing Freedom', 'The Brain of the Firm' and 'Fanfare for Effective Freedom' by Stafford Beer. A cyberneticist, Beer devised the very first networked, real time analogue of a national economy that enabled governance to know exactly what the state of a country's economic output was. These three references are, in my opinion, the foundation stones of any future alternative to capitalism as they encompass both Marx's political economy and Norbert Weiner's understanding of whole systems and the role of feedback and homeostasis. Throw in Buckminster Fuller's concept of 'Spaceship Earth' and J.D. Bernal's revolutionary approach to education and you have the basis for a workable, viable alternative to the present madness. See <http://www.staffordbeer.com/>

5. Supply Chains link together an entire manufacturing process in real time, regardless of the geographical location, by creating a chain of producers who are organically linked via the Web. Dell Computer is an example of this - for Dell is, in actuality, a marketing and distribution entity that utilises the supply chain and just-in-time production and assembly by OEMs (Original Equipment Manufacturer) as it doesn't actually make computers.

6. See Fuller's 'World Design Science Decade' documents <http://www.bfi.org/> for an (almost) complete archive of his groundbreaking work on the role of design and economic re-organisation on a global scale.

7. In 1999, as the dotcom bubble was bursting, a leading investment analyst was being interviewed on CNN and in response to the question as to whether investors should continue to invest in the hi-tech market, he replied that the investor had no choice but invest as it was either

invest or go bust. This was a classic replay of Marx's observations on the revolution in production in the 19th century, when increases in the efficiency of factory production were occurring almost weekly, forcing manufacturers to buy the latest machinery or go bust, because if they didn't buy the latest machines, their competition would.

REFERENCES:

Stafford Beer (1973), 'Fanfare for Effective Freedom', The Richard Goodman Memorial Lecture, Brighton Polytechnic, Sussex; and in Beer (1975), Platform for Change, London & New York: John Wiley.

John D. Bernal (1971), Science in History, Cambridge, Mass.: MIT Press.

Karl Marx (1891), Wage, Labour and Capital, pamphlett, ed./trans. Frederick Engels; online <http://www.marxists.org/archive/marx/works/1847/wage-labour/>

David F. Noble (1986) Forces of Production: A Social History of Industrial Automation, Oxford: Oxford University Press.