All the things that used to be computers,

and all the things that weren’t and still aren’t.

Simon Penny, 2018

Published in
Les Comportment des Choses.
Anthology, Ed E. Quinz.
Les presses du réel. Dijon France. 2021.
Translated into French by E. Quinz.

“It will be all too easy for our somewhat artificial prosperity to collapse overnight when it is realized that the use of a few ‘excited’ words like information, entropy, redundancy, do not solve all our problems.” Claude Shannon, “The Bandwagon” 1956. (a little-known op-ed).

A central and explicit aspect of my robotic art project Petit Mal (1989-95) was to probe *the idea of software* by building a robotic device in which – as I put it at the time – some of the ‘intelligence’ was in the hardware, even the non-electronic hardware.[1] The structure and behavior of Petit Mal was intentionally bottom-up and closed loop, the physical world always informing and constraining what happened in the microcontroller.[2] This way of thinking was, and remains, anathema to most computer scientists and to all of the digital public who have drunk the *information kool-aid*.

One of the strange cultural effects of the digital ‘revolution’ is the idea that information *exists* as something separable from matter. The suggestion that it does not will seem absurd to many. This is, in itself, a testament to just how thorough and substantial an ontological shift has occurred in western culture over the last 50 years. Of course, there have always been ideas, and mathematical concepts and philosophical principles. What I am referring to is a new phenomenon of quantified conceptual (im)material. We *believe* in a numerical commodity that exist without materiality. As if internet server farms were the residences of platonic ideals, immaterial verities. The notion of separability of information (data) as a quantity that can be extracted and packaged is historically novel. As recently as 1970, the term *software* was poorly defined and its nature poorly understood. This is revealed in the catalog notes to Jack Burnham’s famous exhibition ‘*Software - Information Technology: Its New Meaning for Art’* (Jewish Museum NY, 1970) where the term is applied to a grab-bag of notions by various

---

[1] I was aided in this project by brilliant student helpers Jamieson Schulte, Kurt Jurgen Schaefer and Gabriel Brisson.

[2] This microcontroller was a microcephalic by today’s standards: a 68HC11 processor running at 2Mhz with 128 Kilobytes of RAM. By comparison, an iPhone 10 has 3 Gigabytes of RAM (over 2 million times more) and runs at 2.39 Gigahertz (over 1000 times faster). Yet this puny little device permitted real-time interaction in the real world, with moving humans.

commentators: statistical analysis, information communicated electronically, and what we would now derisively call ‘vaporware’ were all referred to as ‘software’.

I came to robotics as a sculptor, performance and installation artist. My transition to using computers and microprocessors was via mechanics, electromechanics and custom-made process-control electronics – in that order. Making things happen in the world has always been my priority. My preoccupation was with materiality, spatiality, embodiment, agency and interaction. My orientation to ‘media’ technology is thus different from most in the field. I did not come to robotics from computer science, so I did not privilege immaterial information. I did not come to media arts via image media so I did not privilege ‘pictures’. Computing was interesting to me primarily because it made complex real-time interaction possible. But in the beginning, those computers were information machines with extremely limited input and output capability. As such they did not interest me until I could hook them up to the world.

Carburetors and distributors

A couple of years ago I was examining a carburetor on an early 1970s car (an MG, not that it matters). (For the following discussion, a general understanding of the basic operation of the – now almost obsolete – four stroke gasoline engine in its earlier, carbureted form will be helpful.) In cars of this vintage, just one carburetor supplies fuel-air mixture to all cylinders (or rather, the descending pistons create a vacuum that pulls gasoline into the airstream via the venturi effect – nothing is pushing the air or the fuel into the cylinders).

I was particularly taken by the ‘spark advance’ and automatic choke mechanisms. When a car engine goes faster, the pistons move faster. One piston stroke - all the way down and back again - takes less time. [3] But it takes a more or less fixed amount of time to ignite and explode the fuel mixture. This means that in order to get all the power out of the explosion – before the piston gets to the bottom of its stroke - you have to ignite the fuel earlier. The distributor shaft is turned by a gear off the crankshaft. An contactor on the shaft sequentially contacts with lines to the spark plug for each cylinder, in a specific firing sequence. But how does the distributor ‘know’- when and how much to ‘advance’ spark timing? [4]

In a modern car, this would involve a ‘sensor’ collecting signals that are turned into ‘data’, processed via algorithms in a microprocessor, the results of which are sent to fuel injectors in each cylinder. How do you advance spark timing without sensors and computation? This is achieved using a ‘vacuum line’. As the car accelerates, more air/fuel mixture is sucked from the carburetor into the cylinders via the inlet manifold.[5] This suction creates a partial vacuum in the manifold. A hose goes from the manifold to a cavity in the distributor, capped by a rubber diaphragm. This diaphragm is pulled by the vacuum, a distance proportional to the degree of vacuum, which is proportional to the speed of the crankshaft. This diaphragm pulls a lever and spring mechanism that advances timing of the spark. From a ‘digital’ perspective this is altogether ontologically *strange*. The system is *computing,* or at least it is making changes to itself in ways that we would call computing if there were symbols involved. But there are no

---

[3] top dead center to bottom dead center (TDC to BDC)

[4] My use of the term ‘know’ is self-consciously and sloppily anthropomorphic. It is meant in a humorous tone. Computers don’t *know* anything either.

[5] the pipe system feeding fuel air mixture to the cylinders, via a system of inlet valves, driven mechanically from the crankshaft. There is a separate exhaust manifold and system of exhaust valves.

numbers, and there are not even varying voltage levels and there is no A-D (analog to digital) conversion. There is no data, no electronics, not even any electricity. The vacuum *just is* the information. But the vacuum is also the motive power - the two are inseparable.

A similar but even more remarkable job is done in the automatic choke. The choke is the mechanism in the throat of the carburetor that increases the richness of the fuel-air mixture by reducing the air flow. It literally 'chokes' the air intake. The resulting 'richer' mixture helps the engine run when it is cold. How does the carburetor know how hot or cold the engine is? In this case, the automatic choke mechanism monitors the temperature of the coolant fluid. Coolant is pumped through engine and the radiator at a rate governed by the thermostat.[6] A small bypass hose feeds some coolant into a cavity by the carburetor which includes a bimetallic spring. This is a strip made of two different metals with different coefficients of thermal expansion, joined at its ends. This strip flexes or bends in proportional to temperature. This bending pushes or pulls the choke more open or more closed. When the coolant is at correct engine operating temperature, the choke is open.

In the carburetor, two other factors must be integrated: a) current engine speed – indicated again by vacuum, and b) desired acceperation – indicated by the driver's foot, communicated by retraction of the throttle cable. These three heterogenous prompts – a temperature, a (negative) pressure and a pull on a cable - are integrated by an ingenious system of levers, springs and ratchets. This automatic choke is an analog computer. It is instantaneously generating a varying product of three constantly changing quantities. There are no numbers, no representations, no bits, no data, no computation. No *information*. There is just carefully contrived mechanical parts obeying the laws of physics.

It is intriguing that, at this point in our technocultural history, this seems to us, to be a sort of *conundrum*, or a mystery of nature. The engine, as an entuned and enmeshed material and informational system, *is* computing, but the inputs and outputs of that computing is material quantities and physical speeds.

But in fact, for centuries before digital computing, machines have made calculations without separation between mater and information – the anitkythera, the astrolabe, the weaving loom, the jacquard loom, the screw-cutting lathe, and the piano.

…..

Information and analogies

This may be a cybernetic way of looking at the matter. It reminds me of Stafford Beer's proposal to use the ecology of a fishpond run a steel mill. Cybernetics, as a mid-century discipline, also predates the invention of the concept of software. The computing technology of cybernetics was *analog* computing, a kind of computing as different from digital computing as geometry is from arithmetic. But, I hasten to add, whereas Artificial Intelligence is contained within digital computing, cybernetics was not contained within analog computing, indeed the opposite is closer to the truth. In 1954, Walter Soroka spoke of analogy this way - *The term* analogy *means similarity of properties or relations without identity. When*

---

[6] The thermostat/engine/radiator system is a classic cybernetic homeostat, deploying negative feedback. Much like the Bolton/Watts steam engine *governor* before it.

analogous systems are found to exist, measurements or observations made on one of these systems may be used to predict the behavior of the others. The systems need not be analogous in every respect, but only in those respects which are of interest.[7]

How naturalized we are to the idea of *information*. The imposition of the hardware/software dualism has encouraged us to think that 'cleverness' is a quality of software. Hardware, if it is flesh, is dumb meat, and if it is non-living matter, it's just lumps of inert stuff lying around, oppressed by gravity, until it is enlivened by pouring software in. This habit of thinking prevents us from comprehending *how stuff can be clever*, and *how we can be clever with stuff*.

Calculating machines – mechanical, electrical, electronic and digital

Until the 1970s, all kinds of 'automation' and 'control' were achieved via non-electronic means. Here we should distinguish between electrical, electronic and digital. The latter part of the industrial revolution was full of such machinery, and the distinction between calculating machines and other kinds of precision machines was not obvious.[8] The results were displayed not as digits on LCD screens but as positions of a rotary needle on a dial. Perhaps the most elegant such machine was the ball and disk integrator – a mechanical computing device still in use into the 1940s. This is pure geometry at work - a sphere, a cylinder and a disc, rolling on the surfaces of each other, doing calculus. James Thomson appears to have originated the idea around 1886 for his Harmonic Analyser, used to calculate the coefficients of a Fourier series.[9] His Harmonic Analyser was probably inspired by Charles Wheatstone's 'wave machine' of the 1840s. [10] Thomson's machine was one in a long series of mechanical geometrical analysis machines, leading all the way to Vannevar Bush's Differential Analyzer, one of the last great mechanical analog computers.[11] Begun about 1928 by Bush's student Harold Hazen, the machine could solve, approximately, an arbitrary sixth-order differential equation.

Prior to electrification in the last C19th, machine logic was always *mechanical*. Charles Babbage's *difference engine* of the 1820s followed earlier work by Pascal and Leibniz. Babbage, amusingly, also invented the cowcatcher for locomotives. Railways and telegraph systems (and timekeeping) developed in parallel, each leveraging qualities of the other. (Think of the stereotypical image of the stationmaster with his pocket watch). Electricity and electrification created a technological and social tumult in the latter C19th. The (electric) telegraph (and the development of submarine cables) created global high-speed communication and was, arguably, far more revolutionary than the internet.

Babbage and Clement

---

[7] Walter Soroka. *Analog methods in computation and simulation* (preface). McGraw Hill 1954.

[8] Mechanical analog computers called 'gun directors' were used for calculating ballistics trajectories in naval battles, when both source and target where moving. cf David Mindell, Between Human and Machine: Feedback, Control and Computing before Cybernetics. MIT Press 2002.

[9] James happened to be brother of William Thomson, a notable scientific and electrical engineering inventor, who was made 1st Baron Kelvin for his efforts, and after whom the Kelvin temperature scale is named.

[10] Wheatstone was one of the great Victorian polymath inventors. He invented the concertina (a popular musical instrument), developed stereoscopy and a system of telegraph, as well as working in other areas such as spectroscopy and precision timekeeping.

[11] Vannnevar Bush was for most of WWII, head of the U.S. Office of Scientific Research and Development (OSRD), and oversaw the Manhattan project. He was the originator of the thought experiment Memex, regarded by many as a vision of the modern multimedia computer.

The building of Babbage's *engine* was stymied by limitations of precision machining at the time. Much has (rightly) been made of Ada Lovelace as the first programmer, but little attention has been paid to the bloke who actually built the machine, and the challenges he faced. Joseph Clement (1779-1844) was a toolmaker and draftsman, son of a Westmoreland weaver. He appears to have gone to school, but later reports suggest he was illiterate. As a young man he worked as a thatcher, a slater and a blacksmith. On the side he built a telescope, a microscope, a clarinet and set of Northumbrian pipes. Around 1807 he designed and built an improved screw cutting lathe. *In 1827 the Society of Arts again (?) gave him their Gold Medal, and in 1828 their Silver Medal, for the invention of several improvements to lathes, the most important of which was a device to change the speed of rotation of the work as the tool came closer to the axis of rotation, as it might when turning the surface of a large flat disk. Another of his successes was the development of a very large and accurate planing machine by means of which the surfaces of metal plates of large dimension could be finished to a fine tolerance.*[12]

…His list of inventions and improvements is impressive, but none has had more impact than his attempt, started in 1828, to produce screws and bolts with standard diameters and with threads of a standard shape and pitch-essential elements for interchangeability. Although his campaign in this area did not immediately result in success, his best journeyman at the time, Mr. (later Sir) Joseph Whitworth, was the man who ultimately established the standard Whitworth thread, which dominated British machine practice for almost 175 years until replaced by the metric standard. [13]

This last remark is crucial as it makes clear the technical challenges that Clement confronted when attempting to build Babbage's engine. The general state of precision machining thwarted him. Standardized drill sizes did not exist.[14] We pay not a thought to the interchangeability of nuts and bolts, regardless of manufacturer, but in the 1830s, standardised screw-threads did not exist. Standardisation of screw threads (in the UK and British Empire) was achieved by Clement's protegé, Whitworth. He was knighted for this substantial contribution to British industrialization, and the thread system was known by his name into the mid C20th when most of the world converted to metric threads.[15] The attempt to manufacture a mechanical computer, and the establishment of precision machines for manufacturing are inextricably linked.

Machines, increasingly, made machines. These machine-making machines were not fully automated, they depended on highly skilled tradesmen – machinists - for control and calibration. A good, even an ordinary, machinist could make new parts for his machine *on his machine*. (Might this be the inspiration for von Neumann's notion of the self-reproducing automaton?) To a modern spectator, a late C19th screw cutting lathe is a quaint (steampunk) chunk of cast iron. But to an informed eye, it is a mechanical calculator - automatically, reliably and precisely carving perfect geometry - computation output as *form*, not as a stream of binary values. These lathes and mills and shapers, and all the arsenal of related

---

[12] https://history.computer.org/pioneers/clement.html
IEEE Computer Society History Committee, Computer Pioneers by J. A. N. Lee, Joseph Clement.

[13] Ibid

[14] The 'number drill' system still used in USA emerged in the C19th but its origins are shrouded in mystery. It includes 60 drill sizes up to around ¼" that fit between the standard fractional sizes in 64ths of an inch. But the differences between drill sizes are not regular. They may have been based on various wire-gauge systems such as the Stubbs Steel Wire Gauge. There is some conjecture it may be based on Swiss clock makers drill systems.

[15] It will come as a surprise to many that the USA is officially on the metric system.

precision machine tools - cut gears, cams and shafts and perfect cylinders and spheres, in cast iron and brass.

An entire class of skilled tradesmen became attuned to manipulating finer and finer material distinctions, increasingly, beyond the realm of human sensing. For these men (and it was mostly men), 1/1000th of an inch (0.0254 mm) was an unacceptable error. These manufacturing machines were sensori-motor prostheses, extending kinesthetic and proprioceptive sensing and action for hands-on manipulation of imperceptible physical differences. The manipulation of these miniscule quantities was thoroughly embodied, made perceptible through gear ratios, vernier scales and simple clockwork scale amplifiers (dial indicators).

How *do* you make a cast-iron cylinder geometrically perfect? How do you make a cylinder for a steam engine, big enough for an adult person to stand in, geometrically perfect? How do you then make a piston that fits it so perfectly, that superheated steam will not escape? A lathe turns a perfect cylinder because it is itself 'true'. It is flat and parallel and level. Because of the precision to which these parts are made, the tool itself must be at least as precise. Over a length of 3ft (1m), the tool may not tilt or bend or twist, say, even one part in 10,000. (That is, over one meter, it must be correct within 0.1mm). How do you measure that error? How do you calibrate the tool that measures that error? Presumably, with a tool that whose precision is an order of magnitude greater! [16] Precision machining grounds out in simple Euclidean geometry, executed to extreme precision. A (very) straight edge. A (very) flat plane.

In order to make precise machines, one needs precise references. Makers who attempt precision of any degree may well wonder how it is that humans have been able to achieve ongoing improvements in precision. It sounds like a paradox. In the shops of the latter industrial revolution, simple hand scraping of hard metal was fundamental to achieving flatness to mind-boggling physical precision, using Joseph Whitworth's 'three plate method'. With the combination of deft craft and geometrical logic, Charles Whitworth was able to perfect a method to produce impeccably flat surfaces of cast iron. With a hard chisel, some paint, and a procedure, a metal scraper can produced three surfaces flat to almost unimaginable precision.

CAD

The development of digital tools has had the effect that today, a majority of folks, even engineers, have forgotten how to think, or work, this way. Digital tools both remove the engineer from materiality and they hide (tedious) procedures, executing them in the 'background'. The end result being that software users do not know they are occurring and do not understand the need for them. But until recently this work was daily business in factories and shops all over the world. And when you made a mistake - like making the piston a hair too small - there was no undo button. Or at least, the undo button was a blast furnace. We are surrounded by the products of these machines of precision production – every nut and bolt, the pistons and cylinders in every internal combustion engine, the shafts and bearings in every electric motor – yet rarely do we give these extraordinary artifacts a thought. Nor do we consider the

---

[16] Browsing ebay the other day, I found an auction for a 'precision bubble level, used' that appeared to be 50 years old or more. On the instrument was engraved – accurate to 0.0005" (half a thousandth of an inch!) in 10" (0.0127mm in about 250mm).

machines that made them. In much of the world, the skills of the precision machinist, the tool and die maker, and related trades are now almost extinct.

Today, our computational appendages encourage us to think that thinking and doing can be separated. Typical of this is the current fad for 3D printing. Dwight D. Eisenhower wisely opined *Plans are useless but planning is indispensable*. As a builder, a sculptor, a maker, I know the process of design problem solving as an embodied and enactive process. I know, through direct experience, the lived process of design and realization. Even with experience, there are aspects of designing/making that must be lived through, that must be modelled, and often, slept on. This is the nature of embodied making as skilled artisans have done for millenia. The experience of making – and, as often as not, failing – is the source of intuitions about the way the material world works, and these intuitions are *fundamental* to effective design. If you've never dropped a brick on your foot, you will not have a visceral understanding of mass and gravity. If you've never tried to erect a flagpole, and had it fall down in the wind, you will not have a visceral understanding of the force of the wind. We get our understanding of the world from acting in it, and we become more adept and more precise by repeated and increasingly refined bodily practice. This is why musicians practice, why artisans have expertise, and why, until recently, there were apprenticeships.

An anonymous sage said: *The different between theory and practice is greater in practice than in theory*. We are accustomed to privileging theory over practice, even if these distinctions, like the distinctions between hardware and software, when examined, are dubious. We are accustomed to thinking that intelligent activity necessarily occurs while sitting at a keyboard. And therefore, work involving physical labor and the use of tools is less intelligent or possibly not intelligent at all. A new generation, for whom 'technology' is electronic, and computing is digital, have minimal appreciation or understanding of a kind of calculation embedded in iron working on iron. In these circumstances, intellectual challenges had *real* weight.

Philip Agre observed *Building things, like fieldwork and meditation and design, is a way of knowing that cannot be reduced to the reading and writing of books.* There is a way of knowing – knowing materiality – that is incommensurable with numbers, symbols, text and code. Andy Pickering captured this incommensurability in his distinction between the *performative idiom* and the *representational idiom*. The current 3D printing/rapid prototyping craze, especially in educational institutions, is dangerous because it elides the process of iteratively accumulated embodied experience that generates we might call *practical intuition*. It teaches that design is a dematerialised intellectual process, at the end of which you press 'print'. This way of proceeding reinforces the hegemony of the information/matter dualism that has been insinuated into culture by digital technologies, long after it became redundant in philosophical circles.

The software industry succeeds by making difficult tasks seem easy. Difficult software fails in the marketplace. But some tasks are irreducibly difficult. Users, especially students, are oblivious to the work the software does for them, in the background. They feel 'enabled', but their sense of achievement is false. Consider the process of simply opening a CAD package: a perfect world is presented, in which planes are flat and infinitely thin and perfectly parallel all the way to infinity, verticals are straight and perpendicular. Values for lengths, areas and angles are automatically given to high precision. In the real world, the apparently simple task of establishing a flat horizontal plane on rough ground is hard work. It requires reasoning, tools and technique. Erecting a vertical plane on it demands a different set of procedures because that plane has mass and is defying gravity. Now build a parabolic dome… out of bricks.

Many tasks involving the configuration of matter are intellectual tasks of the highest order. Since the Enlightenment, from the age of the automaton and the maritime chronometer, exactitude has been synonymous with material precision. Microprocessors are some of the most precise artifacts humans have ever made, and certainly the most precise ever mass-produced. And yet riding on these precise artifacts is an idea that material precision is somehow irrelevant.

Simon Penny, Los Angeles, December 2018