ALMOST LIKE A CAR

Georgina Voss

Software is an atypical technology,[1] becoming present only in the material space of a machine. What does it mean to 'make' a car, when control is increasingly centralized to the computational means that form the guts of automotive machines? And how does this change the idea or the very nature of what a 'car' is?

In his 1999 essay 'In the Beginning was the Command Line',[2] speculative fiction writer Neal Stephenson marshalled operating systems into an automotive analogy. 'Imagine a crossroads where four competing auto dealerships are situated,' he writes. On one corner is Microsoft, which started out selling three-speed bicycles (MS-DOS), which could be fixed when broken. Next door is Apple, offering expensive over-stylized cars whose insides are mysteriously sealed. A newer competitor, Be, Inc. (a now-defunct US computer company), has set up shop full of Batmobiles (their BeOS system). And on the final corner is Linux – not actually a business at all, but a 'bunch of RVs, yurts, teepees, and geodesic domes set up in a field and organized by consensus'. The people there sell tanks – not old Soviet-era juggernauts, but new and better machines, 'jammed with sophisticated technology from one end to the other'.

The passage works in part because of how beautifully it binds the cultural values of each cohort to their products, but also because of the seeming dissonance between computational technologies and automotive engineering. Yet two decades prior to publication of 'In the Beginning', the analogy had already collapsed into the real. In 1977 the first production car to incorporate embedded software rolled off the production line – General Motors' Oldsmobile Toronado, a lovely sleek thing with an almost preternaturally long bonnet and, inside, an electronic control unit (ECU) that managed electronic spark timing [165]. A decade after 'In the Beginning' was published, BMW and Linux were actively trying to develop open software 'for the connected car' through the GENIVI Alliance.[3] And as I write this, autonomous vehicles are being lauded in the media, by governments and through tech PR releases as the next new thing, jammed with sophisticated technology from one end to the other. Yet your average run-of-the-mill car – still on the roads – already contains around one million lines of code.

Several weeks ago, in a tiny vintage shop in Amsterdam, I came across a Bakelite brooch of a car from the 1940s. The shape of this little speckled amber object was instantly recognizable: the bumper; the curvature of the bonnet; semicircles for wheels. Go to any end-of-year vehicle-design degree show and, despite the slick future-shock exteriors, each of the models on show is undeniably a *car*. Yet throughout, what holds increasing mechanical control (through sensors) and centralized control (through locked-in operating systems) are somehow elided. Automotive failure is still seen as the realm of mechanical breakdown rather than computational control: cast your eye, for example, over the six million software-based vehicle recalls in 2016.[4]

(An aside: although we're talking about land vehicles here, and primarily cars, this isn't solely their affliction – aeroplanes, construction cranes and submarines also increasingly marry a familiar outer face with internal embedded controlling lines of predominantly proprietary code; Britain's fleet of nuclear-ballistic missile submarines all run on Windows XP.)

The talk of cars rolling off the 'production line' continues to uphold this imaginary, shielding the presence of computational power. When we think of the work that goes into making an automobile, we are thinking about the finished car, front and centre – sprawling supply chains and just-in-time efficiency dashes are not apparent. Whilst the engine capacity of a vehicle can now be adapted at distance through 'over-the-air' interventions,[5] any vehicle CEO worth their salt knows about the fetishistic power of the assembly line. When Elon Musk brings the press into Tesla, Inc., it's to the factory floor of the Fremont premises: this white-walled cathedral, with Musk himself foregrounded in the polo shirt and cap of the engineer on the floor. In the background there is always some fibreglass skeleton frame or a half-undone car being built up into something whole. The focus is on mechanical assembly, with human workers and red or yellow robotic

↑ **165** The General Motors' 1977 Oldsmobile Torondo was the first production car to include embedded software. It included an electronic control unit that managed electronic spark timing

arms acting – 'working' – together. What work gets done to create the cars' computational guts (all those hundreds of sensors and millions of lines of code, knitting it together on circuit boards, tested to death) is wrapped away, out of view.

Out of sight, we can take the subtle shifts permitted by software infrastructure for granted. Perhaps this isn't really a surprise: after all, functional infrastructure is often invisible, only becoming visible when it breaks.[6] And how better to break systems than through a world-shaking apocalypse?

Come with Me, then, to the Wasteland

> You find what you find. You bother to drag back what strikes a chord in your heart. You repurpose it to war. You fetishize it, because it's more important than you are, and then you build it, stick a cup holder on it, and head out into the wasteland.
>
> Colin Gibson, 2016[7]

In the 2015 film *Mad Max: Fury Road* what is on view is a post-apocalyptic landscape studded with things that are – to paraphrase Charles Darwin[8] – *almost* like cars; and which have been hauled into being by their owners [090].

From the souped-up motors of the sickly War Boys to the sonic aggressions of the Doof Wagon, the vehicles of *Fury Road* are fucking ridiculous. The philosophy of their design was, the film's production designer Colin Gibson says, to serve four or five different purposes, and it shows in their outrageous forms: echoes here of Jeff VandeMeer's biology-through-a-prism, a Laocoön tangle of bodies and parts.

↑ **090** The fleet of re-purposed vehicles in *Mad Max: Fury Road*, 2015

The protagonist Imperator Furiosa's War Rig binds a core Tatra T815 drive with custom front to a widened 1840s Chevrolet Fleetmaster. Other vehicles are equally unsettling: a child's skull with that double row of teeth. Immortan Joe's Gigahorse combines two 1959 Cadillac Coupe de Villes, mounted atop each other in a swaggering slant; the Bullet Farmer's Peacemaker, a Howe & Howe Ripsaw treaded mining rig boshed into a Chrysler Valiant charger, with (of course) part of a Cessna light aircraft strapped on front.

The vehicles of *Fury Road* are made from the last automobiles of their kind, from a time when an energy crisis has put paid to the industrial-scale mining, smelting, engineering, welding and mass production that are needed to make cars of any flavour from scratch.

These mongrel, repurposed vehicles are old – part of what the writer and curator Justin McGuirk describes as the film's 'retro future',[9] where there are 'no new technologies, no new energy sources, and certainly no Tesla home batteries'. (The background presence of petrol everywhere, in a scarcity landscape where water is hoarded like gold, is something that the film quietly moves past.)

But *Fury Road*'s old machines serve another purpose. After an energy crisis has pulled apart the world, it seems that nothing that is left is digital, and the 'retro-future' specifically locks the world in a pre-computational era. Crankshafts and combustion engines, and thousands and thousands of bolts, are fine and dandy; the presence of a functioning motherboard is not. Any surviving machine whose animus depends on computational systems has probably been left to rust, or been gutted for small parts alone. The vehicles that are lovingly built anew take their bulk from the years when cars were analogue: it is a *1959* Coupe de Ville, a *1940s* Chevy Sedan that persist, adored and

worshipped, because they can be repurposed; their action and meaning can be controlled.

Different social groups give different social meanings to things. In the parlance of Science and Technology Studies – aka STS – this is known as 'interpretive flexibility',[10] and it can steer the development of a technology from one terrain into another. In the wasteland, a century-old Chevy becomes something that can be reanimated and defied; the audience can infer that any electric vehicles have long been shredded, scavenged only for parts. By dint of an apocalypse, the psychic conflict between code and car has been shelved: there is no industrial-scale manufacturing or supply chains in the wasteland. What the War Boys, and Vuvalini, and Mauraders, make is a new hybrid breed: imagined through their own needs and desires, restricted only by the material limitations of the harsh landscape around them.

The Car as Agricultural Machinery

What, though, if there is mass manufacturing and complex supply chains – and constraints around how are a vehicle can be made are enforced through *control* of these complex systems?

If we move from the wasteland to the farmyard, to American rural life at the start of the twentieth century, we can see the first cars rolling into view. In 1908, via mass assembly, the Ford Model T set the scene as the first affordable automobile, opening up travel to the newly emerging middle classes.

As STS scholars Trevor Pinch and Ronald Kline note, farmers were at first deeply suspicious of these new noisy, dangerous machines, which scared or even killed their animals[11] – 'Devil wagons'. Cars eventually made their way on to farms, but not as *cars* per se; more as the idea of what a car could be, steered through the needs and desires of rural farmers and actualized through hard mechanical work. If one were to run a belt over an automobile's spinning wheels, it would transform the vehicle into a stationary power device, which – one Kansas farmer noted – would enable a farmer to 'save money and be in style with any city man'. Adapted as such, cars could be used to run agricultural machinery, including water pumps and wood saws, but could also be put to use on domestic chores, such as powering washing machines or running butter churns. By 1915 one Maine farmer had found so many uses for his car that tax assessors didn't know whether to classify it as a 'pleasure vehicle' or agricultural machinery.

Initially Ford seemed easy with, and even gently delighted by, the malleable identities afforded by farmers to their motors. But manufacturers have their own interpretative frame around their expectations of their products, and the slippery pleasures of early rural use didn't last long. As 'barnyard mechanics' spread, a flurry of new accessory companies emerged, offering up modification kits that could, for example, replace rear wheels and re-form a vehicle into something more tractor-like. In 1916 the Ford Motor Company informed its dealers that converting vehicles into a form not sanctioned by the company would cost them their dealership; and two years later it warned owners that altering their cars in this manner would void their warranties.

In limiting interpretative flexibility, car manufacturers flexed their muscles: closing down the meaning of their cars and pushing vehicle owners into buying their own mass-produced items – tractors, for example, rather than conversion kits or stationary power conversion kits. But to exert this control required companies to wield their power relationships at a distance, via nodes in the supply chain. (Compare this to the imaginary wastelands of *Fury Road*, where auto companies have expired along with the rest of the world – no War Boy expects a tap on the shoulder, admonishing him for tearing apart his 1934 Chevrolet 5 Window Coupe). Move to the present day, however, and control of the machine itself could be deployed at a distance.

A century after Ford dealerships were threatened with sanctions, automotive companies have new ways to limit what a person can do with a tractor, or a car, or something like a car. Over the air or through the wiring, control now has the capacity to be centralized, computationally.

Illusions of Adaptive Automation

So the year is 2019, and you own a vehicle; and, one day, it breaks. Even if it's box-fresh from the factory; even if the marketing emphasizes 'intelligence' and 'agility'; even if you think about the astonishingly complex networks of production and just-in-time flow and orchestration that have brought this enormous mobile computer to you, and you have to lie down for a while. Even then, one day, it breaks, and rather than being able to get under the bonnet and fix it, you find that in this nest of sensors and software you're locked out of your own machine. (Is it even 'your' machine?) One day you find yourself using a USB-to-tractor cable to jack a Windows laptop into your John Deere combine harvester to upload an unauthorized version of Service Advisor.[12]

John Deere is the brand name of Deere & Company, which manufactures agricultural vehicles and other heavy machinery, and which has begun transforming into a software company that runs its technology on tractors – somewhere in the balance between hundreds of software engineers developing enormous complex codebases, run through tens of thousands of hours of testing on 30-ton equipment, and farmers having tight timeframes in which to get their equipment up and running to plant or harvest. Some amount of tinkering is permitted by John Deere, but only up to a point – cross that line and farmers can be sued.

Marketing bumph for these machines abstracts away from the computational – one website for John Deere's Service Advisor shows a man in overalls handing a customer an actual cardboard box with 'Service Kit' printed on the side. There are workarounds, because there are always workarounds: a secondary market – an *after-market* – has begun to crawl up through the cracks, to craft engine computers and their ilk to be patched into your car. Modern cars are so tremendously complex that an after-market unit might need to spoof the presence of the original, so that the non-engine parts work properly – a gearbox is often electronically controlled and may expect, or require, the original unit.

The thing is the same, but it is not. These processes are allied to, but distinct from, what the film-maker Astra Taylor describes as 'fauxtomation' – an obfuscation of the work that goes on alongside apparently autonomous machines, giving the illusion that they're smarter than they actually are. The sleight-of-hand that pulls the eye away from the computational systems embedded in automotive engineering is not fauxtomation per se – not only because a vanishingly small percentage of current vehicles are fully autonomous, but because self-driving cars are simply those for which the software already present in so many vehicles today is given primacy.

The locus of power and control in automation has quietly shifted to computational systems over past decades. With this move, the nature of what it means to make, own, and mend a car has radically changed – but the *mis en scene* of the factory floor and production line still calls back to a time of mechanical control.

Automotive work has always been adaptive – there is no point when the car simply *is*. Despite the controls laid down by the design process and threaded into production, vehicles are used in massively disparate spaces and places, to different and physically gruelling ends. They flicker through diverse identities and meanings. You find what you find. The after-market is the space where the adaptive work is done to bring these meanings to fruition. It holds up an angled mirror to the factory floor and shows what is missing from the imaginaries and slick presentations of the automotive industries, but also where seemingly seamless computational command-and-control can have a knife (or new engine part; or chunk of code) jimmied into it. *Here* are the possibilities for doing the work that binds together all parts of the automotive system – dragging it back, repurposing, building and heading back out into the world, on a different path.