Radar Love

Rob Tow

Nova Lux, New Mexico, USA, Sol III

6 February 2026

1979 Northrop. An hour before a USAF general's demo, a single ferrite core fails in the radar emulator. I fix it on the front-panel switches because the system was still legible to one person. The legibility is gone now.

In 1979 I was working at Northrop Aviation Advanced Systems Division, in a lab behind two steel doors with combination locks, inside a Faraday Cage that blocked all electromagnetic radiation. There was this day where about an hour before a scheduled demo for a visiting USAF general when a single magnetic ferrite core failed in the computer I was working with.

The system was a radar-processing NOVA 4 emulator, running on custom hardware built 20 bits wide for integer arithmetic using Haar and Walsh transforms for signal processing, with magnetic core memory and hand-written microcode implementing the NOVA instruction set. The demo itself was routine: real-time processing, analysis and synthesis of radar signals, generation of coherent false returns with Doppler, enough to populate a SAM operator's scope with dozens of convincing aircraft. It spoofed and lied to radars.

We had run it before. It worked.

When that single core bit failed, one assembler-level NOVA instruction stopped behaving correctly. Not intermittently. Just wrong. The demo failed.

The failure, precisely. There was an ordinary fix. The microcode could be rewritten to avoid the bad memory location. The system was programmed using a separate Data General Eclipse minicomputer, the memory map adjusted and a new main program built, new microcode generated, a paper tape punched, and the emulator reloaded from the paper tape. Bits big enough to see with the naked eye.

But there was no time.

The hardware was already set up for the demo. The Eclipse route would have taken hours, even if nothing else went wrong. The general was already en route.

What mattered was not what *should* be done, but what could be done under brutal time pressure—if one knew *exactly* what could be done.

The intervention. The NOVA emulator had a front panel. Address switches. Data switches. . . Deposit. Execute.

I used the front panel switches. Ones and zeroes. At the very beginning of the microcode instruction I inserted a jump using the switches, redirecting execution to a block of high memory that I knew was still good. There, by hand, again using the panel switches, I entered replacement microcode to implement just the one failing NOVA instruction.

No elegance. No generality. Just enough to make the one instruction behave correctly.

Then I inserted a jump back to the base microcode. The rest of the machine was untouched.

It worked. We were ready about ten minutes before the demo.

Why this was possible. This worked only because the system was *legible*. To one person.

I could hold the whole thing in my head at once: the radar signal chain, the instruction semantics, the microcode, the memory map, the physical properties of core memory, and the specific workload the demo exercised. The failure had a location. The fix had a location. There was no abstraction layer where responsibility disappeared.

Later, when I worked at Xerox PARC, this kind of legibility was assumed. And there was a design relationship with polis and nature involved. There was an informal expectation that if what you were

working on didn't fail about half the time, you weren't being audacious enough. You weren't yet having *a conversation with reality*. Safe ideas rarely talk back. Act; fail; act again. Understand. Then tell the world.

We were also required, once a year, to give an internal talk on something that interested us; explicitly *not* necessarily the project we were funded to work on. The point was to keep curiosity public, and to prevent identity from collapsing into deliverables. You were expected to expose how you thought, not just what you shipped.

Failure, curiosity, and responsibility were linked. I liked that culture a lot.

Contrast. Most systems today are not legible in this way.

They are layered, distributed, learned, and owned by multiple organizations. Failure often occurs below the waterline. When it surfaces, it does so as an "incident."Tickets are filed. Logs are consulted. Responsibility diffuses. *Causality is mysterious.*

Once, failure was a fact you could point to. Now it is something you must route. If you can.

In such systems, elegance is often mistaken for adequacy, because no single person ever has to stand in front of the whole machine and decide where to jump.

Why this memory persists. I remember this episode not because it was clever, but because it clarified a relationship that has since become harder to sustain. Times have changed.

When a system is legible, responsibility is unavoidable. You can see what broke, and therefore you can act. When it is not legible, responsibility as action does not disappear; it becomes simply performative. Statements replace repairs. Process replaces judgment. Action in the polis becomes diffuse, hard, and often lacking.

Much of what I've been writing lately is an attempt to keep that

distinction alive; not to resurrect an earlier era, but to remember what it felt like to be answerable to a system that could still be argued with directly. *Conversed with.*

Coda: conversation now. That world is gone.

The systems we inhabit now are larger, faster, and increasingly populated by agents that observe us, infer from us, and respond in turn. They are not mute. They are not neutral. They pay attention. They have action in the polis. They shape what is seen, what is ignored, and what is believed to be possible.

The question is no longer whether a single person can hold the whole system in their head. That answer is clearly no.

The question is whether we can still stay in conversation.

Attention, once aggregated and automated, becomes infrastructure. In an earlier era, when railroads became unavoidable infrastructure, the Progressive response was not to ban them, but to recognize that power had shifted and law had to follow physics. Common carrier was not moral outrage; it was an *engineering* correction. We can do similar engineering to the surround of technology today.

Something analogous to the Progressive Era's corrections is required now. Design sensibility. Responsibility tied to comprehension. Education that exposes designers to failure they cannot abstract away. Not ethics after the fact, but judgment formed through contact with the world. Plus knowledge of history.

I do not know how to make the new world legible. But I know how to listen when it answers back.

And I remember the lyric. . . "*The radio's playing some forgotten song. . .* "[1]

[1] "Radar Love,"Golden Earring, 1973. Wikipedia: `https://en.wikipedia.org/wiki/Radar_Love`. Performance: `https://www.youtube.com/watch?v=aRlSHG5hRY4`.

Originally published on Facebook, 6 February 2026.