

Assembly Required

Read Online

The published version is available [here](#) online.

Bits Versus Electrons

Assembly Required: Rods, Belts, and Electrons

By Bob Frankston

I had relatives in town, and we went to the Charles River Museum of Industry and Innovation. It's a small museum on the banks of the Charles River in Waltham, Massachusetts, also known as Watch City. One of the big advances in the 19th century was the manufacture of watches using standardized parts, and the same for bicycles. The museum is where Henry Ford went to learn about manufacturing. It made me think about how we manufacture consumer devices and our increasing ability to assemble our own devices out of collections of general-purpose parts, such as entire computers and generic connectivity. We also have new kinds of components like the Alexa skills. The value is shifting from the parts to the new entities that we assemble ourselves.

At the museum, I had a chance to speak to the docents and learned about the early history of how power was transmitted. Initially, power from the dam was transmitted using rigid metal rods and gears. The use of leather belts was an advance that made it easier to manage the relationships between the components. A factory is not just a collection of machines but a whole linked together by the belts and pulleys for a common purpose.

I'm impressed by mechanical systems designed to accomplish a particular task. The TV series *How It's Made* shows machines that drop each part into just the right place and position

Digital Object Identifier: 10.1109/ICEE.2017.2755240
Date of publication: 11 December 2017

The design of circuits forced a mathematical approach because electrons aren't as visible as mechanical parts.

using mechanical systems. Class electronic circuits have similar characteristics, using wires to connect tubes, capacity, resistors, inductors, and whatever it takes to accomplish the tasks. Many of the design principles of the mechanical and electrical systems shared concepts of feedback and systems design. Perhaps I have a biased perspective, having studied systems in an electrical engineering department, but my sense is that the design of circuits forced a mathematical approach because electrons aren't as visible as mechanical parts.

Whether designing a factory or a television, each assembly of components was carefully designed for a purpose. As I wrote in my previous column, "Whither Consumer Electronics" (*IEEE Consumer Electronics Magazine*, October 2017 issue), this whole systems approach extended to entire industries designed around a particular technology and market or use case. The design of such factories or devices fits very well into a top-down command-and-control approach to design with layers of abstraction or components. This attitude was even extended to treating people as components in what has been called *scientific*

management or *Taylorism*. Implicit is the assumption that there is a fixed metric (or use) against which one can optimize every element of systems, be they devices or people.

Initially, integrated circuits were seen as a way to improve the efficiency of electronic systems by squeezing more components into less space at a lower cost. Even better, a microprocessor could take rules built into circuits and implement them using programmed logic. There were two versions of some of the manuals for early microprocessors, with one set being aimed at circuit designers rather than computer programmers. Years later, I encountered another version of this in speaking with my seatmate on a flight. He explained the challenge he faced in making computer-based control systems operate like existing mechanical control systems so that they could be understood by the existing maintenance staff. As microcomputers came into their own as software devices, the rest of the computer was built to support the microprocessor and devices were viewed as peripherals.

PEER DEVICES AND COOPERATION

There is another design tradition: peer devices and cooperation. It's not new in and of itself, because people and organizations have always worked together. What is new is that software has enabled us to move beyond mechanical analogs in building systems that are agile and resilient. It's easy to see this in terms of large computers connecting

At the museum, I had a chance to speak to the docents and learned about the earlier history of how power was transmitted. Initially power from the dam was transmitted using rigid metal rods and gears. The use of leather belts was an advance that made it easier to manage the relationships between the components.

A factory is not just a collection of machines but is a whole linked together by the belts and pulleys for a common purpose.

I'm impressed by mechanical systems designed to accomplish a particular task. The TV series "How It's Made" shows machines that will drop each part into just the right place and position using mechanical systems.

Class electronic circuits have similar characteristics using wires to connect tubes, capacity, resistors, inductors and whatever it takes to accomplish the tasks.

Many of the design principles of the mechanical and electrical systems shared concepts of feedback and systems design. Perhaps I have a biased perspective having studied systems in an Electrical Engineering department, but my sense is that the design of circuits forced a mathematical approach because electrons aren't as visible as mechanical parts.

Whether designing a factory or a television each assembly of components was carefully designed for a purpose. As I wrote in my previous column, *Whither Consumer Electronics*, this whole systems approach extended to entire industries designed around a particular technology and market or use case.

The design of such factories or devices fits very well into a top-down command and control approach to design with layers of abstraction or components.

This attitude was even extended to treating people as components in what has been called "Scientific Management" or Taylorism. Implicit is the assumption that there is a fixed metric (or use) against which one can optimize every element of systems be they devices or people.

Initially integrated circuits were seen as a way to improve the efficiency of electronic systems by squeezing more components into less space at a lower cost. Even better, a microprocessor could take rules built into circuits and implement them using programmed logic.

Rods, Belts and Electrons

I had relatives in town and we went to the Charles River Museum of Industry and Innovation (<http://CharlesRiverMuseum.org>). It's a small museum on the banks of the Charles River in Waltham Ma, AKA Watch City. One of the big advances in the 19th century was the manufacture of watches using standardized parts. Same for bicycles. It's where Henry Ford went to learn about manufacturing.

It made me think about how we manufacture consumer devices and our increasing ability to assemble our own devices out of collections of general purpose parts such as entire computers and generic connectivity. We also have new kinds of components such as the Alexa skills. The value is shifting from these parts to the new entities that we assemble ourselves.

Indeed, there were two versions of some of the manuals for early microprocessors with one set being aimed at circuit designers rather than computer programmers. Years later I encountered another version of this in speaking to my seatmate on a flight. He explained the challenge he faced in making computer-based control systems operate like existing mechanical control systems so that they could be understood by the existing maintenance staff.

As microcomputers came into their own as software devices the rest of the computer was built to support the microprocessor and devices were viewed as peripherals.

There is another design tradition – peer devices and cooperation. It's not new in and of itself since people and organizations have always worked together. What is new is that software has enabled us to move beyond mechanical analogs in building systems that are agile and resilient.

It's easy to see this in terms of large computers connecting to each other over networks but it's also true within devices. I remember looking at the design of a VCR (do I need to explain what Video Cassette Recorder was?) and seeing each chip representing a functional unit and they were interconnected by wires used more as networks than as class circuits.

The advantage of this approach is that it allows for arms-length development and independent innovation. Today's smartphones are a powerful example of this. Also, a strange one since, internally, they use a protocol from personal computer modems of the 1980's – the so-called AT protocols – for signaling!

Resilient arms-length interfaces aren't new with software. After all, the reason that ATT lost control of the devices in people's homes is that the red/green interface (AKA tip and ring) of the analog wires was very resilient so they couldn't make a credible claim of harm to the network. It wasn't just good engineering it was necessary.

The newer digital phone network required special gear at the customer's end but that very complexity was its undoing. It was just too difficult to get every element of the system to work just right and alternatives were becoming possible.

The same technologies that enabled the digital network enabled a new approach to "wiring" systems together though it wasn't obvious at first how radical this was. Even today Bluetooth is modeled as replacing the wire between two nearby devices.

To a large degree the Internet is perceived as wire replacement at a larger scale. It is a way to send packets from one place to another very much like a traditional tele-

communications network albeit with software outside the network taking responsibility for delivery rather than relying on network operators.

The technique worked so well we don't necessarily need to exchange packets at all! If I want to watch a video all I need is the name and I can ask around to see if someone (or something) nearby has a copy. Or, perhaps, I want to find Dr. Smith but someone recognizes that Dr. Smith is a dermatologist and tells me whether Dr. Jones, another dermatologist would be acceptable. This kind of evaluation in which we operate on the description, or schema, is the heart of modern databases. A reason why relational databases have worked so well is that we don't actually have to go to the data. If I ask for all records after June 2000 and before April 1999 then I'm asking for the null set and no need to check every record against the criterion.

For connectivity, the wiring diagram comes in the form of the relationships or bindings between end points. With home control systems, we can simply establish a relationship between a control (a switch) and a light by using the identifier of the end points. We can also be more creative and have abstract end points such as "dim the living room lights so I can watch TV". But I'm getting ahead of myself.

In the simple case, we are tying together devices so they can work together. In the days of component Hi-Fi you'd run a wire from the tuner to the amplifier and another set to the speakers.

Today those devices are full powered computers. Even light bulbs (such as my LIFX Wi-Fi bulbs) have full computer capabilities with support for Internet protocols. They are full peers with each other. Yet we still talk about home automation as if there was one device controlling an entire home rather than cooperating systems.

My wall controllers are repurposed generic devices (including smartphones) that act as wall controllers thanks to software. They not only send control messages to the light bulb, they also show me the state of the bulb. This becomes important when I take the controller with me and it still works even when I'm thousands of miles from the bulb.

This creates degrees of freedom and possibilities. But it also presents us with challenges. How do I specify who can control the light? It used to be simple – if I was physically in the room then I had authority to use a switch merely by being there.

We face a challenge as such implicit assumptions require explicit translation into software and policies. Many of these bugs and problems stem from not realizing the possibilities. For example, when I give my printer a public IP address (especially with IPv6) how do I specify who can

use it? Right now, we have security through obscurity which means those devices become vulnerable over time.

Returning to the seemingly simple example of the light bulb, the ability to separate the idea of “light the room” from “turn on a particular bulb” is very powerful. The function might be implemented during the day by opening shades instead of turning on a bulb.

The comingling of function from device is central to the evolution *away* from the idea of a smartphone. Today we treat the smartphone as a single device we carry around for all purposes with apps adding capabilities. This is backwards. The reason I like to write web apps (to use one name for applications written within the browser) is that I can go to any device and use the app and don’t have to worry about installation.

This doesn’t mean that we are totally apart from the physical world. I do need ways to indicate my presence or authority and carrying a device is one option. The device can sense that I am in the same room as my lightbulb and I can bring back some of the physical world knowledge but, this time I am doing so explicitly.

We are also shifting away from carrying physical keys to tokens of authority. The car fob that the car recognizes to unlock in your presence is one way of taking advantage of the new ways of implementing authority. But why do I need a fob per car when I can simply associate a physical device with my presence? If I’m wearing my smartwatch that alone should act like the car fob. I can then grant temporary authority.

As I wrote in <http://rmf.vc/IEEEENotInControl> when you ask a friend to turn on the light so you can read they know implicitly what you mean. When we connect systems, or ask Alexa to turn on the lights we face the challenge of making or incorporating implicit information into the relationships.

Today distance doesn’t matter. In the 1990’s we went from waiting days to get foreign newspapers to reading them online. We didn’t just make it more efficient to reach to distant places, we made distance irrelevant!

All this freedom and flexibility presents a challenge to the consumer electronics industry as value shifts from manufacturing and careful engineering to the ability for anyone to configure sets of devices themselves. Over time we’ll improve software to manage such configurations and the level of expertise will become democratized.

Today everything is becoming consumer electronics as is illustrated in this story [How I replicated an \\$86 million project in 57 lines of code](#). This doesn’t mean we are ready

to shift completely away from large projects. It does, however, illustrate the potential to create our own solutions by assembling available elements and services rather to start afresh for each new project.

In the days of factories with belts and pullies it was difficult (and expensive) to assemble the components into a whole. Today it has become much simpler and we can learn by doing. How long will it take us to realize it and how will the consumer electronics industry adjust to the new landscape? This is an opportunity to lead, not follow.