Whither Consumer Electronics

Online

You can read the published version of this article, here, online at the IEEE website.

Whither?

The word whither means "where to". The old world of consumer electronics with an industry for each technology is fading away. It's the world in which the IEEE was formed.

As that world withers, there is an exciting future of participation and contribution rather than just consumption of content.

A Time of Change

This column, *Bits vs. Electrons*, focuses on the fundamental shift from using electronics to create solutions to a world constantly being redefined by software.

The term "consumer electronics" is framed in the world of electronics and, on the surface, smart products seem like a

continuation of this world. But the way we create consumer products and services has changed in a fundamental way.
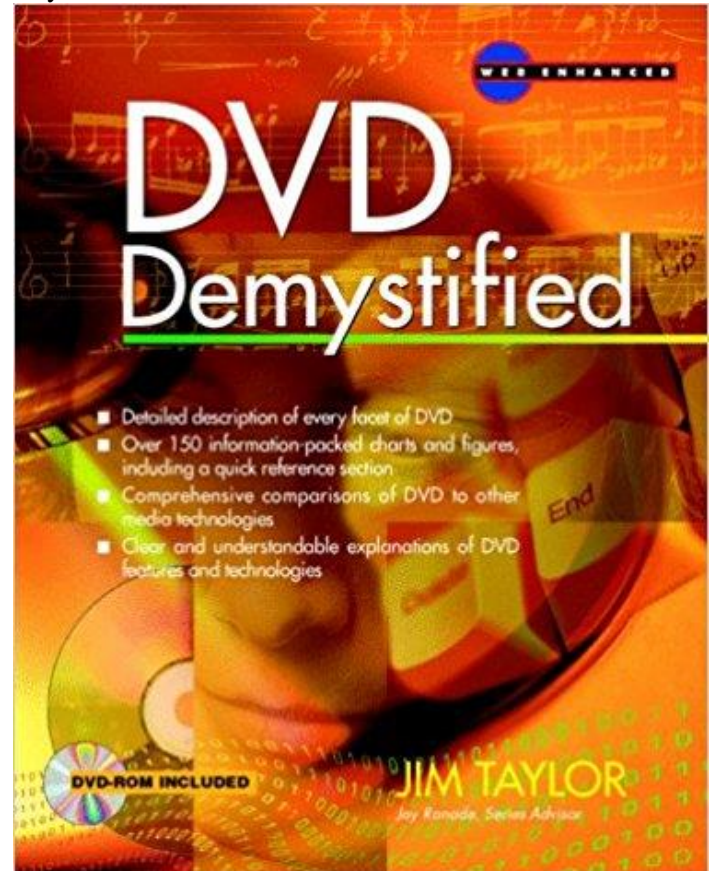
As in any such transition the new technologies and capabilities are seen through the lens of traditional products and their business models. Netflix is treated as television even though it is very different. We fail to see major changes because they are crafted to look familiar. Thus, we continue to apply old policies even as the world changes underneath us.

As the Consumer Electronics Society, we need to understand this new world.

As an organization, the IEEE faces a challenge as the development of products becomes more agile and connected, and less dependent upon standards organizations to coordinate activities.

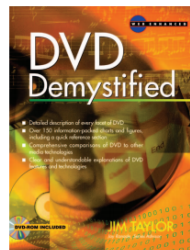But first let's look at what it used to take to create products and services.

Peak CE

DVD to Firewire

In organizing my book shelf (yes, I do have printed books) I came across the 1998 edition of DVD Demystified. The book was written at a time when the Digital Video Disc was trying to prove itself in the marketplace.

It's telling that another definition was Digital Versatile Disc which explained DVDs as not just a transport for bits but rather a complex set of technologies that all had to come together for the product to work.

The book starts out with a fascinating history of digital media with Compact Disc (CD) playing a central role. It's fascinating to read about efforts to treat CDs as a computer storage device with its own file system format. For computer people the major attraction was that a CD could hold half a gigabyte in a time when a floppy disc held less than a megabyte and hard discs with a few hundred megabytes were expensive.

The focus for consumer electronics companies was on selling players rather than computers and CD-I (CD Interactive) defined an on-disc format that allowed for doing more than just menus. The player could support applications such as games and educational programs. Of course, the player was (yet another) special purpose computer.

There was another "format", CD-Plus which was an enhanced CD designed to run on standard personal computers. It could automatically install an application and content on the computer. When I tried one CD Plus disc I found that it didn't observe any of the standard practices for installing software but simply used the XCOPY command to copy files into the places the naïve implementer thought they should go. There was clearly a culture clash.

The DVD industry is built on the technology of CD-I (CD Interactive). CD-I has its own menu structure and other native capabilities. One would author the disc as a complete product.

This clash was further evident in a conversation I had at CES with someone from the content industry during the transition from DVD to higher capacity formats (HD-DVD and Blu-ray). To me the transition was simply one of increased capacity but to the industry each format was a completely different product. The idea of each disc format being a product fit very well into the traditional modeling of selling movies in boxes that had developed during the era of VHS. (VHS having its own history of industry battles and trauma).

This was also the era when IEEE-1394 (Firewire) was vying for adoption. For software people, it was simply a

high-performance network technology but for the traditional CE companies it was the basis for a new line of connected devices. One would be able to buy a Firewire disc drive and use the menu structure presented on a television screen to choose what to play and to manage the media files while respecting the copy protection rules. These rules could limit the number of times the content was played, among other restrictions.

These policies were added at the behest of the stakeholders from the content industries. Such policies add an additional level of complexity in limiting participation to those who are willing to sign very restrictive agreements. Such policies make it difficult to evolve the technology.

This is part of the reason I felt that Firewire was doomed from the beginning and I watched as it played itself out. Each year at CES there were incremental improvements including players that implemented file systems and allowed one to configure multiple devices much like one plugged components together in the early days of Hi-Fi.

The copy protection rules are a case in point. They were added at the instance of the content owners. Building such smarts into the network limit its ability to evolve beyond the initial design point and use cases.

Some of the problems were evident when I attempted to take advantage of the smarts of 1394 and tried to share devices between two PCs. I watched as a camera was shared by two PCs each seeing alternate scan lines. I couldn't control which PC owned which disc drive. It was outside the use cases built into the protocol.

1394 worked passably for using peripherals on a single PC but the extra complexity made it expensive. The protocol also defined very tight response times that limited the length to about six meters. (OK, 18 feet for those who like spurious precision).

USB, SCSI etc.

USB competed with 1394 but was designed by software people as a simple replacement for the serial port on computers. I was in the Windows 95 group at the time so I watched as it came into its own. The goal was to keep it simple to use and inexpensive. It had some useful features such as the ability to automatically identify each device.

But it also suffered from being smart. It had synchronous modes for time sensitive devices that limit the number of active devices the (then) high speed 10 megabits per second wire could support. (yet another example of solving problems in the network)

Devices were organized into class hierarchies and each one needed a special drive to work thus creating opportunities

for failure. It took years to work out successive versions of the protocols. That process was slowed by the need for each hub to support the necessary version of the software. Even today I still face challenges with my USB 3 devices.

At the time, I was working on home networking becoming familiar with the power of the strict separation of packets from their purpose. The many different wiring protocols such as IDE and SCSI were all essentially packet protocols limited by their design points and their stakeholders.

USB has survived, even thrived because it was able to evolve past its original design point.

Today USB has become the de facto standard for powering consumer devices even though it was initially forbidden. This is an example of expanding the use case. Building on what is already available is a powerful dynamic. Instead of developing a new standard, it's easier to agree on an existing one even if in this case it was only an accidental property of the need to power computer peripherals. In return, the accidental adoption of USB for power created a large enough market to invest in USB-C which allows for software-based negotiation of power using the smart chip within the cable.

Building on accidental properties such as USB for power also has its risks. USB as a way to connect devices to a single PC had its own accidental property – security by physical connection. Implicitly trusting peripherals just because they are physically connected has become a vector for attacks.

The relationships were defined by the physical cable. This limits its reach and flexibility. The need for special drivers also adds failure modes.

It has become much easier to connect to printers on a local network though home users may still prefer USB because it's still simpler to use wires. At some point, we'll just be able to say "that printer" and not worry about installing drivers and other spurious complexity.

Peaking

I consider DVD and USB as part of classic consumer electronics with value being created using technology and, in particular, electronics.

The technologies have continued to advance so it seems as if consumer electronics is just moving forward but the value is now being created using software and is not dependent upon any particular technology. Generic connectivity, IP connectivity in particular, has enabled one to create solutions without depending on new technology.

Conversely new technologies can be developed on their own, such as USB-C, without requiring new software. Later software can take advantage of the new opportunities.

This is not about the Internet *per se* but rather the idea behind the Internet – the ability to use software to program around obstacles and to rapidly iterate as we learn from experience.

Electrons ➔ Bits
A World of Software
Just to give a sense of how different the world of bits is think about buying a product from Amazon. The products are stocked in multiple warehouse and shipped from the closest one. This is similar to the technique of caching in software.

In 1998, my mother asked me to send a toy from NY to her grandson in Seattle. Instead of shipping the physical object I simply send the part number to my brother who then bought the identical toy in Seattle. The key is in understanding that the toy he bought in Seattle is indeed the same toy *for this purpose.* Our ability to effect results by simply rebinding identifiers is part of what makes software so very different from the world of physical objects composed of items or changes that require transfer of electrons.

In some ways, this transition is akin to the shift from the days of factories powered by big steam engines or waterfalls with power transmitted by belts and pulleys to a world in which we could simply route electrons around and manage it all with hardwired control logic. Electronics gave us the ability to perform miracles such as creating a television broadcast industry with microsecond coordination around the world. No wonder the IEEE grew to be such a large organization.

The world has changed again. Products are created merely by manipulating rules, AKA using software. There is still a role for complex hardware as in the chips we use to build computers and devices. But now the relationship between applications and particular hardware has been loosened or in many cases eliminated entirely.

With software, we can define relationships abstractly as with the part number for the toy truck.

APIs
APIs, or Application Program Interfaces, are becoming the norm for providing services using web-based protocols. As with the web the relationship between the client and the provider is stateless, assuring a degree of decoupling thus allowing rapid innovation with only limited dependencies.

This is very different from the highly-integrated systems of traditional consumer electronics. This is an architectural principle which is about more than web APIs and is a general principle for network protocols and systems design.

USB-C is a useful example. The idea of having the same wire carry power and signal is very convenient. We need to look beyond the particulars of the current implementation and think about a software-based protocol for power management independent of any wires. The fact that the USB cable acts as a data medium is useful but the negotiation protocol itself should be independent of USB so that it can be adapted to other methods of power delivery. But It's not just power delivery but rather bidirectional power transfer. I should be able to send power from my phone to yours if necessary.

We need to be careful to view the accidental path of history as a source of ideas rather than being prisoners of the past. The idea of using USB as a power source is, well, a powerful idea but we can now think about the more general idea of routable, negotiated, power independently of USB.

The very term API itself is an example. It's one particular example of an arms-length interface but the web protocol is asymmetric. We may continue to use the term even as we shift to symmetric messaging protocols rather than framing interfaces in terms of procedural calls in programming languages.

The Internet itself is a case in point. Its roots lie in an effort to internetwork various networks. But today there is one Internet in the sense of a common address space with the IP packet representing a generic interface independent of the particulars of underlying transports, AKA, networks. To put it another way, the Internet is an API we use to send packets to destinations.

We can view the debate over network neutrality in this context. The constraint of the API is that a network operator doesn't see the meaning of the packets therefore can't implement policies based on the meaning or purpose of the packets. It is a challenging debate because public policy is focused on existing use cases rather than creating opportunity for innovation.

The term "smartphone" is used for a highly integrated physical device rather than an accidental integration of capabilities. Rather than having a dichotomy between mobile and immobile applications we should simply think of factors such as screen size that can be addressed simply as parameters. This is already happening as we shift to using browsers as platforms rather than being tied to the physical device.

Resources

The shift to thinking in terms of APIs or, more generally, architectural boundaries, allows us to think in terms of opportunities rather than creating a carefully crafted solution for each purpose.

It's very expensive and limiting to have to craft a whole industry for solution. It costs essentially nothing to take advantage of opportunities.

The Next Cycle

As the age of specialized hardware gives way to generic hardware tied together by software we're seeing the rise of integrated social networks. But that's another conversation.

We also need to move beyond the legacy of mainframe computing and get past concepts such the need to install software and drivers. I should be able to just pick up the nearest device and use the application or perform my tasks.

Whither Consumer Electronics?

I see consumer electronics as where technology meets user's needs. Traditionally this meant long planning cycles and large investments and a carefully crafted market.

In its place, we're going to see a new world of exciting applications as people are increasingly able to create and share new applications and devices by building on available resources. But first we need to escape the past and recognize that the accidental configurations of hardware and capabilities are just that – accidental. We need to take a fresh look at what is available to realize the new opportunities.