

Appliances vs. PCs: The Future of Computing Platforms

by Michael Slater, for Microprocessor Report, January 2001

In the past few years, a tremendous amount of attention has been focused on the impending shift to the “post-PC” era, in which the PC is replaced by myriad information appliances. This vision has many merits, but its realization is further in the future than its proponents have hoped—and it understates the role that general-purpose platforms will play in the future.

It is easy to hate the PC: there are many, many things wrong with it. Architecturally, the PC hardware design has reached an entirely acceptable level. Much of the legacy hardware (such as the ISA bus) has been purged from the current generation of systems, and 1394 and USB ports eliminate most of the drawbacks of the ancient serial and parallel ports. The PC’s biggest hardware drawback today is one that the industry should be able to fix, if it can break the inertia: the predominance of ugly, bulky, noisy systems.

When we get to software, there is a lot more to hate about the PC. Twenty years of incremental evolution have left us with a system that is much more complex and unreliable than it ought to be. Windows 2000 improves reliability significantly, and in time this technology will trickle down to the consumer market—but the complexity remains. And in PC application software, featuritis has run rampant and quality design with the real needs of users in mind remains rare.

And then there is the business element: Microsoft and Intel earn an awful lot of the profit made in the PC business, and their dominance can’t help but generate a lot of resentment.

As the Web rose to prominence and computing power became cheap enough to distribute into less-expensive devices, the multitude of reasons to hate the PC fueled enthusiasm for a new wave of application-specific, appliance-like devices. As I discussed in my September column, the idea of Web appliances, whether counter-top boxes or mobile tablets, has a lot of appeal—but these devices have largely failed in the market so far. The reasons range from the high cost of displays and the challenges of supporting plug-ins to the difficulty of finding early adopters for products essentially aimed at laggard markets (e.g., the Web appliance for the masses).

Another place where this model has run into trouble is in the idea of a Web browser as a do-all environment. Thirty years ago, computing was done mostly on large, shared computers—mainframes. Then PCs came on the scene, and by the time they had been around 10 years or so nearly all applications were moved off the mainframe and onto the PC. A few years later, with the spectacular rise of the Web, the pendulum—at least in terms of the design point for future products—swung too far back in the other direction. In the future, many thought, applications will run on the Web, data will be stored on the Web, and all we’ll need on the PC is a browser running Java. No more administration of local systems, no more overly-complex PC software, no more local state.

This was a nice theory, and its appeal was strong in light of all the good reasons to hate the PC and the PC business. But it won't succeed as a mainstream replacement for consumer computing for one simple reason: it compromises the user experience unnecessarily.

To fit in the "everything is on the Web" model, users must give up too much. The PC model has lots of drawbacks, but it also has a lot going for it—not the least of which is an installed base that now covers essentially every business desk and around 60% of homes in the U.S. In theory, starting over with Java-based programs would free us all from the PC standard and open up a new wave of innovation. In reality, however, Java programs tend to be annoyingly slow, and the sheer effort to recreate the key base of applications is massive. For a software developer, it is very hard to argue against building your program to make the most of the hardware that the vast majority of potential customers already have—and that, for better or worse, is the Windows PC.

At PhotoTablet, the company I started last spring, we have spent a lot of time talking with non-technical consumers, listening to their needs and desires and attitudes. It was surprising the degree to which they accepted their PCs, despite their frustrations. Consumers have come to accept PCs as a fact of life; they have made invested time and money in them, and it is going to take a very compelling proposition for them to switch to another approach. And the 40% of households without PCs are not a place to build a market; many of these households are saving for a PC and would see it as a failure to accept anything else, and the rest are too technology-resistant for any new product to have much of a chance.

Whether we like it or not, we're stuck with the PC as the platform that will dominate computing for the foreseeable future. The opportunity for appliances is not to replace the PC, but to complement it. Web tablets and other appliances will be sold, for the most part, not to people buying them as alternatives to a PC, but to people who want an additional access device for the Web. Handheld computers, one of today's most successful classes of information appliance, don't replace a PC, they provide another way to interact with information that is stored primarily on the PC.

If the PC is eventually displaced, it will be because the myriad devices that will surround and complement it will gradually become more powerful and more independent and then slowly make the PC less important. A frontal attack is impossible, however; appliances must start by complementing the PC, not replacing it. And no matter what happens, there will always be a strong role for a multifunction, general-purpose computer.

Since the PC will be with us for a long time to come, it is essential that we do everything we can to make it better. A lot is in Microsoft's hands: we need a more reliable, less complex operating system. But application developers play an important role too; they must focus less on feature wars, and more on meeting the user's needs in the simplest possible way. Applications must be built with deep knowledge of the customer, and not just as a way to deliver a technology.

The Web, as wonderful as it is, has brought new challenges. With all the attention focused on the Internet, fewer resources are going into PC applications. (When was the last time you saw an exciting new application?) Consumers' time and attention are consumed by the Web, reducing their ability to absorb new applications. This shift, combined with an inefficient retail channel that pushed price points to unprofitable levels, has gutted the consumer software business. Ironically, the PC industry has managed to get very impressive hardware into the homes of the majority of U.S. consumers—and now few companies are focusing on creating new software to deliver on the potential of those systems. What a shame!

Creating great consumer software is a tough challenge. It requires a deep understanding of the consumers being served, which is all too rare among the software developers who tend to define programs as they build them. Once there is competition within a segment, product managers tend to focus on feature wars; a longer feature list is the easiest way to market a product. Unfortunately, more features often are not what consumers really need; they need the right features, better implemented. And even if you achieve all this, you must somehow get the attention of consumers, convince them that your product is the right one, and find a way to profitably sell the software.

Games are the only recent “killer applications,” in more ways than one, beyond email and the browser. Games appeal only to a minority of the population, however. The central challenge in creating new application categories is that people don't readily add new activities to their lives; they don't have the time. Photography is one mainstream consumer activity that is poised to move into the computer realm, thanks to the spectacular success of digital cameras. Hobbyists have made the switch, but ordinary consumers need better software. It is hard, however, to find other mainstream activities that are ripe for moving onto the PC that aren't already there.

The successful software today provides the computer version of most common activities: writing, managing finances, getting news and product information, and shopping. Creating breakthrough applications in these areas is surely possible, but it is a daunting business challenge to go up against Microsoft or Intuit in the areas that they dominate.

Significant near-term opportunities lie in finding new ways to benefit from Internet connectivity directly to applications, rather than via a browser, and in creating applications that bridge the PC to the emerging world of mobile devices. In the long run, applications that achieve the holy grail of natural language interfaces, and that are intelligent assistants instead of ignorant slaves, may be the most important—but no one yet seems to know how to create them.