



# Teaching Computing's Past

In order for lenders to get the most out of their systems they have to **know how they've evolved** and where that evolution will go next.

**MANY OF US GREW UP WITH PCS ON OUR DESKS.** But not me. I'm older than that. I think it's interesting and instructive to understand how small computing systems have evolved, and where we might be going from here. In the very beginning and through the 1960s computing was dominated by large mainframe computers housed in environmentally controlled computer rooms and monitored by experts in white lab coats. Computer memory was measured in kilobytes, not megabytes, gigabytes or terabytes, with an 8K machine being a true workhorse.

Using these computers was costly and required a great amount of expertise. Programming was done in languages like COBOL and FORTRAN, and work was usually done in batches rather than interactively. I'm old enough to have done my first programming in college by typing out punch cards and submitting them as batch jobs to a CDC computer operator, with my results printed by a "band printer" on green-bar pin-fed paper. In looking back on it, this era could hardly be called small-systems computing.



The first true wave of small systems computing started in the early 1970s when companies located mostly on the Route 128 beltway around Boston came to the forefront. Prominent computer makers from that bygone era include DEC, Wang, Nixdorf, NCR, Honeywell and Burroughs. These companies produced "shared-logic" minicomputers. There was one CPU, and computer memory and other resources were allocated to users, each of whom possessed their own keyboard and "dumb" CRT monitor for input and output, a vast improvement in computing power accessibility. Computer programs could consume a whopping 32K of memory. My company got its start in mortgage loan automation

in this era using Wang minicomputers, which could support a typewriter-quality printer connected to each terminal. You had to insert and carefully line up preprinted forms like the old 1003 into these printers or the forms wouldn't look right.

Home hobbyist computing also began in this era. Intel developed the 4004 microprocessor, followed by the 8008, 8080, 8086, and many others, leading to a large array of small computer companies, led by Altair. I clearly remember when my algebra teacher mother brought home a programmable Monroe calculator from school, giving the family a fun weekend together teaching it how to perform square root calculations by way of the long algorithm. Then my father

bought a \$2,000 Digital Group computer kit which he built from scratch and never did get to work before Digital Group closed its doors. \$2,000 was a whole lot of money back then. Tandy/Radio Shack had great success with the TRS-80 computer (Dad had one of those, too), and Apple Computer got its start with the Apple I and Apple II. These were all single-user computers and were distinguished by their very low cost, but they were considered toys and never accepted into wide-

spread business use. software on a lunchbox portable computer manufactured by a company called Dolch Computer Systems. They were inexpensive, lightweight and (for the era) powerful.

As IBM PCs and their clones became popular, there came a demand for ways to connect them together in networks to share information from a single database. This spawned the second wave of small systems computing: client-server computing. First networking hardware like Arcnet and

HTTP to be easily displayed on monitors of PCs using a new kind of software called a Web browser. NCSA Mosaic was one of the first Web browsers, and its author Marc Andreessen later helped found Netscape Communications Corp. Netscape quickly came to the forefront in Web browser innovation, adding interactive features that allowed simple applications to run over the new World Wide Web. Microsoft later introduced Internet Explorer with its new Windows 95 operating system, bringing the Web to home/everyday computer users who dialed into the Web with their computers connected to their home phone lines.

The ability to browse the Web and access simple applications like Yahoo's search site began the third wave of small systems computing. Web applications became more and more capable over time. Napster allowed you to download music. eBay became a popular auction site and spawned many small businesses. Small investors quit their jobs and became day traders on brokerage sites. This was Alan Greenspan's era of "irrational exuberance," and it culminated when the dot-com bubble burst in 2000. After that it became harder to fund your Internet start-up company, but the usefulness of the Web application deployment model was well established. Several companies developed mortgage loan applications that ran over the Internet in a Web browser; these were mostly borrower-facing applications. For example, back in 1998 I refinanced my mortgage on homeshark.com.

The earliest Web applications were quite crude. In the years since, two forces have caused us to witness a progression of more capable Web applications. First, competition caused Web browsers to become more standardized. This has made it easier for Web application developers to develop and test their software on Internet Explorer, Safari, Firefox, Chrome, Opera, and other browsers. Second, the rise of

The major benefit of client-server computing was that **a single database could be shared** and centrally managed just like in the mainframe days.

spread business use.

That changed in 1981 when IBM introduced the IBM PC. The IBM PC was not much different than single-user hobbyist computers at the time. One difference was that it was a bit more rugged than most, which came at additional expense. Another difference is that rather than the CP/M operating system that most home computers used back then, it used a new operating system called MS-DOS from a small start-up company by the name of Microsoft. But the major difference was that IBM published the PC hardware architecture to encourage hardware add-ons, spawning a new industry of hardware companies building new IBM PC features. It also spawned many companies making IBM PC clones, such as Compaq and Kaypro. Ironically, IBM never intended that the PC be used in business. That happened virally as back-office workers brought them into the workplace to accomplish computing tasks that their computer departments neglected.

The first "portable" computers were launched in this era – first the size of a suitcase, later reduced to "lunchbox" size. My company sold its origination

Ethernet made it possible to connect PCs into networks. Then network operating systems such as Novell Netware and Banyan Vines offered ways to turn one networked PC into a file server that other PCs shared. Finally, the network operating systems ran applications themselves, usually multi-user relational databases, and the connected PCs could run applications individually but share the database among them.

The major benefit of client-server computing was that a single database could be shared and centrally managed, just like in the mainframe days. But the application was still installed on every PC attached to the network, just as before when the PCs were not networked.

That was the 1980s, when local-area PC networks LANs gained widespread use. Then in the 1990s the Internet appeared and allowed all of those LANs to be connected together. A new data transfer protocol called HTTP emerged that permitted any Internet-connected computer to read information stored on others. And a new data format called HTML became popular because it permitted information served over

AJAX technology has made Web applications richer and more interactive.

Yet, Web applications are still significantly constrained today by the capabilities (or lack thereof) of Web browsers and to a lesser extent, by the Internet itself. While browsers are more standardized they still run differently in many ways, complicating Web application development. Internet bandwidth limits make Web applications slow to use.

The stateless nature of the Web and browser UI limitations make Web applications harder to use. The inability of browsers to directly support many common data formats like word processing documents and video is causing a fragmentation in implementation methods. Today powerful vendors offer rich Internet runtime environments like Adobe's Flex, Microsoft's Silverlight, and Sun's JavaFX, but such applications run only on PCs where these environments are installed, which is often not the case on consumer PCs. Web application developers who wish to provide a richer user experience are forced to choose among these incompatible environments and run the risk their applications just will not work on many consumer PCs. That is where Web applications are today, with a deployment model that is again fragmenting and working against the ideal of universal deployment.

Yes, Web applications are easy for your IT department to deploy because unlike client-server applications, all application programming and databases are centralized on a single Web server. But Web applications are still ill-suited for all-day use in a business setting.

That's why we are now seeing the emergence of a fourth wave of small systems computing: application virtualization. Many lenders that deploy internal B2B applications using Web technologies are finding them difficult to develop and use, due to the aforementioned Web browser and other

limitations. It is just too darn hard to originate, underwrite, process and close a mortgage loan with a Web application.

What businesses are realizing is that their goal was not to use Web applications per se, but to use applications that are easy to deploy and administer, as Web applications are. What businesses are finding is that application virtualization provides them the ease of centralized administration, while at the same time supporting the rich applications they want or already use. Application virtualization works somewhat differently than Web deployment, however.

Virtualization requires the business to use specialized software on both

What businesses are realizing is that their goal was not to use Web applications per se, but to use applications that are easy to deploy.

their server and client PCs. For the client PC this means installing a small client program, or using a client applet inside their browser. Once they have done this, any application may be run on the client PC without having to be installed there. Application updates are applied only once to the server and are then used at every client PC without visiting the desktop.

Application virtualization is provided by vendors like Citrix on Windows, NoMachine on Linux, and several other vendors. Even Google is now getting into application virtualization by releasing Neatx, an open source NX server similar to NoMachine's. In fact, the competition among virtualization vendors now assures that the requisite software will always be inexpensive and readily available.

Businesses that ride this fourth wave of small systems computing are finding their systems to be less costly, easier to develop and manage, and most importantly, much easier to use.

Application virtualization will not displace Web applications in the B2C world because B2C applications are only occasionally used and because Web browsers are ubiquitous. But application virtualization is likely to displace Web applications in the B2B world, especially those applications that are used on an all-day/everyday basis.

Now, what's old is new again. Client-server LOSs have always been robust and easy to use. Now, thanks to application virtualization, they are also easy to deploy and administer. Vendors developed their LOS with the needs of the business user uppermost are now being rewarded. Vendors who compromised their business users to make

life easy for the IT department will find waning acceptance of their LOS.

Last June I celebrated my 30th year in the automated lending business. It has surely been a roller coaster. I've seen boom times and bust times and I've seen computing trends come and go.

One thing I have learned is to always consider the needs of the actual end-user foremost. The technology needed to support end-user requirements and ease-of-use will inevitably emerge, as it has in wave after wave. Ease-of-use and robust functionality will always drive technology. It doesn't happen any other way. **MT**

*Tim Liston is president of Associated Software Consultants Inc. (ASC), developer of the PowerLender and PowerSeller systems now used by mortgage lenders nationwide. He co-founded ASC in 1979 and has at one time or another served ASC in almost every capacity, including sales and marketing, administration, and product planning.*