

"Open" Systems: WHAT ARE THEY?

by
Scott McNealy
President and CEO
Sun Microsystems, Inc.
Mountain View, CA

In recent years, most companies that built their fortunes on single-vendor proprietary computing strategies have jumped on the open systems bandwagon. And why not? They've heard the market. Customers are voting for open systems with

their wallets. In droves. They don't want to be locked into the fortunes of one computing architecture or company.

But while the demand for "open systems" is clear, the meaning of the term isn't — at least not anymore. Most proprietary companies singing the "open systems" theme song these days sound suspiciously like they're crooning "My Way" in the background. How truly open are their "open systems?" How committed will they be to open systems in the long run if those systems are stealing sales from their proprietary lines? Or if adopting open systems means giving up control of a lucrative product?

What It Takes to be Open

"Open" is the most used and abused word in computing today. It's become an all-purpose marketing buzzword to sprinkle liberally on virtually any product or strategy. It should be placed on the Endangered Jargon list! Some common misuses include:

"My computer architecture is open because I can network to and get files from the mainframe." This confuses interoperability and portability with openness.

"My computer is open because it uses UNIX." This confuses ease of software portability with openness and doesn't even begin to cover all the other hardware and software interfaces developed by a vendor.

"My PC is open because it runs binary-compatible software from one PC vendor to another." This confuses openness with compatibility — and compatibility with logos. A PC clone is still an Intel/Microsoft computer no matter what distribution channel you buy it from or what logo you put on it.

"My computer complies with standards because it conforms with the XYZ consortium." This confuses openness with press releases. More on this later.

Like many other vendors, Sun is committed to all the criteria listed above. But these are not casual relationships — you can't add up heterogeneity, binary compatibility, UNIX, etc. and have the total equal "open." They are not the same thing.

"Open" is a term worth saving, though. It says a lot about the way computer companies can and should compete; and about the value provided customers through innovation. The theory behind openness is that all technology can benefit from improvement and that all customers benefit from such innovation. Ultimately, the industry benefits, too, as a result of increased demand brought about by more useful products. Basically, openness is a simple concept. It refers to a specification, a blueprint — or as we call it in the computer business, an interface. One that's written down for the world to see and use. It invites improvement and innovation.

The brake pedal is, for example, an open interface in the automobile industry. Every car company knows the description of a brake pedal: it's the three by six-inch pedal located directly to the left of the accelerator that can halt the car when it's depressed

Western Canada Wilderness Committee

Winner of Canada's 1991 Environmental Achievement Award

invites you to

Celebrate Wilderness!

Featuring

Monte Hummel

President of World Wildlife Fund Canada

Norma Kassi

MLA for Yukon

Paul George

Founding WCWC Director

Adriane Carr

WCWC Director of WILD Campaign

useful products. Basically, openness is a simple concept. It refers to a specification, a blueprint — or as we call it in the computer business, an interface. One that's written down for the world to see and use. It invites improvement and innovation.

The brake pedal is, for example, an open interface in the automobile industry. Every car company knows the description of a brake pedal: it's the three by six-inch pedal located directly to the left of the accelerator that can halt the car when it's depressed with x foot-pounds of pressure. Car companies don't pay royalties to one another to use the specification; drivers are comfortable with it because they know the brake pedal is always located in the same place.

The brake pedal specification is a volume open interface. And the automakers implement and innovate on the brake pedal's open interface with competitive products like anti-lock braking systems, hydraulic brakes, disk brakes, etc. Innovative implementations — even of things as mundane as braking systems — enable car manufacturers to differentiate themselves. Having these open interfaces lets automakers improve upon and innovate while still complying with the open interface. And customers don't have to relearn how to drive.

In the computer industry, interfaces describe many of the important elements that tie a computer system together: the microprocessor architecture, the system bus, the operating system and window systems, the graphical user interface, networking and others like the important ABIs (Application Binary Interfaces) and APIs (Application Programming Interfaces).

The fundamental premise in defining openness is a positive answer to a simple question: "Can this technology benefit from innovation; from being faster, more full-function and cheaper?" If you believe the answer is yes, then we can apply the

SPOTLIGHT

question to an actual case — like Microsoft's Windows. Could Windows improve through innovation? Everything I've heard in the market says so. But while Microsoft gives developers the information to create applications that run with Windows, the company does not provide the information to innovate on Windows itself. As a result, Windows is a proprietary (not open) interface.

Here's a simple checklist of criteria for openness:

- 1) The interface must have a specification that's openly available, meaning, published. This allows other vendors to create their own, value-added implementations. By contrast, most vendors release implementations — such as Windows — without making the specifications available. There's a difference between being able to develop applications for the Windows environment and being able to innovate and improve upon Windows itself.
- 2) The specification should be well-written. You should be able to send four — or four hundred — engineers into a room with a good interface specification and have them come out with a reasonable implementation. This is important. Well-written specifications encourage the creation of multiple, compatible implementations. . . a.k.a. competitive products.
- 3) An open interface is available inexpensively or free. No huge royalties. Lots of innovation.
- 4) The interface should be legally clean, i.e., you can't be sued for using it. Who wants to build a product line around a hardware or software interface that will make only lawyers rich?
- 5) Multiple implementations of the interface should be available from multiple companies. Firms that really embrace open interfaces endorse the value of competition. Rarely will there be just one of anything.
- 6) There should be a reference implementation available for a reasonable fee. Seeing how the interface is actually implemented is important to people who are building their own implementations. It's easier to build a car if you've seen one.
- 7) And, finally, there is — best case — an independent organization that performs branding and compatibility testing to ensure the specification is protected and enhanced in an equitable way.

Lots of vendors equate open systems with standards. They're not the same thing. The dictionary definition states that a standard is "regularly or widely used. . . the preferred way of doing things." Standard interfaces are set by high-volume sales (in other words, by users). This means that the Macintosh is a standard (high-volume) but not open (no published interface). Of course, Sun's philosophy is that a robust open interface will be adopted by many vendors and users, eventually achieving the volume to become a standard.

Because standards are set by volume sales, it's illogical that standards can be set by committee. It's also illogical to call organizations like UNIX International, OSF and IEEE "standards committee." They are endorsement bodies and such bodies don't set standards. In theory, if enough companies joined a committee and swore to adopt a given set of technologies, then it's possible that their combined volume could drive standards. Practice proves differently, however. Not only have such companies been unable to predict their sales, they can't even predict their ability to follow through and commit to the consortia's initiatives. Consortia have succeeded in creating customer doubt and paralyzing innovation.

Continued on page 95

SPOTLIGHT

Continued from page 91

What really matters in the end is the customer's vote via purchase order.

Open Systems Lead to Innovation

Open interfaces can be used by more than one vendor; multiple implementations are encouraged. We're talking freedom OF choice here, not freedom FROM choice. It's a scary notion to many companies in the computer business — that's why they try so hard to obfuscate the facts.

Unfortunately for users, the concept of open interfaces doesn't characterize much of today's market. With most hardware and software interfaces, one company is essentially in control of its own proprietary product, thus limiting innovation. Imagine if DOS had been put into the public domain. We'd have realtime DOS, multi-tasking DOS, multi-threaded DOS or DOS with X Windows on top of it. System vendors could pick the implementation, price and time to market they needed.

Controlled Economies vs. Competition

Open systems is the riskiest strategy for a vendor who doesn't excel at innovation and execution. If customers have more than a single source of supply, they will base their buying decision on the product that gives them the best performance, value, customer support and other important aspects. "Security through obscurity" is out the window.

True open systems offer so much for end users that the future is clear; in a few short year, closed interfaces will be forgotten relics from a very interesting past. □

A Killer Quote

When UnixWorld asked Scott McNealy what he and Bill Gates discussed in a recent meeting, Sun's president replied. "I could tell you, but then I'd have to kill you."