

the *Availability Digest*

www.availabilitydigest.com
[@availabilitydig](https://twitter.com/availabilitydig)

Openness Was Not Always Ensured March 2018

An open system is able to interoperate easily with other open systems at both the application and the data level. The attribute of *openness* in systems was not always ensured or even accepted. This was brought to my attention when I came across a paper I had written years ago titled “Can the Computer Industry Truly Support Openness?”



The paper made several assertions that seem surprising in today's technology. It noted that computer hardware vendors placed great value in “owning” an account. They were reluctant to open up their systems for true interoperability with the products of competitors. After all, open standards would make it easy for IT managers to switch products since any product that complied with the standard could be substituted for another such product.

Before Open – Islands of Information

Back at the time the paper was written, different pieces of mission-critical data and applications resided on incompatible systems, forming islands of information. Relocating those islands of highly volatile data that was being continuously updated sounded ideal, but the process was likened to changing a tire on a car moving at sixty miles an hour.

This left open interoperability between incompatible systems as the only route to enterprise integration. Paradoxically, for a machine to be considered for purchase at the time, its manufacturer had to have an “open” story. However, the only openness that was generally provided was workstation access to legacy platforms. Anything more open was threatening to hardware vendors.

To effectively benefit enterprise-wide, mission-critical applications, interoperability had to encompass interprocess messaging and remote database access across all of a company's systems, including open access between legacy mainframes.

Interoperability Standards

True openness occurred only when common interoperability standards were adopted by all computer vendors. Back then, there were accepted standards such as C, SQL, X.25, and TCP/IP. All of these were created as defacto standards and were accepted by the industry before standards committees converted them to formal standards.

Unfortunately, at the time there were no defacto standards for interoperability. The need was just too new. The Distributed Computing Environment (DCE) was looked upon by many as the new interoperability standard, but its high complexity prevented it from being adopted by all but a few pioneers. CORBA, the object-oriented standards body, was another possibility; but the group couldn't agree on specifications and gave up on true interoperability. X/Open was adopting standards, but few vendors offered

implementations. ODBC served as the SQL database standard for interoperability, but it totally ignored legacy databases where the bulk of corporate data resides.

Openness Today

Fortunately, in today's technology, this has all changed. Openness is taken for granted. Openness means that computer programs have some combination of portability (they can run in a variety of environments) and interoperability (they can exchange data with each other).

Open software must adhere to open standards. This means that the software should be free for use, and its source code should be defined by its community of developers and users. The most prominent example of open software is Linux.

Windows, by contrast, is closed, or "proprietary." It is not portable and possesses limited interoperability. It must be licensed, usually for a fee; and its source code has been compiled and hidden from users and developers outside of Microsoft. Developers write to application programming interfaces (APIs), which Microsoft jealously guards.

Ever since the emergence of the Web, whose multitudinous pages are themselves created with open standards, information technology has tended to become more open. Increasingly, software companies stress their openness. Sun Microsystems' Java platform, widely used to create software for devices as different as embedded systems and supercomputers, has been portable and interoperable since it was launched in 1995.

Summary

Openness abounds in today's software technology. Java is the open standard for programming languages. SQL is the open standard for databases. And x86 is the open standard for system architectures. As we move into the future, I anticipate that all software applications will be implemented with these open standards. Hardware systems will comply with them as well. Thus, any application will be able to run on any hardware. We will have achieved the ultimate in operational flexibility.