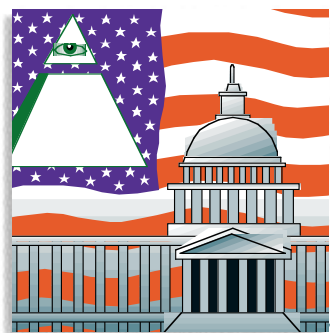


The Promise of Open Documentation



The Open Group's call for commonality of user documentation is mirrored in the federal government's JCALS standard.

One of the most significant events that occurred at this year's UniForum Conference in San Francisco was the announcement of the consolidation between the Open Software Foundation and X/Open Co. into a new organization, The Open Group. The press release detailing this new organization set forth five "market-focused objectives." One of these objectives is the development and implementation of a project that unifies the source and style of open systems vendors' delivered documentation, with the aim of demonstrating the "commonality of the user documentation efforts of all open systems vendors."

The intent of this is to help the Unix market (translation: sales) by having a standard for delivered documentation. The supposition is that, even with a standard set of application programming interfaces (APIs) for the operating system as specified in the Single UNIX Specification from X/Open, and a standard desktop interface as defined by the Common Desktop Environment, it's still necessary to bind the documentation (pun intended) into a unified standard from all Unix vendors.

Fred Dalrymple (fed@veloce.com), an independent consultant for X/Open, says that the goal of the common document effort is the definition of an identical set of documentation from the various vendors, which will enable others, including resellers, to integrate it more easily into their products. It will also enable end

users to choose a new open systems vendor and expect documentation similar to their previous vendor's.

Dalrymple personally does not feel that standard documentation will make or break a deal. But if a reseller has a standard operating system API as described in the Single UNIX Specification and has a common set of documentation across its product line, the impetus to develop in the open systems/Unix environment will be greater.

JCALs

Obviously this is a laudable goal. Not everyone is aware that the federal government has had a project ongoing since 1987 to accomplish a similar purpose: a standard for documentation delivered to customers. This project is the Joint Computer-Aided Acquisition and Logistic Support (JCALS) standard, sponsored by the U.S. Department of Defense. (*Joint* indicates that it serves the joint defense commands of Army, Navy, Air Force and Marine Corps.) The program is designed to enable more effective generation, exchange, management and use of digital data that supports defense systems. The primary goal of JCALS is to migrate from manual, paper-intensive defense system operations to an integrated, highly automated acquisition and support process.

Major defense systems are not unlike major software projects in the commercial market. They are developed at great

expense, usually by outside firms under contract, and have a very long operational life, often measured in decades. Support of these systems is both critical and difficult. JCALS states that, in order for the various services to properly maintain the complex software found in imbedded weapons systems, it needs a standard for delivered documentation. In the case of JCALS, it's the Standard Graphic Markup Language (SGML).

Before going any further, let's consider the problem the Department of Defense is faced with. Let's say it contracts with an integrator to develop a particular weapons system for a fighter aircraft, such as the F-15. Around the same time, another integrator gets another contract to develop a different weapons system for the same aircraft. A third vendor develops a different system, and so forth.

Let's put ourselves in the place of the weapons engineer or technician who must troubleshoot any problems in these three weapons systems. We're given at least three different sets of documentation, with different styles and definitions. Each system may be self-contained in terms of operation but not in terms of maintainability. The poor technician who is responsible for keeping that F-15 flying has a much harder job in repairing complex systems when their documentation is not standardized.

This situation also applies to systems less imposing than custom-designed weapons systems. Think instead about commercial off-the-shelf (COTS) software. The military is trying to convince Congress that it is moving toward the use of COTS products and not spending taxpayers' money on developing "military" versions of products that could be purchased over the counter (or off the shelf).

In fact, there is a Navy submarine project under way, for which an admiral associated with it is telling Congress that over half of the software on this submarine will be COTS. I hope someone has mentioned

By Gary Donnelly

to that admirer the problem in getting commonality of documentation across the installed products from different vendors. When you're a mile under the surface of the ocean, you can't exactly call a firm's 800 support number for technical help. Having a common basis for the third-party software's documentation is not only desirable, it's critical. This is the problem JCALS is addressing.

Project participants have given a lot of thought to the open systems milieu. The JCALS architecture has been designed to be a distributed, open systems environment that makes extensive use of both industry and government standards. These standards include the use of Posix-compliant operating systems, the government's Open Systems Interconnection Profile (GOSIP, for those who want to build international standard networks), TCP/IP (for those who want to build networks today), the X Window System and Motif for user interfaces, and the Ada language for software development. The technology developed in JCALS should be exportable across different platforms as its use widens.

Commonality of Goals

What's the tie-in between the Defense Department's JCALS and The Open Group's common documentation project? First, there is the obvious open systems connection. Both projects are designed for Unix-based systems; both base their documentation standard on SGML; both realize that while it's the software that sells the product, it's the documentation that counts in the long run; and both understand the concept that unless the software is maintainable, the operating costs for that project will be unprofitably large.

By having a common set of documentation for the delivered software, whether it's the operating system or the end user's applications, there will be a clear benefit to the customer. The client base will be able to move across vendors'

offerings without having to worry about the format of their documentation, either for resale or repair.

In the desktop market, one usually assumes that the vendor's documentation will be (A) not very useful and (B) not like that of the other vendors. Often, we simply assume we can go the bookstore and buy yet another vendor's product: a publication specifying how to use the previous vendor's product. This might work in a limited environment, but (as the pundits say) it doesn't scale well. For customers who have to support thousands of systems in which software from dozens of vendors is used, the only answer is a common set of documentation.

The X/Open development team, according to Dalrymple, has borrowed from the work developed in the JCALS project. Perhaps the JCALS effort should also look at where the industry is going and see if its technology can be applied to something other than large weapons systems. If we can have an environment in which we truly have portability of the documentation along with the software, we will have gone a long way down the highway to open systems. **IT**

***Gary Donnelly** teaches and consults in the client/server and open systems arena, focusing on federal marketing issues. He can be reached at gary@donnelly-inc.com.*