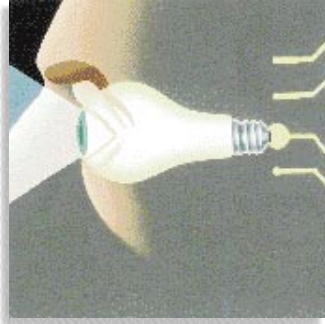


Creating and Implementing Database Standards



If users don't articulate what they need in database standards, they may find themselves with standards that don't solve their problems.

The idea of standardization within the database arena is not new to the IT industry. All sorts of standardization activities have been focused on this arena, probably the best-known of which is the Structured Query Language (SQL) standard produced by the Accredited Standards Committee (ASC) X3 H2 (the database standardization committee of ASC X3). It has been estimated that the successful standardization of SQL was one of the reasons for the success of relational database management systems (RDBMSs) a decade ago. (The standard used was the Standard Query Language from IBM.) While not actually achieving much in the way of true interoperability, the SQL standard provided the market with the assurance that there would be a second source for any relational database. The success of the standard as an implementable piece of technology can be traced to the efforts of the SQL Access Group (SAG), which took the standard, cleaned it up and provided implementation details for companies that would use it.

The standardization effort played an important role in this process. It did not serve as a technical guide to open systems, standardizing the way the databases achieve interoperability. Although this was its original purpose, standardization rarely achieves such results. More often, the standard serves a business function rather than a technical one. In the case of relational databases, the function of the standard was to assure potential pur-

chasers that there were other vendors who could provide a similar capability to the one being bought. This—not the standardized technology—helped move the market to relational databases. (This analysis, of course, ignores the technology and the market need for relational databases in the first place.)

However, I believe that if the standardization had not been in place, one of two things would have happened: either a single major vendor would be providing the technology now (in the manner of Microsoft) or there would be multiple smaller vendors, each with a unique system of doing something, resulting in a fragmented market for relational databases. Another possibility might have been that a different technology would have appeared, around which the majority of vendors would have coalesced. In any case, the success of the market would have depended upon the ability of the vendors and suppliers to work together to solve a business problem.

The proof of this concept will be, of course, whether the same tactic can be duplicated by the market. There is currently a move under way to standardize object-oriented databases. The leading activity in this area has been undertaken by the Object Database Management Group (ODMG), which created a standard for object-oriented databases in 1993. ODMG-93 was published for use and implementation by the industry, with commitments from seven object-oriented database vendors to implement the standard

in 1995. The consortium is now beginning to work with other consortia (primarily the Object Management Group) and with X3 J16 (C++), and X3 H2 to ensure that the work on ODMG-93 is made available to a larger audience for study, review and implementation. The test of the theory will be if the market for the object DBMS expands and both users and providers begin to accept the activities of the ODMG. (For a further description of the activities of the ODMG, see Rick Cattell's account in "Experience with the ODMG Standard," *StandardView*, Sept. 1995, pp. 90-95. It is available from ACM at 800/342-6626.)

DRDA and The Open Group

At the same time as this work, there is a database interface protocol standardization effort occurring in The Open Group, that amalgam of the Open Software Foundation and X/Open Co. In 1995, in a move that surprised X/Open and many independent software vendors (ISVs), IBM announced that it was giving its proprietary distributed relational database architecture (DRDA) specification to X/Open for standardization. The activity kicked around for a while, because X/Open wasn't used to having its technical program driven by single-vendor press announcements. After a while, DRDA activity ended up in the Open Group marketing committee, where I chaired the DRDA task force.

The problem with DRDA was complicated. The RDA standardization effort in ISO and X3 had not been successful, and the earlier effort in X/Open had been largely abandoned. The market demand for these earlier standardization efforts never manifested itself in user demands—the market seemed to be coping with the problem of disparate database interface protocols. The task that faced X/Open was threefold. The first question was to determine if X/Open was the proper forum for this type of standardization; that is, a proprietary specification with a large installed base. The second question was whether or not the market needed a database interface protocol. The third ques-

By Carl Cargill

tion to be considered was whether DRDA was the right one.

The first answer was an easy yes; X/Open had, after all, accepted the Unix specification for standardization. On the next question, there was substantial disagreement on such topics as gateway, user demand and market size. As this was to be a marketing/business issue, the primary consideration was to be given for a marketing case. The real problem here was not so much the technology standardization—that is, the actual modification of the IBM DRDA specification to make it more open—as the issue of the need for the activity at all.

The major database vendors (Informix, Oracle and Sybase) pointed out that their customers were happy with the current situation. IBM and Starware pointed out that the gateway process was inefficient and costly. The retort was that the database vendors and the market have been seeking the “silver bullet” for interface standardization for years and never managed to find one. The discussion became heated, and ultimately the task force voted to reject the submission. The rationale for rejection was simple: There wasn’t enough consensus on the proposal to succeed.

Whose Problems Get Solved?

Two real issues implicit in standardization surfaced in this discussion. The first lay with the nature of the question being solved. The solution provided by IBM was exactly that; a solution proposed before the users were asked if there was a problem or if the IBM answer was correct. The question that came to haunt the discussions was What is the problem? IBM and Starware posited the problem as one that DRDA could solve; the other vendors posited the problem as one that was being solved by gateways. Neither side looked at the issue of user needs; one was going to change the market because they had the answer, and the other was content to let the market react and let the users pay a price in inefficiency. The real question that should have been answered was What is the problem that the users want solved, and how would my solution

help them be more (or less) _____ (enter any one of many user requirements here)?

The second issue derived from the fact that there was no consensus possible in this case. The two sides started from installed-base positions; IBM with its proprietary DRDA base and Informix, Oracle and Sybase with their own solutions. Essentially, IBM put the DRDA issue into X/Open and hoped that the organization would bless it; the ISVs objected, and a “standards war” almost began. Consensus—while not necessarily indicating unanimity—indicates that a majority of the market accepts the standard and will not fight against it. This wasn’t the case.

Database interface protocol efforts seem to have ground to a halt within most standardization organizations. The problem is that users have not clearly elucidated their needs in a way that providers can hear. While this may be changing, it

still remains a major problem for all standardization groups. What users should do—especially those in UniForum, who are, after all, most concerned with open systems—is to state functional requirements collectively and clearly. Part of the response depends on how much you’re willing to pay and when you want it, as well as whether you want a “silver bullet” or would be content with a solution that, like SQL, is “good enough.” The important thing to remember is that users validate standardization and ultimately cause it to happen or not. By not participating, users give away their ability to structure the future direction of the market to either the government or the providers.

IT

Carl Cargill is standards strategist at SunSoft in Mountain View, CA. He can be reached at carl.cargill@eng.sun.com.

COMPANIES MENTIONED IN THIS ISSUE

FEATURES		COLUMNS	
Raise Shields!:	p. 20	Breaking the Curse of Distributed Environments:	p. 30
ANS		BMC	
BBN Planet		Cabletron	
Border Network Technologies		Compuware	
Checkpoint Software Technologies		Hewlett-Packard	
Cisco Systems		IBM	
DEC		Illuminata	
Harris Computer		Innovative Software	
IBM		LPB Systems	
Livingston Enterprises		OpenVision	
Morning Star Technologies		Platinum Technology	
Network Systems		Remedy	
Oracle		Sun Microsystems	
Raptor Systems		Tivoli	
Secure Computing		Unison Software	
Sun Microsystems			
TIS			
V-One			
Attack of the Rampaging Data:	p. 24	Behind the News:	p. 12
Data General		AT&T	
Dimensional Insight		Computer Associates	
Evolutionary Technologies		Hewlett-Packard	
Hewlett-Packard		IBM	
IBM		Lucent Technologies	
KPMG Peat Marwick		Microsoft	
Oracle		Oracle	
Price Waterhouse		SAP America	
Red Brick Systems		Sun Microsystems	
SAS Institute		Tivoli Systems	
Sun Microsystems			
		Information Technology Demystified:	p. 36
		IBM	
		Oracle	
		Standards & Technology:	p. 46
		IBM	
		Informix	
		Oracle	
		StarWare	
		Sybase	
		The Analyst's Couch:	p. 56
		Compaq	
		IBM	
		Intel	
		Silicon Graphics	
		Sun Microsystems	