Universal Business Language: Checkup Time for an XML Vision

Laurianne McLaughlin

XML today routinely helps businesses and governments write applications and do Web-based tasks that were beyond most people’s imaginations back in 1999, when the inklings of an open standard called Universal Business Language first appeared. Crafted by the Organization for the Advancement of Structured Information Standards, UBL provides a royalty-free set of XML-based business documents—such as purchase orders—for e-commerce invoicing and ordering. The UBL’s original goal was to fuel global e-commerce by simplifying paper-heavy processes.

UBL 1.0 became an OASIS standard in November 2004; UBL 2.0 appeared in December 2006. The OASIS UBL effort continues to boast a distinguished technical committee. However, if you’re not familiar with UBL, you’re not alone. The standard hasn’t become well known in the US, nor has it gained traction with US businesses, industry analysts say. The announcement of UBL 2.0 in December made little news.

A difference of opinion

So, what’s ahead for this standard? That question remains open for debate. UBL’s founders say it will play an increasing role for governmental agency IT groups trying to cut the costs of paper-based processes. They point to early successes along these lines in Denmark and elsewhere in Europe.

But some industry analysts call it a well-intentioned standard that simply has become less relevant with time. They believe it’s less important to businesses than other standards around service-oriented architecture (SOA) and Web services, and less key than XML standards related to particular vertical industries such as healthcare.

UBL is one of several “overreaching industry standards that will never gain a meaningful presence in the business world,” says Ken Vollmer, a principal analyst at Forrester Research.

What do UBL’s crafters think of this notion? Jon Bosak, chairman of the OASIS UBL Technical Committee and a lead player in XML’s development, says the goals for UBL have changed a bit, but it still has plenty to offer.

“Vertical-industry standards are more important in the US right now because the verticals had existing EDI [Electronic Data Interchange]-based supply chains in place and because US policy in this area is driven more by corporate interests than by government interests,” Bosak says. “As government bodies in the US—particularly state governments—start taking notice of the savings being demonstrated by UBL adoption in Europe, this will start to change.”

A big vision

From the start, Bosak and the other people working on UBL’s technical aspects signed onto an ambitious job. The effort’s scope crosses national boundaries and business domains. Given the pace of today’s business world,
where quarterly results often dictate actions, complicated open standards that take years to craft face tall hurdles to big success.

And like many industry standards that are crafted in an open process, with much time for input and comments from technology vendors, academics, and interested end users, UBL took quite a while to move from the oven to the table for public consumption.

In 1999, OASIS began working on a set of standard XML office documents. Then OASIS started work on a related standard with the United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT, www.unece.org/cefact). The Electronic Business Using Extensible Markup Language provides standardized XML-based versions of the technologies behind traditional EDI; the one exception is an XML equivalent to EDI's standard X12 and EDIFACT's message sets. That's the gap that UBL fills, Bosak says. ebXML (also known as ISO 15000) seeks to make global commerce simpler for businesses no matter what their locale or size.

An ebXML discussion group led to UBL, and the OASIS UBL Technical Committee was born in late 2001. It sought an XML flavor of certain EDI standards, which companies had been using since e-commerce's early days to exchange sales and supply-chain information.

The challenge, says Tim McGrath, vice chair of the UBL Technical Committee, was to harness the value of much legacy EDI work but exploit the advantages of newer technology to create a well-defined yet versatile standard for business documents.

The committee based its work on a successfully deployed group of XML specifications, xCBL 3.0, which were shaped by CommerceOne and SAP. According to Bosak, this choice let the group start with a mature de facto standard that was already successfully deployed in a number of e-commerce marketplaces. Also, the standard was based on the concept of a component library—an essential idea to UBL's design philosophy. “In the six years since then,” he says, “we've grown UBL in an open-standards process to become a fairly complete solution for a wide variety of generic supply-chain processes.”

**UBL 2.0 adds government twists**

While UBL 1.0 covered the eight supply-chain forms most often used in the business process between ordering and invoicing, UBL 2.0 added 23 trade documents related to a full system for government procurement. These documents covered steps such as sourcing, ordering, fulfillment, billing, and payment.

UBL 2.0’s library offers more than 1,000 XML data elements, based on the ebXML core components technical specification.

The end result is what Bosak calls an “integrated suite of message types that doesn’t meet every conceivable e-commerce need but is comprehensive enough to support a huge number of ordinary commercial transactions.”

UBL 2.0 also provides a way to manage code lists that Bosak feels will be capitalized on by other XML standardization projects.

For examples of UBL in action, you need to look to northern Europe. While some people might judge UBL’s success by its adoption rates, Bosak proposes another measure: the amount of money UBL is saving early users, such as Danish and Swedish government agencies. (Adoption rates, unfortunately, are hard to judge. As with some open source software that’s freely licensed, industry analysts don’t like estimating UBL user numbers.)

In Denmark, which has mandated the use of the UBL Invoice document for all public-sector invoice exchanges monthly, Bosak says. This saves the Danish government an estimated €100 million annually, he adds. This year, the Danish government has begun a larger effort to deploy 12 UBL 2.0 documents into Danish businesses, eyeing a cost-saving national e-commerce infrastructure.

Since October 2005, the Swedish National Financial Management Authority has been recommending Swedi-invoice, a subset of UBL Invoice, for all government use, Bosak notes. He says that the Swedish government hopes to save four billion kroner (more than US$500 million) in the first five years of deployment.

**Slow US adoption**

No similar UBL projects have appeared in the US. Why has UBL adoption been greater in Europe? The EU prioritizes large-scale e-commerce issues, and cross-border e-commerce integration has been on the EU agenda for years. Bosak also says that European governments’ organizational structure makes adopting national technology initiatives simpler than in Canada and the US.

In addition, the OASIS group itself has been focusing much more on Europe, with little marketing or outreach in the Western Hemisphere, says Bosak. “Now that UBL 2.0 is out, I think you’ll see that start to change,” he says. Also, noting that Denmark approaches the size of some US states, he says, “I think that we can expect to see government adoption of UBL in North America happen first at the state or provincial level rather than the federal level.”

But what about business use beyond government agencies? Governments have long been using standardized documents such as tax forms, Bosak says, so “the adoption of UBL is a natural extension of what governments do anyway. The same applies to big businesses, the ones that are big enough to dictate standards to their trading part-

**Like many industry standards crafted in an open process, UBL took quite a while to move from the oven to the table.**
nents—though as far as we know, there haven’t been any UBL adoptions of this kind yet.”

Businesses doing transactions with governments that adopt UBL might spur interest by software vendors, Bosak says. The turning point will likely be when a critical mass of businesses have implemented the same support and realize that they can automate their work with each other, he adds. “That hasn’t happened yet, but the effect when it does will be something like what happened when HTML and HTTP clients achieved critical mass,” says Bosak.

**ebXML to have wider legacy**

While UBL has struggled to win attention, ebXML has won wider adoption, analysts say. Visit the ebXML Forum Web site (www.ebxmlforum.net), and you’ll see recent examples of work with this technology standard—although again, many of these are from government agencies.

The US Centers for Disease Control are using ebXML to communicate with local public-health agencies, hospitals, and related professionals in the Public Health Information Network system. PHIN is designed to ensure quick communication when an infectious disease might be spreading.

Hong Kong’s port uses ebXML messaging standards to help fight terrorist activity involving shipping containers; a system used to exchange dangerous-goods manifests went live in 2003.

Trygdeetaten, Norway’s national insurance program, handles matters such as government-funded healthcare payments. It’s using ebXML messaging to transmit business messages.

The ebXML Forum Web site also includes a white paper detailing how T-Mobile International used ebXML in a B2B (business to business) gateway with communications industry companies.

**Too late to think big?**

While ebXML support continues to pop up in high-profile product lines such as IBM’s recently revamped WebSphere portfolio, UBL’s public profile hasn’t gotten much luster of late. Technology industry analysts don’t paint a rosy picture for UBL. In fact, some say that like other well-crafted and well-meaning but broad industry standards, UBL’s time has come and gone.

“It [UBL] took too long,” says Ronald Schmelzer, a senior analyst and founder at ZapThink, a consultancy specializing in XML, SOA, and Web services. “It was too grandiose.”

“Basically, businesses are having a hard time getting their internal SOA processes straight,” Schmelzer says. “While it would be nice to reduce this inefficiency of working with other companies, that’s not top of mind right now.”

UBL faces a hurdle that many industry standards can’t leap right now, Schmelzer says. “It’s hard now to get people to look at these wide-ranging specs. Businesses are using industry-specific vocabularies instead.”

Narrower industry standards crafted specifically for the healthcare and insurance industries, for example, have become more important than UBL and will continue to play a more significant role in e-commerce, Schmelzer says. He points to HL7 (Health Level 7), a data interchange standard used by medical institutions, and Accord, used by insurance companies. “It’s easier to get smaller groups to agree amongst themselves,” he says.

“The whole premise of standards adoption is you get agreement amongst a large group of people,” Schmelzer says. “The idea of a customer to a bank is very different from a hospital or a government. You end up lumping everyone’s definitions into one large spec, and it’s too complex to use.”

If you’re trying to facilitate communication, there’s a reason we have hundreds of languages, he says: language is bound with culture. “It’s the exact same problem with business,” he says. “Businesses do define things differently. With any specifications for B2B communication, you’re going to face this problem.”

Notably, UBL didn’t face a big technology vendor that worked against it, he adds, noting that some of the most influential vendors—including IBM, Sun, and Oracle—have gotten involved to varying degrees. “It’s not a victim of vendor sabotage. There are a lot of smart people working on this,” he says. “The theory is good—represent information in a standard way and exchange it.” But at present, he says, UBL simply isn’t solving the most pressing problems for business customers.

**UBL’s role in an SOA world**

What do UBL’s founders think its role will be in today’s world of Web services and SOA?

UBL is unique in being an integrated suite of horizontal, XML-based commercial-document standards developed in a transparent, open standards process and licensed royalty-free, Bosak says. This is one factor that’s winning over governments.

Bosak sees SOA and Web services as being more important to matters of message transport, process definition, identity, and trading-partner agreements. This leaves room for UBL to help in areas such as purchasing and invoicing processes.

As for the remaining challenges for the Oasis UBL group, Bosak says they want to help implementers make UBL work in action. A big part of this will be translating the UBL data dictionary into different languages, so that business analysts in various countries can understand the meaning of the tags in a UBL purchase order, invoice, or similar document, he says.

For UBL 1.0, UBL subcommittees translated the more than 600 definitions standardized by the specification into

---

**CURRENTS**

May/June 2007 IEEE SOFTWARE 115
Traditional Chinese, Simplified Chinese, Japanese, Korean, Spanish, and Italian, creating a UBL 1.0 International Data Dictionary that can be freely downloaded. UBL 2.0 will require nearly twice the work, but the committee hopes to have most of it done by late 2007.

Another to-do item: provide recommendations for UBL customization so that the standard can be restricted or extended to meet specific business needs in ways that don’t decrease its integrity, says McGrath.

All in all, has the vision for UBL changed from the original one regarding global commerce? Somewhat, says Bosak. “It’s become clear that the driving force behind its adoption will be the enormous savings realized from the automation of traditional paper-based business processes,” he says. “This doesn’t really have much to do with globalization, since the same kind of savings can be realized just as easily in trade across town as in trade across the world.”

McGrath remains active in forging open standards around ebXML and XML via OASIS and UN/CEFACT. However, the UBL Technical Committee experience has given him a different perspective on the challenges that such an effort faces. Even for the smartest technical minds, the political aspects of seeing through an open standard prove daunting.

“Developing standards is an exercise in consensus building,” McGrath says. “One of the lessons I have learned is that the technology is not the real challenge. UBL is not a technology breakthrough; it is a breakthrough in the application of a technology. So I would say the biggest challenge has been political. Someone once said, ‘There are two things in life you don’t want to see being made, laws and sausages.’ I think we can add standards to that list.”

In Brief

Multitouch Technology: Software Development at Your Fingertips?

Meg Shannon

Keep in touch” might be a trite saying, but for some software developers, the phrase could describe an integral component of future projects. That is, if multitouch technology takes off.

In multitouch technology, a touch screen reads multiple finger functions simultaneously. The aim is to provide more graphical and more intuitive manipulation of computer images, freeing users from the traditional confines of a mouse, stylus, or even keyboard. Because of this freedom to use fingers and hands alone or in combination and the ability of some input devices to accommodate multiple users, proponents believe multitouch technology will stimulate and support more collaborative functions in the computing realm. Such collaboration could be fruitful for software development.

“The primary impact [of multitouch technology] on software developers will be that they need to understand social interactions and team collaboration much better,” says Frank Maurer, head of the University of Calgary’s e-business engineering group. “The assumption that there is a ‘personal computer’ no longer holds. Computers will facilitate interaction and collaboration of groups touching and interacting with a single and, might I say, very large display.”

Old technology, new interest

Multitouch technology has been around since the mid 1980s. However, it has recently catapulted into the spotlight. Notably, Jefferson Han, a researcher at New York University’s Courant Institute of Mathematical Sciences, wowed a crowd at the Technology Entertainment Design show last year when he demonstrated multitouch technology. He seemingly transformed computer graphics into real-world physical objects as he pushed, pulled, and manipulated the images with finger and hand movements.

Likewise, Apple’s implementation of multitouch technology in its much-hailed iPhone has underscored the excitement around the technology. “The litmus test is how well the iPhone does,” says Andreas M. Antonopoulos, founder of and senior vice president at Nemertes Research, a research firm that investigates the impact of technologies on business. The iPhone is “the perfect environment, the perfect platform,” to introduce multitouch technology, he says.

Indeed, if consumers embrace the iPhone, manufacturers might be more willing to use multitouch technology in other platforms and build tools for creating multitouch applications, says Antonopolous. For the technology to take off and be exploited by more software developers, Antonopolous thinks it will have to become part of the operating systems.

“Once the operating system has [multitouch technology], then it’s easier for application developers to develop standard ways of using it,” says Antonopolous.

Touchy tabletops

While the phone is one of the early frontiers for multitouch technology, with PDAs, PCs, and tablet PCs to follow, re-
researchers already have bigger plans for the interface.

Take, for example, Chia Shen, senior research scientist at Mitsubishi Electric Research Laboratories in Cambridge, Massachusetts. As part of her research in human-computer interaction and interface design, Shen is employing DiamondTouch technology. DiamondTouch lets multiple users participate simultaneously using tabletop displays, with each user identifiable by his or her touch. Users can be either in the same room sitting around one table or at multiple tables in separate places.

“We are bringing in a new paradigm where, instead of clicks of the mouse, we allow people to simply reach out and touch to interact with what they see and what they need to do,” Shen says. Shen’s team has also developed DiamondSpin, a Java-based toolkit for creating tabletop interfaces such as DiamondTouch.

Such research underscores the work that could lead applications of multitouch display technology to software development applications. The DiamondTouch tabletop’s design, for example, could help bridge traditional software to new interfaces. Because the tabletop is a USB device that can connect to a PC or laptop, the software running on the computer will be usable on the tabletop, Shen says. However, she adds that current legacy MS Windows applications running on the tabletop would still require users to take turns interacting with the device. However, she claims that software vendors building a software package on their own could easily incorporate multiple touches as the input stream into their application.

Because multitouch technologies—tabletop displays particularly—offer a natural way for people to work together, collaborative functions are a natural for multitouch. Such interaction is especially handy for people working jointly on digital artifacts, says Brian Corrie, a graduate student and researcher in the Department of Computer Science at the University of Victoria, Canada. For software engineers, this could be source code, design documents, user interface mockups, and images, he adds.

“Software development seems to be becoming more of a global process, so supporting distributed teams of software engineers will likely become increasingly important,” says Corrie.

Frank Maurer’s research focuses on digital tables to support agile software development teams in iteration planning. The digital-table technology aims to solve a problem faced by distributed teams, he says. Lightweight agile methods rely on face-to-face communication to speed up and streamline software development. But this approach often has teams using index cards to organize and prioritize development tasks, Maurer says. This solution isn’t efficient for distributed teams. Enter digital-table technology, which can increase the effectiveness of collaboration among distributed teams while reducing the need for hard-copy documentation.

Just a gimmick?

To be sure, multitouch technology faces many challenges. Chen says that new interfaces enabling multiple users to truly simultaneously interact on a large display need nurturing. Other interface issues include how to ensure that each user has equal control and how to display toolbars to give each user access. Maurer says that a key goal is to extend current operating systems to provide a more useful framework for multiuser single-display tabletop applications. He also says that digital tables’ input resolution needs improvement.

Some reports suggest that multitouch might be forever relegated to a niche technique. And not everyone is sold on the idea of multitouch technology as a momentous force in computing. “You can find as many to say multitouch is brilliant as you can who say it’s a gimmick,” says Antonopolous.

Even so, software development is likely already feeling some of the influence of multitouch technology’s progress.

“We must think ‘out of the box,’ getting away from the one-mouse, one-keyboard mind-set,” says Shen. “We also need to understand what it means to have more than one hand and more than one person to interact with an application simultaneously, exactly at the same time.”

We publish IEEE Software as a service to our readers. With each issue, we strive to present timely articles and departments with information you can use.

How are we doing?

Send us your feedback, and help us tailor the magazine to you!

Write us at

Software
@computer.org