WINDOWS NT® IN THE ENTERPRISE

NTty Gritty

by Ross M. Greenberg

A New Universal E-Commerce Standard?

Recently, I started playing with things E-Commerce and discovered what E-Commerce pioneers learned long ago: E-Commerce partners usually do not speak the same data language as another.

Is help on the way? Maybe. Microsoft's Biztalk—and its soon to be released Biztalk Server—has the potential for making things easier, if Microsoft can garner enough support.

Biztalk is a platform-neutral framework for E-Commerce applications, allowing both customers and business/trading partners to conduct business over the Internet. In many ways, Biztalk feels like the merger of grown up versions of EDI (Electronic Data Interchange) plus OAG (Open Application Group) standards, which are then placed in a more standardized and useable form. In product form, it extends upon Site Server 3.0's Commerce Interchange Pipeline features, which are much like an Erector Set of commerce components and integrated tools. As such, if used, it sets a whole new standard in E-Commerce standardization. I can't be too impressed with the Biztalk framework, though: published standards are a dime a dozen.

Biztalk was just released in its Version .8 form recently and it's more difficult than I imagined (or it should be) to get a handle on what it is or what it's for. Just as I was getting comfortable with HTML and Dynamic HTML—as well as with their otherwise orphan step-sibling Cascading Style Sheets—some of their limitations have become bothersome obstacles. I had to wonder: what do these features buy me in a business sense? Not much, frankly, though that sounds like blasphemy to Web-page designers the world over.

It's true they deliver presentation abilities and control and can make Web-pages do things I never imagined they could before. (I kind of like a Web page blinking out of existence with a single click as it gets sucked into my Black Hole function, for example.)

But aside from making pages even slower on the Internet, making standard DHTML or HTML actually do something essential for a business mandates tweaking individual Web pages or .ASP pages. That's true even for mundane but important things such as browser validation of a form's entries before submitting it to the server for further processing. If the business model changes—perhaps adding or subtracting an option to Shipping and Handling charges, for example—each associated page must change. That translates directly into slower builds and deployment, especially on large distributed multi-server Web sites.

The picture gets quite bleak when one business needs to communicate with another. The Internet provides a method of enabling such communication—TCP/IP provides a vanilla common approach—but there is no common language for transmitting information between business entities. If a business cannot even transmit their standards of usage to a remote business—or receive and process that remote business' usage standards—it ends up cloistered in a backwater of E-Commerce, speaking only to itself and to applications entirely in its own control.

Enter the logical extension of HTML, called Extensible Markup Language, otherwise known as XML. XML provides documents with a near infinite variety of homegrown tags, some with imposed functionality, through Document Type Definitions (DTD). DTDs are a type of grammar that describes allowable contents for a field as well as their structure and relationship of one field to another.

For example, a date can be defined to contain months, days, and years, and years might be forced to be Y2K-compliant and be required to be four digits. These constraints are exemplified best in schema such as Biztalk's work-in-progress Framework. The main differences between a schema and DTDs lay in the constraints schema place on DTDs, as well as that DTDs are expressed in a non-XML language and grammar. That's just what developers do not need: yet another meta-language and grammar to learn. XML—which is, by definition, extensible—proves more than good enough. Surprisingly, DTD models do not have extensibility themselves: portions of the DTD model cannot be marked as "open." Adding new elements to an existing DTD programmatically and easily is only possible in a schema based model and that's an important distinction as new models—including Biztalk's Framework—are defined and deployed.

I'm finding XML surprisingly rich, deep and tough to learn. I'm no Web-design guru, but

WINDOWS NT® IN THE ENTERPRISE

HTML—especially with today's tools—is comparatively easy. XML seemed to require a substantial learning curve. And I'm not done yet.

Unless I'm totally missing the point, XML-enabled transmissions of data cannot function in a vacuum. They need a hefty supporting cast of code, including such things as .ASP running on IIS or even on Win95/8, compliant browsers (such as IE5 which can verify XML documents as it determines if they're "well-formed") or the still-being spec'ed Biztalk Server.

But once an XML Data Source is adequately defined through schema, DTD and Document Content Descriptions (which is basically a pre-Biztalk schema for XML documents), then all compliant data sources can be accessed easily. The rule of thumb seems to be that XML needs a lot of up-front design and implementation work, but once done right, it makes future connections near painless, as long as they are with compliant data sources, of course.

For real-life examples of XML in action, fire up a copy of Word 2000 and save a document as a Web page, then read the Web page back in and save as a normal document. Then do a compare document between the old and the new "normal" documents to see differences through the conversion process. Even with the double conversion, there are still no differences. Previous incarnations caused lots of tweaking all over the place to get the converted documents "normalized." Regardless of whether it was XML or something else, now it's a painless conversion and it saves a lot of time. I love it.

What about Biztalk Server, then? This is just software that understands how to apply server software—including homegrown services—to incoming documents that may or may not be in Biztalk schema format. The end result is that there is a single destination address that data from any site on the network can be sent to that should eventually output a Biztalk schema compliant form.

It's an old programmer trick: instead of turning one proprietary format into another, pick something rich but constrained enough so there's no real room for proprietary extensions. Then cut what appears to be proprietary recordsets.

If you move your hands fast enough, no one will notice the global stuff in the middle. That's what the Biztalk schema is to me. The stuff in the middle is the important stuff, though, because using it properly does most of the hard work. Biztalk Server provides several services, including Send and Receive of files (or business documents) using various protocols including HTTP, HTTP/S, SMTP and FTP. Upon receipt of a file, a Biztalk Server component can be called to port the document into a Biztalk Schema compliant form.

Most exciting to me, though, is using DCOM, Message Queue and Microsoft Transaction Server components and transport components, perhaps with a Site Server Commerce Pipeline in the mix, too. The potential is mind boggling, frankly, especially when you consider where E-Commerce is today and how much there is left to do.

The idea of using Biztalk Server to transform data from potentially legacy originated, heterogeneous systems into an MTS based DCOM distributed transactionally-guaranteed post is what I'm so excited about. In business-to-business environments, even designing something like the Biztalk schema specification can be a lot of work, and getting remote data to be compliant can cost more in time and effort than simple transactions may warrant. But I feel it's worth it if tomorrow's customer, somebody unknown today, can talk your talk, and walk your walk.

I have to wonder if Microsoft was so close to the specification trees that they couldn't really see the E-Commerce forest.

I have to wonder if Microsoft was so close to the specification trees that they couldn't really see the E-Commerce forest. Individually, I'm not too excited about the various Biztalk components. But when I stop to consider all the problems of E-Commerce with a variety of trading partners and how well the potential of Biztalk Server addresses them, it looks to be, well, not boring.

There's a lot more to E-Commerce than simply picking out cute shopping cart software, and I think the Biztalk Server software and the Biztalk Schema and specification go a long way to making a real E-Commerce universal language. Stranger things have happened.

Ross M. Greenberg, a software author of communications, security, and ActiveX controls, owns the firm Software Concepts located in the middle of the Catskill mountains. His opinions have appeared in a wide-range of technical and consumer publications.

This column will also appear on Network Computing Online at **http://www.networkcomputing.com/marketing/s_toc.html**

*(Any questions, thoughts, comments or suggestions for this column? Please E-mail **ntcolumn@cmp.com**)*