XML JOURNAL

The Ultimate XML Enterprise Resource

Volume:3 Issue:6

xml-journal.com

THE LARGEST WEB SERVICES CONFERENCE & EXPO IN THE WORLD!
web services EDGE
JDJ EDGE
XML EDGE
www.sys-con.com
Conference: June 24-27 Expo: June 25-27
Jacob Javits Convention Center, New York, New York
.NET
The Future of Enterprise Computing
FINAL PROGRAM
SAVE $100
Call Today!
JACOB JAVITS CONVENTION CENTER

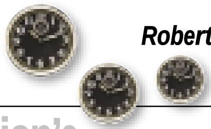
The gaps in Web services, and the steps to resolve them, need to be assessed

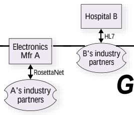
UBL and Web Services

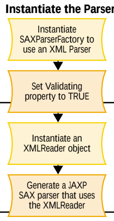
Written by Matthew Gertner 16

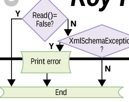
BRAND NEW!

SECTION-SPECIFIC COVERAGE






Instantiate the Parser
Instantiate SAXParserFactory to use an XML Parser
Set Validating property to TRUE
Instantiate an XMLReader object
Generate a JAXP SAX parser that uses the XMLReader



SYS-CON MEDIA

XML JOURNAL

FOUNDING EDITOR
Ajit Sagar ajit@sys-con.com

EDITORIAL ADVISORY BOARD
Graham Glass graham@themindelectric.com
Coco Jaenicke cjaenicke@mediaone.net
Sean McGrath sean.mcgrath@propylon.com
Simeon Simeonov simeons@macromedia.com

EDITORIAL
Coeditor-in-Chief
John Evdemon jevdemon@sys-con.com
Coeditor-in-Chief
JP Morgenthal jpm@sys-con.com
Editorial Director
Jeremy Geelan jeremy@sys-con.com
Executive Editor
M'lou Pinkham mpinkham@sys-con.com
Managing Editor
Cheryl Van Sise cheryl@sys-con.com
Editor
Nancy Valentine nancy@sys-con.com
Associate Editors
Jamie Matusow jamie@sys-con.com
Gail Schultz gail@sys-con.com
Jean Cassidy jean@sys-con.com
Editorial Intern
Stephanie Williams stephanie@sys-con.com

PRODUCTION
VP, Production & Design
Jim Morgan jim@sys-con.com
Lead Designer
Aarathi Venkataraman aarathi@sys-con.com
Art Director
Alex Botero alex@sys-con.com
Associate Art Directors
Cathryn Burak cathyb@sys-con.com
Louis F. Cuffari louis@sys-con.com
Richard Silverberg richards@sys-con.com
Assistant Art Director
Tami Beatty tami@sys-con.com

CONTRIBUTORS TO THIS ISSUE
John Evdemon, Tom Gaven, Jeremy Geelan,
Matthew Gertner, Roy Hoobler, Beverly Hrablook,
Kenneth J. Hughes, Craig King, Robert Lorenz, Greg Mable,
Joseph Moeller, JP Morgenthal, Karl B. Schwamb,
Simeon Simeonov, Robert Unger, John Walker

EDITORIAL OFFICES
SYS-CON MEDIA
135 CHESTNUT RIDGE ROAD, MONTVALE, NJ 07645
TELEPHONE: 201 802-3000 **FAX:** 201 782-9637
XML-JOURNAL (ISSN# 1534-9780)
is published monthly (12 times a year)
by SYS-CON Publications, Inc.
Periodicals postage pending
Montvale, NJ 07645 and additional mailing offices.
POSTMASTER: Send address changes to:
XML-JOURNAL, SYS-CON Publications, Inc.,
135 Chestnut Ridge Road, Montvale, NJ 07645.

XML-J '2.0': Enabling the Extensible Enterprise

WRITTEN BY **JOHN EVDEMON**

XML-Journal recently posted a special article online (at www.sys-con.com.xml) celebrating XML's fourth birthday. It's hard to believe the little toddler has grown up to become a preschooler. When XML was first introduced back in February 1998 it was designed to be a Web-based version of SGML. Indeed, the W3C XML Technical Recommendation states its goal as enabling a "generic SGML to be served, received, and processed on the Web in the way that is now possible with HTML." XML is a subset of SGML. Much the way Java preserved the best parts of C++ concepts and syntax, XML made SGML more digestible for the rest of the world. The hundreds of optional features (such as allowing user-defined delimiters instead of the angle brackets we've all come to know and love) have been discarded. The design of XML took a cue from the success of HTML – make it simple and people will use it. The original designers of XML could never have predicted its runaway success – the rapid adoption and deployment of XML within the IT community came as a bit of a surprise. (Many of these XML pioneers will be profiled in streaming interviews for our newly relaunched Web site – stay tuned for more details.)

Finally, analysts and developers alike could use an open, easy to use syntax for describing and exchanging their data – no more mucking about with CSV (comma-separated value) files or other proprietary implementations. This time we had a well-defined standard explaining how to properly implement the language. Surprisingly, most companies adhered to the standard.

So where do we go from here? Four years have gone by since XML became a Technical Recommendation. A lot has changed since 1998. The marketing community at times seemed more excited about XML than the people actually using it. We saw and read many big promises about XML allowing our data to be "seamlessly integrated" with virtually anyone. Analysts predicted that XML-based B2B exchanges would change the world, generating several million (some said billions) in revenue. XML was suddenly everywhere

(Continued on page 7)

WRITTEN BY **JP MORGENTHAL**

They say change is inevitable. That's hard to dispute, given the economic and technological events of the past five years. So it's inevitable that a publication focused on XML and its related technologies must also change to meet the needs of the expanding XML market.

When I look at XML, I'm always reminded of the beauty in the acronym *KISS* – "Keep it simple, stupid." XML has a simplicity and elegance that has enabled it to become the primary method of data representation and machine integration for the new millennium. It's so simple that you'd have to ponder for hours to truly fathom how integral XML has become to all industries (and our vocabulary) with virtually little effort (in comparison to other technologies). Grandma's even read about it in *Time* – go figure!

During the four-plus years since it was introduced, XML has undergone a significant transformation. If we analyze it with respect to a commonly understood model for the software industry – Geoffrey Moore's *Crossing the Chasm* – XML has moved into the space commonly identified as the *early majority*. This means that XML is no longer the bailiwick of ex-SGML experts and grassroots engineers, but has become the device of choice for captains of industry. Bill Gates and Steve Ballmer wield the acronym XML in front of their customers and Wall Street as they would a financial income statement, while IBM and Sun view it as newfound territory over which to battle for technical superiority.

XML-J has an expanded audience these days that includes some atypical members,

With this issue founding editor-in-chief **Ajit Sagar** *turns over the stewardship of* **XML-Journal** *to* **JP Morgenthal** *and* **John Evdemon**, *longtime members of the* **XML-J** *Editorial Advisory Board. For more on the changes at* **XML-J**, *see editorial director* **Jeremy Geelan**'s *piece on p. 62.*

(Continued on page 7)

SYS-CON MEDIA
135 Chestnut Ridge Rd., Montvale, NJ 07645
TELEPHONE: 201 802-3000 **FAX:** 201 782-9637

PUBLISHER, PRESIDENT, and CEO
Fuat A. Kircaali fuat@sys-con.com

BUSINESS DEVELOPMENT
VP, Business Development
Grisha Davida grisha@sys-con.com
Chief Operating Officer
Mark Harabedian mark@sys-con.com

ADVERTISING
Senior VP, Sales & Marketing
Carmen Gonzalez carmen@sys-con.com
VP, Sales & Marketing
Miles Silverman miles@sys-con.com
Advertising Director
Robyn Forma robyn@sys-con.com
Advertising Account Manager
Megan Ring megan@sys-con.com
Associate Sales Managers
Carrie Gebert carrieg@sys-con.com
Kristin Kuhnle kristin@sys-con.com
Alisa Catalano alisa@sys-con.com
Leah Hittman leah@sys-con.com

SYS-CON EVENTS
VP, Events
Cathy Walters cathyw@sys-con.com
Conference Manager
Michael Lynch mike@sys-con.com
Sales Executives, Exhibits
Michael Pesick michael@sys-con.com
Richard Anderson richarda@sys-con.com

CUSTOMER RELATIONS / JDJ STORE
Manager, Customer Relations / JDJ Store
Anthony D. Spitzer tony@sys-con.com
Customer Service Representative
Margie Downs margie@sys-con.com

WEB SERVICES
Webmaster
Robert Diamond robert@sys-con.com
Web Designers
Stephen Kilmurray stephen@sys-con.com
Christopher Croce chris@sys-con.com
Catalin Stancesco catalin@sys-con.com
Online Editor
Lin Goetz lin@sys-con.com

ACCOUNTING
Chief Financial Officer
Bruce Kanner bruce@sys-con.com
Assistant Controller
Judith Calnan judith@sys-con.com
Accounts Receivable
Jan Braidech jan@sys-con.com
Accounts Payable
Joan LaRose joan@sys-con.com
Accounting Clerk
Betty White betty@sys-con.com

SUBSCRIPTIONS
SUBSCRIBE@SYS-CON.COM
1 888 303-5282
For subscriptions and requests for bulk orders,
please send your letters to Subscription Department
Cover Price: $6.99/issue
Domestic: $77.99/yr (12 issues)
Canada/Mexico: $99.99/yr
all other countries $129.99/yr
(U.S. Banks or Money Orders)

XML-J '2.0': Enabling the Extensible Enterprise

JOHN EVDEMON *(Continued from page 5)*

– billboards, television, newspaper advertisements. Bill Gates proudly proclaimed that he goes to sleep at night "dreaming about XML schemas." Well, if Bill likes XML, it must be worth using, right? Suddenly everyone was wrapping their data in pointy brackets and waiting for the highly touted benefits and revenues to start rolling in.

And then reality set in.

Companies realized that simply adopting XML wasn't enough – XML can be used to solve some difficult problems, but it is, in actuality, nothing more than syntax. Syntax, however, isn't sexy, so the marketing machines revved into high gear, overpromising what could be done simply by switching your entire data infrastructure over to XML.

Four years later we've all grown tired of the hype – we're ready for some much-needed straight talk about what XML and XML-related technologies can and can't do for us. With this goal in mind, **SYS-CON Media** is relaunching *XML-Journal* with a special emphasis on real-world scenarios and implementations. *XML-J* will strive to clearly illustrate how XML is being used to solve issues in both large and small enterprises. Each issue will include actual case studies and thought-provoking analysis from leaders in the industry. We won't ignore our roots, however. Our **XML Labs** section is designed to provide leading-edge product reviews, tutorials, and the types of technical insights you've come to expect.

The past four years have zipped by in a confusing blur: several XML-oriented technical specifications have been released, ebXML continues to mature, and XML-based Web services have become extremely popular. Some of these initiatives will prove quite successful; others will end up on the scrap heap of IT trends past (CASE tools, anyone?). *XML-Journal*'s focus on the practical aspects of XML-based development and implementation can be a valuable tool to help you separate hype from reality.

And that's no hype.

JEVDEMON @ SYS-CON.COM

Author Bio

XML and e-business expert John Evdemon has served as CTO/director of XML-related products for both a large integration platform vendor and a small XML-centric startup. Currently an independent consultant, he has designed and deployed enterprise systems on a wide variety of platforms for well over 15 years, and has worked with XML since its beginning. An experienced speaker and teacher, John is an Invited Expert with the W3C XML Core Syntax Working Group and has chaired several industry-specific XML initiatives. He is a regular contributor to journals, books, and online forums for a wide variety of publishers, and is currently writing a new book on XML.

JP MORGENTHAL *(Continued from page 5)*

such as the chief executive officer, chief information officer, and others who constitute an unmatched source of corporate spending and decision-making. This is a great opportunity for those experienced in and enthusiastic about XML to incorporate this group into the XML fold. We must reach out and provide the knowledge they need to understand the implications of this technological advancement.

We at *XML-J* must leverage our years of publishing experience to make the technology understandable to business readers while making the business needs apparent and clear to the technologists. This will be a symbiotic and fruitful relationship.

To this end, *XML-Journal* is being retrofitted to bring about this harmonious union in printed form. We've created clearly marked sections that allow each reader to best utilize the information contained therein. Our new sections – **Data Management**, **Content Management**, **Enterprise Solutions**, **XML Labs**, and **Home** – identify clear and common uses for XML throughout the industry. In this, our inaugural issue, we've recast the mission statement for the retrofitted magazine as a whole and added a new one for each new section to better help you understand the type of content that will be presented in each area. Each type of article will also be clearly marked with newly created icons.

We hope this new format, direction, and audience bring significantly more value to you, the reader. We want the magazine to be a tool you can use in your daily work and share with your co-workers regardless of their role in the company. Moreover, we hope that a new breed of contributors will step forward and share with us their experiences and the results of using XML in their company, especially the experiences of getting others to see the light and drink the Kool-Aid along with the rest of us.

JPM @ SYS-CON.COM

Author Bio

JP Morgenthal is the managing partner of Comtellect, an IT consultancy in northern Virginia. He has served as CTO of two highly visible startup companies, developing software for supply-chain management and e-commerce. He is an internationally known expert on the design and implementation of enterprise applications and distributed computing technologies, JP is the author of two well-regarded books and numerous articles on XML, EAI, B2B, ERP, XRM, and SCM, JP has served as consultant to Sabre Group, Lockheed-Martin, JP Morgan, Citibank, ADP, Kelloggs, Sun, and Microsoft, among others.

MISSION STATEMENT

XML JOURNAL
The Ultimate XML Enterprise Resource

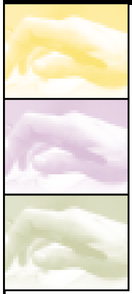
XML-Journal offers its readers an independent perspective into the Extensible Markup Language (XML) specifications, XML's use in business solutions, and the technologies used to create, manage, and use XML documents.

XML-Journal delivers current and valuable knowledge about the XML community, focusing on new developments, emerging standards, and related technologies. ***XML-Journal*** also focuses on delivering to business leaders critical direction and insight from top business leaders and implementers regarding the use and deployment of XML-based solutions. Additionally, ***XML-Journal*** focuses on how XML technologies impact the business community, which includes analyses, return-on-investment, total-cost-of-ownership, and case studies.

WHO WE ARE ***XML-Journal*** is a publication for the CEO, CIO, CTO, CFO, vice president, director, programmer, engineer, and line-of-business manager – all who understand that XML plays a role in how their systems interact with other systems and with the outside world. To this end ***XML-Journal*** covers the impact that XML will have on their companies, the successes and failures of those that have been implementing XML technologies, the skills required to use XML in their organization, and which standards they should watch and, perhaps, participate in developing.

WHAT WE COVER ***XML-Journal*** is broken down into five categories. The **Home** section is devoted to editorials, news, and general commentary not associated with any of the other sections. **Content Management** covers the systems, applications, and best practices pertaining to the organization, dissemination, and presentation of information. **Data Management** covers the systems, applications, and best practices pertaining to the storage, transformation, representation, and general use of structured and unstructured data. **Enterprise Solutions** covers case studies, best practices, applications, and systems that use XML in the delivery of mission-critical enterprise applications, such as enterprise application integration, B2B commerce, supply-chain management, and customer resource management. And **XML Labs** covers implementations of XML applications, the latest on XML standards, reviews of XML products and books, and XML tutorials.

Call for Papers



If you have something of value to share with the IT world, want to make a difference in the XML community, or seek recognition by your industry peers:

We'd like to hear from you!

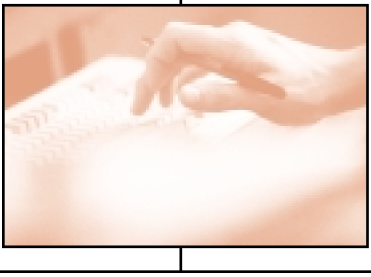
XML-Journal is seeking writers with real-world XML experience to provide articles for the following sections:

- ***Content Management:*** Organization, dissemination, and presentation of information
- ***Data Management:*** Storage, transformation, representation, and general use of structured and unstructured data
- ***Enterprise Solutions:*** Systems and applications that manage mission-critical functions in the enterprise
- ***XML Labs:*** product reviews, book reviews, tutorials, and standards analyses

XML-Journal is the premier publication for CIOs, CTOs, IT managers, enterprise solutions consultants, systems analysts, software engineers, and general XML enthusiasts interested in the applications, solutions, and technologies surrounding XML. The magazine focuses on how XML technologies impact the business community, which includes analyses, return-on-investment, total-cost-of-ownership, and case studies.

XML-Journal offers an independent perspective into the XML specifications, its use in business solutions, and the technologies used to create, manage, and use XML documents. It delivers current and valuable knowledge about the XML community, focusing on new developments, emerging standards, and related technologies. It delivers to business leaders critical direction and insight from top business leaders and implementers regarding the use and deployment of XML-based solutions.

Prospective writers should submit a proposal (75–200 words) that outlines the topic, audience, and section the piece is directed toward to http://editorial.sys-con.com/proposal.cfm. Upon acceptance, an *XML-J* staff member will contact you with your article's due date, a writer's agreement, the writer's guidelines, and a schedule of milestones.



JP Morgenthal / John Evdemon
Coeditors-in-Chief, ***XML-Journal***

WRITTEN BY **ROBERT UNGER**

Is It Time for XML in the ACH?

While currently premature, potential changes in market conditions could tip the balance

The Automated Clearing House (ACH) Network has the flexibility to accommodate XML-formatted payment remittance records in the same manner that it currently transmits EDI-formatted records, according to NACHA – The Electronic Payments Association.

NACHA develops operating rules and business practices for the ACH network. ACH payments include direct deposit of payroll, Social Security benefits and tax refunds, direct payment of consumer bills (preauthorized checking account deductions), B2B payments, federal tax payments, and, increasingly, e-check and e-commerce payments. In 2001 there were almost 8 billion ACH payments made worth $22.2 trillion.

B2B Payments and Remittance in ACH Network

More than 20,000 financial institutions are connected to the ACH network. As B2B e-commerce continues to grow, corporations are increasingly using the ACH for payment transactions. In 2001 corporate electronic payments over the ACH exceeded 1.4 billion, an increase of 11.9% from 2000. The dollar amount of these transactions was more than $16.3 trillion (figures include B2B, business-to-government, and intracash concentration).

In addition to the actual payment, businesses frequently require remittance information in order to reconcile payments within accounts receivable systems; remittance data provides additional information about the payment that is often necessary for reconciling B2B transactions (e.g., why the invoice is not being paid in full).

The ACH permits remittance data to be transmitted with an ACH payment instruction; the remittance data is placed in an "addenda" record, thereby providing additional information about payment as necessary. The NACHA Operating Rules require that the remittance data in the addenda record be structured in Electronic Data Interchange format according to American National Standards Institute (ANSI) American Standards Committee (ASC) X12 Interchange Control Structures (governing body for EDI), or in NACHA-endorsed banking formats. In 2001 there were 143 million such financial EDI payments using the ACH network with 413 million EDI records. Financial EDI payments on the ACH network have grown by 216% since 1997; use of EDI records has grown by 279% during that period.

NACHA's Internet Council, formed in 1996 to facilitate global e-commerce, recently completed a study to assess the potential use of XML-formatted remittance data in the ACH for B2B payment transactions. The white paper on which this article is based – "XML Formatted Remittance Data in the ACH: A Feasibility Assessment" (available at http://internetcouncil.nacha.org) – reviews:

- The current state of ACH B2B payments and remittance processing
- The proposed use of XML in the ACH
- The perceived future of Internet-based B2B payments and remittance

The ACH's capability to accommodate both payment and remittance information has given financial institutions the opportunity to provide value-added services to their corporate clients. Financial institutions, as a potential revenue source, offer a number of EDI-related products and services to support corporate payment and remittance processing. Such services may include data translation, payment origination/receipt, electronic lockbox, translation software, programming, operations, help line, security, value added network (VAN) relationship management, and so on.

Many financial institutions track their B2B/EDI services as a separate profit center. Others may include the profit/loss statement from these services with another bank product, such as ACH services. While it's difficult to obtain data illustrating the "health" of a standalone B2B/EDI business within the financial services community, anecdotal evidence suggests that these services are a high-cost delivery channel to financial institutions and some wouldn't consider them profitable. Generating more volume is potentially a way to increase profitability, but reaching new markets has been problematic.

Despite the growth of ACH/EDI corporate payments, the overall penetration of B2B e-payments remains low when compared to payments made by check. Gartner estimates that only 14% of all corporate payments are made electronically. While ACH accounts for nearly 87% of these electronic payments, the high implementation costs of using EDI in addenda records limits its

AUTHOR BIO

Robert Unger is director of electronic billing and payment for NACHA – The Electronic Payments Association. Previously, Robert directed the Student Aid Modernization Partnership Forum, a public/private initiative led by NACHA focused on developing e-commerce interoperability standards to improve delivery of student financial aid and related services. He holds an MA in technology education.

use to larger corporations that regularly invoice and pay one another, thus excluding most of the small/medium enterprises (SMEs).

As a way to bridge the gap to the lower-volume trading partners, many corporations (through their vendors) now offer Internet-based EDI services that were previously available only through the use of special software and a VAN connection. For example, Web EDI converts online forms data into EDI formats (suited for low-volume activity), while Internet EDI supports file transfer over the Internet, bypassing the need for a VAN (better for high-volume activity). These EDI developments may help lower some of the B2B e-commerce barriers SMEs face when conducting business with larger entities, particularly related to transaction fees, transmission fees, service costs, network availability, IT infrastructure, and data mapping/integration. However, it remains to be seen if the EDI “bridging” opportunities will increase electronic payment transactions among SMEs.

Summing up...

Strengths:

- Corporations prefer “data and dollars” together, and the ACH has a well-established process for facilitating ACH B2B payment and remittance processing.
- The volume of EDI payment/remittance data through the ACH has been increasing considerably in recent years.
- Financial institutions have a range of revenue opportunities available by providing a number of support services in this area.

Weaknesses:

- Using EDI in this process increases costs for some users as well as for the service providers.
- The high costs to financial institutions for providing ACH B2B remittance processing and EDI-based services means that this isn’t a profitable venture for some organizations.
- Overall, the penetration of ACH B2B payments remains low compared to check payments, and usage is primarily confined to larger corporations with established trading partners.

Opportunity:

- Internet and Web-based EDI may be an opportunity to incorporate SMEs into e-commerce programs, and may expand the potential market for financial institution services.

> **“XML is a topic of great interest that the Internet Council will continue to monitor closely. The Council will recommend rule and format changes to NACHA as warranted by developments in the marketplace.”**
>
> David L. Merritt
>
> *Chairman, Internet Council, and VP/product design manager, New Business Development Group, Mellon Global Cash Management*

Proposed Use of XML in the ACH

As noted previously, ACH rules currently permit the transmission of remittance data with a payment file, provided that the remittance data is in an approved EDI X12 format. The essential proposition in the ACH–XML remittance model, as described in the Internet Council’s white paper, is whether XML-formatted remittance data may be used in addition to EDI-formatted remittance data. The difference in the XML model is that the remittance data in the addenda record would be in an XML rather than an EDI format. All of the remaining ACH processes would remain the same.

The tremendous amount of XML activity directed toward B2B supply-chain transactions led the Council to study the potential usage of XML-formatted remittance data in the ACH. A key value proposition to examine is whether this option would increase usage – and revenue opportunities – for ACH stakeholders.

Based on a similar process model, the financial institutions’ revenue model for supporting XML remittance services would also be similar to the EDI remittance services model. That is, financial institutions could still offer the same range of services cited earlier, and the services could be based on XML or EDI.

However, adding XML to the mix may increase the servicing costs for a corporate product that has historically been marginally profitable at best. Supporting XML remittance processing will require a significant investment in startup costs to upgrade corporate payment and remittance services, including software/hardware changes and additional staff with new skills (or training of current staff). This can be profitable only if the service can drive increased volume, and perhaps the more compelling allure of XML for financial institutions is the potential to expand the e-business community and reach new customers.

Similar to the targeted users of Internet EDI services noted earlier, the obvious potential market for XML-based payment and remittance services is SMEs ($1 million to $500 million in annual revenue), which historically have been unable to engage in e-commerce due to the high entry costs associated with EDI and data integration. According to one estimate, only 15% of SMEs have e-commerce capabilities at the moment. While the traditional ACH community is migrating toward Internet and Web EDI as a bridge to SMEs, others see XML as the key enabler for this market.

The assumption here is that a growing installed base of XML-enabled SMEs will translate into a greater demand for electronic payment and remittance services from financial institutions. However, while SMEs transition toward e-commerce for front-end applications (e.g., order placement, negotiation, status check), there remains the more difficult task of integrating data on the back end (e.g., accounts payable and accounts receivable), which affects the payment networks. In fact, according to a survey conducted by the Association of Financial Professionals, “The major barrier to increased use of electronic payments continues to be the lack of integration between an organization’s electronic payment and accounting systems.” Simply allowing SMEs to integrate with larger companies via a Web browser does little to solve the SME data integration issue, and doesn’t necessarily promote more demand for electronic payment and remittance services.

Even if XML availability were to stimulate greater demand for electronic payments, the lack of standards further complicates potential use of XML in payments and remittance processing. In contrast to EDI, there may be as many as 500 XML specifications. A number of these are payment-related DTDs that are in production under the auspices of standards bodies or industry consortia (e.g., Open Financial Exchange [OFX], Interactive Financial Exchange [IFX], RosettaNet, XMLPay, xCBL, ebXML…the list goes on). The lack of a recognized industry-wide standard inhibits the integration of payment systems into the XML supply chain, meaning that pay-

Missed an issue?

We've got 'em all for you on CD!

JAVA DEVELOPER'S JOURNAL

The most complete library of exclusive *JDJ* articles on one CD!

Check out over 500 articles covering topics such as...
Java Fundamentals, Advanced Java, Object Orientation, Java Applets, AWT, Swing, Threads, JavaBeans, Java & Databases, Security, Client/Server, Java Servlets, Server Side, Enterprise Java, Java Native Interface, CORBA, Libraries, Embedded Java, XML, Wireless, IDEs, and much more!

JDJ The Complete Works
~~Reg $119.99~~
Buy Online
Only $71.99

XML JOURNAL

The most complete library of exclusive *XML-J* articles on one CD!

Check out over 150 articles covering topics such as...
XML in Transit, XML B2B, Java & XML, The XML Files, XML & WML, Voice XML, SYS-CON Radio, XML & XSLT, XML & XSL, XML & XHTML, 2B or Not 2B, XML Industry Insider, XML Script, <e-BizML>, XML & Business, XML Demystified, XML & E-Commerce, XML Middleware, and much more!

XML-J The Complete Works
~~Reg $59.99~~
Buy Online
Only $53.99

COLDFUSION Developer's Journal

The most complete library of exclusive *CFDJ* articles!

Check out over 250 articles covering topics such as...
Custom Tags, ColdFusion and Java, Finding a Web Host, Conference Reports, Server Stability, Site Performance, SYS-CON Radio, ColdFusion Tips and Techniques, Using XML and XSLT with ColdFusion, Fusebox, Building E-Business Apps, Application Frameworks, Error Handling, and more!

CFDJ The Complete Works
~~Reg $79.99~~
Buy Online
Only $71.99

SPECIAL OFFER:
Buy CFDJ & CFA The Complete Works
For Only $129.99

CF Advisor

The most complete library of exclusive *CFA* articles!

Check out over 200 articles covering topics such as...
E-Commerce, Interviews, Custom Tags, Fusebox, Editorials, Databases, News, CF & Java, CFBasics, Reviews, Scalability, Enterprise CF, CF Applications, CF Tips & Techniques, Programming Techniques, Forms, Object-Oriented CF, WDDX, Upgrading CF, Programming Tips, Wireless, Verity, Source Code, and more!

CFA The Complete Works
~~Reg $79.99~~
Buy Online
Only $71.99

JDJStore.com

Order Online and Save 10% or More!

www.JDJSTORE.com

OFFER SUBJECT TO CHANGE WITHOUT NOTICE

COLLECT ALL 4
FOR ONLY $229.99
JDJ, XML-J
CFDJ & CFA

> "Various scenarios will have different implications for the applicability of XML-formatted remittance data"

ment will continue to be performed outside the supply chain system.

Summing up...

Strengths:

- The XML remittance proposition corresponds with the EDI model preferred by corporate users, and leverages the current ACH infrastructure for an additional use.
- Financial institutions would be able to offer a range of payment/remittance services as a way to generate revenue.

Weaknesses:

- There is no national standard for using XML in payments and remittance.
- Most businesses (particularly SMEs) don't have integrated systems and aren't ready for "end-to-end" processing.
- The large size of XML files may impact data storage needs and file transmission throughput.
- Costs for everyone will likely increase during a lengthy "migration" period during which both XML and EDI need to be supported.
- Current XML usage in the supply chain is negligible, and demand from end users remains low.

Opportunities:

- XML implementations for payment/remittance may cost less than comparable EDI installations, and software vendors are developing a range of solutions (some free).
- Major corporations have announced their intention to migrate toward XML in supply chain–related activities. If the whole supply chain moves toward XML, it probably makes sense for the payment/remittance function to migrate as well.
- SMEs may become more involved in e-commerce, with XML providing a more flexible interface, thus increasing the potential market for payment/remittance services.

Threats:

- Will the XML remittance option be a sufficient reason to convert the large number of check writers to e-payments?
- Does the proposition make it easier for XML-enabled "intelligent" hubs – or vendor-hosted solutions – to intervene in the current ACH processing and revenue models?

Future of Internet-Based B2B Payments and Remittance

What are the requirements for the future of Internet-based B2B payments and remittance? What are the emerging models for meeting the expectations? What are the prospects for legacy systems like the ACH? These are some of the key questions to evaluate, and the various scenarios described will have different implications for the applicability of XML-formatted remittance data in the ACH.

The anticipated growth in B2B e-commerce indicates these issues will significantly impact how the payments and related-services pie is sliced among current and emerging participants. No one is predicting a decline in B2B e-commerce. At the moment, the ACH is by far the dominant method for facilitating electronic B2B payment and remittance exchanges, but Internet commerce may challenge this status.

It's difficult to divine how the future of B2B payments and remittance services will evolve. However, it's possible to identify some of the key issues that will help determine the future direction of this industry, including, for example, user requirements and network (i.e., private versus public) participation. The question remains whether having XML capability would impact any of these scenarios; a more in-depth discussion is available in the white paper.

Conclusion

Financial institutions, which are responsible for the governance of the ACH, will need a clear understanding of the return-on-investment before promoting a change in the current B2B payment process to support XML remittance data. Coupled with negligible end-user (i.e., corporate ACH users) demand and no clear direction on the future of B2B electronic payments, there are no stakeholders propelling this issue to the forefront. Meanwhile, usage of the current ACH/EDI processes continues to grow.

Payment systems in themselves aren't necessarily e-commerce drivers, and adding the XML remittance data option doesn't present a clear business case at the moment for even early market leaders. Yet significant interest remains in the potential for using XML in the ACH, and there is a desire to track related payment developments. The Internet Council has identified some triggers that may lead the membership to propose that the question of XML remittance data be reevaluated. Triggers are marketplace events that would change the environment, or perception, for answering the XML-in-the-ACH question. Examples include, but are not limited to, the following:

- ***Competitive concerns*** (e.g., a competing bulk-payment network commits to XML remittance data, and the network users adopt this new practice)
- ***Customer/constituent requests*** (e.g., large-volume financial institutions, ACH operators, or other dominant payment industry companies request this capability)
- ***Government/regulatory demands*** (e.g., a new law or regulation requires the use of XML-formatted remittance data)
- ***Legal/risk assessments*** (e.g., if stakeholders are somehow at risk – or have some liability – for not providing this capability)
- ***Related industry developments*** (e.g., the banking community adopts and begins to use a particular XML payment standard).

These examples aren't exhaustive, nor are they weighted in any particular order of importance. The purpose is to provide some general guidance for when a reevaluation of the XML-formatted data question may occur.

RUNGER@NACHA.ORG

> "Significant interest remains in the potential for using XML, and there is a desire to track related payment developments"

THE INSIDER INTELLIGENCE YOU NEED...

JAVA: INDUSTRY NEWSLETTER
Reply
Reply All
Forward
Flag
Print
Email Newsletters
JAVA DEVELOPERS JOURNAL Industry > Newsletter

Monday, December 10, 2001

BROUGHT TO YOU BY...

aligo Wireless Development Strategies - A Free White Paper. How can you use J2EE/J2ME to successfully deploy Wireless Applications? Aligo, the leader in Java based wireless application servers, offers a free technical paper on Java and Wireless Development. Click Here! for more information.

Newsletter

JDJ News
JDJ List
Subscribe Now!
Special Offer!
Privacy Notice

JAVA DUDES

WAKESOFT Learn more now!

JDJ NEWS DESK EXCLUSIVE

Oracle Targets Top Spot With New Oracle9iAS and JDeveloper

by Rick Ross - Founder of Javalobby.org

(December 10, 2001) - I returned just yesterday from a fantastic trip to San Francisco for Oracle's OpenWorld conference. It wasn't clear what to expect when I accepted Oracle's invitation to give a presentation, but I was delighted to find the show had a high-energy crowd that was very tuned in to Java, especially J2EE and Web services. The whole show was great, but the most interesting and important items came from Oracle, itself - Oracle9i Application Server Release 2 and Oracle9i JDeveloper.

Introducing AltoWeb Release 2.5

WEB SERVICES: INDUSTRY NEWSLETTER
Reply
Reply All
Forward
Flag
Print
Email Newsletters
WebServices JOURNAL Industry >Newsletter

Monday, December 3, 2001

Stay Tuned for WS-J's December Issue!
Full Editorial Content Coming December 5th

BROUGHT TO YOU BY...

ORACLE Catch the Oracle9i JDeveloper wave and see how built-in profilers and Code Coach makes your Java code tighter and faster than ever before. Based on the latest J2EE and XML standards, JDeveloper combines Java coding, UML modeling, and RAD on NT/Linux/Unix. Click Here! for more information.

Newsletter

WSJ News
Next Issue
Subscribe Now!
Privacy Notice

ONLINE EXCLUSIVE: FULL "KEYNOTE PANEL" TRANSCRIPT

spiritsoft complete integration

Sean Rhody: Probably the question that is on almost everybody's mind is, what are Web services? I know I have a whole magazine devoted to answering that question, but I think one of the most interesting parts of the magazine is that the answers are different for different people and from different people. So to start things off I'll go down and ask each of the panelists that question. In your mind, Barry, what are Web services?

Barry Morris: I would think we all tend

Tuesday, December 18, 2001

BROUGHT TO YOU BY...

flamenco Secure Web services. Reliable Web services. Manageable Web services. If you need Web services then you need a Web services network. Security, guaranteed delivery, and management services with no code changes. Plug into the Flamenco Network and get going! For a limited time, bring your Web services and test-drive the Flamenco Network for free.

Newsletter

XML News
Subscribe Now!
Special Offer!
Privacy Notice

XML NEWS

Xerces-C++ XML Parser Version 1.6.0 Provides Full Support for W3C XML Schema Recommendation

(Toronto) - The XML Parsers Development division of the IBM Toronto Laboratory has announced the much-anticipated release of the Apache Xerces-C++ XML parser version 1.6.0, including full support for the W3C XML Schema Recommendation version 1.0. Xerces-C++ is a validating XML parser written in a portable subset of C++; Xerces-C++ makes it easy to give applications the ability to read and write XML data. A shared library is provided for parsing, generating, manipulating, and validating XML documents. features: conformance to the XML Specification 1.0; tracking of [Level 1.0], DOM [Level 2.0], SAX/SAX2, namespace, and XML source code, samples, and documentation; and validation of XML; pluggable catalogs, validators performance; and customizable error handling. The latest C++ includes a port to FreeBSD and allows users to through a method call. Source code and binaries Linux, Solaris, and Windows.

performance, modularity, and scalability. Source are provided with the parser. For to make minimal use of templates, no RTTI, use of #ifdefs.

To Keep Ahead of The Curve

FREE E-Newsletters

SIGN UP TODAY!

Go to www.SYS-CON.com

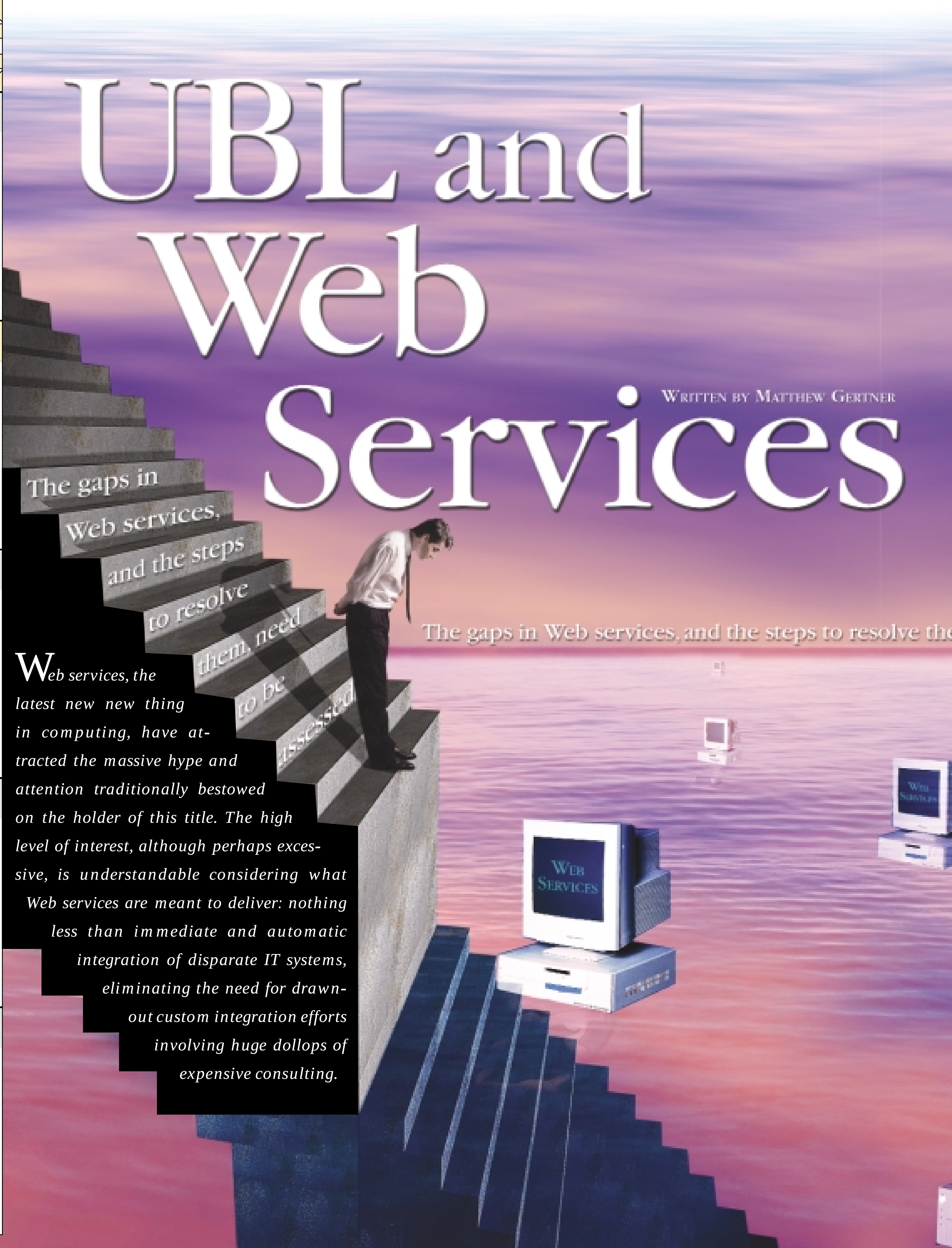
The most innovative products, new releases, interviews, industry developments, and plenty of solid *i*-technology news can be found in SYS-CON Media's Industry Newsletters. Targeted to meet your professional needs, each e-mail is informative, insightful, and to the point. They're free, and your subscription is just a mouse-click away at www.sys-con.com.

SELECT THE INDUSTRY NEWSLETTERS THAT MATCH YOUR NEEDS!
CHOOSE ONE – OR TRY THEM ALL!

JAVA DEVELOPERS JOURNAL Industry > Newsletter	WebLogic Industry > Newsletter
WebServices Industry >Newsletter	WebSphere Industry > Newsletter
XML JOURNAL Industry > Newsletter	COLDFUSION Developer's Journal Industry > Newsletter
wireless Industry > Newsletter	LINUX WEEK Industry > Newsletter

Don't Delay!
Subscribe for FREE!
at www.sys-con.com

Exclusively from the World's Leading *i*-Technology Publisher SYS-CON MEDIA



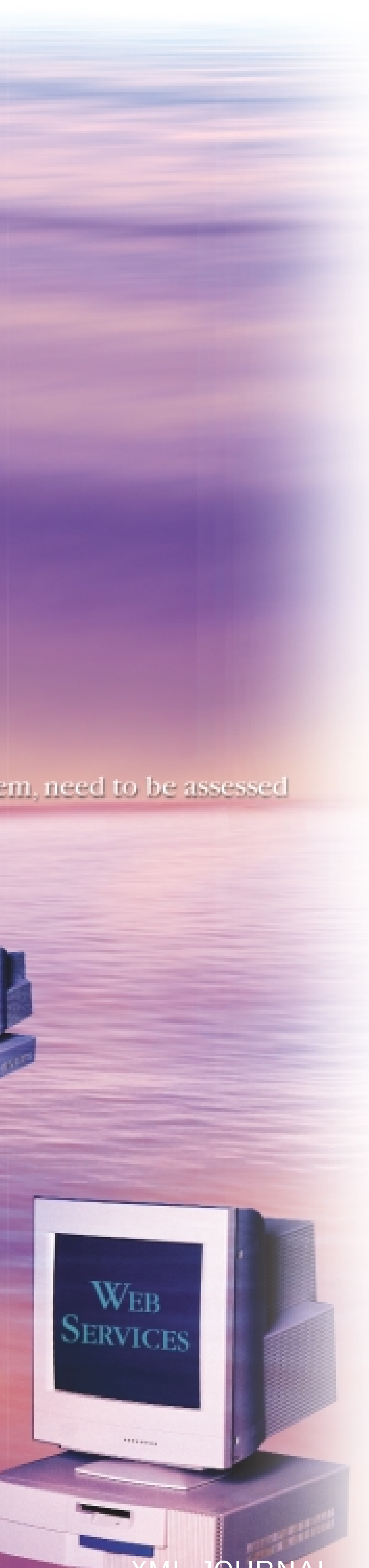
UBL and Web Services

Written by Matthew Gertner

The gaps in Web services, and the steps to resolve them, need to be assessed

The gaps in Web services, and the steps to resolve the

Web services, the latest new new thing in computing, have attracted the massive hype and attention traditionally bestowed on the holder of this title. The high level of interest, although perhaps excessive, is understandable considering what Web services are meant to deliver: nothing less than immediate and automatic integration of disparate IT systems, eliminating the need for drawn-out custom integration efforts involving huge dollops of expensive consulting.



The flood of coverage that Web services has attracted might not be indicative of much, other than the fact that IT journalists haven't had much else to write about lately. Certainly some have started (yet again) to catch on to the fact that if something seems too good to be true, it probably is. Like the graphical user interface, the Internet, Java, and many other technologies, Web services may have a profound, even revolutionary effect on computing. But like these other technologies, Web services will require much work over years, not months, for this potential to be realized.

Among the most significant questions still facing Web services are:

- Who is responsible for turning preliminary Web service specifications into real standards?
- How are Web services chained together to form complete applications?
- How are the semantics of Web service interfaces codified in a way that lets an application access an unknown Web service?

The answers to these questions are slowly becoming clearer. The recently formed Web Services Interoperability Organization (WS-I), which brings together Microsoft, IBM, Oracle, SAP, Intel, HP, and other high-tech leaders, is the most plausible answer yet to the question of who will standardize the hodgepodge of existing Web service–related specs. Recently the World Wide Web Consortium also reaffirmed its commitment to aggressive pursuit of work on Web services standards. And early business process languages like IBM's Web Services Flow Language and BPMI.org's Business Process Modeling Language are increasingly addressing the problem of assembling Web services into applications.

The Semantic Gap

In this article we examine the final question, that of standardizing Web service semantics.

The challenge of representing all but the most trivial Web service semantics in a machine-readable way was well exposed last fall in an article by Clay Shirky. Existing Web service standards, specifically SOAP, WSDL, and UDDI, provide little more than a way for applications to invoke a Web service once they already know what its interface looks like. This is all very well for Web services whose purpose is sufficiently transparent (Shirky uses the example of a Web service that translates centimeters into inches), but it's exactly these services that are the least interesting. After all, we're talking about revolutionizing computing.

Far more significant are Web services that perform an important business function and can be linked automatically to a broader process. Imagine, for example, Web services that deliver catalog information as part of a procurement process, synchronize delivery schedules as part of a materials management solution, or offer financial services for credit, factoring, or payment. This is where the rubber of Web services hits the e-business road: when I can develop my whole business application without worrying about the issue of payment, and then plug in any one of a number of competing payment services at the last minute, or even let the end user decide at runtime which one he or she wants to use.

So far so good, but this is actually a massive effort. It's not impossible to imagine someone coming up with standard definitions of catalogs, orders, advance shipment notices, bills of lading, invoices, and payment requests. But any survivor from the various Electronic Data Interchange (EDI) standards efforts will tell you that it's a daunting technical task to define data formats that come even close to meeting the requirements of a broad range of diverse enterprises, and perhaps an even more daunting political task to get companies to adopt them.

Building a Better Business Vocabulary

Nevertheless, a standard vocabulary for e-business is an essential piece of the Web services puzzle. To the benefit of anyone brave enough to undertake this task, there is a large body of existing work to draw upon. This ranges from business vocabularies predating XML, such as ANSI X.12 and UN/EDIFACT, through horizontal business libraries like Commerce One's xCBL and Ariba's cXML, to vertical business libraries for specific industries, such as RosettaNet for high tech and CIDX for chemicals. Finally, some pioneering work on creating business libraries that can be adapted automatically to different business requirements (or "contexts") has been performed as part of ebXML, a joint effort between OASIS and UN/CEFACT.

How then to turn this alphabet soup into a single, coherent, and widely accepted business vocabulary? By far the most comprehensive effort was launched last year under the name Universal Business Language. Structured first as an independent group, UBL was formally accepted as an OASIS Technical Committee last October. UBL brings to the table a number of strengths, including experienced and proven leadership, broad industry and vendor support, and a solid technical foundation.

UBL takes as its starting point xCBL, widely accepted as one of the most comprehensive XML-based business libraries. UBL's Library Content subcommittee has been entrusted with the task of harmonizing xCBL with the fruits of EDI's Joint Core Components initative and with other business libraries, including vocabularies for industry verticals. Official liaisons have been appointed to UBL from several vertical standards organizations to ensure that the basic UBL business documents will work across multiple industries.

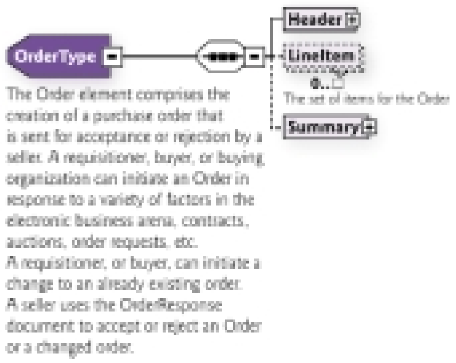
At the same time, UBL's technical subcommittees are specifying the nuts and bolts that will underpin the document library. These include tricky but important decisions about which schema features to use and how to name tags in a clear, concise, and consistent manner. The ebXML context extension methodology is also being adopted and improved in order to produce an automated procedure for creating extended schemas (e.g., for a specific industry, region, or company) that interoperate with the base schemas in the document library.

A first version of the UBL document library is scheduled to be completed 12 months into the effort (a draft of the first schema, for Purchase Order, has recently been released). The context extension methodology will be released approximately one year later – sooner if it turns out that this work can be performed in parallel with the creation of the document library.

Mission Impossible

Some might claim that the task facing UBL is not so much difficult as it is impossible. The real-world requirements for a business vocabulary are such that it isn't feasible to create a single document library that meets the need of every trading party. This is because different companies use different document formats depending on their industry, region, internal conventions, and a range of other factors. Since doing business with companies outside your industry and region is more the rule than the exception, this calls into question the whole premise of a business library that is truly universal.

Various approaches have been taken in previous efforts to address this issue. EDI formats tend to be highly customized for specific pairs of trading partners. Unfortunately, this manual customization is prohibitively expensive for all but the largest enterprises, explaining why EDI adoption has been restricted mainly to big companies with big IT budgets. As for xCBL, it took a slightly different tack, defining schemas that are maximalistic, with a host of optional fields augmenting the core set of required fields. By using object-oriented features, new schemas can be derived that redefine the appropriate optional elements as required and add any new elements that weren't originally anticipated.



The Order element comprises the creation of a purchase order that is sent for acceptance or rejection by a seller. A requisitioner, buyer, or buying organization can initiate an Order in response to a variety of factors in the electronic business arena, contracts, auctions, order requests, etc.
A requisitioner, or buyer, can initiate a change to an already existing order.
A seller uses the OrderResponse document to accept or reject an Order or a changed order.

FIGURE 1 | Preliminary draft schema for Purchase Order

The xCBL approach has a lot of advantages. On the one hand, it maximizes interoperability by using so many optional fields. If you and I require a warehouse party in our purchase orders, chances are we'll both use the one defined in xCBL, so my IT systems will understand the data I receive from you without having to be modified. At the same time, the use of an object-oriented extension means that I can create an extended schema with information no one could have anticipated without breaking the ability of processing engines to understand the standard information in the document. (That's the principle of polymorphism, for you gearheads out there.)

But at the end of the day, xCBL and similar business libraries still bear the genes of their EDI ancestors. Whizzy XML features make it a lot easier (and cheaper) to customize newer libraries like xCBL than EDI, but they still have to be customized through a painstaking manual process. This prevents the creation of cheap, standardized implementations for small businesses and, as with EDI, rules out the possibility of dynamic e-commerce. The latter point is particularly important since much of the Web service hype is premised on the idea that new services can be discovered and invoked without constantly calling in the IT department.

UBL is building on the pioneering work undertaken as part of ebXML to create a "context extension methodology" that accepts the need for customization but turns it into an automated process. This is done by defining rules that can be applied to a base schema to create a new schema automatically. For example, a rule might specify that a required "State" field be added to addresses in the United States.

It's likely that in the short term the context extension methodology will be used mainly at design time to lower the cost and technical expertise needed to create customized schemas. By applying both my context rules and those of my trading partner, I can instantly generate document formats that fill both of our requirements. In the longer term context rules will become a standard part of the trading agreements exchanged at the beginning of trading interactions, and mutually acceptable document formats will be generated in real time. Runtime binding of freshly discovered Web services will then become a real possibility.

Core Library Category	Base-level and aggregate core components needed by the other categories
Trade/Procurement Category	Purchase Order/Purchase Order Response/Purchase Order Change
Materials Management Category	Dispatch Advice (international) or Advance Ship Notice (U.S.) Planning Schedule/Shipping Schedule Goods Receipt
Trade/Payment Category	Commercial Invoice Remittance Advice
Transport/Logistics Category	Consignment Status Request/Consignment Status Report Transport Contract or Bill of Lading
Catalog Category	Price Catalog/Product Catalog
Statistical Reports Category	Accounting Report

TABLE 1 | Document schemas planned as UBL deliverables

UBL Subcommittees

Faced with a challenging job, the UBL Technical Committee has divided its organizational, semantic, and design work among a set of subcommittees working in parallel. Here are brief descriptions of the most interesting ones. For further details, see the UBL Web site at www.oasis-open.org/committees/ubl/.

Context Drivers SC

Part of the context methodology work is determining what are the "drivers" of context, that is, the axes along which context can be plotted. To the most obvious drivers, especially industry and region, have been added a set of additional drivers that can influence document formats: business process, product classification, official constraints, business process role, supporting role, and system capabilities. The job of this SC is to take this list, developed as part of ebXML, and determine whether any further drivers are needed.

UBL Context Methodology SC

This SC is tasked with the finalization of the context extension methodology published as a technical report after the first phase of ebXML. As a first step, the SC has gathered a list of use cases, real-world examples in which a given context requires modification of the standard document formats. These outline the specifics of everything from Brazilian invoice headers to documentation for shipping goods by sea. These use cases will be used to test and, where necessary, refine the ebXML methodology.

UBL Liaison SC

Success rarely comes in a vacuum, and this is doubly so of international standards efforts. UBL can only hope to reach its lofty goals by standing on the shoulders of the giants that came before it, including decades of work on EDI and the many more recent industry-specific groups. This SC is made up of UBL representatives who are charged officially with coordinating with other organizations involved in related work.

UBL Library Content SC

The central goal of UBL is to create a standard set of document schemas that can be used as the basis for e-commerce. The Library Content SC carries the responsibility for the actual design of the document formats. Its task has begun with a thorough review of xCBL, which has been retained as the starting point for the business library. Also essential to the work of this SC is the harmonization of UBL with the work of EDI's Joint Core Components and a whole range of industry-specific standardization efforts. The LC SC has made an early review copy of the UBL schema for Purchase Order available on the UBL Web site.

UBL Marketing SC

It's a well-worn cliché that technical excellence takes a backseat to marketing in achieving adoption in the IT industry. UBL has thus set up this SC to ensure that the goals and benefits of UBL are clearly communicated to outside parties. Responsibilities include drafting press releases and organizing participation in conferences and trade shows, as well as such niceties as chartering the design of a UBL logo.

UBL Naming and Design Rules SC

One of the hardest parts of creating a standard business library is ensuring consistent naming and structure of tags, types, and other constructs across all the document formats. This SC is finalizing a set of recommendations, to be used by the Library Content SC, specifying such things as when to use attributes rather than elements and how the name of a tag should relate to its type. These recommendations will also be made publicly available so that third parties can create their own extensions to UBL while conforming to its overall style. Snapshots of position papers on basic XML design questions are available on the UBL Web site.

UBL Tools and Techniques SC

Document design is difficult, and collaborative document design even more so. There is a wealth of schema editors, context management systems, forms generators, and other tools that can ease this task. This SC is charged with evaluating these tools and specifying a standard toolset to be used by the Library Content SC and recommended to third-party document designers.

Web Services + UBL = Interoperability

By defining what in essence are the basic interfaces for a complete set of business processes, the UBL effort will have huge implications for Web services. Consider, for example, a Web service for online payment. The core functionality of this service is to receive invoices, create payment request documents based on these invoices, and settle the payment through a bank payment gateway.

The defining principle of Web services is that a service of this type should be able to interact in a plug-and-play manner with other Web services, and that it should be replaceable by any other Web service targeting the same functionality. Without UBL this goal isn't achievable because the exact formats of the invoice and payment documents would have to be determined by the implementer of the system, so they wouldn't be interoperable with Web services from other vendors.

UBL solves this problem by providing standard formats for the invoice and payment documents. Any Web service that produces an invoice (a billing service, for example) can thus interface with the payment service. By using the context methodology, subtle differences in invoice and payment formats can be handled without invalidating the overall approach. In addition, the payment service can be swapped for another one, perhaps with some advantages, such as better terms, higher availability, or interfaces to more banks.

• • •

The technical framework being designed for Web services is extremely promising. The combination of a simple but coherent vision for discovering and invoking third-party services, strong industry support from virtually every major technology vendor, and avid attention from the market makes the eventual realization of the Web services vision a strong probability, if not a virtual certainty. But a level-headed assessment of the gaps that still exist, and the steps necessary to resolve them, is essential if disillusionment is to be avoided. The same hype that has made Web services the industry darling could easily turn against them, calling the whole enterprise into question.

One of the major challenges still facing Web services is the creation of a standard vocabulary to convey business semantics. This issue is not even on the table in any of the Web service–specific standardization efforts under way in the W3C or the newly formed WS-I. Fortunately, the UBL work initiated independently as an outgrowth of ebXML is perfectly placed to address Web services' dependence on universally accepted semantics, helping to allay one of the main concerns of Web service skeptics.

AUTHOR BIO

Matthew Gertner is cofounder and chief executive officer of Schemantix, a vendor of business services for enterprises and e-marketplaces. Prior to joining Schemantix, he headed the development of Internet- and XML-related products for POET Software, a leading provider of data management and supplier-enablement solutions. The author of numerous articles and a frequent speaker at industry events, Matt chairs the Context Methodology Subcommittee of OASIS's UBL Technical Committee.

MATTHEW.GERTNER@SCHEMANTIX.COM

Business process context	Official constraints context
Product classification context	Business process role context
Industry classification context	Supporting role context
Geopolitical context	System capabilities context

TABLE 2 Context drivers for context extension methodology

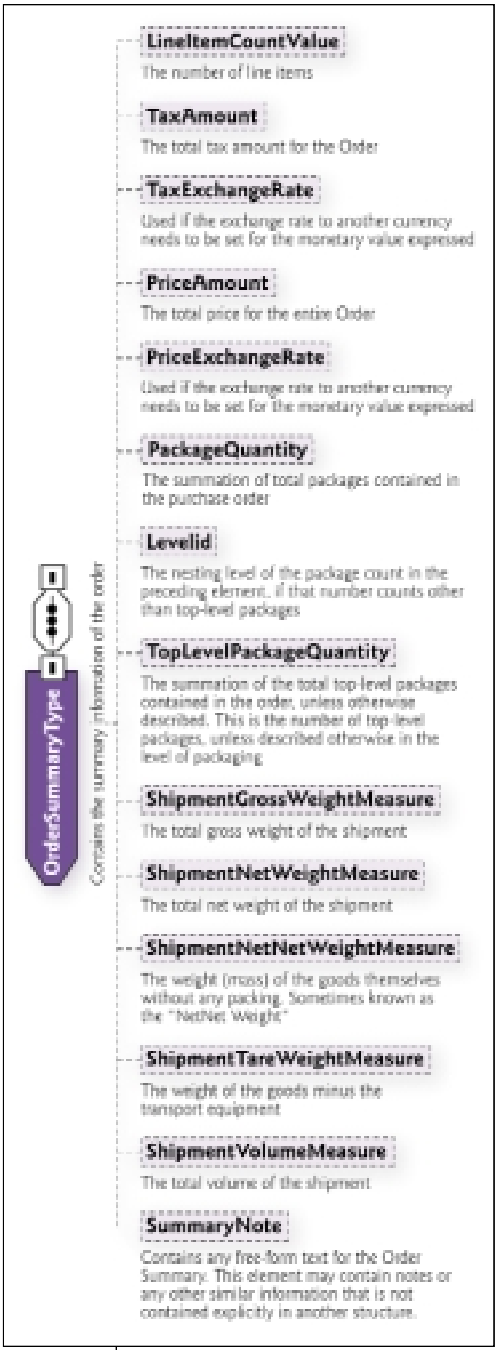

OrderSummaryType
Contains the summary information of the order
LineItemCountValue
The number of line items
TaxAmount
The total tax amount for the Order
TaxExchangeRate
Used if the exchange rate to another currency needs to be set for the monetary value expressed
PriceAmount
The total price for the entire Order
PriceExchangeRate
Used if the exchange rate to another currency needs to be set for the monetary value expressed
PackageQuantity
The summation of total packages contained in the purchase order
LevelId
The nesting level of the package count in the preceding element, if that number counts other than top-level packages
TopLevelPackageQuantity
The summation of the total top-level packages contained in the order, unless otherwise described. This is the number of top-level packages, unless described otherwise in the level of packaging
ShipmentGrossWeightMeasure
The total gross weight of the shipment
ShipmentNetWeightMeasure
The total net weight of the shipment
ShipmentNetNetWeightMeasure
The weight (mass) of the goods themselves without any packing. Sometimes known as the "NetNet Weight"
ShipmentTareWeightMeasure
The weight of the goods minus the transport equipment
ShipmentVolumeMeasure
The total volume of the shipment
SummaryNote
Contains any free-form text for the Order Summary. This element may contain notes or any other similar information that is not contained explicitly in another structure.

FIGURE 2 Preliminary draft schema for Purchase Order Summary

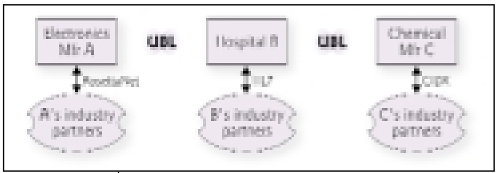


FIGURE 3 UBL as a universal interchange format

HOME
Enterprise Solutions
Content Management
Data Management
XML Labs

WRITTEN BY ROBERT LORENZ

Understanding an Application's True Age

When did you last look at your systems?

If you've been in the enterprise solutions industry for any length of time, you're doubtless familiar with the principle that every packaged application needs to go through a particular process of analysis, selection, and deployment. These are costly endeavors that require many internal and external resources and many months of work.

You're also familiar with the belief that companies desperately seek to enhance or customize these solutions, once installed, to fit their growing needs rather than throw them away and start again. Sometimes, however, companies have to face the difficult task of admitting that they've outgrown an earlier selection and need to move toward a new solution quickly or face the problems associated with undersized and poorly fitting existing applications. At times like this, what's needed is a new type of analysis that can determine the true age of an application based on the needs of the company, not the length of time in service.

Deloitte, Touche & Tomatsu work daily with such customers to assist in the analysis and integration process. We work hard to ensure our customers have the best solutions to fit their current and future needs. But sometimes a successful company will grow at such a fast rate that even the best-laid plans quickly become unsatisfactory to support the business.

We recently worked with one high-growth, privately held distribution company in the Midwest that is struggling to manage electronic data interchange (EDI), inventory, warehousing, vendor-consigned inventory, and distribution resource planning (DRP) systems with a diverse portfolio of homegrown and off-the-shelf applications. The company also has an enterprise resource planning (ERP) system that's less than 30 months old.

The Scenario

The company has grown 200% over the past 18 months and has added five new distribution warehouses. Clearly, this rate of growth was unforeseen at the time the original analyses were done on the systems in use. The company must now face the hard task of properly addressing the age of its application set and determine the next steps for systems development to support the current needs and size of the company.

The following list describes some of the issues the company faces because of its rapid growth:

1. The owners invested millions of dollars to implement and configure an ERP middle-market solution 30 months ago.
2. The company's core systems have limited functionality and quickly became undersized, meaning that it couldn't deliver on mission-critical management aspects of running the company.
3. The information technology group is mobilized to add custom-developed functionality and core module modifications to satisfy business requirements. The systems are being changed and configured in a manner unsupported by the vendor, thus requiring continuous maintenance.
4. Senior management is concerned that the demands on the IT group may be too heavy to support the growing needs of the business and that they're not effectively reporting key performance indicators (KPI) to the management team.
5. Due to a lack of functionality within the core systems, spreadsheets are being implemented in the business units to report results and manage subsets of transactions, making it even more difficult to track and manage a centralized information base.
6. The IT headcount continues to increase, thus raising support costs, yet the backlog of needed functionality is also increasing.
7. The rapid pace of development means less time for design and testing, resulting in increased application downtime.

In a typical large organization these types of issues would lead to finger-pointing and dissension over choices made just a short time ago. That's why it's critical that companies have another metric to point to that justifies the impact being felt: the application's true age. It isn't the fault of any individual or group that made these decisions – it's just that the business grew in an unexpected fashion. The application-age metric allows companies to confirm that the age was appropriate at the time of selection and that it has outlived its expectancy: the company must address the issue head-on...and move on.

Please note, this is not your "typical replace the legacy with new applications" advice. We're talking about very specific instances where the application was properly sized and selected during the analysis phase, but has outlived its life expectancy due to the availability of internal resources and the needs of the business. However, the ability to age

AUTHOR BIO
Robert Lorenz is a manager at Deloitte & Touche, LLP, where he focuses on high-growth industry clients. With over 13 years of industry and consulting experience, Rob brings a technology focus, process improvement experience, business sense, and bottom-line understanding to client engagements.

applications appropriately may help companies that need a methodology to identify if their system can be expanded further or needs to be phased out.

The following is a list of steps and agreements that the company must come to so as to address this issue:

1. Senior management must agree that technology is moving quickly and the right decision 30 months ago may not be the most cost-effective solution today.
2. The useful life of technology is determined by its effectiveness in the marketplace – specifically, to the individual business – not by a single metric of how long an application should remain effective. Be aware that metrics and benchmarks are subjective when it comes to an application's useful life.
3. Build the business case for a possible early retirement.

The Bottom Line

If you find your business struggling to provide IT solutions with a newer suite of technology, look at the application's real age (see Figure 1). You might be surprised. Address the true age of the application. Key areas to explore in your business case should include:

- The required modifications to the system in the pipeline and the growing backlog
- The system's current technology flexibility
- Availability of new or recent versions of off-the-shelf products
- A reduction in supported database platforms and applications
- An increase in the use of flexible technologies, such as XML

Most important, align the business strategic direction with an overall technology direction.

RLORENZ@DELOITTE.COM

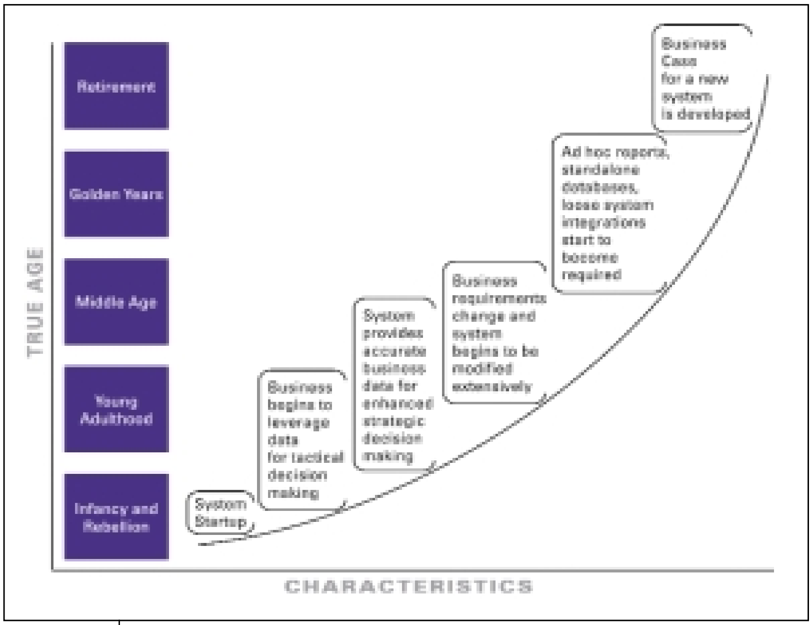

TRUE AGE
Retirement
Golden Years
Middle Age
Young Adulthood
Infancy and Rebellion
System Startup
Business begins to leverage data for tactical decision making
System provides accurate business data for enhanced strategic decision making
Business requirements change and system begins to be modified extensively
Ad hoc reports, standalone databases, loose system integrations start to become required
Business Case for a new system is developed
CHARACTERISTICS

FIGURE 1 | Application aging

WRITTEN BY **SIMEON SIMEONOV**

Can *You* Play the Standards Game?

The real drivers behind Web services standardization

Call me a cynic, but I don't think companies participate in standards development out of altruism. Enterprises are in the business of making their products and technologies successful. In an increasingly interconnected technology landscape, standards are the keys to interoperability.

Applying gross oversimplification, we can identify four types of players in the standards game: leads, coleads, followers, and bystanders. The category within which a company falls typically can hint at its goals in the standards game and can predict its behavior.

Leads need to effect change in the industry. The motivation could be offensive or defensive. Offensive change can help expand the business into new areas. Standards often enable the creation of new markets by gathering a critical mass of industry support and minimizing integration friction. Standards leaders can affect the shape of the future. For the price of heavier investment in the standards game they get the benefits of being perceived as innovators and having the inside track on how to evolve their own products and technologies.

Sometimes platform leaders use standards to simply increase the cost of competition by defining yet another thing that competitors need to comply with. Defensive standards leadership is primarily motivated by one of three forces:

- Building a coalition against one or more competitors
- Changing the rules of the game (changing the perception of what is of real value)
- Stalling the standards process to let internal product development catch up

Coleads are businesses that don't have the power to change the industry themselves but want to place bets on how the future will look. In offensive mode these companies are in the business of quickly implementing standards and sometimes providing value-add on top of them. This is the familiar embrace-and-extend model. More established companies tend to associate new standards with their existing product offerings in a meaningful way. Some startups tend to make standards the key part of their story. Coleads can also use the standards game in defensive mode by attempting to stall the process, just as leads sometimes do.

Followers are often the most altruistic participants in the standards game. These businesses appreciate the benefits that standards bring to products, customers, and the industry as a whole and are willing to help without getting in the limelight. The main benefit followers get is the ability to monitor standards evolution and report back to their product teams. Of course, followers can also do embrace-and-extend. One relatively uncommon type of follower is the spy who listens in on the standards debate and informs product teams that are building competitive technology.

De facto standards consortia, while not standards organizations in the traditional sense (e.g., ISO), provide the stage for the standards development spectacle. The Internet Engineering Task Force (IETF) was the original heavyweight standards body on the Internet. With the emergence of the Web we've started hearing much more about the W3C. Web services brought OASIS and, more recently, UDDI.org and WS-I (Web Services Interoperability Organization), which plans to supply best practices. The Java platform is managed via the JCP (Java Community Process), which is pretty much controlled by Sun. This is an example of how a single company that has some precious intellectual property (IP) often attempts to open the IP to use by others while maintaining control over it. This is typically done via the spinout of a dot-org site. Finally, many standards begin their life as white papers on corporate Web sites. IBM uses its alphaWorks site to stream new ideas and technologies to the public. Microsoft does this via MSDN.

Enlightened or not, self-interest rules in the standards space. The history of the Web alone is full of examples of companies and standards organizations battling for dominance in a rapidly changing environment. What are they fighting for?

What's the Prize?

Web services are fundamentally about connecting, integrating, and orchestrating applications. The Web services architecture supplies rich metadata at every level of the technology stack, making Web services very desirable plumbing for next-generation applications. This is the force driving the evolution of Web services – both the reality and the hype.

Many companies have rushed to provide core Web services capabilities: SOAP engines, WSDL tooling, and UDDI repositories. This is a passing fad. In the long run, few companies will make money on providing core infrastructure. More than a decade ago we used to pay for TCP/IP stacks and there were many companies building them. We no longer pay for TCP/IP stacks and most of these companies are no longer in business. This is how software has evolved – we keep layering abstraction upon abstraction, with every new layer decreasing the value of the layer under it.

Web services aren't an end in themselves; they're tools that will help speed up the inevitable evolution of software. Therefore, what you see in the Web services standards landscape has a lot more to do with the big battles going on in the industry right now than with the Web services standards process itself. These battles are mostly about platform dominance (with the term *platform* used very loosely). Here are some of the big battlefields I see right now:

AUTHOR BIO

Simeon Simeonov, VP, emerging technologies, at Macromedia, Inc., is a member of the W3C working group on XML Protocol and the J2EE expert groups on XML business messaging and XML data binding.

- Core software platform
- Business-to-business integration platform
- Security platform

There are others, but these are the three I'd like to spend some time discussing.

The battle for the core software platform is essentially the battle between Java and .NET. At the end of the last millennium Microsoft was behind the eight ball. The Java camp had a much better openness and interoperability story. The focus on Web services was an excellent strategic move on Microsoft's part. It changed the rules of the game. It led to aggressive innovation in an area that essentially erases many of the advantages of the Java platform with respect to connecting and integrating systems. Follow that with .NET and you see how Microsoft has made an excellent jump shot and is well positioned to get ahead in the game.

If we look at history, a company like Microsoft should be able to execute faster than the loosely coupled Java camp on essentially the same set of platform priorities. Sun, behind in Web services innovation since 1999, is just starting to catch up. This is one reason why we see key members of the Java coalition such as BEA breaking away from the pack and quickly moving to independent Web services innovation with WebLogic Workshop (Cajun). The tension between Sun, the owner of Java, and the rest of the Java camp is palpable. A telling sign is that while IBM is a founder of WS-I and most J2EE vendors are members too, Sun is conspicuously absent. This is a dangerous game that's likely to backfire. IBM deserves praise. ITS researchers are working closely with Microsoft on standards development and with Sun on integrating them with J2EE.

The second battleground is for the EAI/B2B platform of the future. Traditionally, integration has been so difficult that it's been a business only for system integrators and high-end enterprise software vendors. In a Web services world the rules of the game change. If connectivity becomes easy, and if there's plenty of metadata to go around, this type of integration will become much more common. In other words, it will descend into the area where platform companies play. Microsoft's product in this space – BizTalk Server – has been selling quite well. BizTalk's orchestration language is XLang. IBM's standards proposal in this space is WSFL (Web Services Flow Language). The two companies have been working for months on integrating the two into a single proposal. Understandably, Microsoft has been resisting some change since they have a product in the market that's selling quite well. Other players have other approaches, but they're likely to get marginalized following a joint MS/IBM standards proposal. If I were the CTO of a workflow or business process automation (BPA) company, I'd start worrying.

Another important area is security, not in the application-centric authentication/authorization sense but in the much broader platform sense. A platform view of security starts with authentication and authorization and adds confidentiality, privacy, trust, and integrity to all interactions as well as federation, delegation, and audit capabilities to the security infrastructure. It's a difficult problem, one we've carefully sidestepped in the Web services space for more than two years.

In early April Microsoft and IBM published a white paper, "Security in a Web Services World: A Proposed Architecture and Roadmap." It's recommended reading for anyone interested in enterprise-quality Web services. Two days before the publication of this paper, IBM, Microsoft, and VeriSign published a revised version of the WS-Security specification that Microsoft rolled out as part of its Global XML Architecture (GXA) in fall 2001.

It's only going to get more interesting from here since the stakes are so high. The real battle is for ownership of the core security infrastructure in a service-enabled environment. The end goal is ownership of identity and trust brokering services, which all meaningful service interactions require. Microsoft fired the first shot with Passport, which prompted the formation of the Liberty Alliance led by AOL, Sun, and others. Pressure from industry forced Microsoft to move from a "we own it all" model to a federated model for managing identity. Microsoft is willing to bet that it can win on an execution game. To that end, the company needs an open playing field. Platform security standards that define how disparate security models such as Kerberos and PKI can interoperate serve them very well. It's also nice that the whole industry will benefit. The ones that will suffer are security companies that are forging ahead with other models to bring order to the mess that's out there right now. My advice to them: figure out how to get involved; you need to be coleads in these new efforts or you'll get marginalized.

The three battles outlined here set the platform context for a number of battles in vertical segments, such as insurance and health care. Traditionally, XML standards development in these areas has focused on building coalitions around domain-specific XML Schema. In a Web services world the schema will be augmented with WSDL service API descriptions and UDDI category extensions that help service providers and requesters hook up.

Learning to Play

What does this mean for you? Well, if you're in no way connected with establishing technology, product, or business strategy at your company, your time would probably have been better spent not reading this issue's column. However, if you're a strategic thinker, you need to seriously consider how you want to participate in the Web services revolution.

If your company doesn't have at least some past deep experience with standards development, you probably shouldn't get in the game until you bring in talent that can help. The process is long and exhausting. It takes a lot of time and energy to drive standardization. The politics can get nasty. And yes, many standards meetings are boring.

Finally, just for the record, I'm not trying to sound like an anti-Microsoft spokesperson. I just think the company is a great case study in how to play the standards game. Watch and learn.

SIMEONS@MACROMEDIA.COM

How Do *You* Want to Participate in the Web Services Revolution?

1. What are the battles you need to win? Focus on business, not technology/product objectives.
2. If some battles are related to establishing horizontal or vertical platforms where connectivity, integration, and orchestration of applications matter, you need to evaluate the use of Web services standards.
3. How does innovation happen? How is it communicated to the rest of the world? How will you switch between innovation and execution? What is the end goal of the effort and how will it fit within the existing fabric of standards?
4. What do you really need? The perception that something is a standard to eliminate adoption objections or the reality of broadly deployed standards-based implementations so that true interoperability can happen?
5. Consider how you fit in the industry ecosystem. Do you have the influence and resources to be a standards lead? Can you realistically colead an existing effort? Is it worth your effort to actively follow? How will you handle commoditization brought on by standards? Can you meaningfully embrace-and-extend?

WRITTEN BY **BEVERLY HRABLOOK**

Marconi Embraces XML

How one company publishes and delivers complex product information

Every large corporation is a publisher, whether or not it knows it. User manuals, installation guides, repair manuals, corporate information, even internal documents like employee handbooks take weeks to draft, finalize, publish, and distribute. Ask any corporate information specialist or librarian.

As a global provider of advanced communications solutions like optical networks, Internet routers, and intelligent switches for phone companies, Marconi faces the challenge of managing the creation of lengthy, complex product- and service-oriented documents. For example, the creation of a 4,000-page user guide for an optical network demands tremendous collaboration among Marconi's technical writers. Additionally, Marconi solutions are frequently tailored to the specific needs of each customer. This means that technical documents, even within product lines, are rarely the same.

Fortunately, advanced publishing and authoring technology has helped Marconi overcome many of these challenges and has even opened up opportunities, such as being able to simultaneously publish to print, the Web, and CD-ROM. Using the same technology, Marconi will ultimately enable technicians in the field to wirelessly access relevant portions of huge manuals on handheld devices (see Figure 1). For customers this means instant access to technical information that helps keep Marconi systems up and running.

The enabling technology behind this solution is XML, coupled with authoring and publishing software from Adobe Systems, Inc. (specifically, Adobe FrameMaker, Adobe Acrobat 5.0, and the Adobe Portable Document Format, or PDF).

The Challenges of Creation and Delivery

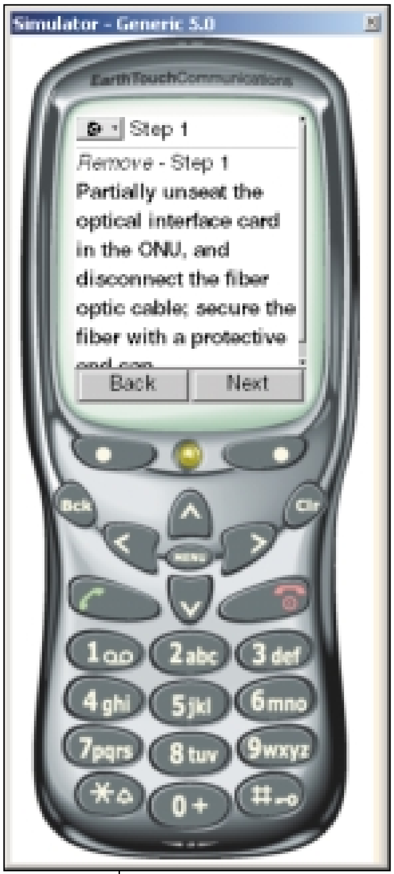
The challenges of Marconi's information management process can be broadly categorized as (1) information creation and (2) information presentation and delivery.

On the creation side, it's crucial to manage collaboration among many authors, maintain document structure consistency, integrate graphic objects and data from external sources, and manage archival and version control of source documents. Before Marconi adopted the XML- and Adobe software-based workflow, dispersed writing teams used a variety of document authoring tools. As a result, sharing information among groups was sometimes difficult and confusing. Assembling the documents was equally challenging. Imagine changing one product modification in every reference in a 4,000-page manual!

On the delivery side there's a need for high-quality print versions as well as publishing to the Web and to CD-ROM. There is also the issue of keeping information current and tailoring it to the specific customers' needs. Finally, information must be available in the form that users require. (It's unrealistic to expect a technician to carry a 10-pound manual up a telephone pole.)

XML, FrameMaker, and Adobe PDF to the Rescue

XML provides the foundation for solving these publishing challenges through its defined document structure, standardization of content imported from diverse systems, and delivery of customized content to multiple output media. For example, XML enables writers to make one change and have it reflected in every passage across multiple chapters or even manuals. XML also enables the creation of complex templates that simplify the authoring of huge texts, and enables those texts to be transferred to other media like the Web or to wireless devices.



Simulator - Generic 5.0

EarthTouchCommunications

Step 1

Remove - Step 1

Partially unseat the optical interface card in the ONU, and disconnect the fiber optic cable; secure the fiber with a protective end cap

FIGURE 1 | WAP phone with WML from XML of the same document

AUTHOR BIO

Beverly Hrablook, Marconi's director, information solutions and global standards, leads a team implementing a global content management system. Marconi provides hardware and software for the communication and information industries. A telecommunications engineer, Beverly is working on revolutionizing the way Marconi's information is created, managed, and delivered.

LARGEST GATHERING OF DEVELOPERS, PROGRAMMERS, AND *i*-TECHNOLOGY PROFESSIONALS IN THE WORLD!!

JUNE 24-27, 2002 • JACOB K. JAVITS CONVENTION CENTER • NEW YORK, NY

WIN A $35,000 LUXURY CAR!

ATTENDEES WILL BE INVITED TO TAKE A GOLF SWING TO WIN AND RIDE OFF IN A $35,000 LUXURY CAR!

Sponsored by webMethods.

REGISTER BY MAY 31 TO SAVE $200!

GO TO WWW.SYS-CON.COM NOW!

JAVA XML WEB SERVICES

Focus on Java

JAVA
IN
JUNE
ESPECIALLY
IN NEW YORK

Java, now mainstream, is the dominant back-end technology upon which next-generation technologies are evolving.

Hear from the leading minds in Java how this essential technology offers robust solutions to *i*-technology professionals and senior IT/IS strategy decision-makers.

JDJ EDGE conference & expo

Java Track: June 25-27

The Java Track, with something for every developer from beginner to advanced, will also look into the role that Java is playing in building up Web Services.

JV1 Java Security Advanced Concepts
JV2 Optimizing Database Performance in J2EE Applications
JV3 Detecting, Diagnosing, and Overcoming the Five Most Common J2EE Application Performance Obstacles
JV4 Building Asynchronous Applications Using Java Messaging
JV5 Building "Smart Client" Applications using J2SE and J2ME
JV6 .NET vs J2EE
JV7 Building Scalable Web Applications and Web Services
JV8 Hot Breaking Session
JV9 Java Tools for Extreme Programming
JV10 Building Truly Portable J2EE Applications
JV11 Security for J2EE Application Servers

Focus on XML

XML
NEXT G
OF
ENTERPRISE
DEPLOYMENT

XML is today's essential technology to develop and deploy Web services. Here's your chance to learn from expert practitioners how XML is making the next phase of the Web a reality.

Focus on standards, interoperability, content management, and today's new quest for more efficient and cost-effective Internet and intranet-enabled business process integration.

XML EDGE conference & expo

XML Track: June 25-27

The XML Track will focus on the various facets of XML technologies as they apply to solving business computing problems. This track targets audiences ranging from beginners to system architects and advanced developers.

XM1 Data - a Key Part of Web Services
XM2 OASIS Standards Update
XM3 A Universal Business Language
XM4 Achieving Standards-Based Mobile eBusiness Success with XML & Web Services
XM5 Using XML for Rapid Application Development and Deployment with Web Services
XM6 Open Up Your RDBMS with Open Standard Technologies
XM7 XML Web Services: Standards Update
XM8 Bringing XML to PKI
XM9 Building a Corporate Information Model in XML
XM10 XML in the Enterprise and Inter-Enterprise World
XM11 XML and RDB Relationships

Focus on Web Services

WEB
SERVICES
SKILLS,
STRATEGY,
AND
VISION

Web Services, the next generation technology that will enable the Internet to work for you and your business, and finally provide that ROI you have been after, will be put under a microscope at Web Services Edge East 2002.

Information-packed sessions, exhibits, and tutorials will examine Web Services from every angle and will provide cutting-edge solutions and a glimpse at current and future implementations. Hear from the innovators and thought leaders in Web Services. Enjoy a highly interactive CEO Keynote panel that will debate and discuss the realities and promise of Web Services.

web services EDGE conference & expo

Web Services Track: June 25-27

The Web Services Track will concentrate on the latest emerging standards. It will include discussions of .NET, Sun ONE, J2EE and App Servers, the role of UDDI, progress of the standards-making bodies, SOAP, and Business Process Management. It is intended for developers, architects, and IT management.

WS1 Starting Out In Web Services: Fundamentals In Web Services
WS2 State of the Web Services Industry
WS3 Web Scripting Languages: Options for Dynamic Web Development
WS4 Building a Web Services Application Infrastructure
WS5 Deploying a Corporate Portal
WS6 The Enterprise Service Bus: (ESB): Leveraging Web Services
WS7 Developments in Web Services Standards
WS8 Unlocking the Value of Enterprise Web Services
WS9 Guarding the Castle: Security and Web Services
WS10 Practical Experiences with Web Services and J2EE
WS11 Designing Web Services Using UML

.NET TRACK: June 25-27

Combining technology, platform, and architecture, the .NET Track provides insight on the latest developments for the Windows platform.

NT1 Configuring .NET for Security, Performance, and Reliability
NT2 Changing Your Environment to .NET
NT3 Going Mobile with .NET
NT4 .NET on Other Platforms (FreeBSD, MONO, Portable .NET)
NT5 Inside the CLR
NT6 Accessing Data from .NET
NT7 Advanced .NET Web Services
NT8 Migrating Legacy Code to .NET
NT9 Advanced User Interfaces with GDI+

IT Strategy Track: June 25-27

The IT Strategy Track will focus on the managerial and design aspects of the various development disciplines. Topics will include Standards, Architecture, Design Patterns and Best Practices, Project Planning and Management, and Extreme Programming.

IT1 Developing, Deploying and Managing Web Services
IT2 Key Trends and Technologies for Building an Enterprise Web Services Architecture
IT3 Selecting a Framework: Toolkit, Platform, or Roll Your Own?
IT4 Getting Started with Web Services
IT6 Overcoming the Web Services Barriers
IT7 The Real Issue: Improving Your Enterprise with Enterprise Web Services
IT8 Minimizing Risks and Maximizing Investments in J2EE Development Through the Use of Reusable Application Architecture and Frameworks
IT9 Application Integration - Building a Flexible Web Services Architecture
IT10 The Economics of Web Services
IT11 Hooking It All Together - Application Integration Case Study

Who Should Attend...

- DEVELOPERS, PROGRAMMERS, ENGINEERS
- *i*-TECHNOLOGY PROFESSIONALS
- SENIOR BUSINESS MANAGEMENT
- SENIOR IT/IS MANAGEMENT/C LEVEL EXECUTIVES
- ANALYSTS, CONSULTANTS

Sun Microsystems at JDJEdge 2002: Java Fast Paths & Java University℠ Program at the Jacob K. Javits Convention Center, NYC

Java Fast Paths

Attend fast-paced, code-level, full-day Java technology developer training designed to provide you with the skills to confidently approach the industry's core Java technology certification exams. Don't just say you know it… prove it! Most developers recognize the expanding importance of gaining Java technology certification. If you're like those who've already taken the exams, the real hurdle to certification is finding time to prepare for the tests.

Monday, June 24, 2002:

- Java™ 2 Platform: Developer Certification Fast Path
- Java™ 2 Platform: Architect Certification Fast Path
- Web Component Developer: Certification Fast Path

Java University℠

The Java University℠ Program complements this year's JDJEdge Conference by offering three aggressive, full-day, code-level training classes for experienced software developers, architects and engineers.

Attend code-level training in sessions designed by industry luminaries, and recognized experts. Sessions cover XML and Web services technology. Whether you're a beginning or a veteran developer, architect, or software engineer, you'll benefit from these value-packed full-day courses. Register now. Seating is limited.

Tuesday, June 25, 2002
Developing Solutions Using Java™ Technology and XML

Wednesday, June 26, 2002
Web Services Programming Using Java™ Technology and XML

Thursday, June 27, 2002
Java™ APIs for Enterprise Web Services

FOR EXHIBIT INFORMATION CONTACT: MICHAEL PESICK 135 CHESTNUT RIDGE RD. • MONTVALE, NJ 07645 • 201 802-3057 MICHAEL@SYS-CON.COM

OWNED AND PRODUCED BY SYS-CON EVENTS SYS-CON MEDIA

Adobe PDF and Acrobat play a critical role in the document review and distribution processes. Documents saved in PDF maintain precise fidelity with the printed page, while Acrobat software provides sophisticated Adobe PDF review and commenting components. XML by itself doesn't yield high-quality formatting for print. PDF provides information managers and publishers with formatted print output and the ability to electronically exchange ready-to-print documents.

FrameMaker (see Figure 2), the critical bridge between XML and PDF, acts as a powerful tool for importing, authoring, and exporting XML. In most cases the document begins as a FrameMaker template, designed to handle the huge amount of linked data that goes into a complex product manual. Tables of contents, cross-references, chapter headings, appendices, and links are easy to set up with the software. Authors can use FrameMaker tools to import graphics, text, and other visuals into the template. FrameMaker templates lend a consistent look and feel to all documentation, and provide easy controls for applying uniform attributes and values to lengthy materials.

Authors also use FrameMaker to manage and monitor the structure of the document. For example, the software instantly informs users if they try to change the paragraph format or other elements that would compromise the overall XML structure. Authors can navigate the tree structure of the XML document, and imported objects can be manipulated within FrameMaker's Structure View. An author can make one change and have it reflected across the entire manual, thanks to XML.

FrameMaker also provides mapping of XML structure and context to styles for print format. FrameMaker's WYSIWYG authoring environment lets writers know that their document conforms to the XML DTD and allows them to see exactly how it will look when printed.

Better Collaboration Between Authors

With Acrobat and PDF (see Figure 3), companies can shorten document review cycles and improve the accuracy and completeness of materials. Marconi distributes technical document drafts in Adobe PDF to approximately 40 reviewers. Because of the tight integration between FrameMaker and Adobe PDF, draft documents retain hyperlinks, tables of contents, and index references – all of which facilitate easy navigation. Using the Acrobat review and commenting tools (such as text highlighting, text strikethrough, and electronic sticky notes), reviewers can comment on PDF drafts that look exactly like the documents in the native application. An additional Acrobat feature – digital signatures – is being considered to further streamline document review and sign-off.

Redefining Real-Time Information Access

This dynamic document creation system will ultimately allow customers to go online, search for information, and instantly generate pages – or entire manuals – containing the technical details they need. Information from a FrameMaker document can be exported as XML, which is then stored in a database publishing system that Marconi customers and staff can access via the Internet. The final documents, which the system can automatically convert to PDF, will be available around the clock to customers worldwide.

The advantages for the Marconi field staff are equally impressive. Technicians will be able to search for and view parts information on handheld devices and troubleshoot problems in real time.

XML Gains Traction in the Enterprise

As Fujitsu Network Communications moves in the direction of XML and WebDAV integration, the company wants to store technical documentation, authored in FrameMaker, as XML files on a Web server. This will enable employees to check in and check out files from within FrameMaker, and allow others in the company to view the stored files from within their Web browsers over the corporate intranet.

Agilent's Learning Products group uses Adobe FrameMaker software to author documents ranging from business card–sized quick references to a 1,700-page dictionary reference for complex software, as well as comprehensive online help. Agilent then uses Adobe Acrobat to convert the documents to Adobe PDF for internal review and for distribution to customers via CD-ROM and the Web.

Single Source, Unlimited Options

Marconi plans to institute the dynamic document assembly workflow company-wide this year. Ultimately, FrameMaker, Acrobat, and XML will enable the generation of an infinite number of documents from the same content repository. Users can assemble the information they need, print it to paper, or convert it to an Adobe PDF document for online delivery. With Adobe software and XML, Marconi is finding new ways to cost-effectively respond to customers' diverse needs.

BEVERLY.HRABLOOK@MARCONI.COM

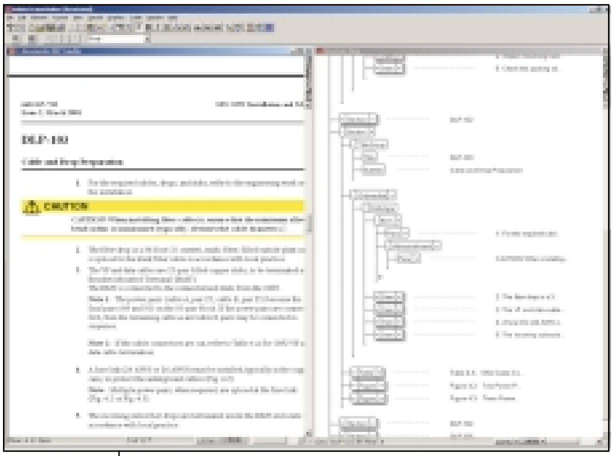


FIGURE 2 | Marconi document in FrameMaker 7

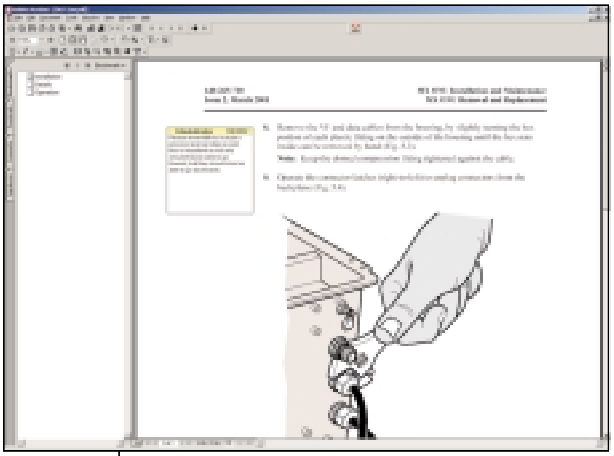
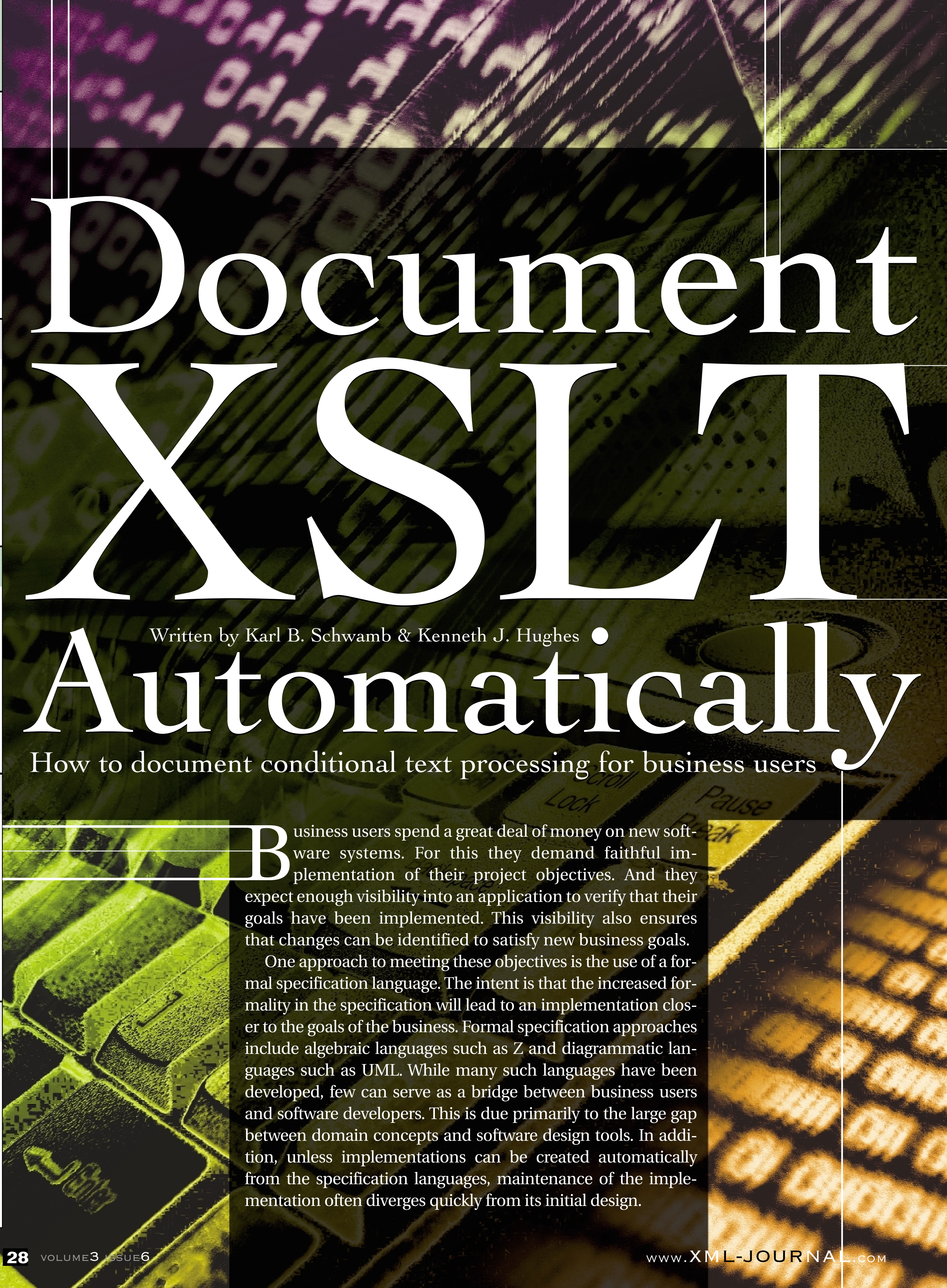


FIGURE 3 | Adobe PDF of the same document in Acrobat 5



Document XSLT Automatically

Written by Karl B. Schwamb & Kenneth J. Hughes

How to document conditional text processing for business users

Business users spend a great deal of money on new software systems. For this they demand faithful implementation of their project objectives. And they expect enough visibility into an application to verify that their goals have been implemented. This visibility also ensures that changes can be identified to satisfy new business goals.

One approach to meeting these objectives is the use of a formal specification language. The intent is that the increased formality in the specification will lead to an implementation closer to the goals of the business. Formal specification approaches include algebraic languages such as Z and diagrammatic languages such as UML. While many such languages have been developed, few can serve as a bridge between business users and software developers. This is due primarily to the large gap between domain concepts and software design tools. In addition, unless implementations can be created automatically from the specification languages, maintenance of the implementation often diverges quickly from its initial design.

Consequently, most business users rely on documentation to explain the inner workings of a system. This addresses the critical need to gain visibility into the system to ensure that their goals have been met and the system can be changed easily to adjust to new business concerns. The value of documentation depends on:

- How well it conveys an understanding of the system
- How easily modifications can be performed
- How well the documentation can be kept up to date with new changes

The area of literate programming addresses this concern somewhat – although the most widely used tools, such as Javadocs, are used by programmers for the benefit of other programmers.

This article discusses a method of automatically documenting, in domain-specific terms, the behavior of conditional text processing applications. The use of such terms, as well as actual text in its domain-specific format, yields a small gap that can be readily bridged by business users. The documentation presents a form that can be marked up by business users with minimal ambiguity. Automatic generation of the documentation ensures that it remains faithful to each build of the application.

Conditional text processing is a very large horizontal application area with potential impact on much of literate society. It affects areas as diverse as traditional print document production, Web page generation, document personalization, targeted advertising, and access-controlled documents. Customized text processing is bound to increase rapidly with the trend toward information delivery that is increasingly personalized, access controlled, and market-segment specific.

Representing Text

There are many ways to describe textual content that can be manipulated programmatically. Historically, this subject has been addressed with a myriad of ad hoc and proprietary formats. However, the SGML community has long recognized the benefits of formats that are open, standard, and domain defined. The phenomenal success of HTML (whose format followed that of SGML) and the need for the flexibility of domain-defined markup combined to motivate the W3C to recommend the similar but streamlined XML format as a basis for all Web-based content.

In the domain of technical documentation, DocBook is a well-accepted XML application that can be used as an intermediate form for generating text in a variety of formats including plaintext, XHTML, RTF, TeX, PDF, and PostScript. DocBook is directed at the production of articles or books. Common tags include <book>, <chapter>, and <para>. Listing 1 presents the skeleton of a document in DocBook format. The document is intended to represent a fragment from a financial planning document.

XSLT for Conditional Text

XSL is the W3C stylesheet standard for XML documents. It includes a language, XSLT, to transform XML documents. XSLT supports several programming styles but we focus on the "fill-in-the-blanks" style as identified by Michael Kay in his book, *XSLT Programmer's Reference*. This style is useful when a target XML document is to be produced by filling in missing items using data provided by an XML data document.

Listing 2 presents an XML data document that includes data about an individual customer. While we won't present the details of the XML DTD (or XML Schema) defining the document structure, it should be evident that it represents properties of a single individual such as age and estimated financial net worth. It's assumed that some other calculation process has determined this information based on data about the individual's financial status.

Listing 3 presents XSLT markup that's been added to Listing 1. The markup includes both a simple data substitution and a simple conditional statement. It uses XPath, another W3C standard and part of XSL, to refer to the data in the XML data document. The XPath reference "customer/name" refers to the customer's name in the XML document given in Listing 2.

Listing 4 illustrates the resulting DocBook document produced by the XSLT transformation of Listing 3 to the XML data in Listing 2. We've used Michael Kay's Saxon processor to generate this example and have rendered this DocBook document into XHTML (see Figure 1) to show how it can be presented to the business user.

Documenting XSLT Processing

Since the XSLT program is itself an XML application, we can apply other XSLT programs to it. In fact, our method maps all XSLT elements used in an XSLT transformation into textual elements used in the target XML document. Since each XSLT element is mapped in this manner, this produces documentation that can be rendered in the same fashion as the documents of the domain. This ensures that the documentation will be familiar to the business users, as it appears as a normal document with the addition of pseudocode annotations.

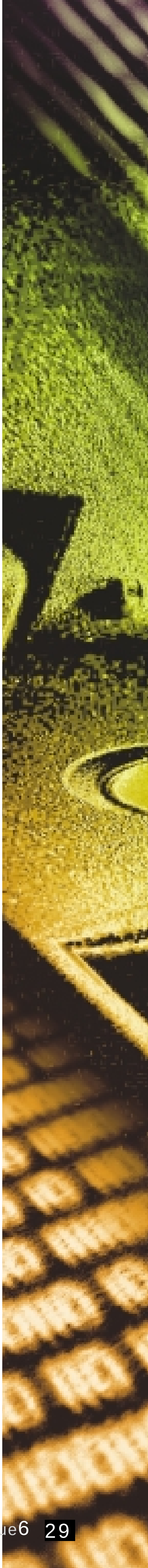
Listing 5 presents the XSLT program that maps the XSLT program of Listing 3 into a target DocBook document. As this is the most critical step in the process, we elaborate on this example – line numbers have been added to the left-hand side of the listing for reference. Line 1 simply identifies the file as an XML document in a Latin1 character encoding. Line 2 declares an XSL transformation using the standard XSL namespace. The next two lines specify the public and system identifiers that the output file should include to identify the document as a DocBook file. Line 6 indicates that excess white space is to be stripped out of all elements.

The remainder of Listing 5 consists of five templates (i.e., specific transformations). The template at Line 7 matches those XSL elements whose content should be processed further but without any special consideration due the containing XSL element. The Line 10 template ignores all markup inside the xsl:output elements. The template at Line 11 performs the first actual metamarkup of the output document by outputting (source-data-name) for each fragment of text to be pulled from <customer> input data. Similarly, at Line 18, metamarkup of the form (IF condition text) is generated to express the condition for which text should be included in the output document. The final template at Line 26 is a common XSLT default processing rule that simply copies unmatched markup to the output file.

Listing 6 presents a mapping from XSLT variable names to domain-specific names used by business users. This step is syntactic sugar for increasing readability beyond that provided by clearly named XML elements in the input data file. Listing 5 Line 32 includes routine XSLT code that performs this mapping. In production, this mapping was performed in a postprocessing phase via a Perl script.

Listing 7 presents the DocBook markup produced by applying Listing 5 to Listings 3 and 6. In practice, we've termed this a *specification* because it precisely specifies the operation of the XSLT program in producing the target documents.

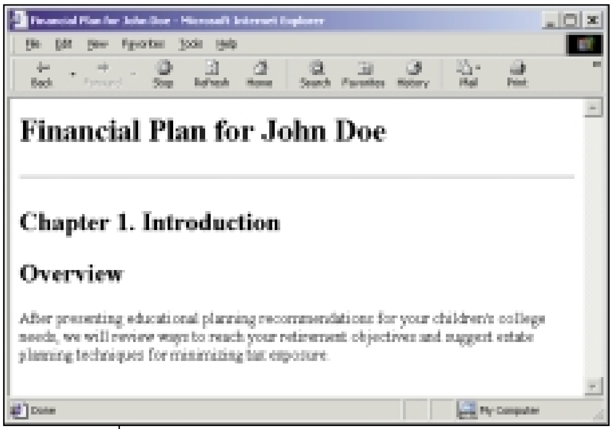
Figure 2 illustrates the XHTML presentation of the DocBook document presented in Listing 7. Two extra files (autodocxslt.xsl and mapnames.xsl) that support the build process are too insignificant to warrant including in the listings. However, they can be downloaded from www.sys-con.com/xml/sourcec.cfm so interested readers can build the examples. (A README file that explains all the files is included.)



Discussion

The method outlined above has been used to map a number of other XSLT constructs into text including variable-length lists and loops. These items may appear in a variety of textual contexts including page headers and footers, section headers, tables, bulleted lists, multicolumn layouts, and glossaries.

However, a more general treatment of the subject is difficult because the method would need to account for the appearance of any XSLT element in any textual context. In practice, the XSLT documentation program has been developed by accounting for every XSLT construct used along with every DocBook context it appears in. To ensure that we've accounted for all possibilities, each specification is tested using James Clark's nsgmls validator to ensure that the markup conforms to the DocBook DTD. Because of this constraint, it isn't sufficient to have a properly working XSLT program – it must be translatable into valid DocBook as well. If DocBook directly supported metamarkup constructs, or if we targeted a different output format that provided such direct support, the challenge of choosing output representations for metalevel markup while simultaneously maintaining validity could have been avoided.


Financial Plan for John Doe - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print
Done
My Computer

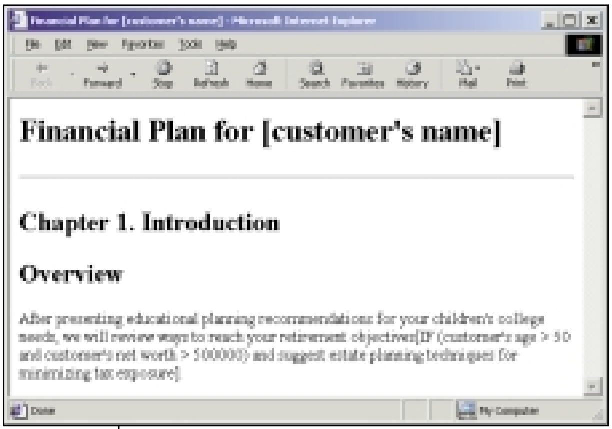
Financial Plan for John Doe

Chapter 1. Introduction

Overview

After presenting educational planning recommendations for your children's college needs, we will review ways to reach your retirement objectives and suggest estate planning techniques for minimizing tax exposure.

FIGURE 1 | XHTML rendering of output document


Financial Plan for [customer's name] - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print
Done
My Computer

Financial Plan for [customer's name]

Chapter 1. Introduction

Overview

After presenting educational planning recommendations for your children's college needs, we will review ways to reach your retirement objectives[IF (customer's age > 50 and customer's net worth > 1000000) and suggest estate planning techniques for minimizing tax exposure].

FIGURE 2 | XHTML rendering of generated documentation

Another aspect of this method that requires attention is the translation of expressions used in conditionals, loops, and other XSL statements. The method assumes that all expressions can be transformed by a simple replacement of XPath references into short, descriptive English names. If the expressions are more complex, additional processing may be needed on the expression to render it into readable form. For example, if a call is made to the XSL "format-number" function to depict a number as a dollar amount, then a U.S business user would prefer to see the expression rendered with a leading dollar sign:

```
format-number("customer/netWorth", #,###,###.00)
  => $customer's net worth
```

Conclusions

This method works well for several reasons.

1. The XSLT transformation itself is represented as markup so that it can be easily manipulated by other XSLT transformations. If the transformation were written in conventional 3GL, a programming language parser would be needed along with custom code for handling the reference to text elements.
2. The programming style used in the XSLT transformation was limited to the "fill-in-the-blanks" style that did not require a general solution to mapping programming constructs into text markup such as DocBook. A rule-based programming style would have been very difficult to map into the proper text structures.
3. There is a small distance between an instance of the target document and its automatically documented specification, making it easier for business users to comprehend.
4. The use of domain-specific business terms instead of program variables helped to make the XSLT control flow statements easier to understand.

This method is very useful for communicating conditional text processing to business users because it expresses processing in terms and context that are familiar to users. The precise translation of each XSLT element into readable text makes the produced documentation an ideal tool for maintaining the XSLT program as the produced documentation always matches the operating program.

For More Information

- Clark, J. XSL *Transformations (XSLT) Version 1.0, W3 Recommendation 16 November 1999*: www.w3.org/TR/xslt
- Walsh, N., and Muellner, L. (1999). *DocBook: The Definitive Guide*. O'Reilly. www.docbook.org/
- Kay, M. (2000). *XSLT Programmer's Reference*. Wrox Press.
- *SAXON XSLT Processor:* http://saxon.sourceforge.net/

AUTHOR BIOS

Karl B. Schwamb is president of Colonnade Software (www.colonnadesoftware.com), a consulting firm specializing in distributed systems. He has led systems design and development efforts for several Fortune 500 companies, primarily in the area of financial services.

Kenneth J. Hughes is president of Entelechy Corporation (www.entel.com), a consulting firm that specializes in XML. He has provided strategic guidance, architectural design, and hands-on development for organizations seeking to apply XML to both traditional publishing and Internet-based systems.

KBS@COLONNADESOFTWARE.COM

KJH@ENTEL.COM

This work was supported by ExpLore Reasoning Systems, Inc., a firm specializing in intelligent systems for financial services applications.

LISTING 1 Plain DocBook sample

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE book
  PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
   "docbookx.dtd">
<book>
  <title>Financial Plan</title>
  <chapter>
    <title>Introduction</title>
    <section>
      <title>Overview</title>
      <para>
        After presenting educational planning
         recommendations for your
        children's college needs, we will review ways to
         reach your
        retirement objectives.
      </para>
    </section>
  </chapter>
</book>
```

LISTING 2 XML input data

```
<?xml version="1.0" encoding="utf-8"?>
<customer>
  <name>John Doe</name>
  <age>52</age>
  <netWorth>2000000</netWorth>
  <!-- other customer data -->
</customer>
```

LISTING 3 XSLT transformation from input data to output document

```
<?xml version="1.0" encoding="iso-8859-1"?>
<xsl:stylesheet xmlns:xsl=
 "http://www.w3.org/1999/XSL/Transform"
                version="1.0">
  <xsl:output doctype-public="-//OASIS//DTD DocBook XML
   V4.1.2//EN"/>
  <xsl:output doctype-system="docbookx.dtd"/>
  <xsl:template match="/">
    <book>
      <title>Financial Plan for <xsl:value-of select=
       "customer/name"/></title>
      <chapter>
        <title>Introduction</title>
        <section>
          <title>Overview</title>
          <para>
            After presenting educational planning
             recommendations for
            your children's college needs, we will
             review ways to
            reach your retirement objectives<xsl:text/>
            <xsl:if test="customer/age &gt; 50
                    and   customer/netWorth &gt; 500000">
              and suggest estate planning techniques
               for minimizing
              tax exposure<xsl:text/>
            </xsl:if>.
          </para>
        </section>
      </chapter>
    </book>
  </xsl:template>
</xsl:stylesheet>
```

LISTING 4 Output document in DocBook

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE book
  PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
   "docbookx.dtd">
<book>
  <title>Financial Plan for John Doe</title>
  <chapter>
    <title>Introduction</title><section>
      <title>Overview</title>
      <para>
        After presenting educational planning
         recommendations for your
        children's college needs, we will review ways
         to reach your
        retirement objectives and suggest estate planning
         techniques
        for minimizing tax exposure.
        </para>
    </section>
  </chapter>
</book>
```

LISTING 5 XSLT documentation generator

```
<?xml version="1.0" encoding="iso-8859-1"?>
<xsl:transform xmlns:xsl=
 "http://www.w3.org/1999/XSL/Transform"
        version="1.0">
  <xsl:output doctype-public=
   "-//OASIS//DTD DocBook XML V4.1.2//EN"/>
  <xsl:output doctype-system="docbookx.dtd"/>
  <xsl:strip-space elements="*"/>
 <xsl:template match=
  "xsl:template|xsl:stylesheet|xsl:text">
   <xsl:apply-templates/>
 </xsl:template>
 <xsl:template match="xsl:output"></xsl:template>
 <xsl:template match="xsl:value-of">
   <xsl:text>[</xsl:text>
   <xsl:call-template name="mapToBetterName">
     <xsl:with-param name="nameIn" select="./@select"/>
   </xsl:call-template>
   <xsl:text>]</xsl:text>
 </xsl:template>
 <xsl:template match="xsl:if">
   <xsl:text>[IF (</xsl:text>
   <xsl:call-template name="mapToBetterName">
     <xsl:with-param name="nameIn" select="./@test"/>
   </xsl:call-template>)
   <xsl:apply-templates/>
   <xsl:text>]</xsl:text>
 </xsl:template>
 <xsl:template match="*">
   <xsl:copy>
     <xsl:copy-of select="@*"/>
     <xsl:apply-templates/>
   </xsl:copy>
 </xsl:template>
 <xsl:include href="mapnames.xsl"/>
</xsl:transform>
```

LISTING 6 XSL variable mapping

```
<?xml version="1.0" encoding="utf-8"?>
<namemaps>
  <namemap from="customer/age">customer's age</namemap>
  <namemap from="customer/netWorth">customer's net worth
   </namemap>
  <!-- other mappings -->
</customer>
```

LISTING 7 Generated documentation in DocBook

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE book
  PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
   "docbookx.dtd">
<book xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <title>Financial Plan for [customer's name]</title>
  <chapter>
    <title>Introduction</title>
    <section>
      <title>Overview</title>
      <para>
        After presenting educational planning
         recommendations for your
        children's college needs, we will review ways to
         reach your
        retirement objectives[IF (customer's age &gt; 50 and
        customer's net worth &gt; 500000) and suggest
         estate planning
        techniques for minimizing tax exposure].
      </para>
    </section>
  </chapter>
</book>
```

Download the Code
www.sys-con.com/xml

FREE TUTORIAL
JUNE 27TH WITH WEB SERVICES EDGE
2002 EAST REGISTRATION. LIMITED OFFER!

TO REGISTER: www.sys-con.com or Call 201 802-3069

web services EDGE world tour 2002

$195
REGISTRATION FOR SYS-CON SUBSCRIBERS
BEST EDUCATIONAL VALUE FOR THE HOTTEST TECHNOLOGY SKILLS!

Take Your Career to the Next Level!

SEATING IS LIMITED. REGISTER NOW FOR THE CITY NEAREST YOU! WWW.SYS-CON.COM

Learn How to Create, Test and Deploy Enterprise-Class Web Services Applications

TAUGHT BY THE INNOVATORS AND THOUGHT LEADERS IN WEB SERVICES

EXPERT PRACTITIONERS TAKING AN APPLIED APPROACH WILL PRESENT TOPICS INCLUDING BASIC TECHNOLOGIES SUCH AS SOAP, WSDL, UDDI AND XML, PLUS MORE ADVANCED ISSUES SUCH AS SECURITY, EXPOSING LEGACY SYSTEMS AND REMOTE REFERENCES.

RECEIVE 4 FREE $345 VALUE!
1-YEAR SUBSCRIPTIONS

REGISTRATION FOR EACH CITY CLOSES THREE BUSINESS DAYS BEFORE EACH TUTORIAL DATE. DON'T DELAY. SEATING IS LIMITED.

NON-SUBSCRIBERS: REGISTER FOR $245 AND RECEIVE THREE FREE ONE-YEAR SUBSCRIPTIONS TO *WEB SERVICES JOURNAL, JAVA DEVELOPER'S JOURNAL*, AND *XML-JOURNAL*, PLUS YOUR CHOICE OF *BEA WEBLOGIC DEVELOPER'S JOURNAL* OR *WEBSPHERE DEVELOPER'S JOURNAL*, A $345 VALUE!

SYS-CON *i*-TECHNOLOGY EDUCATION

SHARPEN YOUR PROFESSIONAL SKILLS.

KEEP UP WITH THE TECHNOLOGY EVOLUTION!

"Presented an excellent overview of Web services. Great inside knowledge of both the new and old protocols. Great insight into the code piece."

– Rodrigo Frontecilla

"Very articulate on the Web services SOAP topic and well-prepared for many questions. I've learned a lot from this seminar and I appreciate this seminar for my job. Thank you!"

– Kenneth Unpingco, Southern Wine & Spirits of America

"I liked the overview of Web services and the use of specific tools to display ways to distribute Web services. Good for getting up to speed on the concepts."

– B. Ashton, Stopjetlag.com

Echoed over and over by Web Services Edge World Tour Attendees:

"Good balance of theory and demonstration."

"Excellent scope and depth for my background at this time. Use of examples was good."

"It was tailored toward my needs as a novice to SOAP Web services – and they explained everything."

WHO SHOULD ATTEND:

- **Architects**
- **Developers**
- **Programmers**
- **IS/IT Managers**
- **C-Level Executives**
- ***i*-Technology Professionals**

IF YOU MISSED THESE...

BOSTON, MA (Boston Marriott Newton) SOLD OUT!
WASHINGTON, DC (Tysons Corner Marriott) SOLD OUT!
NEW YORK, NY (Doubletree Guest Suites) SOLD OUT!
SAN FRANCISCO, CA (Marriott San Francisco) SOLD OUT! CLASSES ADDED

BE SURE NOT TO MISS THESE...

Each city will be sponsored by a leading Web services company

...COMING TO A CITY NEAR YOU

2002

NEW YORK AT WEBSERVICES EDGE 2002 EAST	JUNE 27
BOSTON	JULY 10
SAN FRANCISCO	AUGUST 6
SEATTLE	AUGUST 27
AUSTIN	SEPTEMBER 10
LOS ANGELES	SEPTEMBER 19
SAN JOSE AT WEBSERVICES EDGE 2002 WEST	OCTOBER 8
CHICAGO	OCTOBER 17
ATLANTA	OCTOBER 29
MINNEAPOLIS	NOVEMBER 7
NEW YORK	NOVEMBER 18
SAN FRANCISCO	DECEMBER 3

2003

CHARLOTTE	JANUARY 7
MIAMI	JANUARY 14
DALLAS	FEBRUARY 4
BALTIMORE	FEBRUARY 20
BOSTON	MARCH 11

REGISTER WITH A COLLEAGUE AND SAVE 15% OFF THE LOWEST REGISTRATION FEE.

TOPICS HAVE INCLUDED:
Developing SOAP Web Services
Architecting J2EE Web Services

On May 13th, the San Francisco tutorial drew a record 601 registrations.

TO REGISTER: **www.sys-con.com or Call 201 802-3069**

WRITTEN BY **JOSEPH MOELLER**

Patent Mining Using XML

How one company developed a tool to encourage innovation and collaboration

We developed a "patent mining" tool for our company utilizing XML. Corporate leaders wanted to provide company-wide browser access to our corporate patent portfolio as well as in-depth analysis of our patent data. They saw this as a way to assist our company in strategic and investment decisions and to encourage innovation and collaboration.

The highly structured format of patent data led easily to a hierarchical representation in XML. The use of XML made programming simpler. We only had to load the XML document containing the patents into a browser, so there was no need to extract from a database. In addition, we were able to present the same XML data in multiple ways, using various XSL stylesheets. XML provided a very straightforward, inexpensive, and effective way to meet this requirement for patent mining.

Packaged software is available that imposes structure on a company's content, making it easily searchable and capable of being displayed in various ways in a browser. For this application, however, the volume is relatively low; our company currently has only a few hundred patents totaling about 10MB of data. The content is fairly static; the issued patent data rarely changes, and relatively few new patents are added each month. And the number of concurrent users will remain low.

Because of these factors, we decided we could safely avoid purchasing any content management, database, or other software.

Implementation Steps

The U.S. Patent and Trademark Office (USPTO) provides downloadable access to all patents that have been issued. We wrote a script to access the USPTO Web site, search for our company's patents, and then download them. Once we had them all together, we identified the consistent structure throughout. Listing 1 shows the XDR Schema we created to model the XML markup (all code references in this article can be downloaded at www.sys-con.com/xml/source.cfm).

The header section of a patent contains field data such as patent number, title, issue date, and assignee. The header data also includes repeating groups such as inventors, references cited, and classes. The detailed body of the patent, which includes the "claim" of the patent and a lengthy description, consists largely of unstructured text.

After downloading the patents, analyzing their structure, and creating a schema from them, we were ready to mark them up with XML. To expedite the markup process we wrote a Perl script to tag the structured fields with XML. This was fairly easy to parse with Perl due to the rigid structure of the data.

Figure 1 shows the first page of a patent as it appears on the USPTO site. Listing 2 is the same patent data after being downloaded and marked up with XML. Figure 2 shows the same patent again as displayed in our tool using XSL; it's deliberately similar to the USPTO presentation because we intended to duplicate that look and feel.

The home page of our application lists the critical information for all current patents: patent number, issue date, title, inventor. From this a user is able to click on a patent number and see the detail of an individual patent, such as the one shown in Listing 2. We also created a page that provides field searches and a keyword search of all corporate patents. We used XPath queries to perform the field searches; for keyword search we used regular expression pattern matching. A citation analysis diagram displays which patents reference a patent and which are referenced by a patent. A class distribution histogram groups the patents according to the USPTO classification scheme.

Mining Unstructured Data

Patent mining starts with searchable access to the patent data, but more in-depth analysis requires mining of the unstructured claims and description data in the body of the patent. We ran various cluster analysis programs against the unstructured data and converted the output of these programs into XML. We then displayed the data in various ways using XSL. For example:

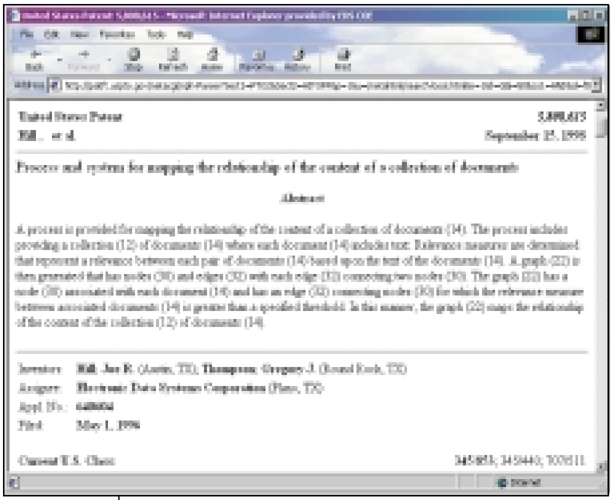
- ***Relevance matrix diagram:*** Uses an algorithm that analyzes overlap in word content and clusters the content accordingly
- ***Optimized cluster analysis diagram:*** Clusters the patents into 20 categories based on word usage
- ***Molecule diagram:*** Shows the clustering in molecules connected to each other based on relevance

Design Decisions

We originally planned to make the patent mining tool a client-side application, which seemed like the simplest and most straightforward route. A client-side application involves downloading the XML data onto each desktop browser, parsing the XML, caching the data in memory, and then transforming it by XSL, all on the client. Some functions, such as keyword search, require loading all 10MB of the data onto the desktop, which became a performance bottleneck. (After the first time, however, the data was cached.)

An additional client-side drawback was that to make effective use of XSLT,

AUTHOR BIO
Joseph Moeller is a consultant specialist in the Enterprise Information Technology division of EDS. Joseph, who has been with EDS for 20 years, currently works in EDS's Austin Innovation Centre, which has made XML an important part of their strategic direction.



United States Patent 5,808,615

Hill , et al. September 15, 1998

Process and system for mapping the relationship of the content of a collection of documents

Abstract

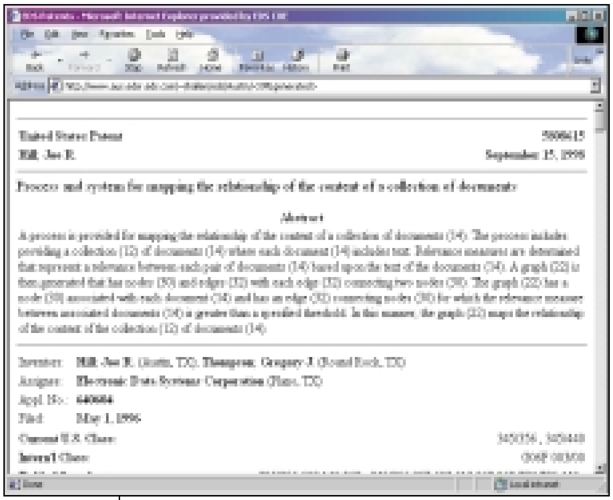
A process is provided for mapping the relationship of the content of a collection of documents (14). The process includes providing a collection (12) of documents (14) where each document (14) includes text. Relevance measures are determined that represent a relevance between each pair of documents (14) based upon the text of the documents (14). A graph (22) is then generated that has nodes (30) and edges (32) with each edge (32) connecting two nodes (30). The graph (22) has a node (30) associated with each document (14) and has an edge (32) connecting nodes (30) for which the relevance measure between associated documents (14) is greater than a specified threshold. In this manner, the graph (22) maps the relationship of the content of the collection (12) of documents (14).

Inventors: **Hill; Joe R.** (Austin, TX); **Thompson; Gregory J.** (Round Rock, TX)
Assignee: **Electronic Data Systems Corporation** (Plano, TX)
Appl. No.: **640604**
Filed: May 1, 1996

Current U.S. Class: 345/853; 345/440; 707/511

File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Favorites History Print
Address
Internet

FIGURE 1 | Patent as it appears at USPTO



United States Patent 5808615

Hill; Joe R. September 15, 1998

Process and system for mapping the relationship of the content of a collection of documents

Abstract

A process is provided for mapping the relationship of the content of a collection of documents (14). The process includes providing a collection (12) of documents (14) where each document (14) includes text. Relevance measures are determined that represent a relevance between each pair of documents (14) based upon the text of the documents (14). A graph (22) is then generated that has nodes (30) and edges (32) with each edge (32) connecting two nodes (30). The graph (22) has a node (30) associated with each document (14) and has an edge (32) connecting nodes (30) for which the relevance measure between associated documents (14) is greater than a specified threshold. In this manner, the graph (22) maps the relationship of the content of the collection (12) of documents (14).

Inventors: **Hill; Joe R.** (Austin, TX); **Thompson; Gregory J.** (Round Rock, TX)
Assignee: **Electronic Data Systems Corporation** (Plano, TX)
Appl. No.: **640604**
Filed: May 1, 1996
Current U.S. Class: 345/356 ; 345/440
Intern'l Class: G06F 003/00

File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Favorites History Print
Address
Done
Local intranet

FIGURE 2 | Patent as it appears in patent mining tool

each client had to have XSLT installed. Yet many Microsoft Internet Explorer users don't yet have MSXML parser 3.0 or above, which is required for XSLT. So we had to backtrack and use the more limited functionality of XSL (a standard component of IE 4.0 and above) on the client desktop.

We'll soon be converting our application to server-side processing, wherein all the processing takes place on the server and only the resultant HTML is sent to the client. Then we won't have a bandwidth/download problem, since the 10MB of data remains on the server. As there's also a persistent DOM (loaded once into cache) ready for processing on the server, performance and scalability are increased. And the processing is no longer browser-dependent. We don't need to be concerned about whether the client Internet Explorer can handle XSLT, or whether the client browser may even be Netscape, which has limited XML support.

• • •

For more complete patent mining functionality, including more analytical tools, we may still buy packaged software. If we decide to expand the scope of patents to include all similar patents issued by the USPTO, which could number in the tens of thousands, we'll definitely need a database. In the short term, however, we have a useful application founded on XML that our company can use to perform basic patent mining.

JOE.MOELLER@EDS.COM

Now in More than 5,000 bookstores worldwide
subscribe Now!
FOR FAST DELIVERY
Go Online and Subscribe Today!
The World's Leading Independent WebLogic Developer Resource
FOR WLS DEVELOPERS BY WLS DEVELOPERS
BEA WebLogic DEVELOPER'S JOURNAL
An introduction to... WebLogic
SPECIAL INTRODUCTORY OFFER
SAVE $31*
HURRY, DON'T DELAY! OFFER EXPIRES AUGUST 31, 2002
Helping you enable inter-company collaboration on a global scale
• Product Reviews
• Case Studies
• Tips, Tricks and more!

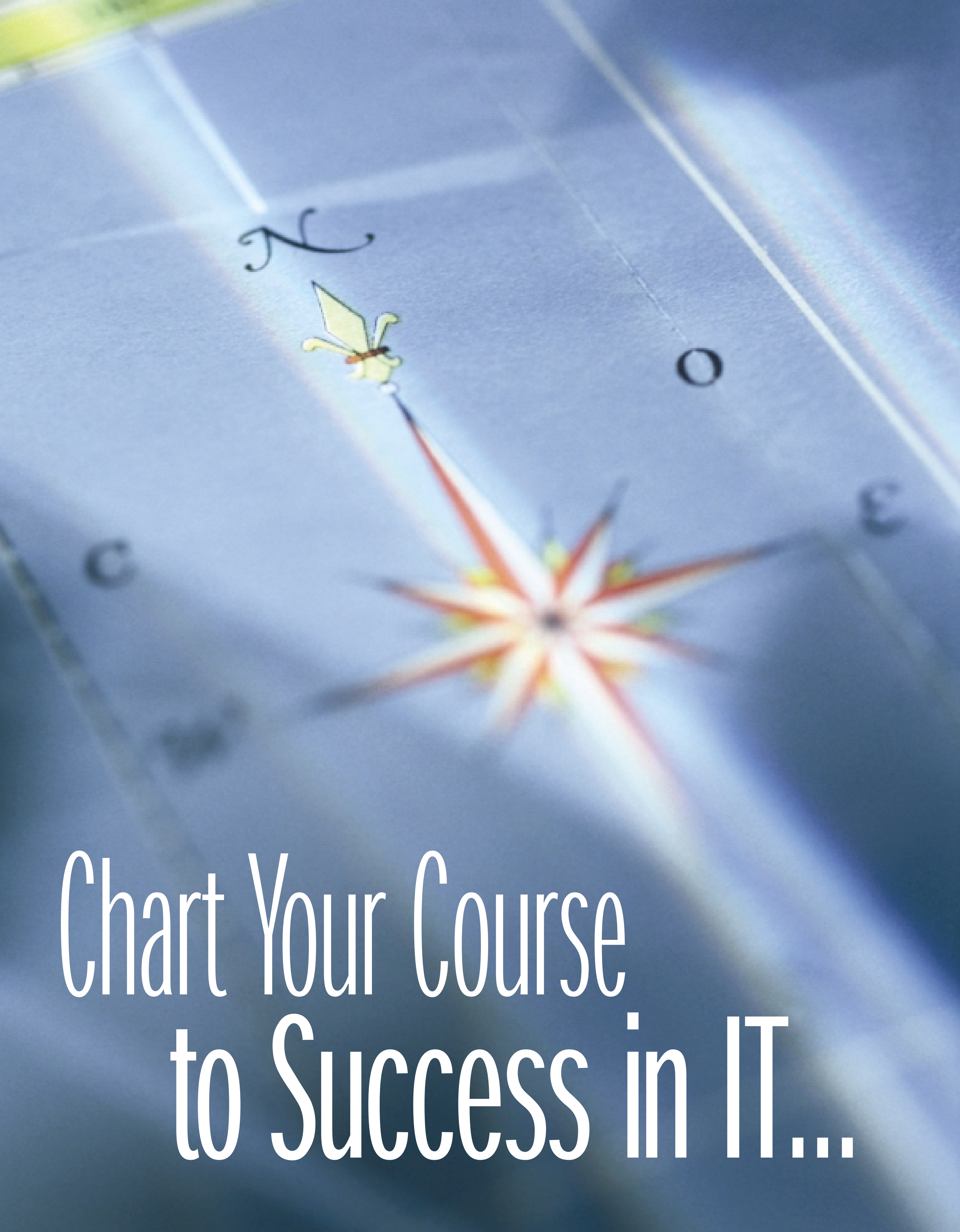


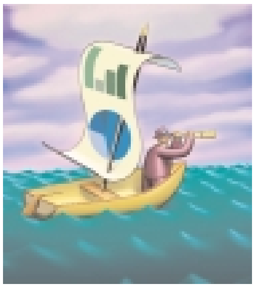
Chart Your Course to Success in IT...

...order your copy of

Java Trends: 2003

Available July 1*

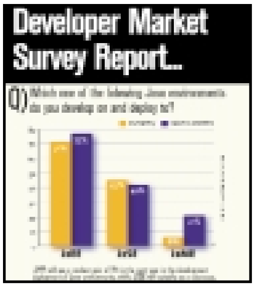
Don't go astray. In the vast sea of Internet technology, market conditions change constantly. Will Java remain the hot platform it is today? Will C# rapidly gain ground? What are the world's foremost Java developers aiming their sights toward? Which companies come to mind when planning their next project? How can their thinking direct your efforts?



EnginData Research has studied the IT industry extensively, spotting many key trends and alerting industry leaders along the way. In the first quarter of 2002, they embarked on their most challenging mission ever: the most in-depth study ever done of the Java development marketplace.

After months of analyzing and cross-referencing more than 10,500 comprehensive questionnaires, the results are now available.

EnginData's research is invaluable to leading IT vendors, integrators, Fortune 500 companies, trade-show organizers, publishers, and e-business organizations worldwide.


Developer Market Survey Report...
Q) Which one of the following Java environments do you develop on and deploy to?

Here's just a sample of the critical data points the Java report delivers...

- ✔ Current and projected usage of languages and platforms
- ✔ Multiple rankings of hundreds of software vendors and tools
- ✔ Types of applications being developed
- ✔ Databases being used
- ✔ Purchasing patterns
- ✔ Company size
- ✔ Development and purchasing timelines
- ✔ Perceptions and usage of Web services
- ✔ Preferred Java IDE
- ✔ J2EE, J2ME, J2SE usage comparisons

As an IT specialist, **EnginData Research** focuses exclusively on three leading drivers in IT today – Java, Web services, and wireless development. Coming soon, our Web services survey will deliver such critical knowledge as:

- ✔ Time frame for developing Web services – enabled apps
- ✔ Percentage of apps with Web services today
- ✔ Sourcing and hosting Web services
- ✔ Perceived leaders in Web services tools and solutions

Navigate your company through the challenging IT marketplace with the trusted knowledge and intelligence of **EnginData Research**. Order your copy of the 2002–2003 Java market study by calling Margie Downs at 201-802-3082, or visiting our Web site.

engindata RESEARCH

www.engindata.com

*A limited number of preview copies will be available at our booth (#624) at JDJEdge International Java Developer's Conference & Expo, June 24–27, at the Jacob Javits Convention Center in New York.

WRITTEN BY **GREG MABLE**

The Next Generation Database – XDB

Combining XML's document flexibility with SQL's data integrity

Imagine the Internet as an ocean and yourself on the beach gazing out on the horizon. If you look far enough, you'll see the crest of a big wave forming. You can see that it's beginning to take shape and that its impact will be major. It will affect all that lies in its path. What is this next big wave? You've probably already guessed: XDB.

AUTHOR BIO

Greg Mable is CTO of NuSoft Solutions, Inc. (www.nusoftsolutions.com, a consulting services company in southeastern Michigan. Coauthor of SQL Server 7.0 Unleashed (SAMS), Greg is a Microsoft certified systems engineer/solutions developer with over 15 years of experience in information technology. He has extensive experience developing a wide range of Web and client/server applications using XML, Java, Microsoft SQL Server, Visual Basic, and Internet-related technologies.

Application processing will change dramatically with the ability to exchange information transparently. With the advent of Web services, applications are now free to communicate in a common format – that of an XML document – anywhere on the Web. Where the Web was once built on static content linked together via hypertext, XML takes it to the next level. Instead of users surfing the Internet via HTML pages linked with hyperlinks, we can now build Web-based applications that can be linked via XML documents. Imagine a user clicking on a link to a Web site. This in turn fires off an exchange of an XML document to another application.

Here's the key: the XML document. This will be the primary means of information exchange and message passing. With the need to process XML documents comes the need to be able to store, retrieve, and report on them. Hence the need for a management system to handle the flood of XML documents that an application will process. This is where an XML database, XDB, comes in.

What Is an XDB?

So what is an XML database, or an XDB? In this article I define what it is, when and why you will need to use one, and what impact it will have on the business world. By the time you finish reading, you just may realize the importance of an XDB and will want to grab your surfboard to ride the next big wave.

Before I define what an XDB is, let's briefly review the evolution of databases. We can consider the first generation of databases as file-based systems. VSAM, ISAM, Btrieve, and dBase – examples of file-based databases – store data in a file and use various types of index methods to retrieve and update it. The second generation of databases is relational database management systems, or RDBMSs. Oracle, SQL Server, and Sybase are examples. They use a common language, SQL, to access the data stored in normalized tables composed of rows and columns. Now, with the emergence of XML, the need to quickly store, retrieve, and update XML documents in real time is imperative. The next generation of databases, XDBs, will need to excel at providing this capability.

We can define an XDB as a database management system that stores, updates, and retrieves XML documents. It uses as its fundamental unit of storage an XML document. Just as the first generation uses a file (or table) as its base unit, and the second generation, RDBMSs, uses a row in a table as its base unit of storage, in an XDB all operations function at the XML document level.

In addition to using the XML document as its base unit, an XDB provides a set of tools to manage the XML documents. These tools conform to a logical data model that abstracts the logical structures from the physical structures. Four major data model specifications are currently proposed by the W3C:

1. Document Object Model (DOM) 1.0 Level 3
2. XML Information Set model
3. XPath 1.0 data model
4. XQuery 1.0 and XPath 2.0 data model

Types of XDBs

There are no requirements for how an XDB is expected to physically store XML documents. Some XDBs are built on an object database, others might use compressed files with an indexing scheme, and still others might be built on top of a relational database. At this time XDBs can be classified into two basic types (with a third type on the horizon): native and XML enabled.

Native XML database

A native XML database (NXDB) is simply one that was designed from the ground up to store XML documents. It might make use of a preexisting technology such as object-oriented data storage techniques, but its mission is to store, retrieve, and update XML documents.

XML-enabled database

In the second type, an XML-enabled database (XEDB), extensions are added to a preexisting database management system to support XML documents. An XEDB can be built on top of an existing object-oriented or relational database management system. An XEDB provides a mapping layer between the XML documents and its database structures as well as support for XML-based tools to retrieve and update XML documents.

Convergence of NXDB and XEDB

The third type of XDB is in its formative stages, and like a wave approaching the beach, it is about to crest. It can be considered a convergence of the two other types: an XDB that is designed to handle XML documents but is built on a preexisting database technology, combining them into a unified data model and a single repository.

In this article I'll briefly describe an example for each of these types. Xindice (pronounced *Zeen-dee-chay*) is an open source Apache project designed as a native XDB. SQLXML 3.0 is a set of XML extensions for use with Microsoft's SQL Server to further support XML processing and Web services. And with release 2 of 9*i*, Oracle announces the XML DB, unifying XML documents with SQL data into a single repository.

Why Use an XDB?

You might ask why you should use an XDB. Couldn't you simply store an XML document in an RDBMS as a BLOB? First, you can most certainly use an RDBMS to store XML documents. In certain applications it may even be more appropriate than an XBD. However, an important distinction needs to be made with regard to how your application makes use of an XML document. The terms *data-centric* and *document-centric* are used to describe the primary function that an XML document provides for an application.

Data-centric

An XML document that's used as the primary vehicle to transport data from one application to another, whether it's a B2B application or simply between the browser and the database, is likely to be data-centric. A data-centric XML document is characterized by its use of highly structured content, fine-grained data, and its focus on transferring data to another application for processing. It's one of the primary strengths of XML, and the one making the biggest impact on the business world today.

Some examples of data-centric XML documents might be a purchase order, stock quote, driver's license, or address book. These documents mostly contain structured content and a minimal amount of descriptive text or mixed content. In addition, the intended recipient of the XML document is another application, such as an Internet browser, a billing system, or a credit card application.

Document-centric

Document-centric XML documents are exactly the opposite: they contain a mixture of structured and semistructured content, and are intended for a person to read rather than for an application to process. Some examples of document-centric XML documents would be books, articles, reports, and medical histories. Document-centric XML documents are characterized by a more informal structure, large-grained data, and a majority of descriptive text or mixed content.

It may not always be easy to make the distinction between data-centric and document-centric XML documents. In fact, you might find that most of your documents contain an equal mix of structured elements and semistructured content, and that the intended recipient is an external third party, which can either be a user surfing your Web site or a Web service responding to a SOAP message.

For example, you might be building a system for a college to use to store, manage, and evaluate student admission applications. Along with the student's high school grades, extracurricular activities, and so on, the admission application contains several personal essays, such as one explaining why he or she would like to attend the school. This document is composed of both data-centric elements (student grades) and semistructured content (personal essays). It's intended for use by college admissions counselors as well as by application programs, such as a financial aid program that evaluates applications based on a combination of financial need and scholastic aptitude.

Even though at times it might be difficult to determine, you'll still need to characterize your application's use of an XML document. If your application uses an XML document primarily as a means of data exchange, then an XML-enabled database would be more appropriate. For example, you're building an application to transmit purchase orders. The XML document merely serves as a way to format the data. In this case your application uses a data-centric model of an XML document. But if your application uses an XML document as a means to capture content, especially human-readable documents, then an XDB would be more suitable. In another example, you're building an application to manage legal contracts. An XDB would be better for this since it can manage its semistructured content and, whenever you need to access a particular contract, it fundamentally preserves the document in its original form. This concept is known as *round-tripping*. For further discussion please refer to the sidebar.

Here are some key advantages of using an XDB to manage your XML documents:

- ***Speed:*** An XDB can retrieve XML documents and its elements much faster than an RDBMS or XML-enabled database since it stores the XML document natively. It doesn't need to reassemble the document or perform any complex joins, as an XML-enabled database would.
- ***XML tools support:*** An RDBMS doesn't have any built-in XML tools support or XML tools that are ancillary to the product, as an XML-based database does.
- ***Handling of semistructured content:*** An XDB is flexible in its handling of semistructured content and its capacity to manage change.
- ***Schema independence:*** An XDB doesn't require any schema to store an XML document, whereas an XML-enabled database does.

In contrast, here are some of the shortcomings using an RDBMS or XML-enabled RDB to support XML documents:

- The mapping between an XML document and the database structures can be quite complex. Maintenance of this mapping can be difficult as the system evolves over time.
- The mapping between an XML document and the database will inevitably lose some of its metadata.
- XML documents can't be round-tripped; the XML documents you put into an RDBMS or an XML-enabled database aren't the same as the documents you get out.

XML DB Architecture

In an XDB, XML documents can be grouped logically into what is termed a *collection*. Just as data in an RDBMS is stored into a set of tables, the architecture of an XDB is built on the notion of collections. XDBs allow you to manage a given group of XML documents in a collection, providing you with the ability to query and update the documents as a whole.

Along with the ability to build and manage XML documents in a collection, an XDB provides you with the ability to create indexes on portions of the XML document. This ensures that key portions of your document are available for fast searching. In addition, since an XDB stores an XML document in its native format, queries that need to return a set of XML documents will also be fast.

Finally, an XDB incorporates the creation of DTDs, XML schemas, or both. This allows you to validate an XML document against a given schema and store that schema with the database.

Let's look at an example of each of the three types of XDBs. In all cases these products are in their infancy and their support for the growing base of XML standards varies significantly from vendor to vendor. In addition, support for other, database-related functions, such as transactional processing, referential integrity, and clustering, is weak. All of the products face a major challenge in both the marketplace and the developer community in that the concepts, techniques, and architectures are new to the industry. With anything new comes a tendency to resist change. And when you're working in the world of DBAs and their focus on data normalization, it's nearly impossible to introduce the concept of a denormalized XML document as a part of their environment.

Xindice

Xindice uses collections to store a group of documents. While these documents can be of the same document type, Xindice also supports well-formed documents that don't have an associated schema.

Xindice supports XPath to search through the XML documents in a collection. Updates are supported through its XUpdate initiative, which allows you to change a portion of the document without having to retrieve the entire document. Xindice also allows you to index documents on both element and attribute values.

The weaknesses of the product include the lack of any GUI-driven user interface. All of the tools operate at the command-line level. An excellent addition to the product would be a set of XML tools, such as a built-in XML editor, a DTD or XML Schema editor, support for XSLT, and so on, but at its base the product is simple to use, has a small footprint, and is very reliable.

Microsoft SQLXML

Microsoft's set of XML services is bundled into an add-on product to their base relational product, SQL Server 2000. The set is referred to as SQLXML Web Services. Basically, these services provide a layer of XML over its relational database engine. XML documents can be stored in the database via a mapping layer, and extensions to its SQL language support queries against the data to return XML documents. Updates are also supported against the XML documents using an Updategram (which is built on the mapping of an XML document to the database structures). Access to the XML documents can be made via multiple methods, including HTTP, ADO, and, with release 3.0, SOAP.

Microsoft's approach to handling XML documents is through a strategy of XML enabling their base relational product. All of the weaknesses described above are immediately evident when using SQLXML. The mapping layer can sometimes be quite complex and can be a nightmare to support for future changes. Its approach lacks a means to make XML document processing transparent to the end user, and its support of XML tools is minor.

*Oracle 9*i *release 2*

Count on the largest database vendor to come up with a vision for supporting XML documents in an XDB. Oracle's latest release brings with it the concept of single repository, uniting document content with the relational database. It has the largest set of XML tools of any product, including an XML editor and XML schema processor. It supports a wide range of features, including XPath search, which can be combined with SQL, ACLs, folders, and global and local schemas. It offers two alternative means of storing XML documents: one in which the XML document is stored as a BLOB within the database, the other in which the XML document is stored within an object-relational storage structure. This gives the the developer the freedom to choose which method is best to store the XML documents.

With this release Oracle has further extended the range of features that its flagship product provides, and has transformed itself into a database that can be used to build the next set of applications.

Why the Future Is the Document (and Not Just the Data)

As the saying goes, It's the document, stupid. No longer is data the fundamental unit of computation; rather, the new paradigm is the XML document. Applications are slowly but steadily moving away from a data-centric point of view to a document-centric–oriented framework.

If you look at the growth of relational database systems, they're reaching a limit on their penetration in the marketplace. The number of companies moving to such a system from a legacy database system grows smaller each year. Recent growth for relational database systems has been in the area of new features, such as online analytical processing (OLAP), data mining, clustering, and caching, to name a few. What were once niche market products like data warehousing are being subsumed into the relational database systems. Witness Microsoft and their incorporation of OLAP services as an enhancement to their base SQL Server product.

The latest area of growth for RDBMSs has been in capitalizing on the wave of XML and Web services. Witness Microsoft's release of SQLXML, a set of features to support XML and Web services. Witness Oracle and the 9*i* product. The strategy of the major database vendors is to better support XML natively in their products. Witness IBM and their support for XML in DB2 with the release of their middleware product, Esperanto. Again, the growth of these products has been to extend the services provided by the base-relational database engine. This time it's in the realm of XML support.

The next-generation databases may be XDBs, and RDBMSs may not be what we think they are now. Perhaps we'll turn to Oracle, DB2, and SQL Server not only as a relational database management system, but we also might look to them as a native repository for XML documents, with all the necessary supporting features built in.

So what features are necessary to support this new requirement of XML document processing? Whether next-generation databases are a whole new set of applications or native XML databases, whether they're hybrids or relational databases with XML support, these are the features any XML-based system will need to work effectively:

- ***XML editor:*** This feature should be built into the product and include support for WYSIWYG XSLT processing. The tool would enable a developer to take raw XML input, build a stylesheet, and output a formatted HTML document.
- ***DTD and XML schema editor:*** This tool should be akin to an ER tool to allow a developer to build XML schemas and validate documents against the schema.
- ***A robust query language:*** Along with the ability to query across a set of XML documents in a collection, this tool would also provide the ability to update documents and portions of documents (without having to update the entire document). Ideally, the query language would be similar in style to SQL so that developers would not have YAQL (yet another query language) to learn. And this language would be able to query against a set of relational database tables, allowing developers to write one query that can pull together data from both data stores.
- ***Repository for XML document collections:*** With this a database developer could create a database in an XDB and populate it with any number of document collections. For example, in the case of an XDB built for the IRS, the repository could be a set of personal tax forms. Each collection in the repository would represent a particular tax form, such as 1040s or Schedule A's.
- ***Browser feature:*** This would allow users to browse the set of documents residing in a collection, much like Windows Explorer allows a user to browse through a set of files in a folder.
- ***Index feature:*** This would allow developers to create and maintain indexes on a set of XML documents.
- ***ACLs:*** This would provide security at the collection and XML document levels, and even for different elements within an XML document.
- ***Transaction support:*** This would provide support for atomic updates on an

e-BUSINESS SOLUTIONS: TAKING BUSINESS TO THE NET
WebSphere
DEVELOPER'S JOURNAL
Development with VisualAge for Java

COLDFUSION
XML-JOURNAL
PowerBuilder Journal
JAVA
wireless

JAVA
wireless
XML-JOURNAL
XML
MESSAGING WITH JAXM

HOME
Enterprise Solutions
Content Management
Data Management
XML Labs

XML document or any portion thereof, and include rollback support.

- ***Clustering:*** This would be similar to database clustering, providing failover support by growing a set of XDBs into a logical unit.
- ***Text searching:*** This would provide all of the advanced searching features we're accustomed to using, such as proximity matching and stem support.

What we really need to be looking at is what XML and Web services open up for application developers. What new line of applications are we enabled to build? Who hasn't read or been inundated for the past year with articles on XML and Web services? If you read any of them, they all talk about the details of building applications that use XML as a form of data exchange and message processing.

What they don't discuss in any great detail is what new applications this technology enables us to build that we couldn't build before. Applications exchanging data via text-based messaging is nothing new. We've been doing that for almost 20 years now using good old EDI technology. Applications exchanging data via messages is nothing new either. SMTP has been around for nearly 20 years as well.

What's new and different about XML and Web services is the ubiquity of the data. Data is now self-describing, and applications now have a universal way of exchanging it. This frees us to think about building new applications we could never seriously attempt to build before, especially applications that weren't well defined, or had well-defined data structures. With XML and Web services we're free to focus on the application at hand. Instead of translating an application's information into data that has to fit nicely into rows and columns of a relational database, we can focus on the documents that an application processes and begin to think in terms that more closely align with the real world.

So what are these new opportunities that XML and Web services make available? Think about any aspect of the real world. Chances are there's an application already built to handle its basic needs for storage and retrieval. For example, an auto insurance company most likely has an application in place that allows an investigator to input an accident report. It will capture the date/time of the accident, the parties involved, the vehicles involved, and so on. It might also include a description of the accident in free-form text. What value can XML document processing and an XDB add to such an application that isn't already being handled adequately?

If we free ourselves from the constraint of data-centric application processing and learn to think about an application in terms of what value the user gets out of using the application, we begin to truly enhance an existing application and take it to the next level. In this instance, what value does the insurance adjuster get out of using the application? In other words, what document does he produce as a result of using the system?

In this case it might be an appraisal document that the insurance adjuster generates as a result of his analysis of the data. The document would be his assessment of why the claim is either denied or approved. This assessment would be an XML document generated not only from the data collected by the system, but also by the evaluation of the facts made by the adjuster. Instead of just using the application as a means to electronically capture the information about the accident (the drivers and vehicles involved), an XDB could capture in a central repository the other semistructured information not normally captured by a claims application – eyewitness statements, a police report, the driver's medical history. The insurance adjuster could then use the application to record the facts about the accident as well as his assessment, and would be able to justify his decision based on information captured in the XDB.

• • •

In summary, when building new applications or enhancing existing ones with XML document processing, keep in mind the purpose of the XML document. Is it data-centric or document-centric? But don't limit yourself (and your users) to using XML documents solely as a means to exchange information. As you embark on a new project, ask yourself, What do the users of the system ultimately produce, that is, what document do they generate as a result of using the system? And bring some wax for your surfboard – the waves are going to be big.

Additional Resources

- *Apache Xindice Web site:* http://xml.apache.org/xindice/
- *Microsoft's SQL Server XML Web resource page:* www.microsoft.com/sql/techinfo/xml/default.asp
- *Oracle's XML DB Web site:* www.oracle.com/ip/index.html?xmldb_intro.html
- *Ronald Bourret's Web site on XML database and products:* www.rpbourret.com/
- *Robin Cover's Web site with links to papers and products:* http://xml.coverpages.org/xmlAndDatabases.html

GMABLE@NUSOFTSOLUTIONS.COM

Round-Tripping

An important factor to consider when deciding what type of XDB to use to store XML documents is its ability to preserve documents as originally generated. This would be important to an application in which the XML documents are considered legal documents or transactions of some sort, such as an insurance claim. In these cases you'll want the ability to retrieve the original XML document as it was transmitted and not have it altered in any way by the database. This concept is known as round-tripping. As shown in Figure 1, a native XDB can receive an XML document, store it, then return it in the same form as it was originally transmitted. This includes preserving all of the white space, processing instructions, and comments. Nothing about the document has been changed as a result of its being stored in the system.

In a document-centric XML application, these parts of the XML document are often just as important as the elements, attributes, and entities. In some cases the application might be required by law to retain exact copies of the documents for a specified amount of time.

On the other hand, an XML-enabled RDBMS takes an incoming XML document, "shreds" it (or breaks it apart and stores the pieces in the relational or object database structures), and then reassembles the document whenever it needs to retrieve it (see Figure 2). Besides the extra processing time it takes to retrieve the document, any white space, processing instructions, or CDATA sections won't be preserved. If your application is primarily data-centric, round-tripping most likely isn't an important factor to consider. An XML-enabled RDBMS is proficient at transferring data from an XML document to the database and back to an XML document without any loss of document content. However, you should keep in mind that not only will an XML-enabled RDBMS not be able to preserve any white space, processing instructions, or comments, but it also won't be able to maintain the same order of child elements. In comparing the original XML document with the reassembled one, your application users might be perplexed by how different the two documents appear, yet still contain the same document content.

The physical representation of an XML document is referred to as its canonical form. The Canonical XML 1.0 W3C recommendation defines a method to generate the canonical form of an XML document. If two documents have the same canonical form, they're considered logically equivalent. In determining which type of XDB works best for your application, you'll need to decide the level of detail that must be preserved when round-tripping XML documents and whether you'll need to compare XML documents for equivalence (see Figure 3).

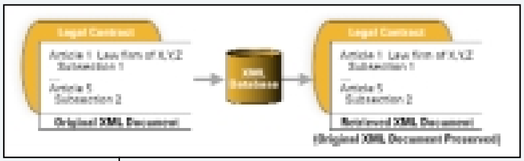


FIGURE 1 | Round-tripping: XML database

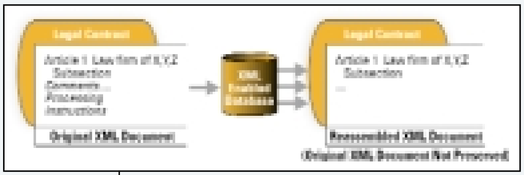


FIGURE 2 | Round-tripping: XML-enabled database

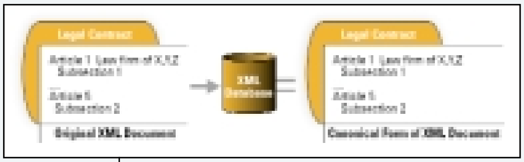


FIGURE 3 | XML document equivalency

WRITTEN BY **CRAIG KING**

Managing Data Sources with XSLT

An approach to transform any type of data source to XML

I and a colleague were working on a research project when we saw an opportunity to approach our data management from a different angle. XML appeared on the scene, and when IBM alphaWorks released its first parser we were on our way, using XML to solve our data access problems.

During several Department of Defense research projects since the mid-1990s, my colleague and I have had the opportunity to explore various techniques for accessing different types of data sources using XML. These data sources include relational databases, CORBA objects, flat files, and, more recently, LDAP and the J2EE Connector architecture. Initially we used JDBC to access the relational data sources. The JDBC code was tedious to write, however, and difficult to update when new database schemas were released.

XML As a Data Access Language

We started out using XML as a way to specify the common parts of our JDBC code. This provided a static approach for defining our basic access requirements – such as the database URL and our queries – as data within elements of the XML document. We could then hand this document, with the embedded instructions, to a database manager and tell it to pass back a result set by running a specified query. This approach did give us the flexibility of placing our queries in an external file, but the source code required to implement the result set processing was more than we wanted to write and maintain. This document structure was also limited because conditional processing and multiple database accesses were difficult to express as straight XML in the document.

The client part of our program was already completely dynamic. Everything about the client, including the menu bar and dialog boxes, was defined by property files, and the code needed to run the interface wasn't known until runtime. We took the lessons learned from the client side and applied them to the server side of the program. Our desire for a completely dynamic processing engine on the server side pushed us to look deeper for ways to use XML technologies to manipulate data sources without writing any Java code.

Along the way we looked at using XSLT to provide a conditional execution and transformation environment. This turned out to be exactly what was needed. XSLT provided the technology necessary to make our static documents dynamic. The next step was to make our instruction set dynamic and build an extensible processing engine for the instructions.

Java As a Dynamic Plug-in Language

I don't want to spend a lot of time expounding on the virtues of Java. Any programming language could have been used to solve the problem. Having said this, however, Java offers a very compelling platform for creating dynamic, pluggable systems. The Java Reflection mechanism allows us to easily defer the loading of the instruction code until runtime.

The processing framework requires two basic interfaces: one to define the instruction set and another to wrap the data source drivers.

Using Java interfaces to define the idea of an "instruction" and a "driver" allows us to define the contract of the handler without requiring the presence of any code. This allows the loading and knowledge of the instruction handlers and data source handlers to be deferred until runtime. A small property file is used to define each instruction:

```
type=Instruction
Instruction=xdl:transaction
Code=com.saic.xdl.Transaction
```

The instruction set can be extended or replaced, and even the XML namespace used to define the instruction set can be changed. The drivers are also created in this way. The JDBC driver that wraps JDBC is defined in a property file as well:

```
type=Driver
DataSource=mydb
Code=com.saic.xdl.jdbc.JDBCDriver
```

Using a pluggable component-based engine is key to providing a dynamic, loosely coupled framework for handling any type of data source through a common interface.

XSLT as a transformation between the data source and the result

Although originally intended to perform complex styling operations, XSLT is really a general-purpose XML processing language. The processing elements defined by XSLT give us a new way to think about our instruction set. The instruction set was changed, modeled after the basic XSLT elements. XSLT gave us the ontology we needed to think about XML as a dynamic specification language. We adapted XSLT elements to provide methods (i.e., templates), loops, conditional statements, and expression evaluation.

XSLT also provides a mechanism for handling XML input documents. Our first efforts centered on the generation of XML content from data sources. Processing an input XML document was tedious. We used the DOM trees to search for relevant data and then we would use this to update the data source. This approach was hardly dynamic. Using XSLT to process an input XML document allows us to transform the input into a set of data processing instructions (let's call this a "map document"). This allows the handling of XML input and output without any requirements for writing code to handle the data.

XSLT is used to express the business rules of our data processing. The result of the XSLT transformation is a set of processing instructions similar to the XSLT

AUTHOR BIO

Craig King is a senior software engineer with SAIC, an employee-owned research and engineering firm. He has over 17 years' experience working on commercial and research projects using object-oriented technologies and practices.

elements. These instructions have their own namespace, "xdl". When the XSL processor is finished transforming the input document, the result is passed to the processing engine, which iterates over the instructions and performs the required data processing. This may or may not result in the generation of a new XML document.

One other very compelling reason for using XSLT: the Java bindings to the language allow us to extend the functionality of XSL. That gives us the extendability of the preprocessing through XSLT and the extendability of the processing engine through dynamic instruction handling and data source handling.

One very simple feature that really makes working with relational data easier is the ability to create the unique sequence numbers required for most tables before we insert the data. This is much easier than using the database to define its own sequences because the ID is immediately available for use in child tables. The namespace for the extension is defined in the XSLT stylesheet tag, and the extension is then used anywhere an expression is valid:

```
<xsl:variable name="myid"
 select="id:getUniqueId()"/>
```

There are many other ways to make use of the XSLT bindings to Java. When working with multiple data sources you can do quick database lookups to provide conditional behavior before the data sources are queried or updated. It's entirely up to your imagination and creativity.

```
<xsl:variable name="pid" select=
 "query: execute('mydb','select …')"/>
```

The best part of using XSLT to handle the business rules is that it allows you to define your data access requirements outside your program.

XML Grammar / Instruction Set

A set of instructions based on the XSLT elements was used as the core instruction set for our requirements. This is obviously not a requirement for other types of systems. The instruction set could be modeled after the business constraints of a particular domain. Statistical instructions could be defined for a baseball stats system, or monetary instructions could be defined for a financial system. The goal is to keep the instructions abstract in relation to the data sources and not to create data source–specific instructions.

We divided up the basic instruction set into four basic core abstractions: instructions, expressions, functions, and extensions. After I define what each of these mean, I'll provide an example that should make everything clear.

Instructions are a child of an xdl:querysheet element. Instructions provide the context for managing flow control, interpreting application logic, and providing data output. The flow control and conditional evaluation instructions are based on XSLT and consist of the following: template, call-template, choose, when, otherwise, cdata, pi (processing instruction), value-of, and variable. We've defined the data source interaction instructions as transaction, session, for-each, bind, and bind-param. And for XML content creation there are the element and attribute instructions.

```
<xdl:call-template name="mytemplate">
```

Expressions are used to provide operators on the values of an instructions attribute. This is the same as the use of the "$" in XSLT to indicate that a variable is being operated on rather than a constant value:

```
<xdl:value-of select="$id"/>
```

Expressions are based on matching of the first character of the expr attribute with a registered expression handler. The ability to define new expression handlers is very powerful. We've defined expressions to allow evaluation of result set data values (*), variables ($), constant values ('), XPath (/), and others. What about creating an expression handler to read in an XML document from a URL so the contents could be added to a database?

```
<xdl:value-of expr="@file://mydoc.xml"/>
```

Functions are addressed by name and can be evaluated with parameters. A function is executed by the use of the function expression handler (:). The functions we've defined are for the most part limited to string manipulation. These functions perform such tasks as concatenation and substring:

```
<xdl:value-of
  expr=":concat('foo','bar')"/>
```

Extensions are globally defined instructions. Where an instruction is used inside an xdl:template, an extension is a child of the xdl:querysheet and therefore global to the entire document. This distinction is subtle, but it's important to know which instructions are global and which are local. Extensions are used to acquire connections to data sources,

THE LARGEST JAVA & WEB SERVICES EXPO IN THE WORLD
JAVA DEVELOPER'S JOURNAL
BETTER SCALING WITH NEW I/O

format queries presented to the data source, and encapsulate the execution of scripting languages such as Python, JavaScript, or Pnuts.

```
<xdl:statement ns="getUsers">
  select id, first, middle, last
  from users
</xdl:statement>
```

Given these definitions, Listing 1 shows what a complete data map might look like. Given a table with Tom and Jane, the output generated by the map would look like this:

```
<Users>
  <User>
    <Id>1</Id>
    <First>Tom</First>
  </User>
  <User>
    <Id>2</Id>
    <First>Jane</First>
  </User>
</Users>
```

By changing a couple of lines in the map, we can generate the ID as an attribute rather than an element:

```
<xdl:attribute name="id" expr="*1"/>
<xdl:attribute name="first" expr="*2"/>
```

With this simple change the output would appear like this.

```
<Users>
  <User id="1" first ="Tom"/>
  <User id="2" first="Jane"/>
</Users>
```

This basic set of instructions is all that's needed to handle most document-processing requirements. What's left is to come up with a way to handle different kinds of data sources in a uniform way.

Wrapping Data Sources

The wrapping of data sources is done much like the J2EE Connector architecture. That is, a standard interface for defining a data source is provided, but the actual data query specification is still data source–specific. There's no really good way around this, but if there's a tool that provides a transparent language for access to different kinds of data, just wrap that as a source and then access it using this framework.

The query for a data source is provided by the xdl:statement extension. This element provides a container for holding the query. This query is passed to the underlying driver when the processing engine encounters an xdl:session element. The query is simply passed off to the underlying driver. If the underlying driver is a JDBC source, the xdl:statement element would contain an SQL query with optional bind parameters:

```
<xdl:statement ns="mymoney">
  select money
  from bank
  where name = ?
</xdl:statement>
```

The statement "ns" attribute provides the namespace value for addressing the statement at a later point in the processing. The xdl:bind would be used to define a session context for binding values, and the xsl:bind-param provides the actual binding values:

```
<xdl:session ns="%mymoney">
 <xdl:bind>
   <xdl:bind-param id="1"
    expr="'{LastName}'"/>
 </xdl:bind>
</xdl:session>
```

The bind variable can come from a variable, an XSLT parameter, or as the result of a previous query. Data sources can easily be integrated using this technique, and the update of the multiple data sources using a single document is just as easy. The transaction control is nested and does not provide a complete two-phase commit capability, but it is appropriate for a majority of the cases.

When constructing a driver, four things have to be taken into consideration: connections, statements, result sets, and metadata.

When managing connections, remember that the connection user name and password may not be the same for each access. It's easy to place the login credentials in a property file, but the connection mechanism should account for server-based use with many different users and roles – pool the connections if possible.

The statements should be pooled and there should be a way to defer the binding of values into the statement. JDBC includes the ability to bind everywhere a "?" appears in the query. The instruction processing is set up to allow binding for any data source, so the statement handler should handle this. In fact, the parameter handling must be part of the driver interface specification.

Metadata must be made available to create dynamic interactions with the data source. The metadata includes things such as the source schema and the size and format of the returned elements. The result set specification of the driver interface must include handling of metadata.

The driver is important in this framework. Developing it is more difficult than developing an instruction handler, but when done correctly, it allows LDAPs to be treated like relational databases, which are then compatible with flat files, and so on.

Output Styles

The output of the processing is not limited to XML documents. Since XSLT is really just an XML processing language, the output can be any ASCII format you choose. Suppose you need to create a comma-separated file of the output for import into another application. The xdl:value-of instruction can be used to map a result into the output space, but the result doesn't have to be contained within XML elements. Use the xdl:value-of instructions instead of the xdl:element. Add a carriage return at the end of each record:

```
<xdl:for-each expr="*">
   <xdl:value-of expr="*1"/>
   <xdl:value-of expr="','"/>
   <xdl:value-of expr="*2"/>
   <xdl:value-of expr="','"/>
   <xdl:value-of expr="*3"/>
   <xdl:value-of expr="'&#13;'"/>
</xdl:for-each>
```

Again, the kinds of things you can do are limited only by your imagination. XSLT provides the business rules and transformation instructions, the processing engine provides the mapping to the data source through the instruction execution, and you supply the ingenuity.

Performance Implications

Some people believe that XML performance is an oxymoron. While this may be true, it may not be relevant. Current XSLT processors suffer when the XML document size gets large. The larger the document and the more references to top-level nodes, the worse the performance. If a system doesn't rely on large documents, and if the XSLT searching elements are used appropriately, you may not experience performance problems with your system.

Many user interface applications or Web-based applications use small data sets, which will perform well in the type of system we've been talking about. In the end it's up to you, the way you've designed your system, and your specific performance needs.

My experience while working on this XSLT framework indicates that the XML processing overhead for each record returned from JDBC is about .1–.5ms per record. This was the case whether we processed 250 records or 25,000 records. The database query execution time more often than not will dominate the overall processing time.

Conclusion

A pluggable framework for combining instruction processing with data source wrapping using XML and XSLT can be extremely powerful. Using XSLT to provide the business rule execution environment provides an approach, based on W3C standards, to transform any type of data source to XML or to transform and import documents into one or more heterogeneous data sources.

To learn more about what I've been working on for the past five years, or if you have questions about the performance of XML/XSLT in data-intensive applications, please send me an e-mail.

Acknowledgment

I'd like to thank my good friend and colleague, Matthew Zager, who implemented most of the ideas presented in this article.

KINGC@SAIC.COM

LISTING 1

```
<xdl:querysheet>

 <!-- The definition of "mydb"
  is defined in a
      property file -->
 <xdl:datasource ns="mydb"/>

 <xdl:statement ns="users">
  select id, first, middle,
   last from users
 </xdl:statement>

 <xdl:template match="/">

  <!-- top level element of
   the output. -->
  <xdl:element name="Users">
   <!-- name space expression
    % attaches to the
        mydb datasource -->
   <xdl:transaction datasource=
    "%mydb">
    <xdl:session statement=
     "%users">
     <!-- query has been
      executed, iterate
          over the result -->
     <xdl:for-each expr="*">
      <xdl:element name="User">
       <!-- bind the results
        to elements. -->
       <xdl:element name="Id"
        expr="*1"/>
       <xdl:element name=
        "First" expr="*2"/>
      </xdl:element>
     </xdl:for-each>
    </xdl:session>
   </xdl:transaction>
  </xdl:element>
 </xdl:template>

</xdl:querysheet>
```

Download the Code
www.sys-con.com/xml

wireless
BUSINESS & TECHNOLOGY
MOBILE VPN TECHNOLOGY
SECURING MAGNOT LINE Wireless LANs
WHAT'S VALUE
GENERATING REVENUE TO ADVANCE THE M COMMERCE
MOBILE OPERATORS AND CONTENT PROVIDERS NEED TO CREATE & SHARE BUSINESS MODELS

HOME
Enterprise Solutions
Content Management
Data Management
XML Labs

WRITTEN BY **TOM GAVEN**

Desperately Seeking… Help for XML Schema

Are the complexities...too complex?

Whether it's the Russian dolls, Salami Slices, or Chameleon Schemas discussed at www.xfront.com, the dizzying array of elements and attributes (and their complex interactions) found in the XML Schema specification, or the multitude of UML stereotypes at www.xmlmodeling.com, W3C XML Schema will disappear into oblivion without a non-XML syntax and a set of easy-to-use and -understand best practices and modeling techniques.

There's been a lot of interest in XML Schema, and well there should be. A lot of very smart people put the XML Schema spec together, and I'm sure it took an amazing amount of effort. But if I can be frank for a moment, XML Schema *is* rocket science. From noNamespaceSchemaLocation to block attributes and substitutionGroups, the XML Schema syntax is simply too complex….STOP the MADNESS and just say NO! While I fault the XML Namespace Recommendation for many of the complexities haunting the XML Schema Spec today, many of the wounds are self-inflicted. One (small) case in point is XML Schema – complexType Inheritance.

complexType Inheritance Is Simply Broken

One of the main object-oriented features of XML Schema is complexType inheritance. Most tutorials (including most introductory tutorials, like the XML Schema Primer) about XML Schema have examples that show off this feature, but complexType inheritance is simply broken. In addition to the syntax being overly complex (extension, restriction, simpleType, complexType, simpleContent, complexContent, block, abstract), it's simply not useful for type reusability.

Why? complexType inheritance has two major limitations:

1. You can extend a base class definition only "to the right."
2. You can extend the definition only with a sequence.

Here's an example that illustrates these problems:

Let's assume I have a complexType called XType.

```
<xsd:complexType name="XType" >
    <xsd:sequence>
         <xsd:element name="a" type=
          "xsd:string" />
         <xsd:element name="b" type=
          "xsd:string" />
         <xsd:element name="c" type=
          "xsd:string" />
    </xsd:sequence>
</xsd:complexType>
```

That is, XType has a content model of (a,b,c). For simplicity I'll use DTD syntax here to demonstrate the problem. (It's unfortunate that I need to resort to DTD syntax to discuss a problem with XML Schema...but I digress.)

Now let's say I wanted to reuse the parent type XType, and model several "subtypes" that correspond to the following content models:

1. (a,b,c),d
2. (a,b,c)|d
3. ((a,b,c),d)+
4. (a,b,c)*
5. ((a,b,c), d) | e)
6. d, (a,b,c)
7. d, (a,b,c)+,e

Unfortunately, only No. 1 above can be modeled using complexType inheritance. That is, a child type can extend the parent's type only "to the right." Moreover, the child type can extend the parent type only with a sequence…that is, a comma between the parent and child definition. (Even No. 3 above, which uses a sequence, is invalid because of the grouping of the child's model with the parent's model.)

Of course, we can support Nos. 1–7 above, and get all the reusability we want with XML Schema model groups. But the problem here is that I can't just throw away the complexType syntax and replace it with model group syntax. If I use model groups, I have to complicate my XML Schema document even more. This is probably why many tutorials on XML Schema avoid discussing model groups altogether, which is unfortunate, because model groups are the best way to get flexibility and reuse with XML Schema. Another restriction of complexType inheritance is that it doesn't support multiple inheritance. Model groups, however, can be used to coalesce any number of other model groups in reusable combinations.

But wait a minute, you say, isn't complexType inheritance useful for other things besides reuse, like "polymorphism" or base type "restriction"?

Well, yes, that's true, but they're also broken.

To achieve base type restriction with XML Schema, we have to duplicate the entire content model of the parent type in the child's type definition. Yes, that's right, you have to copy and paste the parent's definition into the child!

Say *whaaat*?!!??

To achieve polymorphism with XML Schema, we have to place xsi:type attri-

Author Bio

Tom Gaven has authored over 30 courses in many different technologies, including Assembler, C, C++, Java, OS/2, and Windows. He also authored MindQ's Developer Training for Java program. In the past two years he has architected and developed products with XML, XSLT, XML Schema, RELAX NG, Java, and Schematron. Tom is currently working on tools and courseware to make XML easier to use.

	RELAX NG	XML Schema
Size of Spec (pages)	57	258
Length of Schema (lines)*	330	1206
Committee Members	8	47 plus 35 contributors

* Size of Schema represents the length of their respective 'Schema for Schema' documents.

TABLE 1 | Tale of two schema languages, RELAX NG and XML Schema

butes in our instance documents. This means that document authors and designers are forced to use attributes dictated by the W3C in their business vocabularies! It also forces much complexity on the programmers trying to process these documents. Doesn't this fly in the face of what XML is all about?

Desperately Seeking...
Easy-to-understand best practices and modeling techniques

While I greatly appreciate all the effort on best practices and XML modeling at Web sites like www.xfront.com, www.xmlmodeling.com, and www.xmlpatterns.com, I feel they miss the mark. What we need is a set of simplified XML Schema best practices and modeling ideas. We need best practices and modeling initiatives that rein in, rather than accentuate, the numerous complexities of the XML Schema specification.

Desperately Seeking...
NON-XML syntax for XML schema

We also need a compact syntax for XML Schema. Let's take a look at an example, a sample XML instance document for a "course":

```
<course name="XML Schema, Distilled" >
    <module name="Overview of Model
     Groups" >
       <lesson name="Building a Model
        Group" >
           <slide name="What is a
            Model Group" >
             <p> Model Groups are
              quite <b>powerful</b>
               and <i>easy</i> to
                understand.
             </p>
            </slide>
            …
       </lesson>
   </module>
</course>
```

And here's a complete compact schema document that can be used to ultimately validate the XML instance document above (using James Clark's compact syntax for RELAX NG):

```
start = course
course = element course
     { name, module+ }
module = element module
     { name, lesson+ }
lesson = element lesson
     { name, slide* }
slide = element slide { name, content }
name = attribute name { text }
content = ( contentTags )*
contentTags = element (p | b | i )
       { ( text | content ) }
```

One thing to note is that the schema above is seven lines shorter than the equivalent DTD...and look, mom, no parameter entities!

Desperately Seeking...
Help for XML schema

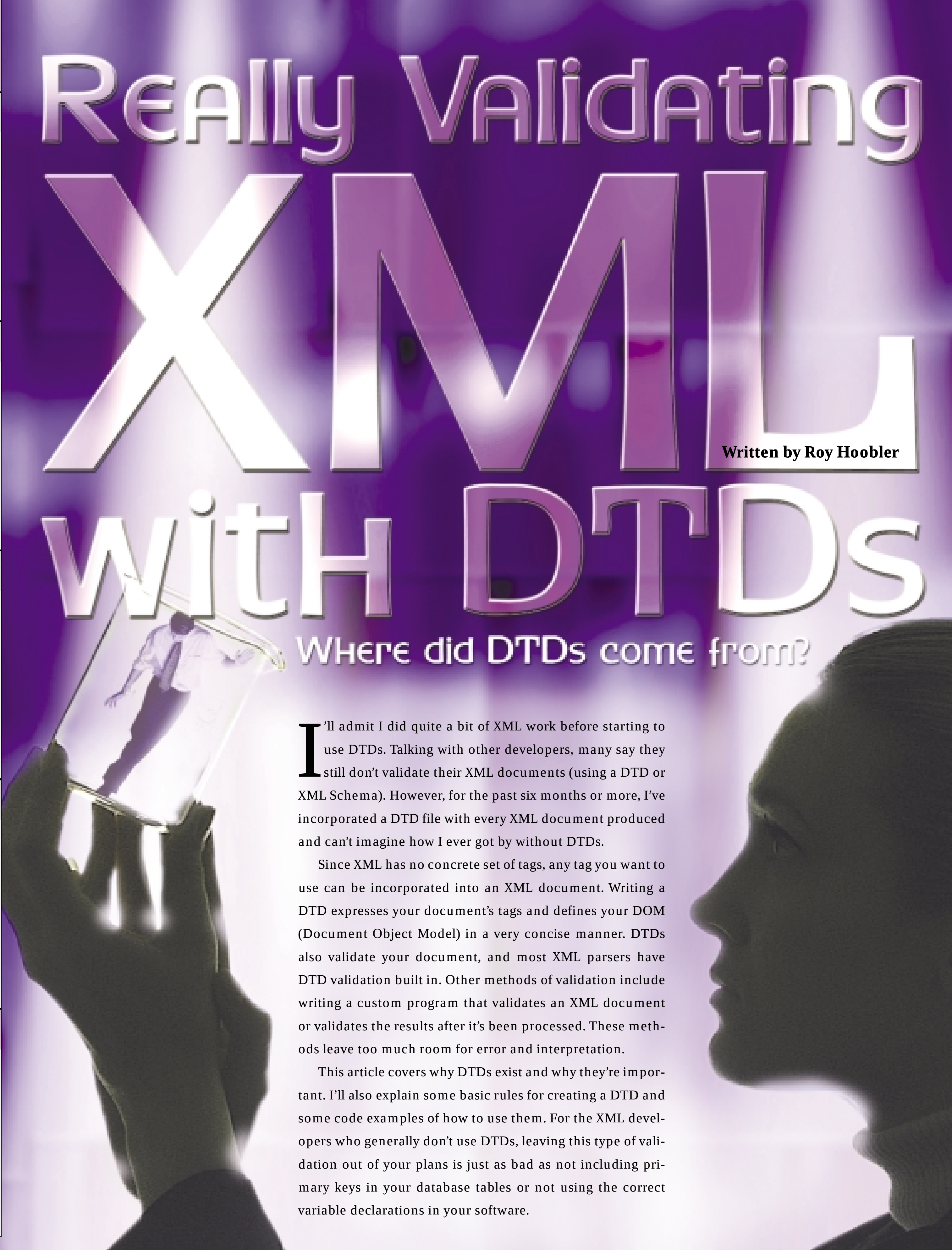
Hopefully, the smart XML Schema folks are working on a non-XML syntax for XML Schema. Perhaps James Clark's syntax can be leveraged here, without even exposing so-called "features" like complexType inheritance, concentrating on creating "easier to use and understand" XML Schemas. In addition, let's create a set of XML Schema best practices and modeling tools built with ease of use in mind. These should be engineered to hide, rather than accentuate, the complexities of XML Schema.

TGAVEN@XMLDISTILLED.COM

The World Wide Web Consortium (W3C)

The W3C was created to lead the Web to its full potential by developing common protocols that promote its evolution and ensure its interoperability. An international industry consortium jointly run by the MIT Laboratory for Computer Science (MIT LCS) in the U.S., the National Institute for Research in Computer Science and Control (INRIA) in France, and Keio University in Japan, services include a repository of information about the Web for developers and users, and various prototype and sample applications to demonstrate the use of new technology. To date, nearly 500 organizations are members of the Consortium. For more information see www.w3.org/.

WebServices JOURNAL
.NET J2EE XML
Web Services for Enterprise Application Integration
C++ vs C#
THE FRAMEWORK BEHIND WEB SERVICES INTEGRATION
EAI FOCUS



Really Validating XML with DTDs

Where did DTDs come from?

Written by Roy Hoobler

I'll admit I did quite a bit of XML work before starting to use DTDs. Talking with other developers, many say they still don't validate their XML documents (using a DTD or XML Schema). However, for the past six months or more, I've incorporated a DTD file with every XML document produced and can't imagine how I ever got by without DTDs.

Since XML has no concrete set of tags, any tag you want to use can be incorporated into an XML document. Writing a DTD expresses your document's tags and defines your DOM (Document Object Model) in a very concise manner. DTDs also validate your document, and most XML parsers have DTD validation built in. Other methods of validation include writing a custom program that validates an XML document or validates the results after it's been processed. These methods leave too much room for error and interpretation.

This article covers why DTDs exist and why they're important. I'll also explain some basic rules for creating a DTD and some code examples of how to use them. For the XML developers who generally don't use DTDs, leaving this type of validation out of your plans is just as bad as not including primary keys in your database tables or not using the correct variable declarations in your software.

DTDs Define and Validate XML

Many organizations today produce their own markup languages such as MathML and DocBook XML. These formats are defined by DTDs, not XML documents. By definition, for any XML document to be valid it must be compared against a DTD. For a document to be a valid MathML it must be validated against the MathML DTD (or Schema), not just look like a MathML document. Another example: if a contact application uses XML data to transfer a phone list, publishing a DTD is a convenient and reliable approach to describe and validate that data. If other business partners or developers want to build an application to read your contact list, giving them a DTD will ensure that their application processes the data correctly.

Planning and creating the DTD takes more time at first but will save development and testing time later. At the end of the development cycle, testing the application will take less time since most "validation problems" will already be resolved. In my own experience, I've been very surprised by validation errors against documents I thought would validate fine – they even worked in the application.

A point I'd like to make is that a DTD is similar to variable declarations. A DTD describes your Document Object Model and checks the XML file before it can be processed. Some programming languages, such as Microsoft Visual Basic 6.0 and lower, allow you to use variables without declaring them. So you can write "height=5" in one line and a few lines later write "height=hieght+5". The compiler won't know you misspelled the second variable and everything will work fine until you run your program and the output for "height" will be 5, not 10 as expected. To fix this problem, VB programmers use "Option Explicit" at the top of the code so all variables must be declared before the code can be compiled. With Option Explicit the VB compiler will return an error "variable height is not defined". A DTD with XML works in a similar fashion. Nodes can be well formed, but are they spelled correctly? Are they in the correct order? Is an attribute missing?

The DTD also acts as a rule book that checks a document to see if it matches specifications set by the creator. A developer, looking at a more complex DTD for the first time, may think DTDs are too verbose and almost cryptic. However, after reading through a few short tutorials, DTDs are simple compared with the XML document and most applications need only a simple DTD to validate XML.

Figure 1 illustrates how DTDs are used to enforce the tag names, attribute names, and structures that might appear in an XML document.

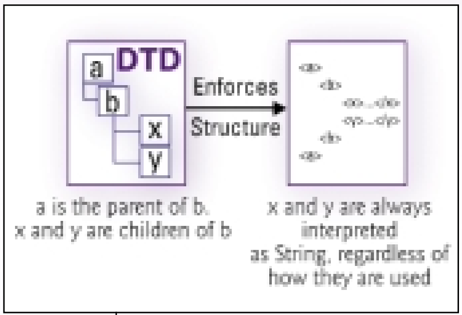

DTD
a
b
x
y
Enforces
Structure
a is the parent of b.
x and y are children of b
x and y are always interpreted
as String, regardless of how they are used

FIGURE 1 | How DTDs are used to enforce structure

Start small; the DTD specification contains a lot of information and functionality, but most of the time using a simple DTD file is appropriate (such as the example in the section "Writing a DTD," below).

DOCTYPE Links XML Documents to the DTD

XML and SGML (and even some HTML) documents can have a simple DOCTYPE declaration at the top of the document. When the XML document is loaded into a validating parser, the DTD is also loaded. The example:

```
<!DOCTYPE article SYSTEM "http://www.mysite.com/dtd/docbook.dtd">
```

tells us that the root element is an "article" and where the DTD can be found. A root element can be any element found in the DTD. The URL needs to be valid and point to a valid DTD. File URLs are also supported, so "/Java/mydtds/docbook.dtd" could replace the URL above.

Authors could state document information with comments, but comments can often be overlooked and there is no standard. Many XML specifications use <book> or <SalesOrder> as a root element; without a DOCTYPE processing instruction and an associated DTD, you'd have to read through every single tag and look for comments to accurately guess what the intended DOM might be.

With a DTD you can tell what type of document the XML is by quickly opening the DTD file, as well as get an overview of elements and attributes (which are required or optional). HTML editors such as MS Interdev and Dreamweaver add a DOCTYPE tag to the top of HTML documents. Fortunately browsers don't validate these HTML documents (since most developers and products don't actually create valid documents in the first place). In this case the Document Type listed does let developers know this is actually an HTML (or XHTML) document and which version of HTML the product supports.

Writing a DTD

If you've never written a DTD before, the example below shows how a simple DTD is put together. Just like XML, elements can contain other elements, text, and attributes. DTDs do more by restricting data, the occurrences of an element, the order in which elements appear, and the selection list for attributes. I'll keep it fairly simple for this article.

Each element in a sequence can be marked as follows:

+ = one or more occurrences
* = zero or more
? = zero or one
| = choice
(nothing) means the element is required and occurs only once

Attributes are listed as #Implied; optional or #Required; required.

#REQUIRED
#IMPLIED

Other options such as #FIXED enable you to supply fixed (default) values for a given attribute.

A rule to remember is that elements in the XML document must be in the same order as in the DTD.

To view a DOM hierarchy, the DTD format is a little easier to get around with than XML. For simple documents you may only have to look at one node. The example in Listing 1, from an application I've been working on, describes a retail Product List XML document and the associated DTD. Browsing the DTD, it's easy to see that a catalog has products, and products have prices and images.

The "catalog" element is our root element. Most other elements, such as description, sizes, and thumbnails, fall under each product. A product itself could be a root element if declared as such in the DOCTYPE declaration of the XML file. A product also has four required attributes: sku, category, price, and isFeatured. Sizes, on the other hand, have an optional "label" attribute.

Even this DTD example is more complicated than I first wanted to show, but it's a good example, coming from an actual application, and it validates over 2,000 lines of XML.

For more information about writing DTDs see Cafe Con Leche at www.ibiblio.org/xml. There's a nice DTD tutorial, usually an XML "quote of the day," and several news articles covering XML and DTDs.

Design a DTD Before Finalizing an XML DOM

Print and online sources admit that it's better to create a sample XML document before creating the DTD. This technique works – as long as you are willing to change. However, creating the DTD before (or alongside) the XML will lead to better design and a more efficient practice for developing applications. If a developer delays in writing the DTD, it might not be integrated into the system until after the application is finished. The end of the development cycle is not the time to find out the XML contains validation errors.

If the XML is created first, then a DTD should be created as soon as possible thereafter. This follows the same principle of OOP design practices and other "best practices." After writing and reviewing a couple of DTDs, I found them easier to use as a reference and maintain than sample XML documents.

Finally, DTDs allow developers to have better control of the design of their DOM/XML documents. Changes made to the DOM structure will work only if incorporated into the DTD.

DTDs vs XML Schemas...

As XML begins to be used for transporting data within applications, many people are opting to use XML Schemas. The argument for using them is that they're in XML format so developers have to learn only one "language" and support data types. DTD files are not XML and do not provide a way to check for data types (Integer, String, Date, etc.). The only type available with a DTD is "character data" – all elements are interpreted as a String.

Producing an XML Schema is much more complicated than writing a DTD, simply due to the complexity of the XML Schema Recommendation. W3C XML Schema is so complex that it had to be broken into three specific documents:

1. XML Schema Part 0: Primer (required because Parts 1 and 2 were deemed too difficult for many people to understand)
2. XML Schema Part 1: Structures
3. XML Schema Part 2: Data Types

If you were to print out the entire XML Schema Technical Recommendation, it would require 264 pages. This is somewhat overwhelming when compared with the XML Technical Recommendation itself (46 pages, including several examples using DTD syntax).

An interesting note: there is a nonnormative DTD that describes the structure of a W3C XML Schema (available at www.w3.org/TR/xmlschema-1/#nonnormative-schemaDTD).

Note that unlike DTDs, XML Schema provides full data type support, potentially offloading some of the data validation requirements from the developer to the schema itself. This may result in a smaller code base, depending on the design of the XML Schema.

Most programs that process XML should be checking for data type validation anyway (especially true for those programs using DTDs). In my view, if the data is wrong, it doesn't matter if the DTD tells you it's wrong or the program does.

An XML Schema or DTD could be used depending on the situation. It's better to think of using DTDs for content-oriented applications and schemas for data-centric applications (or for applications that share specific data types). Being able to validate the document either way is better than nothing at all.

How Validation Works, and Parsing XML Using DTDs

Before a document is "validated," the parser checks the XML for well-formedness (start tags matching correctly with end tags, quotes around attributes, all character data is valid, etc.). Checking for well-formedness is a separate process from validating the document with a DTD (or schema). While all XML parsers perform a well-formedness check, not all are capable of performing validation with a DTD or schema (these are "nonvalidating" parsers; the rest are "validating" parsers).

Once the document is loaded and checked for well-formedness, each node is checked against the DTD. If a node or element appears that doesn't match the DTD, the parser will stop and issue an error. Once everything has been validated, the XML is ready to be transformed, queried, or parsed out to another system.

Validation requires a lot of overhead and definitely slows things down. Validating an XML document against a DTD every time a document is loaded is not the answer for many applications. For Web applications, producing the output in HTML or PDF format solves this problem, sending only HTML pages to the Web site to be viewed by the masses. Creating a "mirrored" XML file that isn't validated for routine system use is another possible solution. Additional initiatives include passing in-memory DOM trees and generating serialized DOM structures.

Yet another consideration is to use a DTD during unit testing, validating the XML only in unit test modules. If a new document is received or generated and causes an error, the unit test can be run with the new document to check for the error or validate the content. In other situations, DTD validation may occur only once, when the document is created or received. If errors occur, the user or system is notified and corrective action can be taken.

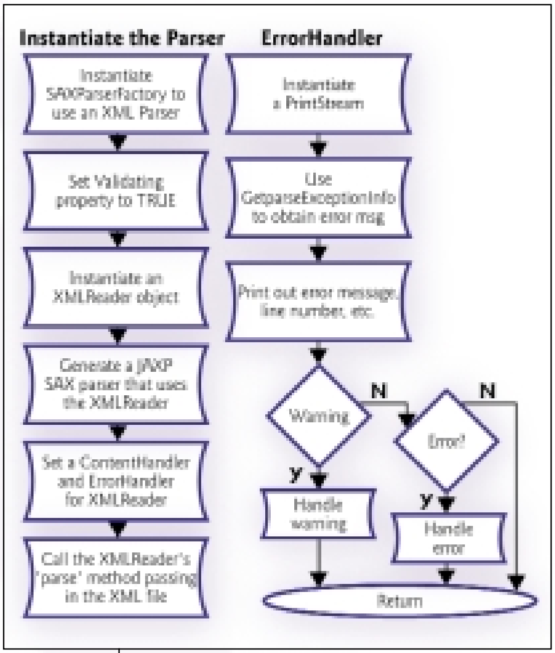

Instantiate the Parser
Instantiate SAXParserFactory to use an XML Parser
Set Validating property to TRUE
Instantiate an XMLReader object
Generate a JAXP SAX parser that uses the XMLReader
Set a ContentHandler and ErrorHandler for XMLReader
Call the XMLReader's 'parse' method passing in the XML file
ErrorHandler
Instantiate a PrintStream
Use GetparseExceptionInfo to obtain error msg
Print out error message, line number, etc.
Warning
N
y
Handle warning
Error?
N
y
Handle error
Return

FIGURE 2 | Instantiating the Xerces XML parser

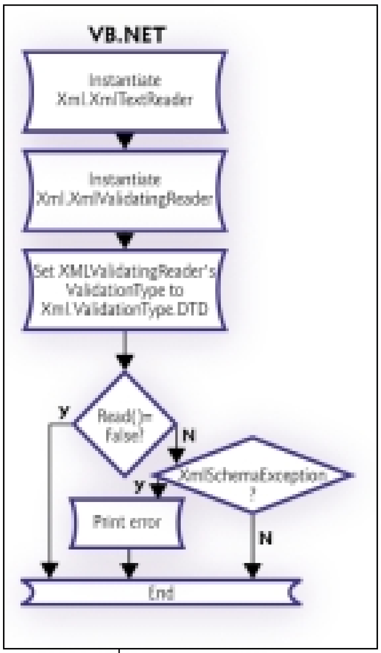

VB.NET
Instantiate Xml.XmlTextReader
Instantiate Xml.XmlValidatingReader
Set XMLValidatingReader's ValidationType to Xml.ValidationType.DTD
Read()= False?
y
N
XmlSchemaException ?
y
N
Print error
End

FIGURE 3 | Visual Basic .NET parser

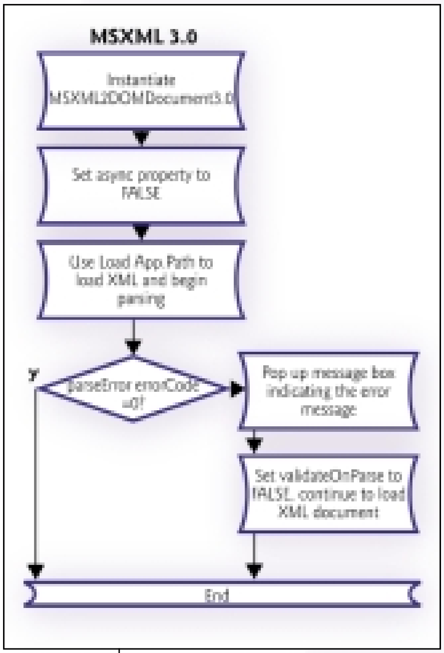

MSXML 3.0
Instantiate MSXML2DOMDocument3.0
Set async property to FALSE
Use Load App.Path to load XML and begin parsing
ParseError errorCode =0?
y
Pop up message box indicating the error message
Set validateOnParse to FALSE, continue to load XML document
End

FIGURE 4 | MSXML3.0 parser

Many XML parsers and transformation (XSLT) parsers are available. Microsoft supports the MSXML3.0 XML and 4.0 parser that work with IE5.0 and above. Also, Visual Studio.NET and the .NET framework have some impressive XML assemblies (.dll components).

Developers working with Java usually begin developing with Xalan (for XSLT) and Xerces (an XML parser) from the Apache Group. Many Java developers also utilize the JAXP classes available from Sun. Since the W3C publishes the XML specification, all parsers have generally the same methods and properties – the major differences tend to involve instantiating the parser and loading the XML document into it.

JAXP, mentioned above, provides a common interface to XML parsers, making it easier to change parsers without changing your implementation – change a system property and you're ready to go. The JAXP classes may help avoid vendor-specific parser methods. If most of your code uses JAXP, no other changes are necessary.

Using DTDs with Parsers to Validate and Transform

In the Java-based Xerces XML parser, the validation can be turned on or off simply by setting a property within the parser. However, to retrieve errors, your class must implement a specific class (known as DefaultHandler) to catch any errors that may be generated by the parser. This is accomplished using the event-driven SAX API instead of W3C's DOM API. The former is used since it can raise events (including error events) that are easily handled by the developer.

Figure 2 provides a simple diagram illustrating the steps to instantiate the Xerces XML parser. The complete code example is available from the **SYS-CON** Web site (see "Resources" section).

By default, for performance issues, most parsers have validation turned off. With Microsoft .NET a separate ValidatingParser class is created for validating an XML document (see Figure 3).

MSXML3.0 (see Figure 4) validates by default. You can receive any errors via the xmldom.parseError object. To turn validation on, set the parserValidation property to True. This example also transforms the document using XSLT.

Tools Available

Development tools such as Forte, Xerces, Visual Basic, and .NET are good for developing software that processes and validates XML documents. Other programs such as EditML Pro, XMetal, and XML Spy produce basic DTDs and schemas for completed XML documents. However, after the DTD is generated, the developer or person working with the document will have to finish the DTD, creating the hierarchy and rules for elements and attributes (allow more than one of this element, and choose child elements and attributes, etc.). Generally, I've built my DTDs by hand and used the .NET code (see Resources) to test and validate the DTD and XML.

Acknowledgment

I'd like to thank Anna Ukhovskaya and Alona Rabin for their help in researching and testing the examples in the listings for this article.

Resources

The following code listings can be downloaded from www.sys-con.com/xml/sourcec.cfm.

1. Sample Java code illustrating the use of JAXP and Xerces
2. Sample VB.NET code illustrating how to validate an XML document
3. Sample script illustrating how to validate XML using MSXML 3.0

AUTHOR BIO

Roy Hoobler has been developing custom Web applications since 1996. After completing his MCSD certification, he spent the mid-'90s at a large consulting firm focused on intranet/extranet applications for Fortune 1000 companies. In 1998 Roy joined Net@Work (www.netatwork.com) as director of Internet technologies, specializing in systems architecture, project management, and research into emerging programming methods.

RHOOBLER@NETATWORK.COM

LISTING 1 XML example with a DTD

```
DTD File (located at c:\products.dtd):
<!ELEMENT catalog (product+)>
<!ELEMENT product
(name,thumbnail*,images*,description?,sizes*,colors*)>
<!ATTLIST product
 sku CDATA #REQUIRED
 category CDATA #REQUIRED
 isFeatured CDATA #REQUIRED
 price CDATA #REQUIRED>
<!ELEMENT name (#PCDATA)>
<!ELEMENT description (#PCDATA)>
<!ELEMENT thumbnail (#PCDATA)>
<!ELEMENT images (image+)>
<!ELEMENT image (#PCDATA)>
<!ATTLIST image
 name CDATA #IMPLIED>

<!ELEMENT sizes (size+)>
<!ELEMENT size (#PCDATA)>
<!ATTLIST sizes
 label CDATA #IMPLIED>
<!ELEMENT colors (color+)>
<!ELEMENT color (#PCDATA)>

XML document using the DTD:

<?xml version="1.0"?>
<!DOCTYPE catalog SYSTEM "c://products.dtd" >
<catalog>
 <product category="soldout" sku="0046" price=
   "20.00" isFeatured="false">
  <name>Fleece Pants</name>
  <thumbnail>0044.jpg</thumbnail>
  <images>
   <image name="Rust">0044rk.jpg</image>
  </images>
  <description>Great casual look sweat pants.
    </description>
  <sizes>
   <size>M</size>
   <size>L</size>
   <size>XL</size>
   <size>XXL</size>
  </sizes>
  <colors>
   <color>Charcoal with Yellow Stitch</color>
   <color>Ash with Orange Stitch</color>
   <color>Navy with Orange Stitch</color>
   <color>Black with Orange Stitch</color>
   <color>Rust with Khaki Stitch</color>
  </colors>
 </product>
</catalog>
```

Download the Code
www.sys-con.com/xml

EDITED BY
SEAN RHODY

THE MOST COMPLETE LIBRARY OF EXCLUSIVE WSJ & XML-J ARTICLES ON ONE CD!

"The Secrets of the Web Services Masters"

CD is edited by well-known WSJ Editor-in-Chief Sean Rhody and organized into more than 40 chapters containing more than 400 exclusive WSJ & XML-J articles.

Easy-to-navigate HTML format!

Bonus:

Full .PDF versions of every WSJ & XML-J published since the first issue

XML in Transit
XML B2B
Java & XML
The XML Files
XML & WML
Voice XML
SYS-CON Radio
XML & XSLT
XML & XSL
XML & XHTML
2B or Not 2B
XML Script

XML Industry Insider
<e-BizML>
XML & Business
XML Demystified
XML & E-Commerce
XML Middleware
XML Modeling
CORBA & XML
Data Transition
XML @ Work

XML & Databases
Electronic Data Interchange
Ubiquitous Computing
Information Management
Objects & XML
XML Pros & Cons
Java Servlets
XML Filter

UML
Integration
WSDL
Beginning Web Services
Web Services Tips & Techniques
Frameworks
Management
Security
UDDI
.NET

3 YEARS
25 ISSUES
400 ARTICLES
ONE CD

WebServices
XML JOURNAL
THE FIRST & ONLY WEB SERVICES RESOURCE CD
WEB SERVICES RESOURCE CD
THE SECRETS OF THE WEB SERVICES MASTERS
INCLUDES EXCLUSIVE .NET ARTICLES
MORE THAN 400 EXCLUSIVE WEB SERVICES & XML ARTICLES
$119 CD VALUE
EDITED BY SEAN RHODY
EVERY ISSUE OF WSJ & XML-J EVER PUBLISHED

Now Shipping

$79

ONLINE

ORDER AT JDJSTORE.COM

SAVE $40

Breeze XML Binder 3.0

by The Breeze Factor

John Walker, an independent software developer in Oakland, CA, has over 19 years of experience in application development and relational databases.

REVIEWED BY **JOHN WALKER**

The Breeze Factor

1042 N. El Camino Real, Suite B418
Encinitas, CA 92024

Phone: 888 965-0088

E-Mail: info@breezefactor.com

Web: www.breezefactor.com

Test Environment

OS: Windows 2000 (Service Pack 2), Dell Precision 340 Workstation, 40 GB Disk

Processor: Pentium IV 1.80 GHz

Memory: 512MB RAM

Specifications

Platforms: Any platform with JDK 1.1, 1.2, 1.3, or 1.4 support

Pricing: Contact The Breeze Factor for licensing costs

One of the first rules any database programmer learns is that, by its very nature, data is dynamic. Therefore, any code acting as the enterprise's conduit to that data, whether it's C/C++, Visual Basic, or Java, needs to be dynamic in its ability to incorporate the changes the data reflects. Enterprises using a great deal of code to manage data create more work for the engineers who are responsible for that code. What's been true of the nature of data in hierarchical and relational databases in the past is now true for the generation of programmers working with XML today.

While XML is universally hailed as the most open methodology for describing and declaring data, technologists and vendors are still trying to find the means to implement the data the standard provides – particularly as the standard has continued to evolve and become more powerful. Two years ago the power and scalability of XML schema files wasn't even being discussed; DTD files were the standard. It's as if the rule should be changed now to say that the code managing the data has to be flexible enough to incorporate the changes that the data and the data's supporting technology present.

The DOM, SAX, and JDOM APIs have each provided a framework that has played an important role in the evolution of XML parsers that programmers have used over the past two years. Each offers an interface that allows for the programmatic traversing, reading, and manipulation of XML documents. While both SAX and JDOM are less demanding of memory and processor resources, all require a good deal of code for their implementations. Since no tool is being used to provide a framework to properly bind this code to its data source, there's a lot of code to change when the underlying data definitions change.

Breeze Factor's XML Schema Binder 3.0 can handle the problems faced by programmers when managing dynamic data schemas by creating the code that allows that data to be accessed in a more intuitive way. Implementing code with either the Document Object Model or the Java Document Object Model results in code that functions in a more procedural manner than object-oriented code. Breeze Factor has created an API that, while it still logically presents the nodes of the XML document as relating to one another in a hierarchical fashion, represents the individual nodes in the API as objects. When the data these nodes represent, or the definition of that data, changes, the objects associated with them can be regenerated. The less complex and more intuitive the code that manages the data is, the easier life will be for the enterprise.

Installation and Configuration of Breeze XML Binder 3.0

The installation set used for this demo was acquired from the download area at the Breeze Factor Web site (www.breezefactor.com/download.html). After I provided an e-mail address to receive the product's activation key, an executable InstallShield installation file was downloaded. The installation set provided the Breeze Schema Binder as well as the Breeze DTD Binder. This was very nice as it allowed a comparison of code generated by both .xsd and .dtd files. Numerous examples were provided of differing .xsd file implementations.

HTML-formatted help and documentation come as part of the installation set and can best be described as robust. Complete Javadocs of the API's class files are provided along with a wonderful explanation of the roles of the Java class files that the XML Binder tool generates. The generated code help file should be read before you attempt any of the examples so you can acquire a general understanding of the functions of the various files provided. Also, the breezetk.jar and xerces.jar files need to be included in the system's CLASSPATH. These files can be found in the /dist directory of the Breeze XML Binder home directory.

The home directory of the generated Java files will also need to be included in the CLASSPATH. The steps described here are really all that needs to be done as the overall installation of the product is straightforward and scarcely labor intensive. Should any problems arise, the online technical support that Breeze Factor provides is terrific and spectacularly prompt.

Working with Breeze XML Binder 3.0

Breeze XML Binder generates the Java classes that manage data in XML documents from XML Schema Definition files that are used to describe that data. The tool contains facilities for both the creation and editing of an .xsd file from scratch or for its simple importation from an existing file. The Schema Editor window provides the interface from which changes can be made to the .xsd's definition. These changes can easily be synced with one button click to the XML Binder window that represents the definition of the .xsd from which the supporting Java files will be generated. It should be mentioned that tools also exist for the creation of .xsds from either .dtd or XML files. Support is also provided for the generation of code to manage data in a relational database. The flexibility of the architecture is then portable to .xsd, .dtd, and relational database formats (see Figure 1).

For the purposes of this article an .xsd was created that defines a golf course scorecard. It serves as an example of how the managing classes Breeze XML Binder generates represent nodes and their children as objects with their own attributes and getter/setter methods. Furthermore, the Breeze XML Binder framework implements child nodes as encapsulated objects in the parent node classes. Golf courses have many holes, and these holes have many tees and handicap ratings. Using an object-oriented implementation, code is generated for a Course class, which contains references to a Hole class. The Hole class contains references to both a Tee and a Handicap class.

This is a departure from the framework that developers have been used to with other DOM APIs in that this is more intuitive and will scale more easily. One of the powerful features of .xsd is that it allows for the definition of native data types for each element. The code generated by Breeze XML Binder takes advantage of this so the resulting API can make reference to these data types. For example, each hole has a par rating that should be represented as an integer (int). The generated Java files will incorporate this definition and allow the par element to be returned to and managed by the implementing application as an integer. With the DOM, SAX, and JDOM, programmers had to remember to perform these casting functions on their own. Now, with Breeze XML Binder, the appropriate wrapper classes are being created automatically to perform these functions. Also, should the data, or its definitions, ever change, a simple regeneration of these supporting classes incorporates the changes into new versions of the supporting files.

The supporting files are generated by simply selecting the Generate Code button at the top of the screen. Using the course.xsd file in the example creates five files for managing the document from an application. A file is generated that represents each complex type in the .xsd file definition along with an additional ObjectFactory file that creates the instance of the XML document to be managed by the application's API. The code in Listing 1 illustrates how these files are used. The ObjectFactory returns an instance of the XMLObject class that in turn represents the root node, CourseCourse. The resulting object allows programmers to traverse and manipulate elements in the XML document in a more familiar object-oriented paradigm via getter and setter methods. Programmers familiar with the DOM, JDOM, and SAX implementations will appreciate the simplicity of the solution Breeze XML Binder provides.

Summary

Breeze Factor has provided a tool that supports the easy implementation of XML documents on many levels. The XML Binder incorporates a Schema Editor that validates .xsd files while allowing for their editing and creation. The code generator constructs classes that logically represent the nodes in the document and facilitate their access and maintenance even as the data, and the elements that define it, scales.

WALK@SBCGLOBAL.NET

Product Snapshot

Pros

- Easy-to-use schema editor
- Support for both XSD and DTD formats
- Generated code well designed and easy to implement
- DOM object not held in memory during use

Cons

- Nothing significant

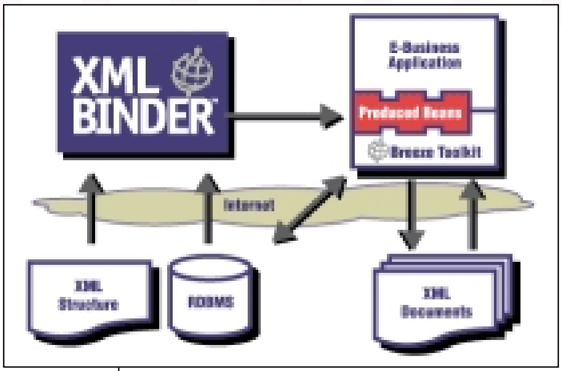


FIGURE 1 Binder architecture provides support for XML and RDBMS data sources.

LISTING 1 BreezeCodeExamples.java file

```
/*
 * Create an XML object from the provided file.
   We use an instance
 * of the ObjectFactory to parse the file and
   create an instance
 * of the appropriate generated class.
*/
String filename = "c:\\walkerbros\\sampledir\\
 golf\\perry_hollow.xml";
ObjectFactory factory =
 ObjectFactory.getInstance();
XmlObject obj = factory.getInstance(filename);
if (obj == null) {
 System.err.println("Error creating object: " +
  factory.getLastError());
 System.exit(1);
}

    /** Begin -- Course Objects generated by
      Breeze XML Binder 3.0 */

    CourseCourse course;
    CourseHole hole;

    HoleHandicap handicap;
HoleTee tee;

/** End -- Course Objects generated by Breeze
 XML Binder 3.0 */


       course = (CourseCourse)obj;

System.out.println("There is course info...");
            System.out.println();
            System.out.println("Location     :
             " + course.getCourseLocation());
            System.out.println("Name         :
             " + course.getCourseName());
            System.out.println("Num. Holes: "
             + course.getCourseNumberOfHoles());
            System.out.println("Par          :
             " + course.getCoursePar());

hole = course.getCourseHoleAt(0);

            System.out.println
             ("There is hole info...");
            System.out.println();
            System.out.println("Number    :
             " + hole.getHoleNumber());
            System.out.println("Par      :
             " + hole.getHolePar());
            System.out.println("Num. tees :
             " + hole.getHoleTeeCount());


}
```

Download the Code
www.sys-con.com/xml

Dot.com
- buyer's guide
- xml forums
- mailing list
- xml jobs
- xml store

Magazine
- advertise
- authors
- customer service
- editorial board
- subscribe

Content
- archives
- digital edition
- editorial
- features
- interviews
- product reviews
- source code

Conferences
- xml edge
  santa clara, ca
  oct 22–25

Search XML-J
Search

Check out these companies and their committed XML technology!

What's Online

www.sys-con.com/xml

Readers' Choice Awards: Nominations Open

Go to www.xml-journal.com today and nominate your favorite XML products, books, and services for the 2002 ***XML-J* Readers' Choice Awards**.

Nominations may be submitted through June 30. Voting will begin on July 1 and the winners will be announced in October. Products that receive the most online votes in each category will be declared the winners.

Last year more than 5,000 *XML-J* readers cast their votes to select the best products of the year from over 135 nominations. Thirteen winners and 39 finalists were recognized.

XML-Journal.com

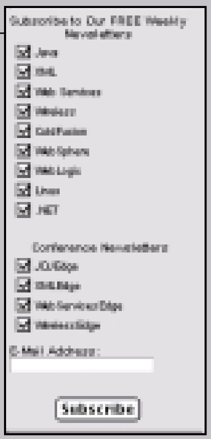
Visit www.xml-journal.com for the latest industry news and events from the world's leading XML resource. Get a leg up in your own job by learning what's happening in the IT world.

Come to Our Archives

Did the dog eat your first issue of ***XML-J***? Did you throw out last July by mistake? All is not lost. Rediscover the past by going to the complete ***XML-J*** archives. Access is **FREE** with your paid subscription. Become the most informed IT professional in the business by reading www.xml-journal.com.

Free Weekly Newsletters

SYS-CON's industry newsletters are informative, convenient, up to the minute – and absolutely **FREE**. Just look in your e-mail inbox every week. Learn what's hot and what's not at your leisure, because these newsletters can be read anywhere you have access to a computer. Subscribe to ***XML-J* Industry Newsletter**, and check out the other **SYS-CON**-family newsletters while you're at it for news on Java, Web services, wireless, Linux, and ColdFusion. Sign up at www.sys-con.com!



Subscribe to Our FREE Weekly Newsletters

☑ Java
☑ XML
☑ Web Services
☑ Wireless
☑ ColdFusion
☑ WebSphere
☑ WebLogic
☑ Linux
☑ .NET

Conference Newsletters

☑ JDJEdge
☑ XMLEdge
☑ Web ServicesEdge
☑ WirelessEdge

E-Mail Address:

Search XML Jobs

XML-Journal is proud to offer our employment portal for IT professionals. Get direct access to the best companies in the nation. Learn about the "hidden job market" and how you can find it. If you're an IT professional curious about the job market, this is the site to visit!

Simply type in the keyword, job title, and location and get instant results. You can search by salary, company, or industry.

Need more help? Our experts can assist you with retirement planning, putting together a résumé, immigration issues, and more.

What's *Your* Opinion

***XML-Journal* Online** offers you not only the opportunity to read our articles online, but to tell us what you think of them. What did you like? What made you mad? What would you like to see in the magazine? You can even e-mail our authors directly with your questions.

Note: Some feedback may be selected for our print edition.

XML-J ADVERTISERINDEX

ADVERTISER	URL	PHONE	PAGE
Altova	www.altova.com		64
ADOS Co., Ltd.	http://www.a-dos.com	81-3-5475-1551	63
Altova	www.xmlspy.com		64
Dynamic Buyer	www.ibm.com/smallbusiness/dynamicbuyer	800-426-7235 ext. 5014	43
engindata Research	www.engindata.com		36, 37
eXcelon	www.exln.com/goNative	800-962-9620	9
iWay Software	www.iwaysoftware.com/guaranteed	866-297-4929	6
Java Developer's Journal	www.sys-con.com	800-513-7111	46
JDJ Store	www.jdjstore.com	888-303-JAVA	54, 55
Macromedia	www.macromedia.com/downloads	888-939-2545	11
Simplex Knowledge	www.skc.com	845-620-3700	27
Sonic Software Corporation	www.sonicsoftware.com		2, 3
Sprint	http://developer.sprintpcs.com		4
SYS-CON Events	www.sys-con.com	201-802-3069	25
SYS-CON Events	www.sys-con.com/education	201-802-3069	32, 33
SYS-CON Media	www.sys-con.com	888-303-5282	15
SYS-CON Reprints	www.sys-con.com	201-802-3026	41
Web Services Journal	www.sys-con.com	888-303-5282	49
WebLogic Developer's Journal	www.weblogicdevelopersjournal.com	888-303-5282	35
WebSphere Developer's Journal	www.sys-con.com/websphere	888-303-5282	41
Wireless Business & Technology	www.sys-con.com/wireless	888-303-5282	47
XML-Journal	www.sys-con.com	888-303-5282	45

General Conditions: The Publisher reserves the right to refuse any advertising not meeting the standards that are set to protect the high editorial quality of ***XML-Journal***. All advertising is subject to approval by the Publisher. The Publisher assumes no liability for any costs or damages incurred if for any reason the Publisher fails to publish an advertisement. In no event shall the Publisher be liable for any costs or damages in excess of the cost of the advertisement as a result of a mistake in the advertisement or for any other reason. The Advertiser is fully responsible for all financial liability and terms of the contract executed by the agents or agencies who are acting on behalf of the Advertiser. Conditions set in this document (except the rates) are subject to change by the Publisher without notice. No conditions other than those set forth in this "General Conditions Document" shall be binding upon the Publisher. Advertisers (and their agencies) are fully responsible for the content of their advertisements printed in ***XML-Journal***. Advertisements are to be printed at the discretion of the Publisher. This discretion includes the positioning of the advertisement, except for "preferred positions" described in the rate table. Cancellations and changes to advertisements must be made in writing before the closing date. "Publisher" in this "General Conditions Document" refers to SYS-CON Publications, Inc.

www.wbt2.com
SUBSCRIBE NOW
www.javadevelopersjournal.com
TO THE
www.sys-con.com/xml
FINEST
www.coldfusionjournal.com
TECHNICAL
www.sys-con.com/pbdj
JOURNALS
www.webspheredevelopersjournal.com
IN THE
www.wldj.com
INDUSTRY!
www.wsj2.com
subscribe online www.sys-con.com or call 800 513-7111
SYS-CON MEDIA
wireless | java | xml | coldfusion | powerbuilder | websphere | weblogic | web services

Next Month...

Emerging Standards for XML-Based Integration
Will we ever realize a "universal business language" in which all participants define transactions in a similar manner?
by John Evdemon

Ultimate Device Support with CC/PP
Make Web applications work better, perform advanced device customization, handle new devices automatically, and more
by Erich Izdepski

XQuery – An Expression Language for XML Data
XQuery is designed for processing XML data – not just files in XML format, but other data (including databases) whose structure is similar to XML.
by Per Bothner

ebXML Evolution
The role of ebXML Registry in e-business framework
by Chaemee Kim

State of the Standards
The industry standards in XML and the consortia that create them
by Mark Spears

XML JOURNAL The Ultimate XML Enterprise Resource

Don't miss the July issue!

HOME
Enterprise Solutions
Content Management
Data Management
XML Labs

XML NEWS

(*San Francisco*) – Oracle presents Web Services Education Session to packed house at Fisherman's Wharf Marriott Grand Ballroom.

Web Services Edge World Tour Attendance Quadruples in San Francisco

(Montvale, NJ) – On May 13 SYS-CON Events, Inc., presented the fourth stop of their Web Services Edge World Tour, SYS-CON's popular one-day tutorial series, to a sold-out audience in San Francisco. The series was organized in response to developers' eagerness to plan ahead for SYS-CON's Web Services Edge 2002 International Conference & Expos East and West, which will take place June 24–27 in New York City and October 1–3 in San Jose, CA, respectively.

www.sys-con.com/WebServicesEdge2002East

News from the World Wide Web Consortium

(Cambridge, MA) – W3C has issued the Platform for Privacy Preferences (P3P) 1.0 as a W3C Recommendation, representing cross-industry agreement on an XML-based language for expressing Web site privacy policies.

P3P 1.0 provides a standard, simple, automated way for users to gain more control over the use of personal information on Web sites they visit. At its most basic level, P3P is a standardized set of multiple-choice questions covering the major aspects of a Web site's privacy policies. P3P-enabled browsers can "read" this snapshot automatically and compare it to the consumer's own set of privacy preferences.

• • •

In other news, W3C has issued Scalable Vector Graphics (SVG) 1.1 and Mobile SVG Profiles as W3C Candidate Recommendations.

Advancement of the documents to this status is a statement that the specifications are stable and an invitation to the Web development community at large to make further implementations of SVG and provide technical feedback. SVG 1.1 separates SVG capabilities into reusable building blocks, and SVG Mobile recombines them in a way optimized for mobile devices.

www.w3.org

New Content Intelligence Services from Documentum

(Pleasanton, CA) – Documentum has announced the availability of Documentum Content Intelligence Services. CIS, an extension to the Documentum platform, automates and controls the enrichment and organization of enterprise content. Documentum CIS streamlines the organization, indexing, and linking of content residing in multiple disparate systems throughout an enterprise, enabling fast, sophisticated searches, easy navigation, and personalization of content.

Key features include auto tagging, auto categorization, auto content linking, and prebuilt departmental and industry taxonomy libraries.

www.documentum.com

CatalystCommand 9.0 Released

(Milwaukee) – Catalyst International Inc. has a new version of its CatalystCommand product, an open technology solution featuring browser independence and new functionality including event management, advanced standard material-handling interfaces, conformance to key industry standards, and enhanced collaborative commerce functionality.

www.catalystwms.com

Formation Systems Introduces New Platform

(Southborough, MA) – Optiva 4.0, a new Web-based Product Lifecycle Management (PLM) platform, is now available from Formation Systems.

The platform provides organizations with a 360-degree view of the PLM process, enabling departments to work in tandem to expedite product introductions, reduce material costs, and improve overall product quality.

New applications include freestyle application adapters, strategic sourcing, bill of information, process definition, and package definition.

www.formationsystems.com

Software Suite Leverages VoiceXML Standard

(Boston) – SpeechWorks International, Inc., is offering a new suite of OpenSpeech DialogModule software for developing commerce applications that leverage the VoiceXML 2.0 standard. OpenSpeech DialogModules are the first standards-based VoiceXML objects available on the market.

The latest release delivers significant benefits for building speech-enabled commerce applications that allow callers to order a product or catalog, change account information, schedule appointments, open a new account, or provide other commerce-related information over the phone using a natural language dialog.

www.speechworks.com

COAST WebMaster 6.0 Product Released

(Ottawa) – COAST Software Inc. has launched the latest version of its automated Web quality testing software, with enhancements in testing and reporting flexibility along with support for Microsoft .NET/XML Web services testing.

COAST WebMaster 6.0 features support for cookies, JavaScript navigation, JavaServer

webMethods, HP Announce OMI Specifications

(Fairfax, VA / Palo Alto, CA) – webMethods, Inc., and Hewlett-Packard have announced the general availability of the Open Management Interface Specification, version 1.0. An open specification jointly authored by webMethods and HP, OMI defines a standards-based (XML/SOAP/HTTP) management interface. With OMI, customers can use leading systems management tools to manage and monitor business processes, Web services, and the underlying integration infrastructure that drives their businesses.

www.webmethods.com/OMI_Spec_index
www.openview.hp.com

Pages, session IDs, and Macromedia Flash, and integration with Rational ClearQuest.
www.coast.com

PeopleSoft Adds Supply-Chain Apps

(Pleasanton, CA) – PeopleSoft Inc.'s new supply-chain applications include strategic sourcing and trading partner management. Strategic sourcing includes functions such as auctions, reverse auctions, e-procurement, and product returns, as well as functions for ranking the quality of suppliers.

PeopleSoft trading partner management is software for communicating and transacting with suppliers using a variety of technologies, including electronic data interchange, XML, Web services, and fax.

PeopleSoft is also upgrading its portal product with a new intelligent context manager that lets users make business decisions by correlating information – such as customer account information pertaining to various transactions – on an ad hoc basis. It also features more sophisticated support for Web services.
www.peoplesoft.com

TrackBase Offers Technology News Tracking

(New York) – TrackBase's technology news tracking service is now available on a month-to-month subscription basis. The service allows companies to track specific corporate, product, and technology news developments according to their needs. With a one-month setup fee, the service has no cancellation penalties, and its comprehensive flat-fee billing structure allows clients to easily budget for its use.

The service is available online, via custom e-mail delivery or XML feed.
www.trackbase.com

Kamoon Updates Enterprise Management Solution

(Fort Lee, NJ) – Kamoon Inc. is shipping a new version of its Kamoon Connect software, which enables EEM activity via any e-mail-enabled device.

Version 2.6 also integrates XForms for a richer user interface, support of different platforms for authentication and SSO, LDAP/NTLM/Active Directory, and an enhanced role-based modular interface.
www.kamoon.com

Macromedia Offers Integrated Product Family

(San Francisco) – Macromedia, Inc., has announced Macromedia MX, a new integrated family of client, tool, and server technologies for creating rich Internet applications that promise more intuitive, responsive, and effective user experiences across platforms and devices.

The products are designed to leverage the capabilities of Macromedia Flash Player, the widely distributed rich client that is deployed to 98% of Web users.

The Macromedia MX product family includes the Macromedia ColdFusion MX server-scripting environment and a suite of development tools: Macromedia Flash MX, Macromedia Dreamweaver MX, Macromedia Fireworks MX, and Macromedia FreeHand 10, which are available together in Macromedia Studio MX.

Developers can use Macromedia MX to build solutions on the Sun Java platform and the Microsoft .NET Framework. The products have strong support for industry standards such as XML, Web services, and accessibility.

Macromedia MX seminars will be held throughout North America. Check www.macromedia.com/go/q103prseminar for more information.

New XML Schema Importer from Mercator

(Wilton, CT) – Mercator's new importer makes it possible for customers to accelerate their complete XML integration solutions to production. With a single keystroke, customers can import an XML schema, automatically creating a complete graphic model of the structure of an XML document. That model can then be used to define business rules for content-based routing, validation, transformation, and process flows.
www.mercator.com

Companies Launch XML Integration Workbench

(Beverly, MA / New York) – Altova Inc. and XML Global Technologies, Inc., have announced the XML Integration Workbench, a data integration solution for small and mid-sized companies, as well as for larger corporations seeking scalable, enterprise-ready data integration tools.

The product bundles three powerful XML-based tools including GoXML Transform Workbench, a modular solution for integrating structured data; GoXML DB Workbench, a native XML database; and the XML Spy Suite from Altova, a strategic XML Global technology partner.

Features of the XML Integration Workbench include a patented transformation engine with an intuitive graphical interface; EDI dictionaries with multiple EDI formats, such as X12, EDIFACT, HIPAA, and SWIFT; XQuery Support; XML content and stylesheet editing; and the XML Spy e-Forms browser plug-in.
www.xmlglobal.com
www.xmlspy.com

FULCRUM Now Makes the U.S. Scene

(Pune, India) – Software tools company AccelTree has debuted FULCRUM, an intelligent Java code builder, in the United States. Neither an IDE nor a code generator, FULCRUM is a Java development tool that uses a proprietary concept of code templates that can be used as building blocks to construct efficient Java objects and applications.

The individualized templates, which serve as guides or examples of a specific Java function that can be adapted to specific needs, can be overwritten and saved. FULCRUM also allows users to switch to manual coding at any time.

Among the reported benefits are faster coding time and the ability to implement your own coding standards. In addition, the need for skilled programmers is reduced, and no runtime software needs to be purchased. The tool also integrates with existing or external objects.
www.AccelTree.com

Lightspeed Interactive, Altova Partner to Deliver XML Authoring Capability

(Pleasanton, CA / Beverly, MA) – Lightspeed Interactive, Inc., a leading provider of information management and delivery solutions, and Altova, Inc., provider of a leading XML development tool suite, have formally established a strategic partnership. The two companies will collaborate to integrate their technologies so as to provide the industry with an advanced XML environment for creating, managing, and delivering advanced XML-based applications.
www.lspeed.com
www.altova.com

Introducing *XML-Journal* ‘2.O’ – Where Four Hands Make Light Work

A new-look *XML-J* for a new phase of the XML-enabled business cycle

WRITTEN BY **JEREMY GEELAN**

AUTHOR BIO

Jeremy Geelan is editorial director of SYS-CON Media.

The inaugural issue of ***XML-Journal*** was published in the first quarter of the new millennium. Then just two years old, XML already seemed to hold almost unlimited promise, and few seemed to doubt that XML technologies had excellent prospects for the 21st century that lay ahead of us. But unlike most other turn-of-the-century technology, XML has flourished slowly and steadily, including through the dot-com meltdown.

How can it be that, for once, the early promise didn't turn out to be just technohype and folderol? In the same time period wireless technologies (in the enterprise, anyway) have distinctly wilted; yet XML technologies have steadily blossomed. Vertical after vertical has spawned its own subset of XML, and within any given area of business, interoperability has moved from being an optional extra to being the centerpiece of the entire new e-commerce landscape.

Most unusual of all, XML has moved far beyond the technical domain and into the business mainstream, and it's as a reflection of this last development that we bring you a more enterprise-oriented *XML-Journal*. In other words, there'll now be as much in each issue for the business audience as for the application developers.

Application developers, of course, will also benefit from this new widening and deepening of scope. On hearing of our plans, the “father” of XML technologies, Charles F. Goldfarb, commented: “Adding a business perspective is a great idea for *XML-J*. I've long preached that developers need to know why their code is wanted to do the best job.”

Henceforth, then, *XML-J* will cover much more closely the impact that XML technologies are having on companies, and the successes and failures of those that have been implementing them. We'll be aiming at the entire enterprise hierarchy – CEOs, CIOs, CTOs, CFOs, vice presidents, directors, programmers, engineers, and line-of-business managers – all who recognize the need to understand that XML plays a role in how their business systems interact with other systems and with the outside world. By the way, in this issue alone are articles from employees of SAIC, EDS, and Deloitte & Touche, some of the largest consulting organizations in the world.

In line with this development, founding editor **Ajit Sagar**, who we “borrowed” from ***Java Developer's Journal*** to get ***XML-J*** up and running, returns exclusively to ***JDJ***, where he has simultaneously been editing the all-important J2EE section. With this current issue he passes the torch of the world's leading XML print (and online) resource to a new pair of hands.

Well, to two new pairs of hands, in fact.

The new editors-in-chief of ***XML-J*** (plural – there are two of them because in our view the scope of XML has widened and deepened beyond the capacity of any sole editor to encompass alone) are **JP Morgenthal** and **John Evdemon**, longtime members of the editorial advisory board that has helped steer the magazine through its first phase. We welcome them both as the new custodians of the magazine in this next exciting phase, the phase we've nicknamed ***XML-J 2.0***.

Both are well known in the industry. They bring to ***XML-J*** an enviable mix of talents. John has served as CTO/director of XML-related products for both a large integration platform vendor and a small XML-centric startup and has been working with XML since its early beginnings. An invited expert with the W3C XML Core Syntax Working Group, he has chaired several industry-specific XML initiatives and has been involved with both ebXML and Web services since their inception.

JP is known to many as coauthor of *Enterprise Application Integration with XML & Java*, one of the most popular books in the Charles Goldfarb XML Series for Prentice-Hall. An independent IT consultant specializing in Web services/SOAP, XML, Java/J2EE, CRM, and supply-chain management, he is an internationally prominent expert on the design and implementation of distributed systems for the enterprise.

Our new coeditors-in-chief join me and everyone else at **SYS-CON Media** in extending our warm thanks to their predecessor for having the vision to lead ***XML-J*** from its tentative inception to its present robust and healthy state.

In his first editorial, Ajit prophetically called XML “a technology that has revolutionized electronic commerce and enterprise computing and is going to completely refurbish the face of business as we know it today.” The hope for XML then matches the reality now, and ***XML-J*** has become the industry's one-stop shop for all matters XML.

But neither Ajit nor anyone else could have known back then that analyst predictions today would say that by 2005, in the financial services industry alone, XML Web services will be an $8 billion industry. That is a great deal of money, and that's just one vertical.

Beyond a doubt, XML is much more now than the technological plat du jour. It's going to be on the menu of international business for a long, long time. And we hope you'll continue to enjoy ***XML-J***'s coverage for just as long.

JEREMY@SYS-CON.COM