SOA Is Here – Are You Ready for IT?

How loosely coupled applications and their need for stronger governance will impact your IT organization pg.10

The State of Web Services Management Protocols

Examining the common ground of the two main specifications pg.26

Web Integration Architecture Patterns for Enterprise Architects

Approaches to Web enablement of legacy systems pg.32

FOCUS ON THE "A" IN SOA

Without a focus on enterprise architecture, your **SOA may be DOA**

pg 22

THE WORLD'S LEADING MAGAZINE DEDICATED TO WEB SERVICES TECHNOLOGIES • WSJ2.COM

Focus on the "A" in SOA

22

Without a focus on enterprise architecture, your SOA may be DOA

By Charles Stack

SOA Is Here – Are You Ready for IT?

10

How loosely coupled applications and their need for stronger governance will impact your IT organization

By Lance Hill

The State of Web Services Management Protocols

26

Examining the common ground of the two main specifications

By William Vambenepe

Web Integration Architecture Patterns for Enterprise Architects

32

Approaches to Web enablement of legacy systems

By Martyn Hill

BPEL is the SQL of SOA

Get started building next-generation SOA applications with the leading vendor of BPEL technologies

Download BPEL tooling & server software today

activebpel.org/soa

*active*BPEL

SYS-CON MEDIA

Lego My SOA

WRITTEN BY
SEAN RHODY

Just about every publication, marketing brochure, and IT commercial currently on parade for your enjoyment contains some mention of service-oriented architecture. If products were still sold in boxes, instead of downloaded, you would see a bright sticker on the box saying "New and Improved: Now with SOA." Instead of Fuller brushes, today's sales wiz now pushes SOA – it's good for what ails you.

Now don't get me wrong – I'm more or less in favor of SOA, although I do believe there are times when it is inappropriate for use. But after all, this is a magazine devoted to SOA, so as its editor, you can bet I find those inappropriate times few and far between. SOA is a concept that works naturally in a partitioned application world – it allows the natural creation of interfaces and construction of software from basic services. A previous generation of coders might have recognized this as the same mantra used by proponents of component-based development, with increased interoperability being the added bonus of the current SOA strategy. And to an extent they'd be right – SOA is a modern software Lego set. Need security? Grab that piece there. Logging – this little piece over here. Have to connect billing to order management? Get that little connector thingamabob. All in all, it's a very neat, logical concept that resonates well with the logical mind of a programmer or architect.

However the joy of SOA tends to blind architects, programmers, and other techies to the cold, hard facts of SOA: once you get it set up, you still have to build the user interface.

I know, SOA is about computer-to-computer communication, right? At least that's what vendors and consultants would like you to believe. But that's a bunch of hogwash. In the end, SOA is about doing business better. If it doesn't help you do that, what possible justification do you have for changing to it? There is none. In fact, justification of the cost of change is often the first thing that kills a prospective SOA implementation. If it doesn't save money or make money, it's not going to happen in today's economy.

But let's get back to the front, or rather the front end, at least. Service-oriented architecture makes it easy to create composite services, or rather, business processes. Processes can be wired together from prebuilt services in a manner that is inconceivable when looked at from the perspective of 40 years of previous software development. And yet with Web services, some transaction management and a little business process management thrown in for good measure, it's very reasonable to rewire an entire process in a very short period of time.

And then it hits: because the reality is that while some business processes don't require the computer to interact with a human being (or reasonable facsimile), most of them (the business processes, not the human or reasonable facsimile) do. That's the wall, the spot where we run out of Lego.

I'll admit there are good tools out there for creating user interfaces in short periods of time. But I'll also challenge you to get past the HUMAN part of the equation – by this I mean getting users to agree on things like colors, field lengths, screen resolutions, order of presentation, etc., ad nauseam. You can't do it – the user interface always takes longer and costs more. As much as you might want to move quickly, user interface is the most contentious task in software design. You need consensus, and sponsorship and time.

And that's where the speed tends to fall off of SOA. It's great for creating the services, but it has yet to really develop fully towards quickly and easily deploying the processes so that people can use them. Guess we still need a few more pieces of Lego.

■ **About the Author**

Sean Rhody is the editor-in-chief of *Web Services Journal*. He is a respected industry expert and a consultant with a leading consulting services company.

■ ■ ■ sean@sys-con.com

But Will It Work?

One of the biggest barriers to SOA adoption is fear of not meeting the high demands of the runtime environment coupled with the need to provide business agility. As more layers have been introduced by the components of the new technology stacks, the points of failure in distributed application have multiplied. While the IT side of the house is very enthusiastic about the plethora of features provided by technologies typically associated with the SOA stack – object-orientation, process orchestration, Web services, business rules, and so on – the business side of the house is usually hesitant to invest substantially in new territories that may lead to high risk for existing businesses. Service orientation promises to bring business agility, but will it continue to sustain the demands of operating the business? In the requirements phase of SOA, the former is associated with functional requirements and the latter with nonfunctional requirements.

Adoption of SOA holds the promise of enabling businesses to more effectively adapt to change – and to add new offerings to their existing products in a more effective fashion. These offerings are implemented on the technology stack as functional requirements. The other side of the equation that allows service enablement to actually achieve realization is the satisfaction of the nonfunctional requirements – requirements that address aspects of the system that do not directly affect the business functionality. Instead they address aspects of the architecture that are essential for the successful operation of the system and its acceptance by the end users as well as the operations and maintenance staff. The satisfaction of nonfunctional requirements is what determines the success or failure of the system when the rubber meets the road.

Clear expression of nonfunctional requirements by business and their successful implementation by IT is crucial for any project that is undertaking service enablement. After all, the main goal of SOA is to bridge the business-IT divide by leading to the establishment of an agile organization. The nonfunctional requirements for a service-oriented architecture address several aspects of the architecture, including the ability to meet the service levels of the business (Service Level Agreements), the ability to maximally leverage the ever-changing technology stack, the ability to perform to RAS (Reliability, Availability, and Scalability) specifications at runtime, the ability to satisfy the needs of effective life-cycle management

WRITTEN BY
AJIT SAGAR

and maintenance, and the ability to effectively adapt to change (leading to business ability).

The expression and realization of nonfunctional requirements is a challenge because while the system can be built to functional specifications, the expression of many of the nonfunctional specifications is not possible in a very precise manner. For example, during the early stages of the architecture design, it may be possible to identify security requirements, but hard to express what constitutes a system failure in the case of a security breach. And at runtime, how can the millions of lines of logs be interpreted to determine whether security thresholds have been crossed? Capacity planning is another example. Systems that are built to satisfy a certain performance criteria are expected to scale to higher volumes in a certain amount of time. How much buffer should be built in to pad the business requirements to make sure that the requirements expressed on paper are actually the ones that will need to be satisfied six months from now?

Fortunately SOA does, to some extent, provide more formal means of expressing, implementing, and monitoring nonfunctional requirements. It is only with the advent of SOA that SLAs have come to mean the same thing to a business user and to an IT specialist. This is probably the main reason why there is so much excitement around the concept surrounding the "Same Old Architecture."

Good sources of information on addressing aspects of SOA such as those discussed in this article are hard to find. I recently read a text, *Service-Oriented Architecture Compass* by IBM press, which covers some of these aspects in a concise fashion, while at the same time addressing the strategy to transition to SOA. If you are interested in this area, you may want to pick up a copy.

■ About the Author
Ajit Sagar is a principal architect with Infosys Technologies, Ltd., a global consulting and IT services company. Ajit has been working with Java since 1997, and has more than 15 years experience in the IT industry. During this tenure, he has been a programmer, lead architect, director of engineering, and product manager for companies from 15 to 25,000 people in size. Ajit has served as *JDJ*'s J2EE editor, was the founding editor of *XML Journal*, and has been a frequent speaker at SYS-CON's Web Services Edge series of conferences. He has published more than 100 articles.

■ ■ ■ ajitsagar@sys-con.com

SOA Is Here – Are You Ready for IT?

How loosely coupled
applications and their need
for stronger governance
will impact your IT organization

■ While significant attention has been paid to the benefits offered by service-oriented architecture (SOA), which has led to an increased understanding of the challenges that SOA poses as well, far less consideration has been given to the changes that this approach will impart on the IT organization itself. With the discussions around SOA having recently shifted from "if" and "why" to "when" and "how," three important questions now need to be addressed by organizations embarking on an SOA strategy: How will you manage your SOA, how will you pay for your SOA, and how will you staff your SOA?

As most would agree, using existing services within an SOA to develop and support new applications provides IT with the opportunity to take a quantum leap forward with regard to productivity and efficiency. As a result, enterprises can address a variety of process requirements faster and more completely than otherwise possible. However, this expanded reuse of existing assets is predicated on a consistent adherence to common standards,

WRITTEN BY
LANCE HILL

which requires most IT organizations to demonstrate far more discipline around governance than they've delivered to date. In reality, this approach produces faster development cycles that are running headfirst into the greater scrutiny required within an SOA, which significantly reduces margin of error as it eliminates many of the safety nets upon which enterprises have come to rely.

Consequently, Eric Austvold of AMR Research recently wrote [*Service-Oriented Architectures: The Promise and the Challenge* (October 6, 2005)], "SOA will expose the gap between the disciplined and undisciplined IT organization, creating the opportunity for fantastic success and spectacular failure." For example, competing SOA fiefdoms are rising in some organizations. At some point, mass confusion will emerge as these systems are unable to reconcile which "get credit" service is which. Instead of using SOA to streamline their operations, these organizations run the risk of adding further complexity as a new layer of middleware – the super SOA – is now needed to coordinate these various initiatives. The end result is that this "hybrid" approach further limits abstraction, cost effectiveness, and enterprise flexibility. What this means is that the approach to developing, deploying, and managing enterprise applications within an SOA needs to change to secure the promised benefits, and this process entails a variety of significant changes that impact the IT organization.

The Rise of the Shared Service Organization

Most IT organizations are already familiar with the concept of a shared service organization, which is often used to support the "common" assets of the enterprise such as mainframe computing, networks,

The Truth? We had zero visibility into Web Services performance.

Wily can handle the truth. We've helped hundreds of organizations around the world manage critical Web Services and proactively identify problems before customers are affected. Using a single solution, you gain control over both your application infrastructure and the customer experience. To learn more about how Wily can help you achieve customer success with Web Services, visit **truth.wilytech.com**.

Get Wily.™

Enterprise Application Management

wily
technology ™

and the corporate database infrastructure. Because applications are now becoming universal enterprise services, there is a need to increasingly view these individual services as a shared corporate asset as well. As such, the rationale of a shared service model as applied to other asset management requirements begins to make sense here as well.

For example, while many application development and deployment functions will remain closely tied to specific business units or operating groups, there is also an overriding need for the enterprise itself to govern the use of these common assets. As a matter of fact, the effectiveness of these governance efforts will be the key determinant of SOA success. Granted, some of these governance issues are technological in nature and can be solved with centralized registries, automated service monitoring, a common metadata repository, or through the use of an enterprise service bus. However, an even more fundamental need exists to simply define the standards that these technologies will use and to monitor and enforce usage requirements across the asset life cycle. To fulfill this requirement, an SOA Center of Excellence is needed.

Depending on the unique parameters of the organization and its culture, the role of the SOA Center of Excellence can range from light oversight or simple coordination through overriding responsibility for the delivery of services. In any of these scenarios, the fundamental goal should be the elimination of any doubt regarding the appropriate usage of a specific asset, and the SOA Center of Excellence should ultimately deliver the

discipline and coordination needed to ensure efficient and effective operations.

As such, an SOA Center of Excellence should be entrusted with maintaining a single view of available services via a common registry along with their definitions. This organization would also be responsible for the enforcement of specific standards that govern usage such as metadata models, versioning standards, release protocols, and testing procedures.

Beyond just managing these services, the SOA Center of Excellence can also be used to deliver the necessary training and additional development standards needed to ensure a common SOA development methodology as well. The most forward-looking enterprises will even look to this organization to help prioritize long-term technology investments against existing business and IT requirements with a goal of ensuring that the SOA fully supports all of these requirements.

Another important role for the SOA Center of Excellence is helping to overcome the human factors that can potentially limit service reuse. As anyone who has ever run a development shop can attest, many projects are hampered by user concerns regarding the quality or suitability of "third-party" services, or by an unwillingness to make up-front investments that might only benefit those who are able to subsequently reuse the service as a result. In regard to overcoming this grassroots resistance by developers, a variety of "carrot & stick" approaches can and should be employed, and many of these enforcement tools fall under the existing mandate for service governance. With regard to the

carrot, other ways to facilitate greater reuse of existing services include the integration of registry information into the development platform to maximize awareness of available services (this approach is typically supported by other forms of educational outreach). Because the ultimate goal is to create a culture in which service reuse is recognized and appreciated, it's not unreasonable to suggest that organizations tally "reusage" and respond and reward accordingly.

Paying the Piper

Of course, these added development and management steps produce additional up-front costs that the enterprise must address if it is to enjoy the benefit of subsequent reuse. With regard to specific models for addressing development costs, a number of approaches have already begun to emerge. The most simplistic and easy to implement is what I would call the "anti-enterprise" model, in which these additional costs are solely borne by the development group because they're the ones in the most immediate need of the core functionality. The additional cost associated with service enablement simply becomes a mandated requirement for all development efforts. Unfortunately, this approach is often shortsighted because it gives little incentive outside of decree for investing the additional funds needed to ensure widespread reuse of the developed service. As such, organizations are left to pursue the bare minimum as oppose to the optimal.

Likewise, some organizations have taken a "head in the sand" approach that completely ignores the issue of added cost, arguing that service reuse is so new a concept that little data exists for developing a cost model. Therefore, the true cost of service enablement is typically ignored within the overall budget. The challenge that this approach creates is that the IT organization or business group may be subsequently unable to show effective ROI for these projects. Thus, users have an incentive to do the bare minimum possible, including avoiding this requirement altogether.

Arguably, the best approach is to recognize these costs up front because this encourages both accountability and efficiency throughout the development process. For

> " Because applications are now becoming universal enterprise services, there is a need to increasingly view these individual services as a shared corporate asset as well "

example, the added cost for service enablement can be defined as a fixed percentage of the total project cost and these additional costs are fully borne by a dedicated source of enterprise funding. With regard to specific budget parameters, a recent study by the Aberdeen Group offers some guidance. According to the research firm, a $10 billion company with a $300 million annual IT budget can save $30 million a year in five years by service-enabling 75 percent of their applications. As such, a $2 million fund for service enablement would result in a very favorable ROI.

In addition, enterprise budget models also need to address the costs associated with actual usage. For example, who bears the budgetary impact when a service developed by your group is subsequently employed as the cornerstone of another group's business model? For most organizations, the chargeback mechanisms or other activity-based pricing that they already employ become the model to be used for funding these ongoing costs. Specific mechanisms could include shared service units in which costs are closely tied to consumption, tiered service units that make allowance for each group's business objectives and modify pricing accordingly, or an enterprise pool model that relies upon headcount or other non-usage–based metrics. The important point to remember is that these fees are in lieu of additional development costs, and therefore represent significant savings for the business.

One way to ensure the success of these programs is through the use of a common fund for supplemental funding. For example, consumption-based pricing typically reflects a pro-rated portion of the overall cost. Therefore, what happens when projected usage falls 50 percent short – do you double your per-usage costs? Not if you hope to encourage long-term support for these services by various business units. As such, the shortfall could be addressed through the use of an enterprise fund for SOA, which would minimize cost fluctuations that would otherwise deter adoption by covering this gap. Likewise, in cases of over-subscription in which usage and therefore fees are higher than projected, this overage should be returned to this common fund.

IT-Centric, Bottom-up	IT-Centric, Top-down	LOB Integrated
1). Business/IT Strategy	1). Business/IT Strategy	1). Business/IT Strategy
2). Business Development	2). Business Development	2). Enterprise Architecture
3). IT Governance Org.	3). Enterprise Architecture	3). Business Development
4). Enterprise Architecture	4). IT Governance Org.	4). IT Governance Org.
5). Enterprise Integration	5). Enterprise Integration	5). Enterprise Integration
6). Application Delivery	6). Application Delivery	6). Application Delivery

TABLE 1 | The role of the enterprise architect

Prepare to Change Your Relationship with the Business

A key topic of late has been the interrelationship between SOA and business process management (BPM). Depending upon where you sit within the organization – IT or line-of-business – you are probably hearing that BPM is a key component of SOA or vice versa. In reality, both perspectives are accurate because the process of decoupling business logic from the application itself is the basis for an SOA, but it is the building block for a successful BPM strategy as well.

What this means is that your shift to a more process-driven approach to IT is poised to accelerate. For example, because a library of services is now readily available, you can move beyond your need to solve underlying integration and development challenges to focus more extensively on stringing together entire business processes. While this is a potentially overused concept, the real payoff is closer alignment between business and IT in which business requirements can now be more quickly, accurately, and fully addressed by IT.

While this closer collaboration with the line-of-business is obviously a positive development, it also needs to be handled with care. Specifically, shorter development times make it harder to shield the business from the real-world hiccups that plague all development efforts because the potential to make up time no longer exists. Likewise, this tighter relationship also creates the potential for more ad hoc development, which can quickly expand the scope of projects to such

a degree that timeline and other objectives are no longer being met consistently.

As such, IT must strive to create even stronger bonds and understanding of its line-of-business counterparts. Specifically, with the line-of-business more directly embedded in the development process, they need to secure a better understanding of what's possible in the near term and what's needed in the long term so that projects can be undertaken in the most efficient and effective manner possible. All people within the IT organization must be prepared to evolve their role beyond simply fulfilling requests on a reactive basis so that they can become relationship managers who support and enable a collaborative partnership with the line-of-business.

Who Owns the SOA?

Over time, a fairly consistent model for the application development team has emerged. The following roles are most common:

- **Business/IT Strategy:** Nominally, the CIO or other senior official who is charged with defining long-term IT strategy in alignment with specific business objectives
- **Business Development:** Typically, a business analyst or similar role that helps to define the specific business parameters and reporting outputs of an application
- **IT Organizational Governance:** Most frequently, a centralized group that creates and enforces both standards and processes that govern application usage
- **Enterprise Architecture:** The architect

> " The real payoff is closer alignment between business and IT in which business requirements can now be more quickly, accurately, and fully addressed by IT "

is responsible for creating the necessary framework for delivering asset functionality with a specific focus on minimizing implementation, runtime, and evolution costs along with complexity, downtime, and technology risk

- **Enterprise Integration:** Project teams that support and enable business processes by connecting together various applications and systems
- **Application Delivery:** Developers that create and implement both packaged applications and customized development

In this hierarchy, the enterprise architect most often serves as the "wild card." Table 1 shows this role and how its importance varies in a variety of scenarios.

In the first scenario, which often represents a highly decentralized organization, the enterprise architects typically serve as "technicians" tasked with supporting the underlying resource teams as they add new functionality to the existing enterprise architecture. As these organizations evolve toward a more centralized and hierarchical approach – the top-down model – the architect often assumes greater responsibility for enterprise governance and standards and the existing IT governance organization becomes more of an enforcement arm as a result. However, even in this model, they fail to assume significant responsibility for the business success of these IT initiatives.

With the rise of SOA, the line-of-business alignment model is most likely to dominate due to the architect's fundamental role in ensuring that services are structured to encourage the most widespread reuse possible. In many ways, the abstraction of business logic

as a service means that architects begin to fulfill many of the functions of the business analyst, which enables their ascension up the stack. For many organizations, this is a fairly progressive model in which the architects play a role that is fully integrated with the business objectives of the enterprise because they are charged with proactively creating the most efficient and effective model for addressing these requirements over the long term.

With this evolution, one can expect SOA adoption to specifically impact each of these roles as follows:

- **Business/IT Strategy:** Managing ongoing relationships with the line-of-business becomes even more critical as the boundary between these two disciplines begins to disappear. Coupled with the shift toward more process-driven approaches to IT, IT leaders need to develop an even deeper understanding of the business so they can proactively plan and provision for requirements that are currently too often met on an ad hoc basis. They should also strive to push this philosophy downstream, eliminating the silos that exist within the IT organization and ultimately ensuring that their staff can be more closely integrated with business requirements.
- **Enterprise Architecture:** As noted earlier, the fundamental role that the architect plays in enabling an SOA means greater accountability because they're now playing a more direct and hands-on role in the successful delivery of services. As a result, instead of a technology-first mindset, they'll be increasingly called upon to take a business-first approach. At the same time, they'll also need to play a greater

operational role as architecture decisions become real time and thus, have a more direct and immediate impact on the business.

- **Business Development:** In addition to their existing responsibilities for defining requirements and analyzing results, these business analysts are now poised to become process architects entrusted with designing, implementing, and monitoring these processes. With this change comes the need for an even greater understanding of business semantics as a requirement for defining process. Their analytical skills will also be increasingly tested by the expanded use of real-time monitoring within an SOA environment, and strict adherence to process improvement disciplines around their use of Key Performance Indicators (KPIs) are an absolute must.
- **IT Organizational Governance:** This role will have the greatest day-to-day impact in determining the scalability of the SOA strategy since they're responsible for enforcing discipline around governance. As such, a relatively minor organization today is likely to increase in importance within an SOA due to their ownership of the underlying registry. That said, they also face the risk of being subsumed by an expanded office of the chief architect. In either case, one of the key roles that they're likely to play is akin to a traffic cop as they prioritize the implementation of various projects in accordance with line-of-business requirements. Because the invocation of services is fundamental to business process management, they'll also be called upon to develop a greater understanding of process parameters with regard to how they impact specific asset management, change management, and configuration management requirements.
- **Enterprise Integration:** Based on their experience in managing Integration Competency Centers, which is one model for an SOA Center of Excellence, these project teams have the potential to expand their influence throughout the enterprise by assuming this complementary function. Of course, this is predicated upon their ability to move beyond a project-based mentality to embrace a more long-term and strategic

perspective in which they serve as the protector of these critical assets. What this entails is an even greater focus on Web services management because this brings forth the required discipline needed to best manage the entire asset life cycle.

- *Application Delivery:* SOA heralds the further industrialization of IT, which challenges these traditional craftsmen. Successful developers will need to take a broader view of the business and look to create and implement applications that address both immediate, tactical requirements as well as more long-term, strategic objectives. These efforts will see their greatest impact within the expanded use of metadata to capture key information, such as policy requirements, service definition, and the abstraction of business rules, within a reusable and independent business container.

The final point to consider is how SOA adoption will impact the overall development methodologies employed throughout the IT organization. Among the key approaches currently being used is the Waterfall methodology, which operates under the premise that changes in development are expensive and disruptive. As such, each project and every application is architected to the "nth" degree to avoid such occurrences. Conversely, the Agile development model recognizes the impact of change, but also understands that it's frequently unavoidable. Therefore, a more modular approach is employed that limits the impact of any individual change, while allowing the organization to benefit from faster development times.

For organizations employing the Waterfall methodology, one of the key challenges that they'll face is that extensive, real-world testing is often delayed until the end of the development cycle. In the more dynamic environment enabled by an SOA, they run the risk of failing to keep pace with changes in key services. Therefore, by the time they're ready to test a particular solution, one of the underlying services may have changed in a number of fundamental ways, leaving the user with a composite application that fails to function as designed. At the same time, the broader shift toward faster development also challenges their ability to keep pace with these escalating demands due to the proclivity for deeper front-loaded design and document work inherent to this methodology.

However, Agile development shops face their own challenges. Principally, this "just-in-time" approach potentially inhibits long-term planning. As a result, reuse can be compromised as little opportunity is given to consider each service's broader, longer-term requirements. In addition, many Agile development organizations have difficulty creating and maintaining long-term documentation regarding applications and assets – a critical need for reuse of services between project teams. That said, their practice of iterative testing serves as a great complement to SOA because it helps to ensure interoperability by enforcing more consistent governance standards.

SOA Means Change – Accept It, Embrace It, and Profit from It

As an IT strategy, SOA offers great promise for helping to overcome some of the most systematic challenges facing IT, and a number of organizations are beginning to show real results. Of course, these benefits do not come without cost, and therefore you need to be prepared to change some existing approaches to accommodate this new strategy.

Arguably, one of the key benefits offered by SOA is the ability to experiment more broadly as feedback can now be more easily captured and incorporated into subsequent versions of a composite application and this entire process can now be rendered in hours or days as opposed to months thanks to SOA. You need to take a similar approach to managing, supporting, and staffing your SOA environment in which you're open to experimentation, but committed to testing to drive the emergence of best practices.

As evinced by such trends as offshore outsourcing, it's also important to note that existing competencies can quickly be commoditized. Regardless of their role, SOA provides IT professionals with the means with which to play a more strategic role within the enterprise, and more important, one in which their unique domain knowledge and understanding of core business processes cannot be easily replicate. This is probably the most important lesson to be learned regarding the transition to SOA, which is the need to develop an even deeper understanding of the business objectives in order to thrive in this environment. ©

■ About the Author

Lance Hill is the vice president of webMethods' product and solution marketing, where he leads a number of strategic initiatives focused on the development, commercialization, and adoption of webMethods' SOA-based technology.

■ ■ ■ lance.hill@webmethods.com

XMetaL Author 4.6 Service Pack 2 DITA Edition

Document preparation just became a whole lot easier

■ Documentation is generally the final step in the software development cycle and while it is a very important component of the entire software package, last-minute changes in software can lead to major alterations in documentation. The inability to quickly incorporate such amendments in software into documentation has made it difficult for technical writers to author and produce quality deliverables applicable to all audiences without delaying the software production cycle. Additionally, ensuring consistency in documentation published across a wide range of media can lead to further delays. XMetaL Author 4.6 Service Pack 2 DITA Edition (XMetaL Author DITA) by Blast Radius does a successful job of alleviating this very problem.

Open source DITA (Darwin Information Typing Architecture) has become the standard for producing technical documentation. Rather than producing a document in a traditional fashion with chapters and pages, DITA individually classifies all of the components in a document, thereby allowing each element to be reusable, customizable, and easily adopted throughout the rest of the document.

Blast Radius has now taken the DITA technical standard and applied it their already popular XMetaL Author. While XMetaL Author DITA is essentially a DTD at its core, on the surface it offers the simple functionality of a basic word processor, thus allowing anyone from amateurs to advanced technical writ-

WRITTEN BY
ROHIT SINGLA

ers to have the ability to create or edit complex documents in XML or SGML. Through DITA the author also has the ability to set predefined DTD or XML Schema rules to allow many users to contribute to a document while ensuring consistency and validation throughout the document.

Content Reuse

As I mentioned, one of the major benefits of DITA is the ability to ensure reusability and consistency of various document components. The most common types of content reuse in XMetaL Author DITA involve topics and content references. Topics are short paragraphs related to a single subject while content references involve specific details that are outlined within each topic. During the preparation of

a technical document in XMetaL Author DITA each topic and content reference is tagged, thereby allowing easy reference throughout the document. A built-in map editor in XMetaL Author DITA allows these tags to then be dragged and dropped in an Explorer-like structure to quickly move content around – all while maintaining consistency with the document based upon the established DTD or XML Schema rules.

DITA Map Creation and Editing

The DITA map creation in XMetaL Author DITA enables users to lay down the foundation for various documents to be created. Figure 1 shows an example of the mapping capabilities of XMetaL Author DITA. In the Resource Manager, subtopics related to changing a printer cartridge are listed, such as preparing the printer to change a cartridge, removing a cartridge, as well as installing a cartridge. Each subtopic is then broken down into individual components as illustrated in the tree diagram to the right of the Resource Manager in Figure 1. Topics can be easily dragged and dropped using an Explorer-like tree diagram. Once complete, this DITA map can be saved and recalled as a template for other documents such as training manuals, knowledge bases, promotional material, or even online help.

Aside from effortlessly moving content around within the map editor, XMetaL Author DITA allows content references from other sources to be added to any documentation. For example, if an individual in the Marketing de-

> " Document preparation will definitely be less of an onerous task with the use of this new product "

partment wished to add technical specifications about a printer to an advertisement, they would be able to simply access a list of reusable content from the Product Development department and simply drag and drop references within their own map, thus quickly adding the content to their advertisement. References throughout the document ensure that any updates made by Product Development will be continually updated correctly in the advertisement produced by Marketing. XMetaL Author DITA's smart insert capabilities allow content to be easily inserted into a document without requiring users to concern themselves with the document structure and more on content.

Publication

XMetaL Author DITA allows documents to be edited in a plain-text XML environment, publication mode with tags displayed, or a normal mode similar to that of a standard word processor. This functionality adds to its user friendliness. While providing all of the traditional formatting options available in standard word processing packages, XMetaL Author DITA also enables deliverables to be published in multiple formats such as HTML and PDF, all the while saving the user time. Other timesaving features of XMetaL Author DITA are:

- *Conditional text formatting:* Depending on the audience or specific publication, cap-
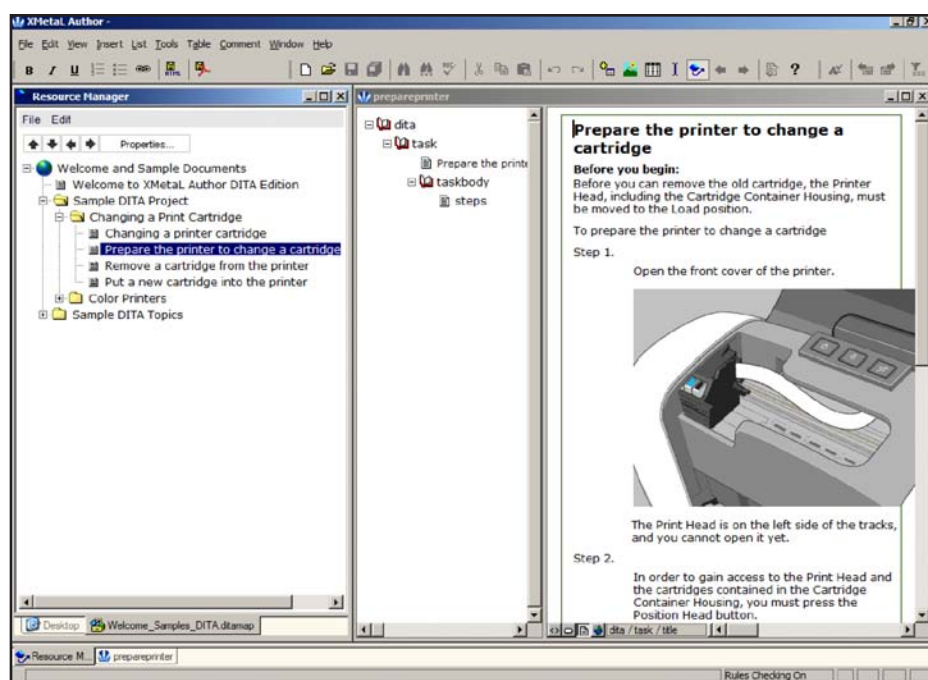
FIGURE 1 | **XMetaL Author DITA's mapping abilities**

tions, headings, or other text can be formatted with specific attributes based upon the XML Schema or DTD.

- *Automatic table of contents and indexes:* As content is added or edited in a document, the table of contents and indexes are automatically generated, which adds to time saved in document publication.

Summary

Blast Radius's decision to extend their XMetaL Author product with the DITA expansion was a great idea. Because DITA has become the new standard for XML document preparation, XMetaL Author DITA should prove to successfully meet the needs of both beginners as well as experienced technical writers. Several users can now easily collaborate on content as well as quickly develop their own customized deliverables, and they can all the while be confident that their documents will be consistent and valid. Document preparation will definitely be less of an onerous task with the use of this new product. A 30-day trial version of XMetaL Author DITA is available online. At the time of this article's publication, a full version of the program can be purchased from the developer online for $895.00 USD. For more information visit www.xmetal.com ℮

■ **About the Author**

As a consultant with Envision Consulting Group, a Unit of IMS Health, Rohit Singla delivers business solutions to many of the world's leading pharmaceutical firms in the areas of government and commercial contracting and compliance.

■ ■ ■ rsingla@gmail.com

> " The inability to quickly incorporate such amendments in software into documentation has made it difficult for technical writers to author and produce quality deliverables applicable to all audiences without delaying the software production cycle "

Will SOA Reduce the Need for Developers?

If you think SOAs will reduce the need for developers, you're dead wrong

■ There is a lot of talk about how SOA will significantly lower the need for developers, thus the savings of SOA. This will be accomplished through the promise of reuse that's driving many toward the SOA light. However, I'm not sure we'll see a reduction in development with the advent of SOA, but perhaps rather a redistribution of talent in the longer term. At the end of the day, the reason for leveraging SOA is agility. Reuse and development savings are a secondary benefit, if they happen at all.

Truth be told, we've been considering the demise of the developer during many "hype phases" over the last 15 years. This included the "component development" phase where I heard not one, but three software executives in keynote speeches talk about how "applications would be assembled like Ford assembles cars, from prebuilt component parts," and thus, the need for fewer developers. The same goes for the distributed object phase, the intranet phase, and now here we are in the SOA phase. The issues are exactly the same, with perhaps the technology being a bit more compelling.

SOA, with all its rich, chewy goodness, has three realities to consider. First, it's something that really has not happened yet; people are talking about it, and in some instances, playing around with it, but true killer SOAs are few and far between right now. This is due to the fact that it's complex, a huge change in thinking, and those things take years to roll out in most enterprises. It's more about people issues than

WRITTEN BY
DAVID LINTHICUM

technology, by the way. Thus, it's too soon to understand what real savings will be realized from the use of SOAs. In other words, it's a bit early to think about how many developers we can fire.

Second, if history is a teacher, we'll find that we actually need more developers – at least at first – with the promise of savings through reuse in the future. However, we've yet to get reuse right with all of the past opportunities such as object-oriented development, distributed objects, and component-based programming, so we're assuming we'll get it right with this technology, standards, and approaches. I'm optimistic, but I'm also a realist here, understanding that true adoption runs about two years behind the hype.

Finally, the use of services over the Internet will create a new generation of developers who build services for applications they'll never see. They build portions of applications for use in many applications as services, typically delivered over the Web, and that industry will be huge. All you need to do is to look at the growth of the major service providers out there and the emerging Web services marketplaces. So, you guys who get fired by the enterprises will have better jobs in these emerging areas.

We're building SOA for many different reasons, including the savings on the development costs, but the primary focus of our SOAs should be on the notion of agility. The end result should be an architecture that's able to change with the needs of the business, and the more your business changes, the more value SOA brings to you. Not to beat a dead horse here, but that's the prize, and where SOA will make its real money for you.

The reduction in development costs will occur at the enterprise levels, but only after SOAs have been implemented and are systemic to the enterprise. This will take some time to accomplish with most businesses – years for many – before you can actually see development costs go down. Indeed, in the short term, development costs will go up.

In the future, more and more development will be occurring outside the enterprise, for consumption by the enterprise. This paradigm will provide even more cost savings, but the need for talented developers will always be there. These developers will be working on other things: service development, and perhaps for other companies, service providers and Web services marketplaces – making more money, I'm sure… that's a win/win as far as I'm concerned. ℮

■ **About the Author**

David S. Linthicum (www.davidlinthicum.com) is the author of three books on application integration and SOA, a frequent speaker at industry conferences, and the host of the "Service-Oriented Architecture Expert Podcast" (www.soaexpertpodcast.com).
■ ■ ■ linthicum@att.net

> " SOA, with all its rich, chewy goodness, has three realities to consider "

Focus on the "A" in SOA

SOA

Without a focus on enterprise architecture, your SOA may be DOA

■ For decades, IT professionals have put up with the headaches of managing a complex and rigid enterprise architecture that has become a Petri dish for the too-familiar misalignment between IT and the business. Enter service-oriented architecture (SOA), which promises to create applications composed of modular software components that are interconnected through well-defined, open Web service standards. Just as companies moved from mainframe to client-server, so must IT move from monolithic applications to a matrix of loosely coupled Web services that enable the composition and recomposition of business processes.

However while many in IT are familiar with and more than eager to realize these benefits, few understand that getting the big SOA payoff requires a fundamental shift in the design, deployment, management, and governance of enterprise architecture.

WRITTEN BY
CHARLES STACK

Architecture: More Than a Document

Typically, an enterprise IT architecture amounts to little more than a static document that describes an equally static environment. This document is often created in isolation, without significant input from business managers, and usually exists as inert PowerPoint slides or Word pages that development groups rarely reference as they create new applications.

And frankly, why should they? Putting such a document in the hands of development teams as a means of communicating and enforcing enterprise architecture is a bit like asking a construction crew to use a single exterior photograph as a blueprint for an office tower. The finished product may look exactly like the photo, but when those top-floor elevator doors open, can you be sure you're not stepping into an empty shaft?

Even in its most streamlined, efficient state, enterprise architecture is far too complex – let alone important – to be expressed in a collection of documents that no one will read, and that are severely limited in their ability to affect how things get built. Can a set of static documents tucked away in a file somewhere offer any assurance that an organization has anything resembling a true enterprise architecture? Yet how many organizations in exactly that situation are already involved in plans to implement an SOA? How much head scratching goes on at these organizations because their SOA efforts appear to have stepped into an empty elevator shaft?

This doesn't mean that these organizations don't place a high value on enterprise architecture. Indeed, few organizations will be given *carte blanche* to revamp the *entire* enterprise architecture around an SOA, regardless of the promised flexibility and agility benefits. Rather than a massive, expensive network overhaul, *the SOA rollout will happen incrementally, within the context of funded projects*. As the business side requests new applications and

new functionality, Web services can be created and reused. In this manner the SOA will emerge over time.

However whether the SOA emerges as a highly organized, loosely coupled collection of services that can be remixed to suit changing business needs, or as a future legacy nightmare, depends on how the SOA integrates with the existing enterprise architecture and its ability to respond to business challenges.

The failure to focus on architecture during this process will ultimately dilute the architecture beyond its ability to have positive impact on the business. It is critical, especially with just such a gradual architectural rollout, that the architecture design exists as more than a static document. It must be a *living* entity that evolves in coordination with changing business needs. This makes the service-oriented enterprise architecture a perpetually moving target. SOA is about anticipating the future, which is as much about where architecture needs to be as it is about where it is today. That is all the more reason to enact measures to ensure that all concerned parties keep that target architecture in their sights at every step.

Balancing Act: Control and Chaos

Effective enterprise architecture requires a balance between control and chaos. It must be dynamic, interactive, and holistic. It must overcome project myopia, spanning space by connecting projects to each other, and spanning time by ensuring that projects are fully aware of what has already been built, what needs to be built now, and what will be built in the future.

The funded project model for SOA deployment underscores the need for architects to confer on an ongoing basis with business managers in order to understand their priorities, model the architecture to support critical business needs, and clearly illustrate how technical assets and services directly affect the ability of business managers to do their jobs

FIGURE 1 | **SOA and Web services are interconnected with a multitude of assets in the IT ecosystem. Managing these relationships and interdependencies is key to the success of your SOA.**

more effectively.

- Architects will have to translate their plans into the bottom-line language of time and money that is spoken by business managers with little technical expertise. Toward that end, dynamic graphical mapping of project and service dependencies will bring the business implications of IT projects vividly to life.
- Only a robust enterprise registry/repository that links the management of services, business processes, and other software assets with business objectives will provide

the kind of long-term impact metrics that IT must supply to guide the business-side decisions that will shape the enterprise architecture (see Figure 1).

Creating the target architecture is only half the battle though. An architecture that no one implements is no architecture at all. Ensuring that developers adhere to the target architecture is absolutely critical to a successful SOA. In this effort, the model for application development in isolation is a recipe for disaster.

Bridging the Silos

Organizations that are adopting SOA must systematically bridge the gaps that separate development silos: those groups of programmers working on projects in almost total isolation, who often insist on building everything from scratch. In the past, the effect of this not-invented-here disease – a malady that afflicts

> " Isolated silos are to SOA what kryptonite is to Superman "

so many corporate development groups – was limited to a severe swelling in the amount of time and money a company needed to spend on any given application. For an SOA, the consequences are life threatening: isolated silos are to SOA what kryptonite is to Superman.

Web services are inherently interdependent, and connect with middleware components, legacy systems, e-business applications, and other software assets in the IT environment. Development teams in an enterprise environment are no less interdependent. Unless these teams communicate and collaborate effectively, the out-of-control cost and complexity of the pre-SOA enterprise infrastructure will become a fond memory to those charged with the responsibility of managing and maintaining the service-oriented enterprise architecture.

In such an environment, what is there to prevent the complexity that is the result of the development of duplicate services? What is there to guard against the disastrous potential of changes to existing services, or to other assets and areas of the business on which those services may depend? There is more to SOA, after all, than the services themselves.

Effective SOA requires the ability to govern the implementation of services from project to project, across silos, and forward into time.

Once silos are truly connected, IT management and development project leads must establish governance practices and systems that require project teams to follow the architecture, adhere to standards, and ensure that services are in compliance with the SOA. Publishing the architecture in a central services registry is a good start, but IT management can go much farther, especially if it has deployed a robust SOA registry/repository.

No "A" in UDDI

Without a central registry/repository that discovers, identifies, and maps services for reuse – one that understands, illustrates, and manages the relationships between services, policies, applications, components, and all of the other software assets in the organization's overall architecture – services will inevitably end up being developed in isolation and misaligned with architecture, despite the best efforts of IT management to connect the silos.

The UDDI standard is useful for the run-time discovery and management of services and for the support of interoperability, but offers little to enforce and govern the overall enterprise architecture. Beyond UDDI, it is essential to understand the interdependencies between services, architecture and the other software assets in the enterprise portfolio. Without these capabilities, there is no chance of eliminating the project myopia that breeds silos, and no way to present the time- and distance-spanning view of the entire enterprise architecture. In the effort to put the "A" in SOA, it is important to recognize that there is no "A" in UDDI.

Head Start: Manage XSD

A critical but often overlooked element of managing and governing the service-oriented enterprise architecture is the institution of early control of the XSD files that represent corporate data. A service-oriented architecture can easily achieve a level of complexity at which XSD files become nearly impossible to untangle. Client-server projects faced much the same problem because IT departments failed to control database schemas from the beginning. Early efforts to manage XSD files will smooth the road to SOA success.

SOA Reality

As yesterday's monolithic IT infrastructure is deconstructed into a dynamic matrix of loosely coupled services, the realization of the dream of the agile, on-demand enterprise inches ever closer to reality. The most important factor in that transformation is that SOA exists within the larger enterprise architecture. The success of the SOA effort depends entirely on the manner in which the enterprise architecture is expressed, communicated, deployed, and governed. ⓔ

■ About the Author

Charles Stack is CEO of Flashline, a provider of SOA and software asset portfolio management solutions. Stack has managed software development for over 20 years and is credited with founding the first Internet retail store. He has been honored with several industry awards, including InfoWorld's Top 10 Innovators in e-business.

■ ■ ■ cstack@flashline.com

SOA Best Practices

1. **Put SOA Within EA**
   SOA will fail unless it is developed within the context of an overall architecture.

2. **Establish a Common Services Group**
   Start small if necessary, perhaps with a single service, but this is a critical element of a successful SOA.

3. **Create a Wiki (or something similar)**
   This is a starting point for the accumulation and enterprise-wide sharing of best practices.

4. **Control Your Schemas**
   XSDs represent your corporate data; in the age of XML, managing schemas is critical.

5. **Design for Multiple Uses**
   Ensure service reusability by creating each service for use in at least two use cases or projects.

6. **Design Large, Cohesive, and Decoupled Services**
   A balancing act, to be sure, but it is important to maximize size and cohesion and minimize coupling.

7. **Align SOA with the Business**
   From the outset, use terminology and definitions that are familiar to business users.

8. **Publish and Promote**
   Use an enterprise registry/repository to publish services, promote those that are the most successful, track usage, and measure the benefits.

9. **Establish at Least Two Measurable Goals**
   Set goals early to insure the verifiable, measurable success of the SOA.

10. **Generate Monthly Reports**
    Frequent is better. Establish monthly reports against the goals established in #9.

OMG ANNOUNCES:

SOA, MDA and Web Services Workshop:
Integrating the Enterprise and Beyond

March 27–30, 2006 • Fairfax, VA

Building on the success of last year's event, Object Management Group™ (OMG™) is pleased to host this workshop featuring presentations from architects, developers, vendors, and end-users of SOA/WS systems, many designed and built using the Model Driven Architecture® (MDA®).

Hear industry experts detail best practices at every stage of the application lifecycle — from modeling for requirements gathering and design, through WS and SOA for implementation and deployment, to the latest in security and semantic interoperability. Case studies by early adopters and presentations by tool developers will show how SOA, WS, and MDA combine to build an interoperability environment that extends business automation beyond the firewall to encompass the enterprise's customers and suppliers.

The program also includes educational tutorials on relevant technologies as well as informative panel discussions and two evening receptions. In addition, the final two days feature dynamic technology exhibits and demonstrations.

HOSTED BY:

OMG®
OBJECT MANAGEMENT GROUP

CO-SPONSORED BY:

M1 Global

Highlights of the program include:

■ **KEYNOTE PRESENTATIONS:**
- Thomas Koulopoulos, CEO, Delphi Group
- Luba Cherbakov, IBM Distinguished Engineer

■ **TUTORIALS:**
- Introduction to Web Services (WS) and Service Oriented Architecture (SOA)
- Modeling Business Process for SOA Applications
- Introduction to Unified Modeling Language™ (UML®) and Model Driven Architecture (MDA)
- MDA, WS and SOA Architecture in the Enterprise

■ **SESSIONS:**
- Extending SOA Throughout and Beyond the Enterprise with WS & MDA
- Best Practice in Design and Implementation of WS and SOA
- Tool Chains & Middleware for MDA, WS and SOA Apps
- Security, Scalability, Performance and other Challenges
- Standards in WS, SOA and MDA
- Semantic Aspects and Ontologies
- Case Studies

■ **PANELS:**
- Integrators and Tool Implementers
- Case Studies: Lessons Learned

FOR MORE INFORMATION OR TO REGISTER:
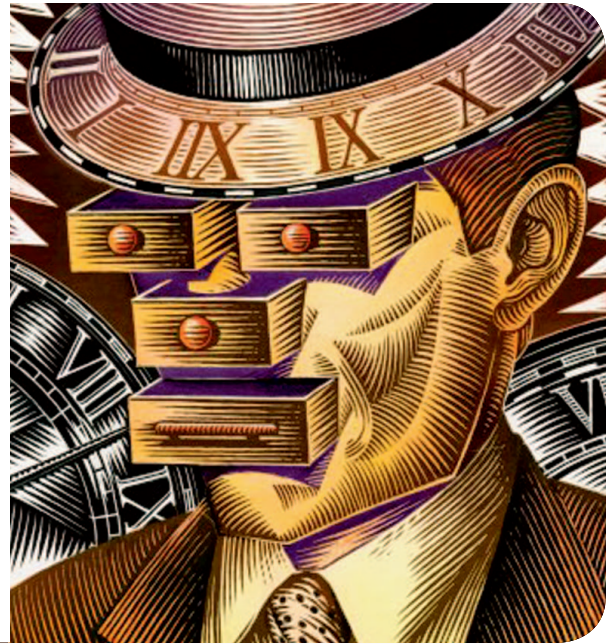
CALL:
+1-781-444-0404

VISIT:
www.omg.org/soa-mda-ws

EMAIL:
loughry@omg.org

The State of Web Services Management Protocols

Examining the common ground of the two main specifications

■ An interesting convergence is taking place in the IT management world, toward Web services–based management protocols. One of the driving factors in this convergence is the effort to improve the agility of enterprise IT, such as HP's Adaptive Enterprise, IBM's On Demand Computing, and Microsoft's DSI.

The convergence also derives from the effort of the Grid community, as seen in Global Grid Forum, where the goal is to build support for large and distributed computing systems. At the lower end of the spectrum, the convergence comes from manufacturers of devices such as printers and phones that acknowledge improved management as a key customer demand. Finally, the traditional IT management community, such as the companies working inside the Distributed Management Task Force (DMTF), is also compelled to use Web services for management.

The reasons for wanting to use a Web services–based standard are multiple, but those most often cited are:

• Interoperability, especially in heterogeneous environments

WRITTEN BY
WILLIAM VAMBENEPE

• Desire to comply with service-oriented architecture (SOA) principles
• Reuse of the large (and increasing) number of Web services tools available
• The need to build efficient, large-scale distributed systems that leverage the increasing number and diversity of nodes on the network
• Protection of investment through the use of industry standards

• Access to the features of other composable Web services specifications in domains such as reliable messaging, security, transactionality, etc.
• Desire to get rid of the need for multiple agents bolted on resources in order to provide manageability for the resource for multiple management products
• Integration of some management functions in business-driven interactions for improved alignment of business and IT

Out of this industry push, two main efforts have emerged. The OASIS WSDM (Web Services Distributed Management) technical committee produced the WSDM MUWS (Management Using Web Services) 1.0 specification as an OASIS standard in March 2005. In August 2005, the WS-Management specification was submitted to the DMTF, which subsequently

> " Those who implement their protocols with versioning in mind will harvest the benefit of their design efforts "

chartered a new subgroup to produce a standard based on the WS-Management submission. WSDM MUWS and WS-Management largely overlap in scope. More important, while the attention is often focused on the differences, there are significant similarities in the two specifications.

Both specifications assume that resources are represented by an XML document and make this document available through SOAP-based mechanisms (allowing the entire document or only parts of it to be retrieved). In addition, both specifications consider that resources are individually addressable and assign to them a WS-Addressing Endpoint Reference (EPR). In doing so, the specifications don't assume that accessing each resource directly is the only way to manage them, but allows the XML representation of several resources to be accessed at once as part of a system. Both specifications also make use of a SOAP-based eventing mechanism to allow managers to subscribe for and receive events of interest. The actual SOAP messages used to execute these actions are different, preventing interoperability across the specifications, but the concepts and approaches are very similar.

The key difference between the specifications comes from the somewhat different perspective under which they were developed. The focus of WS-Management was to optimize for small, well-understood systems, such as the components of a computer or the services inside an operating system. WSDM's focus, as illustrated by the "D" in the acronym, was on allowing and managing distributed systems, composed of resources of very dissimilar types. This division in approach materializes in the only considerable architectural difference between WSDM MUWS and WS-Management: the fact that WS-Management specifies the structure of the EPR while WSDM MUWS doesn't put any constraint on how EPRs are created. The value of treating EPRs as opaque is that it promotes loose coupling between the service and the consumer. The addressing details are hidden in the EPR and never show up in the implementation of the consumer. The consumer only needs to find the EPR to know how to access the service, and if the service changes its addressing mechanism, an updated EPR will work and require no change on the consumer.

> " The key difference between the specifications comes from the somewhat different perspective under which they were developed "

On the other hand, this requires that the consumer first find the EPR of the service, which costs an extra step in the interaction. In management scenarios, an additional aspect to consider is the fact that the invoker usually knows and understands the model of the resource. In this case, if we assume that the addressing mechanism is based on elements of the model (a requirement that WS-Management unfortunately does not impose but that implementers of the specifications would be wise to obey), then exposing this addressing mechanism to the invoker doesn't really add much tightness to the coupling. This is the approach that WS-Management takes, and it is the same one that WSDM MUWS has avoided. In addition to the risk of more brittle systems, this approach creates somewhat of a barrier between manageability Web services and other Web services, even though integrating them is one of the explicit benefits of using Web services for management interactions.

The fact that the specifications have so much in common bodes well for the prospect of gradual alignment, and efforts to make this happen are taking shape, for example in DMTF. Many of these efforts are outside of the control of the DMTF though, since at the XML level most of the content of the different protocols comes from more general specifications, such as WS-Eventing and WS-Notification for events and WS-ResourceProperties and WS-Transfer for defining the messages used for retrieving the XML representation of resources. For these specifications too, the difference is often more in the realm of XML syntax than architecture, such that implementers of either stack should be able to think in terms of having to eventually support a new version of their stack of choice rather than a new stack if and when they rejoin.

Once again, those who implement their protocols with versioning in mind will harvest the benefit of their design efforts. ℮

■ About the Author

William Vambenepe is an HP Distinguished Technologist in the Office of the CTO of the Management Software Business where he is one of the architects of the technical strategy for HP OpenView. In addition to driving technical alignment with partners and key customers, he oversees the standards strategy for OpenView. He was the editor of the WSDM MUWS 1.0 specification from inception to OASIS standards and represents HP in the WS-Management working group.

■ ■ ■ vbp@hp.com
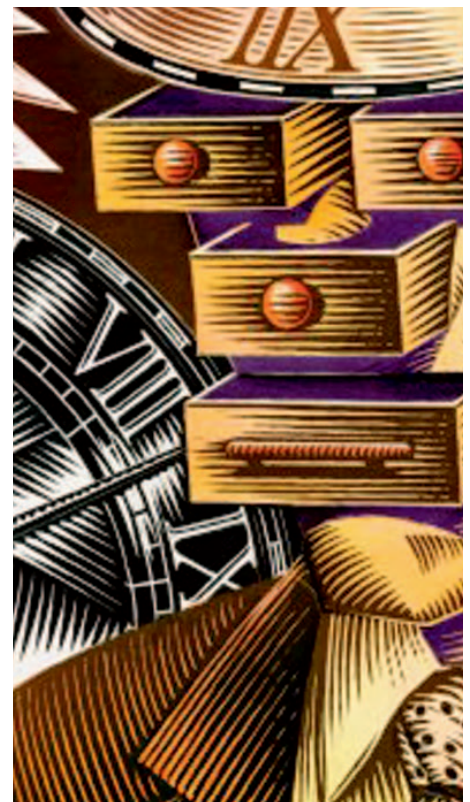
Opening the Borders of Telecom

New telecom advancements open up communications for SOAs

■ In the early days, the birth of desktop computing resulted in technology that provided integration between the computer and the telephone: we call it Computer Telephony Integration or CTI. The concept was delightful to most technology-minded people, but it never really emerged as a ubiquitous desktop solution. Today IP-based telephony for business is a hot and growing market, and many vendors are providing some level of integration with the desktop computer – the most basic is a software-based phone, the most advanced is complete control of user communications and multiple forms of media from which the user may easily select. While this *convergence of communications* at the user's desktop provides value for end-user productivity, these applications are typically independent desktop applications and have little integration with other business applications.

Today CIOs focus a great deal on ways in which their business applications can be better leveraged to improve the efficiency and productivity of the business. Service-oriented architectures (SOA) are increasingly being seen as providing a framework for more effectively integrating disparate business systems. According to IDC, nearly 70 percent of CIOs are planning or implementing SOA as part of their business systems. However, the telecommunications element of their business is rarely even considered in the SOA process.

At a recent presentation I did at an IP communications industry event I asked for a show of hands on how many people knew what "VoIP"

WRITTEN BY
TODD LANDRY

was. To no surprise, all people in the room raised their hands. I then asked how many knew what a "WSDL" was. I could see no hands in the air. Then I asked how many people knew what "SOA" was – no hands. "What you've just experienced is an example of the gap between telecommunications and business applications," I told the audience. Throughout many telephony-focused conferences one of the common themes is what the business value in IP-based telephony is, and a common answer is business applications. While telecommunications was making its move to IP-based software technology, other technologies were born out of a desire to provide a sharing of information and services among business applications, and one

of them is Web services.

The general goal of Web services is to construct elements of business logic – services – that can be very easily used by other applications. The services themselves hide the complexity of their business logic from the consumers through simple interfaces that allow the services to be reused in many different applications. The service and the consumer of the service are described as being loosely coupled, an approach that allows complex composite solutions to be developed through leveraging multiple Web services.

The following are some of the key elements of Web service technology:

- *XML (eXtensible Markup Language)* is the core language of Web service technology. It provides a platform-neutral way to describe the data associated with any service transaction.
- *SOAP (Simple Object Access Protocol)* is the preferred means by which an application invokes a Web service. The protocol itself is written in XML.
- *WSDL (Web Service Definition Language)* is the specification of the interface that a Web service exposes to consumers. It describes the set of operations that the service makes available. The WSDL is also written in XML.

Some of the advantages of utilizing Web services are:

- ***Loosely Coupled:*** The high degree of abstraction that exists between the implementation and the consumption of the service frees the Web service client and the Web service provider from needing any knowledge of each other beyond inputs and outputs.
- ***Platform Neutral:*** Web services can be provided by or invoked on most if not all platforms today, even legacy platforms.
- ***Development Language Neutral:*** People are developing Web services using C/C++, C#, Java, Visual Basic, Python and others.
- ***Standards Are Architecture Neutral:*** They do not enforce client-server, peer-to-peer, or any other model, although practical implementation issues currently drive developers to client-server.
- ***Web Services Leverage Existing Standards:*** Leverage standards that are already in place for conventional Web server applications, including HTTP, secure sockets, and authentication.

Complex JAVA J2EE Hosting made easy.

WebAppCabaret sm

http://www.webappcabaret.com/jdj.jsp
1.866.256.7973

JAVA J2EE-Ready Managed Dedicated Hosting Plans:

Xeon III
SAMEDAY SETUP

Dual 3.2 GHz Xeons
4GB RAM
Dual 73GB SCSI
1U Server
Firewall $399
Linux   *monthly*
Monitoring
NGASI Manager

Pentium 4 I
SAMEDAY SETUP
FREE SETUP

2.4 GHz P4
2GB RAM
Dual 80GB ATA
1U Server
Firewall $199
Linux   *monthly*
Monitoring   2nd month FREE
NGASI Manager

4Balance I

1 Database Server
and 2 Application
Servers connected
to 1
dedicated $1724
load   *monthly*
balancing device.
Dual Xeons.
High-Availability.

NEW! **Geronimo 1.0 Hosting . Virtual Private Servers(VPS) starting at $69/month**

At **WebAppCabaret** we specialize in **JAVA J2EE Hosting**, featuring **managed dedicated servers** preloaded with most open source JAVA technologies.
PRELOADED WITH:
JDK1.4 . JDK1.5 . Tomcat . Geronimo . JBoss . Struts . ANT . Spring . Hibernate
Apache . MySQL . PostgreSQL . Portals . CRM . CMS . Blogs . Frameworks
All easily manage via a web based control panel.
Details:
-All Servers installed with the latest Enterprise Linux
-Firewall Protection
-Up to 60 GB daily on site backup included at no extra charge per server.
-Database on site backup every 2 hours
-Daily off site database backup
-A spare server is always available in case one of the server goes down
-Intrusion detection.
-24x7 Server and application monitoring with automatic self healing
-The Latest Bug fixes and Security updates.
-Tier 1 Data Center. 100% Network Uptime Guarantee
-Guaranteed Reliability backed by industry-leading Service Level Agreements

Log on now at **http://www.webappcabaret.com/jdj.jsp** or call today at
1.866.256.7973

NGASI
POWERED

WebAppCabaret sm

JAVA J2EE Hosting

> "Through the maturing of Web service technology, enterprise applications are now being made to adapt to the specific needs of users rather than users being forced to adapt to the available functionality of applications"

- **Relatively Lightweight:** Therefore they can be easily deployed to small devices, for example PDAs and smartphones.

Through the maturing of Web service technology, enterprise applications are now being made to adapt to the specific needs of users rather than users being forced to adapt to the available functionality of applications. These developments are being applied to all aspects of information technology in the enterprise. It's only natural then to consider how business communications might be part of SOA frameworks, and since communication is a critical part of most business transactions, it seems logical that the enterprise-focused IP Telephony industry would consider its role in the development of SOA.

I have also started speaking to audiences of business software architects and CIOs in venues that are focused on approaches for SOAs. As part of my presentation, I introduce the evolution of telecommunications as a business application and how the result will provide a natural fit into their SOA planning. During these presentations it has become equally clear that that telecommunications are not being considered as part of SOA initiatives. Traditionally, business telecommunications are delivered through PBX systems that are proprietary, closed, and stand-alone. These represent expensive hardware systems installed in separate phone rooms or separate areas of the data center, operated over a separate wiring infrastructure, and managed by a team of highly trained telecom engineers. These systems run independently from every other aspect of the business.

It is well known that traditional PBXs are becoming IP-based PBXs, but more important, PBXs are becoming all software implementations that not only adapt to the business application environment, but are purposely built to align with business applications and the natural interfaces of the business application environment. Now, traditionally closed communications services are being fully exposed through Web services, within an SOA framework, to enable easy and comprehensive integration with other enterprise-class applications.

As an open, software-based business application, the PBX function can be abstracted away from the underlying network infrastructure and any telephones or gateways that it supports. As a software application, it can be architected as an open mission-critical communications solution rather than as a closed stand-alone system. The best approach provides a distributed application that is able to operate on any number of networked servers. The servers can be centralized or set up over any wide geographic area, with no one server acting as a single point of failure.

These new solutions provide a very rich feature set to their users that go far beyond the traditional audio feature sets of legacy PBX systems. At the heart is a presence engine that allows users to view the presence state (on the phone, in a meeting, out to lunch, etc.) of colleagues so that an appropriate means of communication can be chosen. If a colleague is on the phone then a user might choose to send a text message to the colleague requesting a call back when they free up. Additionally, these systems have the ability to "find" (or intelligently route calls to) a user based on a number of criteria, including presence state, time of day, and caller ID. As a business software application, they are able to offer this rich set of capabilities as services to other applications. In essence, IP PBX communications can become part of the core fabric underlying all business processes. Most business applications do not want to know or need to know about the complexity of establishing audio, video, or data communications between users. Web service technology provides an ideal integration approach, whereby the complexity of communications is shielded by the IP PBX from the business application, yet at the same time the technology provides a very simple approach for invoking the services.

Going forward, CIOs will have the power to deploy telecommunications as an enterprise-class software application running on their underlying data network, in order to provide a rich set of communications services, and most important, to integrate with other enterprise-class applications. By doing so, new business-advancing levels of efficiencies and productivity will be achieved – not only in the IT shop but also for end users.

The vision of an SOA providing a framework for next-generation enterprise telecommunications services has become a reality through the maturing of Web service technology. Communications Web services are a new approach for business communications that promise CIOs the opportunity to rethink how the enterprise is architected and to establish communications as the essential fabric supporting all of their business processes.

Such communications services can now take their rightful place alongside collaborating business processes to allow enterprises to achieve the dream of a fully integrated service-oriented architecture, an architecture that can grow with the business and grow with technology. Ultimately, communications Web services create a more integrated, optimized – and more competitive business. ⓔ

■ **About the Author**

Todd Landry is senior vice president of Sphere Communications, providing IP-based software communications technology for mid-size organizations. He welcomes comments at tlandry@spherecom.com, and more reading is available at www.spherecom.com.

■ ■ ■ tlandry@spherecom.com

Don't Miss the 2006 JavaOne℠ Conference

More than 300 technical sessions and Birds-of-a-Feather sessions will be offered at the 11th Annual JavaOne℠ Conference. See and hear from the Industry leaders and technology experts over four content-rich days. Included are:*

Introduction to AJAX
Ben Galbraith, *Consultant* | Dion Almaer, *Adigio, Inc*

This session provides an introduction to AJAX and an orientation to the state of the AJAXian universe. It demonstrates the basic AJAXian techniques through live coding and demonstrates and deconstructs more-advanced examples of AJAX.

Effective Java™ Reloaded
Joshua Bloch, *Google, Inc.*

It has been five years since *Effective Java*™ was released. The Java platform has evolved, and we've learned more about how to use it to best effect. Therefore, a second edition of *Effective Java*™ is being released to coincide with the 2006 JavaOne conference. This presentation covers new material that has been added to the second edition, material that should be useful to every working Java technology programmer.

Spring Framework Update
Rod Johnson, *Interface21*

Rod Johnson, the father of Spring, brings attendees up to date on some of the many significant new features in the Spring 1.3 and 1.4 releases. He discusses what's new and cool in the Spring world and examines the implications of these new features for best practice in developing applications with the Spring Framework. Johnson shows code examples throughout the presentation, leaving attendees ready to try these features out for themselves.

To see more information on the conference offerings visit java.sun.com/javaone/sf

*Content subject to change.

Register by April 14, 2006 and SAVE $100.
java.sun.com/javaone/sf

Web Integration Architecture Patterns for Enterprise Architects

Approaches to Web enablement of legacy systems

■ Most well established large organizations suffer from some level of "Web sprawl." It organically grows for the same reasons as a disparate systems environment:

• Mergers and acquisitions

• Legacy architecture now deprecated

• Tactical solutions circumventing architectural goals

• Disparate IT skill sets

• Lack of architecture steering

• COTS package purchases provide heterogeneous platforms and functional overlaps

• No centralized IT, i.e., different IT organizations across business units

The list could continue. By the same token the enterprise architecture team is tasked with reigning in Web sprawl and stemming its propagation.

Fighting Web sprawl within a large organization is an uphill battle. Experiencing this on a daily basis has shown me that there are some things you can do to help alleviate the issues. It is popular (and rightly so) to have a target unified portal framework somewhere along its migratory path in your organization. This is important, but you also want to ensure

WRITTEN BY
MARTYN HILL

that it is applied against Web enablement requirements as they raise themselves. What is required is a means to describe how to classify the context of a requirement's solution.

What Are Architecture Patterns?

The *architecture patterns* described herein demonstrate an approach to Web enablement of legacy systems and a catalog of options. They are deliberately more iconic in nature, and provide a simple way to communicate an idea to both IT and the business.

When published in an appropriate manner they allow the "requirements front door" (e.g., business analysts) to select from a catalog of choices for Web enablement.

A solution can reside between the options presented and can also mix and match approaches.

There are further *design patterns* applicable within any one of the architecture patterns shown. The catalog serves as a generic overview of the highest-level choices to be made.

It is imperative to the success of this approach that the use of the patterns is added to the IT delivery processes. This will allow the centralizing of exceptions and the creation of new patterns over time to meet changing needs. Also, not least it will give a better understanding of the candidate solution, hence better effort planning and estimating.

The Misunderstood Single Sign-On

If I had a dollar for every time I was asked to build a single sign-on (SSO) interface to an existing Web application, I'd be a…, well I'd have enough at least for a trip to my favorite restaurant.

It is one of those perfect cases of the old saying "when you have a hammer, everything looks like a nail." Indeed, any of the patterns presented later may require an SSO interface

with some external resource or EIS. However this is only part of the solution, and getting the architecture right can greatly simplify the task of unifying the corporate web. While providing a consistent SSO solution is important, so is business process alignment, corporate branding, consistent user journey, data consolidation, usage and marketing reports, etc.

All too often requirements will arrive at IT saying something like *"provide single sign-on to <legacy> application."* Ideally the requirement should have been a functional set of requirements (which can point out *"by the way <legacy> application is already performing this today"*). It should then be an enterprise architecture decision as to whether to:

- Indeed keep the existing legacy application and perform SSO from the strategic portal, but also to consider alternatives such as:
  - Refactor all or part of the legacy application to the target architecture
  - Replace all or part of the legacy application with a target strategic application
  - Migrate the legacy users to an existing strategic application (may involve improvements to support gap analysis from legacy requirements)

Enterprise architects have the job of evangelizing the target architecture and often reasons are found to circumvent this (budget, IT skills, time-to-market, etc.). One of the few times when architecture should be able to be influenced more easily is incrementally "while the hood is up." Requirements of the form described above often hide the underlying business needs that are trying to be met.

Catalog of Web Integration Options

What can we do as enterprise architects and IT professionals? Well we cannot be everywhere at once, particularly in a large organization. The creation, presentation, and *prominent* publication of a set of Web integration options presented as architectural patterns will help to classify your Web enablement solutions. It will have a number of positive effects:

- Business analysts can correctly align solutions, which helps the evolution to your target portal architecture
- Using the correct pattern can help point out the key benefits and hence help the business case for more strategy-based solutions rather than always the tactical
- Many metrics of the solution are easier to establish given the right classification (such as complexity, budget, effort, expected ROI)
- In time the business itself will start to put requirements in the right solution context before IT are even involved

The patterns for Web enablement will vary from organization to organization and can depend on many factors such as: the availability of COTS packages in your industry, the size of your company, the amount of deprecated legacy, the technical architecture of the strategic portal environment, etc. It is important to create the feasible set that represents your
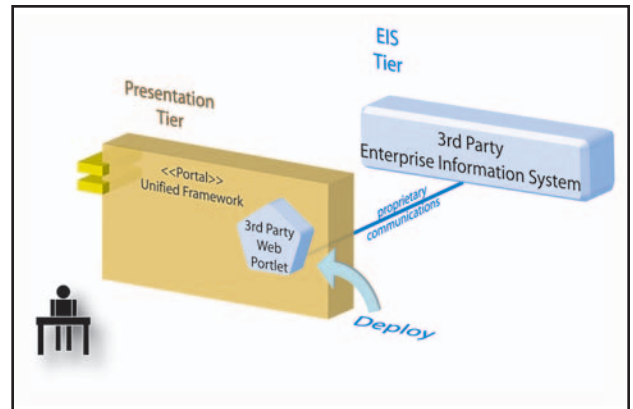
organization along with an understanding of the architectural order of preference (and the business factors that might push a solution down the preference list).

The catalog list presented below illustrates some of the key approaches to Web enablement in a large company:

- Standardized Portlet
- Remote Portlet
- Composite Portal Application
- Web Clipped Portlet
- Direct Content Encapsulation
- Separate Browser Launch

As stated earlier, remember we can mix and match across these patterns to define a candidate solution context.

1. Standardized Portlets (Web Parts)

Problem
Vendors often want to provide both the process logic and the presentation tier of an application but customers have heterogeneous portal environments and this only propagates problem.

Solution
In many large organizations a buy vs build approach is often taken, particularly when the functionality is not a core competency of the business (i.e., systems such as billing, CRM, product catalog, etc.). Just because we have an ideal architectural solution in mind we cannot ignore this fact. We also don't want to reinvent the wheel if a COTS package already provides the necessary presentation-level functionality.

A standardized deployment architecture

> " The creation, presentation, and prominent publication of a set of Web integration options presented as architectural patterns will help to classify your web enablement solutions "
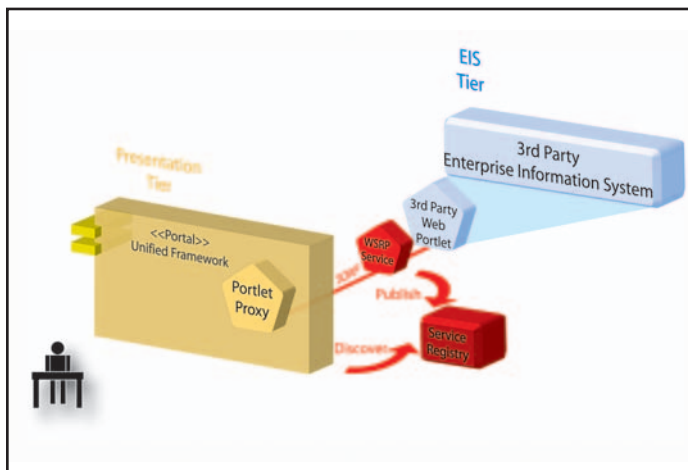
FIGURE 2 | Remote portlet
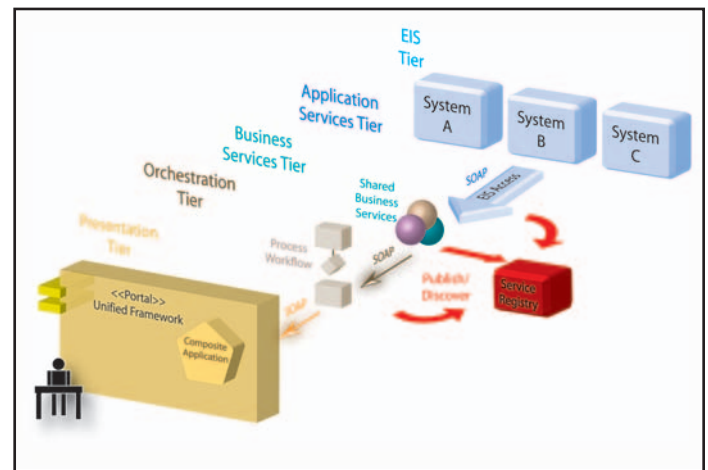
FIGURE 3 | Composite portal application

is provided for Web application vendors to develop upon. Usually the provider of the EIS also provides the standardized portlet/s, but not always. The Web application is deployed locally into the customer's native portal framework. Although largely dependent on the vendor's implementation, the system's being Web enabled acts as a black box to the portal.

Application vendors now only need to support this one standard and can market their applications to any compliant server environment. The solution is close to out-of-the-box and usually requires only deployment and configuration with little-to-no development necessary. Also the consuming portal avoids the need for custom integration code to be built.

Applicability
- The COTS vendor provides all the necessary functionality required (or the ability within their Web framework to create it)
- The processes involved do not generally span other systems or functionally overlap, and there is no strong requirement for inter-portlet communications
- The internal processes of the solution application are not reused elsewhere

Example Known Uses
JSR 168 offers a standard for J2EE portlets to allow deployment of COTS portlets into any compliant application server (BEA, IBM, etc.).

Custom Web Parts and Sharepoint Services offer a consistent framework to extend the Microsoft Sharepoint Portal Server environment.

2. Remote Portlet (Web Parts)
Problem
Vendors and third parties not only require developing their own presentation tier as in #1, but they also only wish to allow deployment within their own application server environment.

Solution
Allow the presentation tier to be accessed via a Web service layer. *Web Services for Remote Portlets* (WSRP) is an OASIS standard that is gaining significant support within the portal market. Vendors can build Web applications that effectively run locally but that also are WSRP enabled so that other portals can access them remotely. There is no visual difference to the user.

It allows a local portal (*consumer*) to reuse and aggregate other portlets within remote portals (*producer*).

This solution avoids the need for integration code to be built. It can be used internally within the enterprise to establish a federated portal environment. There is no reason why the companies own Web applications should not be WSRP enabled also to add greater deployment flexibility in the future.

Applicability
- Consumption of an existing Web application where a producer can still manage presentation but also offer its functionality via a Web service (Web services *used to be* predominantly for business/data exchange)
- Custom produced for long-term legacy Web application where it can remain in its exist-

ing infrastructure and be reused elsewhere (federation)

Known Uses
WSRP is supported by many leading portal vendors, including BEA, Microsoft, Oracle, TIBCO, and IBM.

3. Composite Portal Application
Problem
We may wish to build new applications that extend existing COTS-based systems, but changing them will cause maintenance and upgrade issues. Also the processes involved in a new application may involve many existing systems, and therefore building the solution in any one system will cause tight coupling between the systems.

Solution
A company's operations can be described by a finite set of business processes and data elements. These business process and data elements are implemented in IT systems. The wrapping up of these into "services" that can be easily reused is one of the fundamental tenets of service-oriented Architecture (SOA).

Most SOA strategies include an orchestration capability that allows the composition of a set of shared business services to form an application or new business process. Because of the open nature of Web services these process and business services can be easily accessed from anywhere, including the Web presentation tier, which is termed a *composite portal application*. The composite application is left

only to deal with the presentation aspects of the application.

Applicability
- Preferred when an SOA strategy is in place, in particular to provide a unified front end and to help establish more coverage within service catalog
- Creating Web enablements of EIS tier reusing EAI services
- To extend a systems capability while maintaining its vanilla implementation
- Building applications that span multiple systems and resources
- Providing the greatest business agility once there is a well established and populated service registry
- Building macro-level workflow capabilities across the enterprise

Known Uses

There is a huge market for SOA related products. Business Process Execution Language (BPEL) is a stabilizing and prominent second generation web service technology that provides Orchestration capabilities. BPEL compliant platforms are available from many vendors - Microsoft BizTalk, BEA AquaLogic & WLI, IBM WebSphere to name only a few.

4. Web Clipped Portlet
Problem

An existing application may already func-tionally match your requirements but it is outside of your control: maybe it is provided by an external party, maybe the development team have moved on, and it probably does not strategically match both architecturally and in terms of corporate look and feel.

Solution

Web clipping acts as an intermediary that allows the user to interact with ANY Web application, extract and manipulate its content, and finally present it in a portlet conforming to your branding and style guide. The solution will likely still require investment in an SSO interface.

Web clipping has become quite sophisticated in recent years. Vendors such as Kapow not only offer clipping of all types of Web content (including JavaScript), but also the harvesting of data into a virtual data model, and now more recently the wrapping of Web access behind a more ideally modeled Web services interface. This provides a lot of flexibility in helping to extend the reach of the SOA.

Of course the main issue with clipping another party's content is stability when the content changes. This needs to be addressed upfront in the design both technically and operationally.

Applicability
- When you cannot influence the design of a strategic/longer-term application

- To harvest tactical Web applications with little to no involvement from the development community
- When the Web application comes from a third-party supplier and is unchangeable
- When the layout or style of a Web application would prohibit a direct encapsulation pattern

Known Uses

Kapow offers both stand-alone products and is also integrated into BEA WebLogic Workshop.

5. Direct Content Encapsulation
Problem

A Web application is already available and deployed that offers the required functionality, but is not viewed within the strategic portal environment.

Solution

Create an appropriately sized portlet to host the display of an external Web applications content as-is. You need to employ the use of an HTTP proxy in the portal framework so that it can essentially act as a client to the external Web application and handle cookies if necessary. Still usually requires investment in an SSO interface.

Applicability
- This is a cheap option for integration if
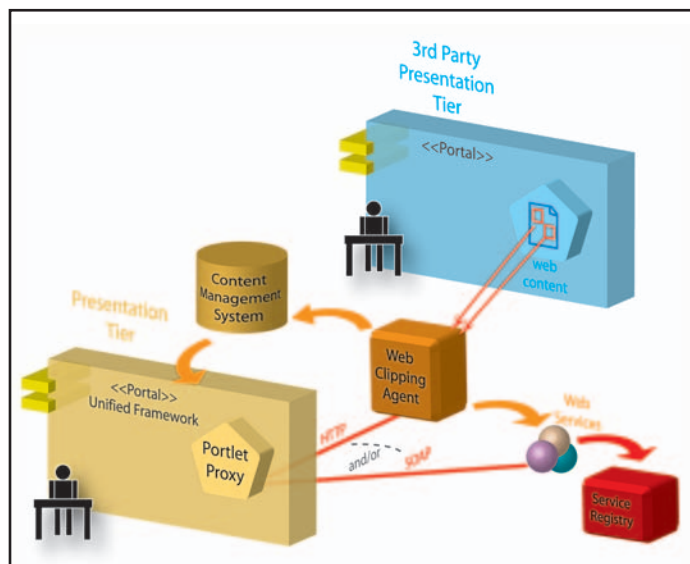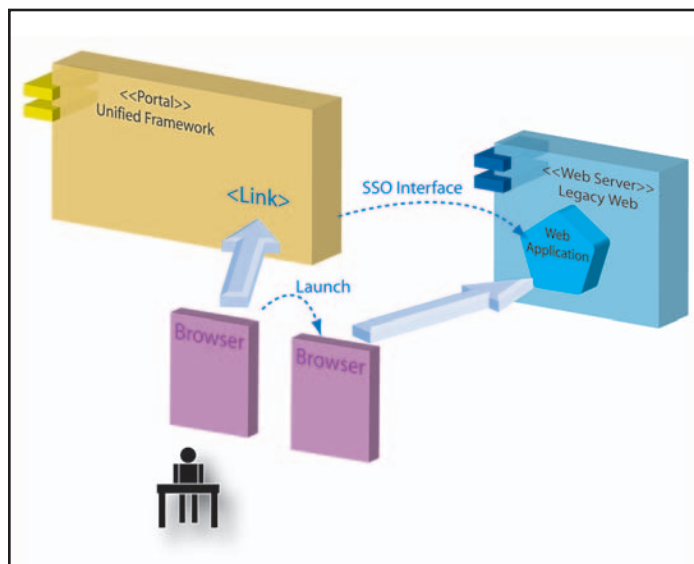
content is already laid out and branded appropriately. Control of format and look and feel is very limited, unlike the Web clipping option.

Known Uses
Most portal vendor platforms provide the ability to easily encapsulate content into a portlet view.

6. Separate Browser Launch
Problem
A legacy Web application already exists and the user community or business is unwilling or unable to migrate to a new strategic portal.

Solution
Provide a simple option to maintain a URL link to a separate Web server and launch the application in a separate browser window. Still usually requires investment in an SSO interface.

The strategic portal environment at least gets to act as an SSO launch pad for all applications and centralizes authorization (to see links).

Applicability
• When the web application is non-strategic or little investment needs to be made in its integration into portal
• When content size/layout precludes inclusion in a portlet

• The business is unwilling/unable to move to a more preferred architectural solution
• The portal environments must be seen to become more unified rapidly

Known Uses
Separate Browser Launch is probably the easiest and cheapest form of Web integration (when a legacy Web app already exists).

An Order of Preference?
It is useful to not only present the choices such as those above but to also let your organization know what the architectural preference would be given all things being equal.

The benefits matrix in Table 1 helps to illustrate when one approach may be more appropriate than another. Given this information, we can establish an order of preference.

We can see that the earlier misguided (or incomplete) requirement *"provide single sign-on to <legacy> application"* with no other architectural considerations applied, e.g., option #6, represents the least preferred solution from an architectural perspective.

Summary
Eradicating Web sprawl within an organization often feels like a never-ending task. One way to help is to undertake the classification of the context of candidate solutions to Web enablement, to catalog them in a simple fashion, to establish an architectural order of preference, to present and publish the approaches, and to make their use part of the delivery processes within your organization. Within time the correct solution context will be selected further upstream in the requirements process and the evolution to the target architecture will accelerate. Ⓔ

■ **About the Author**
Martyn Hill is an enterprise architect with over 19 years of experience in an engineering environment. He is currently a principal architect with CSC Consulting's national practice, specializing in enterprise architecture. He has led the successful development and implementation of strategic architecture and roadmap visions for SOAs, enterprise application integration, Web portals, business gateways, and Web services management platforms for large-scale enterprises.
■ ■ ■ martyn.hill@adelphia.net

	Standardized Portlets	Remote Portlets	Composite Portal Application	Web Clipped Portlet	Direct Encapsulation	Separate Browser Launch
Typically SOA aligned solution		✔	✔	✔		
Interportlet communications	✔	✔	✔			
Fast time to market	✔	✔		✔	✔	✔
Lowest cost (typically)				✔	✔	✔
Centralized intra-app authorization	✔	✔	✔			
Centralized personalization	✔	✔	✔			
Centralized look and feel	✔	✔	✔	✔		
Application events surfaced in portal	✔	✔	✔			
Reduced SSO complexity	✔	✔	✔			
Shared application resources	✔	✔	✔			
Reduced TCO	✔	✔	✔			
Less capital expenditure	✔	✔	✔			
Centralized physical and network architecture	✔	✔	✔			
Centralized platform scaling, admin, and monitoring	✔	✔	✔			
Consistent application-level user journey	✔	✔	✔	✔		

TABLE 1 Benefits matrix

LinuxWorld
CONFERENCE & EXPO

Conferences: April 3 – 6, 2006
Expo: April 4 – 6, 2006

Introducing: **OpenSolutions** WORLD™

Boston Convention
& Exposition Center

BOSTON'06

2 Conferences

10 Tracks

100+ Sessions

Over 200 Exhibitors

OPEN Source.
OPEN Solutions.

OPEN. For Business.

LinuxWorld Conference tracks
- Mobile & Embedded Linux
- Network Management
- Kernel and Driver Development
- Linux on the Desktop

OpenSolutions World Conference tracks
- Business Case
- Security
- Open Source Applications
- Enterprise Application Development
- Dynamic IT: Cluster/Grid/Virtualization
- Emerging Trends

Open source means business.

Find out what it can mean for your business at LinuxWorld Conference & Expo—in ten tracks and more than 100 in-depth sessions, tutorials, and workshops led by the open source movement's top minds.

Go beyond Linux.

Expand your vision at OpenSolutions World, a new conference at LinuxWorld covering the full spectrum of open source solutions and strategies for the enterprise.

Get the full picture.

Explore a comprehensive exhibition of open source products, technologies, and services from more than 200 key Linux and open source vendors—all under one roof.

Put it to work.

Come to LinuxWorld Conference & Expo April 3 – 6, 2006, and harness the power of the entire open source community for your business. The future of enterprise technology is wide open.

Register by 3/3 and save
Enter priority code D0302 and save up to $500
www.linuxworldexpo.com/boston

IDG INTERNATIONAL DATA GROUP
IDG WORLD EXPO

D0302

Save Our Architecture

Driving an SOA strategy top-down from the business

■ As I look upon the enterprise landscape today, I cannot help but wonder if the enterprises are going to be all right. Every couple of years, enterprises have to face the onslaught of their vendors who bring in newly coined phrases, acronyms, and newly minted software platforms, along with the promise of ROI. In the latest onslaught, enterprises are facing terms such as SOA, Web services, BPM, and BAM.

The new buzzword is AJAX, or Asynchronous JavaScript And XML. The speed at which the IT industry coins terms and makes it fashionable to use them gives the cool name creators in the fashion and entertainment industries a run for their money. System integrators and consultants capitalize on the confusion and introduce new solutions and services that range from architecture blueprinting to specific product implementations. Enterprises turn to industry analysts to understand who is the leading vendor of the current hype cycle and eventually cave in to the trend and buy another piece of technology to add to the existing collection. During the course of my career, I have been in all of the camps: vendor, consultant, and system integrator, which enables me to provide some unique and original thoughts on how to save the enterprise architecture.

WRITTEN BY
VENKAT

Understanding Architecture

Architecture is the most misunderstood, yet most widely used word today in IT. This is a word that has been borrowed by IT. This word has its origins from Greek around the 1500s when the related word, architect, meant a

"master builder." Today, architecture, related to usage in IT, is used very loosely to denote the art and discipline of creating software, computers, systems, databases, information, and enterprises themselves.

In order to make the right decision, one has to understand all of the options. To make the right decisions pertaining to SOA, one has to understand what SOA is as well as the available options. Though SOA is looked upon as a technical term, in order to make the right decision, SOA has to encompass business as

well. Even before the term SOA was coined, I had consulted on numerous occasions with large enterprises on their "enterprise architecture." Companies have always been interested in ensuring that their investments in technology, applications, and infrastructure materialize into tangible gains either in productivity and efficiency of operations, or a direct impact to top-line and bottom-line growth. SOA cannot be viewed myopically as only a technology term that impacts the technology architecture of an enterprise.

SOA needs to be divided into four layers and addressed holistically:

- Business architecture
- Functional architecture
- Technical architecture
- Infrastructure architecture

Figure 1 shows that SOA is the most generic (commodity) at the bottom layer and is very specific (unique) to a particular way of doing business at the top layer. Even though two companies may be in the same line of business, industry vertical, and may even have the same business model, the business architecture is very specific and is dependent on the people, business culture, company history, relationships, business processes, and the business characteristics among other things that cannot be duplicated exactly by another business. On the other hand, the lowest layer is the most generic where infrastructure could consist of mainframes, networks, routers, servers, etc. Although the configuration and implementation of the infrastructure elements could be different, the important point is that
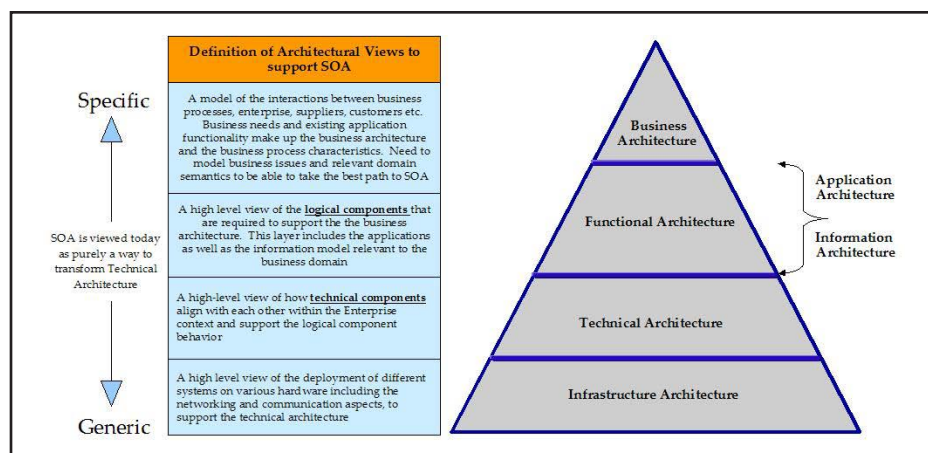

FIGURE 1 | **Layers of SOA**

> "Architecture is the most misunderstood, yet most widely used word today in IT"

as one traverses down the SOA layers, one goes from specific to more generic.

An enterprise has to first identify its business problems and prioritize them. Subsequently, it must figure out the business process characteristics of the business problem and its impact on the business architecture and model that within the context of the business architecture. This is easier said than done.

With this new way to look at architecture in general and SOA in particular, an enterprise now needs to catalog the artifacts that reside in its layers in order to embark upon a successful SOA strategy. The artifacts run the entire gamut, containing business processes, application, services, technical components, documents, policies, infrastructure components such as servers, and anything else that represents enterprise knowledge.

SOA and ROI: A Business Problem?

The SOA vendors sell everything from soup to nuts that fits into the SOA layers. The main point that is harped upon is that SOA will allow enterprises to create and make available "services" that can be put together to provide application functionality. Vendors such as IBM maintain that modular components should be easily defined and manipulated along with dynamic definition of application and operations (see the first entry in the References section). The attractiveness lies in the fact that these services can be located through a registry and that the services can be reused in multiple areas.

The risk of having loosely coupled services that orchestrate business processes is that there is really no control over what is transpiring during process orchestration. BPM vendors realized this and provided products to design, visualize, and orchestrate business processes. However, that was not enough. The IT infrastructure and applications that supported the process orchestration were now loosely coupled and were represented as services. As such, vendors stepped in to manage applications and services and monitor business activities. The vendors that offered products in this area originally were specialized, from managing only infrastructure, servers, network etc. to managing only Web services or applications.

The problem with existing SOA solutions is that there is also no attempt to model the business problem. There is an attempt to model the processes (BPEL, BPMN). There is an attempt to model the software system (UML). Business Process Execution Language
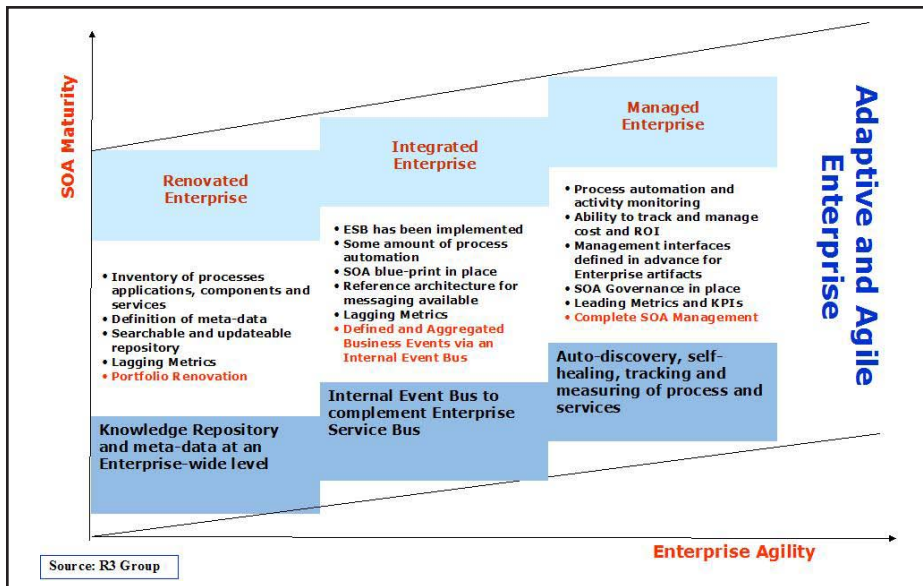
FIGURE 2 | Managed path to SOA

(BPEL) is a way to represent orchestration between two parties. Several vendors jointly submitted this standard to Oasis and the standard is at version 2.0 currently. Business Processing Modeling Notation (BPMN) is a way to represent workflow graphically. As such, BPMN complements BPEL. Neither BPEL nor BPMN provides a way to understand the business issues that may arise inherently from business process orchestration. They also do not provide a way to correlate disparate business-related events that occur as a consequence of process orchestration. There is no attempt to model the specific set of business issues. That is the gap in the current approach to SOA. It is currently being addressed from a purely technological angle with the technology decisions driving the business decisions. This is contrary to what one observes when analyzing SOA as a layered architecture. It is clear that the business architecture is the most specialized and must drive the rest of the architectural decisions. Forget the business-to-IT alignment in the enterprise: there is no business process-to-software systems alignment in the SOA model. Vendors such as Red Rabbit Software have bridged that gap with a business domain modeling studio.

A research field called Complex Event Processing (CEP) has been around for many years now Advances have been made in a related area called Business Event Processing (BEP). Alert readers will immediately notice the introduction of two new terms in keeping with the rapid pace of name generation in IT. BEP is related to CEP in that there is an attempt to relate events in time and space (causality) as they transpire during business process orchestration. There have been attempts to apply CEP to various problem areas, but the application almost always tries to work its way up from the technology layer and tries to bridge the occurrence of technical events to possible business causes. CEP is more applicable, as it stands today, to scientific and technical problems whereas Business Event Processing targets business domains, business pain points, and business models, as well as how to take a top-down approach from the business architecture and make an SOA strategy successful.

By utilizing BEP, an enterprise can first model a set of business events that relates to its business issues. These issues are encountered specifically within a business domain. For example, a consumer packaged goods company may have issues with its inventory levels and safety stock whereas an insurance company may have problems with the cycle time taken to process an insurance policy application. Once a business domain model has been created, it is mapped to the actual business functionality that supports the business processes. By utilizing the BEP platforms, it is now easier to enable only the impacted applications to conform to an SOA strategy while, at the same time, ensuring that the specific business issues that plague a particular domain are better understood. Visibility into the business problems will now be possible, all the way from the business architecture through the entire SOA stack.

In order to realize ROI from investment in SOA, an enterprise needs to clearly articulate its business issues and what is quantifiable and measurable with regard to the issues. Any SOA technology adoption should also include a platform that would help model business issues and then allow an enterprise to track the solution to the modeled issues. Also, there is no single solution to the SOA technology. SOA needs application servers, Enterprise Service Bus, Business Processing Modeling, and Business Event Processing in the technical architecture layer, and these technology components come from different vendors.

The challenge also is whether the newer technologies can work on top of existing technologies. An SOA strategy will take a few years to implement and the new technologies will have to work with the existing (and in most cases legacy and proprietary) technologies to ensure that it is "business as usual" in the enterprise while the architecture is undergoing transformation. Enterprises cannot afford to run a parallel IT department while waiting to convert their existing applications to SOA and

> " If the SOA strategy does not start at the level of the business architecture, you are not going to reap ROI "

then flip a switch to the new architecture. The real problem is that most of the existing SOA approaches take an "all or nothing" approach with respect to enterprises.

How Do I Save My Architecture?

As a practitioner who has implemented enterprise transformation strategies successfully, let me share some knowledge. The key to a successful SOA implementation does not lie in throwing out existing legacy technology and bringing in the latest and greatest technology platform identified in a magic quadrant. If that happens as a consequence of the strategy, so be it. I have never seen it happen that the SOA strategy proved to be successful when it was based on just bringing in new technology. If the SOA strategy does not start at the level of the business architecture, you are not going to reap ROI. An SOA strategy is also not something that is implemented in weeks or months. It is a journey that takes a few years to reach SOA nirvana.

The key questions to ask are:

- What business issues and pain points do I have today?
- How do my business metrics and KPIs compare with my competition and industry benchmarks?
- What is my enterprise's vision and what are the strategic business areas that my company will be in over the next few years?

If your enterprise has zero business issues or pain points, which is quite impossible if you live in the same world as I do, please tell the CEO that you deserve a long vacation that lasts a few months and come back when the next set of terms are coined in IT. If you are like the rest of the world, then the business issues need to be defined and captured in a model, and later when the SOA strategy is implemented, translated into measurable and quantifiable observations.

As important as it is to advance top-down from the business architecture, it is equally important to take a "well-managed" path to an SOA strategy. This path should broadly involve three steps:

- Renovation
- Integration
- Management

The maturity of a business in its SOA adoption increases as the enterprise becomes more agile and adaptive to its business issues. An enterprise will become adaptive and agile only when it embarks upon a managed path to SOA, taking the enterprise through stages of achieving SOA rather than a big bang approach targeted at bringing in new technologies that promise transformation of an enterprise architecture to SOA.

Advancing top-down from the business architecture layer will ensure that you are able to prioritize your business issues and then figure out what artifacts in your other layers will be impacted when you set out to solve your business pain points. Thus, instead of ripping out existing applications, functionality, technology platforms, and infrastructure in the pursuit of SOA nirvana, a managed path will be taken, which will ensure that the relevant portions of each architecture layer are transformed in your SOA quest. This will also allow an enterprise to selectively adopt new technologies that are relevant to the enterprise's needs.

There are still many issues to reflect upon in an SOA strategy. In large multinational enterprises, there are localized IT departments in addition to enterprise IT. Different business divisions have their own strategy and agenda. It is a challenge to even fathom what lies in an enterprise basement (Reference entry 2). Enterprises today have limited budgets, multiple and sometimes redundant applications and technologies, and constant demand from business for new features.

Unless, there is a "managed" SOA initiative, it is doomed to failure. A managed initiative will try to address achieving SOA in three steps: renovation, integration, and management, utiliz-

ing Business Domain Modeling and Business Event Processing so that we can save our architecture.

References

- "Operating Environment essentials for an on-demand breakthrough," March 2004, IBM
- "Red Rabbit Software: SOA The Right Way": www.redrabbitsoftware.com/services/psg
- Venkat, G. "Have You Checked Your Basement Lately? " JDJ Vol. 9, iss. 11. http://jdj. sys-con.com/read/46964.htm ℮

■ About the Author

Venkat has over 13 years of experience in distributed computing, technology strategy, and enterprise architectures. Venkat is the chief technology officer at Red Rabbit Software where he leads the Business and Technology Strategy, the direction of the Red Rabbit Software technology platform, and Research & Development. He has completed PhD courses in Interdisciplinary Studies involving Computers and Aerospace Engineering and has a Bachelor of Technology degree from The Indian Institute of Technology, Madras, India.

■ ■ ■ venkat@sys-con.com

FACULTY

Featuring... Real-World AJAX Rock Stars!

Jesse James Garrett
Father of "AJAX" Who Coined the Term in 2005

Jesse James Garrett is the Director of User Experience Strategy and a founding partner of Adaptive Path, the world's premier user experience consulting company. He is author of The Elements of User Experience (New Riders), and is recognized as a pioneer in the field of information architecture. Jesse's clients include AT&T, Intel, Crayola, Hewlett-Packard, Motorola, and National Public Radio. Since starting in the Internet industry in 1995, Jesse has had a hands-on role in almost every aspect of Web development, from interface design and programming to content development and high-level strategy. Today, information architects around the world depend on the tools and concepts he has developed, including the widely acclaimed "Elements of User Experience" model. He is co-founder of the Information Architecture Institute, the only professional organization dedicated to information architecture. He is also a frequent speaker and writer whose work has appeared in numerous publications, including New Architect, Digital Web, and Boxes and Arrows.

Scott Dietzen
One of the Fathers of WebLogic and J2EE
Ph.D., President and CTO, Zimbra

Scott is widely credited with helping put together the J2EE standard, launching the Web application server category, launching the Java Community Process, and driving the Web services collaboration with Microsoft and IBM. Prior to Zimbra, Scott was CTO of BEA Systems where he was the principal architect of the technology strategy for the WebLogic product family.

Bill Scott
AJAX Evangelist of Yahoo!

Bill Scott, one of the top AJAX experts in the country, is an Interaction Designer and AJAX Evangelist at Yahoo! He is part of the newly formed Design and Practices Team working with teams throughout Yahoo! to create a rich experience on the web. Before joining Yahoo! Bill founded the User Experience Team at Sabre Airline Solutions, part of Sabre Holdings. During that stint he also co-founded Rico, an open source JavaScript framework for creating AJAX & DHTML web applications. Over the past 20 years Bill has been involved in designing and creating user interfaces for video games, military war games, 3D graphics, oil and gas research, software development environments, supply chain planning, and various other scientific and business domains. He posts his musings about user experience on his blog.

David Heinemeier Hansson
Creator of Ruby on Rails

U.S.-based since November, David Heinemeier Hansson is the development lead of Rails, also known as Ruby on Rails, which the official Ruby on Rails website irreverently describes as "a full-stack, open-source web framework in Ruby for writing real-world applications with joy and less code than most frameworks spend doing XML sit-ups." He has had help from a lot of contributors. David is the creator of applications like Instiki, Basecamp, and Ta-da, and has now joined the Chicago-based team of 37signals.com, a privately held company founded in 1999 and committed to building the best web-based software products possible with the least number of features necessary. David will give his first presentation on "AJAX in Rails" at the "Real-World AJAX" one-day seminar.

Adam Bosworth
Vice President of Engineering, Google
One of the Fathers of XML & the Creator of MS Access

Adam Bosworth is Vice President of Engineering, Google. He joined Google in 2005 from BEA Systems, where he was Chief Architect & Senior VP of Advanced Development. Prior to joining BEA, Bosworth co-founded Crossgain, a software development firm acquired by BEA. Known as one of the pioneers of XML, he previously held various senior management positions at Microsoft, including General Manager of the WebData group, a team focused on defining and driving XML strategy. While at Microsoft he was also responsible for designing and delivering the Microsoft Access PC Database product and assembling and driving the team that developed the HTML engine of Internet Explorer 4.0.

Rob Gonda
Bestselling AJAX Author, CTO, iChameleon Group
Editor-in-Chief, AJAX Developer's Journal

Rob Gonda, newly appointed to the helm of SYS-CON Media's AJAX Developer's Journal, is the CTO for iChameleon Group. He is an Advanced Certified Coldfusion Developer, holds a BS in computer science and engineering and an MBA with a specialization in entrepreneurship. He recently wrote a two-part feature article on AJAX for ColdFusion Developer's Journal that in one month became the most-read article in the magazine's history.

Dion Hinchcliffe
Cofounder & CTO, Sphere of Influence Inc.
Editor-in-Chief, Web 2.0 Journal

Dion Hinchcliffe, newly appointed editor-in-chief of SYS-CON's pioneering Web 2.0 Journal, is cofounder and chief technology officer for the enterprise architecture firm Sphere of Influence Inc., in McLean, Virginia. A veteran of software development, Dion works with leading-edge technologies to accelerate project schedules and raise the bar for software quality. He is highly experienced with enterprise technologies and he designs, consults, and writes prolifically. Dion actively consults with enterprise IT clients in the federal government and Fortune 1000. He is a frequent speaker on AJAX, Web 2.0 and SOA and is currently the top-read SYS-CON.com blogger.

Paul Rademacher
Google, Creator of HousingMaps.com

Paul Rademacher is the creator of HousingMaps.com, which combined Craigslist and Google Maps for the first web mashup. Paul holds a Ph.D. in Computer Science from UNC-Chapel Hill, and worked as an R&D Engineer at Dreamworks Animation on such movies as Shrek 2 and Madagascar. Since creating HousingMaps, Paul is now at Google.

Ross Dargahi
Well-known AJAX Evangelist
Co-founder and VP of Engineering, Zimbra

Ross Dargahi, a co-founder of Zimbra, is presently the company's VP of Engineering. Prior to Zimbra, he was Director of Engineering and Director of Product Management with the Messaging Products Group at Openwave Systems - where he built and led the engineering team that designed and architected large-scale messaging subsystems and was responsible for unified messaging, multimedia messaging (MMS), and voicemail. Ross joined Openwave as part of the 1999 acquisition of Onebox where he was a founding engineer and senior architect. Prior to Onebox, Ross was at Sun Microsystems' JavaSoft division where he was a lead engineer with the Java Server Group responsible for network computer and embedded server products.

Dave Crane
Co-author of Bestselling AJAX Book 'AJAX in Action'
Dave will autograph his bestselling book for all delegates!

Dave Crane is senior developer and architect for Historic Futures Ltd., a UK firm specializing in web-based supply chain management solutions. He is also lead author of the bestselling 'AJAX in Action', and has been using the technologies that we now know as AJAX in production for several years. He has worked extensively in the IT industry over the last ten years, in areas as diverse as home automation, finance and national policy planning, for companies ranging from start-ups to FTSE 100 market leaders, using technologies as diverse as J2EE, Python, Ruby, PHP, Linux and .NET. He holds degrees in environmental science, parallel computing, and a Ph.D. in simulation modelling and philosophy. He is active as a writer, trainer and mentor in the AJAX commun

Christian Cantrell
Coauthor of JavaScript/Flash Integration Kit
AJAX/Flash Integration Guru

Christian Cantrell is a Product Manager for Developer Relations with Adobe and a big fan of AJAX. He has been developing large-scale, web-based applications in ColdFusion, Java, JSP, and Flash for more than six years. He is the author of numerous tutorials and white papers, and is coauthor of Flash Enabled: Flash Design & Development for Devices as well as the JavaScript/Flash Integration Kit. Christian is a leading expert on AJAX to Flash integration.

Sahil Malik
telerik Tech Evangelist, Microsoft MVP
Author of Bestselling ADO.NET 2.0 book

Sahil Malik is an independent consultant, trainer and mentor in various Microsoft Technologies and has been closely involved with telerik as a technology strategist. He has worked for many large clients across the globe including a good deal of Fortune 100 companies and US government organizations. He is currently leading the office of Emerging Technologies at a prominent government office where he is in charge of reviewing, assessing and recommending various technologies to support the organization. Malik frequently speaks on a variety of .NET related topics at local user group meetings and industry events. For his community involvement and contribution, he has been awarded the Microsoft MVP award. He can be reached at www.winsmarts.com.

Jouk Pleiter
Co-Founder & CEO of Backbase

Jouk Pleiter is the CEO of Backbase, a leader in the field of Rich Internet Applications and AJAX development software. Backbase's clients include ING, ABN AMRO, TNT, KPN, Comsys and Heineken. Backbase operates globally with offices in San Mateo (North America) and Amsterdam (Europe). Since 1995, Jouk has been an entrepreneur: he founded three successful software companies. Prior to Backbase, Jouk was part of the founding team at the web content management company Tridion, where he led the product management operations, and was driving the company's efforts to become a leader in the European WCM software market. Jouk previously was part of the founding team at the Interactive Agency Twinspark where he grew the company to a leading market position in Europe and was instrumental in the sale of Twinspark to Agency.com. He has an MBA from the University of Groningen.

Kevin Hakman
Director of Product Marketing for TIBCO
General Interface TIBCO Software

Kevin Hakman is the director of product marketing for TIBCO General Interface, the award winning AJAX and Rich Internet Application framework and toolkit. Kevin Hakman pioneered AJAX in the enterprise co-founding General Interface in 2001. Since that time General Interface (aka 'GI') has been powering Web applications that look, feel and perform like desktop applications, but run in the browser at Fortune 500 and U.S. Government organizations. General Interface was also the first to use its own toolkit to provide full visual tooling for AJAX when it released it's 2.0 Version in 2003. TIBCO acquired General Interface in 2004 to extend its vision for service oriented applications to the end user. Kevin is a contributor to the SOA Web Services Journal and the AJAX Developer's Journal.

Shanku Niyogi
Product Unit Manager of the UI Framework and Services Team Microsoft Corporation

Shanku is Product Unit Manager of the UI Framework and Services (UiFX) team, which is responsible for delivering high-productivity UI framework technologies for the .NET platform, including ASP.NET, Atlas, Windows Forms, and frameworks for smart clients. Prior to his current role, Shanku was Group Program Manager of the Web Platform and Tools team on the Whidbey release of ASP.NET and Visual Web Developer. Shanku joined Microsoft in 1998 as a developer, having spent several years shipping products in the Windows ISV industry. Shanku holds a Bachelor of Mathematics degree in Computer Science from the University of Waterloo.

What you'll walk away with...
A New understanding of AJAX and how to make it work for you, plus:

A) **Conference Proceedings Binder**
Full-conference proceedings in a self-contained commemorative binder.

B) ***Ajax in Action* Book**
A copy of the Best-Selling AJAX book by Dave Crane.

C) ***Real-World AJAX "Secrets of the Masters"* Book + DVD**
New AJAX book edited by Dion Hinchcliffe (Release Date: Summer 2006)

D) **Notebook**
Executive style keeps you looking sharp for note-taking wherever your business might take you!

E) **Computer Back-Pack**
A hip new way to conceal your laptop! The Urban Wonder Compu-Pack is an ingenious creation - a 600-denier polycanvas and nylon knapsack!

F) **T-Shirt**
100% preshrunk 6.1 oz. heavyweight cotton tagless shirt features shoulder-to-shoulder tape, and double-needle sleeves and bottom.

G) **Porcelain Mug**
Lightweight and durable, you can use it as a teacup, coffee mug or keep it as a pencil holder.

H) **Satin Silver Contemporary Pen**
Satin-silver plastic barrel with the look and feel of brass. Rubber comfort grip improves writing control and comfort.

I) **Real-World AJAX Seminar on DVD**
Watch complete seminar video and slide presentations (Release Date: Summer 2006)

For more information...
Call 201-802-3022 or email events@sys-con.com

SYS-CON EVENTS   **For more great events visit www.EVENTS.SYS-CON.com**

NOTE: SPEAKER LINE-UP SUBJECT TO CHANGE WITHOUT NOTICE
VISIT WWW.AJAXSEMINAR.COM FOR THE MOST COMPLETE UP-TO-DATE INFORMATION

Welcome to the Future of Video on the Web!

LIVE SIMULCAST!
AROUND THE WORLD ON SYS-CON.TV

iTVCON.com
INTERNET TV CONFERENCE & EXPO 2006

Coming in 2006 to New York City!

"Internet TV is wide open, it is global, and in true 'Web 2.0' spirit it is a direct-to-consumer opportunity!"

For More Information, Call 201-802-3023 or Email itvcon@sys-con.com

List of Topics:
> Advertising Models for Video-on-demand (VOD)
> Internet TV Commercials
> Mastering Adobe Flash Video
> How to Harness Open Media Formats (DVB, etc)
> Multicasting
> Extending Internet TV to Windows CE-based Devices
> Live Polling During Webcasts
> Video Press Releases
> Pay-Per-View
> Screencasting
> Video Search & Search Optimization
> Syndication of Video Assets
> V-Blogs & Videoblogging
> Choosing Your PVR
> Product Placement in Video Content
> UK Perspective: BBC's "Dirac Project"
> Case Study: SuperSun, Hong Kong

Welcome to the Future!
Did you already purchase your ".tv" domain name?
You can't afford not to add Internet TV to your Website in 2006!

2005 was the year of streaming video and the birth of **Internet TV**, the long-awaited convergence of television and the Internet. Now that broadband is available to more than 100 million households worldwide, every corporate Website and every media company must now provide video content to remain competitive, not to mention live and interactive video Webinars and on-demand Webcasts.

20 years ago the advent of desktop publishing tools opened the doors for the creation of some of today's well-known traditional print media companies as well as revolutionized corporate print communications. Today, with maturing digital video production, the advent of fully featured PVRs, and significant advances in streaming video technologies, **Internet TV** is here to stay and grow and will be a critical part of every Website and every business in the years to come.

It will also very rapidly become a huge challenge to network and cable television stations: **Internet TV** is about to change forever the $300BN television industry, too.

The Internet killed most of print media (even though many publishers don't realize it yet), Google killed traditional advertising models, and **Internet TV** will revolutionize television the way we watch it today. You need to be part of this change!

Jeremy Geelan
Conference Chair, iTVCon.com
jeremy@sys-con.com

PRODUCED BY
SYS-CON
EVENTS

Track 1	Corporate marketing, advertising, product and brand managers
Track 2	Software programmers, developers, Website owners and operators
Track 3	Advertising agencies, advertisers and video content producers
Track 4	Print and online content providers, representatives from traditional media companies, print and online magazine and newspaper publishers, network and cable television business managers

XML JOURNAL SUPPLEMENT

HOME

ENTERPRISE SOLUTIONS

CONTENT MANAGEMENT

DATA MANAGEMENT

XML LABS

This Month

Getting to Know XForms
Craig Caulfield

HTML forms are one of the best-known techniques for gathering data from a user and submitting that data to a server. However, HTML forms are only simple tools and don't natively support some of the features needed by current Web applications such as sophisticated data validation. Also, the user interface created by HTML forms is essentially hard coded for one device, meaning the same form can't be easily re-tasked for, say, PDAs or mobile phones.

Storing XML in Relational Database Management Systems
Selim Mimaroglu

When we need to store XML in a Relational Database Management System (RDBMS) we have to think of nonnative ways to store it, since the structure of XML does not fit the flat relations of the RDBMSs. Although some RDBMS vendors support XML storage and XQuery such as Oracle 10g (release 2), this support is not yet common. Some of the well-known RDBMSs are Oracle, DB2, Informix, Sybase IQ, SQL Server, MySQL, and PostgreSQL. If you have one of these products, or a similar product, you should be interested in reading this article.

Getting to Know XForms

Find out about XForms and why they are the perfect partner for Web services

XML-Based Interop, Close up

In addition to the strategy side of Web services, there is also the protocol-oriented side of things, the XML side. Embracing not only XML itself but also the full range of mainstream XML-based technologies like XPath, XSLT, XML Schema, and SOAP. *XML Journal* has been delivering insightful articles to the world of developers and development managers since the year 2000.

It is our privilege to bring XML-Journal directly to readers of Web Services Journal, and vice versa. Anyone already familiar with the Web services world of SOAP, UDDI, and WSDL will find here articles and features each month that will interest them – about the cutting-edge technologies and latest products that are changing not only our industry, but the way the world exchanges information. To make it easy for you to find your way around, we have four distinct 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 analysis

Getting to Know XForms

Find out about XForms and why they are the perfect partner for Web services

WRITTEN BY
Craig Caulfield

HTML forms are one of the best-known techniques for gathering data from a user and submitting that data to a server. However, HTML forms are only simple tools and don't natively support some of the features needed by current Web applications such as sophisticated data validation. Also, the user interface created by HTML forms is essentially hard coded for one device, meaning the same form can't be easily re-tasked for, say, PDAs or mobile phones.

The W3C XForms Recommendation is one way of addressing some of these issues. (XForms is both plural and singular, so there's no XForm, only XForms.) XForms are XML tags embedded in host documents such as XHTML that, when rendered by an XForms-aware browser, give applications some rich and dynamic capabilities such as:

- XForms can take advantage of the strong data typing offered by XML Schemas to validate user input at the client without using any scripting. More sophisticated data validation is also possible, such as enforcing relationships between different form values.
- XForms' user interface components are device-independent, meaning they are rendered according to whatever device they are being displayed on.
- XForms create and consume XML, rather than name/value pairs, making them the ideal client for Web services.

To show how some of these capabilities can be used, this article walks through the development of a simple XForms application that uses an Amazon Web service to query and display a typical book search.

Setting Up for XForms

To take advantage of the power of XForms, we first of all need something over and above the simple HTML forms processing model – an XForms engine. XForms engines are programs that are usually installed on the client side, for example

as browser plug-ins, and typically work in the following way:
- A request is made for a host document containing XForms markup by, for example, opening a document on a file system or issuing an HTTP GET.
- The XForms engine reads the markup and renders the form. Because XForms' user interface components are defined in an abstract way, the same XForms will be rendered in a way that is appropriate to the client device. Therefore, while a date input field might be rendered as a calendar picker on a desktop browser, on a more constrained device such as a PDA, the same input field may be rendered as a single-line text box.
- Once the form has been rendered, the users work with it to accomplish what they need to. As they are doing this, events will be fired to, amongst other things, enforce data validation rules, conditionally display different user interface components, or perform calculations. (XForms calculations and field updates are performed using the same depth-first search and topological sorting as spreadsheets.)
- When the user has completed the form, he or she clicks a button to submit it. The XForms engine will then create an XML instance document and deal with this payload according to the rules defined by the form. The target of a submission is always a URI, which may mean that the XML payload is written to a file, sent to an e-mail address, submitted to a Web service, or to any other HTTP or HTTPS endpoint.

The advantage of having the XForms engine on the client is that all processing happens in one place and unnecessary traffic with the server is avoided. However, until XForms implementations become more common, developers can't always assume that their clients will have the right installation. Alternatively, server-side XForms engines can deliver equivalent XForms functionality to plain clients, usually through a combination of scripting and the normal markup of the host language. This means that clients don't need their own XForms engine, but the markup delivered to the client may be bulky and may not support the full range of XForms features.

For the examples used in this article, I'll be using a client-side XForms engine called FormsPlayer (www.formsplayer.com/content/index.html). Sidebar 1 has more details about some of the XForms engines that are available today.

XForms Basics

XForms are not intended to be stand-alone XML documents, but rather are embedded in what is known as a host "language." Often, this host language will be XHTML, but it might also be Wireless Markup Language (WML) or Scalable Vector Graphics (SVG), among others. Within their host languages, XForms closely follow the Model View Controller (MVC) pattern, which means there is a clean separation of data, presentation, and logic.

The XForms equivalent of the MVC model is, appropriately enough, the model element. An XForms model is really a template for the XML payload that will eventually be created, and it has a dual role in loading the model with any initial data. Listing 1, a basic Hello World XForms, shows a simple model inside the head of an XHTML document. Models are typically placed in the head to emphasize the fact that they are a non-rendered component. Any well-formed XML document can be placed within the model's instance sub-element, or it can contain a pointer to an external instance document.

One of the key advantages XForms have over HTML forms is the ability to perform data validation, constraint checking, and calculations without the need for client-side scripting. XForms allows this to be done in a number of ways.

- An XML Schema can be associated with the instance data elements. For small models, these constraints and validations can also be specified inline rather than referring to a separate XML Schema.
- The bind element can be used to establish a data binding between user interface controls and elements in the instance document and at the same time specify a data type for the instance element. The bind element also contains any calculations or restrictions (as XPath expressions) that need to be applied.

The last part of the XForms model is the submission element, which uses a URI to describe what should be done with the XML payload once it has been populated. In our Hello World example, the XML payload is simply written out to a named file when the user clicks on the Write to Disk button.

Once the model is defined, all or part of it is exposed through user interface components. Following the MVC pattern, XForms user interface components provide a view onto the model, with the two being linked

"One of the key advantages XForms have over HTML forms is the ability to perform data validation, constraint checking, and calculations without the need for client-side scripting"
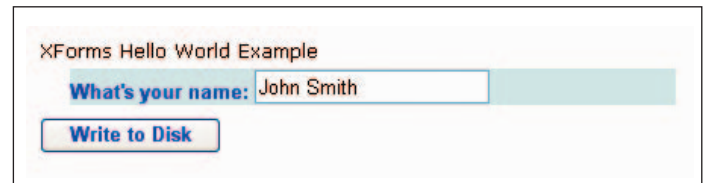

Figure 1 • A simple Hello World XForms

by either XPath expressions as in Listing 1, or through a bind element. Whereas HTML defines explicit user interface components such as radio buttons and check boxes, XForms defines a set of *abstract* components. These abstract components say what a component should do, but the actual rendering will depend on the host language and the device on which the components will be displayed.

While the XForms engine takes care of most of the rendering of the XForms user interface, developers can still have a style influence in a couple of ways. XForms can be directly styled according to the features available in the host language, so when using XHTML, this might mean using tables, headings, and other XHTML tags, with the XForms components arranged within these. However, by hard coding the styling details in this way, the device-independence of the form is limited.

The alternative styling approach is to use only the minimum host-language structure necessary to accommodate the XForms markup. Again, when using XHTML, this would mean placing the XForms model in the head section and the abstract user interface definition and bindings in the body section. An external stylesheet, such as Listing 2, would then provide the colors, fonts, and other formatting for each particular device.

The final part of the MVC pattern, the controller, equates to XForms events and actions. Events in XForms are defined by the XML Events specification, which in turn provides an element-based interface to the Document Object Model (DOM) Level 2 event syntax (see the Resources for more details). Meanwhile, XForms actions are the handlers that respond to XForms events. This means that XForms follow the well-known Gang of Four Observer pattern in which observers are attached to particular elements, which are notified when nominated events occurs, and then certain actions are performed.

XForms events and actions can be combined to do some of the following:

- Display a message to the user with varying degrees of intrusiveness ranging from modal dialog boxes to fleeting tool tips
- Change some value in the instance document
- Dynamically insert or delete line items in tables
- Navigate to a new URI, possibly in a new window and possibly replacing the active form
- Submit all or part of an XML instance document
- Toggle between different component renderings to implement multi-stage forms or tabbed applications

Introduction to Amazon Web Services

To demonstrate some of the features of XForms and how well suited they are to Web services, I've created a simple form (Listing 3) that uses an Amazon.com Web service to provide a simple book search facility.

As some background, Amazon Web services were launched in mid-2002 as a way of allowing their business partners to create their own content-rich applications based on Amazon's product and pricing information and e-commerce model. Amazon now offers a range of Web services, including access to Web site information and traffic history charts (Alexa

Figure 2 • The Amazon book search criteria

Figure 3 • The Amazon search results

Web Information Service) and a distributed message queue service (Amazon Simple Queue Service). Check out the Amazon Web service references in the Resources section for details about all of Amazon's Web services.

For the XForms here, I'll be using the primary Amazon E-Commerce Service (ECS), version 4.0. The ECS exposes 18 methods that allow developers to search for Amazon and third-party product and seller information; details about customer content such as reviews, wish lists, and listmania lists; and to manage shopping carts of products for purchase through Amazon. If you would like to implement this XForms, you will need to create a free Amazon Web services account. Besides giving you access to forums, blogs, code samples, and a technical library, you will also receive an Access ID Key, which is a 20-character code that must be included as a parameter for each Web service call to uniquely identify the caller.

While it's free to use the ECS, there is a restriction that requests should be limited to no more than one per second. The licensing agreement gives the fine detail and other restrictions.

Building and Running the Web Service Client

Creating an XForms Web service client involves defining a SOAP payload as the XForms model, populating this via the XForms user interface controls, and then submitting the payload to a Web service endpoint. When the Web service response is received it can be processed using XPath or XSLT.

Listing 3 shows an XForms model that contains a SOAP payload that initiates one of the most common ECS requests, an item search. Besides the mandatory Amazon Access Key ID, nearly all of the parameters for an item search are optional, so only the absolute minimum needs to be submitted. For this particular XForms, users will be able to search Amazon for books based on keyword, author, and title criteria, so these elements are included in the SOAP payload. The SearchIndex element contains a value of Books meaning the search will target only books, but this could easily be broadened to cover all product types. In addition, the results of a search can be sorted according to relevance, price, or title. This sorting happens on Amazon's side before the Web service response is returned, so it's one fewer processing activity for the client. The Availability, Condition, and MerchantID have been pre-filled with values that should return the widest range of search items. Finally, the ResponseGroup indicates what and how much information will be returned. For an item search and a ResponseGroup value of Medium, this will include full product and pricing details, links to images, and editorial reviews.

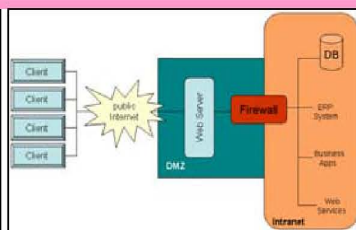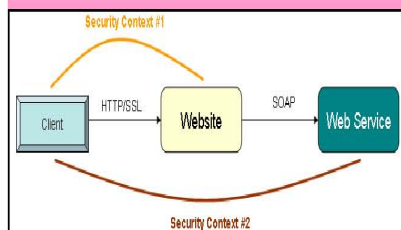Figure 2 shows the XHTML document in Listing 3 open in Internet Explorer.

The search criteria entered by the user populates the XForms model (the SOAP payload) by way of XPath expressions in the body of the XHTML document. Following recommended XForms practice, only enough host language markup is used to correctly position the

The 2006 Web Services Security Conference & Exhibition
May 25-26 Marriott Conference Center, University of Maryland, College Park, Maryland

THE INN AND CONFERENCE CENTER
UNIVERSITY OF MARYLAND UNIVERSITY COLLEGE
Marriott Conference Centers

Get the Details and Register at: http://www.unatek.com/conference.htm or Call 301-583-4629

XML Gateway
High performance for XML traffic

Keynotes and Power Panels' Topics Include: State-of-the-Art in Intrusion Prevention, XML Cutting-Edge Products, SOA Web Services, Single Sign On Authentication, Identity, Data Theft, Risk Management and Legal Issues, *And Many More!!!*

ABOUT THE CONFERENCE

The 2006 Web Services Security Conference (WSSC) is the forum for senior executives, industry and government leaders, and legal and academic experts to present the state-of-the-art of online transactions security practice. With the mix of strong developer, infrastructure and messaging content, WSSC 2006 is the principal educational conference for all those who engage in online transactions and a technical training conference for a wide range of professionals.

In addition to the in-depth breakout sessions, attend a selection of Hands-on demos, Chalk-&-Talks, Panel Discussions and more!

WHO SHOULD ATTEND?
- Business and Corporate Executives, CIOs and CTOs, Financial and Banking Professionals, Government Representatives
- Technical Decision Makers, IT Managers, Network and System Administrators, Developers, Software Architects, Web Development Managers, Security Specialists, and anyone looking to stay current with the latest in Web services and online transactions security.

FUN AND INFORMATIVE NETWORKING EVENTS
- Birds-of-Feather Gatherings, Roundtables and Panels
- Expo Floor Opening Party & the Conference Party

POWER SPEAKERS

Melise R. Blakeslee
Partner, McDermott Will & Emery LLP

Dr. Anton Chuvakin
Chief Strategist, netForensics, Inc.

John Starke
VP, *TransGlobal Business Systems*

Kerry G. (Kwasi) Holman
President / CEO, PGCEDC

Dr. Charles Iheagwara
CTO, Unatek, Inc

Ralph Bazilio
President, TCBA

Mamoon Yunus
Founder and CTO, Forum Systems, Inc.

Andre Yee
President/CEO, NFR

Martin Roesch
Founder and CTO, Sourcefire, Inc.

Dr. Ephraim Feig
CTO, Kintera, Inc.

Donald Debolt
Director, Computer Associates

Register Sponsor Panels Speakers Agenda Exhibit Conference Call for paper Hotel Info

Sponsors: **Unatek   Forum Systems   netForensics   Transglobal   SOAWebServicesJournal + More!**

XForms components, and any styling is handled by an external cascading style sheet (Listing 2).

Once the XForms model has been populated, it is be submitted to the ECS endpoint (shown in the submission element) just like any other Web service payload. The submission element wraps around a toggles element, which switches to the response user interface when the Web service response is ready (Figure 3).

Again, XPath is used, this time with output elements, to extract the necessary information from Web service response payload for display to the user. Notice the repeat element in the responseGUI case element: this simply iterates over a homogeneous collection of nodes from the Web service response, in this case creating a table-like display.

The Search Again button then resets the XForms and the search can be performed again.

Conclusion

HTML forms have long been a standard part of Web development, but they are showing their age when confronted with omnipresent XML, the almost irresistible rise of Web services, and more demanding user-interface requirements.

The simple Web service client demonstrated here only hints at the advantages XForms offer over HTML forms. In reality, XForms are a powerful tool for building device-independent, declarative, form-based applications that don't need scripting or programming to deliver rich user interfaces. More important, when compared to other ways of creating Web service clients such as using the SOAP APIs, the

Sidebar 2: XForms Development Environments

Because XForms are plain XML, they can be developed with any text or XML editor but, as always, an intelligent development environment will make the job much easier.

- XFormation (www.xformation.com) is a sophisticated visual XForms development environment created in partnership with the developers of the FormsPlayer engine.
- OpenOffice (www.openoffice.org) and StarOffice (www.sun.com/software/star/staroffice/index.jsp) have quietly provided XForms tools for some time. These tools are both forms designers and form engines.
- IBM XForms Generator (www.alphaworks.ibm.com/tech/xfg) is an Eclipse 3.1 plug-in that generates XForms and optional JSP response handlers directly from XML instance documents and Web Service Description Language (WSDL) documents. Also, check out IBM's developerWorks (www-136.ibm.com/developerworks/xml) Web site for a good range of articles and tutorials on XForms.
- Microsoft InfoPath 2003 (www.microsoft.com/office/infopath/prodinfo/default.mspx). InfoPath is an odd addition to a list of XForms development tools because it isn't really XForms compliant. Still, it is an XML-based forms processor and it comes with the usual strong Microsoft WYSIWYG design interface. It's an excellent tool for creating form interfaces to document-style Web services. If you use Microsoft Office 2003, you will probably already have it.

As with the XForms engines, keep an eye on the W3C Web site for movement.

"XForms are a powerful tool for building device-independent, declarative, form-based applications that don't need scripting or programming to deliver rich user interfaces"

XForms alternative is concise and comes with a built-in means of getting a window onto the request and response payloads.

Even so, creating XForms is not as simple as defining tags in an XHTML page. The development model provides a sometimes daunting range of possibilities, and until XForms engines are more widely supported, developers need to critically consider how they will deliver XForms functionality to their clients.

Still, the benefits of using XForms more than balance this effort.

References

- *Amazon Web Services:* www.amazon.com/gp/aws/landing.html
- Dubinko, M. *XForms Essentials.* O'Reilly & Associates, 2003.
- Halstead, P., Blain, M., and Manava-Teredesai, V. *Developing Solutions with Microsoft InfoPath.* Microsoft Press, 2005. If you are looking at a range of form-based technologies besides XForms, it might be worth evaluating InfoPath 2003 to see if it fits your requirements.
- Levitt, J. *The Web Developer's Guide to Amazon E-Commerce Service.* J9T Publications, 2005. Until recently, Amazon's Web services have been a work-in-progress, with significant changes made between releases. Thus, finding good examples and version-right information has been frustrating. Although written specifically for PHP developers, this book is still a fantastic reference for all developers looking for the fine details of using the Amazon Web services.
- Musciano, C. and Kennedy, B. *HTML and XHTML: The Definitive Guide*, 5th edition. O'Reilly & Associates, 2002.
- Raman, T. V. *XForms: XML Powered Web Forms.* Addison Wesley Professional. 2003.
- Simpson, J. E. *XPath and XPointer* O'Reilly & Associates, 2002.
- *W3C XForms 1.0 (Second Edition) Specification:* www.w3.org/TR/2005/PER-xforms-20051006
- *W3C XForms 1.1 Working Draft:* www.w3.org/TR/xforms11. Among the features planned for the next XForms version are utility functions for XPath expressions such as the Luhn algorithm for credit card number validation, and better support for interacting with Web services.
- *W3C XML Events Recommendation:* www.w3.org/TR/xml-events
- *W3C Document Object Model (DOM) Level 2 Events Specification:* www.w3.org/TR/2000/REC-DOM-Level-2-Events-20001113

AUTHOR BIO

Craig Caulfield is a senior software engineer for a defense and commercial software house in Perth, Western Australia. He has a Bachelors degree in Computer Science, a Masters degree in Software Engineering, and holds certifications in Java, XML, DB2, UML, MySQL, and WebSphere.

ccaulfi1@bigpond.net.au

```
<?xml version="1.0" encoding="utf-8"?>
<html xmlns:fp="urn:formsplayer.com" xmlns="http://
www.w3.org/1999/xhtml" xmlns:xforms="http://www.
w3.org/2002/xforms" xmlns:xsd="http://www.w3.org/2001/
XMLSchema" xmlns:ev="http://www.w3.org/2001/xml-
events">
 <head>
  <object width="0" height="0" id="FormsPlayer"
classid="CLSID:4D0ABA11-C5F0-4478-991A-375C4B648F58">
   <b>FormsPlayer has failed to load! Please check
your installation.</b>
   <br />
  </object>
  <?import namespace="xforms" implementation="#FormsPla
yer"?>
  <xforms:model id="default">
   <xforms:instance xmlns="">
    <data>
     <name/>
    </data>
   </xforms:instance>
   <xforms:submission action="output.xml" method="put"
id="submit" />
  </xforms:model>
  <link rel="stylesheet" href="style.css" type="text/
css" />
 </head>
 <body>
  <xforms:output>XForms Hello World Example</xforms:
output>
  <xforms:input ref="/data/name">
   <xforms:label>What's your name: </xforms:label>
   <xforms:hint>Your name</xforms:hint>
  </xforms:input>

  <xforms:group>
   <xforms:submit submission="submit">
    <xforms:label>Write to Disk</xforms:label>
   </xforms:submit>
  </xforms:group>
</body>
</html>
```

Listing 2: XForms Style Sheet (style.css)

```
body {
  padding-left: 6px;
  color: black;
  font-family: verdana, arial, sans-serif;
  font-size: 10pt;
  background-color: white;
}
xforms\:label {
  font-weight: bold;
  font-size: 10pt;
  color: #0000FF;
  font-family: sans-serif;
}
xforms\:group {
  display: block;
}
xforms\:input {
 display: block;
 margin-top: 4px;
 margin-bottom: 4px;
 margin-left: 20px;
 width: 400px;
 background-color: rgb(210,230,230);
}
xforms\:select1 {
```

```
 width: 400px;
 background-color: rgb(210,230,230);
 margin-top: 1px;
 margin-bottom: 4px;
 margin-left: 20px;
}
xforms\:repeat {
  margin: 5px;
  padding: 5px;
}
xforms\:trigger {
 margin-top: 4px;
}
xforms\:submit {
 margin-top: 4px;
}
.repeat-item {
 display: list-item;
 width: 660px;
 border-top: 1px thin solid gray;
 border-bottom: 1px thin solid gray;
}
.repeat-index {
 background-color: lightblue;
}
.invalid {
 background-color: #FFFF66;
 width: 500px;
}
```

Listing 3: Amazon Web Service
XForms (AmazonXForms.html)

```
<?xml version="1.0" encoding="utf-8"?>
<html xmlns:fp="urn:formsplayer.com" xmlns="http://www.
w3.org/1999/xhtml"
              xmlns:xforms="http://www.w3.org/2002/
xforms"
            xmlns:ev="http://www.w3.org/2001/xml-
events"
            xmlns:soap="http://schemas.xmlsoap.org/
soap/envelope/"
            xmlns:tns="http://webservices.amazon.com/
AWSECommerceService/2005-10-05">
 <head>
  <title>Amazon XForms Client</title>

  <!-- I'm using FormsPlayer as the XForms client
engine. Have to make sure it's installed first. -->
  <object width="0" height="0" id="FormsPlayer"
classid="CLSID:4D0ABA11-C5F0-4478-991A-375C4B648F58">
   <b>FormsPlayer has failed to load! Please check
your installation.</b>
   <br />
  </object>
  <?import namespace="xforms" implementation="#FormsPla
yer"?>

  <!--  The main form payload. It is this SOAP enve-
lope that the XForms populates and submits to Amazon
-->
  <xforms:model id="SOAPEnvelope" functions="fp:
HTMLserialise">
   <xforms:instance id="AmazonRequestPayload">
    <soap:Envelope xmlns:soap="http://schemas.xmlsoap.
org/soap/envelope/"
          xmlns:tns="http://webservices.amazon.com/
AWSECommerceService/2005-10-05"
            xmlns:xs="http://www.w3.org/2001/
XMLSchema">
     <soap:Body>
      <tns:ItemSearch>
       <!-- Insert your own Amazon Access Key ID
```

```
here. -->
        <tns:AWSAccessKeyId>01234567890ABCDEFGHI</tns:
AWSAccessKeyId>
          <tns:Shared>

            <tns:Availability>Available</tns:
Availability>
            <tns:Author>Dan Brown</tns:Author>
            <tns:Condition>All</tns:Condition>
            <tns:Keywords></tns:Keywords>
            <tns:MerchantId>All</tns:MerchantId>
            <tns:ResponseGroup>Medium</tns:ResponseGroup>
            <tns:SearchIndex>Books</tns:SearchIndex>
            <tns:Sort>relevancerank</tns:Sort>
            <tns:Title></tns:Title>
          </tns:Shared>
        </tns:ItemSearch>
      </soap:Body>
    </soap:Envelope>
  </xforms:instance>
  <xforms:submission method="post" mediatype="text/
xml" replace="instance"
           action="http://soap.amazon.com/onca/soap?Ser
vice=AWSECommerceService"
              id="AmazonRequest">
    <xforms:toggle case="responseGUI" ev:event="xforms-
submit-done" />
  </xforms:submission>
  </xforms:model>

  <!-- A model for internal use only. Populates the
sort list-box with standard values.  -->
  <xforms:model id="SortOptionsData">
    <xforms:instance>
     <options>
      <sortOption description="Relevance">relevancerank<
/sortOption>
      <sortOption description="Price">pricerank</sortOp-
tion>
      <sortOption description="Title">titlerank</sortOp-
tion>
     </options>
    </xforms:instance>
  </xforms:model>

  <link href="style.css" rel="stylesheet" />

 </head>
 <body>
  <xforms:switch id="switchView">
   <xforms:case id="requestGUI">

    <xforms:output>Amazon Book Search Criteria</xforms:
output>

    <!-- Gather the book search criteria from the user
and populate the linked elements in the model -->
    <xforms:input ref="instance('AmazonRequestPayload')
//tns:Title">
      <xforms:label>Title:     </xforms:label>
      <xforms:hint>Enter a title.</xforms:hint>
    </xforms:input>

    <xforms:input ref="instance('AmazonRequestPayload')
//tns:Keywords">
      <xforms:label>Keywords: </xforms:label>
      <xforms:hint>Enter some keywords.</xforms:hint>
    </xforms:input>

    <xforms:input ref="instance('AmazonRequestPayload')
//tns:Author">
      <xforms:label>Author:     </xforms:label>
      <xforms:hint>Enter an authors name.</xforms:hint>
    </xforms:input>
```

```
    <!--  Amazon will sort the data for us before
returning the results. -->
    <xforms:select1 ref="instance('AmazonRequestPayload
')//tns:Sort" appearance="minimal">
      <xforms:label>Sort by: </xforms:label>
      <xforms:itemset nodeset="sortOption"
model="SortOptionsData">
          <xforms:value ref="." />
          <xforms:label ref="@description" />
      </xforms:itemset>
    </xforms:select1>

    <xforms:group>
     <xforms:submit submission="AmazonRequest">
      <xforms:label>Search</xforms:label>
     </xforms:submit>
    </xforms:group>

   </xforms:case>

   <!-- If the request is valid, display what Amazon
has found. -->
   <xforms:case id="responseGUI">

    <xforms:group>
     <xforms:output>Search Results</xforms:output>
    </xforms:group>

    <xforms:repeat nodeset="*//tns:Item"
id="searchItems">
        <xforms:group>
          <xforms:output ref="tns:ItemAttributes/
tns:Author">
            <xforms:label>Author: </xforms:label>
          </xforms:output>

          <xforms:output ref="tns:ItemAttributes/
tns:Title">
            <xforms:label> Title: </xforms:label>
          </xforms:output>
          <xforms:output value="concat(tns:
OfferSummary/tns:LowestNewPrice/tns:CurrencyCode,
            tns:OfferSummary/tns:
LowestNewPrice/tns:FormattedPrice)">
            <xforms:label> Price: </xforms:label>
          </xforms:output>

          <xforms:output ref="tns:ItemAttributes/
tns:ISBN">
            <xforms:label> ISBN: </xforms:label>
          </xforms:output>

        </xforms:group>
    </xforms:repeat>

    <xforms:group>
     <xforms:output value="concat('Showing the first
', count(*//tns:Item), ' items.')" />
    </xforms:group>

    <xforms:group>
     <xforms:trigger>
        <xforms:label>Search Again</xforms:label>
        <xforms:toggle ev:event="DOMActivate"
case="requestGUI" />
     </xforms:trigger>
    </xforms:group>

   </xforms:case>

  </xforms:switch>
 </body>
</html>
```
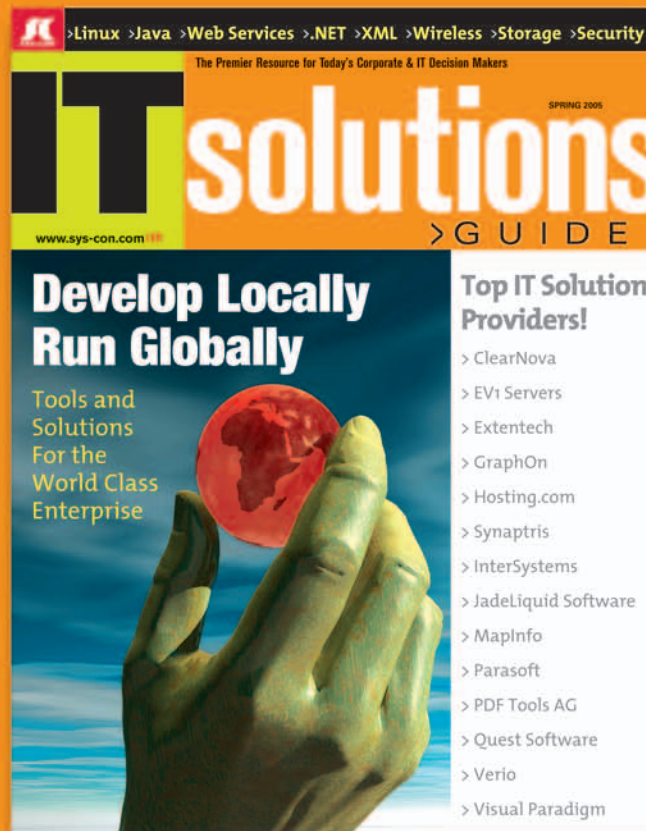
WRITTEN BY **SELIM MIMAROGLU**

Storing XML in Relational Database Management Systems

Using the Edge table approach

When we need to store XML in a Relational Database Management System (RDBMS) we have to think of nonnative ways to store it, since the structure of XML does not fit the flat relations of the RDBMSs. Although some RDBMS vendors support XML storage and XQuery such as Oracle 10g (release 2), this support is not yet common. Some of the well-known RDBMSs are Oracle, DB2, Informix, Sybase IQ, SQL Server, MySQL, and PostgreSQL. If you have one of these products, or a similar product, you should be interested in reading this article.

There are several ways to store XML in an RDMS. One possibility is to store XML in big text columns such as CLOB (Character Large Object). Some advantages of storing XML as CLOB are: there is no preprocessing, and the originality of the XML document is preserved. Unfortunately, this method has some major disadvantages – neither XQuery nor SQL will work on the text columns. We would like to query the stored data, because reading the whole document and extracting needed parts manually is simply impractical. We would like to find ways to store XML in a database that would allow us to query the data easily. In this article I am going to show you a method that is known as *shredding XML into the edge table*. This is a very general method that will work on any XML document regardless of having a schema.

XML has a lot of bells and whistles, but it can be viewed simply as a tree. It's possible to represent the XML nodes as vertices of the tree and the relations between the XML nodes as the edges. Let's look at an example of how we can take an XML fragment and represent it as a tree. Figure 1 shows a simple XML document and Figure 2 shows its tree representation. The root of the XML document is the root of the tree too. Note that in XML order is very important, therefore when we create a tree of XML we have to capture the order, which is done in Figure 2. For example, the attribute *isbn* comes before the element *title*, which comes before the element *authors*.

Figure 2 shows the elements and attributes in ovals, and their corresponding values in rectangular boxes. The tree you see in Figure 2 is a nice representation of XML that makes it easier for us to gather the information presented in Figure 1 in a graphical way. The attribute

```
<book isbn='XY023'>
  <title>XML in Use</title>
  <authors>
    <author>Author1</author>
    <author>Author2</author>
  </authors>
</book>
```
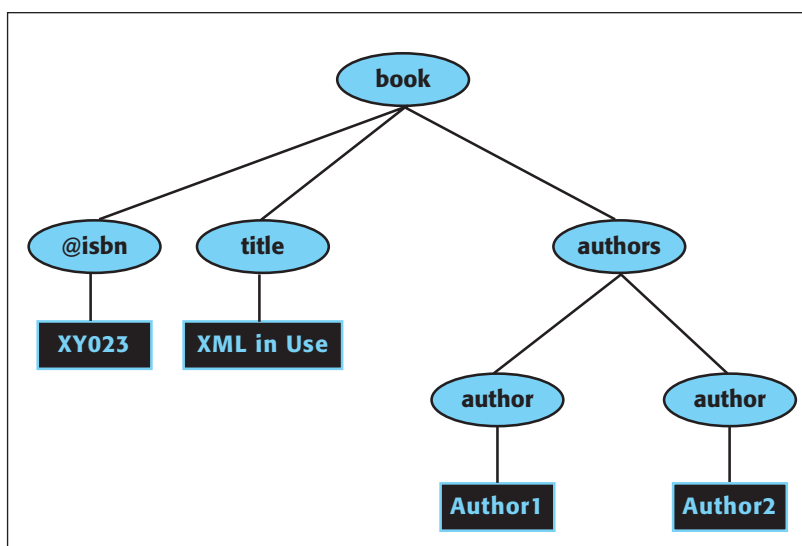
Figure 1 • Simple XML document

Figure 2 • Tree representation of the XML document in Figure 1

AUTHOR BIO

Selim Mimaroglu is a PhD candidate in computer science at the University of Massachusetts in Boston. He holds an MS in computer science from that school and has a BS in electrical engineering.

isbn carries a flag of @, which indicates that it is an attribute. Unfortunately, storing Figure 2 as a graph into a database isn't of much use either. We will *shred* the XML tree into pieces, each of which can be stored in the rows of a database table. However, in order to do this we first have to number all of the vertices of the tree as shown in Figure 3. Numbering is essential for capturing the parent-child and ancestor-descendant relations, which will be used for answering XML queries.

Numbering an XML Tree

We will number the vertices in a top-down, left-to-right fashion using *preorder traversal*, which is numbering a vertex when we see it for the first time and then repeating this process until all vertices are numbered.

The red dashed line shows how to number the XML tree. It traverses the tree in a top-down and left-to-right fashion. Note that the root element *book* has the lowest number in the tree. The numbers of *@isbn*, *title*, and *authors* are 2, 4, and 6, which in fact capture the order of these nodes.

Edge Table

Now that we have numbered the XML tree, we can store it in the *Edge* table. An *Edge* table has *id, parentid, name, value,* and *lastdesc* attributes (columns) as shown in Figure 4.

id is the identification number of a vertex (element, attribute, text box, etc.). Identification numbers are shown in Figure 3. *id* is generally integer type.

parentid is the parent's identification number. For example, the parentid of the *title* element is 1 (same for *@isbn* and *authors*). The type integer is the usual choice.

name is the name of an element or an attribute. Text nodes (rectangular boxes) do not have names, but we name them as *text* for better performance. *name* can be char or varchar type. Data type char stands for character, and varchar stands for variable character. Element tag names and attribute names are generally
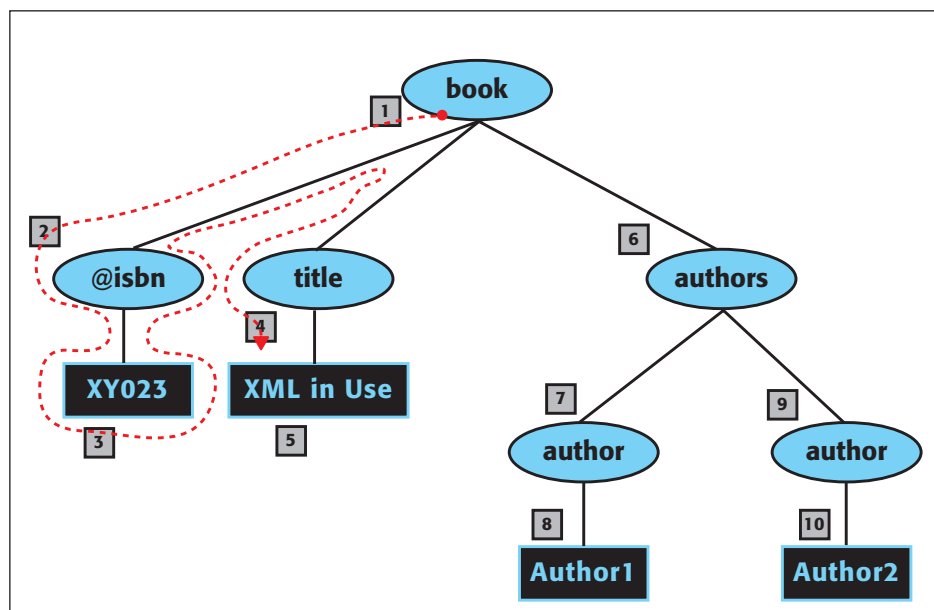
> ## "XML has a lot of bells and whistles, but it can be viewed simply as a tree"

Figure 3 • Numbering tree of Figure 2

short, usually not more than 50 characters.

value is the value of an element or an attribute. The content of this column is generally larger than the other columns. In most cases using varchar is the most efficient solution. For example in Figure 1, the value of the title element, <title>XML in Use</title>, has the largest character content: a total of 10 characters.

For simplicity, in this article I have been using a small XML sample (and I will continue using the same XML below). For interested readers, I have provided much larger XML samples at www.cs.umb.edu/~smimarog/xmlsample. For these examples I set the *value* type to varchar(550) in my database. varchar(550) is space saver compared to char(550). varchar(550) indicates that the maximum length of the characters will be no more than 550, but when it is less than 550, say 35, then the remaining 515 characters will not be wasted. However when you have the type char(550) and put content into it that has only 35 characters, unfortunately 515 characters will be wasted.

lastdesc is the number that holds the last descendant of a node. Figure 3 shows that *lastdesc* of *book* is 10, the *lastdesc* of *title* is 5, etc. We are going to use the last descendant information for answering XPath queries that contain // (double slash). Remember that // is the shortcut of *descendant-or-self::node()*. I will show how this is used in an example later in this article.

id	parentid	name	value	lastdesc

Figure 4 • Empty *Edge* table

Parsing XML

Parsing XML for storage into an RDBMS is known as *shredding*, as I mentioned earlier. There are several dozens of XML parsers available, which can be categorized into two main groups: SAX (Simple API for XML) parsers, and DOM (Document Object Model) parsers. SAX is the gold standard of XML APIs. SAX implementations are fast and memory efficient. SAX is available for most of the popular programming languages such as Java, C++, Visual Basic, Python, and Perl. Java comes with a built-in SAX parser. Apache's Xerces (http://xerces.apache.org) provides a SAX parser for C++, Java, and Perl.

SAX is an event-driven API. It notifies the user of various events when parsing the XML document in top-down fashion, such as startDocument, endDocument, startElement, endElement, processingInstructions, etc. It's the programmer's responsibility to take action upon notification of these events. In Java this is done by implementing the ContentHandler interface or by extending the DefaultHandler class. SAX parsers check for well-formedness and optionally for validity. In Java the errors and the warnings are provided by the ErrorHandler interface. The parser will send a "Fatal Error" when it encounters a well-formedness error,

id	parentid	name	value	lastdesc
1	0	book	null	10
2	1	@isbn	null	3
3	2	text	XY023	3
4	1	title	null	5
5	4	text	XML in Use	5
6	1	authors	null	10
7	6	author	null	8
8	7	text	Author1	8
9	6	author	null	10
10	9	text	Author2	10

Figure 5 • Edge table storing shredded XML of Figure 3

and an "Error" for all other errors.

Let's look at one of the events closely:

```
public void startElement
        (String namespaceURI,
         String localName,
         String qName,
         Attributes atts)
    throws SAXException
```

The SAX parser will invoke this event at the beginning of every element. The SAX parser provides the namespace URI, the local name, the qualified name (qName), and the list of attributes of this element. This event is accompanied with the endElement event when the parser reaches the end of this element, but before invoking the endElement the parser will provide the contents of the element in order.

Xerces-J is my personal favorite. I used it for parsing the sample XML file shown in Figure 1. You might use a DOM parser instead of a SAX parser. A DOM parser will scan the XML and construct the object model for the whole document in the memory, creating a structure similar to Figure 2. Generally speaking, it's more convenient to work with DOM parsers, because they make the whole object model of the XML available in the memory. For large documents this is a problem because an object models of a big XML document may not fit into the main memory, therefore I prefer working with SAX parsers.

Querying XML Using SQL

Most of the RDBMSs do not support XQuery or XPath. In this section, I will show you how to use SQL for responding to some XPath queries. After shredding the XML document we have the table shown in Figure 5.

In the following examples I will offer several XPath queries and convert them to SQL queries. Note that each parent-child or ancestor-descendant relation corresponds to a self join of the *Edge* table (that's joining *Edge* table with itself). The representation in Figure 5 is not the most space efficient one, but it can handle mixed XML content nicely. Although values of the attributes and elements do not have unique names, note that we named them as "text" for better performance.

```
<book isbn='XY023'>
<title>XML in Use</title>
<authors>
  <author>Author1</author>
  <author>Author2</author>
</authors>
</book>
```

Example 1: Find the title of the book.

```
XPath:
/book/title/text()

SQL:
select
 a.value
from
 Edge a, Edge b, Edge c
where
 a.name = 'text' and
 a.parentid = b.id and
 b.name = 'title' and
 b.parentid = c.id and
 c.name = 'book' and
 c.id = 1
```

There are two self joins for two parent-child relations: the first one is between "title" and its text, and the second one is between the root element "book" and "title."

Example 2: Find all of the authors of the book.

```
XPath:
/book/authors/author/text()

SQL:
select
 a.value
from
 Edge a, Edge b, Edge c, Edge d
where
 a.name = 'text' and
 a.parentid = b.id and
 b.name = 'author' and
 b.parentid = c.id and
 c.name = 'authors' and
 c.parentid = d.id and
 d.name = 'book' and
 d.id = 1
```

Example 3: This example is similar to Example 2. It demonstrates the use of lastdesc column. It skips the "authors" element by using // (double slash) instead. Find all of the authors of the book.

```
XPath:
/book//author/text()

SQL:
select
 a.value
from
 Edge a, Edge b, Edge c
where
 a.name = 'text' and
 a.parentid = b.id and
 b.name = 'author' and
 b.id > c.id and b.id <= c.lastdesc
and
 c.name = 'book' and
 c.id = 1
```

Note that in this example we have b.id greater than c.id and less than or equal to c.lastdesc. That's because all of the descendants of c (root "book") will be between c.id and c.lastdesc.

Acknowledgements

smimarog@cs.umb.edu

HOME

ENTERPRISE SOLUTIONS

CONTENT MANAGEMENT

DATA MANAGEMENT

XML LABS