



## **Web 2.0? It's the Universal SOA**

**David S. Linthicum**

With all of the talk about Web 2.0, I was wondering where the heck that term came from. The term Web 2.0 was originally coined by O'Reilly's Dale Dougherty to describe the forces behind the huge post-dot-com success of Internet companies like Google, eBay, Amazon, and iTunes, as well as noncommercial, emergent Web-born creations such as Wikipedia and BitTorrent. Clearly, the Web has become the platform of choice for many organizations, and with success of SaaS (Software as a Service) companies such as Salesforce.com and NetSuite, this is quickly becoming true. Also, we are clearly moving from dynamic visual interfaces to dynamic non-visual services, all accessible over the Internet. All of this is Web 2.0, or better yet the Universal SOA.

So, what does this mean to those standing up SOAs today? It's clear that many of the services we consume and manage going forward will be services that exist outside of the enterprise, such as subscription services from guys like Salesforce.com, or perhaps emerging Web services marketplaces from guys like StrikeIron. This is outside-in SOA, in essence reusing service in an enterprise not created by that enterprise, much as we do today with information on the Web. Thus, those services outside of the enterprise existing on the Internet create a Universal SOA, ready to connect to your enterprise SOA, perhaps providing more value. This is nothing new, by the way, we been talking Universal SOA for some time now, at least the notion, we are just seeing bits and pieces appearing today.

So, how do you prepare yourself? I have a few suggestions:

- First, accept the notion that it's okay to leverage services that are hosted on the Internet as part of your SOA. Normal security management needs to apply, of course.
- Second, create a strategy for the consumption and management of outside-in services, including how you'll deal with semantic management, security, transactions, etc.
- Finally, create a proof of concept now. This does a few things including getting you through the initial learning process and providing proof points as to the feasibility of leveraging outside-in services.

Other, more technical issues you need to consider:

- Consider your Firewall Issues. Look at how easily you can consume services from sources that are outside of your enterprise, and what has to be done to make this possible without compromising security. Typically you'll have to upgrade your infrastructure to allow for outside-in Web services.
- Consider Connectivity. So, how do you make Web services known to your SAP applications? How about your legacy systems? You need a strategy there, and connectivity is not as easy as it seems. You have to account for the differences in interfaces and protocols.
- Consider Semantics. So, how do you account for the differences in applications semantics within the consumed services, and your native application semantics? The answer is an integration server or appliance. Do you have one, and is it the right one?
- Consider Governance. You need a governance strategy to make sure the services you leverage are secure, and won't become another problem. Remember, we're attempting to solve problems here, not create new ones.
- Consider Change. This is a fundamental change in the way things are done in most enterprises, and now is the best time to change both understanding and attitude towards this huge, dare I say it, paradigm shift. Most enterprise architects and developers I run into can't comprehend that services from the internet could replace applications developed internally. However, those that embrace this first will have a huge market advantage. This will prove out as we move forward. Will you be ready?