WOA: A New Enterprise Partner for Linked Data

by Mike Bergman - Sunday, October 12, 2008

http://www.mkbergman.com/459/woa-a-new-enterprise-partner-for-linked-data/

Web-Oriented Architecture (and REST) is Gaining Enterprise Mindshare

Nick Gall, a VP of Gartner, first coined the TLA (three-letter acronym) for WOA (Web-oriented architecture) in late 2005. In further describing it, Nick simply defines it as:

WOA = SOA + WWW + REST

In the longer version, Nick describes WOA as based on the architecture of the Web that he further characterizes as "globally linked, decentralized, and [with] uniform intermediary processing of application state via self-describing messages."

WOA is a subset of the service-oriented architectural style. He describes SOA as comprising discrete functions that are packaged into modular and shareable elements ("services") that are made available in a distributed and loosely coupled manner.

Representational state transfer (REST) is an architectural style for distributed hypermedia systems such as the World Wide Web. It was named and defined in Roy Fielding's 2000 doctoral thesis; Roy is also one of the principal authors of the Hypertext Transfer Protocol (HTTP) specification.

REST provides principles for how resources are defined and used and addressed with simple interfaces without additional messaging layers such as SOAP or RPC. The principles are couched within the framework of a generalized architectural style and are not limited to the Web, though they are a foundation to it.

REST and WOA stand in contrast to earlier Web service styles that are often known by the WS* acronym (such as WSDL, etc.). (Much has been written on RESTful Web services *v.* "big" WS*-based ones; one of my own postings goes back to an interview with Tim Bray back in November 2006.)

While there are dozens of well-known methods for connecting distributed systems together, protocols based on HTTP will be the ones that stand the test of time. And since HTTP is the fundamental protocol of the Web, those protocols most closely aligned with its essential nature will likely be the most successful.

-- Dion Hinchliffe [2]

Shortly after Nick coined the WOA acronym, REST luminaries such as Sam Ruby gave the meme some airplay [1]. From an enterprise and client perspective, Dion Hinchliffe in particular has expanded and written extensively on WOA. Besides his own blog, Dion has also discussed WOA several times on his Enterprise Web 2.0 blog for ZDNet.

Largely due to these efforts (and -- some would claim -- the difficulties associated with earlier WS* Web services) enterprises are paying much greater heed to WOA. It is increasingly being blogged about and highlighted at enterprise conferences [3].

While exciting, that is not what is most important in my view. What is important is that the natural connection between WOA and linked data is now beginning to be made.

Analogies to Linked Data

Linked data is a set of best practices for publishing and deploying data on the Web using the RDF data model. The data objects are named using Web uniform resource identifiers (URIs), emphasize data interconnections, and adhere to REST principles.

Most recently, Nick began picking up the theme of linked data on his new Gartner blog. Enterprises now appreciate the value of an emerging service aspect based on HTTP and accessible by URIs. The idea is jelling that enterprises can now process linked data architected in the same manner.

I think the similar perspectives between REST Web services and linked data become a very natural and easily digested concept for enterprise IT architects. This is a receptive audience because it is these same individuals who have experienced first-hand the challenges and failures of past hype and complexity from non-RESTful designs.

It helps immensely, of course, that we can now look at the major Web players such as Google and Amazon and others -- not to mention the overall success of the Web itself -- to validate the architecture and associated protocols for the Web. The Web is now understood as the largest Machine designed by humans and one that has been operational every second of its existence.

Many of the same internal enterprise arguments that are being made in support of WOA as a service architecture can be applied to linked data as a data framework. For example, look at Dion's 12 Things You Should Know About REST and WOA and see how most of the points can be readily adopted to linked data.

So, enterprise thought leaders are moving closer to what we now see as the reality and scalability of the Web done right. They are getting close, but there is still one piece missing.

False Dichotomies

I admit that I have sometimes tended to think of enterprise systems as distinct from the public Web. And, for sure, there are real and important distinctions. But from an architecture and design perspective, enterprises have much to learn from the Web's success.

With the Web we see the advantages of a simple design, of universal identifiers, of idempotent operations, of simple messaging, of distributed and modular services, of simple interfaces, and, frankly, of openness and decentralization. The core foundations of HTTP and adherence to REST principles have led to a system of such scale and innovation and (growing) ubiquity as to defy belief.

So, the first observation is that the Web will be the central computing touchstone and framework for all computing systems for the foreseeable future. There simply is no question that interoperating with the Web is now an enterprise imperative. This truth has been evident for some time.

But the reciprocal truth is that these realities are themselves a direct outcome of the Web's architecture and basic protocol, HTTP. The false dichotomy of enterprise systems as being distinct from the Web arises from seeing the Web solely as a phenomenon and not as one whose basic success should be giving us lessons in architecture and design.

Thus, we first saw the emergence of Web services as an important enteprise thrust -- we wanted to be *on* the Web. But that was not initially undertaken consistent with Web design -- which is REST or WOA -- but rather as another "layer" in the historical way of doing enterprise IT. We were not *of* the Web. As the error of that approach became evident, we began to see the trend toward "true" Web services that are now consonant with the architecture and design of the actual Web.

So, why should these same lessons and principles not apply as well to data? And, of course, they do.

If there is one area that enterprises have been abject failures in for more than 30 years it is data interoperability. ETL and enterprise busses and all sorts of complex data warehousing and EAI and EIA mumbo jumbo have kept many vendors fat and happy, but few enterprise customers so. On almost every single dimension, these failed systems have violated the basic principles now in force on the Web based on simplicity, uniform interfaces, etc.

The Starting Foundation: HTTP 1.1

OK, so how many of you have read the HTTP specifications [4]? How many understand them? What do you think the fundamental operational and architectural and design basis of the Web is?

HTTP is often described as a communications protocol, but it really is much more. It represents the operating system of the Web as well as the embodiment of a design philosophy and architecture. Within its specification lies the secret of the Web's success. REST and WOA quite possibly require nothing more to understand than the HTTP specification.

Of course, the HTTP specification is not the end of the story, just the essential beginning for adaptive design. Other specifications and systems layer upon this foundation. But, the key point is that if you can be cool with HTTP, you are doing it right to be a Web actor. And being a cool Web actor means you will meet many other cool actors and be around for a long, long time to come.

Concept "Routers" for Information

An understanding of HTTP can provide similar insights with respect to data and data interoperability. Indeed, the fancy name of linked data is nothing more than data on the Web done right -- that is, according to the HTTP specifications.

Just as packets need their routers to get to their proper location based on resolving the names of a URI to a physical device, data or information on the Web needs similar context. And, one mechanism by which such context can be provided is through some form of logical referencing framework by which information can be routed to its right "neighborhood".

I am not speaking of routing to physical locations now, but the routing to the logical locations about what information "is about" and what it "means". On the simple level of language, a dictionary provides such a function by giving us the definition of what a word "means". Similar coherent and contextual frameworks can be designed for any information requirement and scope.

Of course, enterprises have been doing similar things internally for years by adopting common vocabularies and the like. Relational data schema are one such framework even if they are not always codified or understood by their enterprises as such.

Over the past decade or two we have seen trade and industry associations and standards bodies, among others, extend these ideas of common vocabularies and information structures such as taxonomies and metadata to work across enterprises. This investment is meaningful and can be quite easily leveraged.

As Nick notes, efforts such as what surrounds XBRL are one vocabulary that can help provide this "routing" in the context of financial data and reporting. So, too, can UMBEL as a general reference framework of 20,000 subject concepts. Indeed, our unveiling of the recent LOD constellation points to a growing set of vocabularies and classes available for such contexts. Literally thousands and thousands of such existing structures can be converted to Web-compliant linked data to provide the information routing hubs necessary for global interoperability.

And, so now we come down to that missing piece. Once we add *context* as the third leg to this framework stool to provide semantic grounding, I think we are now seeing the full formula powerfully emerge for the semantic Web:

SW = WOA + linked data + *coherent context*

This simple formula becomes a very powerful combination.

Just as older legacy systems can be exposed as Web services, and older Web services can be turned into WOA ones compliant with the Web's architecture, we can transition our data in similar ways.

The Web has been pointing us to adaptive design for both services and data since its inception. It is time to finally pay attention.

[1] Sam and his co-author Leonard Richardson of RESTful Web Services (O'Reilly Media Inc., 446 pp, May 2007;

ISBN 0596529260) have preferred the label ROA, for Resource-oriented Architecture.

[2] D. Hinchcliffe, "A Search for REST Frameworks for Exploring WOA Patterns -- And Current Speaking Schedule", Sept. 10, 2006; see http://hinchcliffe.org/archive/2006/09/10/9275.aspx

[3] The Linked Data community should pay much closer attention to existing and well-attended enterprise conferences in which the topic can be inserted as a natural complement rather than trying to start entire new venues.

[4] The current specification is RFC 2616 (June 1999), which defines HTTP/1.1; see http://tools.ietf.org/html/rfc2616 . For those wanting an easier printed copy, a good source in PDF is http://www.faqs.org/ftp/rfc/rfc2616.pdf.

PDF generated by *AI3:::Adaptive Information* blog