

Weighing SOA and Web Services: the IFs, ANDs, and BUTs, Part I

If you thought Y2K was expensive, just imagine a change to a postal code structure. Consider these scenarios when employing these system designs.

By Naomi Bloom

Service Oriented Architecture (SOA) and its essential building blocks, web services, are a great idea in systems design and software development with significant potential business benefits. Business applications are faster to market, less costly to develop and maintain, more robust, less costly to acquire and operate, and far more adaptable to changing business needs using SOA and web services.

But as so often happens in the world of software, these business benefits ONLY accrue if web services and SOA are done well, which means:

- The defined web services are sufficiently granular, well-documented, properly version controlled, etc. to be usable and are re-used everywhere that's relevant;
- The defined web services are sufficiently robust to meet the needs of all users without modification and are easily modified as business needs change;
- The data and process semantics of the defined web services are agreed upon by all users or there are agreed upon industry standard semantics;
- The underlying business applications are rebuilt, from the ground up, upon models of the domain, which have been designed expressly to recognize those objects that are great candidates for re-use and reincarnation as web services; and
- There is sufficiently powerful foundational software (often called middleware) to orchestrate the merger of web services to fulfill a business process, using the right, effective-dated versions of the applicable web services without undue system or human overhead.

And did I mention that each web service must be wrapped with the organizational, geographic, and demographic applicability rules (e.g., this web service is only applicable to those parts of our sales organization that operate without a physical work location) and then wrapped further with their effective dates? Or for maximum benefit in applying SOA and web services to a BPO platform, the domain model and the web services must be designed from the start for multi-tenancy?

Fortunately, as a buyer/licensor/subscriber of HRM software (however delivered), or of BPO services, it's not your job to figure out how to do all of this. But it is impor-

tant that you assess how well your BPO provider and—where commercial software is used—their ERP/HRMS vendor are using web services and SOA. That assessment is relatively easy to do if you choose the right scenarios and insist on scenario testing the relevant software or service delivery. So here is one of Naomi's killer SOA and web services scenarios to get your started.

■ **Killer scenario No. 1: the postal code problem.** In this scenario, the U.S. zip code structure, which is now five numeric characters as the base code, a hyphen, and four numeric characters as the refinement code, is changed on short notice to become eight alphanumeric characters, a percentage sign, and then six more alphanumeric characters. If the underlying HRM software represents a rigorous SOA and web services implementation, then there will be only one web service—manage intergalactic addresses—that will have to be redone and regression tested not only to structure the data in this new way but also to include updated reporting formats, data capture, presentation formats, edit rules (e.g., pointing to a new URL for those edits), etc. In this desired outcome, which could be demonstrated while you wait, all those places in the software that need this web service would simply get the new one. Done!

If a rigorous SOA and web services implementation has not been done, then there may be a much bigger web service (e.g., manage all intergalactic person contact information) that will have to be redone and regression tested, thus introducing much greater costs, elapsed times, and possibilities for errors. Also, the relevant web service may not handle all the relevant pieces such as structuring the data as well as reporting formats, data capture and presentation formats, edit rules (e.g. pointing to a new URL for those edits), etc., so that there are multiple web services to be updated and regression tested. The web services overhaul of the software might not be complete so that changes must be made in the shiny new web service part of the software but also in the not-yet-overhauled procedural logic part of the software. And these are just a few of the potential problems with vendor's introduction of SOA and web services. See my column next month for more SOA and web services killer scenarios. **HRO**



Naomi Lee Bloom,
Managing Partner, Bloom
& Wallace, may be
reached at 239-454-7305
or [naomibloom@
mindspring.com](mailto:naomibloom@mindspring.com).