

SOI vs. SOA

Alejandro Schwed



En este artículo revisaremos las principales diferencias entre estos dos enfoques.

Durante años de trabajo en implementaciones de arquitecturas empresariales para grandes clientes, hemos identificado que en el mercado existe bastante confusión sobre lo que es una implementación de una arquitectura basada en servicios (SOA) y lo que se ha denominado como Integración basada en Servicios (SOI, Service Oriented Integration).

Mirada a SOI

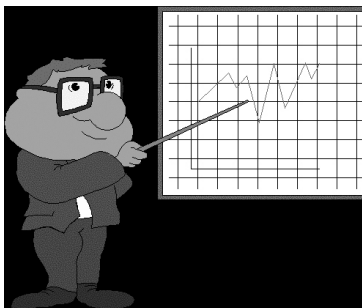
La Integración basada en Servicios o SOI, aunque es una aproximación a la visión SOA, no cumple con todas las características de este tipo de arquitectura. En SOI, aunque se utilizan profusamente los servicios,

éstos, son utilizados para enviar información entre aplicaciones y no para ofrecer funcionalidad explícita a los procesos de negocio.

Es decir, los servicios SOI siguen siendo interfaces de envío de información como las que estábamos acostumbrados a ver en el mundo EAI (Enterprise Application Integration), sólo que ahora, bajo un formato estandarizado (por ejemplo, SOAP).

En contraposición, en SOA lo que se debe ofrecer como servicio es una funcionalidad específica, aunque la misma contenga un intercambio de datos (por ejemplo, unos parámetros

de entrada y unos datos de salida). Es así, como en SOA puro, se debe eliminar el envío de datos de una aplicación a otra para que ambas contengan en sus bases de datos propias información idéntica sobre un mismo objeto de negocio.



Un ejemplo práctico

Para facilitar la diferencia, usemos un ejemplo práctico. En una empresa cualquiera, hoy en día, se envía constantemente información de clientes entre varios sistemas. Podríamos lograr una primera aproximación a servicios automatizando estos intercambios y colocándolos bajo una interfaz estándar XML y SOAP.

Sin embargo, esta aproximación, sería una Integración basada en Servicios (SOI). Par lograr una verdadera solución 100% SOA se debería tener una aplicación maestra de clientes que ofrezca servicios de consulta de datos del cliente, de actualización de datos del cliente, etc. y las demás aplicaciones (que no contendrían datos de cliente),

utilizarán estos servicios dentro de sus procesos.

Por supuesto, esta última visión es bastante utópica ya que la mayoría de las empresas ya cuentan con una serie de aplicaciones cuyo funcionamiento depende de poseer en sus propias bases de datos la información de los clientes en mayor o menor grado.

Una mezcla

Es así como la mayoría de las soluciones de hoy en día están compuestas más bien de una mezcla entre SOI y SOA, donde si bien existe un único servicio de actualización de datos del cliente, por ejemplo, el mismo va a las diferentes aplicaciones que contienen estos datos y los actualiza cada vez que surge una modificación. Esto implica un intercambio de datos que se parece más a SOI que a SOA, sin embargo, también ofrece una funcionalidad única y reutilizable que podría argumentarse como una implementación SOA.

En términos propios de SOA, el desacoplamiento de SOI es menor que el logrado en servicios SOA.

¿Qué quiere decir esto?

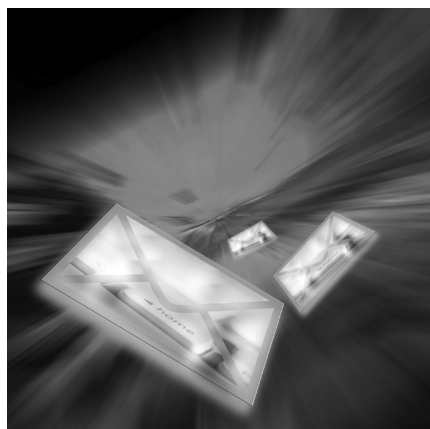
Simplemente que al integrar datos de aplicaciones (enviar datos de un lado a otro), todavía no se logra un desacopla-



miento total. Cada servicio SOI estará circunscrito a las limitaciones que tengan las aplicaciones que generen y reciban los datos.

Adicionalmente, en muchas ocasiones se sufrirá el efecto del Mínimo Común Denominador o MCM (para los matemáticos, lo anterior no es un error de tipografía) donde el servicio SOI se verá limitado a la menor característica de las aplicaciones que atienda.

Por ejemplo, volviendo al caso de un servicio de actualización de datos del cliente, si una de las aplicaciones sólo puede recibir tres dígitos como campo de identificación del cliente, la opción de implementación más simple sería limitar este campo a tres dígitos aunque otras aplicaciones puedan manejar más dígitos o incluso caracteres alfanuméricos (claro, también existe la opción de implementar reglas complejas de transformaciones para sobrepasar el efecto MCM).



En SOA

En contraposición, en una implementación pura SOA, la arquitectura debe estar totalmente desacoplada. Es decir, ninguna aplicación debe poseer “datos” propios de otra aplicación, sino que el proceso basado en servicios debe ser el que consulta y actualiza los datos donde sea necesario.

Dada esta situación, el efecto MCM se minimiza y se logran procesos más flexibles y más eficientes.

En este contexto, uno podría preguntarse cómo lograr una implementación enteramente desacoplada.

La respuesta pasa por utilizar una de las herramientas del mundo SOA más complejas en elaborar, no por la complejidad de sus formas, sino por la abstracción que requiere en su elaboración. Esta herramienta es el diagrama de dominios de gestión.

En este diagrama, se busca representar la información que gestiona una empresa (o que gestionará, en el caso de una empresa nueva) mediante constructos denominados dominios.

Para cada proceso de la empresa, se debe estudiar la relación de información que hay entre los dominios y es así cómo se identifican los servicios 100% SOA que una empresa debe o debería tener.

Conclusiones

Cómo se habrá podido deducir, en conclusión, una arquitectura basada en servicios pura, va mucho más allá de revisar y re-organizar los procesos de integración de datos de una empresa.

Una arquitectura SOA pura, debe incluso llegar a re-organizar las funcionalidades de las aplicaciones. Sin embargo, muchas aplicaciones utilizadas por las empresas ya están desarrolladas de una forma que su funcionamiento depende de tener datos directamente en bases de datos propias de la aplicación, y cambiarlas, requeriría involucrar a los proveedores de estas herramientas para que modifiquen sus aplicación.

Por supuesto, esto resulta casi imposible en cualquier organización en marcha, entender la visión utópica de SOA, permite tener un “norte” que nos ayuda a tomar decisiones mucho más robustas arquitectónicamente en proyectos de implementación

Referencias

[1] <http://www.slideshare.net/slaha-nas/services-soa-oriented-integration-soi-279154>

[2] <http://www.ebizq.net/webinars/6798.html>

[3] <http://www.itbusinessedge.com/cm/blogs/lawson/the-limitations-of-service-oriented-integration/?cs=31250>



Alejandro Schwed. Es el director de consultoría de la empresa CTP. Tiene más de 10 años de experiencia en implementación de herramientas tecnológicas en grandes empresas, con principal énfasis en el sector de telecomunicaciones y de banca y finanzas. Ha sido parte de proyectos de implementación en diferentes roles, lo que le ha permitido tener una visión tanto funcional como técnica de estas implementaciones. Hoy en día, coordina un equipo de más de 60 consultores que se enfocan en la implementación de soluciones tecnológicas de punta, que ayuden a sus clientes a lograr ventajas competitivas en el corto y mediano plazo. Posee un Magister en Gerencia de Sistemas de Información y un MBA de Boston University.