

## SOA: ¿una moda o algo más?

“La arquitectura orientada a servicios es tanto un marco de trabajo para el desarrollo de software como un marco de trabajo de implantación. Para que un proyecto SOA tenga éxito los desarrolladores de software deben orientarse ellos mismos a esta mentalidad de crear servicios comunes que son orquestados por clientes o middleware para implementar los procesos de negocio. El desarrollo de sistemas usando SOA requiere un compromiso con este modelo en términos de planificación, herramientas e infraestructura.” (Wikipedia)

La integración de aplicaciones independientemente de la tecnología empleada no es una idea nueva, cuya aproximación más notable fue, quizá, CORBA. Lógicamente para que esta integración sea posible debe ser una integración débil, de forma que los componentes no queden atados el uno al otro.

Recientemente con las mejoras producidas en las telecomunicaciones y los mercados cada vez más exigentes, esta arquitectura de aplicaciones ha cobrado más fuerza, primero porque es técnicamente posible, y segundo, pero más importante, porque permite a las empresas adaptarse a las nuevas características del entorno (tecnológicas y de mercado), lo que ha venido a denominarse por algunos como empresas ON-DEMAND.

Los elementos básicos que intervienen en una arquitectura SOA son:

➔ Web Service: colección de protocolos y estándares (OASIS y W3C) orientado al intercambio de información entre aplicaciones.

➔ Servicio: un servicio es una funcionalidad diseñada, codificada y encapsulada de forma que sea reutilizable por otros servicios y aplicaciones, normalmente recubiertos por un Web Service. No es importante la tecnología con la que está construido.

➔ Agentes: proveedor del servicio y consumidor del Servicio.

Asimismo, con la generación de nuevos y nuevos servicios web, surge la necesidad de organización y conocimiento: ¿qué servicios puedo utilizar? ¿dónde y cómo puedo utilizarlos? Quiero cambiar uno de mis servicios, ¿quién lo usa? Por todo ello se crearon los servicios de Registro, que habitualmente se ha implementado con el estándar de OASIS UDDI.



Mientras se acordaban estos estándares y se realizaban las primeras aproximaciones, también surgió la necesidad de delegar la comunicación entre todos los servicios, así como de encapsular las funcionalidades entre otras de enrutado y transformación, en un producto específicamente diseñado para ello, lo que ha venido a denominarse como ESB (Enterprise Service BUS), que se convierte en una de las piezas más importantes de la arquitectura, ya que evita que cada servicio tenga que preocuparse de cómo comunicarse con el resto.

Como complemento a esta arquitectura se han definido elementos que aportan nuevas funcionalidades como monitorización de procesos, servidores de reglas de negocio, y sobre todo motores de procesos,... Estos últimos han cobrado un especial auge con la aparición de nuevos estándares como BPEL, que permite orquestar la ejecución de diferentes servicios Web de forma que se puedan implementar los procesos de negocio de la organización de una forma rápida y flexible, mejorando el ROI respecto a enfoques mas tradicionales. Sus principales características son:

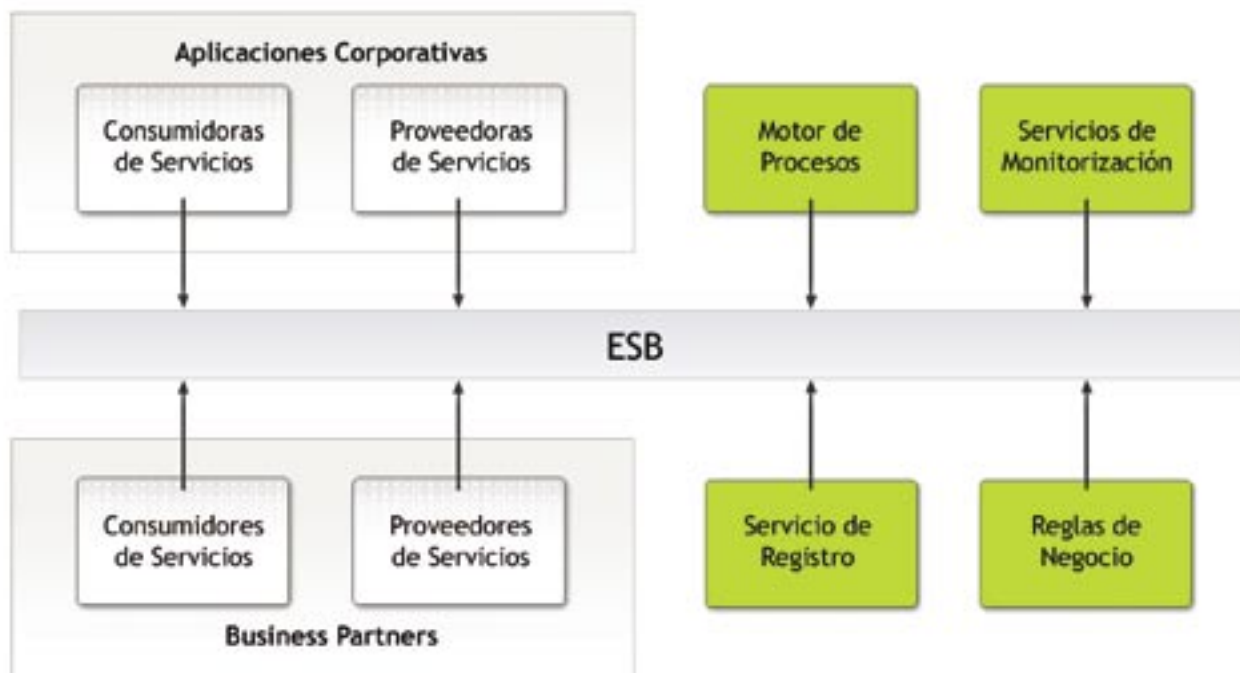
- ➔ Independencia tecnológica de la implementación de cada servicio: tecnologías Java, Microsoft,...
- ➔ Basado en estándares abiertos: XML, SOAP, WSDL,... y utilizados por los principales fabricantes de software Oracle, IBM, BEA, Microsoft,...
- ➔ Orquestación de ejecuciones distribuidas, lo que facilita la escalabilidad, mantenibilidad, rendimiento,...

➔ Componentes avanzados de workflow como ramas paralelas, timers, interacción humana, gestión de excepciones,...

➔ Permite las "transacciones empresariales" ya que a menudo un proceso de negocio engloba varias aplicaciones de diferentes organizaciones.

En resumen, se trata de un paso mas en la evolución del diseño del software. Se empezaron construyendo aplicaciones monolíticas de costosa integración. Mas tarde se dividieron las aplicaciones en capas de forma que se mejoraba su mantenimiento e integración. El último paso, de momento, ha sido aplicar el mismo enfoque de capas pero no dentro de las aplicaciones sino a toda la organización.

*Andeka Puente,  
Director de Proyectos*



Arquitectura orientada a Servicios