

Universidad Latinoamericana de Ciencia y Tecnología



Facultad de Ingeniería

Escuela de Ingeniería Informática

**Trabajo final para optar al grado de licenciado en
Ingeniería Informática con énfasis en Desarrollo de
Software**

Tema:

**Adopción de una Arquitectura Orientada a Servicios
(SOA): retos, estrategias y mejores prácticas**

Estudiante: Luis Andrés Vásquez Badilla

Cédula: 2-599-473

Profesor: Lic. Miguel Pérez Montero

I Cuatrimestre del 2008

Índice

Resumen.....	iii
Abstract.....	v
Palabras clave.....	vi
Introducción	1
Entendiendo SOA	3
Los principios fundamentales de SOA	8
Beneficios e implicaciones.....	12
El punto de partida	17
Definiendo servicios.....	22
Consideraciones para la implementación.....	27
Conclusiones.....	33
Referencias bibliográficas	35

Adopción de una Arquitectura Orientada a Servicios (SOA): retos, estrategias y mejores prácticas

Luis Andrés Vásquez Badilla¹

Resumen

La arquitectura orientada a servicios (SOA, por sus siglas en inglés) es un concepto relativamente nuevo que está adquiriendo un gran auge en el ámbito de las tecnologías de información. Este concepto se fundamenta en la implementación de un diseño arquitectónico de software basado en elementos modulares, independientes y desacoplados llamados 'servicios', los cuales representan una función repetitiva del negocio.

El éxito de SOA en los últimos años se debe principalmente a que se enfoca en varios problemas clave que las organizaciones están enfrentando. Uno de los principales beneficios que SOA provee consiste en la capacidad que brinda para adecuar rápidamente el software existente a los rápidos giros estratégicos y, de esta manera, responder de una manera más ágil a los requerimientos cambiantes del negocio e influir de manera positiva en la agilidad empresarial y su competitividad. Por otra parte, el diseño modular y desacoplado de SOA permite reducir la curva de costos asociados a los cambios

¹ Bachiller en Informática Empresarial y candidato a licenciado en Ingeniería Informática con énfasis en Desarrollo de Software, ULACIT. Correo electrónico: lvasquezb@gmail.com

futuros en los sistemas de información. Así, la construcción de una arquitectura orientada a servicios facilita la modificación de los sistemas, disminuye el tiempo de construcción y ensamble de nuevos módulos, permite la interoperabilidad y distribución de los sistemas e incrementa la reutilización, entre algunos de sus beneficios.

Sin embargo, el proceso de adopción de una arquitectura orientada a los servicios tiene grandes costos asociados. Las dificultades que las empresas enfrentan en el camino de adopción de SOA incluyen retos en varios ámbitos, principalmente en las áreas de gobernabilidad, configuración, administración de los metadatos, seguridad, monitoreo e interoperabilidad. Por ello, la identificación y ejecución de una estrategia adecuada de adopción de SOA resulta primordial para garantizar el éxito del proyecto.

En este sentido, este documento identifica y define una serie de aspectos clave para la implementación apropiada de una arquitectura orientada a servicios. Estas mejores prácticas incluyen aspectos relacionados con estandarización, principios fundamentales de la arquitectura, consideraciones técnicas de integración, mecanismos para dar valor agregado a la arquitectura y otros elementos que deben acompañar el proceso de planificación. Todas estas consideraciones constituyen una guía para atacar de manera efectiva los retos más comunes, presentes en el camino de adopción de SOA.

Abstract

The Service Oriented Architecture (SOA) is a relatively new concept that is becoming more important everyday in information technologies. The foundation of this concept is the implementation of an architectural software design, based on modular, independent, and loosely-coupled pieces of software, called services, which represent a repetitive business task.

The success of SOA in recent years has to do with its focus on a variety of problems organizations are currently facing. One of the major benefits that SOA provides is the ability of quickly adapting the existing software to the strategic changes of the company, and thus, responds in a more flexible manner to business requirements, improving agility and competitiveness. Furthermore, the modular and detached design of an SOA can reduce costs associated with future changes in information systems. Hence, the construction of service oriented architecture facilitates the modification of the systems, decreases the time of construction and assembly of new modules, facilitates the interoperability and distribution of systems, and increase reuse, among some of its benefits.

However, the process of SOA adoption has a lot of costs. The difficulties companies go through in their way to SOA include challenges in several areas of this doctrine, primarily in the areas of governance, configuration, metadata management, security,

monitoring and interoperability. Therefore, identification and implementation of a proper strategy of SOA adoption becomes essential to ensure the success of the project.

In this sense, this document identifies and describes a set of key areas for the proper implementation of service oriented architecture. These best practices deal with standardization, fundamental principles of the architecture, technical considerations for integration, mechanisms to give added value to the architecture and elements of the planning process. All these considerations are supposed to work as a guideline to tackle the most common problems that surround the adoption of SOA.

Palabras clave

Arquitectura orientada a servicios, SOA, estrategia, mejores prácticas.

Introducción

El mundo de los negocios está cambiando aceleradamente, y las empresas aprovechan cada vez más la tecnología para obtener la eficiencia que les permite sobrevivir en un ambiente competitivo. Las tecnologías de información brindan a las empresas la capacidad para automatizar sus procesos de negocio, con el fin de lograr así una mayor eficiencia en su operación; sin embargo, una de las principales causas de atrasos en la implementación de los cambios del negocio, es la dificultad para adaptar una plataforma de software compleja a estos giros estratégicos.

Por otra parte, los cambios acelerados, producto de la globalización y los avances tecnológicos, están provocado un gran impacto en el comportamiento de las relaciones *business-to-business* (empresa-empresa) y *business-to-customer* (empresa-cliente). Esto ha generado una necesidad constante de realizar cambios en las tecnologías de información que soportan los procesos de negocio respectivos.

Adicionalmente, a lo largo de los años, las empresas han implementado diferentes arquitecturas de información y estilos de programación de software, lo que ha provocado una divergencia de estándares en los sistemas de información, que hacen difícil compartir datos y lógica entre ellos.

La arquitectura orientada a servicios (SOA por sus siglas en inglés) consiste en un estilo de diseño arquitectónico de software que se basa en la utilización de servicios independientes con funciones específicas, los cuales pueden ser reutilizados y orquestados para satisfacer las necesidades de cada proceso de negocios.

SOA permite a las empresas (entre otros beneficios) agilizar los cambios en la plataforma de software, para responder a los frecuentes giros organizacionales, promoviendo la reutilización de datos y de lógica de negocio.

Por el valor que SOA provee, este nuevo concepto es en este momento uno de los mayores factores en los que las grandes y medianas empresas están invirtiendo importantes cantidades de tiempo y dinero. No obstante, la implementación de este tipo de arquitectura es muy compleja. Las dificultades enfrentadas durante el proceso de adopción de SOA son grandes debido, entre otras cosas, a que la variación del paradigma es sustancial y a la cantidad de cambios requeridos en aspectos, no sólo de informática, sino de toda la organización.

El presente artículo pretende ser una guía para las empresas que desean implementar una arquitectura orientada a servicios o que se encuentran en sus etapas iniciales, al presentar una síntesis de recomendaciones y consideraciones importantes para enfrentar los retos que se deben afrontar al adoptar una arquitectura de este tipo.

En las primeras secciones se comentan los antecedentes de SOA, así como sus beneficios, sus principios esenciales y algunos conceptos necesarios para la comprensión del tema. Posteriormente, se presenta una síntesis de recomendaciones para iniciar con un proyecto de implementación de SOA y para la definición de los servicios. Por último, se describen los aspectos que se consideran los factores claves que se deben tomar en cuenta durante la implementación de la plataforma de SOA.

Entendiendo SOA

La arquitectura orientada a servicios es un concepto relativamente nuevo. Las primeras definiciones de SOA datan del año 1996 (Schulte y Natis, 1996) y fue alcanzando su apogeo de manera progresiva desde entonces. El concepto de SOA fue adquiriendo mayor importancia a medida que las necesidades de flexibilidad, facilidad de mantenimiento, integración, reutilización de lógica y datos, y adaptación de la plataforma de tecnologías de información a los procesos de negocio se hacían más evidentes en las empresas.

Para entender mejor el concepto, debe primero comprenderse el fundamento base de esta arquitectura: los servicios. En el contexto empresarial, el concepto de servicios es utilizado muy frecuentemente desde hace varios años, ya que permite una mejor flexibilidad y eficiencia en su operación. Dejando de lado el ámbito

de las tecnologías de información, las empresas utilizan servicios en casi todos los procesos. Estos servicios son proporcionados por alguna entidad (proveedor del servicio) interna o externa a la organización, la cual estará encargada de entregar al solicitante (consumidor del servicio) el insumo necesario para continuar la operación. De esta manera, las empresas utilizan servicios de banca, facturación, contabilidad, entrega de paquetes, de limpieza, de información de crédito y de seguridad, entre muchos otros.

Desde este enfoque, los procesos de negocios se encuentran formados por una serie de pasos en los que interactúan varios servicios, cada uno de ellos con una función específica o una serie de funciones relacionadas. De esta forma se logra convertir trabajos complejos en la suma de servicios más simples. En este contexto, puede afirmarse que un servicio es una tarea repetitiva dentro de un proceso de negocio (High, Kinder y Graham, 2005). Estos mismos fundamentos son utilizados en el diseño de arquitecturas orientadas a servicios.

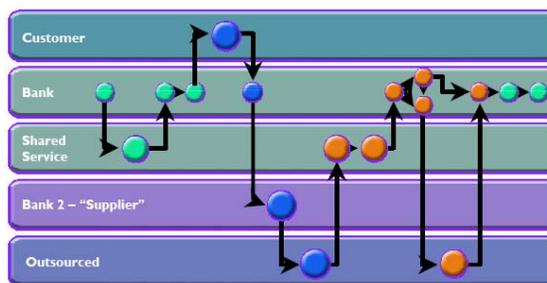


Ilustración 1: Proceso de negocios compuesto por servicios.

Fuente: High et al., 2005.

Ahora bien, en el ámbito de las tecnologías de información (TI), se podría definir un servicio como la abstracción de una función auto-contenida, con una interfaz claramente definida, que puede ser reutilizada por distintos consumidores. La definición de servicios en el contexto de SOA debe considerar varios principios que serán explicados más adelante.

Inicialmente, SOA fue influenciado por el concepto de Servicios Web (*Web Services*), al ser estos las primeras implementaciones de servicios en el ámbito del software. Sin embargo, se debe tener claro que una arquitectura SOA no implica la implementación de servicios web, y la implementación de servicios web no necesariamente implica una arquitectura SOA.

Existen varias definiciones de SOA. Una de ellas, por ejemplo, nos habla de SOA como un diseño arquitectónico modular para las aplicaciones empresariales, que puede ser distribuido y que está basado en componentes compartidos y desacoplados (Schulte y Abrams, 2006). Otra definición, describe SOA como una arquitectura de sistemas distribuidos "multicapa", que encapsula las partes de la arquitectura de la aplicación como servicios (Legner, 2005).

Adicionalmente, se han definido varios meta-modelos o arquitecturas de referencia de SOA. Cada una detalla una serie de capas y define la interacción de sus componentes. Uno de los

modelos más aceptados define 5 capas esenciales: la capa de sistemas operacionales, la capa de componentes, la capa de servicios, la capa de coreografía de procesos del negocio y la capa de presentación, también llamada capa de consumidores (Arsanjani, 2004).

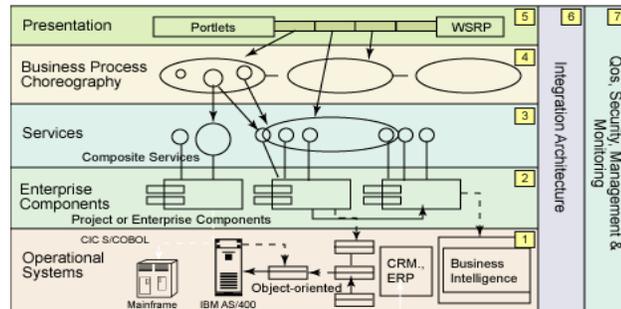


Ilustración 2: Las capas de SOA.

Fuente: Arsanjani, 2004.

La capa de sistemas operacionales incluye los sistemas legados, aplicaciones a la medida y los paquetes de software que han sido implementados en la empresa. Estos administran sus propios datos y serán incorporados por medio de técnicas de integración orientadas a servicios. La capa de componentes comprende unidades modulares de software que contienen la implementación de los servicios.

Por otra parte, la capa de servicios abarca todos los servicios que han sido definidos en la arquitectura SOA, y que representan una abstracción de alguna función o un conjunto de funciones del negocio.

Cada uno de estos servicios posee una interfaz expuesta preferiblemente con una tecnología estándar e independiente de la plataforma. Estos servicios pueden ser entidades individuales o combinadas para formar un servicio compuesto.

Las siguientes capas no son en sentido estricto parte de la implementación básica de SOA; sin embargo, tienen gran importancia para una implementación integral, que permita sacar el mejor provecho a una arquitectura de este tipo. Así, la capa de procesos del negocio se encuentra relacionada con la disciplina BPM (*Business Process Management*), ya que esta comprende las representaciones de los procesos del negocio como una orquestación o secuencia de los servicios expuestos en la capa inferior. Esta capa juega un papel importante de coordinación y mediación entre las necesidades a nivel del negocio y las implementaciones de TI.

Por último, la capa de presentación proporciona a los usuarios la interfaz a las implementaciones de los procesos del negocio y los servicios definidos, a través de tecnologías como portales, aplicaciones cliente o cualquier otro mecanismo.

Adicionalmente, SOA define una serie de patrones para los elementos de la arquitectura, entre ellos: la interacción de los componentes, y el diseño y exposición de los servicios. El modelo de interacción de SOA especifica, en principio, la comunicación entre tres

componentes básicos: el proveedor del servicio, el consumidor del servicio y el *broker* de servicios (Arsanjani, 2004).

El *broker* de servicios ha llegado a evolucionar en su funcionalidad, y se ha transformado en uno de los componentes más importantes de una arquitectura orientada a servicios: el Bus de Servicios Empresarial (ESB, por sus siglas en inglés). Este elemento, en general, provee funcionalidades de conectividad, mediación y registro de servicios; sin embargo, las funcionalidades incluidas en el ESB dependerán de cada implementación.

Por medio de sus capacidades de mediación, el ESB disminuye la complejidad de las conexiones. Además, facilita la interoperabilidad entre los distintos elementos de la arquitectura, aunque estos pueden ser totalmente heterogéneos en su tecnología de implementación.

Adicionalmente, el ESB reduce las dependencias entre los componentes, lo cual permite desacoplar las implementaciones de los servicios de su utilización y, de esta manera, facilitar el reemplazo futuro de componentes y la adaptación de nuevos servicios.

Los principios fundamentales de SOA

La construcción de una arquitectura orientada a servicios requiere tiempo y esfuerzo. Una adecuada comprensión de los

principios que rigen este concepto es de suma importancia para dar un buen enfoque durante su adopción y, de esta manera, minimizar la posibilidad de un eventual fracaso.

Adicionalmente, deben considerarse algunos conceptos como computación distribuida, mensajería de integración, metadatos e interfaces, BPM y gobernabilidad, pues se encuentran estrechamente relacionados con la disciplina de construcción de una arquitectura SOA.

Como se comentó anteriormente, el concepto de servicios es aplicado en el contexto empresarial desde hace ya varios años, y esto ha permitido una mayor flexibilidad en los procesos de negocio; pero aplicar este concepto a arquitecturas de software conlleva una dificultad un poco mayor debido a los aspectos técnicos que son fundamento de la teoría de SOA.

De la misma forma, la implementación de diseños de software orientados a servicios, en comparación con los diseños existentes, requiere un poco más de rigor y esfuerzo en su construcción, dada su complejidad y las dificultades enfrentadas durante las fases iniciales.

En principio, el diseño del software que formará parte de la arquitectura debe cumplir con la siguiente serie de principios fundamentales de diseño:

- Modularidad: La arquitectura se encuentra formada por una serie de servicios, cada uno construido para ejecutar una función específica o una serie de funciones relacionadas. Una aplicación empresarial orientada a servicios (SOBA, por sus siglas en inglés) se encuentra compuesta por una serie de módulos autónomos (servicios) que ejecutan tareas específicas. Cada uno de los módulos del sistema SOA, durante la ejecución de algún proceso, ejecuta funciones de consumidor o proveedor de un servicio específico.
- Distribución: En muchas ocasiones, el consumidor del servicio y el proveedor de este se ejecutan en máquinas distintas, por lo cual, los módulos que forman parte del sistema deben tener la capacidad de ser ejecutados en distintos computadores. Esto implica la necesidad de utilizar un mecanismo de comunicación estándar que permita la comunicación entre ellos.
- Abstracción: Similar al concepto de abstracción en la teoría de orientación a objetos, la lógica de realización de los servicios no debe ser visible a sus consumidores. Adicionalmente, es conveniente que los datos persistentes relacionados con un servicio sean encapsulados y accedidos solamente a través de este.
- Independencia (bajo acoplamiento): Deben reducirse las dependencias entre servicios. El desacoplamiento entre servicios se obtiene por medio de la definición de interfaces para cada

servicio. La implementación de la lógica de un servicio no debe depender de la de otro. La interfaz de un servicio debe definir los metadatos necesarios para la invocación del servicio y algunos datos descriptivos de este.

- Interfaz independiente: La interfaz del servicio debe estar separada de su implementación. Esta interfaz debe ser accesible para el programador del servicio consumidor, independientemente del código de implementación del servicio proveedor. En general, es conveniente que estas interfaces residan en un repositorio o registro formal de servicios.
- Composición: Los servicios deben ser diseñados de forma que puedan combinarse entre ellos para producir servicios compuestos. El principio de composición es de suma importancia, ya que permite construir nuevos servicios con base en los servicios existentes, y de esta forma responder más ágilmente a necesidades futuras.
- Reutilización: El diseño de los servicios debe ser construido con base en el principio de reutilización. Un servicio debe ser implementado tomando en cuenta que puede formar parte de varios flujos o procesos de negocio y, por lo tanto, debe poder ser llamado por múltiples consumidores. En este sentido, es importante estudiar detalladamente la 'granularidad' conveniente de un servicio antes de su construcción.

Adicionalmente, existe una serie de consideraciones importantes para la identificación y construcción de los servicios que formarán parte de la arquitectura. Estas consideraciones serán descritas más adelante.

También es importante considerar que la construcción de una arquitectura orientada a servicios debe promover la colaboración entre los diferentes servicios para resolver las distintas tareas del negocio. La interoperabilidad entre los distintos elementos de la arquitectura es uno de los fundamentos más importantes de SOA. Por ello, la implementación de un ESB como mecanismo de conectividad y mediación entre las partes es uno de los aspectos fundamentales en la adopción de esta arquitectura.

Por último, no debemos perder de vista que el objetivo de la implementación de los servicios en este contexto es soportar alguna tarea de proceso del negocio. En este sentido, es esencial que los servicios construidos estén enfocados hacia un concepto de negocio.

Beneficios e implicaciones

El concepto de SOA está adquiriendo un éxito cada vez mayor. La cantidad de empresas interesadas en implementaciones de este tipo está aumentando rápidamente. Algunas empresas ya se han colocado a la vanguardia con implementaciones exitosas de SOA, y

muchas otras ya han planteado en su cartera de proyectos la implementación de una arquitectura orientada a servicios, o por lo menos, la ejecución de algún plan piloto o prueba de concepto relacionada.

Por otra parte, desde hace ya varios años, muchos de los grandes proveedores de soluciones de TI han enfocado sus esfuerzos a la creación y venta de software para la implementación de SOA (software que soporta la arquitectura y explota sus capacidades). De esta forma, las grandes empresas proveedoras de TI han participado activamente en un rol promotor de esta nueva tecnología.

Sin duda alguna, el apoyo que SOA ha recibido en los últimos años se debe, en gran parte, a que este nuevo concepto se enfoca en varios problemas clave que las organizaciones están enfrentando. Hace algunos años, el interés principal de las empresas era automatizar sus procesos de negocio mediante el uso de tecnologías de información y comunicación. Hoy, los factores clave para la competitividad empresarial incluyen el estado de sinergia entre los procesos de negocio, la tecnología que los soporta y la agilidad para responder a los cambios en el mercado. Precisamente estos son dos de los principales beneficios de una buena implementación de SOA.

Adicionalmente, se debe tener claro que la tecnología es una herramienta para respaldar y mejorar las operaciones del negocio, por lo cual debe estar siempre alineada con los objetivos estratégicos

de la organización. En sentido estricto, los intereses de las empresas no se centran en la adquisición de tecnología para soportar sus procesos, sino en los procesos mismos y sus servicios, conceptualizados desde el punto de vista empresarial y no de TI. En este sentido, el diseño de arquitecturas orientadas a servicios contribuye, en gran medida, a dar una mejor orientación a las tecnologías de información, de manera que exista una mejor integración y simbiosis entre los sistemas de información y los procesos de negocio. Esto se debe, principalmente, a la correlación que existe entre los servicios definidos en la arquitectura SOA y las disciplinas de la capa superior, como BPM. De esta manera, SOA promueve una importante interacción entre los conocedores del negocio y los implementadores de las tecnologías de información, relación fundamental para mantener la alineación con el negocio.

Otra de las principales atracciones de SOA es su capacidad para reducir la curva de costos asociados a los cambios futuros en los sistemas de información. Este es un punto importante por considerar, ya que justifica las grandes inversiones de tiempo y dinero necesarias para la implementación de SOA en nuestra empresa.

En las arquitecturas tradicionales, el crecimiento de los sistemas de información provoca un aumento exponencial en la complejidad y, por lo tanto, en el esfuerzo y dinero necesario para adecuar los sistemas a los cambiantes requerimientos del negocio. El

concepto de SOA pretende reducir este aumento de costos por medio de una arquitectura compuesta por servicios interoperables e independientes; la publicación de "contratos" o metadatos; y, adicionalmente, la integración por medio de mecanismos de interoperabilidad "inteligentes" entre las partes. Así, aunque la complejidad del sistema en su totalidad crezca, los costos asociados a los cambios futuros serán siempre los mismos. Esta es una de las ventajas de la construcción de una arquitectura basada en piezas de software independientes. Estas mismas premisas conducen a la capacidad de realizar cambios mayores y más frecuentes en los sistemas de información y, consecuentemente, hacia una mejor alineación de los sistemas de información con los requerimientos del negocio.

De este modo, los conceptos fundamentales de SOA (modularidad, distribución, abstracción, independencia, interfaces, composición y reutilización), propician una serie de ahorros en costos de mantenimiento, administración y operación de los sistemas de información. La construcción de una arquitectura basada en estos conceptos facilita la modificación de los sistemas, disminuye el tiempo de construcción y ensamble de nuevos módulos, facilita la interoperabilidad y distribución de los sistemas e incrementa la reutilización.

Por otra parte, la capacidad de una arquitectura orientada a servicios para adecuarse rápidamente a nuevos requerimientos tiene repercusiones directas sobre la competitividad de la organización, debido a que le permite a la empresa preparar la entrega de nuevos productos o servicios rápidamente, agregar nuevos canales de distribución con facilidad, optimizar la interoperabilidad con proveedores y mejorar la visibilidad de los procesos del negocio, entre otros beneficios.

Sin embargo, debemos también considerar que la implementación de SOA no solamente trae beneficios. El proceso de adopción de esta arquitectura tiene varios costos asociados. Por una parte, a pesar de que los cambios evolutivos son de menor costo, la construcción inicial de una arquitectura distribuida como SOA es mucho más difícil de diseñar, implementar y probar que las arquitecturas de aplicaciones tradicionales, por tratarse de una arquitectura compuesta por una gran cantidad de componentes heterogéneos que deben diseñarse y construirse para trabajar de forma integral.

Por otro lado, deben realizarse cambios sustanciales a nivel operativo para que la implementación de esta tecnología sea exitosa, no solo en el área de TI, sino también en algunas otras áreas de la empresa. Esta alteración inherente en el modo de trabajo habitual

puede inducir a la falta de aceptación y, consecuentemente, provocar un eventual fracaso.

Igualmente, debe adoptarse una metodología y un enfoque diferente para el desarrollo de las soluciones de software. Esta nueva metodología involucra al área de TI y a otros sectores, los cuales interactúan en el modelado de los procesos de negocio, que serán posteriormente convertidos en secuencias o flujos de servicios.

Los esfuerzos y cambios requeridos son grandes, al igual que los beneficios obtenidos; y como cualquier proyecto de implementación de tecnología, la adopción de una arquitectura orientada a servicios resultará satisfactoria únicamente cuando los objetivos, estrategias y pasos por seguir se encuentren claramente definidos desde el principio. En las siguientes secciones se detallará una serie de aspectos importantes por considerar para la definición de la estrategia de adopción de SOA.

El punto de partida

Los esfuerzos que actualmente las empresas dedican a la implementación de SOA son grandes; no obstante, dada la complejidad de su adopción y los retos que deben enfrentarse, es muy probable que muchos de los esfuerzos fracasen o que las implementaciones no produzcan los beneficios esperados.

Por esto, es realmente importante definir previamente el camino por seguir, cada detalle, paso a paso, antes de empezar a caminar por él. Igualmente, como cualquier proyecto, la definición de los objetivos es un aspecto crítico para no caer en el error de "caminar sin rumbo". En este sentido, debe tenerse claro que un proyecto de adopción de SOA es un proyecto de grandes dimensiones y, por consiguiente, una buena estrategia consiste en dividir el proyecto en segmentos. De esta manera es posible atacar problemas pequeños y menos complejos hasta resolver el problema en su totalidad. Así, los patrocinadores e involucrados en el proyecto pueden ver resultados en un plazo no tan extenso, lo que evitaría una decepción o falta de interés que puede llevar al fracaso.

Siguiendo con la estrategia, para la primera implementación de SOA es recomendable escoger inteligentemente algún proceso de negocio relativamente pequeño y bien definido dentro de la empresa. Es importante que la ejecución de este primer proyecto involucre todas las etapas de ejecución de SOA: modelado, integración, publicación y administración (Keen, Adamski, Basu, Chilcott, Eames, Endrei, Fagalde, Raszca y Seabury, 2007). Para esto, la posibilidad de adquirir software para soportar cada una de estas etapas debe ser evaluada, por lo cual deben estudiarse las alternativas existentes según las necesidades de la empresa, considerando, entre otras cosas, la interoperabilidad entre los paquetes de software y la utilización de estándares como BPEL (formato estándar para el

modelado de procesos de negocio) y WSDL (formato estándar para la descripción de servicios publicados).

Con esto, surge la necesidad de capacitar al personal en la utilización de las nuevas herramientas de software, para lo cual es conveniente recurrir a asesorías de proveedores de TI reconocidos. Sin embargo, es sumamente importante que antes de abordar cualquier capacitación sobre la utilización del software, todos los involucrados tengan claros los conceptos básicos de SOA y, sobre todo, los objetivos del proyecto y su responsabilidad dentro del ciclo de vida de la implementación.

En este sentido, uno de los aspectos más importantes para iniciar un proyecto de SOA es que este debe ser visualizado como un proyecto a nivel empresarial y no del área de tecnología de información. Es muy probable que si el proyecto de implementación de SOA es conceptualizado como un proyecto típico de TI, este fracase por la falta de interés del nivel empresarial y, sobre todo, porque por su naturaleza, la ejecución de un proyecto de SOA requiere la colaboración de distintas partes de la organización.

De la misma forma, el proyecto debe ser visualizado en conjunto con la disciplina de administración de procesos del negocio (BPM), por ser este un elemento imprescindible en el ciclo de vida de SOA. Por estas razones, debe involucrarse activamente a todo el

personal necesario de las áreas de negocio y, sobre todo, estimular su confianza en el proyecto y su compromiso con este.

Por otra parte, debe considerarse que el costo involucrado en las primeras etapas de SOA será mucho mayor que los beneficios obtenidos. En las fases iniciales de implementación, SOA no producirá una reducción significativa de costos, ni en el ámbito de TI, ni en ningún otro ámbito de empresa. El valor asociado a SOA se adquiere en el factor "agilidad" para los cambios futuros, y en la medida en que esta nueva arquitectura vaya adquiriendo madurez y una adecuada orientación. Es importante que los patrocinadores del proyecto tengan esto presente, para evitar crear falsas esperanzas, por lo cual, una buena justificación de las inversiones es primordial.

Las dificultades que las empresas enfrentan en el camino de adopción de SOA incluyen retos en varios ámbitos de esta doctrina, principalmente en las áreas de gobernabilidad, configuración, administración de los metadatos, seguridad, monitoreo e interoperabilidad. Existe un gran riesgo asociado a la posibilidad de que alguno de estos problemas deteriore la estrategia y disminuya la posibilidad de éxito en la implementación. Estos riesgos deben ser claramente identificados desde el principio, para ser atacados con un plan de acción debidamente orientado.

Adicionalmente, existen varias estrategias que deben ser consideradas para abordar el problema de construcción de servicios.

La estrategia puede ser de tipo *"top-down"*, *"bottom-up"* o una combinación de ambas. Por una parte, la estrategia *"top-down"* implica la construcción de los servicios a partir de los insumos de la capa superior de procesos, es decir, los servicios son construidos con base en procesos previamente modelados, lo que permite agregar los componentes de software existentes de acuerdo con las necesidades que se vayan presentando. La utilización de una estrategia de este tipo requiere un modelo de procesos maduro (procesos claramente definidos) en la empresa. Es conveniente seguir la estrategia *"top-down"* para los casos de construcción de servicios totalmente nuevos.

Por otra parte, la estrategia *"bottom-up"* implica tomar los componentes de software existentes que pueden ser reutilizados, y construir servicios a partir de ellos, integrándolos de manera progresiva a la arquitectura SOA. Estos servicios serán posteriormente combinados y utilizados según demanden los procesos de negocio. Esta estrategia es bastante popular, ya que permite iniciar el proceso de construcción de servicios reutilizando los componentes de software que la empresa utiliza actualmente. Sin embargo, debe tenerse claro que seguir una estrategia de tipo *"bottom-up"* puede provocar la necesidad posterior de modificar los servicios construidos para adecuarse a los requerimientos del negocio, lo que implicaría un esfuerzo adicional.

De cualquier forma, es fundamental realizar un estudio minucioso de la madurez del modelo de procesos, y de manera similar, revisar el modelo de componentes de software actual de la empresa. Este examen ayudará a definir la mejor estrategia por seguir para la construcción de los servicios.

Definiendo servicios

La definición de servicios de una arquitectura orientada a servicios es un proceso complejo. Inicialmente, es importante tener una visión clara del concepto de servicios y demás componentes que interactúan en una arquitectura orientada a servicios. Una adecuada comprensión de los conceptos y la utilización de la técnica correcta para la identificación y construcción de los servicios es esencial para la implementación de una arquitectura apropiada.

Conceptualmente, las unidades de trabajo en una arquitectura de servicios pueden visualizarse en 3 niveles de abstracción: las operaciones, los servicios y los procesos de negocio. Por una parte, las operaciones consisten en transacciones que representan una unidad lógica de trabajo. Estas generalmente involucran el acceso o modificación de datos persistentes y son comparables con los métodos de las aplicaciones orientadas a objetos. Por otra parte, los servicios consisten en una agrupación lógica de operaciones y, por último, los procesos del negocio representan una serie de actividades

desde el punto de vista empresarial, basadas en un conjunto de reglas de negocio, y que generalmente involucran la utilización de varios servicios. La secuencia, llamado y ejecución de estos servicios es llamada 'coreografía de servicios o procesos' (Zimmermann, Olaf y Gee, 2004).

Desde el punto de vista de diseño de software, la "granularidad" o nivel de abstracción en una arquitectura SOA puede visualizarse por medio de 3 capas: la capa de clases, la capa de componentes y la capa de servicios (Zimmermann, Olaf y Gee, 2004).

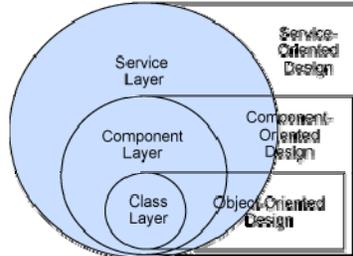


Ilustración 3: Capas de diseño

Fuente: Zimmermann et al., 2004.

Ahora bien, como se comentó anteriormente, se puede abordar la creación de los servicios desde diferentes perspectivas. Por un lado, tener procesos claramente definidos en la empresa o la referencia de algún mapa de procesos estándar de la industria (como

eTOM para telecomunicaciones, por ejemplo) resulta de gran utilidad para un enfoque de tipo "*top-down*". En este caso, se debe velar por un correcto modelado de los procesos del negocio, ya que estos modelos serán los insumos para identificar y construir los servicios.

Por otra parte, para una estrategia de tipo "*bottom-up*" no existe alguna herramienta o procedimiento que permita mapear elementos existentes (clases, componentes) a servicios de manera automática y sin un análisis previo. Sin embargo, existe una serie de patrones, técnicas y consejos por seguir para lograr identificar y exponer los servicios de la manera más adecuada.

Desde cualquiera de los enfoques debe alcanzarse un nivel de granularidad de los servicios balanceado y acorde con las necesidades del negocio. Esto sólo se consigue mediante un análisis profundo y utilizando las técnicas apropiadas.

Definir este nivel de granularidad es un procedimiento de análisis complejo, que debe tomar en cuenta los siguientes consejos:

1. Preferiblemente, diseñar los servicios generales en lugar de específicos: es conveniente que los servicios sean construidos con funciones no demasiado específicas, sin perder de vista la completitud y consistencia. La creación de servicios con una granularidad gruesa reduce la complejidad, aumenta la eficiencia, reduce la redundancia y el acoplamiento. La mejor manera de manejar servicios dentro de una arquitectura SOA consiste en

tener pocos servicios que puedan ser reutilizados en varios escenarios. Sin embargo, pueden definirse servicios con una granularidad fina, si una necesidad del negocio lo amerita. Debe conseguirse el balance apropiado.

2. Tomar en cuenta las necesidades del negocio, que al final son las que determinan el nivel de granularidad de un servicio más eficiente para la organización. Los servicios deben ser diseñados desde el punto de vista del negocio, ya que atenderán una necesidad específica de un proceso empresarial.

Adicionalmente, para la incorporación de software existente en la arquitectura SOA, puede utilizarse alguna técnica como la descomposición, la cual consiste en tomar las aplicaciones y paquetes de software existentes y descomponerlos en una serie de servicios que representan un conjunto de operaciones relacionadas. Por otra parte, el modelo de diseño de aplicaciones orientado a componentes es una técnica que permite la definición de servicios de una manera sencilla y eficiente. Con base en esta técnica, es posible definir componentes, y a partir de esto, diseñar los servicios como una agrupación de componentes que interactúan entre sí para realizar una función del negocio. La facilidad para la identificación e implementación de los servicios utilizando este tipo de técnicas, dependerá directamente del diseño actual de las aplicaciones.

De cualquier forma, la definición de servicios en una arquitectura orientada a servicios puede hacerse más eficiente si se toman en cuenta los siguientes puntos:

1. Orientado al reuso: uno de los principales objetivos de una arquitectura orientada a servicios es definir servicios que puedan ser reutilizados. Si un componente no es apto para la reutilización, no debe ser definido como servicio.
2. Deben tener operaciones identificables, que permitan que todos los elementos que lo requieran utilicen el servicio.
3. Debe tener interfaces bien definidas que presenten toda la información necesaria sobre el servicio. Esta información incluye: descripción del servicio, información de los mensajes, cómo construir y a dónde enviar los mensajes.
4. Debe transformar los datos de un estado válido a otro: deben ser totalmente independientes, esto quiere decir que el servicio no debe depender del estado o información de otro servicio.
5. Disminuir el acoplamiento: los servicios deben definirse como cajas negras para el solicitante, de manera que la implementación del servicio no sea importante para este.

El proceso de migración a SOA, y en especial la identificación de servicios, es un proceso que requiere un análisis y que sin duda representa un punto clave para la construcción de una arquitectura conveniente. Es esencial tener una visión de todo el proceso del

negocio y tomar en cuenta las técnicas y consejos para la identificación y definición de los servicios.

Consideraciones para la implementación

Existe una gran cantidad de factores que deben considerarse para lograr una buena implementación de SOA. Cabe aclarar que cada proyecto de adopción de SOA tiene sus elementos particulares, y estos deben ser estudiados junto con cada una de las opciones para su resolución. El estudio de cada alternativa debe realizarse con visión de largo plazo y sin perder de vista los objetivos de la arquitectura y del proyecto en general. Pese a esto, las recomendaciones presentadas a continuación constituyen una serie de buenas prácticas que deberían considerarse en todos los proyectos de adopción de una arquitectura orientada a servicios.

1. Utilizar estándares en toda la implementación.

Utilizar estándares de la industria en cualquier ámbito reduce la dependencia y promueve la interoperabilidad. Por esto, la utilización de protocolos y formatos estándares es uno de los aspectos más importantes en la implementación de SOA. Es conveniente utilizar estándares para los modelos de procesos (BPEL, BPMN), los metadatos (XML, WSDL), el acceso a los servicios (SOAP/HTTP, JMS) y así en todos los ámbitos de la arquitectura en los que sea posible, siempre que no exista alguna

razón de rendimiento o calidad de servicio que justifique la utilización de un método propietario.

De igual manera, en caso de adquirirse un producto para soportar algún proceso del ciclo de vida de SOA, debe evaluarse con rigidez la compatibilidad e integración con los productos ya implementados, y la utilización de estándares de la industria como una característica esencial.

2. Nunca perder de vista los principios fundamentales.

La construcción de una buena arquitectura orientada a servicios debe respetar, en todo caso, sus principios fundamentales. De esta forma, el diseño de cualquier servicio debe obedecer los principios de modularidad, distribución, abstracción, independencia, interfaz independiente, composición y reutilización.

3. No desechar las aplicaciones actuales.

Se debe evitar adoptar una estrategia de sustitución total de las aplicaciones existentes por aplicaciones orientadas a servicios. Es recomendable integrar los paquetes de software adquiridos, las aplicaciones legadas y los servicios existentes a la arquitectura de manera progresiva. Para la integración de las aplicaciones y paquetes de software, conviene utilizar una capa de abstracción adicional para la interconexión con bus de servicios empresarial (conectores/adaptadores).

4. Utilizar conectores para toda la lógica de transporte.

Durante la integración de las aplicaciones legadas a la capa de servicios, es recomendable abstraer la lógica de transporte (para la interconexión con el bus de servicios empresarial) de las aplicaciones legadas o el software existente, por medio de una pieza de software independiente, encargada de esta función. Es importante que estos conectores implementen exclusivamente la lógica de transporte, sin incluir lógica del servicio o del proceso. Los conectores para algunos paquetes de software pueden ser comprados a algunos proveedores de TI. Es conveniente realizar un análisis de esta posibilidad para esos casos.

5. Utilizar adaptadores en donde sea necesario.

Durante la integración de las aplicaciones legadas a la capa de servicios, y en los casos en que sea necesaria la transformación de formato de los mensajes de invocación o respuesta de servicios, debe utilizarse una pieza de software independiente, similar a los conectores, que incluya la lógica de transformación pertinente (por ejemplo de XML a CWF y viceversa). Igualmente, es primordial que estos adaptadores contengan solamente la lógica de conexión-adaptación, sin incluir lógica del servicio, ni mucho menos, lógica del proceso.

6. Conservar la abstracción entre las capas.

Un diseño correcto de una arquitectura SOA debe establecer claramente los límites entre cada una de las capas que la conforman y evitar el acoplamiento entre ellas. Así, por ejemplo, el diseño de un servicio (de la capa de servicios) debe evitar la incorporación de la lógica de un proceso de negocio (de la capa superior de procesos). De igual forma, el diseño de las interfaces entre cada una de las capas debe evitar al máximo la dependencia entre los elementos que se comunican a través de ella. La utilización de conectores y estándares favorece esta abstracción.

7. Utilizar un bus de servicio empresarial (ESB) y un registro formal de servicios para implementaciones relativamente grandes o en crecimiento constante.

Para arquitecturas SOA de escala mediana-grande o en situaciones en las que se visualiza un crecimiento constante en la cantidad de servicios, es recomendable utilizar un ESB como mecanismo centralizado de conectividad y mediación, favoreciendo la interoperabilidad, disminuyendo las dependencias y facilitando la adaptación o reemplazo de nuevos componentes.

En ese mismo contexto, es conveniente administrar un repositorio o registro de servicios centralizado y formal. Para esto, cabe la posibilidad de adquirir algún producto que provea esta funcionalidad, por lo que es conveniente analizar esta alternativa,

siempre tomando en cuenta la utilización de estándares (WSDL, XML y XSD). Adicionalmente, es recomendable que los procedimientos de descubrimiento y publicación o registro de servicios sea claramente definido desde el inicio de la implementación.

8. Implementar un mecanismo para el monitoreo de la calidad del servicio.

En muchas ocasiones, la disponibilidad y el rendimiento de los servicios es un factor crítico con respecto a la ejecución de los procesos de negocio. En tales casos, la implementación de un mecanismo para el monitoreo de los servicios resulta de gran utilidad.

9. Implementar un mecanismo de seguridad en las capas donde sea necesario.

Dado que SOA difiere de una arquitectura común, los nuevos mecanismos de seguridad por implementar serán un tanto diferentes a los ya conocidos. No obstante, la seguridad es un factor crítico en la implementación de SOA, por lo que debe considerarse, sin excepción, la incorporación de algún mecanismo de autorización en las capas pertinentes y de acuerdo con la granularidad de seguridad que sea conveniente.

10. Aplicar una estrategia para mantener una buena gobernabilidad.

La gobernabilidad es un factor clave para el éxito de la adopción de una arquitectura orientada a servicios. Al inicio del proyecto de implementación de SOA, debe definirse de una manera clara y formal cada uno de los roles y responsabilidades, así como los procedimientos por seguir para cada etapa del ciclo de vida de SOA. Mantener una buena comunicación los involucrados es un aspecto igualmente indispensable.

11. Separar las reglas de negocio de la implementación de los procesos en casos de gran volumen de procesos o reglas sumamente cambiantes.

La separación de las reglas de negocio de la lógica de los procesos facilita y agiliza los cambios en dichas reglas, con lo que se evita la necesidad de recurrir nuevamente al ciclo de implementación, prueba y publicación del proceso. Esta disociación puede llevarse a cabo por medio de la utilización de un repositorio de reglas de negocio.

12. Aplicar los conceptos de SOA de forma selectiva.

No es recomendable utilizar un diseño orientado a servicios para aplicaciones pequeñas o temporales. La utilización de SOA es

recomendable cuando se diseñen aplicaciones nuevas de tamaño relativamente grande, en diseños de soluciones para un largo período, en integración de aplicaciones legadas, paquetes de software adquiridos o servicios existentes, o para aplicaciones propensas al cambio o reutilización. De cualquier manera, es recomendable realizar un análisis de cada caso particular.

13. Seguir el ciclo de mejora continua.

Es conveniente incorporar al ciclo de vida de SOA, un mecanismo para el monitoreo y seguimiento de los procesos de negocio. Para ello, debe analizarse la posibilidad de adquisición de software que proporcione esta funcionalidad o, en su defecto, la implementación de este mecanismo por otro medio. De cualquier forma, el monitoreo de los procesos de negocio utilizando métricas definidas da un gran valor agregado a la administración de los procesos del negocio, como complemento de SOA, habilitando un ciclo de mejora continua y, por consiguiente, mejorando la eficiencia y competitividad de la organización.

Conclusiones

La adopción de una arquitectura orientada a servicios produce gran cantidad de beneficios; sin embargo, los retos y complicaciones

que pueden presentarse durante su implementación son grandes y pueden afectar la estrategia de forma contraproducente. Por ello, es indispensable que el plan de adopción de SOA considere los riesgos inherentes al proceso de implementación, y de esta manera, se puedan tomar las acciones necesarias para disminuir la posibilidad de un fracaso.

Por otra parte, existe una serie de aspectos que han sido identificados como factores claves para una buena implementación de SOA. Estos elementos constituyen un conjunto de mejores prácticas para la adopción de la arquitectura y deben ser considerados durante el proceso de planificación e implementación. No obstante, cada proyecto tiene sus elementos particulares, por lo que estas mejores prácticas no son una garantía del éxito de la implementación, sin embargo, constituyen una guía por seguir, para atacar de manera efectiva los retos más comunes, presentes en el camino de adopción de una arquitectura orientada a servicios.

Referencias bibliográficas

- Arsanjani, A. (2004). *Service-oriented modeling and architecture*.
Recuperado el 26 de enero de 2008, de
<http://www.ibm.com/developerworks/library/ws-soa-design1/>.
- High, R.J., Kinder, S. y Graham, S. (2005). *IBM SOA Foundation – Architecture Overview* [Versión electrónica]. IBM Business Consulting Services.
- Keen, M., Adamski, D., Basu, I., Chilcott, P., Eames, M., Endrei, M., Fagalde, B., Raszca, R. y Seabury, S.D. (2007). *Implementing Technology to Support SOA Governance and Management* [Versión electrónica]. IBM Redbooks.
- Legner C. y Heutschi R. (2007). *SOA Adoption in Practice - Findings from Early SOA Implementations*. Recuperado el 2 de febrero de 2008, de
http://www.alexandria.unisg.ch/publications/person/O/Hubert_Oesterle/35214.

- Schulte, R. y Abrams, C. (2006). *SOA Overview and Guide to SOA Research* [Versión electrónica]. Gartner Inc.
- Shulte, R. y Natis, Y. (1996). *"Service Oriented" Architectures, Part 1.* [Versión electrónica]. Gartner Inc.
- Zimmermann, O., Krogdahl, P. y Gee, C. (2004). *Elements of Service-Oriented Analysis and Design.* Recuperado el 2 de febrero de 2008, de <http://www-128.ibm.com/developerworks/webservices/library/ws-soad1/>.