

INTEGRACION DE PROCESOS UTILIZANDO LA ARQUITECTURA ORIENTADA A SERVICIOS - SOA

Process Integration Making Use of Service Oriented Architecture SOA

RESUMEN

La Arquitectura de desarrollo de software Orientada a Servicios viene de una evolución de orientación a objetos, luego a componentes para convertirse en un nuevo paradigma que propone convertir los recursos de software en servicios disponibles que puedan ser reutilizables y fáciles de integrar en grandes y complejas aplicaciones, facilitando la interoperabilidad entre las aplicaciones y el web service. Así se integran diferentes ciclos de vida e implementaciones de soluciones con múltiples tecnologías.

PALABRAS CLAVES: Integración de procesos, integración de tecnología, granularidad, modularidad, orientación a servicios, procesos de negocios, servicios reutilizable, servicios Web, SOA,

ABSTRACT

Service Oriented Architecture, SOA, comes from a evolution of object oriented and component Oriented to become a new paradigm that proposes to turn the software resources into services available that can be re-usable and easy to integrate in great and complex applications, being allowed the interoperability between applications and web service, integrating different life cycles and obtaining implementations with multiple technologies.

KEYWORDS: Process integration, technology integration, granularity, modularity, business processes, Service-orientation, Web services, SOA.

1. INTRODUCCIÓN

Arquitectura es un concepto indispensable entre las aplicaciones hoy en día, se refiere a la combinación e integración de múltiples ambientes o plataformas. [8]

La integración y coordinación de procesos puede verse simplificada utilizando Arquitecturas de integración como SOA y Web Services. [1]

Desde la aparición de los computadores y el software se ha trabajado en diversos métodos para facilitar y mejorar la producción de código ejecutable y la creación de aplicaciones. Como resultado se han ido consiguiendo diferentes técnicas y herramientas como ensambladores, compiladores, herramientas gráficas de desarrollo rápido de aplicaciones, herramientas de generación de código a partir de modelos y métodos basados en la aplicación de patrones, entre otros.

Actualmente las técnicas y métodos avanzados de desarrollo de software se basan en el uso de modelos abstractos a partir de los cuales es más sencillo, rápido y económico generar código, como el Desarrollo Dirigido

SANDRA MILENA CAICEDO

Ing. de Sistemas.
Maestrante Ciencias
Computacionales
Profesor Auxiliar
Universidad Tecnológica de Pereira
caimilsa@gmail.com

LIGIA STELLA BUSTOS R, M.Sc.

Ph.D. (c) Ingeniería Informática.
Ingeniería del Software.
Magíster en Administración
Económ. y Financiera.
Profesor Asistente
Universidad Tecnológica de Pereira
ligias@utp.edu.co

JERÓNIMO ROJAS DIAZ

Ing. Electrónico
Maestrante Ciencias
Computacionales
Profesor Auxiliar
Universidad Tecnológica de Pereira
Profesor Universidad de Caldas
jeroxx@gmail.com

por Modelos, más conocida como el estándar Model Driven Architecture (MDA). Es una especificación que representa una alternativa más para desarrollar software, en este caso a partir de modelos muy abstractos.

En otro ámbito y desde un punto de vista tecnológico orientado a flexibilizar la construcción e integración de software, en los últimos años ha crecido el número de desarrollos basados en el uso de servicios de software y servicios Web. Esta forma de desarrollo cambia el enfoque tradicional ya que permite construir aplicaciones a base de pequeños componentes configurados como servicios.

Dentro del área de desarrollo basada en servicios debemos destacar la arquitectura orientada a servicios, denominada Service Oriented Architecture (SOA), ya que permite realizar una localización adecuada de estos componentes independiente de su ubicación, integrar funcionalidades realizadas con diferentes tecnologías de implementación y realizar una ejecución que no depende de una u otra plataforma de hardware, sistema operativo o middleware.

De forma general podemos considerar que un servicio, ya sea del negocio, software o Web, está basado en una estructura sencilla donde existen dos partes. Una realiza solicitudes de servicio y otra representa el proveedor del servicio, el cual recibe las peticiones, las procesa y envía las respuestas adecuadas. Esto se lleva a cabo de forma ordenada mediante el paso de mensajes entre el solicitante y el proveedor.

El concepto de servicio del negocio puede entenderse como un mecanismo básico que facilita la organización y la ejecución de los intercambios habituales del negocio.

Por ejemplo, en una empresa dedicada a la venta de productos podemos considerar que el pago de un artículo es un servicio del negocio, es decir, un mecanismo básico mediante el cual el cliente que desea comprar hace una solicitud al encargado de ventas. De acuerdo a unas reglas de negocio asociadas a este servicio, entre ambos intercambian información con el objetivo de realizar correctamente esta operación.

El concepto de servicio software, es similar al de componente de software ya que ambos ofrecen determinadas funcionalidades a través de una interfaz externa que permite acceder a sus funciones sin tener en cuenta sus detalles internos.

La arquitectura orientada a servicios o SOA permite interconectar y acceder a los servicios a través de una interfaz común independientemente de la tecnología con la que se ha desarrollado el servicio y de su localización (ver figura 1).

3. DESARROLLO DE PROYECTOS CON ARQUITECTURA ORIENTADA A SERVICIO SOA

Los modelos de ciclo de vida (MCVS) para SOA utiliza pilares de ciclos de vida para soluciones distribuidos a continuación enunciaremos cada una de esas etapas. [9][1]

3.1 MCVS SOA con enfoque Top – Down

La estrategia top-down utilizada para construir soluciones SOA genera productos de alta calidad. La arquitectura resultante con este enfoque será óptima debido a que se comienza analizando el flujo de negocio de manera integral, para luego bajar el nivel de detalle hasta los servicios a implementar. La principal desventaja de este enfoque es el presupuesto y tiempos. [9][1]

Realizar un análisis global del negocio y del conjunto completo de servicios a implementar implica un gran costo inicial que no todas las organizaciones se encuentran dispuestas a enfrentar.

Las fases que integran este MCVS aplicado a proyectos SOA, son las siguientes:

a. Análisis

En esta primera fase se determina el alcance del proyecto SOA. Antes de modelar el esquema de servicios se comienza por analizar en detalle el flujo y reglas de negocio de la organización. Luego surgen los principales servicios candidatos, y se definen las capas a utilizar.

b. Diseño

Una vez definido el análisis, se puede comenzar a diseñar de qué forma implementarlo. Esta fase es por lo general dirigida en base a estándares e incorpora principios y convenciones establecidas para sistemas orientados a servicios.

c. Desarrollo

Una vez determinadas las tecnologías sobre las cuales se construirán los componentes de la arquitectura orientada a servicios, sólo basta construirlos.

d. Pruebas

Debido a que los servicios serán potencialmente módulos reutilizables en una gran variedad de escenarios, su calidad debe ser rigurosamente controlada.

e. Implantación

Con base a las tecnologías específicas seleccionadas para el desarrollo se podrán definir las actividades que formen parte de la fase de implantación.

f. Administración

La naturaleza de los factores a administrar para este tipo de sistemas va a ser muy similar a la utilizada para sistemas distribuidos basados en componentes.

La administración incluye el monitoreo de servicios, control de versiones, seguimiento de mensajes, detección de cuellos de botella.

3.2. MCVS SOA con enfoque Bottom-Up

El enfoque bottom-up establece una perspectiva diferente durante el análisis. El mismo propone comenzar a construir los servicios a partir de requerimientos puntuales, como por ejemplo, establecer canales de integración punto a punto entre sistemas, o reemplazar soluciones de comunicación remota de aplicaciones por un protocolo multiplataforma como SOAP (Simple Object Access Protocol). Muchas veces estos requerimientos pueden resolverse simplemente

implementando servicios sobre módulos de un sistema ya existente. [9][1]

Las organizaciones podrían ver ventajoso a este modelo ya que les permite integrar sus sistemas utilizando nuevas tecnologías a bajo costo. A pesar de que las implementaciones de este tipo podrían resultar exitosas, es decir, lograr su objetivo de integración puntual, no se encontrarían enmarcadas en una arquitectura diseñada para aprovechar la Orientación a Servicios en su máxima expresión. Las soluciones desarrolladas bajo este Modelo no están concebidas para soportar un gran número de servicios de forma consistente, robusta y ágil.

3.3. MCVS SOA con enfoque Ágil

Con la finalidad de encontrar un enfoque que permita incorporar los principios de arquitectura orientada a servicios en los ambientes de negocio, sin necesidad de esperar que se haya finalizado el proceso en toda la organización, ha surgido el MCVS con enfoque ágil.

La modalidad de trabajo de este modelo difiere de las anteriores ampliamente ya que se ocupa de ejecutar el análisis del negocio en paralelo al diseño de servicios y desarrollo.

Esta forma de trabajo tiene una componente de esfuerzo adicional, con el lógico costo asociado. Esto se debe a la necesidad de tener que ajustar los servicios construidos para alinearlos con los modelos de negocio que pueden ir cambiando a medida que se avanza con el análisis. Las fases que integran este MCVS aplicado a proyectos SOA, son las siguientes:

La fase de análisis debe focalizar en el modelo de negocio. En el momento en que se tiene suficiente conocimiento de las áreas del negocio, se va a comenzar con el trabajo en paralelo de modelado de servicios de negocio. Este punto de inflexión, denominado el punto de maduración del análisis de negocio, debe ser determinado apelando al sentido común y experiencia.

a. Análisis

Si es muy temprano para comenzar con el modelado de servicios, seguramente se requerirá un trabajo de reingeniería para adaptar los servicios al modelo final de negocio, y si por el contrario, se espera demasiado para comenzar por los servicios, se estará perdiendo la agilidad que podría destacar a este modelo frente a los dos precedentes.

b. Diseño, Desarrollo, Pruebas e Implantación

Estas fases van a ser ejecutadas en paralelo a la etapa de análisis a medida que nuevos servicios se incorporan a la arquitectura. Se trabaja con el diseño de cada componente de servicio que surge del análisis. Luego se desarrollan, prueban e implantan los servicios diseñados.

c. Revisión

Se deben efectuar revisiones periódicas de la arquitectura actual contra los modelos de negocio obtenidos. A partir de estas revisiones, cuya intención es encontrar inconsistencias entre la implementación y la realidad, surgirán planes de ejecución de adaptaciones de los servicios construidos para alinearlos con las necesidades actuales. Cada servicio que deba ser modificado, tendrá que pasar por las etapas de diseño (o mejor dicho rediseño), desarrollo, pruebas e implantación nuevamente

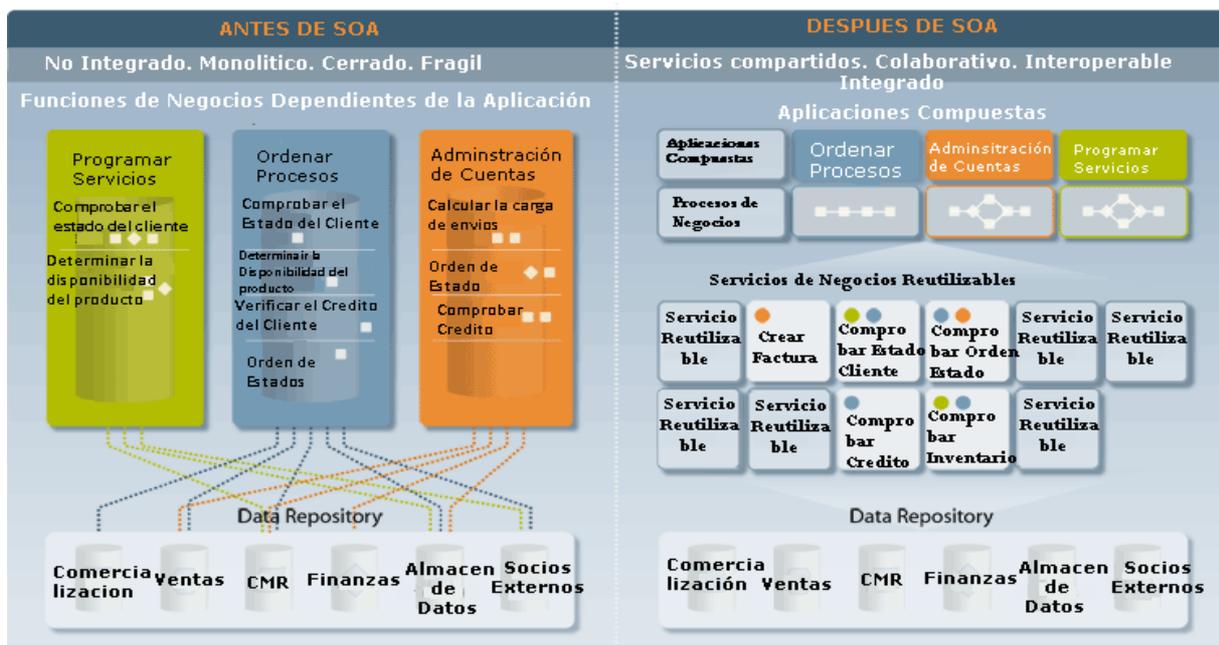


Figura 1. Arquitectura Orientada a Servicios. www.es.sun.com/practice/software/soa. Septiembre-2008

3.4. MCVS con enfoque RUP + XP

Existe una tendencia por la adopción de una metodología mixta, la cual toma una u otra forma dependiendo del momento del ciclo de vida en el cual se encuentra el proyecto SOA.

Un proyecto SOA consta de dos grandes fases de alto nivel. La primera es la construcción de la plataforma SOA. Esta fase requiere la utilización de una metodología como RUP (Rational Unified Process). La segunda fase es la de mantenimiento, en la cual nuevos proyectos son construidos sobre la arquitectura inicial. En esta fase se pueden aprovechar las ventajas de una metodología liviana como XP (eXtreme Programming).

3.5. MCVS SOA con enfoque de Gobierno constante

El modelo con enfoque de gobierno SOA no solo ayuda a las empresas y TI a identificar los mejores proyectos que contribuyan a las metas de la empresa, si no que también habilita a los empleados a trabajar y colaborar de manera más eficiente de definir claramente sus funciones y responsabilidades. [4][1]

a. Modelado

El modelado es el proceso de capturar el diseño de sus negocios a partir de una comprensión de los requisitos y los objetivos de los mismos, y convertir eso en una especificación de los procesos. [5]

Durante esta fase se obtienen y analizan los requerimientos de negocio con el fin de llegar a un modelo de procesos de negocio que será la base para el diseño de servicios y niveles de servicio.

Estos procesos también serán útiles a la hora de medir la eficiencia del negocio.

Es de gran importancia que en esta etapa se fije un modelo común que sea comprendido tanto por gente de IT como de negocio.

b. Ensamble

Una vez que se han definido los procesos de negocio, se deben obtener los servicios necesarios para que los mismos puedan ser incorporados a la solución. [5][1] Para esto puede ser necesaria la construcción de nuevos servicios, utilizar servicios ya existentes, o encapsular funcionalidades de sistemas existentes. Por último, se utilizará un mecanismo de orquestación de servicios que permita que los procesos de negocio cobren vida.

c. Implantación

En esta etapa se debe configurar el ambiente de ejecución de los servicios para lograr cumplir con los niveles de calidad fijados y así poder ejecutar exitosamente los procesos de negocio. Es fundamental que el ambiente de

servicios sea robusto, escalable y seguro. Este ambiente debe estar preparado tanto para correr procesos de misión-crítica como para aceptar cambios de forma flexible.

d. Administración

La fase de Administración incluye establecer y mantener la disponibilidad de los servicios y sus tiempos de respuesta. Se deben monitorear los KPI (Key Performance Indicators) en tiempo real para prevenir, aislar, diagnosticar y solucionar problemas. Es también una tarea a llevar a cabo en esta etapa la de administrar y mantener un control de versiones sobre los servicios que corren los procesos de negocio.

e. Gobierno y Procesos

Este proceso debe ser ejecutado durante todo el ciclo de vida. Se deben establecer políticas y procesos que aseguren el éxito del proyecto SOA. Por ejemplo, se puede crear un centro de excelencia para implementar políticas de gobierno y controlar que los estándares se cumplan.

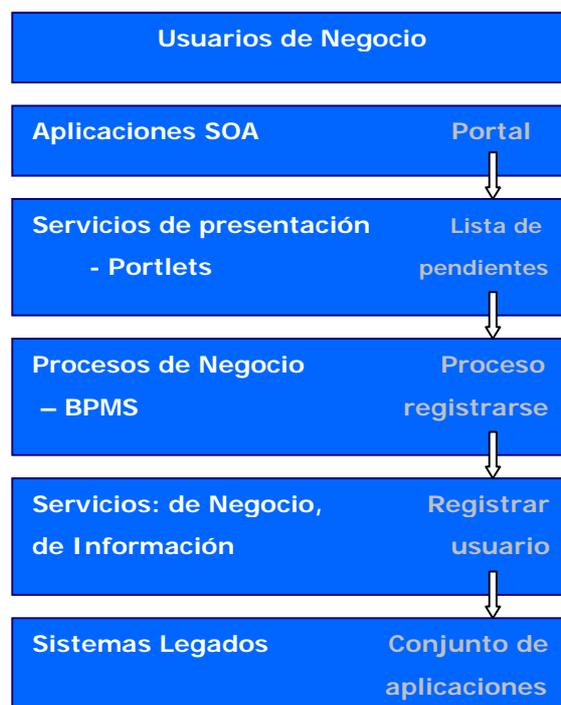


Figura 2. Arquitectura de Referencia SOA[6]

4. RETOS AL ADOPTAR SOA

Uno de los más obvios y comunes retos es el manejo de servicios de metadata. Los ambientes basados en SOA pueden incluir muchos servicios que intercambian mensajes para realizar tareas. Dependiendo del diseño una simple aplicación puede generar millones de mensajes. Manejar y proveer la información sobre como

deben interactuar los servicios puede ser una tarea muy compleja.

Otro de los retos es proveer los niveles apropiados de seguridad. Un modelo de seguridad integrado dentro de una aplicación, puede ya no ser apropiado cuando las funciones de la aplicación están expuestas como servicios que pueden ser usados por otras aplicaciones. Esto significa que el modelo de seguridad basado en aplicaciones ya no es el modelo adecuado para manejar la seguridad de los servicios. Actualmente están surgiendo nuevos modelos de seguridad para SOA.

De igual manera las especificaciones de SOA y Web services están constantemente en expansión, actualización y refinamiento.

5. SOA Y WEB SERVICES

Diferencia entre SOA y Web services.

Desde el punto de vista conceptual, SOA se puede implementar con múltiples tecnologías [13]

- MOM, POJOs, EJBs, DCOM, CORBA, COBOL, ...
- Aunque cuando se consigue realmente la máxima reutilización es cuando se utilizan los estándares más ampliamente soportados [12] [15].
- Reutilización de funcionalidades por otros consumidores.
- Reutilización de funcionalidades de otros servicios.
- Aprovechamiento de otras herramientas, al poder interactuar con ellas.
- Conocimiento del personal.
- Capacidades ofrecidas por estándares adicionales.

Web Services representan una forma de poder realizar ciertos aspectos técnicos de SOA, mas ellos mismos te pueden introducir ciertos problemas. [16] Primero, los estándares no son lo suficientemente maduros para garantizar interoperabilidad. Segundo, Web Services por sí solos no garantizan un nivel de bajo acoplamiento apropiado.

Sin embargo, a pesar que SOA es una tendencia de la arquitectura de software que enfatiza en la construcción de aplicaciones intercomunicables y con bajo sentido de acople, es ampliamente aceptado decir que un web service es un SOA, los cuales deben cumplir estos dos requerimientos:

- Las interfaces de comunicación deben estar basadas en un protocolo de transporte como http, ftp, smtp, CORBA.
- Los mensajes deben ser en formato XML.

Por lo tanto, es necesario hacer la advertencia en no caer en aspectos muy específicos de web services, ya que los

mismos no serán el estándar final para la integración de sistemas, por lo que web services se deben de utilizar solamente cuando aspectos de infraestructura específicos son de gran importancia. Además, se debe reconocer cuando no es prudente optar por un SOA, lo que resultaría en un enfoque erróneo para solucionar un problema.

	SOA	Web Service
Arquitectura	Aplicaciones y servicios de datos	Grano fino, poco acoplamiento, Localización de servicios.
Administración	Plan de gobierno: directrices, controles y mecanismos de cumplimiento	WS-Discovery para localizar servicios Web en tiempo de ejecución
Protocolos	Cualquiera es apropiado (SOAP, FTP, JMS, AJAX, REST, SMTP, CICS, etc.)	SOAP, WSDL, UDDI, WS-*
Formato de mensajes	Frecuentemente XML	XML EN SOAP
Estándares	SOA es una metodología, no un estándar	W3C OASIS

Tabla 1. Diferencias entre SOA y Servicios Web

6. CUÁNDO NO ES APTO OPTAR POR UNA ARQUITECTURA SOA [14][16]

Existen situaciones en las cuales no es recomendable optar por un SOA, las cuales son:

1. Cuando se tiene un ambiente de TI homogéneos

Si se utilizan las tecnologías de un solo proveedor, entonces es posible que la sobrecarga adicional de una SOA no fuera eficaz en función del costo, una arquitectura SOA es muchas veces poco práctica. Además, entornos heterogéneos de hardware no podrán beneficiarse de una SOA, a menos que también tenga una infraestructura heterogénea de software, es decir, diferentes sistemas operativos o de middleware.

2. Cuando ocurre en tiempo real el rendimiento es crítico

Confiar en la comunicación asíncrona para proporcionar acoplamiento entre los consumidores y los productores de servicios, SOA no está bien adaptada a las situaciones que requieren aplicar estrictamente los tiempos de respuesta. Mas, SOA es un excelente método para las empresas que buscan formas de acelerar el procesamiento

de sus archivos, sin tener que deshacerse de sus aplicaciones.

3. Cuando las cosas no cambian

Si hay pocas razones para cambiar la lógica de negocio, presentación, flujo de datos, proceso, o cualquier otro aspecto de la aplicación, conversión de estas aplicaciones a un SOA podría no regresar valor suficiente para hacer que el esfuerzo valga la pena.

Claramente, no se puede garantizar que este paradigma es apto para todas las soluciones de software, ya que pueden existir condiciones que impidan que una solución basada en este paradigma tenga realmente el impacto deseado. Sin embargo, la arquitectura orientada a servicios está emergiendo como una solución muy interesante cuando las necesidades de los clientes radican en la flexibilidad, reutilización y bajo acoplamiento.

7. CONCLUSIONES Y RECOMENDACIONES

SOA es una evolución, que permite integrar diferentes tecnologías, buscando interoperabilidad, calidad, control, eficiencia y seguridad; esta arquitectura propone soluciones para construir sistemas capaces de independizar la tecnología del servicio.

Todo esto se puede lograr desarrollando un plan estratégico bien ejecutado que permita alinear los recursos de IT de forma más directa con los objetivos de negocio, logrando así una mayor integración con los clientes y proveedores.

SOA requiere formación para llevarse a cabo, implica cambios y avances, se necesita cultura frente al cambio, metodología e indicadores del mismo para que sea todo un éxito.

Un proyecto SOA se concibe desde la necesidad de mejorar, no es algo puntual, ya que afecta en sus tiempos y medidas a toda la organización, se aborda desde la visión de servicios.

Esta estrategia ha de ser independiente de la tecnología, requiriendo de un plan directo concreto y práctico

8. BIBLIOGRAFIA

- [1] Krafzig et al., IBM. Una propuesta de modelos de ciclo de vida (MCVS) para la integración de los procesos de negocio utilizando Service Oriented Architecture (SOA). 2005.
- [2] SUN, IBM, Reelectura otra mira de la tecnología Informe Dinámico: SOA IBM y sus "10 años" con SOA (diciembre 2006).
- [3] Revista BYTE La Nueva Avalancha SOA febrero 2008.
- [4] Sandy Poi, Betsy Sleight, Jeffrey Viesel, David Snavely, Strategy & Change, (November 2007) IBM, Enabling SOA through organizational change and governance Management.
- [5] IBM, Infraestructura y software [OnLine] <http://www-306.ibm.com/e-business/la/pe/soa/>
- [6] SOA Agenda, Arquitectura de Referencia soa, [OnLine] <http://soaagenda.com/journal/articulos/arquitectura-de-referencia-soa/>
- [7] SUN, The Soa Platform Guide: Evaluate, Extend, Embrace, White Paper February 2006.
- [8] B2B Solutions Group, artículo . Migrating to a Service Oriented Architecture, by Kishore Channanbasavaiah and Kerrie Holley, IBM Global Service, and Edward M. Tuggle, jr. IBM Software Group.
- [9] Doddavula S.K., Karamongikar S. "Designing an Enterprise Application Framework for Service-Oriented Architecture". White Paper. Infosys. Agosto 2005.
- [10] Josuttis, Nicolai M. SOA in Practice. O'Reilly Media, Inc. Agosto, 2007
- [11] Massuthe, Meter y Wolf, Karsten. An Algorithm for Matching Nondeterministic Services with Operating Guidelines. 2006
- [12] Gebhardt, Mike. Serviceorientierte vs. Eventbasierte Architekturen. Johann Wolfgang Goethe – Universität. 2005
- [13] Javier Cámara Coordinador del grupo de "SOA Infrastructure and Governance, SOA y estándares: una pareja inseparable (mayo 2007).
- [14] Blomberg, Jason. When not to use an SOA. <http://www.zapthink.com/report.html?id=ZAPFLA-SH-02162004>. (29.10.2007)
- [15] He, Hao. What is Service Oriented Architecture? <http://webservices.xml.com/pub/a/ws/2003/09/30/soa.html>. (29.10.2007)
- [16] Julián Astorga Campos, Juan Luis Quirós Venegas, Escuela de Ciencias de la Computación e Informática, Un Enfoque Teórico e Intuitivo a la Arquitectura Orientada a Servicios (SOA).