Introducción al SOA

Jesús David Calle Calle [1] and Javier Trujillo Hernández [1]

[1] GMV Innovating solutions
jdcalle@gmvsolutions.es
[2] INDRA
jtrujillo@indra.com

Resumen. La Arquitectura Orientada a Servicios (SOA por sus siglas en inglés) es un estilo de arquitectura de TI que se apoya en la orientación a servicios. La orientación a servicios es una forma de pensar en servicios, su construcción y sus resultados. Un servicio es una representación lógica de una actividad de negocio que tiene un resultado de negocio específico. Los departamentos de IT aún necesitan responder de forma rápida a los nuevos requerimientos de negocio, reduciendo continuamente los costes de IT y absorbiendo e integrando a la perfección nuevos partners y clientes para el negocio. En este capitulo describimos en detalle la SOA y el impacto que ha tenido y tiene actualmente en las empresas. Además, dedicamos un amplio apartado a tratar los beneficios que aporta SOA a los negocios que han decidido utilizarla.

Palabras clave: SOA.

Abstract. Service Oriented Architecture (SOA) is a style of IT architecture that relies on service-orientation. Service-orientation is a way of thinking about services, their construction, and their results. A service is a logical representation of a business activity that has a specific business outcome. IT departments still need to respond quickly to new business requirements, continuously reducing IT costs and seamlessly absorbing and integrating new partners and customers for the business. In this chapter we describe in detail the SOA and the impact it has had and currently has on companies. In addition, we dedicate an extensive section to dealing with the benefits that SOA brings to the businesses that have decided to use it.

Keywords: SOA

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

1 Introdución a SOA

A lo largo de los últimos 40 años los sistemas de IT han crecido exponencialmente, dejando a las empresas manejar la creciente complejidad de las arquitecturas de software. Las arquitecturas tradicionales han alcanzado sus límites, mientras que las necesidades tradicionales de IT aún persisten. Los departamentos de IT aún necesitan responder de forma rápida a los nuevos requerimientos de negocio, reduciendo continuamente los costes de IT y absorbiendo e integrando a la perfección nuevos partners y clientes para el negocio.

La industria del software se ha movido a lo largo de múltiples arquitecturas diseñadas para permitir un procesamiento distribuido completo, lenguajes de programación diseñados para ejecutar sobre cualquier plataforma y para reducir drásticamente los tiempos de desarrollo, así como los millares de productos de conectividad diseñados para permitir una integración de aplicaciones más rápida y mejor. Sin embargo, la solución completa aún está por llegar [1-10].

Ahora, las arquitecturas orientadas al servicio se están promocionando como el siguiente paso en la evolución para ayudar a que las organizaciones de IT alcancen sus más altos retos. Pero las preguntas todavía persisten: ¿Son las SOAs reales? E incluso si se pueden trazar y describir,
¿pueden implementarse en la actualidad? En los últimos tiempos, los WebServices han estimulado las discusiones sobre SOA, pero esta discusión, no es nueva, ya que muchos de sus conceptos han sido perseguidos a lo largo de la última década con tecnologías como CORBA, que extendía la promesa de integrar aplicaciones en plataformas heterogéneas

1.1 El problema de base

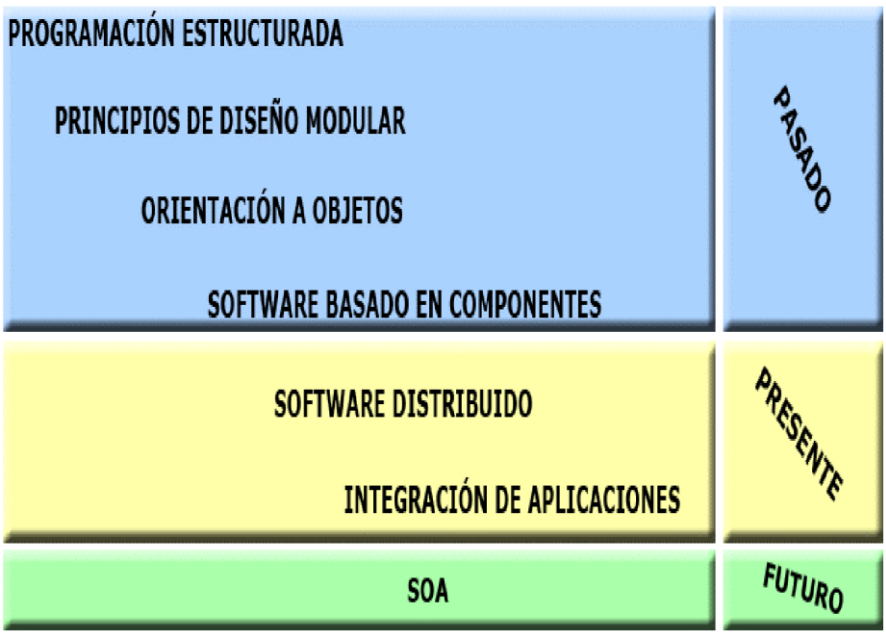

Figura 1: Evolución de las arquitecturas de Software

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

1.2 Los problemas princpales

- La complejidad del entorno tecnológico actual.
- Programación repetida y no reutilizable.
- Múltiples interfaces.

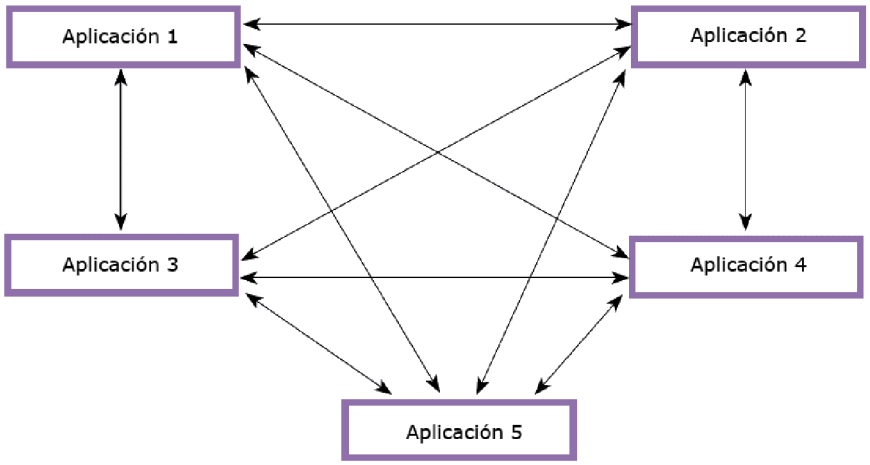

Figura 2: Sistema de múltiples interfaces

1.3 Conceptos fundamentales de SOA

Servicio

Un servicio en SOA es una pieza que expone una funcionalidad con tres grandes características:

- El contrato de interfaz del componente es independiente de toda plataforma.
- El servicio puede ser localizado en invocado dinámicamente.
- El servicio es auto contenido, es decir, el servicio mantiene su propio estado.

Un contrato de interfaz independiente de cualquier plataforma implica que cualquier cliente procedente de cualquier lugar, cualquier sistema operativo, y en cualquier lenguaje, puede utilizar el servicio.

El descubrimiento dinámico indica que un servicio de descubrimiento (por ejemplo, un servicio de directorio) está disponible. El servicio de directorio habilita un mecanismo de localización al que los consumidores pueden acudir para buscar un servicio en base a un determinado criterio.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

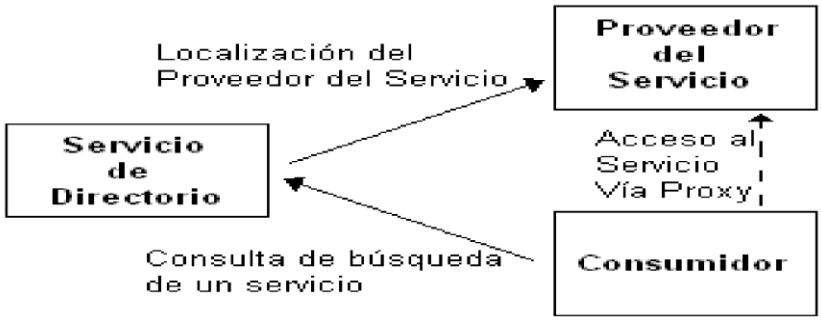

Figura 3: Servicio en SOA

Las funcionalidades de negocio son, desde el punto de vista de las aplicaciones, funciones atómicas no pertenecientes al sistema. Las transacciones de negocio pueden parecer una simple función para la aplicación solicitante, pero deben ser implementadas como una composición de funciones cubiertas por su contexto transaccional individual. Pueden involucrar múltiples funciones de bajo nivel, transparentes para el consumidor.

Las funciones del sistema son funciones genéricas que pueden ser abstraídas de una plataforma en particular como Microsoft Windows o Linux. Esto puede parecer una distinción de servicios algo artificial. Podemos estar de acuerdo en que, desde el punto de vista de las aplicaciones, todos los servicios son atómicos; es irrelevante si son funciones de negocio o de sistema. La distinción se hace necesaria cuando intentamos introducir el concepto de granularidad. Los servicios pueden ser funciones de bajo-nivel (simples) o de alto-nivel (complejas), es decir, de grado fino o grano grueso, y suele haber una diferencia importante en el rendimiento, flexibilidad, reutilización y mantenibilidad basándonos en estas definiciones. Este proceso de definición de servicio viene acompañado normalmente de una visión a mayor plazo.

Esta es la tarea real que debemos hacer; es decir, la aplicación de una arquitectura de desarrollo basada en componentes.

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239

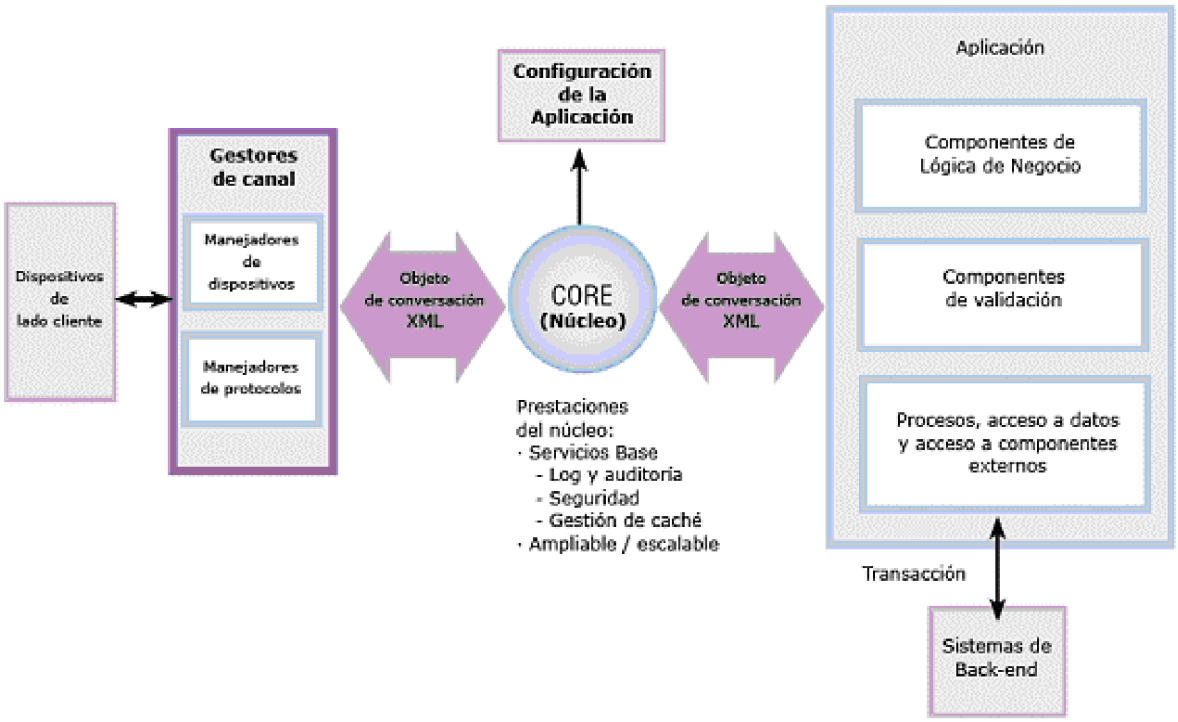

Figura 4: Naturaleza del servicio de SOA

Mensaje

Los proveedores de servicios y sus consumidores se comunican vía mensajes. Los servicios exponen un contrato de interfaz que define el comportamiento del servicio y los mensajes que recibe y entrega. Gracias a que el contrato de interfaz es independiente de cualquier plataforma o lenguaje, la tecnología empleada para la definición de los mensajes puede ser igualmente independiente. Por lo tanto, los mensajes son construidos típicamente con documentos XML que conforman un esquema XMLXML ofrece toda la funcionalidad, granularidad y escalabilidad que los mensajes necesitan. Así, para que los proveedores y los consumidores se comuniquen de forma eficaz, necesitan un tipo de sistema no restrictivo que les permita definir mensajes con total claridad, y XML lo es.

Como los mensajes son el mecanismo de comunicación entre proveedor y consumidor, la definición de los mensajes no puede ser tomada a la ligera. Los mensajes tendrán que ser implementados posteriormente con una tecnología concreta que permita cumplir con los requerimientos escalables de los mensajes. Si los mensajes tienen que ser redefinidos, es muy probable que tengamos que readaptar la interfaz del servicio, lo cual, seguramente sea un proceso costoso.

Descubrimiento dinámico

El descubrimiento dinámico es una pieza fundamental de SOA. En un alto nivel, SOA se compone de tres piezas centrales: proveedores de servicios, consumidores de servicios y servicio de directorio. El rol de los proveedores y los consumidores es evidente, pero el rol del servicio de directorio requiere que se entre un poco más en detalle.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

El servicio de directorio es un intermediario entre proveedores y consumidores ofreciendo un listado de los servicios disponibles, donde los proveedores publican sus servicios a ofrecer y los consumidores acuden para localizar a los proveedores de los servicios consultados. La mayoría de los servicios de directorio organizan y clasifican sus servicios mediante algún criterio. Los consumidores de servicios pueden entonces utilizar las capacidades de búsqueda de los directorios de servicios para localizar a dichos proveedores.

Incorporar un servicio de directorio a SOA implica lo siguiente:

- Escalabilidad de servicios; se pueden añadir servicios de forma incremental.
- Desacoplamiento entre consumidores y proveedores.
- Permite la actualización en caliente de los servicios.
- Ofrece servicios de localización a los consumidores.
- Permite a los consumidores elegir entre proveedores en tiempo de ejecución más allá de escribir en código la localización de un proveedor (hard-code).

WebService

Pese a que los conceptos que sustentan SOA fueron establecidos mucho antes que los WebServices apareciesen, éstos juegan un rol prioritario dentro de SOA. Esto se debe a que los WebServices se construyen empleando protocolos muy bien conocidos e independientes de toda plataforma [11-17].

Estos protocolos incluyen HTTP, XML, UDDI (Universal Description Discover and Integration), WSDL (Web Services Description Language) y SOAP (Simple Object Access Protocol). Es la combinación de estos protocolos lo que hace tan atractivos los WebServices.

Además, son estos protocolos los que cumplen los requerimientos fundamentales de SOA, ya que SOA requiere que un servicio pueda ser descubierto e invocado dinámicamente. Este requerimiento es cumplimentado por UDDI, WSDL y SOAP.

Además éste servicio requerido por SOA ha de contar con un contrato de interfaz independiente de toda plataforma, lo cual es satisfecho mediante XML. SOA hace hincapié en la interoperabilidad, la cual, es satisfecha mediante HTTP. Por esta razón, los WebServices reposan en el corazón de SOA.

1.4 Requerimientos para una SOA

Aprovechamiento de los Assets existentes

Este es el requerimiento más importante. Rara vez podemos prescindir de los sistemas existentes de golpe, y probablemente contengan en su interior datos de gran valor para la empresa.

De forma estratégica, el objetivo es construir una nueva arquitectura que ofrezca todo el valor que se espere de ella, a la que los sistemas existentes cedan todos sus datos y servicios, pero de forma

táctica, los sistemas existentes tienen que integrarse con la nueva arquitectura y con el tiempo, podrán ser componentizados o reemplazados por proyectos incrementales más manejables.

Soporte a todo tipo de integración necesaria

Esto incluye la integración de los usuarios (para ofrecer una experiencia al usuario unifica y simple), conectividad entre aplicaciones (habrá que desarrollar una capa de comunicaciones sobre la que repose toda la arquitectura), integración de procesos (para orquestar aplicaciones y servicios), integración de la información (para clasificar y mover toda la información de la compañía) y construir para

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239

integrar (construir y explotar nuevas aplicaciones y servicios) [18-26].

Permitir la implementación y migración de Assets incremental

Completando este requerimiento podremos cubrir uno de los aspectos más críticos en el desarrollo de una arquitectura: la capacidad de producir ROI incremental. Innumerables proyectos de integración han fracasado por culpa de su complejidad, coste y planificaciones imposibles.

Construir un framework que cumpla con los estándares de componentes

Se debe incluir un entorno de desarrollo construido en torno a un framework que cumpla estándares de componentes para lograr una mayor reutilización de módulos y sistemas, permitir que Assets tradicionales puedan migrarse al nuevo framework y estar abierta a la implementación futura de nuevas tecnologías.

Permitir la implementación de nuevos modelos de computación

Ejemplos específicos de este requerimiento incluyen los nuevos modelos basados en portales para clientes, computación en red y computación bajo demanda.

1.5 SOA, el cambio en el enfoque

Cualquier organización dedicada a la IT consta de muy diferentes partes, cada una de las cuales contribuye de forma equitativa a los objetivos de que la IT cubra las necesidades del negocio. Cada una de estas partes tiene unos requerimientos específicos que gestionar, como se muestran a continuación:

- Sistemas	- Redes	- Aplicaciones
- Información	- Servicios	- Procesos
- Bases de datos	- Repositorios	- Integración
- Almacenes de conocimiento	- Migraciones	- Y más

Para que este mundo de orientación al servicio se haga realidad, las compañías tienen que moverse hacia un nuevo enfoque arquitectural conocido como Arquitectura orientada al servicio (SOA). SOA es la arquitectura que representa la funcionalidad del software como servicios que pueden descubrirse en la red.

Una definición pura de la arquitectura SOA podría ser “una arquitectura de aplicaciones en la cual todas las funcionalidades son definidas como servicios independientes con interfaces bien definidas, las cuales pueden ser requeridas en secuencias establecidas como parte de procesos de negocio”. (Sólo una persona técnica puede comprender esta definición).

Las arquitecturas orientadas al servicio no son nuevas; CORBA (Common Object Request Broker Architecture) y DCOM (Distributed Component Object Model) han provisto funcionalidad similar por largo tiempo. Estos acercamientos a la orientación al servicio ya existentes, sin embargo, sufrieron ciertas dificultades debido a su excesiva dependencia de los escenarios donde aplican.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

1.6 SOA, afrontando viejos problemas

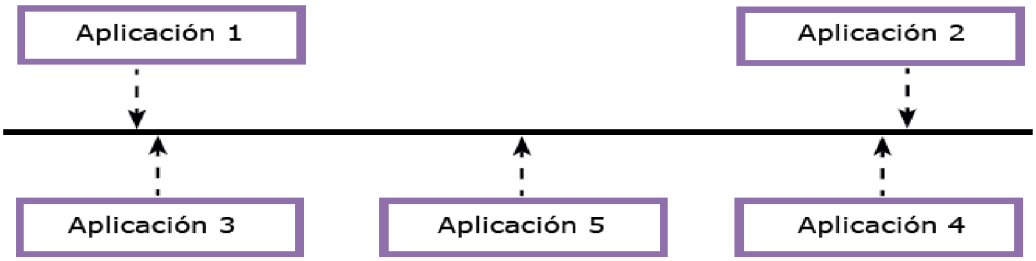

Figura 5: Punta de partida

Este es el punto de partida. Ahora, podemos añadir el punto de vista arquitectural de bus de servicios, representado en la figura siguiente mediante una línea gruesa y un servicio de control de flujos que conecta los servicios y dicta la ruta para cursar las peticiones. El gestor de flujo procesa una secuencia de ejecución, o flujo del servicio, que invocará los servicios en la secuencia apropiada para producir el resultado final.

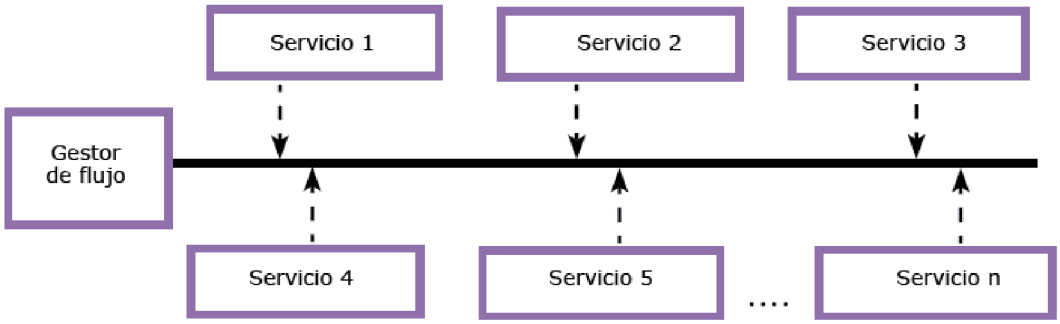

Figura 6: Bus de servicios

En este punto, necesitamos determinar cómo se llama a los servicios, así hay que añadir configuración de las aplicaciones. Entonces, abstraemos las entradas y salidas. Finalmente, tenemos que construir los procesos de conexión con el back-end. El resultado es un framework integral que permite a los procesos ejecutar como tales y los permite evolucionar o migrar en el futuro [27-35].

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

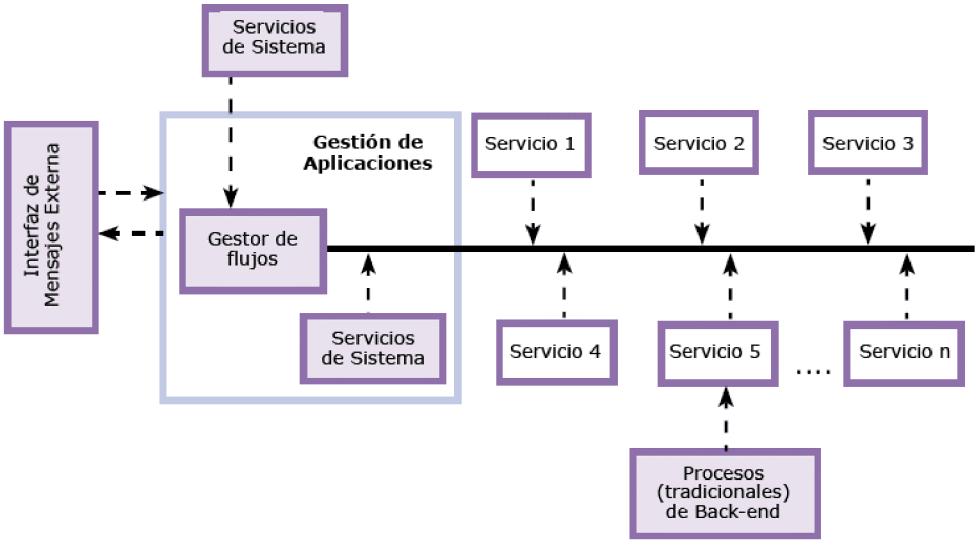

Figura 7: Resultado final: framework integral

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

1.7 El futuro: nuevos modelos y requerimientos

La velocidad a la que se producen los cambios en las empresas deberá aumentar para poder aumentar su eficiencia y rendimiento. Estos requerimientos establecen un conjunto de imperativos de IT en busca de flexibilidad, donde SOA aparece como el elemento capacitador.

Pero, ¿qué sucederá si aparece de pronto un nuevo modelo para el desarrollo de aplicaciones? ¿Seguirán siendo las doctrinas de SOA necesarias o de interés? La respuesta es un rotundo si.

Dos nuevos conceptos han emergido como orientación de la IT, la computación distribuida y la computación bajo demanda. Mientras estos modelos son complementarios y deben implementarse por separado, no dejan de tener una relación muy estrecha; y ambas hacen que la evolución hacia SOA sea incluso más imperativa.

Representando a cada aplicación, recurso o capacidad de negocio como un servicio con una interfaz estandarizada que permite una rápida combinación de aplicaciones nuevas o existentes, las aplicaciones pueden adaptarse fácilmente a las nuevas necesidades de negocio y mejorar la eficiencia operativa.

Computación distribuida

Vamos a mencionar un par de puntos esenciales que merece la pena citar. Primero, la computación distribuida es mucho más que una simple aplicación con un largo número de millones de operaciones por segundo (MIPS) para lograr calcular el resultado de un problema complejo.

Nos permite dividir los recursos en múltiples entornos de ejecución mediante la aplicación de uno o más criterios, tales como el particionamiento del software o el hardware, o tiempo-compartido, simulación de computadoras o la calidad del servicio. Éste despliegue a medida de los recursos compartidos, nos permite usarlos en cualquier momento, estén donde estén dentro de la red.

Esta virtualización es una forma simple de gestionar los dispositivos, almacenes, aplicaciones, servicios y objetos de datos. De ahí que la aplicación de SOA maximice la utilización de los recursos en un entorno distribuido. Permite desplegar y migrar servicios entre grupos de nodos creando pequeños ecosistemas para responder eficientemente a cambios en los entornos tanto interno como externo.

Computación bajo demanda

SOA es un prerrequisito esencial para la computación bajo demanda. SOA es una arquitectura que promociona las aplicaciones bajo demanda. De este modo, las aplicaciones deben operar en una SOA para beneficiarse de los beneficios de la computación bajo demanda.

WebServices es una tecnología que implementa los conceptos de SOA. Como subconjunto de la computación bajo demanda, WebServices bajo demanda son simplemente servicios de negocio publicado usando los WebServices estándar.

La computación bajo demanda puede cubrir un amplio espectro. Uno de los extremos de su espectro apunta al entorno de la aplicación; el otro extremo apunta al entorno operativo, el cual incluye elementos como la infraestructura y la computación autónoma.

La transformación de los negocios significa el aprovechamiento de ambos entornos (aplicación y operativo) para crear un negocio a medida. En el corazón de un negocio a la medida encontramos servicios de negocio bajo demanda donde los servicios a nivel de aplicación pueden ser descubiertos, reconfigurados, ensamblados y servidos bajo demanda, con capacidades de integración just-in-time.

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

2 Beneficios de SOA

Para la arquitectura

- Orientación a Servicio
- Envío de mensajes
- Débil acoplamiento
- Alineación con el ciclo de vida de los procesos de negocio
- Capacidad de integración
- Capacidad para evolucionar aplicaciones e infraestructura existente

Para la computación distribuida

- Ofrece servicios de negocio a través de la plataformas
- Ofrece localización independiente
- Los servicios no necesitan ser específicos de un sistema o red
- Nos acerca totalmente al débil acoplamiento
- Autenticación y autorización a todos los niveles
- La búsqueda y conexión a otros servicios es diná-

mica Implementación a corto plazo

- Aumenta la rentabilidad
- Reduce los costes de la adquisición de hardware
- Aprovechamiento de los skills de desarrollo existentes
- Acelera la adopción de servidores basados en Standard y la consolidación de aplicaciones
- Ofrece un puente de datos entre tecnologías incompatibles

Implementación a largo plazo

- Ofrece la posibilidad de construir aplicaciones compuestas
- Crea un infraestructura automantenible que reduce los costes de mantenimiento
- Permite las aplicaciones de toma de decisión en tiempo real de forma efectiva
- Habilita la compilación de una taxonomía de la información unificada a lo largo de la empresa y sus consumidores y alimentadores

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

3 Visión de negocio

3.1 Visión de mercado: los costes

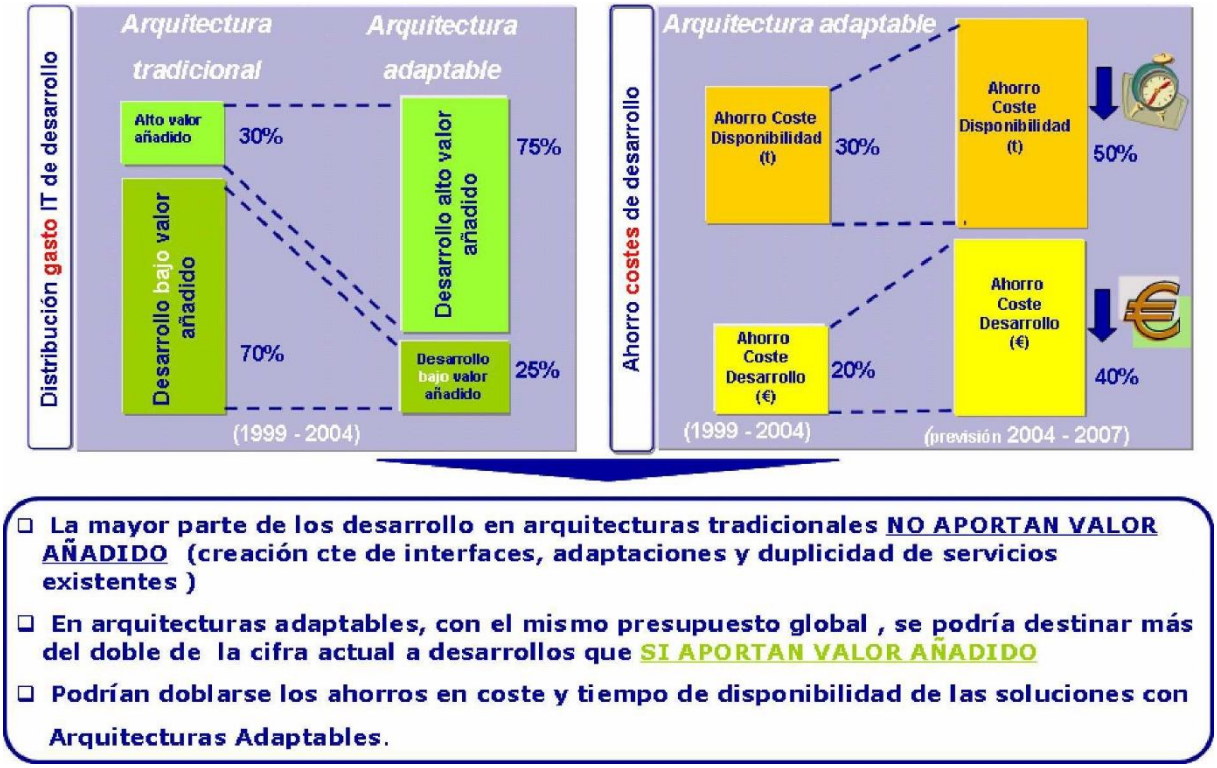

Figura 8: Costes de implementación de distintas arquitecturas

3.2 Organizaciones y arquitecturas adaptables

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

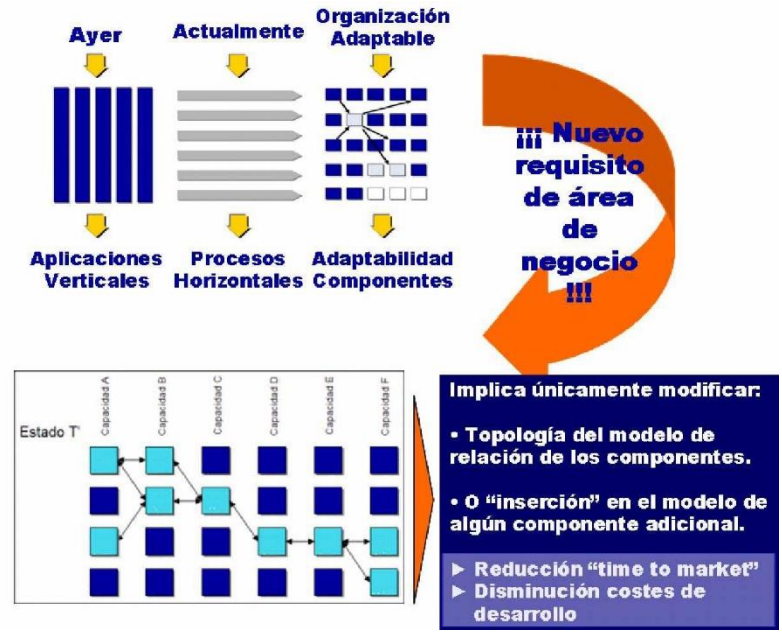

Figura 9: Organizaciones y arquitecturas adaptables

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

3.3 Beneficios de SOA, para el negocio

Aprovechamiento de Assets existentes

Este es el principal beneficio y el más importante. Se puede construir un servicio de negocio como un agregado de componentes existentes, utilizando un framework SOA adecuado para hacerlo y para la compañía. Usar este nuevo servicio requiere conocer únicamente su nombre y su interfaz.

Las especificaciones de la implementación del servicio (tal como su estructura de componentes o el manejo interno de los datos) son transparentes para los usuarios del servicio. Éste anonimato permite a las organizaciones el aprovechamiento de sus inversiones actuales, construyendo servicios desde un conglomerado de componentes construidos en diferentes computadoras, ejecutando diferentes sistemas operativos, desarrollados en diferentes lenguajes [36-40].

Los sistemas tradicionales pueden ser encapsulados y accedidos vía WebServices. Más importante, los sistemas tradicionales pueden ser transformados, añadiendo valor a medida que sus funcionalidades son transformadas en servicios.

Infraestructura como producto

El desarrollo y despliegue sobre la infraestructura se hará más consistente a lo largo de las diferentes aplicaciones de la compañía. Los componentes existentes, los nuevos componentes y los componentes comprados entre una gran variedad de proveedores pueden consolidarse dentro de una SOA bien definida. Tal agregación de componentes será desplegada como servicios en la infraestructura existente.

Como resultado, la capa de infraestructura va convirtiéndose en un producto en sí mismo.

Con el tiempo, a medida que los servicios vayan siendo más desacoplados e independientes del hardware que los ejecuta, podremos optimizar el hardware ya que sólo se conocerá a bajo nivel en tiempo de ejecución.

Time-2-Market más rápido

Las librerías que estructuran los WebServices se convertirán en los Assets del núcleo de la organización como una parte más de nuestra SOA. Construir y desplegar servicios con estas librerías de WebServices reducirá drásticamente el tiempo de posicionamiento de los productos en el mercado, y así como las nuevas iniciativas reutilicen servicios y componentes existentes, se irán reduciendo los tiempos de diseño, desarrollo, testeo y despliegue.

A medida que los servicios en la organización alcancen el nivel deseado dentro de la organización o en las zonas de confianza, aparecerá un ecosistema de servicios que permitirá ensamblar aplicaciones compuestas por servicios, en lugar de desarrollar aplicaciones personalizadas.

Coste reducido

A la vez que el negocio exige adoptar los nuevos requerimientos y tecnologías que emerjan, los costes de crear o mejorar los servicios dentro de nuestra SOA, tanto para aplicaciones existentes como nuevas, se reducen considerablemente, en primer lugar, la curva de aprendizaje del equipo de desarrollo se va reduciendo por el conocimiento adquirido en la producción de los servicios anteriores y la familiaridad con los componentes existentes.

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239

Reducción del riesgo

Reutilizar componentes existentes reduce el riesgo de introducir fallos en los procesos de mejora o creación de nuevos servicios de negocio. También se reduce el riesgo del mantenimiento de la infraestructura que soporta los servicios.

Mejora continua de los procesos de negocio

Una SOA permite representar de forma clara el flujo de los procesos a través del orden de los componentes que se utilizan en un servicio de negocio particular, y ofrece un entorno de auditoria ideal para monitorizar la ejecución de las operaciones de negocio.

El modelado de los procesos queda reflejado en el servicio de negocio. Reajustar un proceso es tan sencillo como reorganizar las piezas de su interior, que puede hacerse mientras se monitoriza, permitiendo alcanzar los resultados óptimos en su ajuste y mejora continua.

Arquitectura centrada en los procesos

Las arquitecturas existentes tienen la tendencia a ser centradas en las aplicaciones. Las aplicaciones son desarrolladas según la conveniencia de los desarrolladores de modo que, a menudo, el conocimiento de un proceso se encuentra repartido entre varios componentes. Una aplicación es como una caja negra para las demás, pero sin granularidad accesible desde su exterior. La reutilización requiere de la copia de código, la incorporación de librerías compartidas o de objetos heredados.

En una arquitectura centrada en el proceso, la aplicación es desarrollada para el proceso, que se descompone en una sucesión de pasos, cada uno de los cuales representa un servicio de negocio. En efecto, cada servicio o componente es una subaplicación. Las subaplicaciones se encadenan para crear un flujo de proceso capaz de satisfacer una necesidad del negocio. Esta granularidad permite a los procesos el aprovechamiento y reutilización de cada subaplicación a lo largo de toda la compañía.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

4 La suma de filosofías, el éxito

4.1 La arquitectura de Software

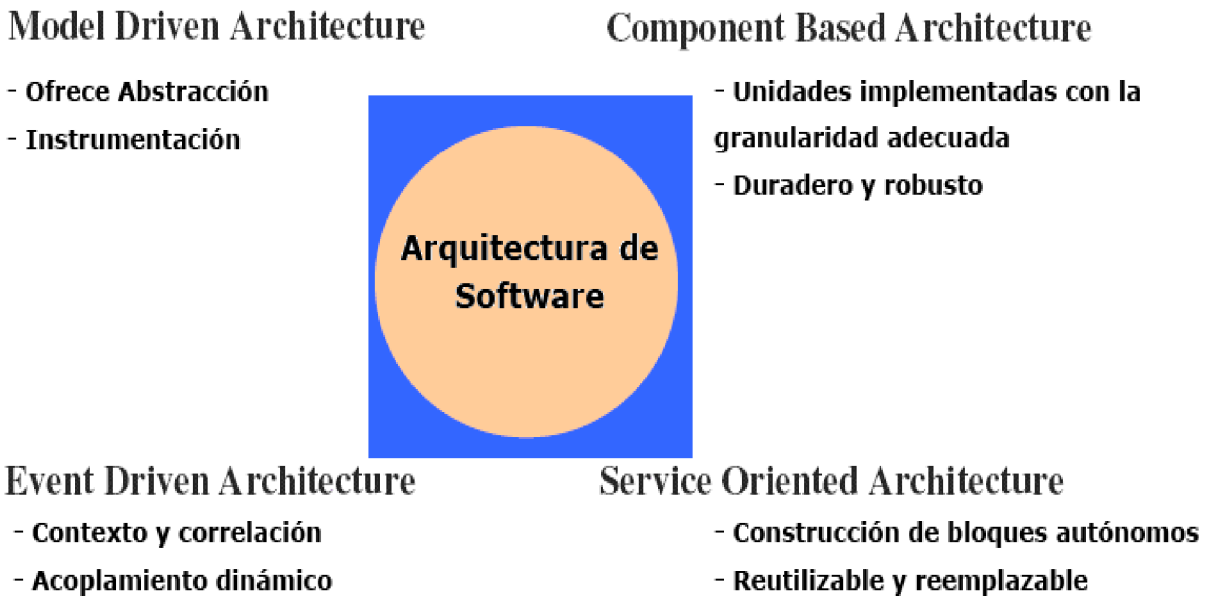

Figura 10: Arquitectura de Software

4.2 Component Based Architecture (CBA)

- Abarca mecanismos y tecnologías para desarrollar componentes de grano grueso vinculados a su entorno/contenedor.
- Los componentes se descomponen en servicios asociados a la lógica de presentación, negocio, acceso a recursos, seguridad, …
- Procede de una evolución de las premisas de la orientación a objetos y aprovecha una extensa infraestructura de tecnologías (como J2EE/J2ME/JAIN o .NET/OSA/Parlay) para la integración.
- Modulariza sistemas grandes (monolíticos) en unidades técnicas de grano grueso, duraderas y reutilizables.
- Los componentes pueden ser desarrollados por terceros e integrados en los sistemas de la empresa.
- La forma de instrumentar los componentes aplicando políticas de seguridad, ha hecho a CBA más poderosa.

4.3 ¿Qué aporta CBA a SOA?

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

- Implementación de unidades con la granularidad adecuada.
- Las unidades siguen orientadas a los objetivos del negocio cuando son desplegadas en los contenedores.
- Construcción de unidades cohesivas, reutilizables y accesibles.
- Construcción de lógica distribuida y reubicable.
- Componentes auto-descritos que soportan la instrumentalización.
- Implementación de servicios complejos mediante el ensamblado de componentes menores.
- Suaviza los trastornos provocados por los cambios
- Y más…

4.4 ¿Cómo aporta valor CBA a SOA?

Componentes cohesivos y robustos generan servicios duraderos; la posibilidad de despliegue "en caliente" permite la refactorización continua.

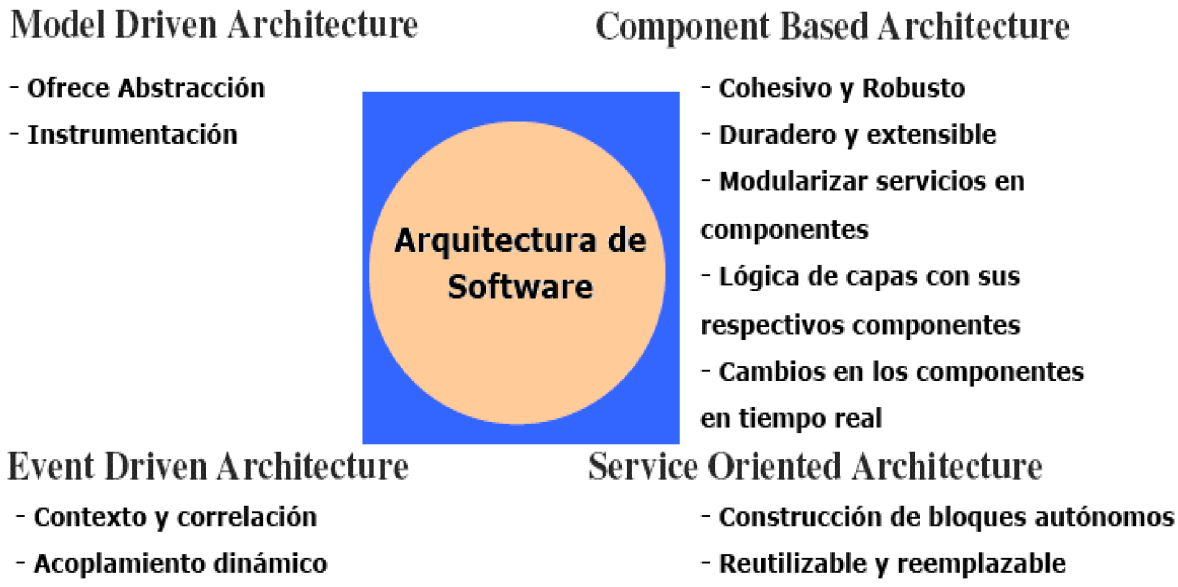

Figura 11: Aportes de valor de CBA a SOA (I)

Cuando se analizan los componentes necesarios para construir una aplicación, elegimos los componentes clave y los promocionamos como core (núcleo) de nuestra aplicación. Al refinar los requerimientos, iremos añadiendo el servicio de la aplicación alrededor de estos componentes, haciéndolos cohesivos y consistentes [41-43].

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

Cuando las necesidades del negocio cambian, y tenemos que introducir nuevas necesidades de negocio en la aplicación, si la abstracción de los componentes de núcleo está bien realizada, el impacto sobre ellos será mínimo, y podremos afrontar los cambios mediante la adición de nuevos componentes a su alrededor (cambio no invasivo).

Esto amplia el tiempo útil de las aplicaciones y soporta algunos conceptos de SOA.

Es importante que los cambios puedan efectuarse en una localización específica de la arquitectura sin afectar al resto de secciones, lo cual posibilita la refactorización constante.

Los cambios lógicos (lógica de negocio, lógica de presentación) pueden afrontarse adecuadamente si aplicamos los principios básicos de CBA, modularizar los componentes en función de la capa que ocupan, lo cual desacopla la lógica global de la aplicación y suaviza el impacto de los cambios.

Gracias a las capacidades de los contenedores de componentes, podemos efectuar cambios en tiempo real (despliegue en caliente) sin afectar a todo el sistema.

Las amplias capacidades de los contenedores de componentes permiten configurar la calidad de los servicios.

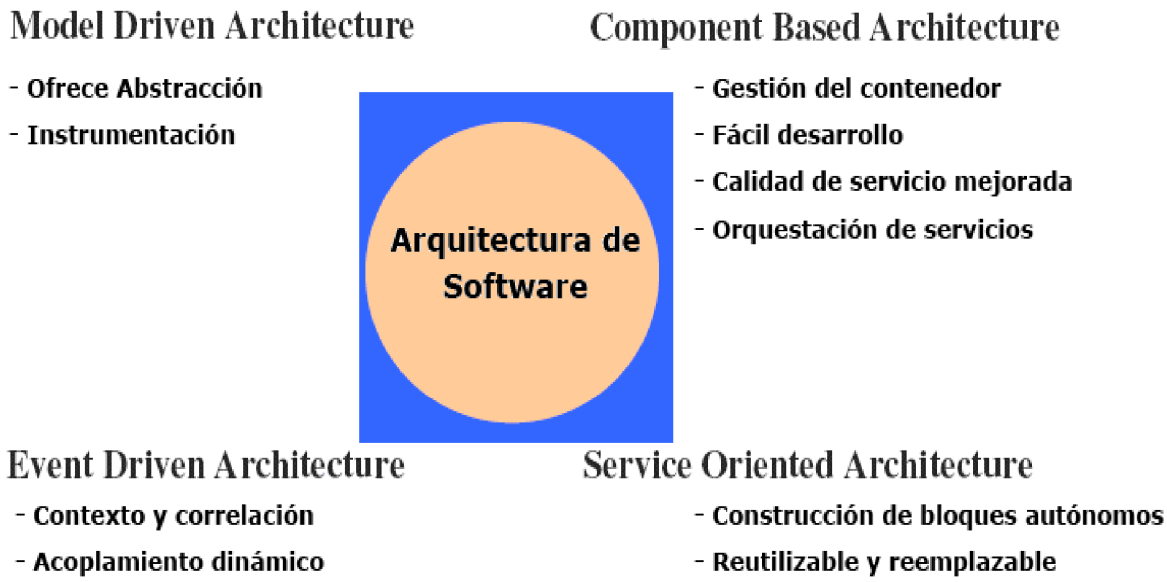

Figura 12: Aportes de valor de CBA a SOA (II)

En CBA, los componentes bien desplegados encajan con los servicios del contenedor (servicios infraestructurales y ortogonales) cobrando sentido en el entorno y contexto de la ejecución. Los componentes pueden utilizar servicios complejos como la gestión de pools mediante simple configuración. Esto permite un desarrollo de componentes rápido y eficiente reutilizando los servicios ortogonales como gestión de recursos, seguridad, … Esto aumenta la calidad del servicio.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

Los servicios empresariales gestionados por los contenedores se han convertido en una estrategia básica de todas las empresas.

Basándonos en la premisa de “un sistema óptimo sólo puede componerse de piezas óptimas”, la orquestación de componentes ofrece un servicio de mayor calidad.

La idea fundamental es la noción de construir sistemas de software con funciones de servicio más funciones de gestión. Las primeras suelen recaer en los arquitectos de aplicaciones, mientras que las segundas lo hacen en los arquitectos de infraestructuras. La tendencia debe ser reducir la separación entre ambos roles.

Para conseguir el desarrollo de servicios centrados en el usuario, debemos :

- instrumentalizar con telemetría (reconocer las condiciones cambiantes)
- instrumentalizar con código para responder bajo condiciones cambiantes.

Distribución de componentes para las interacciones de servicio dinámicas.

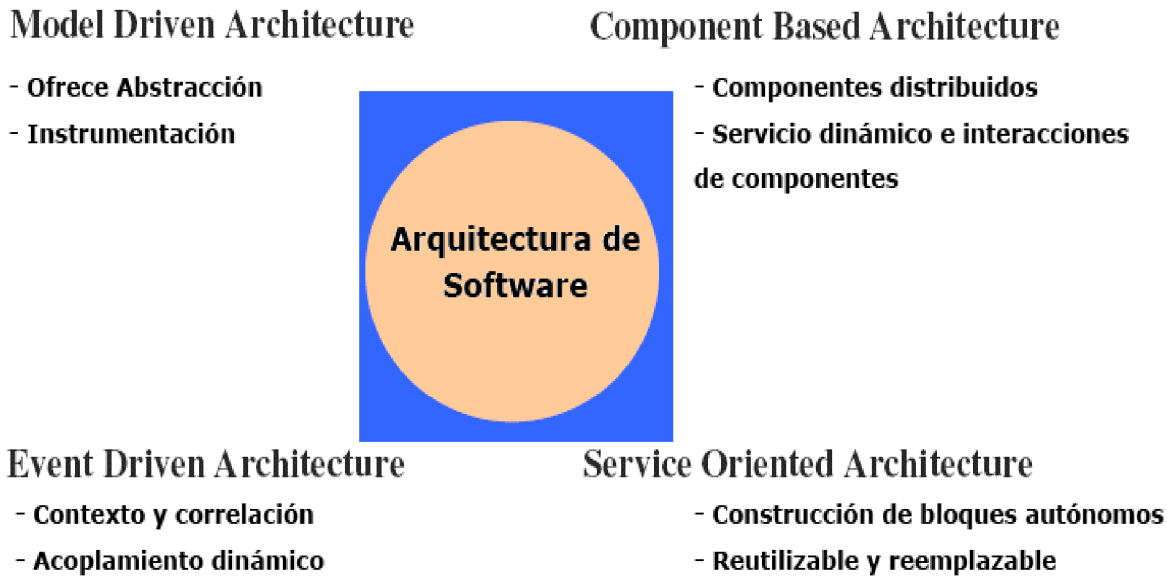

Figura 13: Aportes de valor de CBA a SOA (III)

Tanto en el interior como en el exterior de un servicio encontramos componentes, que se encuentran distribuidos a lo largo y ancho de las redes de computación (intra e inter redes) y entre los servicios de acceso (Cable, WiFi, 3G,…) y los servicios del núcleo (eventos). Esto hace que el servicio sea verdaderamente móvil, accediendo a componentes residentes en las diferentes redes de la compañía Desde el punto de vista de SOA, un componente (de negocio habitualmente) publicado (encontrable y usable) que soporte una interfaz serializada (SOAP/XML), constituye un servicio web. No

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

obstante, le entrega basada en el contexto de este servicio, puede beneficiarse de otros componentes externos para asegurar una interacción dinámica con el usuario. Esta interacción de servicio debe ser controlada y tarificada adecuadamente [44-48].

Los servicios modularizados como componentes distribuidos en la red aseguran:

- El cumplimiento de las premisas de SOA referentes a la movilidad del servicio
- El cumplimiento de las premisas de SOA al respecto del servicio bajo demanda (afectado por los privilegios y configuraciones del perfil del usuario demandante.

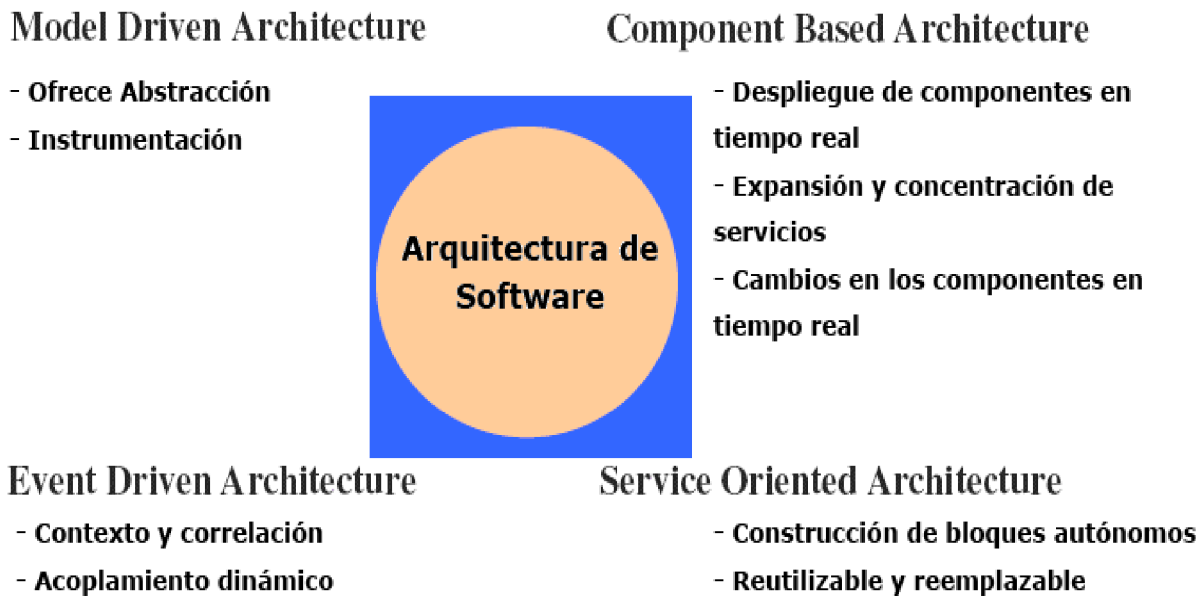

Figura 14: Aportes de valor de CBA a SOA (IV)

Tradicionalmente, cuando construimos un sistema basado en componentes para un determinado público objetivo, el máximo de accesos concurrentes queda controlado, pero cuando desplegamos un servicio como servicio web, los consumidores potenciales aumentan exponencialmente, generando fluctuaciones extremas desde el punto de vista de la calidad de servicio. Es evidente que no podemos afrontarlo sin la elasticidad adecuada del servicio.

Con la movilidad del código y el despliegue en tiempo real, podemos alcanzar una verdadera elasticidad en los servicios. En un momento concreto, los componentes de servicio pueden estar ejecutando en 2 contenedores, y en el momento siguiente en 200.

La componentización de los servicios asegura la alineación de los contenedores de servicios de aplicación con los contenedores de servicios del sistema.

La elasticidad se logra cuando nos damos cuenta de que determinados componentes de servicio mejoran sus prestaciones si son ejecutados en determinados contenedores de la red (por ejemplo, un componente de servicio de criptografía debería ser ejecutado sobre un hardware con cripto- aceleradores.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

Descomposición funcional de los servicios para poder alinearlos.

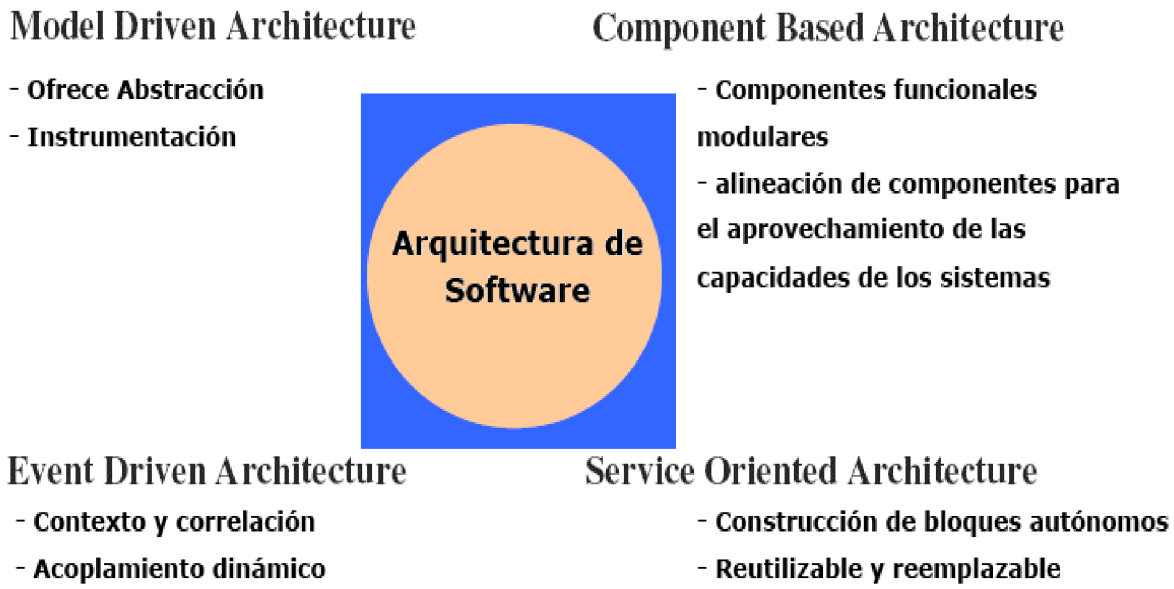

Figura 15: Aportes de valor de CBA a SOA (V)

La descomposición funcional que tiene lugar cuando se construye un servicio basándose en componentes modulares, facilita la adaptación del mismo a los cambios en los requerimientos. Si el servicio se construye de forma monolítica, las operaciones básicas de que consta se enlazarán mediante código (hard-code), por lo que un cambio en una operación nos obligará a recorrer todas las aplicaciones del sistema para repetir el cambio.

El beneficio fundamental de la descomposición funcional se aprecia cuando observamos las posibilidades del aprovechamiento de servicios externos como la seguridad, escalabilidad, disponibilidad o gestión.

La descomposición funcional opera en conjunción con la movilidad del código, orquestación de componentes y su naturaleza distribuida [49-51].

4.5 Event Driven Architecture (EDA)

- Ofrece los mecanismos para coordinar a los solicitantes y los proveedores de servicios (productores y consumidores).

- Los eventos de software permiten una acoplamiento comunicativo a varios niveles, con un amplio espectro de mensajes correlacionados.

- Se crea una red capaz de escuchar miles de eventos procedentes de diferentes orígenes, generando las respuestas adecuadas.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

- Soporta flujos de mensajes dinámicos, paralelos, asíncronos y reacciona a los inputs externos que pueden ser de naturaleza impredecible de antemano.
- Recibida una petición, se elegirá un proveedor del servicio demandado balanceando la carga y ajustándose a los requerimientos de calidad de servicio del demandante.

- El mismo software del bus de eventos puede monitorizar las peticiones y respuestas generando una información de valor añadido tanto para los consumidores como para el negocio.

- La coordinación entre peticiones y respuestas puede ser síncrona o asíncrona.

- Los flujos de ejecución simultánea pueden correr de forma independiente.

4.6 ¿Qué aporta EDA a SOA?

Afinidad de usuarios a la obtención del servicio

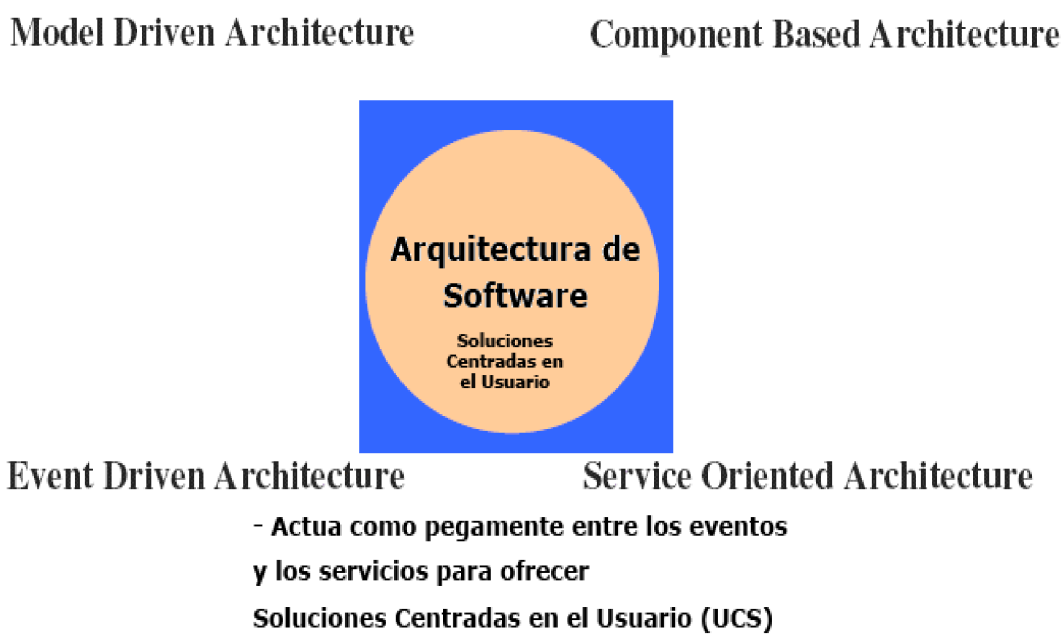

Figura 16: Aportes de valor de EDA a SOA (I)

Las experiencias de los usuarios cuando interactúan con los sistemas abarcan un amplio espectro; la generación de servicios centrados en el usuario, aumentan la calidad del servicio.

Los eventos ofrecen la oportunidad de gestionar el ciclo de vida de un servicio incluso entre sesiones y a través de dispositivos diferentes.

Desde el punto de vista de los usuarios, su interacción con un servicio (servicio web, servicio de comunicación, servicio de negocio, etc.,) tiene lugar sobre una red e implica, en la mayoría de los casos, interacciones complejas.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

La experiencia de interacción del usuario abarca un amplio espectro:

- llamada a servicio simple, llamada – respuesta
- interacción múltiple del usuario para lograr un objetivo del mismo
- múltiples objetivos del usuario realizados a lo largo del tiempo y de las sesiones para obtener una interacción de negocio completa.
- múltiples interacciones de negocio con el usuario, analizándolas para anticipar oportunidades de servicio (creación de portafolio de servicios para el usuario).

Generación de servicio de respuesta ajustado al contexto

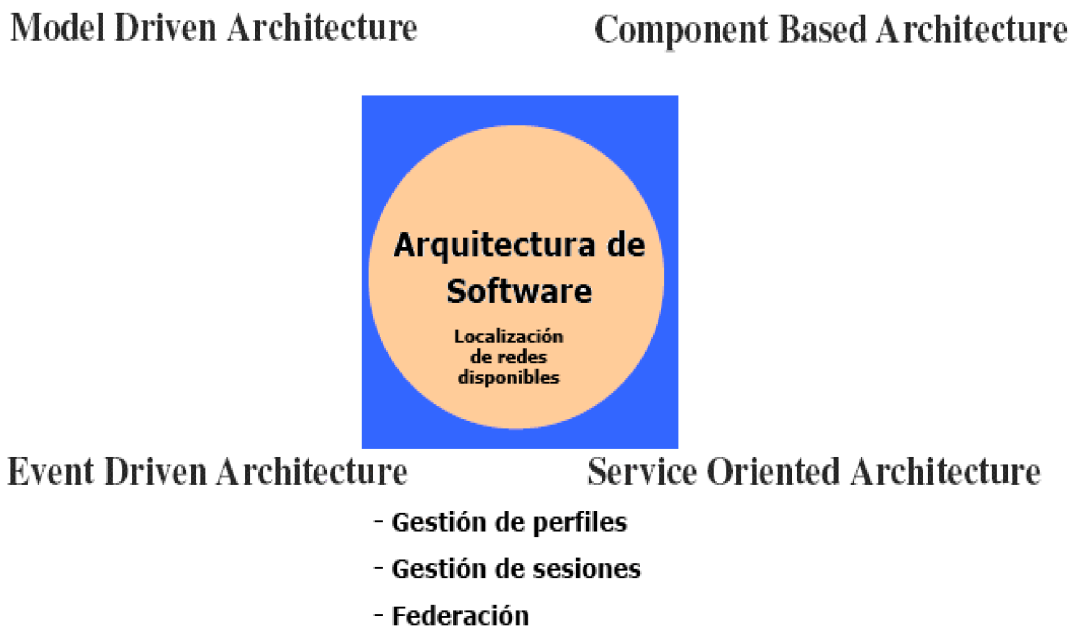

Figura 17: Aportes de valor de EDA a SOA (II)

La solicitud de servicio de un usuario puede desligarse del tiempo (como un servicio de suscripción) y la cumplimentación del servicio puede extenderse en el tiempo.

El mecanismo de eventos transforma el acceso a los servicios de las redes en respuestas ajustadas al contexto particular cuando se realizan los servicios.

Desde un punto de vista global del sistema, la entrega de los servicios a las peticiones de los usuarios pueden involucrar una larga coreografía de servicios para preparar una respuesta simple, de forma inmediata o a lo largo de un plazo de tiempo, o una respuesta compleja puede ser devuelta tras múltiples interacciones, sesiones y formatos. Esta generación de servicios es posible sólo cuando:

- Petición-respuesta se transforma en el paradigma de servicios sentir-entregar
- Redes de servicio pasivas se convierten en proactivas (mediante una combinación con eventos ejecutados en el núcleo)

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

La entrega de agrupaciones de servicios, donde cada uno de ellos persigue objetivos individuales, encajando los unos con los otros, permite por ejemplo que un servicio para un único usuario se convierta en un servicio para otros usuarios concurrentes gestionados por el sistema.

Orquestación y ejecución coherente de servicios

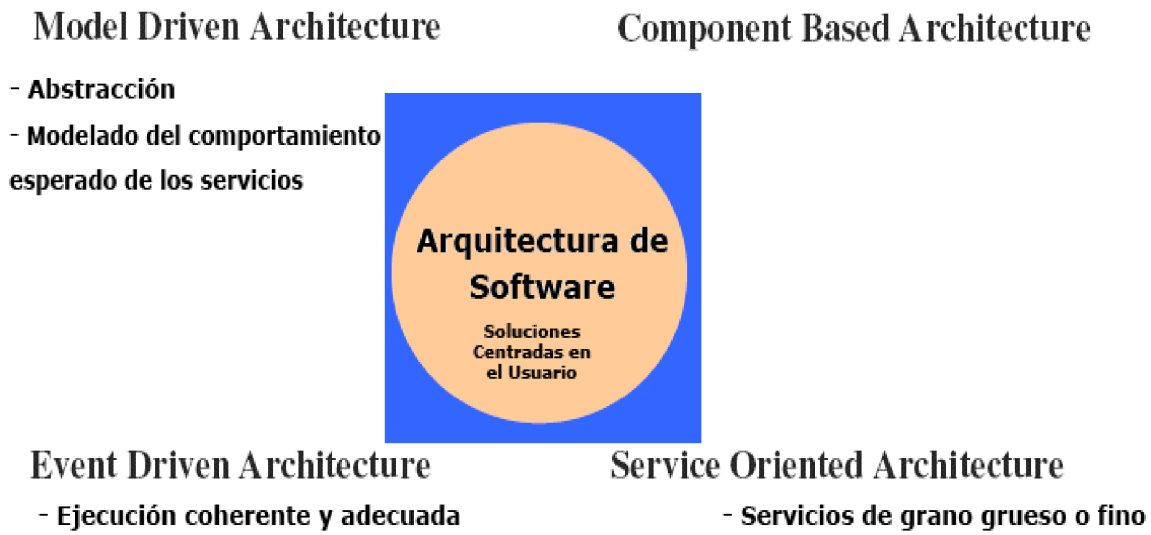

Figura 18: Aportes de valor de EDA a SOA (III)

Los servicios software durante el tiempo de ejecución son siempre parte de un proceso de negocio mayor que satisface un objetivo del mundo real.

Los eventos virtualizados en el sistema representan los eventos del mundo real que al correlacionarse entre ellos, ayudan a agrupar servicios para realizar un proceso de negocio.

La idea de ejecución coherente y orquestación de servicios ofrece una aproximación estandarizada y abierta para conectar diferentes servicios web para ejecutar un servicio de negocio de alto nivel. Esta es una necesidad urgente del mundo actual debido a la amplia modularidad ofrecida por los WebServices en una SOA.

El problema de la correlación de eventos abarca un amplio espectro:

- Correlación entre petición y respuesta a nivel de una simple llamada en modo asíncrono
- Correlación de múltiples recursos a nivel de actividad para realizar transacción
- Correlación de varios estados de transición a nivel de un proceso de negocio

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

Soluciones ágiles componiendo bloques de servicio dinámicamente

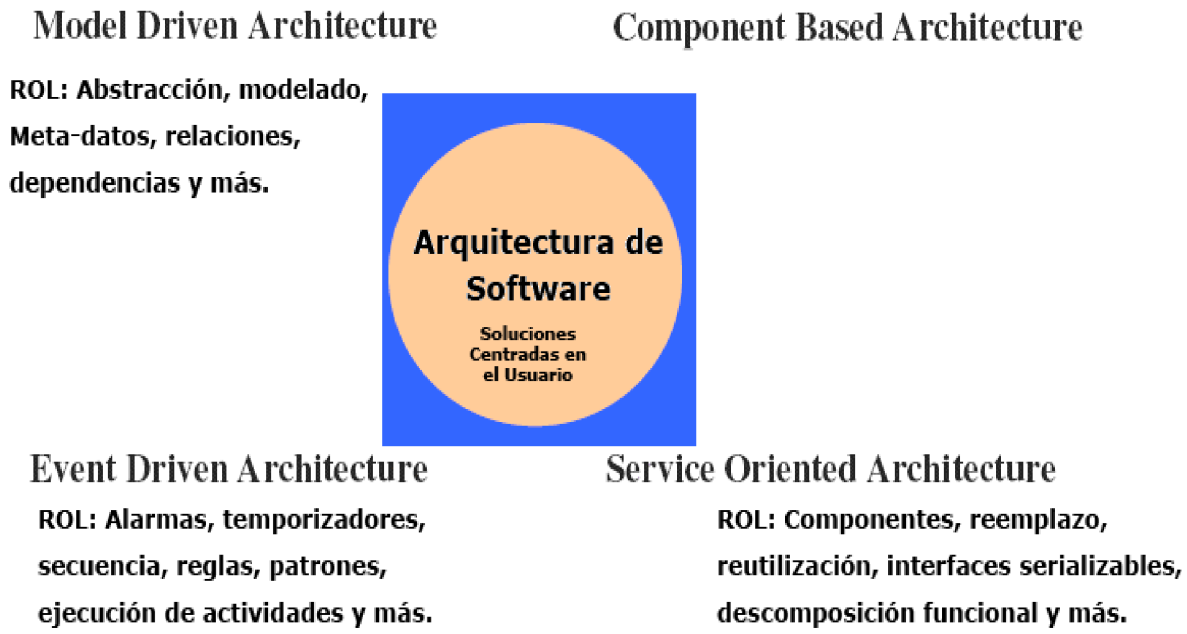

Figura 19: Aportes de valor de EDA a SOA (IV)

Los constructores de SOA deben focalizar el modelado basándose en componentes para cohesionar la arquitectura, y no centrarse en el contexto de su uso o ensamblado. Modelar eventos junto a componentes con meta-datos, permite evaluar el contexto de la ejecución y aumentar la capacidad de ensamblar componentes para completar una solución variable.

En el desarrollo de sistemas vemos a menudo servicios construidos para realizar una parte concreta de una actividad o un proceso de negocios reutilizable en otro u otros contextos. Esta es una solución muy natural en una SOA, en una metodología incremental e iterativa.

Algunas actividades que entorpecen este dinamismo son:

- Desarrollo de componentes para un contexto inicial y un único sistema
- Aplicación de modelos que acoplan los componentes
- El miedo al desarrollo de componentes nuevos que dejen enlaces perdidos dañando la integridad del sistema, pospone la refactorización indefinidamente.

Combinando EDA con SOA, utilizando eventos para coordinar los servicios, obtenemos:

- La granularidad quita el obstáculo del ensamblado de componentes
- Aumentar las capacidades de un sistema horizontal de forma no invasiva. El sistema debe mantener información sobre los usuarios y los servicios que

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

consumen.

4.7 Model Driven Architecture (MDA)

- Los meta-datos guardan un registro de la arquitectura corporativa en términos de datos, información, aplicaciones, servicios, tecnología e implementación.

- El modelado de los sistemas empresariales debe estar en conjunción con los meta-datos.

- Las herramientas básicas necesarias son:

    - Meta-Object Facility (MOF), que define un repositorio estándar para modelos y meta-modelos (un meta-modelo es un caso especial de modelo).

    - XML Meta-Data Interchange (XMI), que define un formato de intercambio de información basado en XML para los modelos y meta-modelos UML.

    - Common Warehouse Meta-model (CWM), que estandariza un meta-modelo completo de la arquitectura que permite realizar minería de datos a través de los límites de las bases de datos. (Como un perfil de UML para el espacio de datos en lugar del de aplicaciones).

4.8 ¿Cómo aporta valor MDA a SOA?

Agrupando, interconectando y emparejando servicios (atómicos)

Un valor fundamental de los meta-datos y meta-modelos para la SOA es la capacidad de interconectar y emparejar servicios a la perfección, a través de una solución de Servicio Integral. El meta-dato/meta-modelo del servicio es la representación de los aspectos comunes de los componentes de servicio (incluyendo el modelado de su funcionalidad y la forma de ejecutarla), con esto, el integrador de servicio puede agrupar, interconectar y emparejar servicios sin interferir en la complejidad interna de tales servicios.

Agrupar e interconectar estos servicios de forma adecuada permite proveer servicios complejos como voz, video y datos a los consumidores.

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

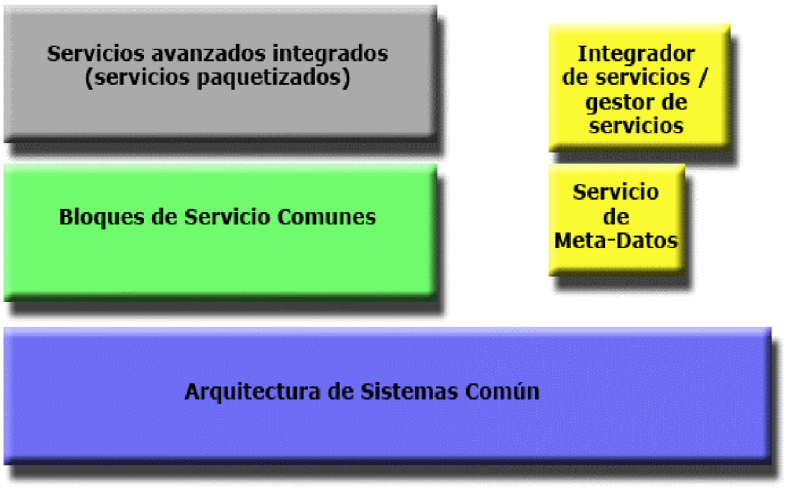

Figura 20: Solución de Servicio Integral

Integración de servicios implementados con múltiples tecnologías

La mayoría de las empresas de cualquier sector se están moviendo hacia una arquitectura de sistemas común (como J2EE). Sin embargo, existen muchas con implementaciones de servicios sobre diferentes plataformas (J2EE y .NET, por ejemplo).

MDA aporta un gran valor sobre las multiplataformas con el PIM (platform independent model) que mapea a PSM (platform specific models) en aplicaciones, interfaces, código, GUI, descriptores, etc. Con PIM como servidor de meta-datos, los integradores de servicio pueden agrupar e interconectar servicios no sólo procedentes de una única plataforma, sino desde plataformas heterogéneas.

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

El único reto de la integración es continuar cumpliendo los requerimientos de nivel de servicio y operación (calidad del servicio) que cada uno de los servicios debía cumplir por separado.

También es posible realizar ingeniería inversa de las aplicaciones existentes para convertirlas en un modelo PIM y redesplegarlo como componentes de servicio.

Las tecnologías existentes detrás de MDA ayudan a traducir modelos de alto nivel de una infraestructura de IT completa, en vez de traducir tan sólo código a código máquina Los generadores de alto nivel cubren una superficie mucho mayor que los recompiladores de fuentes.

Esto permite la creación de servicios y componentes y una SOA totalmente desacoplada de las plataformas e infraestructuras de bajo nivel.

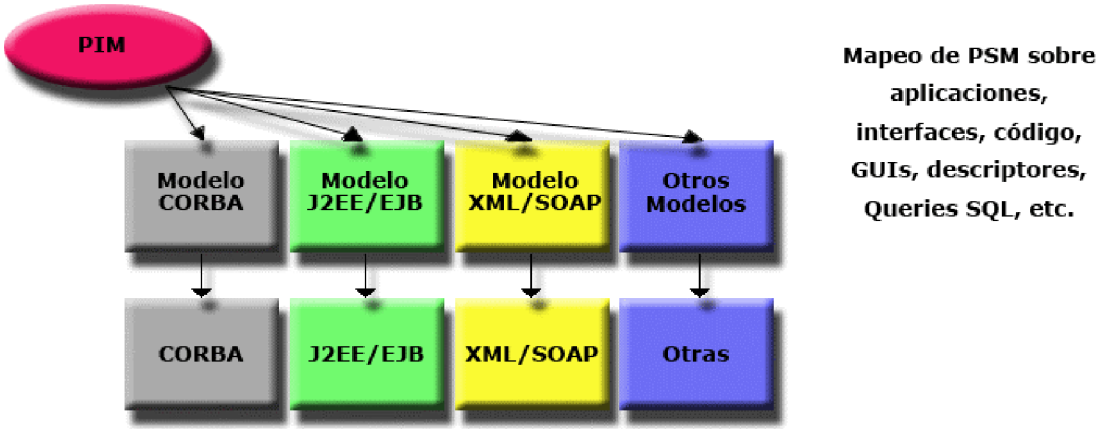

Figura 21:
Mapeo de PSM

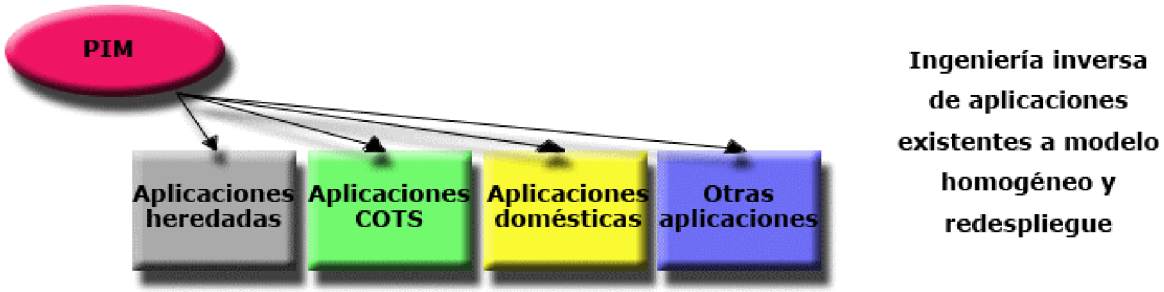

Figura 22: Ingeniería inversa

Construcción de servicios de datos de valor añadido

En un mercado tan competitivo como el nuestro, las empresas que destacan son aquellas que

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

pueden introducir un valor añadido (sensible a los datos) en sus productos/servicios para sus consumidores, empleados y asociados. Los meta-datos y meta-modelos ofrecen un enorme valor a estas empresas (servicios que pueden cambiar cada semana o cada día).

Cuando se trata de empleados o ejecutivos, los servicios que entregan información crítica sobre el rendimiento, inteligencia empresarial, etc. en tiempo real (Business Activity Management) que es derivado del repositorio de meta-datos/meta-modelos (tal como CWM), permitiendo la minería de datos y consultas a las bases de datos actualizadas en tiempo real.

Las técnicas empleadas cuando se utiliza MDA están allanando el camino para desarrollar y proveer servicios bajo demanda y de tiempo real. La mayoría de los datos asociados a estos servicios se obtienen con herramientas de business intelligence (+ BPM y BAM) situados en lo mas alto de los data Warehouse y data marts.

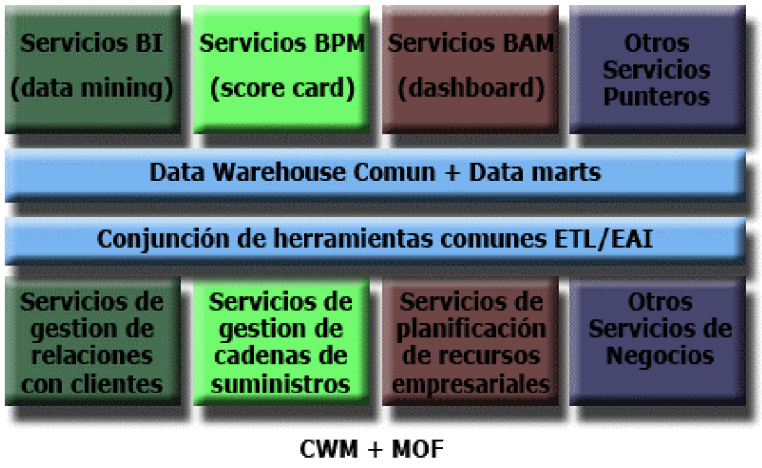

Figura 23: CWM + MOF

Entrega de servicios contextualizados y perfilados al usuario

Otro de los mayores retos de la provisión de servicios en una SOA es la adición de sensitividad respecto al contexto y orientar estos servicios al perfil del usuario.

Los pasos más grandes que se están dando al respecto son los esfuerzos de avanzar hacia un modelo de información común para gestionar la información. Entran en juego las nociones de redes con directorio (DEN) y redes con identidades (IDEN), diseñadas para proveer los bloques de servicio para mejorar la gestión de la inteligencia empresarial (con la aplicación de políticas adecuadas) e integrar estos elementos con otros pertenecientes a la infraestructura.

Esto utiliza los datos existentes de los usuarios y de la empresa presentes en el directorio de la compañía, refuerza los servicios finales (extremo a extremo) y permite la creación de servicios distribuidos a lo largo de la red.

El objetivo es usar el directorio primero dirigiendo a los usuarios a los servicios relevantes y segundo, mantener un subconjunto de datos de gestión. Esto incluye:

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

- Identidad común y administración de seguridad.
- Comprensión común de la gestión de sistemas y servicios.
- Información referente a localizaciones, agrupaciones y políticas.
- Información relativa a la presencia.

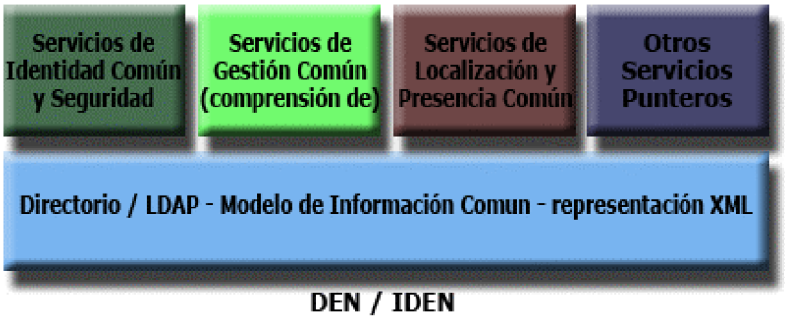

Figura 24: DEN/IDEN

Integración total del negocio

Si una empresa adopta las cuatro áreas indicadas en esta sección;

Agrupación, interconexión y emparejamiento de servicios (integración de bloques de servicio) con servicios de Meta-datos/Meta-modelos y un servicio de integración, integrando servicios implementados con tecnologías heterogéneas usando modelos independientes de plataforma, construyendo servicios dirigidos a ofrecer información de valor añadido con CWM/MOF y ofreciendo servicios ajustados al contexto y perfil de los usuarios aprovechando una DEN/IDEN, la empresa está preparada para la integración total del negocio.

5 Conclusiones

- SOA no es un proyecto SÓLO de tecnología sino que implica cambios organizativos, culturales y metodológicos
- El objetivo es conseguir una arquitectura adaptable que permita ser adaptable a la propia empresa
- La implicación de la dirección en el proyecto es un factor clave de éxito
- El concepto de desarrollo de aplicaciones desaparece para dar paso al de orquestación de servicios que dan soporte a un proceso de negocio
- El diseño de una estrategia de implantación de SOA es crítico para el éxito del proyecto
- ROI: Reducción de costes de desarrollo y mejora de la eficiencia operativa
- Web Services es una implementación física de SOA pero no es SOA en sí mismo. SOA = Web Services

6 La búsqueda de implementación de una política de web services sin una estrategia SOA

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

puede generar duplicidades y falta de eficiencia

7 Agradecimientos y referencias

- Badri Sriraman, Lead IT Architect de Unisys
- Rakesh Radhakrishnan, Enterprise IT Architect de Sun MicroSystems
- Mike Wookey, Modeling engineer de Sun MicroSystems
- IBM, Microsoft y Sun MicroSystems

Direcciones de interés:

- http://www.ondotnet.com
- http://www.devshed.com
- ftp://ftp.software.ibm.com
- http://www.microsoft.com
- http://www.morfeo-project.org

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

References

1. Baruque, B., Corchado, E., Mata, A., & Corchado, J. M. (2010). A forecasting solution to the oil spill problem based on a hybrid intelligent system. Information Sciences, 180(10), 2029–2043. https://doi.org/10.1016/j.ins.2009.12.032
2. Canizes, B., Pinto, T., Soares, J., Vale, Z., Chamoso, P., & Santos, D. (2017). Smart City: A GECAD-BISITE Energy Management Case Study. In 15th International Conference on Practical Applications of Agents and Multi-Agent Systems PAAMS 2017, Trends in Cyber-Physical Multi-Agent Systems (Vol. 2, pp. 92–100). https://doi.org/10.1007/978-3-319-61578-3_9
3. Casado-Vara, R., & Corchado, J. (2019). Distributed e-health wide-world accounting ledger via blockchain. Journal of Intelligent & Fuzzy Systems, 36(3), 2381-2386.
4. Casado-Vara, R., Chamoso, P., De la Prieta, F., Prieto J., & Corchado J.M. (2019). Non-linear adaptive closed-loop control system for improved efficiency in IoT-blockchain management. Information Fusion.
5. Casado-Vara, R., de la Prieta, F., Prieto, J., & Corchado, J. M. (2018, November). Blockchain framework for IoT data quality via edge computing. In Proceedings of the 1st Workshop on Blockchain-enabled Networked Sensor Systems (pp. 19-24). ACM.
6. Casado-Vara, R., Novais, P., Gil, A. B., Prieto, J., & Corchado, J. M. (2019). Distributed continuous-time fault estimation control for multiple devices in IoT networks. IEEE Access.
7. Casado-Vara, R., Vale, Z., Prieto, J., & Corchado, J. (2018). Fault-tolerant temperature control algorithm for IoT networks in smart buildings. Energies, 11(12), 3430.
8. Casado-Vara, R., Prieto-Castrillo, F., & Corchado, J. M. (2018). A game theory approach for cooperative control to improve data quality and false data detection in WSN. International Journal of Robust and Nonlinear Control, 28(16), 5087-5102.
9. Chamoso, P., de La Prieta, F., Eibenstein, A., Santos-Santos, D., Tizio, A., & Vittorini, P. (2017). A device supporting the self-management of tinnitus. In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (Vol. 10209 LNCS, pp. 399–410). https://doi.org/10.1007/978-3-319-56154-7_36
10. Chamoso, P., González-Briones, A., Rivas, A., De La Prieta, F., & Corchado J.M. (2019). Social computing in currency exchange. Knowledge and Information Systems.
11. Chamoso, P., González-Briones, A., Rivas, A., De La Prieta, F., & Corchado, J. M. (2019). Social computing in currency exchange. Knowledge and Information Systems, 1-21.
12. Chamoso, P., González-Briones, A., Rodríguez, S., & Corchado, J. M. (2018). Tendencies of technologies and platforms in smart cities: A state-of-the-art review. Wireless Communications and Mobile Computing, 2018.
13. Chamoso, P., Rodríguez, S., de la Prieta, F., & Bajo, J. (2018). Classification of retinal vessels using a collaborative agent-based architecture. AI Communications, (Preprint), 1-18.
14. Choon, Y. W., Mohamad, M. S., Deris, S., Illias, R. M., Chong, C. K., Chai, L. E., … Corchado, J. M. (2014). Differential bees flux balance analysis with OptKnock for in silico microbial strains optimization. PLoS ONE, 9(7). https://doi.org/10.1371/journal.pone.0102744
15. Christian Paulo Villavicencio, Silvia Schiaffino, J. Andrés Díaz-Pace, Ariel Monteserin (2016). A Group Recommendation System for Movies based on MAS. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 5, n. 3
16. Corchado, J. A., Aiken, J., Corchado, E. S., Lefevre, N., & Smyth, T. (2004). Quantifying the Ocean's CO2 budget with a CoHeL-IBR system. In Advances in Case-Based Reasoning, Proceedings (Vol. 3155, pp. 533–546).
17. Corchado, J. M., & Aiken, J. (2002). Hybrid artificial intelligence methods in oceanographic forecast models. Ieee Transactions on Systems Man and Cybernetics Part C-Applications and Reviews, 32(4), 307–313. https://doi.org/10.1109/tsmcc.2002.806072
18. Corchado, J. M., & Fyfe, C. (1999). Unsupervised neural method for temperature forecasting. Artificial Intelligence in Engineering, 13(4), 351–357. https://doi.org/10.1016/S0954-1810(99)00007-2
19. Corchado, J. M., Borrajo, M. L., Pellicer, M. A., & Yáñez, J. C. (2004). Neuro-symbolic System for Business Internal Control. In Industrial Conference on Data Mining (pp. 1–10). https://doi.org/10.1007/978-3-540-30185-1_1
20. Corchado, J. M., Corchado, E. S., Aiken, J., Fyfe, C., Fernandez, F., & Gonzalez, M. (2003). Maximum likelihood hebbian learning based retrieval method for CBR systems. In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (Vol. 2689, pp. 107–121). https://doi.org/10.1007/3-540-45006-8_11

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239

21. Corchado, J. M., Pavón, J., Corchado, E. S., & Castillo, L. F. (2004). Development of CBR-BDI agents: A tourist guide application. In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (Vol. 3155, pp. 547–559). https://doi.org/10.1007/978-3-540-28631-8
22. Corchado, J., Fyfe, C., & Lees, B. (1998). Unsupervised learning for financial forecasting. In Proceedings of the IEEE/IAFE/INFORMS 1998 Conference on Computational Intelligence for Financial Engineering (CIFEr) (Cat. No.98TH8367) (pp. 259–263). https://doi.org/10.1109/CIFER.1998.690316
23. Costa, Â., Novais, P., Corchado, J. M., & Neves, J. (2012). Increased performance and better patient attendance in an hospital with the use of smart agendas. Logic Journal of the IGPL, 20(4), 689–698. https://doi.org/10.1093/jigpal/jzr021
24. Eduardo Porto Teixeira, Eder M. N. Goncalves, Diana F. Adamatti (2017). Ulises: A Agent-Based System For Timbre Classification. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 6, n. 2
25. Enyo Gonçalves, Mariela Cortés, Marcos De Oliveira, Nécio Veras, Mário Falcão, Jaelson Castro (2017). An Analysis of Software Agents, Environments and Applications School: Retrospective, Relevance, and Trends. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 6, n. 2
26. Fdez-Riverola, F., & Corchado, J. M. (2003). CBR based system for forecasting red tides. Knowledge-Based Systems, 16(5–6 SPEC.), 321–328. https://doi.org/10.1016/S0950-7051(03)00034-0
27. Fernández-Riverola, F., Díaz, F., & Corchado, J. M. (2007). Reducing the memory size of a Fuzzy case-based reasoning system applying rough set techniques. IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews, 37(1), 138–146. https://doi.org/10.1109/TSMCC.2006.876058
28. Fyfe, C., & Corchado, J. (2002). A comparison of Kernel methods for instantiating case based reasoning systems. Advanced Engineering Informatics, 16(3), 165–178. https://doi.org/10.1016/S1474-0346(02)00008-3
29. Fyfe, C., & Corchado, J. M. (2001). Automating the construction of CBR systems using kernel methods. International Journal of Intelligent Systems, 16(4), 571–586. https://doi.org/10.1002/int.1024
30. García Coria, J. A., Castellanos-Garzón, J. A., & Corchado, J. M. (2014). Intelligent business processes composition based on multi-agent systems. Expert Systems with Applications, 41(4 PART 1), 1189–1205. https://doi.org/10.1016/j.eswa.2013.08.003
31. García, O., Chamoso, P., Prieto, J., Rodríguez, S., & De La Prieta, F. (2017). A serious game to reduce consumption in smart buildings. In Communications in Computer and Information Science (Vol. 722, pp. 481–493). https://doi.org/10.1007/978-3-319-60285-1_41
32. Glez-Bedia, M., Corchado, J. M., Corchado, E. S., & Fyfe, C. (2002). Analytical model for constructing deliberative agents. International Journal of Engineering Intelligent Systems for Electrical Engineering and Communications, 10(3).
33. Glez-Peña, D., Díaz, F., Hernández, J. M., Corchado, J. M., & Fdez-Riverola, F. (2009). geneCBR: A translational tool for multiple-microarray analysis and integrative information retrieval for aiding diagnosis in cancer research. BMC Bioinformatics, 10. https://doi.org/10.1186/1471-2105-10-187
34. Gonzalez-Briones, A., Chamoso, P., De La Prieta, F., Demazeau, Y., & Corchado, J. M. (2018). Agreement Technologies for Energy Optimization at Home. Sensors (Basel), 18(5), 1633-1633. doi:10.3390/s18051633
35. González-Briones, A., Chamoso, P., Yoe, H., & Corchado, J. M. (2018). GreenVMAS: virtual organization-based platform for heating greenhouses using waste energy from power plants. Sensors, 18(3), 861.
36. Gonzalez-Briones, A., Prieto, J., De La Prieta, F., Herrera-Viedma, E., & Corchado, J. M. (2018). Energy Optimization Using a Case-Based Reasoning Strategy. Sensors (Basel), 18(3), 865-865. doi:10.3390/s18030865
37. Jaime Rincón, Jose Luis Poza, Juan Luis Posadas, Vicente Julián, Carlos Carrascosa (2016). Adding real data to detect emotions by means of smart resource artifacts in MAS. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 5, n. 4
38. Laza, R., Pavn, R., & Corchado, J. M. (2004). A reasoning model for CBR_BDI agents using an adaptable fuzzy inference system. In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (Vol. 3040, pp. 96–106). Springer, Berlin, Heidelberg.
39. Li, T., Sun, S., Corchado, J. M., & Siyau, M. F. (2014). A particle dyeing approach for track continuity for the SMC-PHD filter. In FUSION 2014 - 17th International Conference on Information Fusion. Retrieved from https://www.scopus.com/inward/record.uri?eid=2-s2.0-84910637583&partnerID=40&md5=709eb4815eaf544ce01a2c21aa749d8f

ISBN: 978-84-9012-859-6
Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc

40. Li, T., Sun, S., Corchado, J. M., & Siyau, M. F. (2014). Random finite set-based Bayesian filters using magnitude-adaptive target birth intensity. In FUSION 2014 - 17th International Conference on Information Fusion. Retrieved from https://www.scopus.com/inward/record.uri?eid=2-s2.0-84910637788&partnerID=40&md5=bd8602d6146b014266cf07dc35a681e0
41. Lima, A. C. E. S., De Castro, L. N., & Corchado, J. M. (2015). A polarity analysis framework for Twitter messages. Applied Mathematics and Computation, 270, 756–767. https://doi.org/10.1016/j.amc.2015.08.059
42. Sittón-Candanedo, I., Alonso, R. S., Corchado, J. M., Rodríguez-González, S., & Casado-Vara, R. (2019). A review of edge computing reference architectures and a new global edge proposal. Future Generation Computer Systems, 99, 278-294.
43. Mata, A., & Corchado, J. M. (2009). Forecasting the probability of finding oil slicks using a CBR system. Expert Systems with Applications, 36(4), 8239–8246. https://doi.org/10.1016/j.eswa.2008.10.003
44. Méndez, J. R., Fdez-Riverola, F., Díaz, F., Iglesias, E. L., & Corchado, J. M. (2006). A comparative performance study of feature selection methods for the anti-spam filtering domain. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 4065 LNAI, 106–120. Retrieved from https://www.scopus.com/inward/record.uri?eid=2-s2.0-33746435792&partnerID=40&md5=25345ac884f61c182680241828d448c5
45. Palomino, C. G., Nunes, C. S., Silveira, R. A., González, S. R., & Nakayama, M. K. (2017). Adaptive agent-based environment model to enable the teacher to create an adaptive class. Advances in Intelligent Systems and Computing (Vol. 617). https://doi.org/10.1007/978-3-319-60819-8_3
46. Rafael Cauê Cardoso, Rafael Heitor Bordini. (2017) A Multi-Agent Extension of a Hierarchical Task Network Planning Formalism. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 6, n. 2
47. Rafhael Cunha, Cleo Billa, Diana Adamatti (2017). Development of a Graphical Tool to integrate the Prometheus AEOlus methodology and Jason Platform. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 6, n. 2
48. Rodríguez, S., De La Prieta, F., Tapia, D. I., & Corchado, J. M. (2010). Agents and computer vision for processing stereoscopic images. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (Vol. 6077 LNAI). https://doi.org/10.1007/978-3-642-13803-4_12
49. Rodríguez, S., Gil, O., De La Prieta, F., Zato, C., Corchado, J. M., Vega, P., & Francisco, M. (2010). People detection and stereoscopic analysis using MAS. In INES 2010 - 14th International Conference on Intelligent Engineering Systems, Proceedings. https://doi.org/10.1109/INES.2010.5483855
50. Román, J. A., Rodríguez, S., & de da Prieta, F. (2016). Improving the distribution of services in MAS. Communications in Computer and Information Science (Vol. 616). https://doi.org/10.1007/978-3-319-39387-2_4
51. Tapia, D. I., & Corchado, J. M. (2009). An ambient intelligence based multi-agent system for alzheimer health care. International Journal of Ambient Computing and Intelligence, v 1, n 1(1), 15–26. https://doi.org/10.4018/jaci.2009010102

ISBN: 978-84-9012-859-6

Colección Aquilafuente: AQ-239
Ediciones Universidad de Salamanca - cc by nc dc