

Servicios Web

Desarrollo de Software Basado en Componentes y Servicios

M.I. Capel

ETS Ingenierías Informática y
Telecomunicación

Departamento de Lenguajes y Sistemas Informáticos
Universidad de Granada
Email: manuelcapel@ugr.es

DSBCS

Máster en Ingeniería Informática



Índice

- 1 **Introducción**
- 2 Sistemas basados en interacción entre pares (P2P)
 - Tecnologías para SW
 - Tecnologías para SW II
- 3 Desarrollo Basado en Servicios Web de Procesos de Negocio
- 4 Composición de Servicios Web
 - Orquestación
 - Coreografía

Índice

- 1 Introducción
- 2 Sistemas basados en interacción entre pares (P2P)
 - Tecnologías para SW
 - Tecnologías para SW II
- 3 Desarrollo Basado en Servicios Web de Procesos de Negocio
- 4 Composición de Servicios Web
 - Orquestación
 - Coreografía

Índice

- 1 Introducción
- 2 Sistemas basados en interacción entre pares (P2P)
 - Tecnologías para SW
 - Tecnologías para SW II
- 3 Desarrollo Basado en Servicios Web de Procesos de Negocio
- 4 Composición de Servicios Web
 - Orquestación
 - Coreografía

Índice

- 1 Introducción
- 2 Sistemas basados en interacción entre pares (P2P)
 - Tecnologías para SW
 - Tecnologías para SW II
- 3 Desarrollo Basado en Servicios Web de Procesos de Negocio
- 4 Composición de Servicios Web
 - Orquestación
 - Coreografía

Fundamental



- Se habla de primera y segunda generación de Servicios Web
- SOA *primitivo* se corresponde con Servicios Web de Primera Generación
- SOA *contemporáneo* se corresponde con Servicios Web de Segunda Generación

Recapitulación: Caracterización de SOA

¿Qué es SOA?

Una arquitectura flexible y estándar que unifica los procesos de negocio estructurando grandes aplicaciones como colecciones de servicios.

Estas aplicaciones pueden ser usadas por usuarios dentro o fuera de la organización.

Estilo arquitectónico

SOA es también un estilo de arquitectura donde todas las funcionalidades, ya sean existentes o nuevas, son agrupadas en servicios que se comunican entre sí.

Recapitulación: Principios de Diseño de SOA

Reglas para desarrollar, mantener y usar servicios

- *Encapsulamiento* (en servicios SOA): Para los servicios que no estén diseñados para ser parte de SOA
- *Acoplamiento Débil*: Independencia entre servicios, sólo se “conocen”
- *Contratos*: Los servicios deben cumplir con unas reglas de comunicación preestablecidos en documentos
- *Abstracción*: La lógica interna de los servicios no tiene por qué ser conocida por el *ambiente* en el que se ejecutan

Recapitulación: Principios de Diseño de SOA II

Reglas para desarrollar, mantener y usar servicios

- *Reutilización*: Los servicios deben ser diseñados siempre para su reutilización por varias aplicaciones *remotas*
- *Composición*: Colecciones de servicios han de poder ser compuestos para formar otros servicios
- *Autonomía*: Los servicios son los únicos que controlan su lógica interna
- *Optimización*: Deben ser lo más eficientes posible
- *Descubrimiento*: Los servicios son diseñados para facilitar su descubrimiento en el ambiente en que se ejecutan

Tecnologías de soporte SOAs

Modelos basados en objetos distribuidos	Sistemas basados en Infraestructuras componentes software (Middleware)	<u>Sistemas Basados en Agentes</u>	Sistemas Cliente- Servidor	<i>Transportabilidad</i>
				<i>Extensibilidad</i>
				<i>Concurrencia</i>
				<i>Confiabilidad</i>
				<i>Apertura</i>
				<i>Interoperabilidad</i>
				<i>Reconfiguración Dinámica</i>
				<i>Autoadaptación</i>
				<i>Colaboración</i>

Figura: Grado de cobertura de propiedades requeridas

Inconvenientes derivados de utilizar software intermediario

- Costes de desarrollo prohibitivamente altos
- Existe poco personal con experiencia en el desarrollo
- Pocos estándares totalmente satisfactorios para poder desarrollar sistemas y aplicaciones
- Las herramientas actuales no son lo suficientemente buenas
- Pone en peligro las prestaciones de *tiempo real* de un sistema
- Los productos software relacionados con el *software intermediario* no están aún demasiado maduros

Servicios Web

Servicios Web

Los Servicios Web son servicios implementados utilizando estándares como SOAP, WSDL y UDDI para ser accedidos a través de una red, preferiblemente Internet, y ejecutados de manera remota

Protocolo estándar

Service Oriented Architecture Protocol (SOAP)

Lenguaje de descripción de servicios

Web Service Description Language (WSDL)

Descubrimiento de servicios disponibles

Universal Description Discovery and Integration (UDDI)

SOA y Servicios Web

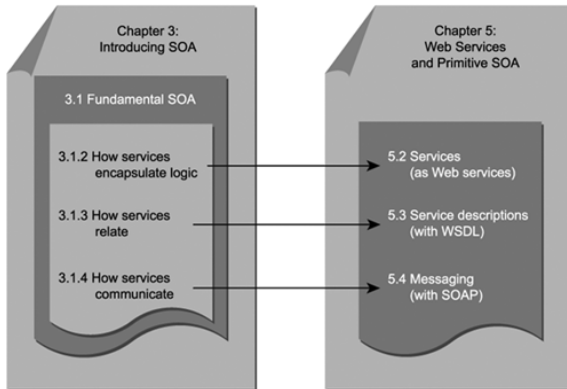


Figura: Relación estructural entre SOA y Servicios Web

Implementación de SW

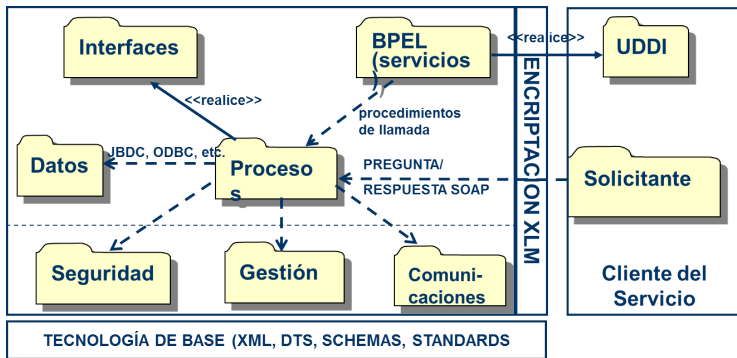


Figura: Implementación de una arquitectura para Servicios-Web

Marco de implementación de SW

Caracterización Servicios Web *de base*

- Ha de ser *abstracto*:
 - definido por organización de estándares (ISO, ...)
 - implementado en todas las plataformas
- Bloques constructivos, descripciones de servicios y de mensajes, basados en WSDL
- Componente de mensajería basado en SOAP y sus conceptos
- Registro de descripción de servicios, a veces se incluye descubrimiento de servicios, basado en UDDI
- Arquitectura software bien definida con mensajería, patrones y composición

Tecnologías

XML

El lenguaje de marcas XML se basa en la combinación de texto junto con información que describe ese texto. En XML se utilizan las etiquetas (tags) para describir bloques de texto. Fue diseñado para compartir fácilmente las estructuras de datos a través de la red.

SOAP

Simple Object Access Protocol es un protocolo estándar bajo el auspicio de la W3C que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML

Tecnologías – II

WSDL

Web Service Description Language es un lenguaje basado en XML utilizado para describir los Servicios Web. Este lenguaje define a los servicios como colecciones de puertos, donde cada puerto indica una función del servicio que está siendo descrito. Así, cualquier usuario de un servicio puede leer el WSDL y saber qué funciones se pueden llamar utilizando SOAP.

UDDI

Universal Description, Discovery and Integration es un registro basado en XML para listar servicios en Internet. Está diseñado para ser interrogado con mensajes SOAP y proveer acceso a los WSDL de los servicios que se encuentran listados.

Tecnologías – III

BPEL

Business Process Execution Language es un lenguaje ejecutable de procesamiento de negocios basado en XML. BPEL nació como la combinación de WSFL (IBM) y XLANG (Microsoft) para crear un estándar mundial de ejecución de procesos de negocio. Este lenguaje se utiliza para orquestar la ejecución de un proceso, llamando a los servicios que va necesitando.

Clasificación de los Servicios

Atendiendo a los roles que puede asumir un Servicio Web

- Proveedor de servicios
- Solicitante de servicio
- Intermediario
- Emisor inicial y receptor final

Proveedor de servicios

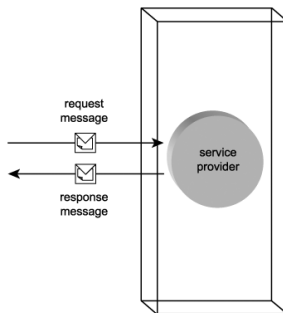


Figura: Un SW actuando como proveedor de servicio al recibir una petición

El concepto de proveedor de servicio es diferente del de agente del servicio

Solicitante de servicios

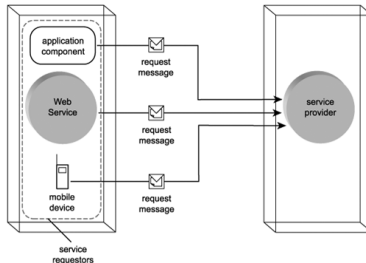


Figura: Un SW actuando como solicitante de servicio al enviar una petición

El servicio web busca y evalúa al proveedor más adecuado a través de las descripciones WSDL de los servicios

Intermediación de servicios

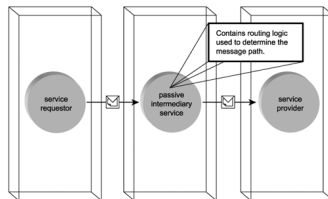


Figura: intermediario pasivo no altera contenidos del mensaje

El servicio web busca y evalúa al proveedor más adecuado a través de las descripciones WSDL de los servicios

- Comunicación no predecible, basada en caminos *punto-a-**
- SW actuando como intermediarios, enrutan y procesan mensajes en tránsito
- Intermediarios, actúan como proveedores y solicitantes

Intermediación de servicios – II

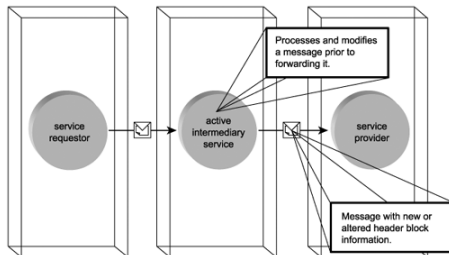


Figura: Un SW actuando como intermediario activo

La intermediación activa utiliza la información en el encabezado SOAP de mensajes:

- buscan bloques de datos en encabezados y los alteran
- realizan alguna acción concreta

Composición de servicios

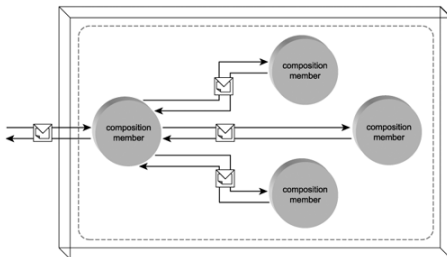


Figura: Composición de servicios de 4 miembros

- Los SW han de ser *fuertemente composicionales*
- Extensiones para SW: orquestación y coreografía
- Lenguajes WS-BPEL y WS-CDL para “*service assemblies*”
- Composición *transaccional*: posibilidad de “roll-back”

Arquitectura simple de aplicación basada en SW

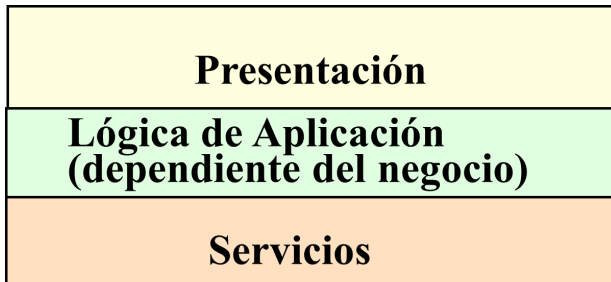


Figura: Aplicación de negocios

Modelos de servicios

Concepto

- Clasificación de los servicios atendiendo a la forma en que se utilizan
- Roles que los SW desempeñan en los *procesos de negocio* y en los *sistemas de información*
 - Modelo de servicio de negocio
 - Modelo de utilidades para aplicaciones
 - Modelo de controlador

Modelo de servicio de negocio

Suele corresponder con la *capa de actividades de negocio*

El servicio de negocio actúa como un controlador

- como bloques constructivos fundamentales para la representación de toda la lógica de negocio del sistema,
- para representar una entidad (de análisis) de la empresa o a una parte que pueda considerarse un conjunto de información con entidad propia dentro del sistema,
- para representar la lógica de proceso de negocio como miembros de una composición de servicios.

Modelo de utilidades para aplicaciones

Suele corresponder con la capa concreta denominada *servicios de aplicaciones*

- como servicios que hacen posible fomentar la *reutilización* en un SOA,
- como servicios intermediarios que desconocen la aplicación en la que se vayan a utilizar,
- como servicios que promueven la característica intrínseca de un SOA denominada *interoperabilidad*,
- como servicios con el mayor grado posible de autonomía.

Modelo de controlador

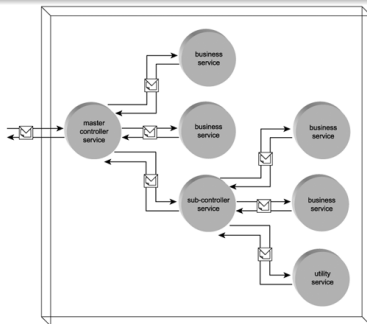


Figura: Composición de servicios con un controlador–maestro

- para apoyar e implementar la composicionalidad,
- para hacer uso de las oportunidades de reutilización de servicios más específicos,
- en apoyo de otros servicios, para apoyar su autonomía.

Descripciones de servicios con WSDL

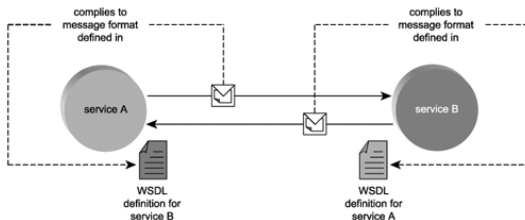
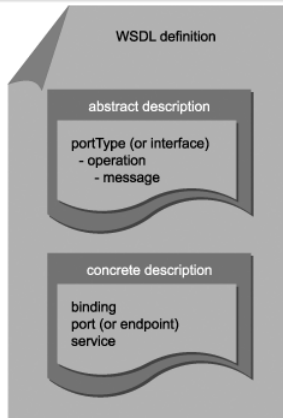


Figura: Acoplamiento débil de servicios utilizando WSDL

- Las descripciones de servicio son fundamentales para establecer una forma de comunicación débilmente acoplada
- Fundamentales para obtener una infraestructura de desarrollo basada en SW

Descripción de servicios con WSDL – II



- Descripción abstracta
- Descripción concreta

Descripción de servicios abstracta con WSDL

Establece las características de la interfaz del servicio sin ninguna referencia a la tecnología de implementación del mismo

Estructuración de la descripción abstracta

- *PortType* (“Interface” en WSDL 2.0): ordenación de mensajes en grupos de operaciones
- *Operation*: conjunto de mensajes de entrada/salida (≡ parámetros de las operaciones)
- *Message* y los patrones que gobiernan la secuencia de transmisión de mensajes

Descripción de servicios concreta con WSDL

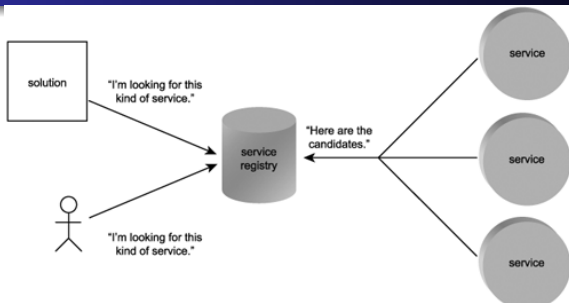
Objetivo

Conecta la definición de una interfaz abstracta con una tecnología real implementable. Suele conectarse a una tecnología específica de protocolo de transporte físico en red

Estructuración de la descripción concreta

- *Binding*: ligadura que establece las condiciones para que se puedan establecer las conexiones físicas del servicio
- *Port*: dirección física a través de la cual se puede acceder a un servicio utilizando un protocolo específico.
Renombrado como *endpoint* en WSDL 2.0
- *Service*: grupos de *extremos* que están relacionados para proporcionar un servicio en el otro extremo

Descubrimiento de servicios



- Para acceder a un SW, el único requisito es poder acceder a su descripción
- Repositorios de servicios:
 - Localizar las últimas versiones de SW disponibles
 - Descubrir nuevos SW de acuerdo con unos objetivos
 - UDDI está presente desde la primera generación de SW

Registro de SW con UDDI

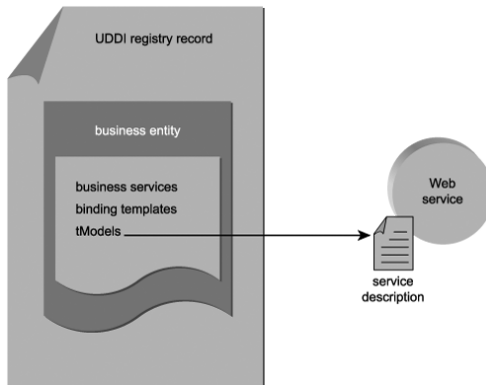


Figura: Estructura básica de un registro UDDI de entidad de negocio

Ejemplo de código para UDDI

```
1 POST /save_business HTTP/1.1
2 Host: www.XYZ.com
3 Content-Type: text/xml; charset="utf-8"
4 Content-Length: nnnn
5 SOAPAction: "save_business"
6 <?xml version="1.0" encoding="UTF-8" ?>
7 <Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope/">
8   <Body>
9     <save_business generic="2.0" xmlns="urn:uddi-org:api_v2">
10       <businessKey="">
11       </businessKey>
12       <name>
13       Enagas, S.L.
14       </name>
```

Ejemplo de código para UDDI –II

```
1      <description>
2          La empresa esta interesada en proporcionar lo mas
3              avanzado de ....
4      </description>
5      <identifierBag> ... </identifierBag>
6      ...
7      </save_business>
8  </Body>
</Envelope>
```

Descubrimiento de un SW con UDDI

Estructura básica de un registro UDDI

- *Business Services*: perfil básico del proveedor; descripción de los servicios ofrecidos
- *Binding Template*: guarda la información de enlace físico con el servicio. Cada servicio puede hacer referencia a 1 ó más *plantillas vinculantes*.
- *tModel*: proporciona punteros a direcciones de descripción del SW real

Sintaxis lenguaje UDDI

www.ws-standards.com/uddi.asp

Paso de mensajes con SOAP

- Los SW son unidades independientes de la lógica de proceso en los marcos de trabajo y en las aplicaciones
- Los mensajes SOAP ofrecen la posibilidad de agregar inteligencia y autonomía a los mensajes que se transmiten
- SOAP, diseñado para reemplazar a los protocolos RPC privativos, permiten serialización de llamadas entre componentes (distribuidos) mediante documentos XML

Mensajes de SOAP

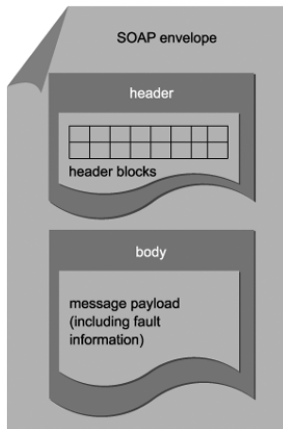


Figura: Estructura básica de un mensaje

Independencia de la lógica de proceso de las aplicaciones

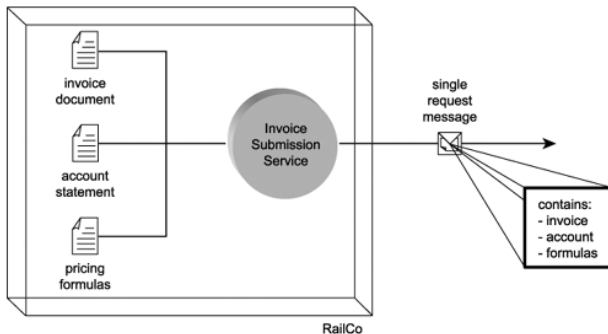


Figura: Mensaje SOAP y lógica de servicio

Nodos-SOAP

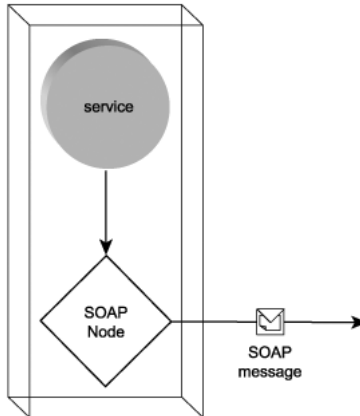


Figura: Nodos SOAP implicados en el proceso de un mensaje

Caminos de mensajes I

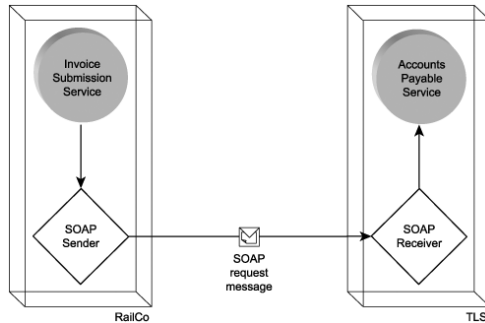


Figura: Camino de mensaje de 3 servicios Web

Caminos de mensajes II

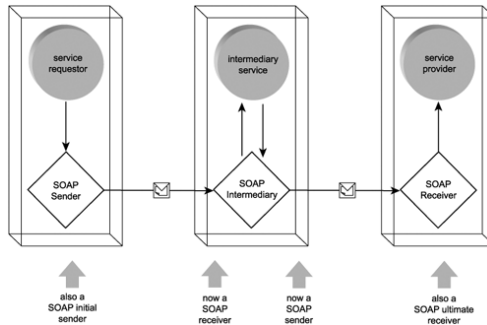


Figura: Camino de mensaje determinado dinámicamente

Patrones de intercambio de mensajes I

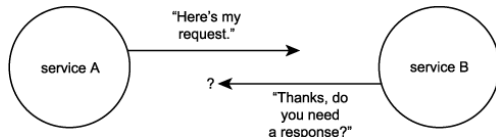


Figura: No todos los intercambios (MEPs) son solicitantes y receptores

Patrones de intercambio de mensajes II

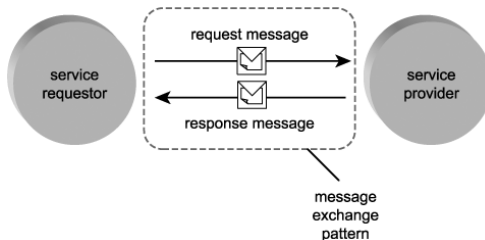


Figura: MEP petición-respuesta

Patrones de intercambio de mensajes III

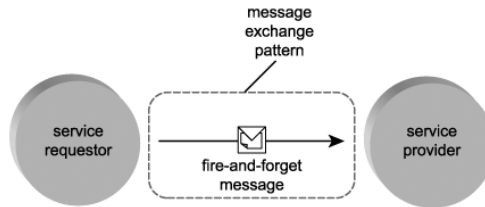


Figura: MEP disparar–olvidar

Patrones de intercambio de mensajes IV

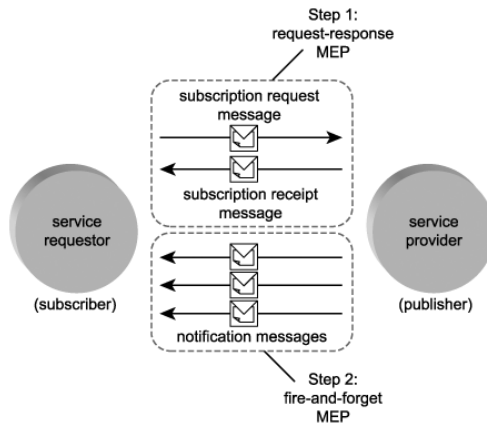
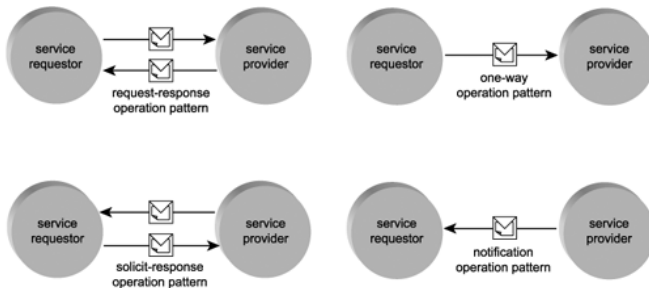


Figura: MEP publicar-suscribir

Patrones de intercambio respaldados por WSDL 1.1



Ontología general de un SOA

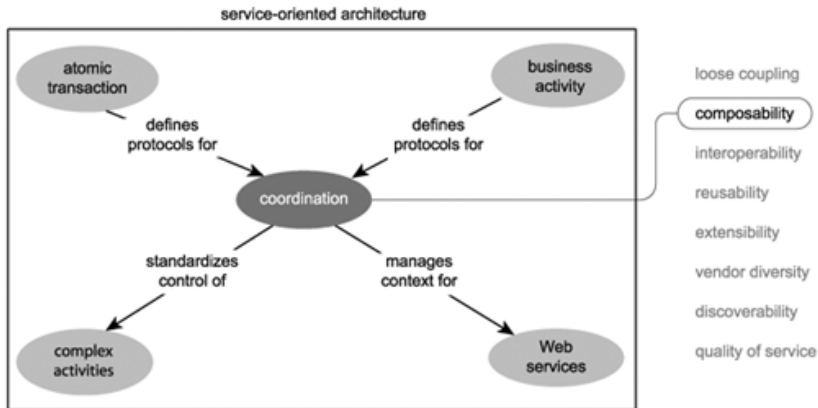


Figura: Relación con las otras partes de SOA

Actividad de servicio

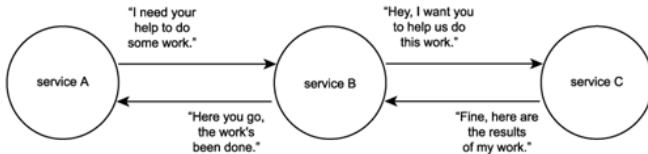


Figura: Colaboración entre múltiples servicios Web

Actividad de servicio II

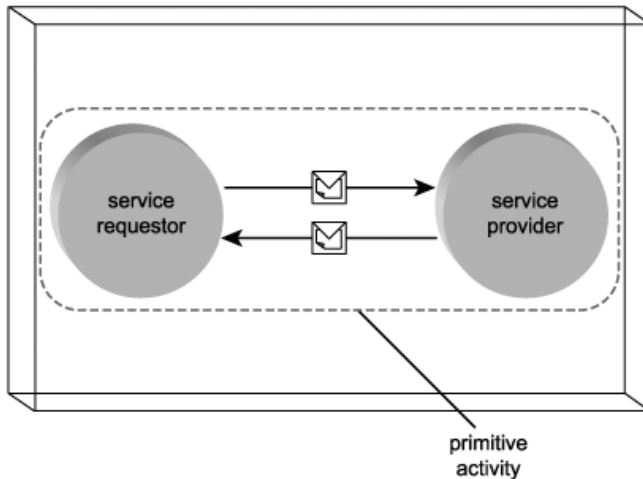


Figura: Actividad de servicio primitiva

Actividad de servicio III

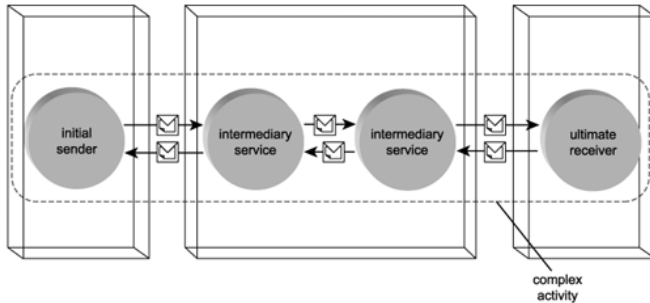


Figura: Actividad de servicio compleja

Coordinación

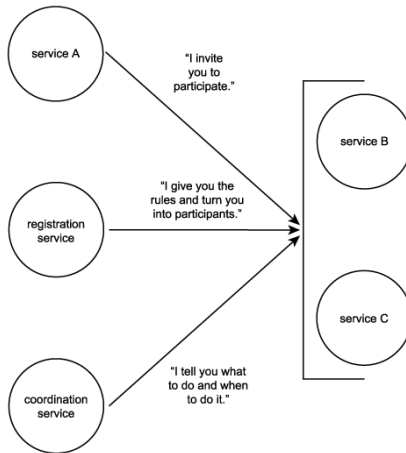


Figura: Estructura controlada de actividades

Coordinación II

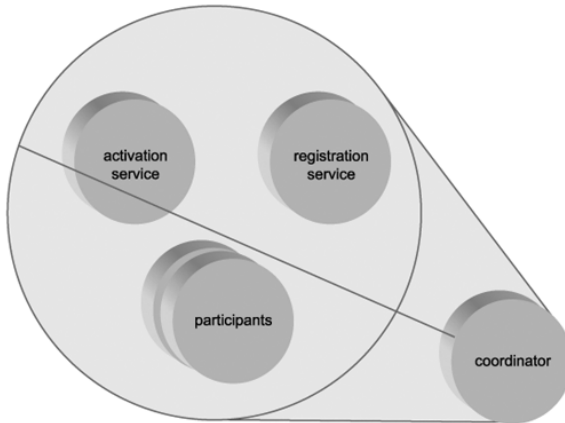


Figura: Composición de servicio de coordinación

Registro y activación

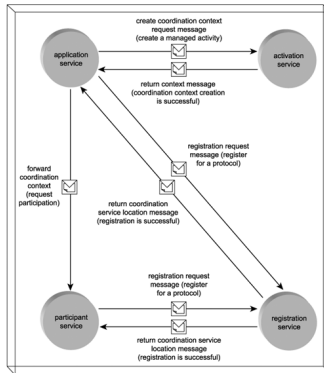


Figura: Registro de un SW de coordinación

Registro y activación II

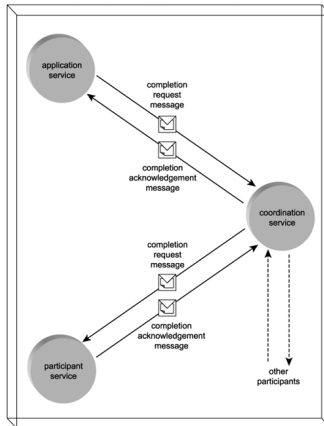
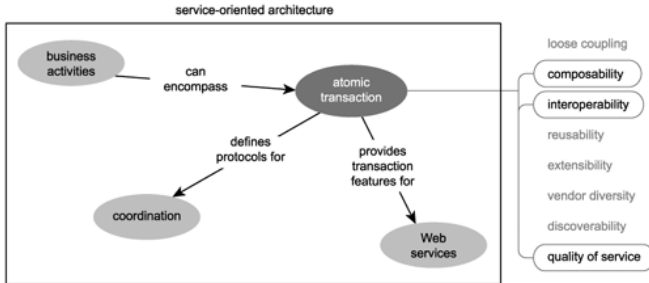
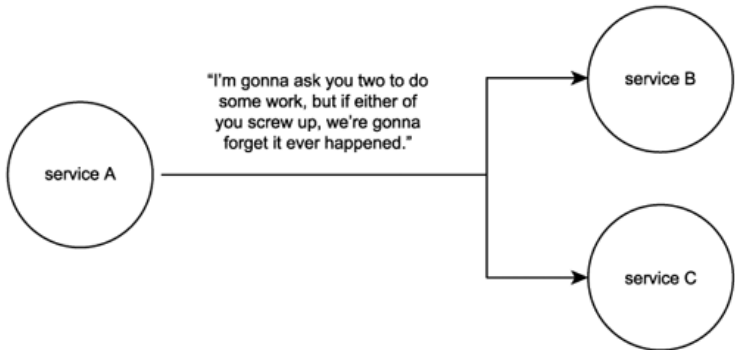


Figura: Completación de un SW de coordinación

Ontología de la transacción atómica



Transacciones atómicas



Coordinación de transacciones atómicas

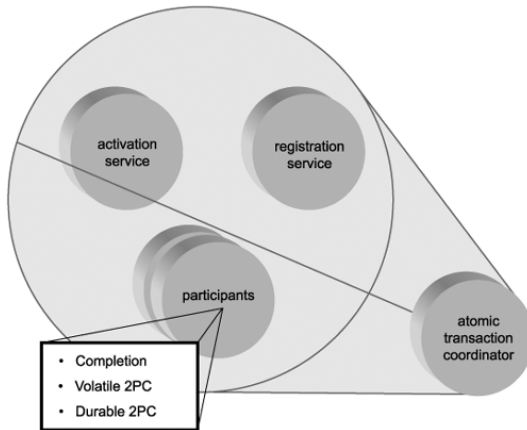


Figura: Servicio coordinador de transacciones atómicas

Protocolo de compromiso en transacciones

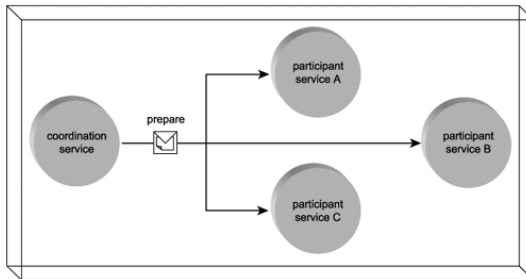


Figura: Coordinador pide a los participantes que se preparen para votar

Protocolo de compromiso en transacciones II

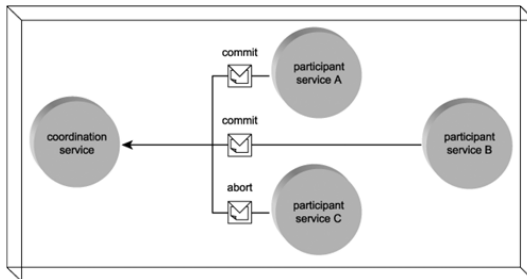


Figura: Los participantes de una transacción votando

Protocolo de compromiso en transacciones III

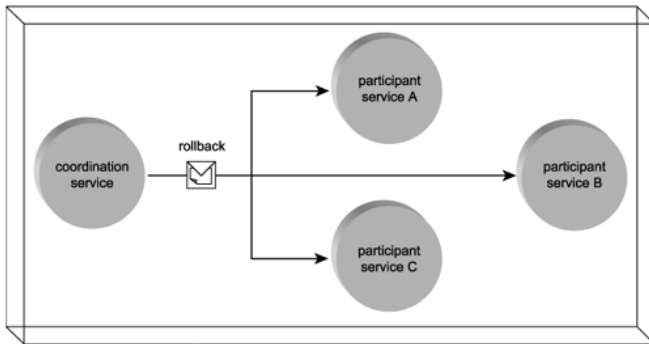
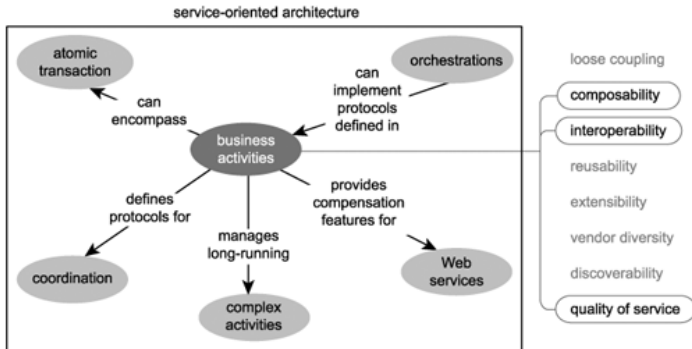


Figura: Coordinador abortando la transacción y notificando “roll-back”

Ontología de las *actividades de negocio*



- Regularn actividades complejas gracias a su autonomía y composicionalidad
- Limitan el tiempo de vida de los servicios a lo que sea necesario
- Mediante el proceso de compensación aumentan QoS de un SOA
- Propician interoperabilidad y facilitan integración de arquitecturas software

Actividades de negocio

Ideas fundamentales

- Se encargan de gerenciar las actividades de servicio complejas y de larga duración
- Los participantes en estas actividades están obligados a seguir reglas complejas dictadas por los *protocolos de una clase de negocio*
- Se diferencian de la gestión de transacciones en:
 - Utilizan procesos de compensación en lugar de “roll-back”
 - Diferentes métodos de tratamiento de fallas
- Carecen de estado
- Altamente composicionales y autónomas
- Complementan muy bien las responsabilidades de un SOA

Protocolos de Actividad de Negocio

Business Activity de WS-*

- Tipo de coordinación diseñado para la gestión del contexto entre actividades de negocio de WS
- Los participantes normalmente se basan en un *coordinador* para determinar que ya no tienen que realizar más procesamiento
- Los participantes se inscriben en un protocolo estándar
- El coordinador y los participantes transicionan a través de varios estados
- Los participantes pueden entrar en *estados de compensación*, donde llevan a cabo acciones relacionadas con el tratamiento de fallas y control de excepciones

Actividades de negocio vs transacciones

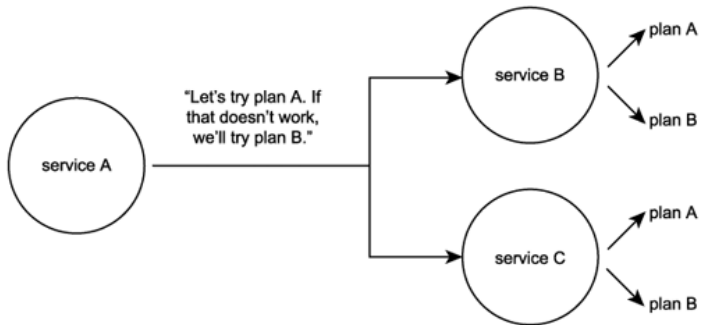


Figura: Control de la integridad de actividad de servicio

Actividades de negocio y transacciones atómicas

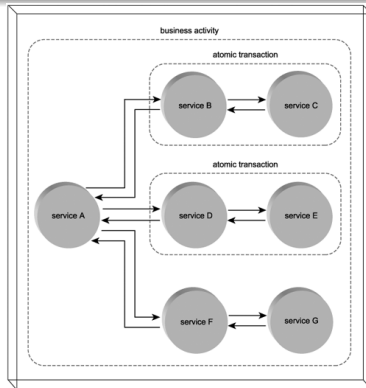


Figura: 2 transacciones atómicas dentro de una actividad de negocio

El uso de una actividad de negocio no excluye el uso de las transacciones atómicas en aplicaciones y marcos de trabajo

Introducción a la composición de SW



Figura: Colaboración entre SW

Composición de SW

“Proceso de agregar múltiples servicios en un único servicio para realizar funciones más complejas en una aplicación o modelo de negocio”.

Orquestación de SW

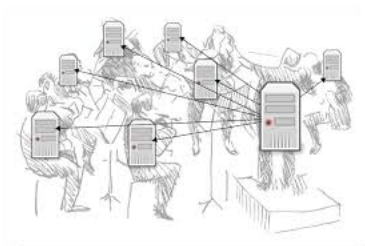


Figura: Orquestación de SW

Concepto

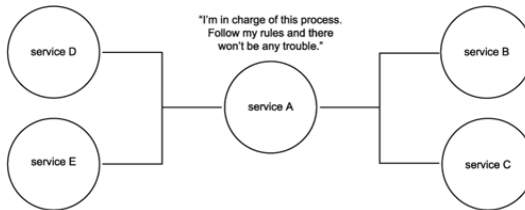
Se trata de una parte centralizada bien definida de la *lógica de flujo de trabajos* de un sistema de información de negocios que facilita la *interoperabilidad* entre 2 ó más aplicaciones diferentes.

Orquestación de SW

Objetivos

- Hacer posible la fusión de grandes procesos de negocio sin tener que reconstruir las aplicaciones
- Se consigue la colaboración de las aplicaciones mediante la introducción de una nueva *lógica de flujo de trabajos*
- Sirve para abstraer de lógica de flujo de trabajos de la solución
- La orquestación es más importante en entornos de desarrollo orientados a servicios
 - La lógica del proceso de negocio se consiga expresar mediante servicios

Orquestación como servicio



- Federación de empresas y la orientación a servicios
- El diseño de servicios fomenta la *interoperabilidad*
- Siendo un servicio más, se consigue extender un sistema sin afectar a la interoperabilidad de los componentes
- La lógica incluida en una orquestación permite normalizar la representación de toda una empresa como modelos

Web Services Business Process Execution Language (WS-BPEL)

Características de WS-BPEL/BPEL4WS/BPEL

- Extensión clave para poder obtener WS de segunda generación
- Utiliza conceptos y terminología de WS-* para el modelado de procesos de negocio
- Estándar actualmente vigente: WS-BPEL 2.0 (ver www.oasis-open.org)

Protocolos de negocio y definición de procesos

Características de una orquestación

- La lógica de flujo de trabajos de una orquestación se encuentra contenida dentro de una definición de proceso
- La lógica de flujo de trabajos: eventos, condiciones y reglas de negocio
- Se define un protocolo: cómo interactúan los participantes de la orquestación para realizar una tarea de negocio

Servicios de procesos y entre asociados

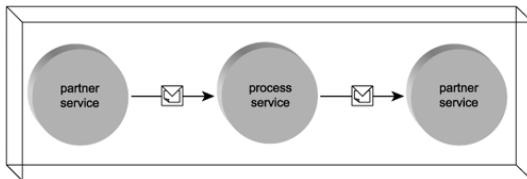


Figura: Un servicio de proceso inicialmente llamado por un servicio asociado y luego invocado por otro

Semántica de una orquestación

- Los procesos participantes se representan como servicios
- El *servicio de procesos* pueden asociarse y/o ser llamados externamente

Descripción del flujo de trabajos con WS-BPEL

Idea fundamental de la notación BPEL

- Descomponer la lógica de flujo de trabajos en una serie de acciones básicas pre-establecidas
- Identificar acciones fundamentales de flujo de trabajos que se puedan *ensamblar* utilizando la lógica de actividades estructuradas: secuencia, interruptor, repetición, secuencia, etc.
- Es importante conseguir que el orden de ejecución de las actividades no sea ambiguo y esté predefinido

Descripción del flujo de trabajos con WS-BPEL II

Orquestación y sincronización

- Un flujo de trabajos no termina hasta que todas las actividades que contiene han terminado su procesamiento
- Se consigue, por tanto, una lógica de orquestación/sincronización global basada en los flujos individuales
- El principio de modularidad entre flujos se consigue con la definición de *enlaces*: dependencias formales entre actividades de los flujos
 - Un actividad no puede completarse hasta que se cumplan los requisitos de sus enlaces salientes
 - Antes de comenzar cualquier actividad han de cumplirse los contenidos de los enlaces entrantes

Orquestación y SOA

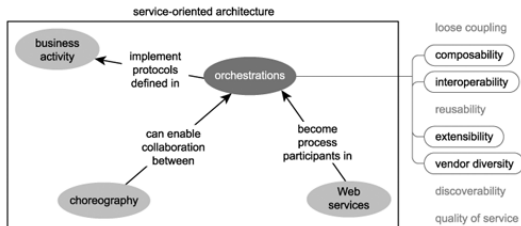


Figura: Orquestación relacionada con otras partes de un SOA

Principios de automatización de empresas

- La lógica de proceso de negocio está centralizada en una orquestación
- Las orquestaciones propician entornos de aplicaciones orientados a servicios extensibles y adaptativos

Beneficios de utilizar orquestación en SOA

Resumen

- Modificación centralizada de la lógica de flujo de trabajos
- Facilitar la fusión de procesos de negocio
- Definición de SOAs a gran escala respalda la evolución hacia una empresa federada
- La tecnología actual ayuda: middleware actual permite integrar motores de orquestación en entornos de aplicaciones orientadas a servicios

Coreografía de SW

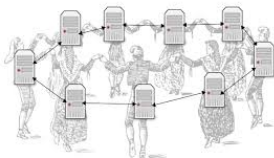


Figura: Orquestación de SW

Concepto

Se trata de una parte bien definida de la *lógica de negocio* que facilita la *interoperabilidad* de servicios cuando la colaboración se extiende más allá de los límites de una organización.

Se pueden ver como patrones de colaboración e interoperabilidad universal para la realización de tareas de negocio comunes a organizaciones.

Coreografía de SW – II

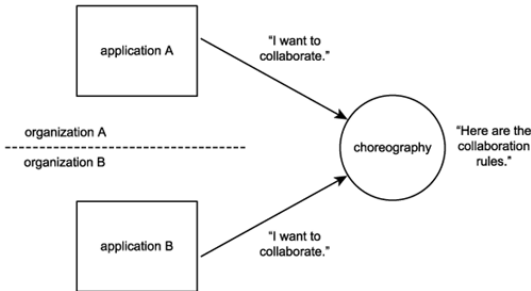


Figura: Una coreografía posibilita la colaboración entre participantes.

Motivación

Se necesita “coreografiar” múltiples servicios de diferentes organizaciones que necesitan trabajar juntos para lograr un objetivo común en el desempeño de una tarea de negocio.

Coreografía como servicio

Características de una coreografía

- Sirven para intercambio de *mensajes públicos*: colaboración entre SW que pertenecen a diferentes organizaciones
- A diferencia de las orquestaciones, ninguna organización controla la *lógica de colaboración*
- Se podrían utilizar también para establecer colaboración entre aplicaciones de una misma organización
- *Acciones* dentro de una coreografía se estructuran como secuencias de intercambios de mensajes entre SW

Coreografía como servicio – II

Canales

- Un intercambio de mensajes se define como una relación entre pares de roles, cada uno asumido por un SW
- Una acción de coreografía viene definida por un conjunto de pares de roles de SW
- Un canal define la representación del intercambio de mensajes entre los miembros de un par de roles
- La información sobre un canal se puede pasar en un mensaje para conseguir *descubrimiento de servicios*
- Las *unidades de trabajo* de una coreografía incluyen interacciones entre SW a las que imponen condiciones para conseguir completarse con éxito

Descripción de coreografías con WS-CDL

Web Services Choreography Description Language

- Lenguaje basado en XML que describe la colaboración entre pares mediante la definición de comportamientos comunes y observables de cada participante en una relación
- La especificación del lenguaje se debe a W3C (W3C Web Services Choreography Working Group, cerrada en julio de 2009), ver: <http://www.w3.org/TR/ws-cdl-10/>
- Actualmente existen solapamientos con el lenguaje WS-BPEL para orquestaciones
- La especificación del lenguaje fomenta el descubrimiento dinámico de servicios y propicia la colaboración entre muchos participantes a gran escala

Reusabilidad, composicionalidad y modularidad

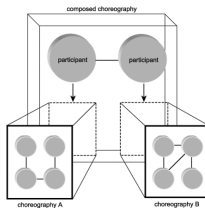
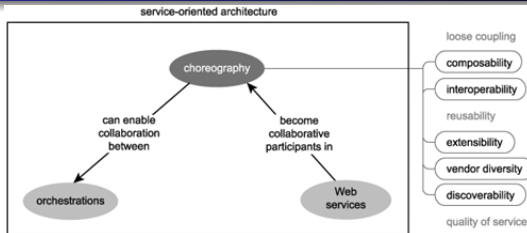


Figura: Una coreografía compuesta por 2 coreografías más pequeñas.

- Se diseñan de forma *reutilizable* para ser aplicadas a distintas tareas de negocio que comparten acciones
- Ensamblado de coreografías mediante un servicio de importación del marco de trabajo
- Se pueden estructurar en módulos que incluyen tareas para ser usados por una jerarquía de coreografías

Coreografía y SOA



Conceptos fundamentales

- Permiten ampliar o modificar dinámicamente los procesos de negocio participantes
- Ayudan a configurar SOA complejos que trascienden las fronteras de una organización
- Descubrimiento de SW y el diseño y desempeño ágil de tareas de negocio dentro de una organización

Orquestaciones vs. coreografías

Coincidencias y diferencias

- Una orquestación se podría entender como la aplicación específica de una coreografía a un determinado negocio
- Tanto las orquestaciones como las coreografías se utilizan para patrones complejos de intercambio de mensajes
- Una orquestación normalmente representa el flujo de trabajos de una organización:
 - La organización posee y controla la lógica de colaboración
- La lógica de control de una coreografía no suele ser propiedad de una sola organización:
 - el patrón de intercambio que se propone se utiliza para la colaboración entre SW de distintas organizaciones