Listado de Buenas Prácticas

Arquitectura de Servicios Web y Marco de Trabajo

Rodrigo Frez

21 Julio de 2009

Temario

- Introducción
- Listado de Buenas Prácticas
  - Prácticas para planear proyectos con Servicios Web
  - Prácticas relacionadas con la estandarización
  - Prácticas en la implementación

Repaso

- Arquitectura Servicios Web
  - Basado en estándares.
  - Muy buen soporte es crítico.
  - Se asume una muy pequeña infraestructura.
    - Sólo un pequeño número de estándares debe ser implementado.
  - Se enfoca en MENSAJES y DOCUMENTOS, no en componentes de Software.

Repaso

- Servicio Web:
- Es una arquitectura de computación distribuida
- Usa sus propias interfaces, protocolos y servicios de registro, para que distintas aplicaciones de distintas plataformas puedan usar sus procedimientos
- En constante evolución

Las aplicaciones de Servicios Web son encapsuladas, componentes Web bajamente acoplados que se pueden vincular dinámicamente entre sí

Introducción

- En general, debido a que la tecnología de Servicios Web reside sobre XML, las buenas prácticas para XML son válidas también.

Listado Prácticas para planear proyectos con Servicios Web

Saber cuando usar Servicios Web

- Un pequeño proyecto de bajo riesgo es una buena oportunidad para dar el primer paso dentro del mundo de los Servicios Web.
- En este ámbito será posible integrar esta tecnología de manera limitada y controlada.
- El ideal es no ir muy lejos esforzándose en integraciones que en poco tiempo quedarán obsoletas.
  - Por cambios en los estándares o por mejores alternativas de integración

Saber cuando usar Servicios Web

- Un conjunto de razones para considerar Servicios Web serían las siguientes:
  - La industria está adoptando y promoviendo el uso de Servicios Web. Mientras antes se incorpore al equipo de desarrollo, mejor será el entendimiento acerca de las nuevas arquitecturas de aplicaciones.
  - No es necesario crear una nueva arquitectura para usar Servicios Web. El diseño desacoplado de ellos permite agregar un número importante de servicios sencillos sin traer mayor impacto en el resto del sistema.

Saber cuando usar Servicios Web

- Si ya se está utilizando diseños orientados a servicios o modelos de negocio, los Servicios Web es el siguiente paso lógico. La incorporación de los paradigmas de orientación a servicios puede motivar técnicamente migrar a Servicios Web.
- Muchas herramientas de desarrollo incluyen soporte para creación y uso de Servicios Web, ocultando al desarrollador el detalle técnico de bajo nivel. Esto permite tener una pronunciada y favorable curva de aprendizaje.

Saber cómo usar Servicios Web

- Siempre es mejor limitar el ámbito de implementación de Servicios Web
  - de manera que esté a la par con los límites del propio conocimiento.

- Aunque el concepto fundamental tras los Servicios Web tiene mucho en común con el diseño basado en componentes, posee diferencias significativas.

- Agregar Servicios Web de manera incorrecta a una aplicación, lleva generalmente a desarrollar desde cero mucho antes de lo esperado.

Saber cómo usar Servicios Web

- Si las funcionalidades a desarrollar son críticas para la organización lo mejor es
  - Esperar hasta tener la absoluta certeza de la manera en que los Servicios Web deben ser integrados.
  - Idealmente limitar el uso de servicios a prototipos de bajo riesgo que lleven a entender de mejor manera donde los Servicios Web son un mayor aporte dentro la propia tecnología.

Saber cuando evitar Servicios Web

- Incorporar Servicios Web sólo tiene sentido cuando su inclusión provee valor. Si no son o serán parte importante de la organización en un futuro cercano es mejor evitarlos, principalmente teniendo en cuenta:
  - Evolución y diferencias en la implementación de las especificaciones de la arquitectura de Servicios Web
  - Es mejor esperar hasta que cada especificación posea una amplio respaldo de la industria
  - El peligro de utilizar ciegamente las herramientas que asisten al desarrollador en la creación de Servicios Web.
    - Aunque ahorran trabajo en el lenguaje de marcado (WSDL, SOAP), éstos no asisten en optimizar o mejorar el diseño de la aplicación.

Saber cuando evitar Servicios Web

- Las extensiones no estandarizadas que agregan soluciones de desarrollo propietarias crean dependencia en el proveedor.
  - Seguir el camino de estas soluciones puede comprometer las oportunidades futuras de interoperabilidad.

- Las aplicaciones autónomas no poseen como requisito el uso de Servicios Web.
  - El uso de éstos se toma en consideración principalmente cuando la interoperabilidad es el tema central.

Actualizarse de forma incremental

- Para realizar la transición a una arquitectura que considere el uso de Servicios Web, es mejor:
  - Comenzar con una solución que utilice conceptos de orientación a servicios.
  - Que no dependa de la tecnología de Servicios Web para su implementación.

- Con esta aproximación es posible revertir el diseño al modelo basado en componentes, sin que exista gran impacto en el diseño general de la aplicación.

- De esta manera también será posible migrar de forma definitiva y natural a una arquitectura que descanse tecnológicamente en los Servicios Web en el futuro.

Considerar el verdadero peso de los sistemas legados

- Siempre es mejor considerar la reutilización de la lógica de los sistemas legados, antes de reemplazarla por completo.

- Los Servicios Web pueden entregar una ventaja comparativa tanto en costo como en tiempo:
  - al entregar las herramientas para incorporar la lógica de los sistemas legados a una organización que necesita un alto grado de interoperabilidad

Considerar el verdadero peso de los sistemas legados

- Adicionalmente es importante tener en cuenta las limitaciones propias de incorporar un sistema legado que no fue diseñado para integraciones externas.

  - En los casos en que estas limitaciones estén claras, adaptar un Servicio Web para que entregue cierta lógica de un sistema legado tiene mucho sentido.

El costo y retorno de inversión

- Un buen Servicio Web requiere una gran cantidad de trabajo para asegurar que verdaderamente funciona y es confiable.
- Cada servicio desarrollado puede convertirse una pieza clave dentro de toda una infraestructura tecnológica.
- Aparte de exponer interoperablemente aplicaciones legadas y variados tipos de funcionalidades reutilizables dentro de un negocio, los Servicios Web pueden representar e incluso constituir procesos de negocios completos.
- Si se construyen Servicios Web, el costo del desarrollo puede ser significativamente menor si se utilizan las actuales herramientas de desarrollo que la organización posee, siempre que ésta soporte la creación de Servicios Web.

El costo y retorno de inversión

- La calidad de la interfaz que representa a un Servicio Web es muy importante tomando en cuenta las nuevas posibilidades de integración.

- Aunque los Servicios Web son catalogados como una tecnología muy poco acoplada, una vez que una serie de servicios se encuentran altamente integrados a una plataforma organizacional mayor es la posibilidad de encontrar dependencias sobre las interfaces.

- Cambiar una interfaz una vez que ya ha sido construida, puede ser una tarea costosa y sucia, especialmente en ambientes en que un servicio depende funcionalmente de otro.

El costo y retorno de inversión

- Sin embargo en la mayoría de los casos, ensamblar sistemas nuevos y sistemas legados a una arquitectura de Servicios Web requerirá una fracción del costo y el esfuerzo de un proyecto de integración tradicional.
  - Debido a esto existirán retornos concretos y medibles a la inversión que se deberán tomar en consideración. Lo ideal es hacerlo bien al primer intento, pero lograrlo tendrá sus costos.

- Aunque muchas organizaciones ya han invertido en proyectos orientados a arquitecturas XML, dar el paso a integrar tecnologías basadas en Servicios Web necesita de justificaciones adicionales:
  - es especialmente cierto cuando inversiones significativas ya se han dedicado a proyectos que dentro de una organización cumplen de buena forma su objetivo.

El costo y retorno de inversión

- La receta general para mantener visible el retorno a la inversión es evaluarla cada vez que nueva información se encuentre disponible.

- Es muy probable que los beneficios de una arquitectura que utilice Servicios Web sean mucho mayores que los originalmente previstos.
  - Esto debido que los beneficios de la migración son esencialmente tecnológicos y de implementación, y no dirigidos a los beneficios organizacionales de alto nivel.

- El retorno de la inversión en la implementación de Servicios Web puede en algunos casos ser mas fácil de justificar que otras tecnologías basadas en XML.

- Los beneficios tienden a ser más tangibles debido a que la interoperabilidad resultante presenta ahorros que son fácil e inmediatamente identificables.

Diseñar mirando el futuro

- Se puede tomar ventaja del potencial de interoperabilidad de los Servicios Web, aún si una integración no es demandada inmediatamente.
- Según esto es recomendable diseñar aplicaciones pensando en las futuras posibilidades de integración.
- Introducir conceptos integradores requerirá cambios en los estándares de diseño, la arquitectura de desarrollo y la manera de pensar del equipo desarrollador.
- De esta manera se podrá facilitar, por ejemplo, que clientes locales y futuros clientes remotos posean interfaces de Servicios Web estandarizadas y con convenciones de nombre.

Listado Prácticas relacionadas con la estandarización

Incorporar el estándar

- Como en cualquier proyecto de desarrollo en una organización, donde existen:
  - distintos grupos desarrolladores
  - construyendo distintas partes de un sistema

  estandarizar el diseño de cada una de ellas es importante por todos los motivos tradicionales (robustez, mantenimiento, etc.).

- En el mundo de los Servicios Web, el beneficio real viene en el establecimiento de una interfaz estandarizada.

- En una organización esto puede traducirse en sistemas estandarizados para navegar a través de la lógica de una aplicación, la arquitectura de la integración, los repositorios de datos corporativos y las partes de la infraestructura de la organización.

Incorporar el estándar

- Poseer ambientes no estandarizados siempre retrasará el progreso e introducirá nuevos riesgos.

- Cuando se desarrollan Servicios Web dentro de una organización, se establece una infraestructura en la cual desarrolladores, integradores, y tal vez usuarios remotos, podrán utilizar de forma común en años venideros.

- La sola inclusión de una plataforma común para el intercambio de datos no es suficiente para asegurar la calidad de un ambiente orientado a los Servicios Web.

Tomar en cuenta las convenciones de nombre

- Cuando se ensamblan las partes de una arquitectura integrada pueden aparecer múltiples componentes interdependientes, siendo cada uno de ellos necesario para la solución.

- Los ambientes de trabajo consisten en la mayoría de los casos en una mezcla de componentes de aplicación legados y contemporáneos, lo que por si sólo puede presentar algunas inconsistencias.

- En un proyecto es muy fácil etiquetar los componentes de una aplicación de forma arbitraria. Un nombre es algo que se agrega rápidamente, de manera de poder concentrarse en tareas funcionales más importantes.

- Los beneficios de la estandarización de nombres no son evidentes hasta bien avanzados los ciclos de un proyecto, cuando es necesario comenzar a ensamblar los componentes.

Tomar en cuenta las convenciones de nombre

- Los nombres utilizados para identificar componentes públicos de Servicios Web, actúan como puntos de referencia para otros componentes y Servicios Web.

- Cuando se modifica uno de estos nombres, se necesita cambiar todas las referencias a él.

- Como resultado, renombrar todos los componentes y Servicios Web se convierte en un proyecto en si mismo, durante el cual todo otro desarrollo es puesto en espera.

Tomar en cuenta las convenciones de nombre

- Utilizar una convención de nombres no sólo mejorará la eficiencia de la administración de una solución, hará de la migración y la implementación de los nuevos proyectos de integración, una tarea mucho más sencilla.

- Las convenciones de nombre reducen el riesgo de error humano y la posibilidad de que un pequeño cambio lleve a la solución a misteriosas caídas.

Separar el diseño de la implementación

- Diseñar un Servicio Web es una tarea aparte de la implementación.
- Un diseñador de interfaces de Servicios Web es responsable de asegurar que las interfaces externas de todos los Servicios Web son consistentes y representan claramente la función de negocio del Servicio Web.
- Este diseñador típicamente será el “dueño del documento WSDL”, al cual los desarrolladores proveerán de código para su implementación.
- También puede estar a cargo de los documentos SOAP para asegurar también un formato consistente de mensajes.

Separar el diseño de la implementación

- Responsabilidades típicas de un diseñador de interfaces de Servicios Web:
  - Documento WSDL
  - Documento de mensajería SOAP
  - Claridad de la interfaz
  - Extensión de la interfaz
  - Estandarización y convenciones de nombre de la interfaz
- Los típicos requisitos para este rol son poseer conocimientos en diseño de componentes, habilidades y conocimientos de SOAP y WSDL, capacidad de análisis del negocio y un buen entendimiento del ámbito de negocio de la organización.

Utilizar algún mecanismo de clasificación

- Cada Servicio Web es único, pero muchos de ellos terminan realizando funciones similares y exhibiendo características comunes, lo que permite pensar en la clasificación de ellos.

- A continuación se expresan algunos beneficios del uso de clasificaciones:

  - Es posible aplicar estándares de diseño específico a las distintas clasificaciones de Servicios Web.
  - El tipo de clasificación comunica instantáneamente el rol del Servicio Web en la arquitectura.
  - Las clasificaciones pueden alinearse con las políticas y estándares de seguridad de la organización.

Listado Prácticas en la implementación

Utilizar un registro privado de Servicios Web

- Una vez que los Servicios Web se establezcan como una entidad común dentro de la organización, naturalmente estos comenzarán a evolucionar:
  - requiriendo actualizaciones a las interfaces o la irrupción de nuevas generaciones de Servicios Web.

- Muy pronto se hará difícil mantener registro de todas las interfaces de cada servicio:
  - especialmente tomando en cuenta que muchos de ellos siempre se encontrarán en estado de transición (debido a las migraciones desde los sistemas legados).

Utilizar un registro privado de Servicios Web

- Un registro privado puede almacenar las descripciones de todos los Servicios Web que posea la organización.

- Actúa como un repositorio central para las actuales interfaces de servicios, la cual puede acceder, previa autentificación, cualquier interesado en conocer acerca del marco de Servicios Web de la organización.

- Algunos de los beneficios inmediatos son:
  - Acceso eficiente a las interfaces de cada servicio.
  - Se preserva la integridad de todas las interfaces de servicio publicadas en el registro.
  - Se motiva y promueve el descubrimiento de Servicios Web genéricos y reutilizables.

Utilizar un registro privado de Servicios Web

- El registro de servicio debiera ser un proceso obligatorio dentro del marco de Servicios Web en una organización.

- Los usuarios no deben perder confianza en el registro, por lo que deben tener las garantías de que las interfaces publicadas corresponden a las últimas versiones disponibles.

- Otro punto importante para hacer de un registro una parte importante y fundamental del marco de Servicios Web, es asignar un rol que sea responsable de mantener el registro de Servicios Web.

- En resumen es responsable de la administración del repositorio del registro de Servicios Web.

Considerar la administración

- Una tarea subestimada en los proyectos de sistemas orientados al uso de Servicios Web es la administración necesaria para mantener la solución funcionando en el tiempo.
- Al aumentar la interoperabilidad tiene como riesgo la más alta dependencia entre ambientes de aplicación.
- Con un mayor nivel de integración mayor es la responsabilidad de mantener los Servicios Web funcionando de forma correcta, independiente de los parámetros que reciba o devuelva.
- Los altos volúmenes de uso, las condiciones de error y otras variables de ambiente son situaciones que deben ser consideradas y anticipadas.

Considerar la administración

- En este caso también es recomendable crear un rol de Administrador de Servicios, el cual es responsable de la mantención y monitoreo de las variables de ambiente.
- En ambientes donde un gran número de Servicios Web son utilizados debe existir una administración de sistema para garantizar la ejecución confiable dentro del ambiente que acoge a los Servicios Web.
- Este rol es especialmente relevante en organizaciones que ofrecen Servicios Web que pueden ser accesados externamente.
- Para manejar efectivamente volúmenes de uso impredecibles, este rol debe ser capaz de responder rápidamente cuando el rendimiento comienza a caer.

Considerar la administración

- El rol de Administrador de Servicios debiera ser responsable entre otros de:
  - Solucionar la necesidad de contar con herramientas y máquinas que permitan realizar la administración, proponiendo y evaluando productos.
  - Monitorear el uso de Servicios Web en forma individual.
- Analizar estadísticas de uso, para identificar patrones.
  - Prever el impacto del uso de un Servicio Web sobre aplicaciones o componentes legados
  - Identificar y mantener registro de las dependencias entre los Servicios Web implementados.
  - Manejar el control de versiones sobre los Servicios Web.
  - Estar involucrado con el control de versiones de aplicaciones legadas representadas por Servicios Web.