

Visión unificada de datos en web services

Por Agustín Conseglieri y Ángel García Crespo

1. Introducción

MIENTRAS QUE LAS SIGLAS EAI (*Enterprise application integration*), o el término middleware hacen referencia a una clase de aplicaciones de software que tienen como finalidad la interconexión de sistemas heterogéneos, comunicándose mediante un protocolo común al que suelen acceder mediante adaptadores, y con el objetivo final de ofrecer un soporte a procesos de negocio que afectan a más de uno de los sistemas de información de la compañía, los web services utilizan un esquema de peticiones http y respuestas xml para comunicar sistemas entre sí.



Agustín Conseglieri, ingeniero en informática por la Universidad Carlos III de Madrid. Especialista en diseño y desarrollo de sistemas de información, arquitecturas EAI y web services.

En los últimos años ha trabajado como técnico de sistemas en el ISP Madritel y como arquitecto de sistemas en Wanadoo España. En la actualidad realiza su tesis doctoral analizando la implantación de sistemas expertos en entornos empresariales, e imparte clases de estructura de datos en la universidad Francisco de Vitoria. Además colabora con programas de formación a empresas.

El sistema proveedor ofrece una serie de funcionalidades o servicios que pueden ser invocados mediante http como si se hiciera desde un navegador, y la página xml que es devuelta como respuesta no está pensada para ser presentada en pantalla, sino para que sea parseada y procesada como resultado del servicio que se ha solicitado. En concreto los middleware,



Ángel García Crespo, doctor ingeniero industrial, Universidad Politécnica de Madrid con la calificación "cum laude" por unanimidad. Premio del instituto J. A. Artigas. Executive MBA por el Instituto de Empresa. Profesor titular de la Universidad Carlos III de Madrid, profesor de la asignatura 'Planificación estratégica de sistemas de información'. Subdirector de Organización Docente de la Escuela Politécnica Superior, director del grupo de integración de sistemas avanzados. Definición de estrategias, gestión, coordinación y control de varios proyectos europeos, consultoría y desarrollo a empresas. Secretario del Instituto de Desarrollo Tecnológico y Promoción de la Innovación. Gestión, administración, promoción y consolidación del mismo.

han surgido como forma de acceso para otras aplicaciones a los entornos host, especialmente durante la época dorada del modelo cliente/servidor. Posteriormente se ha utilizado el mismo término para designar a los modelos de objetos distribuidos como Corba (*common object request broker architecture*) o RMI (*remote method invocation*), haciendo referencia en este contexto a una tecnología en la que la comunicación está embebida realmente en la lógica de la aplicación, al establecerse relaciones entre objetos que forman parte de distintas aplicaciones.

2. Estado del arte

Los EAI actuales contemplan la comunicación de aplicaciones pero de forma totalmente desacoplada, al menos en teoría, de la lógica interna de la aplicación, y la granularidad de la comunicación es sobre funcionalidades que el sistema en cuestión puede realizar de forma completa.

Entre las principales ventajas de los web services, destaca que aprovechan la tecnología internet existente, por lo que es fácil comunicar aplicaciones ubicadas en servidores remotos, asumiendo un canal seguro mediante la utilización de https (*hypertext transfer protocol secure*). Ofrecen un interfaz conceptualmente coherente para distintos canales, como extranets de ventas, intranets, portal de clientes, etc. Por otro lado, es bastante seguro para el servidor ofrecer el acceso a sus servicios mediante peticiones http, ya que aísla relativamente bien la lógica interna de la aplicación de la visión de la parte llamante, si lo comparamos por ejemplo con la posibilidad de invocar un objeto de su aplicación por otro objeto de la aplicación llamante, en el caso de utilizar Corba o DCOM (*distributed component object model*). Como inconvenientes más destacados nos encontramos con que no es fácil asegurar la integridad transaccional con este esquema, lo que obliga a buscar una atomicidad en métodos de granularidad elevada, también otros problemas (como la velocidad de los parsers, el intercambio de esquemas o plantillas de definición de los mensajes, la unificación en la descripción de los servicios ofrecidos mediante lenguajes estándar, etc.) están siendo solucionados de forma más o menos exitosa en las soluciones comerciales actuales.

3. Middleware como complemento de los web services

La idea fundamental cuando se utilizan web services es invocar llamadas a métodos o procedimientos remotos que utilizan modelos de información conceptualmente disjuntos de los que se manejan por los servicios de otro

componente. Sin embargo, en la práctica ocurre que los servicios web, ofrecen la posibilidad de acceder a sistemas de información que, aunque tienen su modelo de datos propietario, independientemente de que se trate de una customización o no de la que traiga por defecto la aplicación, no deja de formar parte desde el punto de vista conceptual del mapa de datos de toda la empresa, incluso aunque éste no se encuentre definido formalmente, con lo cual es muy interesante la posibilidad de contar con esta visión añadiendo un middleware.

«El web service en sí mismo es bastante restrictivo en el sentido de mostrar cómo maneja la información»

El web service en sí mismo es bastante restrictivo en el sentido de mostrar cómo maneja la información y, en cierta medida, eso es lo que posibilita el desacoplamiento de los sistemas, identificando sólo una serie de parámetros de entrada obligatorios y proporcionando cierta funcionalidad que es totalmente conocida a priori. Esta visión unificada e integradora de los datos en el middleware posibilita ir un poco más allá y, estableciendo un símil, tener la visión del modelo de procesos de la compañía, en la parte soportada por los sistemas de información, incluyendo la relación con los sistemas e incluso los datos que consultan o modifican. De esta forma, puede pensarse en una distribución de sistemas de información que ofrecen un *API (application programming interface)* o interfaz de servicios web, que se integran mediante un middleware, constituyendo una interfaz común del más alto nivel. El middleware proporcionaría en esta arquitectura, integridad, control transaccional, gestión de errores y visión de los

Más información

—*COM (component object model)*, de Microsoft.

<http://www.microsoft.com/com/>

—*Corba (common object request broker architecture)*, es un estándar definido por el *OMG (Object Management Group)*.

http://www.omg.org/technology/documents/corba_spec_catalog.htm

—*RMI (remote method invocation)*, desarrollado por Sun Microsystems.

<http://www.sun.com/>

—*SOAP (simple object access protocol)*, definido por el *W3 Consortium*.

<http://www.w3.org/>

procesos, mientras que los web services de los sistemas facilitan el desacoplamiento tanto desde el punto de vista funcional como técnico.

Existe una evolución desde el enfoque de la integración de sistemas de información que supone como primer nivel la integración de plataformas desde el punto de vista del hardware, la posterior integración a nivel de datos considerando en este estadio la mera comunicación de los mismos de forma homogénea. La integración de las aplicaciones consigue ya no sólo la compartición de la información, sino el funcionamiento de aplicaciones transversales que puedan dar soporte a un proceso de negocio de forma completa. Si se ofrece soporte de los *SI (sistemas de información)* a todos los procesos, se puede hablar de un entorno de procesos integrados que puede evolucionar hasta lo que se denomina “total business integration” cuando se puede tener una visión de los procesos que realiza la compañía de forma homogénea y en un único punto de los sistemas de información. Si se añade la componente de tiempo real, se podría mo-

nitórizarse la actividad de la compañía y se habría llegado al estadio más evolucionado, denominado “business activity monitoring”.

La introducción de un middleware no sólo permite tener un *API* unificado de los sistemas de información, sino que además ofrece una visión global del modelo de datos de toda la información que maneja la compañía, como se ha expuesto. En algunos casos puede que antes de la introducción de un middleware no se tenga esta visión, conceptualmente hablando, y así ocurre en arquitecturas existentes en las que un proyecto de implantación de un middleware se utiliza muchas veces o genera una revisión o análisis de todo el modelo de datos de la empresa detectándose inconsistencias, redundancias o carencias que hasta el momento parecían ocultas. Es frecuente, especialmente en proyectos de hace algunos años, que los sistemas de información se hayan desarrollado ad hoc, como islas o setas, para resolver problemáticas concretas y en relación a partes de algún proceso de negocio, generalmente bastante crítico. Ocurre entonces que se introduce un middleware para conectar estas islas, pero utilizándolo para crear interfaces punto a punto entre cada una, mientras que lo deseable es reflejar el proceso íntegramente y tener una visión completa del mismo en toda la parte soportada por los sistemas de información. De esta forma el middleware puede mantener y ayudar en la monitorización del proceso, e incluso garantizar el cumplimiento de determinado *SLA (service-level agreements)*, generando alarmas o activando mecanismos de contingencia cuando proceda. Es interesante, incluso la posibilidad de introducir en este punto la lógica predictiva para detectar situaciones anómalas que puedan tener cierta probabilidad de que, si no se actúa en los sistemas, se llegue a situaciones de fallo; por

ejemplo un aumento en la latencia con un determinado sistema, puede indicar que ese sistema esté próximo a la saturación, o si el proceso de altas está cursando más de las habituales, se podría llegar a colapsar el proceso, siendo conveniente en tales casos activar mecanismos de contingencia antes de llegar a una situación de error y de más costosa resolución.

«La integración de las aplicaciones consigue ya no sólo la compartición de la información, sino el funcionamiento de aplicaciones transversales que puedan dar soporte a un proceso de negocio de forma completa»

Por otro lado, la lógica predictiva en el middleware puede sobrecargar el funcionamiento del mismo, lo que puede hacer desaconsejable la utilización de dicha funcionalidad en algunos entornos. Sin embargo, sí que parece imprescindible introducir reglas sobre cómo se debe actuar cuando una parte del proceso no puede continuarse porque, por ejemplo, un sistema no esté disponible, siendo lo más frecuente en estos casos el encolamiento y posterior lógica de reintentos.

Existe un trabajo realizado para integrar web services mediante un middleware con el objetivo de ofrecer una catálogo virtual de diversas bibliotecas que ofrecen información de sus catálogos locales mediante web services y que fue presentado en el *Congreso iberoamericano de telemática, CI-TA'2002*, que aplica esta idea pero a un ámbito más específico, aunque bien puede observarse que se obtiene un modelo de datos global, independientemente de los modelos de datos de cada una de las bi-

bliotecas. En este sentido es un avance permitir la generalización de la idea citada a cualquier sistema de información de cualquier entorno.

4. Conclusiones

La tendencia actual de desarrollo mediante web services en el ámbito de los sistemas de información presenta una característica fundamental que es el desacoplamiento de los datos de cada uno de los sistemas, llegando a ocultar en cierta medida la visión del modelo de datos global. Una aproximación para suplir esta carencia es la introducción de un middleware como elemento integrador de los web services que ofrecen cada uno de los sistemas, este componente podrá implementar los procesos de negocio mediante la combinación lineal de los distintos web services y además ofrecer una visión del modelo de datos global.

Bibliografía

- Álvarez, G.** "Seguridad en servicios web xml". En: *Boletín del criptonomicón*, 2003, febrero, v. 5, pp. 2-10.
<http://www.iec.csic.es/criptonomicon/susurros/susurros37.html>
- Charleswoth, I.; Jones, T.** "The web services and EAI report". En: *EAI journal*, 2003, marzo, v. 3, pp. 11-18.
- Dang Tran, F.; Gerodolle, A.** "A flexible Java middleware for building large-scale virtual environments on the internet". En: "Work in progress" de *IFIP/ACM International conference on distributed systems platforms and open distributed processing (Middleware2000)*, 2000, abril, pp. 4-8.
- Scott, S.** "Building xml web services for the Microsoft.NET platform". Microsoft Press, 2002.
- Zagalo, H. T.; Martins, J. A.; Pinto, J. S.; Nascimento, R.** "Um catálogo bibliográfico virtual e o seu acesso através de middleware baseado em web services". En: *Actas de II Congreso iberoamericano de telemática (CI-TA'2002)*, 2002.
- Agustín Conseglieri**, Universidad Francisco de Vitoria.
aconseglieri@bpe.es
- Ángel García Crespo**, Universidad Carlos III de Madrid.
acrespo@ia.uc3m.es