

INTEGRACIÓN DE SISTEMAS MEDIANTE MESSAGE-ORIENTED MIDDLEWARE (MOM)

Una aplicación práctica en el Nuevo Área Terminal del Aeropuerto de Madrid-Barajas

M. MUÑOZ

En este artículo se plantea la necesidad de integrar los diferentes sistemas informáticos de una empresa para poder ofrecer una visión corporativa y única. Para ello se presenta un tipo de software, denominado Message-Oriented Middleware (MOM), que puede ser empleado para realizar dicha integración. A continuación se realiza una comparación con otra posibilidad, denominada RPCs (Remote Procedure Calls), que constituye un procedimiento alternativo para integrar sistemas informáticos. Después se realizan algunas consideraciones sobre los principales aspectos funcionales que deben ser tenidos en cuenta cuando se prepara una integración de sistemas y, finalmente, se hace un breve resumen de un proyecto real: la integración del Sistema de Telefonía del Nuevo Área Terminal del Aeropuerto de Madrid-Barajas con el resto de los sistemas del aeropuerto.

This article presents the need for the integration of the various computer systems of a company to provide a single corporate vision. For this purpose, a type of software known as Message-Oriented Middleware (MOM), is presented. A comparison is then established with an alternative possibility for the integration of computer systems, known as RPCs (Remote Procedure Calls). Some considerations are then detailed on the main functional aspects to be taken into account when preparing a system integration, and, finally, a brief summary of a real project (the integration of the Telephone System of the New Terminal Area of the Madrid-Barajas Airport) is presented.

INTEGRACIÓN DE SISTEMAS INFORMÁTICOS MEDIANTE MOM

La rápida evolución de la Tecnología de Información ha producido en las empresas una enorme variedad de aplicaciones informáticas que deben coexistir. Se trata de sistemas desarrollados por diversos proveedores con tecnologías muy heterogéneas que, a pesar de ello, deben ser integradas. Además de la complejidad de diálogo derivada de la distinta naturaleza tecnológica y funcional de las aplicaciones, suele existir un problema adicional no menos importante: los sistemas se explotan en entornos diferentes entre los que, con frecuencia, tampoco

existe una fácil comunicación.

Se llama middleware a un tipo de software que, precisamente, está destinado a disminuir el problema de interacción entre sistemas. En realidad se llama middleware a cualquier software destinado a facilitar la conectividad mediante una serie de servicios, que permiten que varios procesos que se ejecutan en una o más máquinas puedan interactuar entre ellos a través de la red.

Existen varias clases de middleware. Según el tipo de conectividad en que se especializan, podemos hablar de Servicios de Presentación, cuando el objetivo del software consiste en facilitar la gestión de formularios, gráficos, impresiones o hiperenlaces; de Servicios de Control, cuando la misión del middleware es la gestión de transacciones, de recursos o la planificación de atención de peticiones; de Servicios de Información, cuando se trata de gestionar ficheros, bases de datos o repositorios; y de Servicios de Comunicación, cuando se facilita el envío de mensajes puerto-a-puerto, llamadas a procedimientos remotos, intercambio electrónico de datos o gestión de colas de mensajes.

Precisamente dentro de este último grupo de middleware se encuentra lo que finalmente se ha convenido en llamar Message-Oriented Middleware o, de forma más resumida, simplemente MOM. Como su propio nombre indica, este tipo de Servicios de Comunicación se basan en un paradigma realmente sencillo:

El MOM soporta llamadas asíncronas entre aplicaciones informáticas, constituyéndose en un bus lógico (no físico, ya que sólo es software) que permite la emisión y recepción de mensajes controlados por tópicos.

De esta forma se logra una fácil integración entre diferentes sistemas, que simplemente se relacionan emitiendo mensajes a través del bus, despreocupándose de todos los problemas inherentes a la comunicación, ya que será el MOM quién se responsabilice de hacer llegar el mensaje a todos los sistemas que estén suscritos al correspondiente tópico.

Los paquetes de MOM gestionan una cola de mensajes entre diferentes procesos que deben interactuar entre sí, de esta forma, si el proceso destinatario del mensaje está ocupado o no activado, el mensaje será almacenado en un dispositivo temporal hasta que pueda ser procesado adecuadamente. (figura 1).

Aunque el MOM es normalmente asíncrono y puerto-a-puerto, si se considera necesario y la implementación lo soporta, puede gestionar también mensajes síncronos o, al menos, funciones que permiten que la aplicación emisora del mensaje reciba las correspondientes confirmaciones de recepción por el resto de sistemas que estén suscritos al tópico con que se emitió el mensaje.

En resumen, todo muy parecido al correo electrónico convencional, con sólo dos diferencias: (1) los mensajes son intercambiado por sistemas, en vez

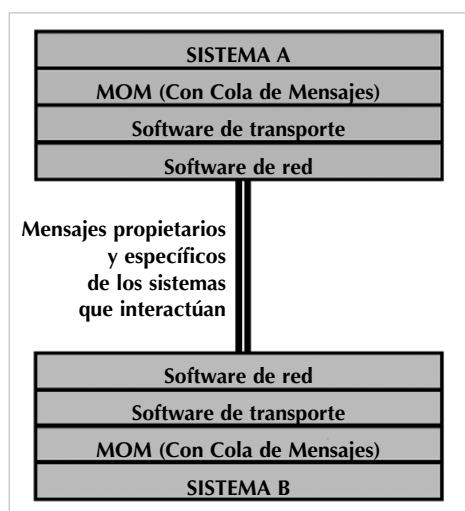


Figura 1. Integración de Sistemas mediante Message-Oriented Middleware.

de por personas, y (2) los mensajes pueden ir formateados de acuerdo a criterios preestablecidos para que los sistemas que los intercambian sean capaces de leerlos y procesarlos.

UN PROCEDIMIENTO ALTERNATIVO MEDIANTE EL USO DE RPCs

Antes de continuar hablando del MOM vamos a realizar una comparación con el procedimiento síncrono por excelencia para intercomunicar aplicaciones: se trata de las Llamadas a Procedimientos Remotos (o RPCs, por su versión inglesa "Remote Procedure Calls").

De forma resumida, el esquema de funcionamiento de una RPC es el siguiente: cuando un desarrollador implementa una aplicación y desea que aplicaciones de terceros puedan interactuar con la suya, desarrolla también unas funciones RPCs que pueden ser invocadas por cualquier otro sistema. Estas funciones, aparentemente, pueden ejecutarse como cualquier otra función propia, de forma que las llamadas son síncronas. Pero la situación es diferente, ya que cuando se realiza una llamada a una función propia, esa función, normalmente, reside en el mismo entorno de trabajo, mientras que al llamar a una función en otro entorno diferente, la agilidad en la respuesta no está garantizada.

Por lo tanto, cuando una aplicación invoca a una función RPC, se produce lo que podríamos describir como "llamar y esperar" y este protocolo implica que la aplicación que invoca a la función RPC quedará bloqueada hasta que la otra aplicación responda. Resulta evidente que utilizar RPCs o MOM para establecer una comunicación entre dos siste-

mas es completamente diferente, tan diferente como es emplear procedimientos síncronos o asíncronos, respectivamente. (figura 2).

Si se compara RPC con MOM, como no podía ser de otra forma, encontramos ventajas e inconvenientes en cada caso:

A favor de MOM debemos destacar la posibilidad de que el sistema "llamante" pueda realizar otras operaciones sin quedar en espera de una respuesta. También se debe considerar que en este caso puede haber varias respuestas para una única llamada o, por el contrario, puede haber una sola respuesta para muchas llamadas, dándose así una gran flexibilidad funcional. Además MOM es muy estable, incluso en el caso de una caída del sistema operativo, ya que la cola de mensajes siempre podrá ser recuperada. En definitiva, las soluciones basadas en MOM son especialmente recomendables cuando se deben integrar grandes sistemas propietarios o sistemas ampliamente distribuidos.

A favor de las RPCs, sin ninguna duda, debe considerarse que permiten un mayor nivel de abstracción que redundará en una mejor integración funcional entre los sistemas que interactúan. También es importante destacar que este método de trabajo está optimizado para funcionar en arquitecturas cliente/servidor, finalmente su naturaleza síncrona hace que la programación sea más sencilla, ya que el programa llamante queda bloqueado mientras se espera la respuesta, resultando de esta manera menos complejo de programar.

En definitiva, haciendo un resumen algo simplista, las RPCs son adecuadas para situaciones sencillas, pero, cuando el número de sistemas a integrar es alto, o cuando la complejidad de la integración es grande, parece más apropiado utilizar MOM. Adicionalmente, la tecnología MOM ha tenido un fuerte respaldo últimamente al ser adoptada como procedimiento estándar de comunicación asíncrona por el Object Management Group (OMG).

ASPECTOS FUNCIONALES DE LA INTEGRACIÓN DE SISTEMAS

Cuando se diseña un conjunto de sistemas nuevos que deben integrarse, bien utilizando MOM o bien por cualquier otro procedimiento, el problema es fácilmente abordable ya que se pueden presentar dos escenarios diferentes, pero ambos controlables:

Si se trata de nuevos sistemas que se están desarrollando partiendo de cero, o bien a partir de la adaptación de una solución básica, siempre será posible considerar los requisitos de integración dentro del ciclo de vida del proyecto de Ingeniería del Software, por lo que el sistema "nacerá" incluyendo todas las funcionalidades de comunicación que se consideren necesarias, tanto en cuanto a la emisión de mensajes como en cuanto a la recepción de los mismos.

El segundo escenario que se puede presentar es cuando no se realiza un proyecto de Ingeniería del Software para desarrollar el nuevo sistema, sino que

	ASPECTOS	MOM	RPCs
A Favor de MOM	Bloqueo por falta de respuesta	NO	SI
	Respuestas / Llamadas múltiples	SI	NO
	Resistente ante caídas del sistema	+	-
	Apto para integraciones complejas	+	-
A Favor de RPCs	Nivel de abstracción	-	+
	Nivel de integración	-	+
	Optimizado para plataforma Cliente / Servidor	-	+
	Fácil de programar	-	+

Figura 2. MOM versus RPCs, ventajas e inconvenientes.

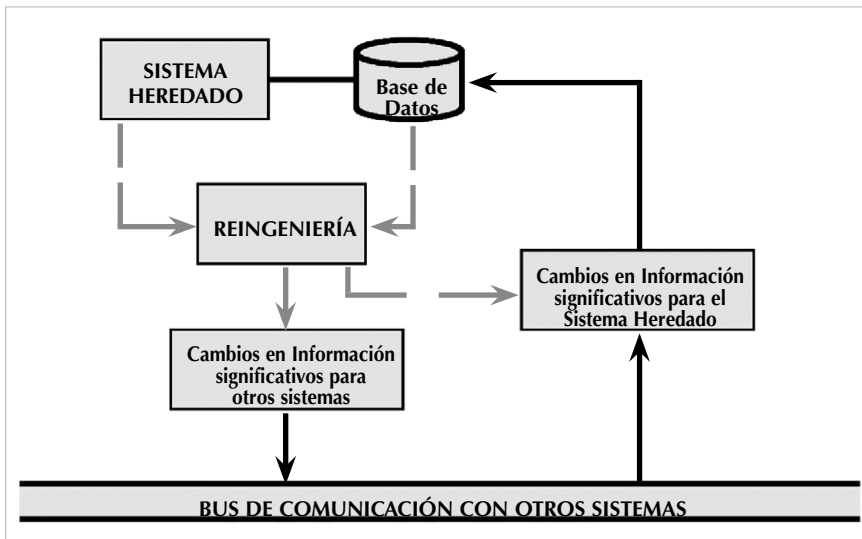


Figura 3. Integración de sistemas heredados.

se adquiere un producto estándar para integrar junto al resto de los sistemas. Tampoco en este caso se presentará ningún obstáculo insalvable, ya que a la hora de seleccionar el producto a integrar se tendrá muy en cuenta la posibilidad de comunicación que aporte la herramienta que finalmente se adquiera.

El auténtico problema se presenta cuando se deben integrar sistemas heredados, probablemente antiguos, en cuyo diseño no se contemplaron las exigencias de comunicación que ahora se plantean.

En este caso, lo primero que se debe realizar es un proyecto de Ingeniería del Software específico para la integración. Evidentemente, dentro de este proyecto, habrá que hacer especial énfasis en las fases de Análisis de Requisitos (requisitos específicos de integración, es decir requisitos funcionales para establecer una comunicación lógica entre sistemas) y Estudio de la Arquitectura (que estará especialmente enfocado a resolver el problema de la comunicación física entre sistemas).

Una vez determinado qué mensajes queremos utilizar (emitir o recibir) para integrar el sistema y cómo se realizará físicamente esa comunicación, sólo queda un problema de diseño que, dependiendo de la naturaleza del sistema a integrar será más o menos sencillo de resolver. (figura 3).

Si el sistema a integrar incluye un API (Application Programming Interface) el diseño será muy sencillo de realizar, ya que sólo será necesario aplicar ese interfase según las pautas funcionales especificadas durante el Análisis de Requisitos.

Si el sistema a integrar no incluye un API, pero es lo suficientemente abierto como para desarrollar uno, sin duda la

mejor solución será desarrollar el API y hacer el diseño tal como se decía en el párrafo anterior.

La tercera situación es sin duda la más compleja, ya que si no se dispone de API ni es posible desarrollarlo, la única solución consiste en realizar un proceso de Ingeniería Inversa que permita, normalmente a través del modelo de datos, producir mensajes de salida para otros sistemas o incluso alterar el contenido de la base de datos propia en función de los mensajes de entrada que provengan de otros sistemas.

SISTEMAS DE INFORMACIÓN Y CONTROL EN EL NUEVO ÁREA TERMINAL DE MADRID-BARAJAS

Aeropuertos Españoles y Navegación Aérea (Aena) se ha encontrado recientemente en una situación de máxima exigencia en cuanto a la necesidad de integrar una gran cantidad de sistemas que, en algunos casos, son sistemas de una gran complejidad. El proyecto de integración ha sido desarrollado con éxito, en buena medida, gracias a la utilización de MOM para la creación de un bus lógico de comunicación a través del que interactúan todos los sistemas del Nuevo Área Terminal (NAT) del Aeropuerto de Madrid-Barajas.

Para dar una idea de la magnitud del problema vale con considerar que estamos hablando de uno de mayores proyectos de ingeniería que se ha abordado en Europa en la última década. El NAT de Madrid-Barajas incluye un Edificio Terminal de casi medio millón de metros cuadrados y un edificio satélite con más de un cuarto de millón de metros cuadrados. También se ha construido un APM (Automated People

Mover) que transportará pasajeros entre ambos edificios, un Sistema Automático de Transporte de Equipajes (SATE) y otras muchas instalaciones y sistemas también de gran complejidad: un túnel de servicios aeroportuarios, seis áreas de aparcamiento con una superficie total de más de trescientos mil metros cuadrados, una central eléctrica, sistemas de control de las instalaciones de las nuevas pistas, (figura 4).

Cada una de estas grandes instalaciones está controlada mediante una serie de sistemas específicos instalados por los distintos suministradores y contratistas. Si bien es cierto que cada sistema de control tiene una función autónoma, por ejemplo el sistema de control del SATE sólo se ocupa de controlar el estado de los 78 km. de cintas autotransportadoras, así como de la situación de cada uno de los equipajes perfectamente identificados individualmente; o el sistema CONOPER (Control de Operaciones) sólo se ocupa de gestionar los vuelos que transportarán a los setenta millones de pasajeros que utilizarán las instalaciones del NAT cada año. No es menos cierto que el aeropuerto en su totalidad necesita gestionar integrada a los pasajeros y a sus equipajes. Es evidentemente necesario que SATE y CONOPER estén integrados.

Esta necesidad de integrar decenas de sistemas funcionalmente muy complejos y de orígenes muy diversos, ha implicado la necesidad de desarrollar un proyecto de integración en el que hemos participado varias empresas, utilizando MOM como base tecnológica para construir la plataforma lógica de comunicaciones.

UN CASO PRÁCTICO: INTEGRACIÓN DEL SISTEMA DE TELEFONÍA

Para dar una idea de la arquitectura informática empleada en el proceso de integración de sistemas, podemos centrarnos en uno de los sistemas más sencillos y elementales del NAT, como es el Sistema de Telefonía.

La integración del Sistema de Telefonía con el resto de los sistemas del NAT logra mejorar la gestión global de la instalación de los y grupos de trabajo del aeropuerto en los siguientes aspectos:

- (1) Permite a los gestores de aeropuerto conocer en todo momento y en tiempo real el estado de disponibilidad y rendimiento del sistema telefónico;
- (2) Suministra información a todos los sistemas y grupos de trabajo relacionados con la telefonía, informando



Figura 4. NAT del aeropuerto Madrid-Barajas.

de los eventos acaecidos e incluso anticipándose a los que puedan ocurrir en el futuro;

(3) Da a conocer al propio Sistema de Telefonía los eventos producidos en otros sistemas de los que, en alguna medida, depende (electricidad, ventilación, etc.) de modo que se pueda optimizar la disponibilidad del servicio telefónico.

El Sistema de Telefonía, desde el punto de vista funcional, podemos considerarlo descompuesto en dos subsistemas diferentes: por un lado el Subsistema de Transporte, que está formado por un anillo principal y un sub-anillo auxiliar de telefonía, así como por un Subsistema de Centralitas.

Desde el punto de vista de control, cada uno de los subsistema es muy di-

ferente, por un lado el Subsistema de Transporte es gestionado directamente por parte de un Operador de Telefonía desde el exterior del aeropuerto, por lo que no se dispone de información para integrar con el resto de sistemas, en cambio, el Subsistema de Centralitas está formado por una serie de equipos físicos, que sí son gestionados por el propio aeropuerto, disponiéndose de un conjunto de componentes software, aportados por el suministrador de las centralitas, que permiten acceder a la información sobre la situación de las centralitas. Se trata del DNA Event Manager de Ericsson y del servidor de control de eventos HP Open View de Hewlett Packard.

El tráfico de información de los elementos de gestión de las centralitas se realiza a través del protocolo SNMP (Simple Network Management Protocol), que interactúa con los componentes de Open View relacionados con la disponibilidad de la centralita (componente EventManager) y con el rendimiento de la centralita (componente Performance Manager). Estos componentes mantienen actualizada una base de datos del Sistema de Telefonía, en este caso utilizando como gestor de base de datos SQL Server de Microsoft, donde se registra periódicamente la situación de todas las centralitas. (figura 5).

El mecanismo de integración establecido por el Plan Barajas de Aena es el MOM de TIBCO que, mediante la publicación y suscripción de mensajes en un bus lógico, permite la integración de los diferentes sistemas de control del aeropuerto. Precisamente estos mensajes informarán sobre fallos, disponibilidades, anomalías y, en general, cualquiera de los parámetros manejados por los correspondientes sistemas informáticos que puedan ser de interés para terceros sistemas.

En esta situación, para integrar el Sistema de Telefonía con el resto de los sistemas del NAT, se dispone de dos posibilidades diferentes: bien utilizar las propias llamadas realizadas a través de SNMP o bien emplear los eventos registrados en la base de datos donde el sistema de control almacena una traza de la situación de las centralitas.

En el primer caso, para la obtención de información a través de SNMP, el elemento de integración serían las funciones propias del API de SNMP, que incluye comandos tales como GET o SET, para consultar o establecer valores específicos de las variables de control de las centralitas, respectivamente. El único problema adicional consiste en realizar un filtrado de los más de 400 códigos de fallo de componentes de las

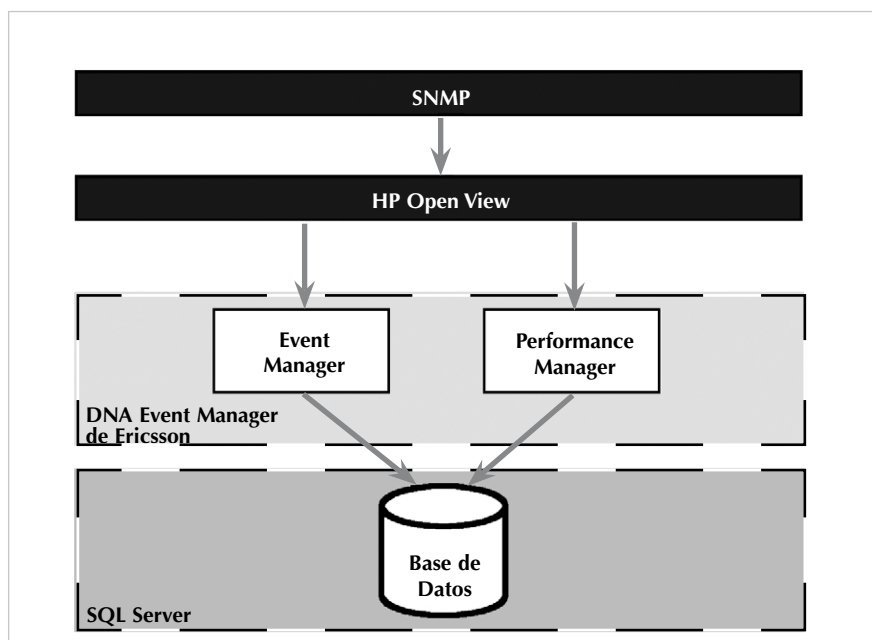


Figura 5. Arquitectura de Integración del Sistema de Telefonía.

centralitas telefónicas, para resumirlos en los que realmente son útiles como información susceptible de interesar a otros sistemas del NAT.

La segunda opción disponible consiste en obtener la información necesaria para publicar mensajes en el bus lógico directamente desde la base de datos de control (LOG del sistema). En este caso la única labor a realizar sería el mismo tipo de filtrado descrito en el párrafo anterior para simplificar la codificación exhaustiva, propia del Sistema de Telefonía, y convertirla en un código estándar e interpretable por el resto de los sistemas.

CONCLUSIONES

Probablemente la principal conclusión de este artículo sea que, cuando hablamos de grandes empresas o instalaciones, donde la propia naturaleza del entorno a gestionar implica la coexistencia de diversos sistemas informáticos, la importancia y complejidad de la integración de sistemas crece proporcionalmente a la cantidad y com-

plejidad de los propios sistemas a integrar.

Desde hace muchos años, ha existido una enorme inquietud ante esta necesidad de integración y, fruto de esa inquietud, han surgido diversos procedimientos de trabajo, entre los que hemos destacado los RPCs y el propio MOM, del que hemos esbozado incluso un breve y simplificado ejemplo.

Tal vez la asignatura pendiente esté en la integración de sistemas antiguos o, en cualquier caso, sistemas que en su diseño inicial no contemplaron ningún mecanismo sofisticado de integración. Hemos visto como se debe acudir a la Reingeniería para, a partir de la propia información gestionada por el sistema, obtener outputs para terceros, e incluso permitir la actualización de la base de datos del sistema en cuestión a partir de inputs recibidos de otros sistemas. Ahora sólo faltaría facilitar esta labor mediante el desarrollo de herramientas mas sofisticadas para la implementación de estos Proyectos de Ingeniería Inversa.



Manuel MUÑOZ GARCÍA. Después de licenciarse en Matemáticas, especialidad de Ciencias de la Computación, por la Universidad Complutense de Madrid en el año 1980, ingresa en el Departamento de Informática de EPTISA, donde llega a desempeñar el puesto de Jefe del Servicio de Mecanización Interior. En 1983 se incorpora al Departamento de Informática de EMPRESARIOS AGRUPADOS. Tras ocupar diversos puestos en esta empresa, donde ha transcurrido toda su carrera profesional, en 1997 es nombrado Director del Departamento de Informática, puesto que continúa desempeñando en la actualidad. También es profesor de Ingeniería del Software en el Departamento de Sistemas Informáticos de la Escuela Técnica Superior de Ingeniería - ICAI, de la Universidad Pontificia de Comillas.

31 Reunión Anual Sociedad Nuclear Española

LOGROÑO

19-21 de octubre de 2005

**Reserve estas fechas
en su agenda**

OCTUBRE						2005	
L	M	M	J	V	S	D	
					1	2	
3	4	5	6	7	8	9	
10	11	12	13	14	15	16	
17	18	19	20	21	22	23	
24	25	26	27	28	29	30	
31							