RM-ODP: El Modelo de Referencia de ISO para el Procesamiento Abierto y Distribuido

Antonio Vallecillo Moreno
ETSI Informática. Universidad de Málaga
av@lcc.uma.es

1. Introducción

La gran difusión de los ordenadores en nuestra sociedad, junto con el aumento de la eficiencia del hardware y las comunicaciones, han ocasionado el establecimiento de una infraestructura excepcionalmente propicia para el desarrollo de aplicaciones distribuidas en sistemas abiertos. Sin embargo, las características específicas de estos sistemas y el tipo de problemas que en ellos se plantean suponen un serio reto para la Ingeniería del Software tradicional, cuyos métodos y herramientas se han visto desbordados por estos nuevos requisitos.

Según un análisis de la revista *ComputerWorld* (en su número del 11 de octubre de 1999), el 85% de los departamentos de tecnologías de la información (IT) de las empresas norteamericanas no son capaces de satisfacer las necesidades estratégicas impuestas por sus propias compañías. Aunque no se ha realizado un informe similar en España, seguro que la situación no es sensiblemente mejor aquí.

Muchas son las razones que pueden esgrimirse para tratar de explicar estos hechos, aunque lo que no podemos ignorar es el continuo cambio al que se ven sometidos los requisitos que propician el actual desarrollo de tecnología. De ser meros centros de cálculo, los departamentos de IT de las organizaciones tradicionales se han visto abocados a producir, mantener y gestionar una tecnología cada vez más compleja, a la vez que heterogénea, abierta, distribuida y en continua evolución. Además, los nuevos sistemas de información han de responder también a las necesidades cada vez más complejas por parte de los usuarios, y dentro de entornos de negocio que evolucionan a un ritmo cada vez más rápido. Desmond D'Souza resume muy bien los requisitos que los negocios demandan de la tecnología en las empresas actuales [2]:

- En primer lugar, los sistemas software de una empresa están para soportar a sus negocios y resolver los problemas reales que éstos plantean, con independencia de si se crean para cubrir una necesidad concreta, o con motivo de una oportunidad tecnológica.
- En segundo lugar, los sistemas han de ser capaces de adaptarse y evolucionar de una forma rápida. En general, se ha visto que es bueno que estén basados en componentes reutilizables, definidos sobre una arquitectura que separe las diferentes áreas del negocio, las distintas reglas que lo rigen, y construida independientemente de la infraestructura tecnológica subyacente. Además, esta arquitectura debe poder acomodar no sólo componentes heterogéneos y distribuidos, sino también aquellos heredados de las aplicaciones ya existentes (*legacy components*).
- Por otro lado, debe utilizarse un vocabulario preciso, con semántica claramente definida, y suficientemente extendido dentro la empresa de forma que sea entendido y compartido por todas las áreas de la organización, eliminando así las tradicionales barreras de comunicación y ambigüedades en el lenguaje que utilizan los distintos departamentos de una empresa (dirección, administración, producción, recursos humanos, ventas, IT, etc.).
- En cuarto lugar, deben utilizarse métodos y arquitecturas "escalables", capaces de crecer desde pequeños proyectos a grandes aplicaciones, y de poder incorporar de forma progresiva los distintos requisitos que va imponiendo el mercado o el propio crecimiento del negocio: robustez, seguridad, mantenibilidad, alta disponibilidad, etc.
- Y por último, los sistemas han de ser "integrables", tanto con otros sistemas de la misma empresa, como con aquellos externos a ella: proveedores, clientes, auditores, incluso con los de la competencia.

Por supuesto, la mayor parte de las empresas son conscientes de estos requisitos, aunque distan mucho de estar en disposición de satisfacerlos. Sin embargo, tampoco pueden abordarlos de forma

individual por los elevados costes que esto puede conllevar. La solución no está en 'reinventar la rueda' por separado, sino que es preciso trabajar conjuntamente para tratar de dar solución a estos problemas de una forma consensuada, global, y definitiva (esto es, con perspectivas y opciones de futuro). Y para eso es precisamente para lo que están los estándares, pues son ellos los que nos ofrecen estas ventajas que necesitamos.

En general, el desarrollo de estándares es fundamental para cualquier cuerpo de Ingeniería, y la Ingeniería del Software no es una excepción. Quizá la juventud de esta disciplina, junto con el tamaño (¿manejable?) de los sistemas y aplicaciones desarrollados hasta ahora no haya propiciado la aparición de estándares 'oficiales', y aun menos el uso de los pocos existentes. Sin embargo, la creciente complejidad de los sistemas abiertos y distribuidos, junto con los requisitos mencionados anteriormente, hacen imposible la construcción de aplicaciones software sin la ayuda de normas internacionales que permitan reutilizar el conocimiento y experiencia ya existente en estos campos. Y no sólo se obtiene la ventaja de aprovecharse de esa experiencia a la hora de construir una aplicación, sino también de que el resultado obtenido pueda ser fácilmente integrado con otras aplicaciones que sigan también esos estándares.

Las organizaciones ISO (*International Standards Organization*, http://www.iso.ch/) e ITU-T (*International Telecommunication Union*, conocido antes como CCITT, http://www.itu.int/) se plantearon ya hace unos años la elaboración conjunta de una serie de estándares para el desarrollo de aplicaciones abiertas y distribuidas. El objetivo es definir un modelo de referencia (conocido como RM-ODP, *Reference Model – Open Distributed Processing*) que permita integrar toda una serie de estándares sobre estos temas, manteniendo consistencia entre ellos. Su punto de partida es la necesidad de tratar la complejidad intrínseca de este tipo de aplicaciones, cuyas especificaciones, diseño e implementación necesitan de un marco de trabajo que las estructure convenientemente para poder abordarlas con éxito.

Muchas son las contribuciones que aporta RM-ODP para el desarrollo de aplicaciones abiertas y distribuidas, entre las que destacamos las siguientes:

- RM-ODP proporciona un marco de trabajo conceptual y una arquitectura que integra aspectos relacionados con la distribución, interoperabilidad y portabilidad de sistemas software, y de forma que la heterogeneidad del hardware, sistemas operativos, redes, lenguajes de programación, bases de datos y distintas formas de gestión sean transparentes al usuario.
- RM-ODP proporciona un marco de coordinación para la normalización del procesamiento abierto y distribuido, que no sólo trata de aunar aquellos estándares actuales propios de estos temas, sino dar también cabida de forma natural al desarrollo de otros nuevos conforme vaya surgiendo su necesidad.
- RM-ODP define de forma clara y precisa aquellos conceptos que aparecen en el desarrollo de plataformas de componentes distribuidos, proporcionando un vocabulario y un marco semántico común a todos los participantes y usuarios de las aplicaciones (desde los distintos departamentos de la empresa hasta los desarrolladores y usuarios finales). RM-ODP propicia, dentro de lo posible, el uso de técnicas de descripción formal para la especificación de estos conceptos y de la propia arquitectura, como pueden ser ESTELLE, LOTOS, SDL, o Z.

En lo que sigue vamos a explorar un poco más a fondo la estructura de RM-ODP, sus principales elementos, los estándares que lo componen, y las ventajas que ofrece al desarrollador de aplicaciones distribuidas.

2. El Modelo de referencia RM-ODP

Tal y como lo definen ISO e ITU-T, el modelo de referencia de procesamiento abierto y distribuido RM-ODP proporciona un marco de coordinación para la normalización del desarrollo de aplicaciones abiertas y distribuidas, creando una arquitectura capaz de soportar de forma integrada aspectos tales como la distribución, interoperabilidad o portabilidad de los sistemas, objetos y componentes.

2.1 Normas básicas

El núcleo central del modelo de referencia RM-ODP está recogido en cuatro normas fundamentales, que definen y especifican su estructura y elementos básicos:

- **Visión de conjunto** (ISO/IEC 10746-1; ITU-T X.901). Esta norma contiene una visión de conjunto de las motivaciones de ODP, presentando el alcance, la justificación y la explicación de sus conceptos esenciales, así como una descripción de su arquitectura.
- **Fundamentos** (ISO/IEC 10746-2; ITU-T X.902). Esta norma contiene la definición (en lenguaje natural) de los conceptos básicos de RM-ODP, así como un marco analítico para la descripción normalizada de sistemas de procesamiento abierto y distribuido. Presenta también los principios de conformidad con las normas ODP, y la forma en que éstos deben aplicarse. En sólo 18 páginas, esta norma es capaz de sentar las bases de todo el modelo, de una forma clara, concreta, y precisa.
- **Arquitectura** (ISO/IEC 10746-3; ITU-T X.903). Esta norma contiene la especificación de las características que debe tener un procesamiento distribuido para que pueda ser considerado como abierto, así como las restricciones que deben cumplir aquellas normas que deseen integrarse en este modelo de referencia. Asimismo, define los distintas *puntos de vista* (*viewpoints*) o subdivisiones que pueden hacerse de un sistema desde diferentes perspectivas, y que ayudan a su comprensión y especificación global.
- **Semántica arquitectural** (ISO/IEC 10746-4; ITU-T X.904). Esta norma contiene una formalización de los conceptos del modelo, utilizando para ello diferentes técnicas de descripción formal.

2.2 Conceptos fundamentales

Las normas anteriores establecen una serie de conceptos fundamentales sobre los que se apoya el modelo para la normalización de sistemas abiertos y distribuidos, y que describimos a continuación:

- La especificación de un sistema en términos de especificaciones de diferentes *puntos de vista* (*viewpoints*), que están interrelacionados entre sí.
- El uso de un *modelo de objetos común* para la especificación del sistema desde cada uno de los puntos de vista.
- La definición de una infraestructura que permita ocultar ciertas complejidades inherentes a los sistemas distribuidos, simplificando la especificación y diseño de las aplicaciones (*transparencias de distribución*).
- La definición de un conjunto de *funciones comunes* que son de utilidad general para la especificación y construcción de sistemas abiertos y distribuidos.
- Un marco para evaluar la *conformidad* del sistema respecto a las propias normas que define ODP.

Pasemos a describir estos conceptos con un poco más de detalle.

2.2.1 Puntos de vista

En general, la extensión y complejidad de un sistema de información impiden que una sola persona pueda abarcar todos y cada uno de sus aspectos. Por otro lado, cada persona tiene también sus propios intereses, necesidades, y perspectivas distintas desde donde abordar y examinar las especificaciones de un sistema: un ejecutivo hace preguntas muy distintas de las que hace un administrativo, o uno de los implementadores del sistema. Lo que RM-ODP proporciona es un marco de referencia mediante el cual poder examinar, describir y especificar un sistema desde distintas perspectivas, denominadas *puntos de vista*. Cada uno de estos puntos de vista trata de satisfacer a una audiencia distinta, cada una interesada en aspectos diferentes del sistema. Y asociado a cada uno de los puntos de vista se define un *lenguaje* especializado, que recoge el vocabulario y la forma de expresarse de la audiencia concreta a la que se dirige.

RM-ODP define cinco puntos de vista genéricos, y que considera básicos a la hora de definir un sistema:

- El punto de vista de la **empresa**, que describe los requisitos desde la perspectiva del propio negocio, así como la manera en la que pretende satisfacerlos. Se centra pues en la finalidad, alcance, entorno y políticas que rigen las actividades del sistema especificado, dentro de la organización de la que forma parte.
- El punto de vista de la **información**, que escribe el tipo de información que va a manejar el sistema, así como la estructura de los datos y sus posibles valores. Se centra en las clases de información

tratadas por el sistema, su semántica, y las restricciones impuestas sobre la utilización e interpretación de dicha información.

- El punto de vista **computacional**, que describe la funcionalidad que ha de ofrecer el sistema, así como su descomposición y organización funcional. Para ello trata de describir el sistema como un conjunto de objetos que interactúan entre sí, definidos mediante interfaces.
- El punto de vista de la **ingeniería**, que describe la infraestructura necesaria para soportar el procesamiento distribuido del sistema, así como la forma de distribución de los datos y operaciones que permitan al sistema proporcionar la funcionalidad requerida.
- El punto de vista de la **tecnología**, encargado de describir la tecnología que soportará el sistema en base a la infraestructura de hardware, software y comunicaciones que permita el procesamiento y la funcionalidad necesaria, así como la representación y distribución de los datos.

Aunque estos puntos de vista son diferentes y se especifican por separado, es importante señalar que todos ellos especifican a *un mismo* sistema, y por tanto son complementarios, nunca contradictorios entre sí; la segunda norma de RM-ODP establece un marco común que garantiza la coherencia mutua entre estos cinco puntos de vista. Para ello utiliza el modelado basado en objetos que define RM-ODP (ver siguiente punto), y que proporciona una base común para los distintos lenguajes de los puntos de vista. Esto hace posible identificar relaciones y correspondencias entre las diferentes especificaciones y representaciones del sistema que se obtienen desde cada uno de ellos.

2.2.2 Uso de un modelo de objetos común y de técnicas de orientación a objetos

RM-ODP utiliza un modelo de objetos común como base para las distintas especificaciones, así como técnicas de orientación a objetos para modelar el sistema general y desde cada uno de los puntos de vista. Estos métodos se muestran muy apropiados por las ventajas en cuanto a abstracción y encapsulación que proporcionan. La abstracción permite resaltar los detalles interesantes del sistema sobre aquellos que no se consideran relevantes a un determinado nivel, y describirlos de forma independiente de cualquier implementación concreta. La encapsulación permite definir servicios y funciones ocultando detalles innecesarios sobre su implementación, su posible heterogeneidad, o los mecanismos de provisión de los servicios concretos utilizados por los clientes y servidores.

2.2.3 Transparencias

Las transparencias permiten ocultar las complejidades inherentes a cualquier sistema distribuido a la hora de especificarlo, permitiendo una mayor capacidad de abstracción. Entre ellas destacamos los siguientes aspectos:

- las diferencias de representación de datos y mecanismos de invocación para servicios entre sistemas heterogéneos (transparencia de acceso);
- la ocultación de los posibles fallos que puedan ocurrir en otros objetos, así como de su recuperación, para permitir ciertos aspectos de tolerancia a fallos (transparencia de fallo);
- la búsqueda e invocación de servicios sin necesidad de especificar su ubicación, y con independencia de su migración o posible reubicación (transparencias de ubicación, migración y reubicación);
- la existencia de múltiples copias de un mismo servicio para proporcionar fiabilidad y disponibilidad (transparencia de replicación);
- la coordinación, supervisión y recuperación de las transacciones entre objetos de forma externa y sin tener que involucrar a los propios objetos (transparencia de transacciones).

RM-ODP define una serie de funciones y estructuras para realizar este tipo de transparencias. Sin embargo, cada una de ellas puede llevar asociado un compromiso entre la calidad de su funcionamiento requerido y su coste (ya sea en tiempo o en recursos); de igual forma, puede que no se desee que todas las transparencias sean relevantes en todos los casos. Por tanto, ODP no obliga a soportar todas estas transparencias, aunque sí indica que, en caso de soportar alguna de ellas, debe hacerse de acuerdo a sus normas para garantizar la conformidad con este estándar, y la integración con otros sistemas ODP.

2.2.4 Funciones comunes

Una ventaja muy importante en el desarrollo de aplicaciones distribuidas es poder contar con una serie de funciones generales para realizar muchos de los servicios que son precisos en estos ambientes. RM-ODP

describe, como parte de su arquitectura básica, 24 funciones comunes a todos los sistemas abiertos y distribuidos. Dichas funciones están organizadas en cuatro grupos –funciones de gestión, coordinación, repositorio y seguridad–, y se utilizan también internamente en ODP para soportar algunos requisitos del lenguaje del punto de vista de ingeniería, y para implementar algunas de las transparencias. La existencia de esos servicios o funciones comunes ya definidos va a permitir la construcción de aplicaciones distribuidas de una forma modular, y poder mejorar notablemente el tiempo de desarrollo y la fiabilidad del sistema resultante. En la arquitectura de RM-ODP sólo se describen brevemente estas funciones, dejando la especificación detallada de cada una de ellas para normas separadas. Una de las más importantes es la función de intermediación (*trading*), que por su relevancia será discutida más adelante.

2.2.5 Conformidad

Como mencionamos al principio, una de la ventaja de la existencia de normas internacionales es la posibilidad que ofrecen a los sistemas que son conformes a ellas de integrarse entre sí y asegurar su interoperabilidad. En los sistemas abiertos y distribuidos, la evolución y heterogeneidad de sus componentes obligan a disponer de mecanismos que permitan incorporar diferentes partes por separado, tanto en tiempo como en espacio, así como independientemente de los proveedores que las hayan construido [5]. Por ello es muy importante que los comportamientos de las diferentes partes de un sistema estén claramente definidos, y que sea posible identificar mecanismos para garantizar que un cierto componente del sistema cumple sus especificaciones. El marco definido en RM-ODP para regir la evaluación de la conformidad trata estas cuestiones, abarcando distintas facetas: desde la identificación de los denominados puntos de conformidad hasta la especificación de la naturaleza de los enunciados de conformidad que habrán de hacerse en cada punto de vista, y la relación entre ellos.

2.3 Otras normas de RM-ODP

Aparte de las cuatro normas básicas que definen los fundamentos del modelo de referencia, RM-ODP va construyéndose y completándose mediante nuevas normas que describen con detalle algunos de los aspectos y conceptos que inicialmente se mencionan en el modelo básico. En este apartado vamos a describir otros estándares que RM-ODP ofrece actualmente, y que corresponden a varios conceptos muy importantes dentro del modelo: la función de intermediación, el esquema de asignación de nombres, el lenguaje de definición de interfaces, y la forma de referenciar a dichas interfaces y vincular los correspondientes objetos.

2.3.1 La función de intermediación

El concepto de intermediación (*trading*) es fundamental en RM-ODP para la oferta y demanda de servicios distribuidos, y constituye una de las funciones comunes que hemos mencionado anteriormente. Un intermediario o corredor de servicios (*trader*) es una entidad capaz de almacenar información acerca de posibles servicios, de forma que potenciales proveedores de los mismos puedan registrarse en él. Así, los clientes pueden enviar sus peticiones a dicho intermediario, quien se encarga de localizar a un proveedor de entre los que tiene registrados. Una vez ha localizado a alguno que satisfaga los requisitos que impone el cliente, envía la referencia de dicho servidor al cliente para que ellos puedan interactuar entre sí, sin necesidad del intermediario.

La norma que describe esta función de intermediación en RM-ODP (ISO/IEC 13235-1; ITU-T X.950) es la encargada de especificarla de forma independiente de cualquier implementación, asegurar que el servicio puede hacerse de forma federada entre varias funciones de intermediación distribuidas (sean o no del mismo fabricante), y ofrecer suficientes detalles que permitan evaluar alegaciones de conformidad a esta norma. Una característica importante de esta norma es que es capaz de especificar la funcionalidad y el comportamiento de la función de intermediación utilizando un lenguaje de notación formal como es Z.

Por otro lado, la segunda de las normas que describen esta función (ISO/IEC 13235-3; ITU-T X.951) complementa a la anterior, especificando cómo la función de intermediación puede hacer uso del servicio de directorio X.500 tal y como lo define OSI (*Open Systems Interconnection*).

2.3.2 El esquema de asignación de nombres

La asignación de nombres de forma global es un serio problema en los grandes sistemas abiertos y distribuidos, más aún cuando éstos pueden estar gestionados simultáneamente por más de una entidad. De

ahí que RM-ODP especifique que los nombres han de asignarse de una forma que dependa del contexto, y en este sentido la norma ISO/IEC 14771 (ITU-T X.910) proporciona un esquema de asignación de nombres general dentro del marco de referencia.

2.3.3 El lenguaje de definición de interfaces de ODP

Para poder especificar los servicios que ofrecen los objetos que forman parte un sistema abierto y distribuido, es preciso contar con algún lenguaje preciso, bien definido, e independiente de cualquier posible representación de los datos o estructuras que define, así como de la futura implementación de los objetos que especifica. La norma ISO/IEC 14750 (ITU-T X.920) define dicho lenguaje, al que se conoce como *lenguaje de definición de interfaces* de ODP, o ODP IDL por su acrónimo en inglés. Su principal objetivo es describir la signatura de los objetos que especifica, en términos de las estructuras de datos que se manejan y el perfil de las operaciones que definen sus servicios. De esta forma se consigue la ocultación necesaria para el desarrollo de aplicaciones abiertas. Además, el ODP IDL está totalmente en consonancia con el lenguaje de IDL que se define para CORBA, desarrollado por el consorcio internacional OMG (*Object Management Group*, http://www.omg.org/).

2.3.4 Referencias a interfaces y vinculación

Puesto que en un modelo de objetos los servicios vienen definidos por una interfaz, la forma de referenciar interfaces se convierte en fundamental para la interacción de objetos y sistemas en ODP, así como para conseguir la federación de grupos de sistemas. En RM-ODP, una referencia a una interfaz incluye la información necesaria para establecer vinculaciones entre objetos, incluidas aquellas entre objetos que soporten varios protocolos de comunicación, o que se encuentren ubicados en dominios de gestión distintos. Asimismo, una referencia a una interfaz ha de contener suficiente información para soportar la transparencia de reubicación, ya comentada anteriormente. La norma ISO/IEC 14753 (ITU-T X.930) es la encargada de especificar todos estos aspectos, fundamentales a la hora de construir aplicaciones distribuidas.

2.4 Normas actualmente en proceso

Aparte de las normas que acabamos de describir, los comités de normalización de ISO e ITU-T trabajan actualmente en la elaboración de otras normas que formarán parte del modelo de referencia RM-ODP una vez sean consensuadas y publicadas. Estas normas profundizan sobre más aspectos fundamentales del modelo, como pueden ser la forma de interactuar de los objetos computacionales del mismo, la especificación de ciertas funciones comunes, o sobre cómo manejar los aspectos de calidad de servicio de una aplicación. En concreto, algunas de las normas actualmente en proceso de elaboración son:

- ODP-Protocol Support for Computational Interactions (ISO/IEC 14752; ITU-T X.931)
- ODP-Type Repository Function (ISO/IEC 14769; ITU-T X.960)
- ODP-Reference Model: Enterprise Viewpoint (ISO/IEC 15414; ITU-T X.911)
- ODP-Reference Model: Quality of Service (ISO/IEC 15935; ITU-T X.905)

3. Utilidad práctica del modelo

Una vez presentada la estructura y composición básica de ODP, su motivación, y las principales ventajas que ofrece este modelo de referencia para el desarrollo de sistemas abiertos y distribuidos, cabe preguntarse por la utilidad práctica que tiene. De hecho, ODP suele verse a un nivel demasiado abstracto y elevado para que pueda tener aplicación directa en nuestras empresas, o en los desarrollos que acometemos. Sin embargo, esto no tiene por qué ser así, como pretendemos mostrar aquí.

En primer lugar, ya existen aplicaciones concretas que se han basado en RM-ODP para ser especificadas, diseñadas, y construidas, así como empresas que están utilizando este modelo de referencia para tratar de organizar sus aplicaciones de IT de acuerdo a un norma consensuada internacionalmente y con amplias perspectivas de futuro (desde bancos suizos a empresas de telecomunicaciones norteamericanas).

En segundo lugar, ya existe tecnología que soporta este modelo de referencia lo suficientemente madura y extendida como para que construir aplicaciones basándose en él no sea una apuesta arriesgada.

Por ejemplo, la plataforma de objetos distribuidos CORBA (*Common Object Request Broker Architecture)* definida por OMG, está totalmente en consonancia con RM-ODP, y presenta soluciones tecnológicas muy probadas que soportan varios de los puntos de vista. En particular, CORBA ofrece una plataforma de componentes distribuidos junto con una serie de servicios y facilidades comunes (similares a las funciones comunes de RM-ODP) para el desarrollo de aplicaciones distribuidas en ambientes heterogéneos y multidisciplinares. Y en el campo de las telecomunicaciones, TINA (*Telecommunications Information Networking Architecture*) definida por TINA-C (TINA *Consortium*) es una arquitectura para el desarrollo de aplicaciones de este tipo basada en los conceptos que ofrece RM-ODP, y que actualmente proporciona la infraestructura más rica y consolidada dentro de esta disciplina.

En cuanto al uso que podemos darle desde nuestra situación particular, RM-ODP nos ofrece varias ventajas muy interesantes, entre las que destacamos las siguientes:

- En primer lugar, nos obliga a pensar desde diferentes perspectivas o puntos de vista, lo que permite un mejor análisis y recolección de los requisitos que hemos de imponerle a nuestros sistemas.
- En segundo lugar, proporciona una infraestructura y un modelo común desde donde los requisitos expresados en diferentes lenguajes (los de los distintos puntos de vista) pueden ser integrados para formar un sistema globalmente consistente.
- En tercer lugar, proporciona toda una serie de *patrones de razonamiento* ya establecidos sobre los que apoyarnos a la hora de especificar y diseñar el sistema, y poder identificar sus elementos fundamentales y las relaciones entre ellos. En este sentido, RM-ODP nos va a ayudar a hacer las preguntas adecuadas a los responsables apropiados, con un nivel suficiente de abstracción y de precisión para razonar de forma correcta sobre los sistemas.
- Y por último, RM-ODP ofrece un conjunto de mecanismos muy útiles a la hora de diseñar y desarrollar aplicaciones distribuidas, junto con un soporte tecnológico suficientemente maduro como para construir aplicaciones robustas, eficientes y competitivas, a la vez que integrables con otros sistemas que cumplan estos estándares.

Para saber más sobre RM-ODP, aparte de leer las propias normas, existen varios foros en donde se discute sobre estos temas. Uno de ellos lo constituyen los seminarios que Haim Kilov suele organizar anualmente sobre semántica de comportamiento en conferencias como ECOOP y OOPSLA, y que tratan de incorporar semántica precisa a las especificaciones y el diseño de los sistemas [1,3,4]. Pero también están las reuniones plenarias del subcomité SC7 de ISO (p.e. la que da origen al número especial de esta revista), que sirven para congregar a los miembros de todos los comités implicados en la normalización de RM-ODP, hacerles discutir sobre estos temas, e impulsar el avance de estas normas.

4. Referencias

1. K. Baclawski, H. Kilov, A.E. Thalassinidis, K. Tyson (Eds). *Proceedings of the 8th OOPSLA Workshop on Behavioral Semantics*. Denver (CO). November 1999.
2. D. D'Souza. Enterprise integration. Enterprise components with Catalysis/UML. En [1] pp. 49-63.
3. H. Kilov, W. Harvey (Eds.). *Object Oriented Behavioral Specifications.* Kluwer Academic Publishers, 1996.
4. H. Kilov, B. Rumpe, I. Simmonds (Eds.). *Behavioral Specifications of Business and Systems*. Kluwer Academic Publishers, 1999.
5. C. Szyperski. *Component Software. Beyond Object-Oriented Programming*. Addison-Wesley Longman, 1998.

Biografía

Antonio Vallecillo Moreno es licenciado en Matemáticas y Doctor Ingeniero en Informática por la Universidad de Málaga. Actualmente es profesor de esa Universidad y Director de su Servicio Central de Informática. Aparte de su experiencia docente, su carrera profesional ha estado muy ligada a la empresa privada, en donde ha trabajado en diversas multinacionales de informática y telecomunicaciones, tanto en España como en Inglaterra. Su investigación se centra actualmente en los modelos de componentes para sistemas abiertos, y en el uso industrial de los métodos formales.