

**XXVI Convocatoria de Premios “Ingenieros de
Telecomunicación” 07!**



**Premio BANESTO a la Mejor Tesis Doctoral en Tecnologías de
la Información y las Comunicaciones en la Banca**

**José Luis Ruiz Revuelta
Doctor Ingeniero de Telecomunicación**

1. Resumen de la Tesis Doctoral

1.1. Introducción

El presente documento es un resumen de la tesis doctoral que tiene como objetivo presentar la candidatura de la tesis cuyos datos aparecen reflejados en la Tabla 1 a la XXVIII Convocatoria “Ingenieros de telecomunicación”, convocada por el COIT en el año 2007. Tal y como establece dicha convocatoria el resumen consta de cuatro apartados principales: resumen (dividido en origen, objetivos, desarrollo y conclusiones), originalidad, resultados y aplicabilidad.

Tabla 1 Datos de la Tesis Doctoral

<i>Título</i>	A POLICY-DRIVEN, MODEL-BASED SOFTWARE AND SERVICES DEPLOYMENT ARCHITECTURE FOR HETEROGENEOUS ENVIRONMENTS
<i>Programa de Doctorado</i>	Doctorado en Ingeniería de Sistemas y Servicios Telemáticos, departamento DIT, de la ETSIT, UPM. Programa con Mención de Calidad concedida por el MEC (Ref. MCD 2007-00131)
<i>Fecha de Lectura</i>	Jueves 5 de julio de 2007
<i>Calificación</i>	Sobresaliente <i>Cum Laude</i>
<i>Director</i>	Dr. Juan Carlos Dueñas López, Profesor Titular del departamento DIT de la ETSIT de la UPM.
<i>Autor</i>	José Luis Ruiz Revuelta, Doctor Ingeniero de Telecomunicación

1.2. Origen

El origen de la tesis doctoral está principalmente relacionado con la problemática del despliegue de software y servicios, la cual se presenta de forma horizontal en múltiples ámbitos de negocio. En particular en lo que concierne a esta tesis el contexto de desarrollo ha estado marcado por el aprovisionamiento de servicios en el sector residencial y el mercado de banca. En lo que concierne al despliegue de servicios en el mercado de banca el desarrollo estuvo en su etapa final ligado a la participación del autor en el proyecto de I+D nacional CENIT ITECBAN que tiene como principal objetivo la creación de un nuevo núcleo bancario basándose en principios de arquitecturas orientadas a servicios y desarrollos dirigidos por modelos (MDA - *Model Driven Architecture*). La aplicación de la arquitectura de despliegue desarrollada en la tesis tuvo como contexto de desarrollo original el sector residencial, para lo cual tuvo especial relevancia la participación del autor en los proyectos de investigación europeos ITEA Osmose (*Open Source Middleware for Open Systems in Europe*) e ITEA Osiris (*Open Source Infrastructure for Runtime Integration of Services*). ITECBAN y OSIRIS son proyectos aún en curso y su desarrollo final incorporará en ambos sectores industriales (banca y sector residencial) resultados de esta tesis doctoral.

El despliegue de software y servicios, por simplificar nos referiremos a él simplemente como despliegue, se asocia al conjunto de actividades relacionadas con la transición de activos de servicio desde los entornos de producción a los entornos de consumo,

incluyendo su implantación y posterior operación. La amplia adopción de la ingeniería software basada en componentes, CBSE (*Component-Based Software Engineering*), es uno de los factores que ha contribuido al aumento del interés en el despliegue de software. Destaca de los componentes el hecho de que se pueden considerar como unidades independientes de despliegue. Existe cierta confusión entre la orientación a objetos (OO) y el desarrollo basado en componentes. En realidad la orientación a componentes no obliga a utilizar lenguajes OO, pero en la práctica existe cierta relación entre ambos, puesto que en la actualidad son muchos los sistemas construidos con lenguajes OO. Incluso en este último caso, son los componentes y no los objetos las unidades que se despliegan realmente, aun cuando éstos se componen generalmente de clases, desagregadas o empaquetadas en bibliotecas. CBSE fomenta la creación de aplicaciones mediante la composición de componentes, de manera análoga a un modelo de Lego. La construcción de aplicaciones mediante ensamblado de componentes promueve la reutilización del software y así la posibilidad de crear nuevas aplicaciones con menos esfuerzo.

Las Arquitecturas Orientadas a Servicios, en inglés SOA (*Service Oriented Architectures*), han ido un paso más lejos que la CBSE. La orientación a servicios promueve la creación de nuevas aplicaciones mediante la combinación de servicios proporcionados por distintos actores. Adicionalmente, las SOA agregan dos aspectos beneficiosos al puzzle de la Ingeniería de Servicios, la capacidad de dinámicamente activar y desactivar servicios, y la posibilidad de que el ensamblaje del sistema pueda tener lugar al vuelo sobre sistemas en ejecución.

Juntos, CBSE y SOA configuran un soporte arquitectónico básico para que los proveedores de software y servicios ofrezcan sus productos - empaquetados como componentes a los usuarios finales o a actores intermedios que crean servicios mediante servicios preexistentes. Además, Internet permite que los proveedores de servicio proporcionen remotamente a sus clientes nuevos servicios o actualicen los que ya tienen disponibles. Gracias a Internet, los servicios pueden ser ofrecidos en cualquier parte del planeta, a bajo coste y rápidamente. La conectividad ubicua de la red de banda ancha ha propiciado una gran variedad modelos de negocio en el aprovisionamiento de servicios, entre ellos el paradigma de los Servicios Web (Web Services). Sin embargo, el soporte para la automatización de los procesos de despliegue es esencial para alcanzar un mercado de servicios maduro. La ejecución manual de los procesos del despliegue no es una opción viable en entornos de aplicación a gran escala, tales como el aprovisionamiento del servicio en entornos residenciales o en organizaciones con una clara orientación de servicio. Claros ejemplos de ellos son el sector de banca o el de telecomunicaciones. Todo ello no sólo porque la ejecución manual es propensa a errores, sino también porque no es rentable, puesto que los costes crecen exponencialmente con el número de ubicaciones sobre las cuales hemos de desplegar servicios. La automatización completa de los procesos de despliegue es sin duda una característica deseada, pero aún sigue siendo un problema sin resolver. La gestión de dependencias es uno de los factores que hacen que la automatización continúe siendo un desafío. Hoy en día, la mayor parte de los sistemas no se construyen desde cero, sino como una combinación de piezas de software, algunas desarrolladas especialmente para el sistema, pero muchas de ellas disponibles de antemano. Una aplicación puede tener dependencias que afectan a múltiples capas en el entorno de despliegue; el sistema operativo, bibliotecas comunes u otros servicios de alto nivel. Por consiguiente, para desplegar con éxito un servicio, todos sus elementos requeridos, de aquí en adelante

dependencias, deben ser desplegados. Existe un factor que complica aún más el proceso de resolución de dependencias, el problema del versionado de los componentes software. El versionado es la estrategia básica de control en la evolución de los sistemas. Versiones más modernas son desarrolladas para proporcionar nuevas características, para mejorar el funcionamiento o para corregir errores. Los cambios entre las versiones del mismo software pueden incluir cambios en los servicios ofrecidos que afectarían obviamente a componentes dependientes. Hasta ahora, solamente se han mencionado las dependencias a nivel de software. Sin embargo, un proceso avanzado de despliegue debe también considerar otros factores relevantes, en particular: la adaptación a las capacidades del contexto de despliegue (incluyendo nodos y redes de comunicaciones) y las preferencias y la localización del usuario. Conceptualmente es bastante similar a manejar dependencias software, pero en un dominio diferente, y por tanto necesitaremos un vocabulario diferente. Una de las metas de esta tesis es proponer una arquitectura de despliegue que pueda hacer frente a la heterogeneidad de los entornos de despliegue. Varias iniciativas han intentado automatizar los procesos del despliegue: basados en *scripts*, basados en lenguajes OO y dirigidos por modelos. Los basados en *scripts* fueron históricamente la primera tentativa de automatización de procesos. Los scripts son una solución muy buena cuando hay poca variabilidad en el proceso del despliegue. La reutilización mejora usando lenguajes de programación OO, aunque a costa de suponer un coste de aprendizaje más alto. La razón es que usar lenguajes de programación OO pone en nuestras manos mecanismos potentes para la reutilización como la herencia y la composición, por ejemplo es posible crear una nueva tarea extendiendo una preexistente. La utilización de modelos abre nuevas posibilidades de automatización. Los recursos, los sistemas y los servicios pueden ser modelados por separado. Por lo tanto, herramientas más específicas e inteligentes pueden ser aplicadas. Combinando estas herramientas es posible crear una solución mucho mejor que los bloques monolíticos que se están utilizando hoy en día.

1.3. Objetivos

El despliegue de sistemas es la disciplina encargada de controlar aquellos aspectos directa e indirectamente relacionados con los procesos de distribución de software. Desde las sedes de proveedores de software y servicios hasta las sedes de los consumidores. La mayor parte de los sistemas de información actuales están constituidos por la interacción de múltiples componentes en tiempo de ejecución. Además, a menudo estos sistemas están sometidos a modificaciones frecuentes. Estos cambios pueden tener múltiples causas: la mejora de prestaciones, corrección de errores o la inclusión de nuevas funcionalidades.

En consecuencia, los aspectos relacionados con el despliegue de software y servicios tienen un impacto innegable sobre los costes de implantación, mantenimiento y operación de los sistemas de información y comunicaciones.

La automatización de procesos relacionados con el despliegue tiene múltiples ventajas; el ahorro de costes y la disminución de errores son algunas de las más importantes. Lamentablemente existen muchos factores que hacen que la tarea de automatización de estos procesos sea una actividad compleja. Básicamente, las dificultades radican en el complejo patrón de dependencias de recursos que presentan las unidades de software. Es necesario modelar y dar una respuesta a estas dependencias, que por desgracia

abarcan aspectos a diferentes niveles, pero interrelacionados: necesidades de recursos lógicos (software), físicos (hardware) y de red. Todos ellos han de ser tenidos en cuenta si pretendemos ejecutar operaciones automáticas de despliegue de manera completa y correcta.

La contribución principal de esta tesis es la propuesta de un sistema de despliegue dotado de una arquitectura que permite ofrecer adaptación contextual en sistemas distribuidos. Para que un sistema de despliegue se adapte al contexto es necesario que sea capaz de reaccionar y adaptarse a los recursos disponibles en el objetivo de una operación de despliegue. Y con ello me refiero a los anteriormente mencionados, recursos lógicos (*software*), físicos (*hardware*) y de red (*networking*). En cuanto a la distribución, en realidad nos estamos refiriendo a varios aspectos. En primer lugar a la ejecución de procesos de despliegue, que pueden iniciarse en sedes remotas. En segundo lugar, nos referimos al acceso vía red a repositorios de recursos lógicos. Y en tercer y último lugar nos estamos refiriendo al hecho de que se pretende ofrecer una propuesta sobre cómo gestionar procesos de despliegue sobre redes de nodos interconectados. Debido a este último aspecto de distribución, será necesario tomar decisiones sobre cómo distribuir los componentes de un producto o servicio entre los nodos de la red, en función de sus recursos y de ciertas políticas de distribución.

La arquitectura de despliegue propuesta incluye modelos para la descripción de sistemas software y de redes de nodos de forma que sea posible adaptar los procesos de despliegue a los recursos disponibles. Adicionalmente los procesos incluidos en la arquitectura proponen la utilización de mecanismos de personalización basados en políticas que permiten adaptar el comportamiento del sistema a los intereses de los actores involucrados en múltiples escenarios.

1.4. Desarrollo

El desarrollo de esta tesis doctoral ha generado cuatro contribuciones principales. Primero, la relación entre los aspectos no funcionales y el ciclo de vida de despliegue. Particularmente: la adaptación al contexto, la planificación de despliegue, la automatización de los procesos y la reacción a los cambios del contexto de despliegue.

En segundo lugar, se han creado modelos de información para describir el software y los sistemas distribuidos. Los modelos están disponibles como esquemas XML y como modelos EMF (*Eclipse Modelling Framework*). EMF genera modelos eMOF, el cual es a su vez un subconjunto del lenguaje de meta-modelado OMG MOF (Meta Object Facility). El modelo de software distingue básicamente entre productos software, características del software y unidades de despliegue. Los productos software son las unidades de adquisición. Las características del software son unidades seleccionables dentro de un producto de software. Las unidades de despliegue son las que proporcionan en ejecución las características del software y por tanto también los productos. El modelo de sistema distribuido para despliegue distingue básicamente nodos, routers y enlaces. Ambos modelos se han diseñado para facilitar la adaptación al contexto de los procesos de despliegue. En particular ambos proporcionan información relativa a los recursos ofrecidos y requeridos por los elementos de modelado. Esto permite procesar automáticamente requisitos software en base a los recursos disponibles en el sistema sobre el cual se ejecuta el despliegue.

Tercero, se ha propuesto un proceso de despliegue adaptable mediante políticas. La descripción del proceso incluye la identificación de los agentes principales implicados en el ciclo de vida del despliegue, a saber los proveedores de software, los ejecutores de procesos de despliegue y los contextos de despliegue. La ejecución de los procesos se puede modular mediante la aplicación de políticas de despliegue. Las políticas guían la selección entre unidades válidas de despliegue (es decir unidades que satisfacen los recursos requeridos pero proporcionan conjuntos de recursos diferentes), ayudan a la resolución de conflictos entre unidades y determinan la distribución de unidades en base a ciertas estrategias. La descripción del proceso de despliegue identifica las actividades donde las políticas del despliegue son aplicables, se les ha llamado PEP (Policy Enforcement Points). Las políticas del despliegue hacen uso de la información que describe a las unidades de software y el contexto de despliegue para modificar los resultados de una acción de despliegue. Además, los medios para expresar y hacer cumplir políticas de despliegue para los casos de uso de selección de unidades y distribución.

Por último, una implementación de referencia de la arquitectura del despliegue se ha desarrollado y demostrado con casos de estudio. La implementación de la arquitectura ha proporcionado información valiosa para determinar la viabilidad de las contribuciones ya mencionadas. Los siguientes componentes se han creado para la implementación de referencia:

- *Gestor de Entorno*, este componente coordina acciones del despliegue dentro de un entorno de despliegue. El gestor de entorno descubre y mantiene el control de nodos en el dominio e interactúa con los gestores de nodo. Este componente es el punto de entrada para las operaciones del despliegue en un dominio. La entidad ejecutora de procesos de despliegue entra en contacto con este componente para emprender acciones del despliegue. En nuestra implementación de referencia este mismo componente se encarga de iniciar los procesos de despliegue, pero desde el punto de vista arquitectónico otro actor podría iniciar el proceso.
- *Gestor de Nodo*, este componente es un agente de gestión JMX (*Java Management Extensions*). Supervisa los recursos del nodo y ejecuta operaciones del despliegue en el nodo. Los servicios ofrecidos por este elemento se registran usando el protocolo DNS-SD (*DNS Service Discovery*).
- *Repositorio Activo*, proporciona operaciones de inserción, búsqueda, eliminación y actualización para productos de software y unidades de despliegue. Es capaz de crear conjuntos válidos (con dependencias resueltas) para productos y unidades y también de preparar los planes de despliegue sobre sistemas distribuidos. Además, ejecuta gestores de políticas que determinan la selección y distribución de unidades de despliegue.
- *Editores Gráficos*, para la creación y la edición de modelos de software y de sistemas distribuidos. Los editores facilitan la generación en formato XML de instancias de modelo. Estas herramientas son útiles para los proveedores de software y también para las entidades encargadas de ejecutar los procesos de despliegue.

1.5. Conclusiones

La complejidad creciente de los servicios y sistemas de información y comunicaciones ha exacerbado la necesidad de automatización de los procesos. Los sistemas modernos son creados mediante la combinación de múltiples componentes software en tiempo de ejecución. Adicionalmente, se actualizan, se extienden y se eliminan con frecuencia cada vez mayor. En consecuencia; los aspectos relacionados con el despliegue de software y servicios tienen un impacto innegable en los costes de operación de los sistemas de información y comunicaciones. A pesar de este hecho, los desafíos presentados por el despliegue de software han sido descuidados tradicionalmente por la ingeniería. Al menos, si lo comparamos con la atención prestada a otras actividades de *tiempo de desarrollo* tales como los procesos de análisis de requisitos, diseño, codificación, o pruebas. Muchos factores complican la tarea de automatización del despliegue de software. Las unidades del despliegue exhiben patrones complejos de dependencias que deben ser entendidos y solucionados. Las dependencias abarcan aspectos asociados a niveles distintos, pero interrelacionados; a saber el software, hardware y los elementos de red. Todos estos aspectos se deben satisfacer para realizar una acción correcta de despliegue.

Esta tesis contribuye al área de despliegue de software con una arquitectura de despliegue que soporta la adaptación al contexto y ejecución de procesos de despliegue sobre sistemas distribuidos. Un sistema de despliegue se adapta al contexto si es capaz de modificar su operación en función del estado actual del sistema objeto del proceso de despliegue, en cuanto a sus recursos disponibles (hardware, software y red). Las actividades de despliegue se pueden iniciar desde ubicaciones remotas y las unidades del despliegue se pueden extraer de múltiples depósitos en la red. Además, se ha intentado manejar un aspecto adicional de distribución; distribución sobre el dominio de despliegue. Lo cual implica que se puede estar tratando un sistema distribuido, de nodos interconectados. Eventualmente, esto implica que las unidades de despliegue que colaboran para proporcionar cierto servicio pueden ser asignadas a varios nodos en tiempo de despliegue.

Los objetivos detallados de este trabajo de investigación son los siguientes:

1. *Desarrollar un modelo para expresar y ejecutar políticas del despliegue.* Por lo tanto, permitiendo que las tareas de despliegue sean adaptables mediante políticas.

Este objetivo tiene que ver con la carencia de soporte en sistemas actuales para combinar los intereses múltiples y potencialmente conflictivos de cada uno de los actores implicados en el mercado de aprovisionamiento de servicios, entre otros los proveedores de servicios, los proveedores de contenidos, los usuarios finales y los operadores de red.

2. *Proponer un modelo para describir y manejar formalmente contextos distribuidos y heterogéneos.*

Uno de los desafíos en la automatización del despliegue de software radica en la adaptación al contexto de la ejecución del proceso. En último extremo es necesario resolver un problema de dependencias entre elementos a distintos niveles: gestión de dependencias entre componentes o servicios software (con la complejidad añadida de que existen varias versiones de los elementos), de requisitos sobre el hardware y de la

topología de la red sobre la que se ejecuta el proceso (que en el peor de los casos puede ser un sistema distribuido).

Un modelo para describir formalmente el contexto de despliegue ayuda a proporcionar automatización a los procesos de resolución de dependencias

3. Incorporar los dos modelos ya mencionados en la arquitectura de despliegue.

La capacidad de distribuir funcionalidades del despliegue entre los actores implicados en procesos de despliegue ayuda a solucionar tareas complejas. Los principios de orientación a servicios (SOA) e ingeniería basada en componentes (CBSE) fueron aplicados en la definición de la arquitectura. Una vez más el estándar de despliegue de OMG fue una referencia valiosa, aun cuando el OMG D&C en sí mismo no se puede considerar orientado a servicios.

4. Validar la viabilidad de la estrategia de investigación mediante el desarrollo y prueba de una implementación de referencia de la arquitectura de despliegue.

Una implementación de referencia de la arquitectura de despliegue ha sido desarrollada. La implementación de referencia proporciona detalles más ricos de los elementos principales incluidos en la arquitectura. La realimentación obtenida con experimentos de uso de la implementación de referencia fue útil para mejorar la implementación y la propia arquitectura en un proceso iterativo.

2. Originalidad

La tesis propone una arquitectura de despliegue dirigida por políticas, y basada en modelos que pueda adaptarse a situaciones de despliegue distribuido y heterogéneo. Además de la especificación de la arquitectura una implementación de referencia de los elementos principales incluidos en la arquitectura se ha desarrollado y se ha validado con casos de estudio.

El hecho de que la arquitectura de despliegue sea adaptable mediante políticas tiene como finalidad proporcionar un mejor soporte para la automatización de las tareas del despliegue. La explicación radica en que el uso de políticas permite representar las necesidades y requisitos de los múltiples actores implicados en un escenario de aprovisionamiento de servicios, entre otros los proveedores de contenidos, servicios y aplicaciones, los operadores de red y los usuarios finales.

Se ha dicho ya que la arquitectura de despliegue está basada en modelos. Antes de explicar las ventajas de usar modelos es conveniente clarificar el término modelo. [IEEE610.12 90] define un modelo como *una aproximación, representación, o idealización de ciertos aspectos de la estructura, comportamiento, operación o de otras características de un proceso, concepto o sistema del mundo real*. Análogamente, el OMG define modelo como *una descripción o especificación de un sistema y su entorno con un propósito determinado. Un modelo se presenta a menudo como una combinación de dibujos y texto. El texto puede estar en un lenguaje de modelado o en lenguaje natural*.

Una estrategia basada en modelos permite la formalización y conceptualización de todos los aspectos asociados a los sistemas y herramientas de despliegue. La posibilidad de que los modelos que representan las unidades de despliegue y los objetivos de los procesos (redes de nodos interconectados) sean interpretables por máquinas facilita el proceso automatizado, la integración entre herramientas, las pruebas de sistema, su mantenimiento y su evolución en el tiempo. Los modelos propuestos en la tesis se basan en estándares de la OMG y DMTF, pero los amplía para mejorar el tratamiento de aspectos tales como la adaptación a contextos heterogéneos y los problemas relacionados con la gestión de versiones de software en tiempo de despliegue. Al hablar de contextos heterogéneos se hace referencia al hecho de que las estructuras objeto de los procesos de despliegue puedan estar compuestas por redes de dispositivos con diferentes características y prestaciones. Será natural por tanto el que tengamos que enfrentarnos a situaciones en las que podemos enfrentarnos a una gran variabilidad en cuanto a los recursos disponibles, en términos de software, hardware y red.

3. Resultados

3.1. Publicaciones

En el desarrollo de la tesis doctoral se han generado más de diez publicaciones en revistas y congresos internacionales:

Apache and Eclipse: Comparing Open Source Project Incubators. Juan C. Dueñas, Hugo A. Parada, Félix Cuadrado, Manuel Santillán and José L. Ruiz. IEEE Software magazine. November/December 2007 (Vol. 24, No. 6) pp. 90-98 ISSN: 0740-7459.

A Service Component Deployment Architecture for e-Banking. José L. Ruiz, Juan C. Dueñas and Félix Cuadrado. Third International IEEE Workshop on Service Oriented Architectures in Converging Networked Environments (SOCNE 08), Okinawa, Japan. To appear in March 2008.

An Open Source Platform for the Integration of Distributed Services. Félix Cuadrado, Juan C. Dueñas, José L. Ruiz, Jesús Bermejo and Miguel García. Third International IEEE Workshop on Service Oriented Architectures in Converging Networked Environments (SOCNE 08), Okinawa, Japan. To appear in March 2008.

JMX Redux. Juan C. Dueñas, Manuel Santillán and José L. Ruiz. Dr. Dobbs Journal. July 2005. Pages 50-57.

An End-to-End Service Provisioning Scenario for the Residential Environment. Juan C. Dueñas, José L. Ruiz and Manuel Santillán. IEEE Communications Magazine. September 2005, Vol. 43, No. 9. Pages 94-100

Open Platforms for service provision. J. C. Dueñas, J. L. Ruiz, J. Bermejo, J. A. Alonso, C. Acuña, C. Díaz, M. González, M. García, A. Alvarez. European Wireless 2005. 1st Workshop on Wireless Vehicular Communications and Services for Breakdown Support and Car Maintenance (W-CarsCare05). April 2005, Nicosia, Cyprus.

Experiencias en la utilización de middleware de código abierto para el aprovisionamiento de servicios extremo a extremo en el sector residencial. Jose L. Ruiz, Juan C. Dueñas, Manuel Santillán. V Jornadas de Ingeniería Telemática JITEL 2005. Septiembre 2005, Vigo, España.

Deployment in dynamic environments. Jose L. Ruiz, Juan C. Dueñas, Fernando Usero and Cristina Díaz. DECOR, 1st francophone conference on deployment and (re)configuration. ISBN 2-7261-1276-5. Grenoble, France. October, 2004.

OSGi Service Platform in Integrated Management Environments, OSGi Alliance World Congress. Barcelona, Spain, October 11-15, 2004. C. Ruiz, C. Acuña, J. Bermejo, J. C. Dueñas, J. L. Ruiz.

A Framework for Resolution of Deployment Dependencies in Java-Enabled Service Gateways. Third International Workshop on Scientific Engineering of Distributed Java Applications. LNCS 2952. Springer Verlag, ISSN 0302-9743. ISBN: 3-540-21091-1. Pg 1-12. Luxembourg, November 2003.

3.2. Contribuciones a proyectos de I+D+i

Las contribuciones de la tesis doctoral se han visto reflejadas en los siguientes proyectos de ámbito europeo y nacional:

- EUREKA ITEA Osmose (2003-2005) Open Source Middleware for Open systems in Europe, official web site at: <http://itea-osmose.org>
- EUREKA ITEA Osiris (2005-2008), Open Source Infrastructure for Runtime Integration of Services, official web site at <http://www.itea-osiris.org>
- CENIT ITECBAN (2006- 2009)

3.3. Proyectos Fin de Carrera

Durante el desarrollo de la tesis doctoral he sido tutor de seis proyectos fin de carrera, dos de ellos premiados por el COIT:

- *Análisis y desarrollo de un sistema distribuido de telemando y telecontrol con tecnología de componentes de software y servicios Web,* Carmen África Lahera (2004).
- *Desarrollo de un sistema de control de intrusos para pasarelas domésticas,* Rafael Barriuso (2004). Premio COIT 2005 al mejor PFC en seguridad.
- *Desarrollo de una herramienta de gestión remota de pasarelas domésticas,* Manuel Santillán (2004)
- *Desarrollo de un servicio de acceso y presentación de contenidos multimedia personales e interactivos,* Bonifacio García (2005). Premio Cátedra Telefónica UPM (2005) a la mejor aplicación de la Internet de Nueva Generación y Premio COIT al mejor PFC en aplicaciones multimedia (2005).
- *Desarrollo de un sistema de gestión de perfiles de usuario para la personalización en el acceso de contenidos digitales,* Carolina Antón (2005).
- *Desarrollo de un sistema de despliegue optimizado de servicios sobre plataformas distribuidas,* Sergio Tur (2007).

3.4. Contribuciones a Comunidades de Código Abierto

Algunos de los resultados de la tesis doctoral han sido donados como contribuciones a comunidades de código libre, entre las que destacan:

- Apache Foundation. Se ha contribuido parte del código de la implementación de la arquitectura al proyecto Apache Felix <http://felix.apache.org>
- OS4OS (Open Source for Open Systems). Parte del código de la implementación de la arquitectura ha dado lugar a la creación del proyecto JBones <http://jbones.forge.os4os.org>

4. Aplicabilidad

En todo momento el objeto de la tesis doctoral ha tenido una orientación práctica en lo que concierne a la resolución de un problema, el despliegue de software y servicios, que como principal consecuencia tiene como resultado una mejora de la productividad de los ciclos de desarrollo y una reducción de costes de mantenimiento, gracias a la reducción de pasos manuales de los procesos. Inicialmente el primer punto de aplicación de los resultados de la tesis doctoral estuvieron ligados al mercado residencial, lo cual fue canalizado a través de la participación del autor en los proyectos ITEA Osmose e ITEA Osiris, en los cuales participaron importantes empresas españolas como Telefónica o Telvent.

En lo que concierne al despliegue de servicios en el mercado de banca la aplicabilidad del trabajo ha estado ligada a en su etapa final al proyecto de I+D nacional CENIT ITECBAN que tiene como principal objetivo la creación de un nuevo núcleo bancario. Actualmente desarrollo mi carrera profesional trabajando para Indra y en colaboración con Caja Madrid. Es destacable que como resultado de dicha colaboración parte de los resultados de la tesis estén siendo trasladados a través de ITECBAN al mercado bancario. Los bancos, al igual que otras muchas organizaciones con un largo recorrido, mantienen infraestructuras en las que conviven sistemas que llevan funcionando décadas, típicamente con arquitecturas *Host*, con nuevas aplicaciones y servicios desarrollados con tecnologías JEE y .Net. Por otro lado típicamente las aplicaciones y servicios atraviesan un circuito de promoción entre diversos entornos, hasta que alcanzan un cierto grado de madurez para poder ser llevados a entornos de producción. La heterogeneidad y diversidad de los entornos complican la gestión de los procesos de despliegue. A todo ello debemos añadirle el hecho de que los ciclos de producción son cada vez más cortos, aún más con filosofías orientadas a servicios. Por todas estas razones, el sector bancario es una de las vías claras de aplicabilidad de esta tesis doctoral. La disponibilidad de una arquitectura de despliegue que permita controlar y automatizar los procesos de despliegue es una gran ventaja en cuanto a costes y permite mantener bajo control arquitecturas de servicios, típicamente distribuidas, en un contexto de heterogeneidad.