



Comunicación

105

UNA EXPERIENCIA PRÁCTICA DE INTEGRACIÓN DE SISTEMAS HETEROGÉNEOS DIRIGIDA POR MODELOS

Jose Luis Mendiola

Gerente de Informática de Diputación
IZFE, S.A. (Diputación Foral de Gipuzkoa)

Beatriz Alustiza

Analista
IZFE, S.A. (Diputación Foral de Gipuzkoa)

Antonio Estévez

Open Canarias S.L.

Marko Txopitea

Open Norte S.L.

Palabras clave

MDA, Integración, J2EE, generación de código, modelos, metamodelos.

Resumen de su Comunicación

La construcción de aplicaciones bajo nuevos estándares y la integración de sistemas heterogéneos es una tarea compleja. En este trabajo presentamos la estrategia seguida para la integración de un framework basado en Struts y J2EE, el sistema transaccional CICS y el gestor de contenidos FileNet. El objetivo principal del proyecto ha sido redefinir la metodología de trabajo de los desarrolladores para mejorar su productividad. Siguiendo estrategias fundamentadas en el desarrollo basado en los modelos, y concretamente en MDA, se ha logrado desarrollar un framework único para los tres entornos. Se obtuvieron metamodelos independientes de cada uno de ellos para finalmente obtener un metamodelo unificado, flexible y abierto. El desarrollador aumenta su productividad ya que se abstrae de detalles de implementación de cada entorno, centrándose en obtener un modelo de negocio que represente su nuevo sistema, mediante UML, a partir del cual generar el código necesario.

UNA EXPERIENCIA PRÁCTICA DE INTEGRACIÓN DE SISTEMAS HETEROGÉNEOS DIRIGIDA POR MODELOS

1. Introducción.

Los grandes entornos corporativos disponen de frameworks y herramientas heterogéneas para la gestión de los sistemas que poseen. La mayoría de estos entornos necesitan integrar diferentes arquitecturas, tecnologías y sistemas de información. En la actualidad, existen escasas soluciones consolidadas para resolver a costes razonables este tipo de problemas, que además de ser propietarias, dependen de las tecnologías empleadas.

Las últimas tendencias proponen seguir estrategias basadas en el desarrollo de software dirigido por modelos (DSDM [2]), y más concretamente MDA [1] como posible solución a gran parte de los problemas existentes. Bajo la estrategia MDA las empresas consiguen que sus modelos de negocio sean válidos independizándolos de los frecuentes cambios tecnológicos.

En este trabajo se describe una metodología general para la integración de sistemas complejos, basándose en los fundamentos del DSDM y aplicándolo a un entorno real corporativo. El principal objetivo de este proyecto ha sido implantar un sistema de alta productividad para desarrollar aplicaciones J2EE [3] que interoperan en sistemas transaccionales como CICS y con gestores de contenidos como FileNet [4].

En el apartado 2, se describe el entorno tecnológico con las diferentes plataformas a integrar. A continuación se desarrollarán los tres casos prácticos para las plataformas seleccionadas. Los resultados de la aplicación de la metodología, las conclusiones y los trabajos futuros cierran este trabajo.

2. Plataformas objeto del proyecto

La Sociedad Foral de Servicios Informáticos de la Diputación Foral de Gipuzkoa (IZFE) gestiona para la Diputación Foral y los Ayuntamientos de Gipuzkoa un parque de máquinas y servidores compuesto por un mainframe IBM y más de 130 servidores Windows, Unix y GNU/Linux. IZFE acomete más de 90 nuevos desarrollos al año y actualmente posee más de 300 sistemas vigentes y en permanente evolución, correspondientes a ámbitos tan diversos como Hacienda, Carreteras, Cultura, Juventud, Servicios sociales, Emergencias, Montes e Innovación. El número de personas dedicadas directamente a estos proyectos de desarrollo es 165, además de colaboraciones cerradas de diferentes proveedores.

En cuanto a las tecnologías utilizadas por IZFE relacionadas con este proyecto, se puede destacar el servidor de aplicaciones Websphere Application Server for z/OS, versión 5.1. La base de datos relacional utilizada es DB2 Server for z/OS, versión 7.1.0.

Entre las tecnologías que se desean integrar en el proyecto, se encuentra el gestor documental FileNet versión 3.0, el servidor transaccional CICS Transaction Server for z/OS, versión 2.3 y el framework IZFE basado en Struts que permite el desarrollo de aplicaciones J2EE.

Este conjunto variado y complejo de plataformas conforman el escenario ideal para el desarrollo de este proyecto y poder aplicar la metodología y las diferentes estrategias de integración para el desarrollo de un software eficiente.

3. Metodología seguida

Dada la variedad y la complejidad de los entornos con los que se trabajaba se decidió utilizar metodología bottom-up, comenzando por aspectos más específicos para después generalizar y poder abstraer los aspectos comunes. Para ello, se planificaron una serie de tareas repetibles diferenciadas por las tecnologías sujetas a estudio. Al final de estas tareas, se procedió a la integración de las diferentes tecnologías, para acoplarse en un modelo de utilización común e integrado de todas ellas. La figura 1 indica la secuencia de estas tareas así como las actividades que se realizaron en cada una de ellas.

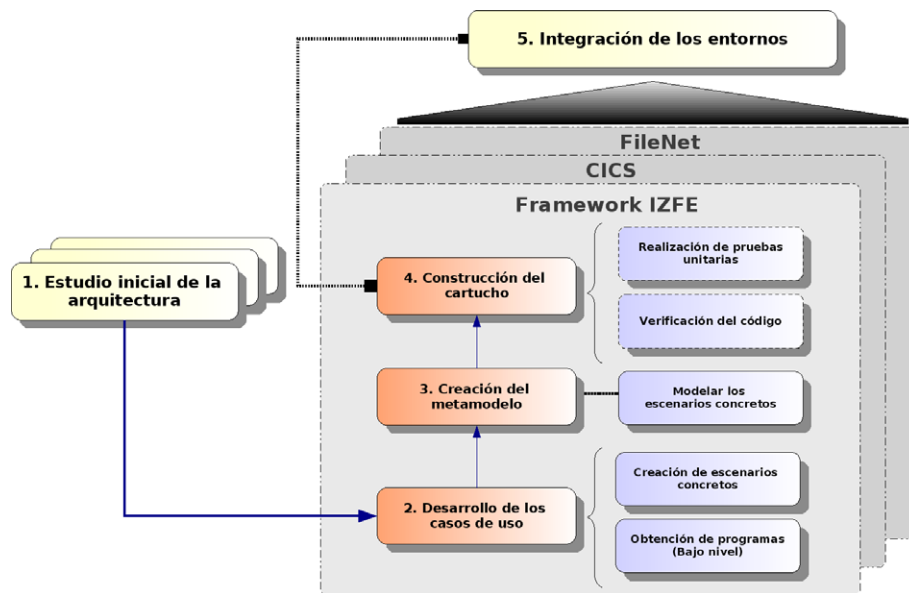


Figura 1. Disposición de las tareas en la metodología seguida

Tarea 1: Estudio inicial de la tecnología y de la arquitectura.

En esta etapa se obtiene y se estudia toda la información disponible. Fundamentalmente toda la información relativa al entorno tecnológico sujeto a estudio. Dicha documentación es revisada, validada y desarrollada. En la medida de lo posible, se intenta la duplicación de un entorno que simule al entorno real.

Tarea 2: Desarrollo de los casos de uso.

En esta etapa se obtienen diferentes aplicaciones (código) para su análisis y definición de la funcionalidad requerida. Como resultado, se deben desarrollar una serie de escenarios concretos que cumpla con la mayor parte de los requisitos funcionales analizados. Se requieren programas de amplio espectro sobre la tecnología implantada. Dichas aplicaciones deben cubrir la mayor parte de las necesidades intrínsecas de IZFE, cubriendo las tareas más comunes a realizar por los mismos. Se analizan y verifican los programas en los entornos de prueba (en aquellas tecnologías que existan). Estos programas se abstraen y/o se recodifican para facilitar la creación del metamodelo, identificando estructuras y componentes susceptibles a ser abstraídos, conceptos generales que posteriormente puedan ser modelados mediante la creación de plantillas. Unas plantillas que, posteriormente, se utilizarán para la generación automática del código, en fases subsiguientes.

Tarea 3: Creación del Metamodelo.

El alineamiento del proyecto con DSDM y en particular con MDA gira en torno al concepto de los Perfiles UML (5), un mecanismo de especialización definido como parte del propio UML. Los perfiles facilitan el modelado de aspectos específicos de un sistema de software. El principio básico para la obtención de cada uno de los perfiles es indagar generalizaciones entre diferentes lenguajes de programación, plataformas, tecnologías; además de incorporar otros aspectos relevantes relacionados con la integración con sistemas y aplicaciones heredadas.

Tarea 4: Construcción del cartucho.

Definidas las funcionalidades y el metamodelo se comienza con la construcción del cartucho, cuya función es la de dirigir la generación, compilación y empaquetado del modelo exportado en XMI (6). En esencia, un cartucho reúne la implementación de los Perfiles UML en el contexto de la plataforma y lenguaje de programación sobre el que se generará el código, que en este caso será Java. Dicho cartucho contendrá un descriptor donde se definen los perfiles con cada uno de sus estereotipos, asignándoles las plantillas correspondientes. La nueva aplicación se verifica en el entorno tecnológico de IZFE.

Estas 4 tareas se han desarrollado para los entornos del framework IZFE, CICS y FileNet que a continuación detallamos, y que en cada caso se han de particularizar al sistema en cuestión.

4. Aplicación de la metodología

El desarrollo del proyecto se secuenció entre los diferentes entornos tecnológicos. Se comenzó por el framework de IZFE, ya que se identificó como factor clave para el éxito, así como por su elevada complejidad. El framework IZFE utiliza la plataforma J2EE que se ejecuta sobre un WebSphere Application Server. Actualmente se utiliza para el desarrollo de aplicaciones web corporativas. El segundo entorno que se abordó fue el gestor transaccional, identificado como entorno CICS. En este entorno existen procesos y lógica de negocio heredadas a nivel corporativo con un alto valor estratégico. Como tercer entorno se abordó el desarrollo de un gestor documental propietario, que se denominó entorno FileNet. En este entorno se almacenan gran cantidad de contenidos de alta criticidad, utilizados en varios ámbitos de IZFE.

En cada entorno se aplicaron las tareas definidas en la metodología previamente explicada. En la última fase del proyecto se dedicó un esfuerzo importante a la integración de todos los entornos en un metamodelo común e integrado.

4.1. Framework IZFE

Tarea 1. Estudio de la arquitectura.

El framework IZFE se utiliza para la creación de aplicaciones J2EE en un entorno corporativo. Básicamente es una bifurcación (fork) del framework Struts en su versión 1.1. Dicho framework está dividido en una serie de subsistemas, entre los que destaca el subsistema de escucha (listener), el de control, el de presentación, el de negocio y el subsistema de seguridad especializado a un entorno corporativo.

Tarea 2. Desarrollo de los casos de uso.

Se seleccionaron dos aplicaciones utilizadas para la propia administración del framework. Estas aplicaciones se testearon y se pusieron en explotación en los entornos de simulación. Teniendo dichas aplicaciones como referencia, se definieron los requisitos de una nueva aplicación y se utilizaron técnicas de reingeniería para

su implementación. Durante esta fase se identificaron componentes unitarios del framework, susceptibles a ser incluidos como componentes parametrizables en el metamodelo.

Tarea 3. Creación del metamodelo.

El objetivo del metamodelo es la creación de un sistema con una arquitectura simplificada, que cumpla los requerimientos del framework IZFE. Para lograr esta simplificación, se mapearon los componentes que ofrecen el framework a un modelo más orientado al patrón MVC (7), en el que los dominios estén claramente definidos y enfocados a las funcionalidades de aplicaciones web autocontenidas. Con esta simplificación se consigue una disminución de elementos que el arquitecto MDA debe dominar para la creación de una aplicación más integrada. También permite una mejor distribución del trabajo a desempeñar en áreas especializadas, dividiendo el conocimiento entre las diferentes personas que conforman el equipo. Se definieron los siguientes dominios y/o capas: dominio de inicialización, presentación, lógica de negocio y el dominio de persistencia.

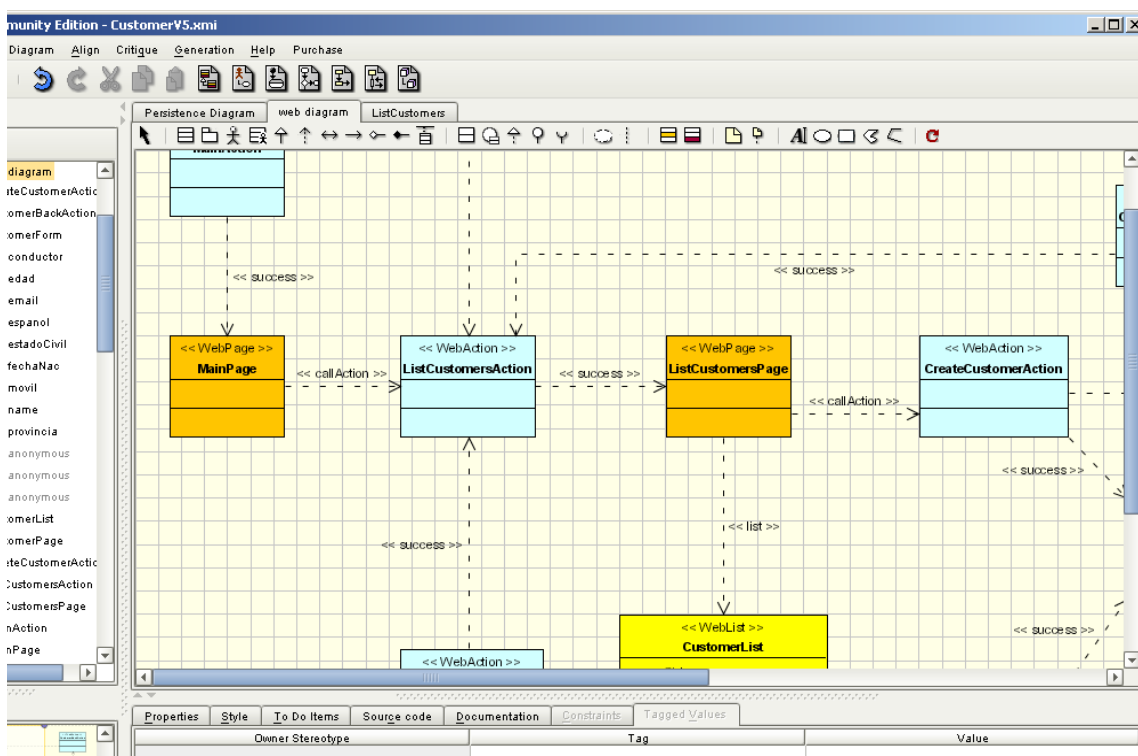


Figura 2. Ejemplo de un modelo concreto en el dominio de presentación

Tarea 4. Construcción del cartucho.

En esta etapa se propuso como objetivo la generación del 100% del código. Esto aumentó considerablemente la complejidad del problema, sobre todo en la definición de la lógica de negocio. Para este objetivo se usaron los diagramas de estado, incorporando en dichos estados semántica de acciones, tal y como se describe en la especificación 1.5 del UML. Para lograr esta aproximación se utilizó el Action Specification Language (ASL) (9), y con algunas modificaciones se desarrolló una gramática sobre la que se generó un parser, utilizando el compilador de compiladores SableCC (10). De esta forma se obtuvo un cartucho que generaba aplicaciones de forma completamente automatizada. Ahora IZFE en lugar de programar directamente estas interfaces utiliza los metamodelos definidos en UML para representar gráficamente

sus necesidades. El sistema, por tanto, es capaz de generar automáticamente el código a partir de estos diagramas.

4.2. Entorno CICS

Tarea 1. Estudio de la arquitectura.

El objetivo que se planteaba en este entorno era la ejecución de programas complejos bajo CICS a través de componentes J2EE. Se realizó un estudio exhaustivo en este ámbito, debido a la inexistencia de desarrollos previos con los requisitos planteados. Se identificaron dos problemas claves a resolver; la comunicación con el EIS y el formateo de tipos entre dominios.

Tarea 2. Desarrollo de los casos de uso.

Para la resolución del problema de comunicación se ha hecho uso de CICS ECI Resource Adapter, proporcionado a través de CTG (CICS Transaction Gateway, de IBM). El segundo problema identificado en la tarea anterior se resolvió utilizando la librería JRIO [13]. Finalmente se obtuvieron las funcionalidades mínimas exigidas y se realizaron pruebas unitarias para validar la solución escogida.

Tarea 3. Creación del metamodelo.

El metamodelo se desarrolló identificando componentes funcionales a ser generalizados, y parametrizando la información mínima a ser aportada a los modelos para su correcta generación posterior. Todo esto se realizó tomando como referencia los casos de usos elaborados en la etapa anterior.

Tarea 4. Construcción del cartucho.

Finalmente se implementó el cartucho, gracias al cual se genera de forma automatizada el código en el contexto de la plataforma, incluido con un pequeño test unitario. Dicho cartucho contiene un descriptor donde se definieron los perfiles con cada uno de sus estereotipos, asignándoles las plantillas correspondientes. Posteriormente se validaron los sistemas generados. De este modo, y utilizando el motor de generación, IZFE es capaz de describir el modelo gráficamente a través de diagramas sencillos de UML, y generar en una pasada el 100% de código necesario para su conexión con el gestor transaccional.

4.3. Framework FileNet

Tarea 1. Estudio de la arquitectura.

FileNet es un gestor documental propietario con funciones añadidas de workflow y con un framework propietario basado en Struts. Dispone de una API para Java que permite acceder a prácticamente todas sus funcionalidades. A su vez, IZFE ha desarrollado y mantiene una pequeña API simplificada que facilita la gestión de contenidos según los usos internos de la organización.

Tarea 2. Desarrollo de los casos de uso.

Se seleccionaron dos extractos de dos aplicaciones diferentes en las que se hacía uso de la API de IZFE. Basándose en los ejemplos facilitados y utilizando reingeniería inversa se extrajo las funcionalidades comunes en escenarios concretos. Finalmente se realizaron una serie de pruebas unitarias en el entorno de IZFE.

Tarea 3. Creación del metamodelo.

Partiendo de los escenarios definidos en la tarea anterior, se procedió a la construcción del correspondiente metamodelo, incluyendo los estereotipos y los valores etiquetados. El metamodelo fue generalizado para independizarlo del gestor de contenido concreto.

Tarea 4. Construcción del cartucho.

Para la construcción del cartucho se mapearon cada uno de los estereotipos a unidades de generación. Por cada unidad de generación fue definida una plantilla, la cual permitía al motor de generación la creación del código. La utilización del cartucho permite, a través de la definición de diagramas UML, la generación del 100% de código de acceso al contenido de los recursos definidos en el correspondiente gestor documental.

4.4. Unificación de metamodelos.

La última etapa definida en el apartado 3 se corresponde con la integración de los diferentes entornos. Para ello se plantea una integración a nivel de aplicativo web con el framework IZFE.

Inicialmente se plantea la integración con el CICS. En un principio se obtuvo un metamodelo y un cartucho para el modelado independiente de un programa concreto al que se le pasaban unos parámetros de entrada y se obtenían unos parámetros de salida. Posteriormente se pasó a incorporar esta funcionalidad como un elemento más, integrado dentro del metamodelo utilizado en el diseño de los modelos que se generen para el framework IZFE.

De esta forma, se acometieron dos tipos de integración en el metamodelo del framework IZFE. Una integración en el dominio de presentación, la cual permite la invocación de programa CICS utilizando un formulario web. Y una segunda integración en el dominio de lógica de negocio. Esta última de integración se realizó utilizando diagramas de estado.

La integración con FileNet se planteó de forma similar a la de CICS. Una vez creado el metamodelo y cartucho que se puede utilizar de forma independiente y aislada, se acometió la integración con el metamodelo del framework IZFE. La cual implicaba la ampliación, tanto en el dominio de persistencia (persistencia de recursos), como en el dominio de presentación. Para ello se definió un entorno de mantenimiento y gestión de recursos generalista, el cual permitiera de forma sencilla la creación, modificación, borrado y búsqueda de recursos tipo FileNet. Se utilizó una estrategia de dos pasos para resolver el problema. En un primer paso se genera el sistema completo utilizando un modelo tradicional para el framework IZFE. Posteriormente se utiliza el sistema generado para crear las plantillas, con un mayor nivel abstracción, la cual permitió la definición de una serie de estereotipos que simplificaron aún más la definición de modelos de integración con FileNet. Esta estrategia tuvo éxito debido a la poca variabilidad en los requisitos iniciales de utilización de los recursos de FileNet.

5. Resultados obtenidos

Utilizando este sistema, un arquitecto puede elaborar una aplicación completa diseñando los modelos adecuados basados en UML con perfiles. Para ello no hace falta ser un experto en J2EE, ni en el framework IZFE, ni en CICS ni en FileNet, simplemente tener unas nociones básicas de estas tecnologías y en UML.

Realizando un correcto modelado, el motor en el que se basa el proyecto es capaz de generar toda la

estructura y código necesario para el despliegue y desarrollo de una aplicación completa en un servidor J2EE, en este caso un Websphere Application Server (WAS). Esta aplicación generada es totalmente compatible con el framework corporativo de IZFE, y podrá haber utilizado, en la capa de negocio, funciones albergadas en sistemas CICS, o en la capa de persistencia, recursos definidos en la base de datos documental FileNet. Todo ello sin tener que introducir una sola línea de código y sin necesidad de ser un experto en las tecnologías utilizadas, simplemente modelando de una forma correcta a través de diagramas UML.

Destacar, que en el apartado de persistencia del modelo de negocio en base de datos relacionales, el framework IZFE da libertad para que los programadores utilicen la solución que ellos estimen oportuna. Para la ejecución de este proyecto se propuso la utilización de Hibernate [11] como solución viable y eficiente para la persistencia de dichos objetos con cualquier base de datos que se tenga (en nuestro caso, DB2). Este componente fue añadido, y correspondientemente modelado e implementado dentro del cartucho adecuado.

Para la conexión con los EIS, en este caso CICS, se hizo uso de las librerías aportadas por JCA [12], de las cuales existen numerosas referencias, documentación y librerías actualizadas. Y para la conversión de datos Cobol a Java y viceversa, una funcionalidad crítica en este ámbito, se utilizaron las librerías y clases proporcionadas por JRIO.

También se ha modelado la persistencia de recursos a través del gestor de contenido FileNet. El metamodelo se ha abstraído lo suficiente como para generalizarlo a otros gestores de contenidos, únicamente bastaría con modificar el cartucho.

Por último, indicar que como resultado final de este proyecto se obtiene la integración de todos los metamodelos anteriormente nombrados en un metamodelo común y unificado. Con ello se logra la integración completa de los entornos especificados, a priori, completamente heterogéneos en un modelo unificado, eficiente y ordenado, que permite el desarrollo de nuevos sistemas. El desarrollador se encuentra con un framework basado en UML y con perfiles bien definidos en las tres plataformas, con el que desarrollará a un mayor nivel de abstracción independizándose de los aspectos tecnológicos.

6. Conclusiones

Este nuevo enfoque en la creación de sistemas representa un gran cambio en la forma de trabajo tradicional, que actualmente existe en los equipos de desarrollo de IZFE. Ello exige el planteamiento de una nueva metodología y forma de trabajo, así como la planificación de una gestión del cambio que permita una adaptación rápida al nuevo escenario y que suponga un mejor aprovechamiento de las ventajas de este nuevo entorno. Los modeladores de los nuevos sistemas deben poseer un mayor conocimiento, principalmente de UML, a la hora de abordar la abstracción y creación de dichos sistemas.

Aparte de las ventajas intrínsecas que se derivan de un sistema basado en una aproximación MDA, destacamos:

- La normalización de los sistemas mediante modelos UML.
- La integración de sistemas heterogéneos, ocultando las complejidades de cada una de las tecnologías en particular.
- Desarrollo de un sistema único basado en la Web.
- Aumento considerable en la calidad de los sistemas desarrollados, ya que el código generado ha sido exhaustivamente probado.
- Posibilita el desarrollo rápido de prototipos, para que se conviertan fácilmente en sistemas y aplicativos finales.

-
- Mejora la facilidad de la implementación de la persistencia de los modelos, independizándolos de las continuas evoluciones y cambios tecnológicos.

7. Futuros trabajos

Entre los futuros trabajos, podríamos señalar el mantenimiento adaptativo del cartucho generado, según vayan evolucionando los sistemas ya integrados (framework IZFE, CICS y FileNet), además de otras tecnologías. Se podrían generar diferentes cartuchos para otros lenguajes de programación distintos a Java (.NET, por ejemplo), la incorporación de otras herramientas (Spring) en el framework corporativo, o la interacción con nuevos sistemas diferentes a los ya integrados, que implicaría las correspondientes modificaciones en los cartuchos generados.

También se podría abarcar la posibilidad de la integración de la tecnología de portlets , como nuevo alcance en el dominio de la aplicación. Para ello, se recomendaría usar el portlet suministrado por Struts, para evitar problemas de compatibilidad con el controlador del framework IZFE.

Dentro del ámbito MDA, cabría destacar la puesta en marcha de nuevos motores para la generación del código. En este caso, se ha utilizado una metodología “traslacionista”, donde a partir de las plantillas, incluidas en el correspondiente cartucho, conseguimos trasladar del modelo diseñado a código. En este momento existen otras aproximaciones de tipo “elaboracionista”, como pueden ser las transformaciones de modelos basadas en QVT , con un amplio futuro dentro de la arquitectura MDA.

8. Referencias

- MDA - Model Driven Architecture. <http://www.omg.org/mda/>
- DSDM – Desarrollo de software dirigido por modelos. <http://www.mdsd.info/>
- J2EE. <http://java.sun.com/javaee/index.jsp>
- FileNet <http://www.filenet.com> P8 3.0.0 Documentation
- UML Unified Modelling Language <http://www.uml.org/>
- XML. <http://www.omg.org>
- Framework de Struts. <http://struts.apache.org/>
- Action Semantics Revised Final Submission. OMG document ad/01-08-04.SL
- The Action Specification Language Reference Manual. <http://www.kc.com>
- SableCC Parser generator. <http://sablecc.org>
- Hibernate: <http://www.hibernate.org>
- JCA: J2EE Connector Architecture. <http://java.sun.com/j2ee/connector/>
- JRIQ. <http://www-03.ibm.com/servers/eserver/zseries/software/java/jrio/overview.html>