

Ingeniería Web Dirigida por Modelos

Nora Koch

Web Engineering Group

Programming and Software Engineering Unit, Institute for Informatics

Ludwig-Maximilians-Universität München

e-Mail: kochn@pst.ifi.lmu.de

Oettingenstr. 67. 80538 München (Germany)

Santiago Meliá

Grupo de Investigación IWAD

Departamento de Lenguajes y Sistemas Informáticos

Universidad de Alicante

e-Mail: santi@dlsi.ua.es

C/ San Vicente S/N, 03690 Alicante (Spain)

Nathalie Moreno Vergara

Atenea System Modeling Group

Departamento de Lenguajes y Ciencias de la Computación

Universidad de Málaga, ETSI Informatica.

e-Mail: vergara@lcc.uma.es

Bulevar Louis Pasteur, 35. (29071) Malaga (Spain)

Vicente Pelechano Ferragud

OO-Method Research Group

Departamento de Sistemas Informáticos y Computación

Escuela Técnica Superior de Informática Aplicada

e-Mail: pele@dsic.upv.es

Cami de Vera, s/n 46022, Valencia (Spain)

Fernando Sánchez Figueroa
QUERCUS Software Engineering Group
Escuela Politécnica
Universidad de Extremadura
e-Mail: fernando@unex.es
Avda. Universidad s/n,
10071 – Cáceres (Spain)

Juan M. Vara Mesa
Kybele Research Group
Universidad Rey Juan Carlos
e- Mail: juanmanuel.vara@urjc.es
C/ Tulipán S/N (28933),
Móstoles - Madrid (Spain)

Ingeniería Web Dirigida por Modelos

Nora Koch, Santiago Meliá, Nathalie Moreno, Vicente Pelechano,
Fernando Sánchez-Figueroa, Juan M. Vara

Web Engineering Group (Ludwig-Maximilians-Universität München)
kochn@pst.ifi.lmu.de

IWAD Research Group (Universidad de Alicante)
santi@dlsi.ua.es

Atenea System Modeling Group (Universidad de Málaga)
vergara@lcc.uma.es

OO-Method Research Group (Universidad Politécnica de Valencia)
pele@dsic.upv.es

QUERCUS Software Engineering Group (Universidad de Extremadura)
fernando@unex.es

Kybele Research Group (Universidad Rey Juan Carlos)
juanmanuel.vara@urjc.es

Resumen: La continua aparición de nuevas plataformas y tecnologías de implementación ha provocado que en los últimos años las aplicaciones Web se asemejen cada vez más a complejas aplicaciones de escritorio. Por ello, el proceso de diseñar, implementar y adaptar sistemas para este dominio de aplicación se está convirtiendo en una ardua y laboriosa tarea. Así, en el contexto de la Ingeniería Web, los principios del Desarrollo Dirigido por Modelos (DDM) están siendo aplicados con éxito para abordar el diseño, evolución y adaptación de aplicaciones en este entorno. En este artículo, se muestra cómo la Ingeniería Web Dirigida por Modelos (IWDM) ha surgido reciente como una nueva disciplina que aplica los principios fundamentales del DDM/MDA para definir modelos, metamodelos y transformaciones que permitan resolver los problemas de interoperabilidad existentes entre las distintas propuestas metodológicas, y abordar la construcción de herramientas que soporten procesos de desarrollo basados en las iniciativas anteriormente citadas.

Palabras Clave: Ingeniería Web, Desarrollo Dirigido por Modelos, MDA, Metamodelos, Transformaciones de Modelos, Herramientas CASE, Interoperabilidad

1 Introducción

En los últimos siete años, la *Arquitectura Dirigida por Modelos* (Model-Driven Architecture o MDA) se ha aplicado a diversos dominios resultando que, en general, funciona mejor en aquellos ámbitos en los que predominan los requisitos funcionales, trabajan con modelos bien estructurados, existe una clara separación de conceptos y usan plataformas estándares. La Ingeniería Web (IW) ha resultado ser, en ese sentido, uno de esos dominios en los que la aplicación de MDA ha generado mayores beneficios. Esta iniciativa está permitiendo abordar con éxito problemas de interoperabilidad, de evolución de modelos y de adaptación o adopción de nuevas plataformas y tecnologías de implementación.

Ante los beneficios que promete MDA, muchos métodos de IW están reformulando sus propuestas originales para incorporar los principios de este paradigma a sus metodologías; aún cuando ello conlleva cambios muy significativos en sus notaciones, procesos de desarrollo y herramientas. En particular, algunos de estos enfoques han tenido que: (a) rediseñar sus lenguajes de modelado utilizando técnicas basadas en metamodelos; (b) reorganizar el conjunto de modelos original de manera que éstos sean independientes de la plataforma; (c) reformular sus procesos de desarrollo en términos de transformaciones y combinaciones de modelos, o; (d) incorporar y adoptar, a sus procesos de desarrollo, los estándares que sustentan la iniciativa de MDA: UML® (Unified Modeling Language), MOF (Meta-Object

Facility), XMI (XML Metadata Interchange), QVT (Query / View / Transformations), etc.¹

Sin embargo, a pesar de todos los cambios y retos enumerados, los beneficios que está proporcionando la adopción de DDM/MDA a la Ingeniería Web compensan el coste y el esfuerzo realizado. MDA ha brindado la oportunidad de incorporar técnicas más eficientes para el desarrollo de aplicaciones Web, acortando la distancia existente entre los modelos de alto nivel de abstracción y el bajo nivel de abstracción de sus implementaciones [8]. Estos cambios han originado una nueva disciplina dentro de la IW llamada Ingeniería Web Dirigida por Modelos (IWDM), que se centra principalmente en la interoperabilidad de las actuales metodologías para el desarrollo de aplicaciones Web. En concreto, MDWEnet es una nueva iniciativa surgida entre un grupo de investigadores europeos que trabajan en IWDM con el objetivo de fomentar la interoperabilidad entre las diferentes propuestas y herramientas, a fin de ofrecer mejores herramientas y métodos para la industria².

Por supuesto, la adopción de MDD/MDA no ha estado exenta de problemas y limitaciones. El objetivo de este trabajo es el de ofrecer una visión sobre el estado actual de MDA en el ámbito de la IW desde el punto de vista de algunos de los grupos que trabajan activamente en este dominio de aplicación. No sólo se enumerarán los esfuerzos y los resultados alcanzados, sino también los problemas, dificultades y debilidades encontradas en la aplicación práctica de MDA. Estas conclusiones han

¹ OMG, www.omg.org

² MDWEnet, <http://www.pst.ifi.lmu.de/projekte/mdwenet/>

sido recogidas en base a la experiencia recopilada por estos grupos de investigación sobre los proyectos en los que se ha aplicado MDA.

El documento está estructurado de la siguiente manera. La Sección 2 describe el papel que actualmente desempeñan los modelos y metamodelos en los métodos de IW como consecuencia directa de uno de los principales objetivos de MDA: la generación automática de código a partir de modelos. En este contexto, la Sección 3 ilustra la importancia que han adquirido las transformaciones entre modelos dentro del proceso de desarrollo de aplicaciones. Seguidamente, se describe en la Sección 4 algunas de las herramientas que soportan actualmente los principios de DDM/MDA en el dominio de la Web. La Sección 5 mostrará cómo la comunidad de IW está aplicando los conceptos y mecanismos proporcionados por MDA para resolver los conflictos de interoperabilidad entre las distintas propuestas metodológicas. Finalmente, la Sección 6 destacará los desafíos más importantes a los que debe hacer frente MDA para mejorar la eficiencia y utilización de esta iniciativa en el contexto de las aplicaciones Web.

2 Modelos y Metamodelos en el Dominio Web

El paradigma de MDA se basa en la construcción de modelos y en la transformación de estos modelos, que representan puntos de vistas independientes del aspecto computacional (CIM), independientes de la plataforma (PIM) o específicos para una plataforma (PSM). Durante la última década la comunidad de Ingeniería Web ha definido varios métodos para el modelado de aplicaciones en este entorno, que

principalmente se centran en la construcción de PIMs. Los métodos incluidos en el libro de Rossi y otros [19] tales como Hera, OOHDM, OOWS, UWE, WebML, WSDM, y métodos como MIDAS [3], OO-H y WebSA (para ambos ver [15]) proponen la construcción de tipos similares de modelos pero usando diferente notación para la representación gráfica de los modelos. La mayoría de ellos utilizan una notación propietaria; otros combinan el uso de UML con su propia notación, y solamente UWE y WebSA hacen uso de UML 2.0 para todos sus modelos. UWE usa UML siempre que sea posible y define para los conceptos específicos del dominio Web un perfil UML basándose en los mecanismos de extensión que el lenguaje provee. El resultado es un language específico para el dominio Web conforme con la notación de UML. Existe una clara tendencia en el uso de UML también por otros métodos, principalmente por las ventajas que proporciona respecto al uso de herramientas CASE y en el metamodelado para propuestas dirigidas por modelos.

La principal característica de las propuestas en el entorno Web es la separación de conceptos (*separation of concerns*) construyendo diferentes modelos para el contenido de aplicaciones Web, estructura de hipertexto, presentación, adaptividad, en particular personalización y dependencia del contexto, y aspectos relacionados con la arquitectura de la aplicación Web. Los diferentes modelos son integrados en base a mecanismos implementados en herramientas propietarias o por medio de la aplicación de reglas definidas con lenguajes de transformación de modelos. Además, las propuestas IWDM se apoyan en los principios de MDA creando CIM como los modelos de requisitos construidos con WebRE [6] y OOWS; construyendo modelos PIM para navegación, presentación y procesos de negocio (casi todos los métodos);

construyendo PIM para modelos de arquitectura (WEI y WebSA); y obteniendo modelos PSM o transformando en código para plataformas específicas como Java EE, Struts, Spring y .NET. El objetivo principal es posponer, lo más posible en el proceso de desarrollo, la creación de modelos que tienen en cuenta aspectos tecnológicos. La ventaja es poder reaccionar eficientemente y a bajo costo a cambios tecnológicos.

Las entidades usadas en todos los modelos específicos para el entorno Web y las relaciones entre las entidades también pueden ser representadas como un modelo – llamado metamodelo, que se necesita para la especificación de las transformaciones de modelos. El primer metamodelo en el área Web fue presentado para la propuesta UWE en la 3ra Conferencia Internacional de Ingeniería Web (ICWE 2003) [11]; a posteriori otros métodos definieron sus metamodelos; recientemente [17] propone un metamodelo de referencia común llamado Web Engineering Interoperability (WEI), que es descrito como un metamodelo basado en el estándar MOF. WEI también define extensiones *lightweight* de UML, es decir perfiles UML para la representación de la sintaxis concreta de cada uno de sus metamodelos.

3 Transformaciones de Modelos para Generar Aplicaciones Web

El framework MDA propone el uso de las transformaciones de modelos para su aplicación en un proceso de desarrollo trazable desde los modelos abstractos (CIM o PIM), pasando por los modelos dependientes de plataforma (PSM) y llegar hasta la implementación. De esta forma, muchas aproximaciones de ingeniería Web han definido las transformaciones para obtener parte o toda la implementación de una

aplicación Web. Una estrategia de dominio específico, como es el dominio de las aplicaciones Web, permite generar automáticamente partes significativas de la implementación y así reducir el esfuerzo de desarrollo [2]. En este sentido, algunas metodologías Web ya han tomado ventaja de ello desarrollando herramientas CASE comerciales (que se presentarán en la siguiente sección). Hasta ahora, estas herramientas CASE usaban técnicas de generación de código que permitían obtener aplicaciones Web a partir un conjunto reducido de modelos de diseño. Sin embargo, muchas metodologías Web están comenzando a usar las transformaciones de modelos para extender o implementar herramientas CASE y así obtener las ventajas que proporcionan los lenguajes de transformación.

A continuación, se presenta cómo los métodos Web han aplicado los dos tipos de transformaciones de modelos, es decir, transformaciones modelo-a-modelo y transformaciones modelo-a-texto, con el objetivo de generar aplicaciones Web.

3.1 Transformaciones Modelo-a-Modelo

Las transformaciones modelo-a-modelo se clasifican en dos tipos diferentes: (1) transformaciones verticales que convierten los modelos desde un nivel de abstracción superior a un nivel de abstracción inferior (p.e. PIM-a-PSM, PSM-a-Código), (2) transformaciones horizontales que son usadas para describir correspondencias entre modelos del mismo nivel de abstracción (PIM-a-PIM, PSM-a-PSM). En muchas ocasiones, las transformaciones verticales del dominio de las aplicaciones Web se han desarrollado usando lenguajes específicos de herramienta. Sin embargo, metodologías

Web recientes como UWE [10], OOWS [20], WebSA [15] y MIDAS [3] han formalizado todo o parte de sus procesos de desarrollo usando transformaciones modelo-a-modelo mediante lenguajes de transformación como QVT, ATL o AGG. En algunos casos, las transformaciones modelo-a-modelo son definidas como mecanismo de integración para introducir nuevos aspectos como estilos arquitectónicos (WebSA), requisitos de usuario (WebRE [6] y OOWS Requirements Analysis [20]) y medidas de calidad [4]. En otros casos, se han establecido para formalizar las correspondencias de proceso de desarrollo original (UWE [12]).

Por otro lado, las transformaciones horizontales han sido aplicadas en el dominio Web para mantener la consistencia en la especificación de modelos, comprobando que los distintos modelos no imponen requisitos contradictorios en los elementos que comparten [17]. Sin embargo, debido a lo novedoso de estos artefactos, existe una considerable falta de madurez en los estándares actuales y en las herramientas que las implementan. Así, algunas de las propiedades de las transformaciones modelo-a-modelo como reusabilidad, facilidad de uso y de mantenimiento, no han sido abordadas. De hecho, hasta la fecha, no se conocen herramientas comerciales que hagan uso de las transformaciones modelo-a-modelo para obtener aplicaciones Web.

3.2 1.2 Transformaciones Modelo-a-Texto

La comunidad de Ingeniería Web tiene una larga experiencia en las transformaciones modelo-a-texto o generación de código. Aproximaciones como WebML, OOH y OO-Method/OOWS llevan generando aplicaciones Web desde hace diez años. En algunos

casos, la generación de código ha sido realizada mediante el uso de lenguajes de propósito general (C++, XSLT, Java o Python) que no poseen la mantenibilidad y extensibilidad adecuadas para realizar transformaciones modelo-a-texto. Recientemente, el estándar de la OMG Model to Text RFP ha establecido las características apropiadas para los lenguajes modelo a texto (p.e. Round-Trip engineering) y algunas propuestas como Xpand y MOFScript han sido adaptadas a dicho estándar con más o menos éxito. Recientemente, OOWS y WebSA han usado Xpand y MOFScript en la implementación de sus generadores de código para beneficiarse así de sus propiedades.

4 Herramientas CASE para la IWDM

Esta sección proporciona una visión general de las soluciones que a día de hoy dan soporte CASE a las propuestas enmarcadas en la IWDM. Además, se hace un breve repaso de los principales problemas y retos a abordar en este ámbito durante los próximos años.

En un primer grupo incluimos aquellas herramientas pensadas para trabajar con lenguajes específicos de dominio (DSLs) definidos como una extensión a UML (perfiles UML). Probablemente la más destacada en este primer grupo sea ArgoUWE [10], resultado de extender la herramienta de libre distribución ArgoUML con capacidades para el modelado del contenido y las estructuras de navegación para aplicaciones Web. Más tarde se le añadieron nuevas funcionalidades para modelar también la lógica de negocio y el comportamiento en aplicaciones Web dirigidas por

el flujo de trabajo, así como para la detección de inconsistencias en modelos basados en el metamodelo de UWE, con respecto a las reglas OCL definidas sobre dicho metamodelo [10]. En la actualidad el principal inconveniente de ArgoUML, y por tanto de ArgoUWE, es que no soportan la versión 2.0 de UML. En este grupo de herramientas se incluye también WebTE [16], una herramienta UML que soporta el estándar XML. WebTE utiliza un motor de transformaciones para ejecutar las transformaciones de modelos definidas en OO-H y WebSA y, a partir de los modelos definidos en ambos métodos, generar una aplicación Web Java EE.

Aunque a día de hoy existen otras soluciones, como AriadneTool [5], el entorno de desarrollo DaVinci [13], MIDAS-CASE [21] o VisualWade [9], estas herramientas aún no son capaces de proporcionar toda la funcionalidad que se espera de ellas o, en el peor de los casos, son parte de proyectos que han sido abandonados.

En un segundo grupo englobamos todas aquellas herramientas pensadas para trabajar con lenguajes definidos por medio de metamodelos basados en MOF. Dado que no están basadas en el uso de UML, podemos decir que estas herramientas no son totalmente *MDA compliant*, puesto que no utilizan todos los estándares de la OMG para el desarrollo de aplicaciones dirigido por modelos. A día de hoy, la más representativa de estas herramientas es WebRatio [23], una herramienta comercial que soporta el método WebML. Con WebRatio los objetos de negocio se modelan usando notación UML o E/R, mientras que la interfaz se modela usando WebML. A partir de estos modelos se genera una aplicación Web para arquitecturas Java EE que se conecta a orígenes de datos SQL y/o XML. Existen propuestas más recientes basadas

en el uso de nuevos metamodelos MOF, como M2DAT [3] o el entorno OOWS Suite [7], pero todavía están en fase de desarrollo.

Cabe señalar que el soporte CASE para la IWDM sigue una tendencia generalizada en el DDM: la proliferación de herramientas y entornos para el desarrollo de nuevas herramientas ad-hoc que den soporte a las propuestas en el marco del DDM. Así, numerosas organizaciones se están planteando el desarrollo de su propia herramienta para dar soporte a su propio método de IWDM (siguiendo cualquiera de las propuestas para DDM: MDA, Factorías Software, Líneas de Producto, Programación Generativa, etc.). De este modo, la tecnología juega un papel clave a la hora de distinguir entre herramientas UML (basadas en el uso de UML) y aquellas que no lo son: las facilidades que se proporcionan en el contexto del *Eclipse Modelling Project* (EMP) y otros entornos para el desarrollo de DSLs, como el *Generic Modelling Environment* (GME) o las *DSL Tools*, han hecho que la atención se centre ahora en desarrollar herramientas basadas en la definición de nuevos metamodelos MOF, en lugar de herramientas basadas en UML. En este sentido, merece atención especial el proyecto EMP, por la cantidad y calidad de las facilidades para el DDM que proporciona, y que han dado lugar a la aparición de una nueva generación de herramientas Eclipse: un marco común, ideal para la interoperabilidad de herramientas y modelos como EMF, meta-editores como GMF, motores de transformación como ATL o VIATRA, generadores de código como MOFScript, etc. Como consecuencia, las propuestas más recientes del ámbito de la IWDM desarrollan sus herramientas como plug-ins de Eclipse, como el entorno OOWS Suite y M2DAT,

mientras que las que ya existían son actualizadas o redefinidas para encajar en la plataforma Eclipse, como WebRatio [1].

Aún así, todavía queda mucho trabajo por hacer. Un problema muy típico, planteado por Moreno y Vallecillo en [19], es el hecho de que las reglas de transformación se incluyan en el propio código de la herramienta (este es el caso de ArgoUWE y WebRatio), lo que genera un vacío entre el diseño de la aplicación y su implementación. De acuerdo a los principios de MDA, esas reglas deberían definirse con un mayor nivel de abstracción, utilizando para ello el estándar QVT. Aunque algunas propuestas ya han comenzado a solventar ese problema ([10] muestra cómo en UWE, [15] para WebSA y [17] para WEI), todavía tienen que incorporarse esas mejoras en las correspondientes herramientas. Además, la ausencia de una implementación de referencia para QVT (lo que ha llevado a la aparición de varias implementaciones parciales [17]) complica la tarea de integración. Otro problema recurrente es la cuestión de la interoperabilidad entre herramientas: los modelos elaborados con una herramienta no pueden utilizarse desde otra. El uso de un marco de modelado común como EMF ha ayudado a paliar el problema. Igualmente, la utilización de modelos de *weaving*, que empieza a ser una propuesta aceptada para automatizar la migración de modelos [22], podría extrapolarse como estrategia para resolver el problema de la interoperabilidad. Finalmente, el lector debe comprender que aunque el DDM es un paradigma de desarrollo consolidado y que goza de gran aceptación, la IWDM es relativamente nueva, como muestra el hecho de que todas las herramientas mencionadas en esta sección provienen del ámbito académico. Por lo tanto, podemos concluir que el reto más importante para los desarrolladores de

herramientas CASE que den soporte a la IWDM es ser capaces de atraer la atención de la industria.

5 Interoperabilidad en IWDM

Las actuales propuestas de IWDM proporcionan un conjunto de métodos y herramientas para un diseño y desarrollo sistemático de aplicaciones Web. Sin embargo, estas propuestas tienen algunas limitaciones, especialmente para intercambiar modelos o representar otros aspectos de modelado, tales como estilos arquitectónicos, independencia de la tecnología o distribución. Una posible solución a estos problemas es proporcionar interoperabilidad entre las propuestas de IWDM, de tal forma que se puedan complementar unas con otras. La interoperabilidad forma parte de la esencia de MDA a diferentes niveles: modelos y metamodelos, transformaciones y herramientas.

En relación a los modelos y metamodelos, existen tres posibilidades para conseguir esta interoperabilidad: a) obtener un método unificado basado en las fortalezas de los diferentes métodos, b) obtener puentes de interoperabilidad entre modelos individuales y sus correspondientes herramientas y c) obtener un metamodelo común. Estas posibilidades tienen cada una de ellas sus propias ventajas y desventajas. Actualmente hay dos trabajos relacionados con las opciones b) y c). En [24] Wimmer et al., presentan una metodología basada en MDA para hacer interoperables tres propuestas de modelado Web (WebML, OO-H y UWE). Los autores usan la implementación Ecore del estándar MOF para la definición de los tres metamodelos.

ATL se usa como lenguaje de transformación de modelos para implementar las reglas de transformación y el motor ATL ejecuta la transformación. Su siguiente meta es definir un metamodelo común para modelado Web. Otra propuesta interesante es WEI [17], un framework basado en modelos para construir aplicaciones Web que, entre otros objetivos, trata de proporcionar un framework común (y un metamodelo) en el que las actuales propuestas puedan ser integradas y formuladas en términos de MDA. WEI puede ser utilizado tanto para construir aplicaciones Web desde el principio como para construir aplicaciones Web basadas en modelos existentes (incluidos aquellos definidos usando otras metodologías como UWE, OO-H o WebML).

En relación a las transformaciones, existe un problema debido a que cada propuesta usa su propio lenguaje de script que resulta incompatible con otros lenguajes y herramientas que los usuarios normalmente utilizan. En este sentido, QVT no está siendo usado tal y como la OMG podía esperar.

En relación a la interoperabilidad de herramientas, a pesar de los esfuerzos de la OMG, el estándar XMI no es del todo satisfactorio, especialmente al trabajar con perfiles UML. Así pues, por el momento parece más conveniente aprovechar las transformaciones modelo-a-modelo para conseguir la interoperabilidad entre herramientas CASE, estén o no basadas en UML.

6 Desafíos Futuros

Los métodos de IWDM están evolucionando para adaptarse adecuadamente a los continuos avances tecnológicos y a los nuevos requisitos de las aplicaciones Web. En los últimos años han surgido un nuevo tipo de aplicaciones Web conocidas como Web 2.0. Estas aplicaciones permiten a las personas colaborar y compartir información de manera online. Tal y como comenta Murugesan en [19] la *Web Centrada en la Gente* y la *Web de Lectura/Escritura* ofrece interfaces de usuario inteligentes y facilidades incorporadas para que los usuarios generen y editen la información que se presenta en la Web y de ese modo enriquecer el contenido.

Las aplicaciones Orientadas a Servicios y la Web 2.0 están proporcionando una clara infraestructura para integrar multiples servicios software bajo una interfaz de usuario rica. Las aplicaciones con Clientes Ricos basados en AJAX (Javascript Asíncrono y XML) o RIA (acrónimo en inglés), los Mashups de Servicios, la arquitectura REST o los Servicios Web XML permiten integrar las aplicaciones Web actuales con servicios de terceros, portales y también sistemas legados. Las Aplicaciones de Internet Ricas (RIA) están cambiando los navegadores, pasando de un interfaz estático petición-respuesta a un interfaz dinámico asíncrono. Las RIA prometen unos beneficios y una experiencia de usuario más rica que afectan a los métodos de Ingeniería Web [18].

La amplia aceptación de los dispositivos móviles permite a los usuarios acceder a la Web usando este tipo de dispositivos de mano (pocket PC, PDA, smartphones, etc). Las aplicaciones Web Móviles ofrecen algunas características adicionales comparadas

con las aplicaciones Web tradicionales como servicios sensibles a la localización, capacidades de sensibilidad al contexto, y personalización. No nos olvidamos de que el desarrollo de aplicaciones Web es una tarea compleja que debe tener en cuenta muchos aspectos diversos y requisitos no funcionales como escalabilidad, fiabilidad, disponibilidad, evolución y mantenibilidad, usabilidad, seguridad, accesibilidad, movilidad, localización, personalización, adaptatividad, etc.

La Usabilidad de la Web es uno de los factores de calidad más relevantes que deben tener en cuenta los métodos de IWDM actuales y futuros. La usabilidad debería integrarse de forma obligatoria en cada una de las etapas de un método de desarrollo de software. Las aplicaciones Web generadas por los métodos de IWDM deberían satisfacer las expectativas y necesidades de los distintos tipos de usuarios (capacitados y discapacitados) con diferentes habilidades. La aparición continua de nuevas tecnologías de moda como las RIA y los dispositivos móviles no garantiza que las guías de diseño y las pruebas de usabilidad actuales sirvan para las nuevas aproximaciones con el propósito intención de garantizar una mayor experiencia de usuario.

MDA todavía sigue siendo una aproximación relativamente joven, existiendo algunos aspectos alrededor del desarrollo de software basado en MDA que no están suficientemente claros. En particular, MDA se concibió originalmente para desarrollos nuevos y no está claro cómo MDA maneja la producción de versiones, parches o actualizaciones, que constituyen una parte importante del mantenimiento de las aplicaciones Web. En lo que respecta a los requisitos no funcionales, MDA trata

bien las propiedades funcionales, pero es menos capaz de abordar los requisitos no funcionales. En este sentido, existe un debate sobre si la especificación de requisitos no funcionales está en el ámbito de MDA o no. En resumen, mientras los principios subyacentes a MDA son sólidos, todavía existen aspectos que deben resolverse para poder llevar MDA su máximo potencial, particularmente en el dominio de las aplicaciones Web.

Referencias

1. Acerbis, R., Bongio, A., Brambilla, M. and Butti, S., *WebRatio 5: An Eclipse-Based CASE Tool for Engineering Web Applications*, in Web Engineering, 2007, 501-505.
2. Bettin, J.: *Measuring the Potential of Domain-Specific Modeling Techniques*, in Proc. of the Second Domain-Specific Modeling Languages Workshop, OOPSLA, Working Papers W-334. Helsinki School of Economics, 2002, 39-44.
3. Cáceres, P., De Castro, V., Vara, J.M. and Marcos, E.: *Model Transformations for Hypertext Modelling on Web Information Systems*, in Proc. of the ACM/SAC 2006 Track on Model Transformations (MT2006), Dijon, France, 2006, 1256-1261.
4. Cachero C., Melia S., Genero M., Poels G., Calero C.: *Towards Improving the Navigability of Web Applications: A Model-Driven Approach*, in European Journal of Information Systems, 2007, Vol. 16, 420-447.
5. Diaz, P., Montero, S. and Aedo, I.: *Modelling Hypermedia and Web Applications: the Ariadne Development Method*, in Information Systems, Vol.30(8), 2005, 649-673.
6. Escalona, M. J. and Koch, N.: *Metamodelling the Requirements of Web Systems*, in Web Information Systems and Technologies: Int. Conferences WEBIST 2005 and WEBIST 2006. Revised Selected Papers, Springer LNBI, Vol. 1, 2007, 267-280
7. Fons, J., Pelechano, V., Pastor, O., Valderas, P., Torres, V.: *Applying the OOWS Model-Driven Approach for Developing Web Applications. The Internet Movie Database Case*

- Study*, in Web Engineering: Modelling and Implementing Web Applications. Springer Human-Computer Interaction Series. Rossi, G.; Pastor, O.; Schwabe, D.; Olsina, L. (Eds.), 2007.
8. Gellersen, H. W., Gaedke, M.: *Object-Oriented Web Application Development*, in IEEE Internet Computing, 1999. Vol 3(1), 60-68.
 9. Gómez, J., Bia, A. and Parraga, A.: *Tool Support for Model-Driven Development of Web Applications*, in Web Information Systems Engineering (WISE 2005), 2005, 721-730.
 10. Knapp, A., Koch, N., Zhang, G. and Hassler, H.-M.: *Modeling Business Processes in Web Applications with ArgoUWE*, in – Proc of Int. Conf. Unified Modeling Language (UML 2004), Springer LNCS 3273, 2004, 69-83.
 11. Koch, N.: *Transformation Techniques in the Model-Driven Development Process of UWE*, in Workshop Proc. of the 6th Int. Conf. on Web Engineering (ICWE 2006), ACM Vol. 155, Palo Alto, California, 2006.
 12. Koch, N., Kraus, A.: *Towards a Common Metamodel for the Development of Web Applications*, in Proc. 3rd Int. Conf. Web Engineering (ICWE 2003), LNCS 2722, 497-506. Springer Verlag, 2003.
 13. Kraus, A. : *Model Driven Software Engineering for Web Applications*, PhD Thesis, Institut für Informatik, Ludwig-Maximilians-Universität München, 2007.
 14. Langegger, A., Palkoska, J.r. and Wagner, R.: *DaVinci - A Model-driven Web Engineering Framework*, in J. Web. Infor. Syst Vol.2(2), 2006 119-132.
 15. Meliá, S. and Gómez, J.: *The WebSA Approach: Applying Model-Driven Engineering To Web Applications*. in Journal of Web Engineering (JWE), 5(2), 2006 121–149.
 16. Meliá, S. , Gómez J. and Serrano J.L.: *WebTE: MDA Transformation Engine for Web Applications*, in Proc. 7th Int. Conf. Web Engineering (ICWE 2007), Springer LCNS 4607, Como, Italy, 2007.

17. Moreno, N. and Vallecillo, A.: *Towards Interoperable Web Engineering Methods*, accepted for publication at the Journal of the American Society for Information Science and Technology (JASIST), 2008
18. Preciado, J.C., Linaje, M., Sánchez, F.: *Designing Rich Internet Applications with engineering methodologies*, in Proc. of the 9th IEEE Int. Symposium on Web Site Evolution. IEEE Computer Society, 2007, 23-30.
19. Rossi, G. Pastor, O.; Schwabe, D.; Olsina, L. (Eds.): *Web Engineering: Modelling and Implementing Web Applications*. Springer Human-Computer Interaction Series, 2007.
20. Valderas, P., Pelechano, V., Pastor, O.: *A Transformational Approach to Produce Web Application Prototypes from a Web Requirements Model*. In International Journal of Web Engineering and Technology (IJWET), Vol. 3, No. 1, 2007, 4-42.
21. Vara, J.M. De Castro, V., Marcos, E. :, *WSDL Automatic Generation from UML Models in a MDA Framework*, in International Journal of Web Services Practices, Vol.1, No.1-2, 2005, 1-12.
22. Vara, J.M., Didonet Del Fabro, M., Joualt, F. and Bezivin, J.: *Model Weaving Support for Migrating Software Artifacts from AUTOSAR 2.0 to AUTOSAR 2.1*, in Int. Conf. on Embedded Real Time Software (ERTS 2008), Toulouse (France), 2008.
23. Webratio. URL, <http://www.webratio.com>. Last time visited: 29th of January 2007.
24. Wimmer, M., Schauerhuber, A., Schwinger, W., Kargl, H.: *On the Integration of Web Modeling Languages: Preliminary Results and Future Challenges*, in Workshop on Model-driven Web Engineering (MDWE 2007), held in conjunction with ICWE, Como, Italy, 2007.

Nora Koch es investigadora de la Ludwig-Maximilians-Universität (LMU) de Munich desde 1995 y trabaja en F.A.S.T. Applied Software Technology GmbH en consulting y dirección de proyectos desde 1998. Se doctoró en ciencias de la computación en la LMU en el año 2000. Nora es líder del Grupo Web Engineering Group de la LMU, que es responsable del desarrollo de la metodología UWE. Su interés en la investigación se centra en métodos para el desarrollo de aplicaciones para el entorno Web, personalización de sistemas Web e ingeniería dirigida por modelos. Nora organizó la 4ta Conferencia Internacional de Web Engineering (ICWE 2004) en Munich, dos ediciones del Workshop Ingeniería Web dirigida por Modelos (MDWE 2005 y MDWE 2007) y es miembro fundador de la red MDWEnet. Además es miembro del IEEE y de la Sociedad Alemana de Informática. Nora ha publicado más de 70 publicaciones. Más referencias en <http://www.pst.informatik.uni-muenchen.de/~kochn>.

Santiago Meliá es profesor colaborador en el departamento de Lenguajes y sistemas Informáticos en la Universidad de Alicante, donde dirige y enseña en un Master Oficial en Ingeniería del Software. Además, es miembro del grupo de investigación IWAD de la misma universidad. Sus áreas de investigación incluyen el desarrollo dirigido por modelos, las metodologías de Ingeniería Web, las técnicas de generación automática y la arquitectura del software, todas ellas son parte de su trabajo de tesis presentado en Alicante en 2007. Ha publicado en prestigiosas revistas (European Journal of Information Systems, Journal of Web Engineering, Revue d'Interaction Homme-Machine and Lecture Notes in Computer Science) y conferencias (OOPSLA, EC-MDA, ER, EC-Web, ICWE, CADUI, etc.). También ha participado como comité de programa en diferentes conferencias y talleres internacionales. Si desea más información consulte su página personal <http://www.dlsi.ua.es/~santi/>

Nathalie Moreno es profesora ayudante en el departamento de Lenguajes y Ciencias de la Computación de la Universidad de Málaga, donde obtuvo el título de Ingeniera y realiza actualmente su tesis doctoral. Su investigación se centra en el desarrollo dirigido por modelos y, en particular, en metodologías de modelado conceptual, de procesos de negocio, lenguajes de transformación de modelos y técnicas de generación automática de código en el ámbito de las aplicaciones Web. Algunas de sus publicaciones más relevantes figuran en revistas de impacto como European Journal of Information Systems, Journal of the American Society for Information Science o Technology or ET Software; así como en capítulos de libro y conferencias internacionales tales como ICWE, DEXA, MODELS, ICSE, etc. Habitualmente participa como revisor y miembro del comité de programa en reconocidas conferencias y talleres relacionados con su área de investigación. Puede contactar con ella en vergara@lcc.uma.es

Vicente Pelechado es Titular de Universidad en el Departamento de Sistemas Informáticos y Computación (DSIC) de la Universidad Politécnica de Valencia. Sus intereses de investigación abarcan ingeniería web, modelado conceptual, ingeniería de requisitos, patrones software, servicios web, sistemas pervasivos y desarrollo dirigido por modelos. Obtuvo el título de Doctor por la Universidad Politécnica de Valencia

en 2001. Actualmente imparte clases de ingeniería del software, desarrollo dirigido por modelos, diseño e implementación de servicios web y patrones de diseño en la Universidad Politécnica de Valencia. Es miembro del Grupo de Investigación OO-Method. Ha publicado artículos en revistas científicas de reconocido prestigio (Information Systems, Data & Knowledge Engineering, Information and Software Technology, International Journal of Web Engineering and Technology, etc.), capítulos de libro en Springer, IDEA, M. Sharpe y ha publicado en diversas conferencias internacionales (ER, CAiSE, WWW, ICWE, EC-WEB, WISE, AH, ICSOC, DEXA, etc.) Ha sido miembro de Comités Científicos y de Organización de reconocidas Conferencias Internacionales y Workshops como CAiSE, ER, ICWE, ICEIS, ACM MT y IADIS.

Fernando Sánchez Figueroa es profesor en el Departamento de Ingeniería de Sistemas Informáticos y Telemáticos de la Universidad de Extremadura. Su actividad docente abarca las asignaturas de Programación Concurrente y Patrones de Diseño. Fernando Sánchez es Ingeniero en Informática por la Universidad de Sevilla y Doctor por la Universidad de Extremadura. Su actividad investigadora se centra en la Ingeniería Web, especialmente en las Rich Internet Applications, accesibilidad a la Web e Ingeniería Web dirigida por modelos. Ha sido Vicerrector de Nuevas Tecnologías durante cuatro años. Actualmente es director de la cátedra de "Ingeniería del Software" patrocinada por INSA-IBM.

Juan M. Vara es Ingeniero en Informática por la Universidad Rey Juan Carlos, donde realizó los cursos de Doctorado en el programa de Informática y Modelización Matemática. En la actualidad es profesor ayudante en el departamento de Lenguajes y Sistemas Informáticos II de dicha universidad y ha impartido clase en varios masters y cursos de especialización. Es integrante del grupo de investigación Kybele, donde realiza su tesis doctoral en el área del Desarrollo Dirigido por Modelos para sistemas de Información Web. Es co-autor de varios libros y capítulos de libro, así como de numerosas publicaciones en congresos y revistas nacionales e internacionales. Ha participado activamente en varios congresos en el ámbito de la Ingeniería del software, bien como organizador o como miembro del comité de programa y participa en varios proyectos de carácter regional, nacional y europeo. Puede contactar con él en juanmanuel.vara@urjc.es