

ADM: MÉTODO DE DISEÑO PARA LA GENERACIÓN DE PROTOTIPOS WEB RÁPIDOS A PARTIR DE MODELOS

Susana Montero, Paloma Díaz, Ignacio Aedo y Laura Montells

Laboratorio DEI. Dpto. de Informática

Escuela Politécnica Superior

Universidad Carlos III de Madrid

Av. Universidad 30, 28911 - Leganés, (Madrid)

e-mail: {smontero, pdp, lmontell }@inf.uc3m.es, {aedo}@ia.uc3m.es web:

www.dei.inf.uc3m.es

Palabras clave: Métodos de diseño dirigidos por modelos, meta-modelado, prototipado rápido

***Resumen.** El desarrollo basado en modelos resulta idóneo en el caso de los sistemas web en el que tanto las tecnologías de implementación como las propias funcionalidades ofrecidas están en permanente evolución. El método ADM sigue un enfoque MDD para afrontar el modelado de las aplicaciones hipermedia y web que permite representar todas las características del sistema, utilizando abstracciones pertenecientes al dominio de la hipermedia, interrelacionándolas entre sí y consiguiendo una especificación independiente de plataforma. En este artículo se presenta una panorámica de los diferentes trabajos realizados en el ámbito de este método de diseño que permiten considerarlo como una aproximación de MDD para ingeniería de la web.*

1. INTRODUCCIÓN

El desarrollo basado en modelos (MDD- “Model-Driven Development”) es un enfoque para la construcción de software que ha ganado en popularidad en los últimos años merced al esfuerzo del grupo OMG y, en particular, a su propuesta MDA (“Model Driven Architecture”) [1]. Este paradigma aboga por trasladar el peso del desarrollo desde la implementación hacia la creación de modelos basados en algún meta-modelo formal, de tal forma que se pueda realizar una especificación abstracta del software en términos de elementos y servicios del dominio de la aplicación a partir de los cuáles se puede generar código de forma automática o semi-automática. El valor fundamental de los meta-modelos formales reside en su capacidad para garantizar la independencia de la plataforma de implementación y, por ende, para facilitar la interoperabilidad entre sistemas heterogéneos, características ambas que han hecho que algunas comunidades de desarrollo, como la de la hipermedia a comienzos de los noventa [2],

hayamos visto en ellos una herramienta básica para poder llevar a cabo un desarrollo de calidad. Un paradigma de desarrollo como éste resulta idóneo en el caso de los sistemas web en el que tanto las tecnologías de implementación como las propias funcionalidades ofrecidas están en permanente evolución, de manera que la interoperabilidad y la flexibilidad se convierten en requisitos ineludibles de cualquier desarrollo.

En el área de la Ingeniería Web podemos encontrar métodos y herramientas que proporcionan modelos separados para tratar con diferentes aspectos de diseño (la navegación, la presentación, la estructura de la información, etc.), y que cuentan con herramientas para compilar dichos modelos y producir aplicaciones web basadas en estos modelos, como es el caso de OO-H [3], UWE [4] o WebML [5]. En este artículo se presenta el proceso de desarrollo de ADM (Ariadne Development Method) [6] que también sigue un enfoque MDD para afrontar el modelado de las aplicaciones hipermedia/web permitiendo representar todas las características del sistema relacionadas con la estructura, la navegación, la presentación, la interacción y el control del acceso. Para ello hace uso de abstracciones pertenecientes al dominio de la hipermedia que proceden de un meta-modelo denominado Labyrinth y las interrelaciona mediante una serie de reglas de validación consiguiendo una especificación independiente de plataforma. Además, el método cuenta con una herramienta software, llamada AriadneTool, que asiste al diseñador durante el proceso de desarrollo y transformación de los modelos.

Desde su concepción, el método ADM adoptó un enfoque basado en modelos que es, precisamente, el que se describe en este artículo. El resto del artículo se organiza de la siguiente forma: la sección 2 presenta el proceso metodológico de ADM y el meta-modelo en el cual se basa; las secciones 3, 4 y 5 describen cada una los niveles de abstracción junto con los modelos y transformaciones de modelo que conforman el método según la propuesta MDA; finalmente, las conclusiones y trabajos futuros se presentan en la sección 6.

2. PROCESO METODOLÓGICO DE ADM

En esta sección se presenta el proceso metodológico conocido como ADM (Ariadne Development Method) para el desarrollo de sistemas hipermedia/web.

2.1 Breve descripción de ADM

ADM combina el proceso de diseño centrado en el usuario y flexible con una aproximación de ingeniería de modelos hipermedia basado en el proceso de desarrollo MDA, tanto en cuanto los productos que resultan de cada fase son modelos de diferentes vistas de diseño (estructura, navegación, presentación, comportamiento, procesos y acceso), abordadas desde diferentes niveles de abstracción. Por lo tanto, ADM no está orientado a la implementación en una tecnología específica, sino que ofrece productos de diseño generales que pueden ser trasladados a diferentes entornos de implementación.

Para conseguir esta independencia entre la especificación del sistema y su implementación es necesario que los modelos tengan una sintaxis y una semántica bien definidas que, en el

caso de ADM, son proporcionadas ambas por el metameta-modelo de referencia Labyrinth [7,8]. Las entidades y relaciones de Labyrinth representan los componentes de cualquier sistema hipermedia/web (p.e. nodos, contenidos, enlaces, anclas, atributos, eventos y usuarios) así como las construcciones básicas (p.e. función de localización, mecanismo de composición, sistema de anclado, restricciones temporales y espaciales) ofreciendo un lenguaje común válido para diferentes plataformas de implementación. Sobre la base de este metameta-modelo se definieron una serie de modelos de diseño orientados a ayudar al diseñador a especificar los requisitos de su sistema a dos niveles de abstracción: el nivel conceptual, cuyo objetivo es identificar entidades tipo y el nivel detallado en el que se hace una especificación exhaustiva de los distintos modelos. Por ejemplo, modelos de diseño definidos sobre los elementos del meta-modelos en el nivel conceptual son el Diagrama Estructural, el Diagrama Navegacional, las Especificaciones Funcionales, el Catálogo de Accesos, los Diagramas Internos, los Catálogos de Atributos y Eventos, el Diagrama de Usuarios o la Tabla de Autorizaciones. Modelos propios del nivel detallado serían, entre otros, el Diagrama de Instancias, el Diagrama de Usuarios Instanciado, las Especificaciones de Presentación, los Diagramas Internos Detallados, las Reglas de Autorización Detalladas, la Tabla de Accesos o las Especificaciones Detalladas de Funciones. Más detalles sobre los distintos modelos de ADM se puede encontrar en [6, 9].

En la figura 1 se muestra cada una de las fases que componen el proceso de desarrollo de estos modelos y las relaciones existentes entre dichas fases.

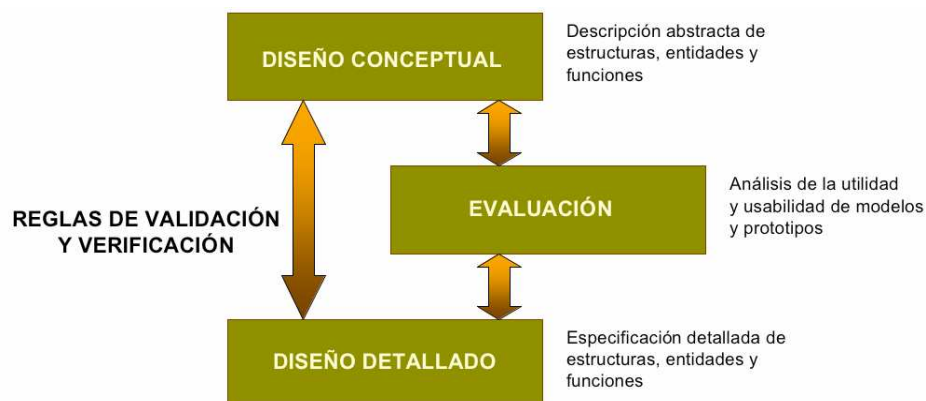


Figura 1: El proceso de desarrollo en ADM

El Diseño Conceptual tiene como objetivo describir las características estáticas y dinámicas del sistema desde una perspectiva muy abstracta, en términos de entidades-tipo, mientras que el Diseño Detallado da lugar a una especificación muy concreta casi trasladable automáticamente a unidades de implementación. Finalmente, la Evaluación hace posible verificar la validez y utilidad de las soluciones propuestas, ya sea en forma de productos de diseño o de prototipos, involucrando al usuario en este proceso. ADM cuenta con una herramienta de automatización, AriadneTool, que permite llevar a cabo transformaciones semi-automáticas y asistidas entre los diferentes niveles de abstracción, proporcionando así la

generación de prototipos ligeros en lenguajes de marcado como HTML, XML, SMIL y RDF/S.

2.2 El enfoque dirigido por modelos de ADM

Como se ha mencionado anteriormente, ADM puede considerarse un método de desarrollo dirigido por modelos para sistemas hipermedia/web, cuyo proceso metodológico se representa de forma esquemática en la figura 2.

En primer lugar, el método se sirve de un conjunto de patrones de diseño para ayudar al diseñador a identificar los requisitos de su sistema. Estos patrones, que constituyen por tanto los Modelos Independientes de Computación (CIM -Computational Independent Model), están formalizados mediante ontologías, permitiendo transformar un conjunto de requisitos en un conjunto de modelos de ADM a través de las soluciones propuestas en los patrones seleccionados. En segundo lugar, los Modelos Independientes de la Plataforma (PIM-Platform-Independent Model) son los productos del modelado de ADM, tanto conceptual como detallado, y representan soluciones de diseño a los requisitos expresadas de forma independiente a la plataforma de implementación y ejecución. Finalmente, los Modelos Específicos de la Plataforma (PSM- Platform-Specific Model), que proporcionan modelos en términos de constructores de implementación propios de una tecnología específica, son generados de manera automática a partir de los PIM mediante una herramienta software llamada AriadneTool.

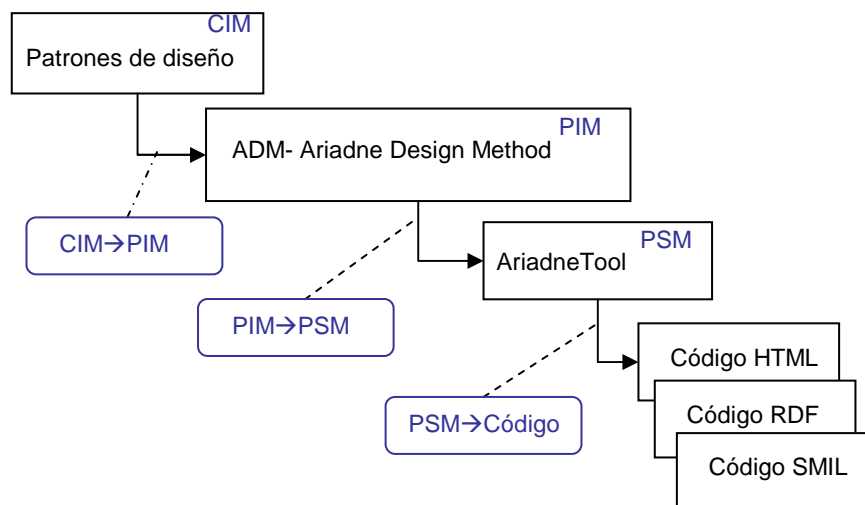


Figura 2. Proceso metodológico de ADM basado en MDA

Entre cada nivel existen mecanismos de transformación automática. Así, la transformación entre modelos CIM y modelos PIM se realiza mediante la traducción de las soluciones incluidas en los patrones de diseño seleccionados a entidades de diseño de ADM, para lo cual recurre al uso de ontologías y a la composición de los diferentes modelos PIM en que se puede

transformar una misma solución. Dichos modelos son una primera aproximación al diseño conceptual que es refinada por el diseñador para dar lugar a unos modelos más detallados. Los modelos PIM resultantes de las iteraciones realizadas durante el modelado son transformados en modelos PSM mediante la aplicación del patrón Builder [14], con el que se generan modelos específicos en HTML, XML, RDF y SMIL. Todas estas transformaciones se realizan gracias al meta-metamodelo y meta-modelo en los que se sustenta ADM.

2.3 Meta-Modelo

En el marco MDA es muy importante contar con el meta-modelado, ya que actúa como mecanismo para definir modelos no ambiguos. Esta no ambigüedad va a permitir que una herramienta pueda leer, escribir y entender modelos construidos para definir una aplicación de manera conceptual, y también que se puedan especificar las reglas de transformación para pasar de un modelo a otro de diferente nivel de abstracción. En este contexto ADM está soportado por dos niveles de abstracción que se corresponden con los niveles M3 y M2 de las capas de modelado propuestas por OMG.

- Nivel de meta-metamodelo (nivel M3). Representan las entidades fundamentales de cualquier sistema hipermedia (p.e. nodos, contenidos, enlaces, anclas, eventos, usuarios, etc.) así como las construcciones básicas (p.e. función de localización, mecanismo de composición, sistema de anclado, restricciones temporales y espaciales). Estas entidades y funciones se encuentran especificadas en el modelo de referencia Labyrinth [7,8] y por lo tanto este meta-metamodelo puede ser reutilizado para la especificación de otros meta-modelos pertenecientes a métodos distintos a ADM. Además dicho meta-metamodelo ha sido representado mediante una ontología [10] con el objeto de dar soporte a las transformaciones CIM-PIM, a la vez que permitiría el intercambio de modelos definidos para otros métodos.
- Nivel de meta-modelo (nivel M2). Son las entidades-tipo de las que se componen los diferentes modelos de diseño del método ADM y que están basadas en las descritas por el nivel M3. Por ejemplo, el Diagrama de Navegación de ADM proporciona componentes del nivel M3 como el nodo, el enlace y el ancla. A su vez, un nodo del nivel M3 se corresponde con una composición de varias entidades-tipo en el nivel M2 que proceden de tres productos de diseño: el Diagrama Estructural y de Navegación (de dónde se consigue el identificador y la etiqueta) y Tabla de Categorizaciones (de dónde se extrae la categoría de acceso).
- Nivel de modelado (nivel M1). Se corresponde con los modelos realizados por los diseñadores realizando instancias de los meta-modelos de M2 que para dar solución a los requisitos de su sistema. Gracias a la definición ontológica del meta-metamodelo se han podido añadir un conjunto de reglas que permiten comprobar la completitud, la consistencia y la corrección de estos modelos de nivel M1, de tal manera que se restringe la forma de utilización de las entidades-tipo acorde a la semántica definida por cada uno de los meta-modelos de por ADM.

En las siguientes secciones se procede a describir cada uno de los niveles de abstracción según el marco MDA para el método de diseño ADM.

3. CONSTRUCCIÓN DE MODELOS DE DISEÑO CON ADM

ADM no propone una técnica concreta para la especificación funcional de requisitos, pudiendo elegir el diseñador entre alguno de los recopilados en [11]. Sin embargo si propone un mecanismo de transformación para pasar del nivel CIM al nivel PIM.

Los patrones de diseño en el área de la hipermedia siempre han sido propuestos como una técnica complementaria a los métodos de diseño: “Los patrones pueden ayudar a rellenar el hueco entre la especificación de requisitos y el modelado conceptual”[12]. Además, el conocimiento registrado por los patrones se presenta como una regla que guía la transformación entre un problema dado y una solución.

Partiendo de este contexto, se puede establecer una correspondencia entre problemas conceptuales y soluciones de modelado, es decir, los requisitos de la aplicación son identificados con el problema del patrón, mientras que la solución del patrón permite generar una primera aproximación a modelado conceptual. El método ADM proporciona un lenguaje de patrones hipermedia y web que están organizados para cubrir diferentes vistas de diseño (navegación, estructura, presentación, interacción, personalización y seguridad). Para cada uno de los requisitos, el diseñador selecciona el patrón que mejor lo cubre, y así la solución del patrón guía el desarrollo de los diferentes modelos PIM. Además, el diseñador puede utilizar estos patrones para identificar sus propios requisitos [13].

Con el objeto de automatizar este proceso, los patrones fueron formalizados mediante ontologías. Estas ontologías representan tanto la estructura del patrón como las entidades fundamentales del dominio hipermedia, y gracias a que el meta-modelo de ADM está también formalizado mediante ontologías, las soluciones de los patrones pueden ser transformadas en entidades de diseño que son situadas en su correspondiente modelo PIM, generando así una primera aproximación al modelado conceptual.

Esta unión entre requisitos y patrones va a permitir la trazabilidad de los primeros a través de los modelos PIM, gracias a una de las principales características de los patrones de diseño: *permiten documentar soluciones software* [14].

4. REFINANDO LOS MODELOS DE DISEÑO

El siguiente nivel de abstracción tiene como objeto especificar de manera conceptual la especificación de requisitos utilizando como representación formal el meta-modelado. ADM aborda el modelado de las aplicaciones hipermedia y web desde seis vistas de diseño independientes y complementarias que se traducen en los siguientes grupos de modelos:

- **Modelo de navegación.** Establece la estructura de enlaces y las herramientas de navegación que conforman el *hiperespacio* por el cual el usuario navegará para conseguir sus objetivos en la aplicación. Este es uno de los productos clave en un buen diseño, puesto que la navegación constituye el verdadero valor añadido de esta tecnología por lo que el diseñador deberá definir una estructura útil y usable

que evite la desorientación del usuario.

- **Modelo de presentación.** Los nodos incluyen contenidos multimedia que deben organizarse y armonizarse en diferentes dimensiones, tales como el tiempo o el espacio bi- o tri-dimensional. La forma en la que se orqueste la presentación multimedia puede determinar la utilidad y usabilidad del sistema.
- **Modelo estructural.** Este modelo permite definir la estructura jerárquica intrínseca a muchos sistemas web, que no debe confundirse con la estructura de navegación. Además, la utilización de este modelo ayuda a analizar el universo del discurso, en términos de entidades relevantes y relaciones existentes, así como a adquirir un conocimiento más profundo de él.
- **Modelo de comportamiento.** Los sistemas hipermedia son altamente interactivos, incluyendo aspectos dinámicos como la reacción del sistema a eventos concretos, el acceso a aplicaciones externas, como bases de datos o motores de inferencia, o la inclusión de objetos virtuales y estructuras que son creadas y modificadas en tiempo de ejecución.
- **Modelo de procesos.** Además de las funcionalidades de navegación, las aplicaciones hipermedia incluyen cada vez más otro tipo de servicios como procesos de negocio, motores de búsqueda o funcionalidades de personalización y adaptación, que deben ser modeladas en relación con el resto de los componentes del sistema.
- **Modelo de control de acceso.** Debido a la proliferación de aplicaciones hipermedia multi-usuario, se ha incrementado la necesidad de preservar la seguridad de la información, así como adaptar los sistemas a las necesidades y preferencias de los usuarios. Con este propósito, se identifican los tipos de usuarios o roles, y no los usuarios individuales, que utilizarán el sistema así como sus funciones y los permisos que se les concederán.

4. GENERANDO PROTOTIPOS

Todos los niveles de abstracción anteriormente descritos están soportados por una herramienta software conocida como AriadneTool. Esta herramienta ha sido desarrollada tanto para asistir al diseñador en los diferentes niveles de modelado como en la transformación de los modelos entre diferentes niveles. La herramienta trabaja mediante proyectos y almacena toda la información de los mismos en un repositorio XML acorde al meta-modelado descrito en la sección 2.1. Los requisitos son capturados por la herramienta a través de formularios acorde al modelo Volere [11]. Cada requisito puede ser enlazado con un determinado patrón de diseño y generar así una primera aproximación al modelado conceptual. Posteriormente, el diseñador puede refinar los diferentes modelos de diseño, moviéndose entre dos niveles de abstracción (conceptual y detallado) a la vez que comprueba la completitud y correctitud de los mismos o genera la documentación asociada al proyecto. Finalmente, el diseñador puede generar un prototipo inicial de la aplicación diseñada de manera automática en alguno o todos de los lenguajes de marcado para la web soportados por

la herramienta, como son HTML, XML, SMIL y RDF.

4.1. El patrón Builder

Para poder generar código a partir de los modelos PIM se ha desarrollado un módulo en la herramienta AriadneTool cuya estructura está basada en el patrón de diseño Builder [14]. Este patrón se usa cuando el algoritmo para crear un objeto complejo debe ser independiente de las partes de que se compone dicho objeto y de cómo se ensamblan, y cuando el proceso de construcción debe permitir diferentes representaciones del objeto que está siendo construido. La figura 3 muestra el diagrama de clases para el patrón Builder aplicado a nuestro contexto.

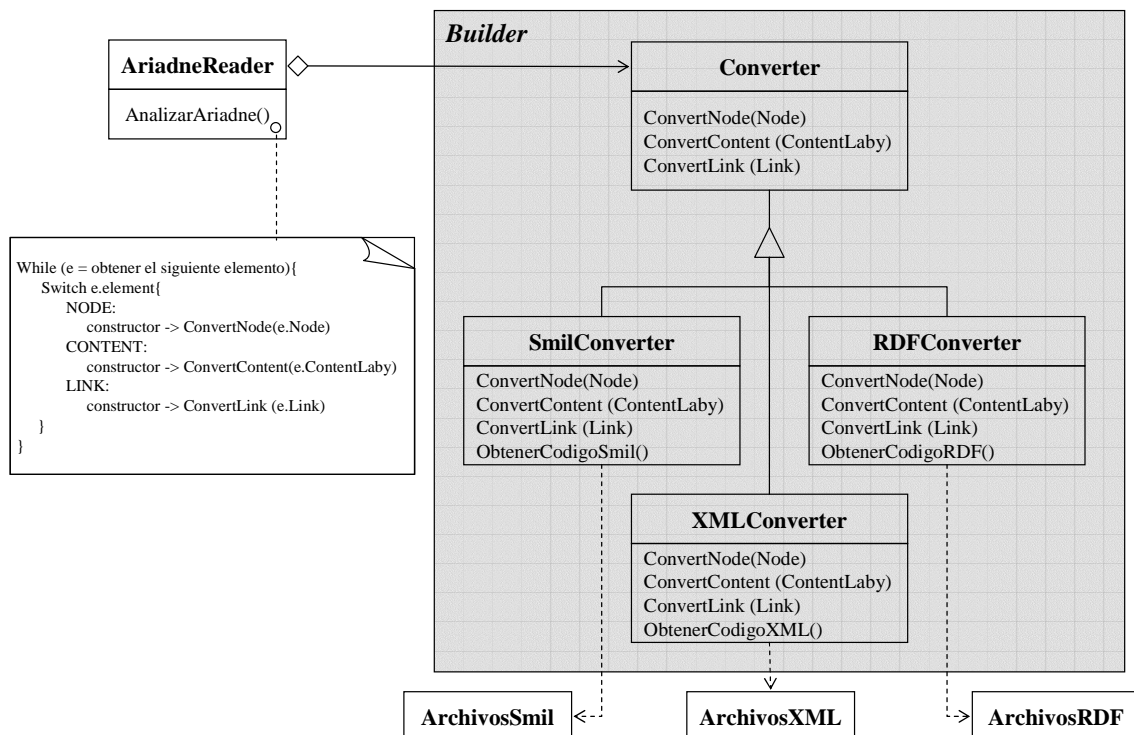


Figura 3. Diagrama del patrón *Builder* adaptado al módulo generador de prototipos

En este caso, nuestro objeto complejo es el prototipo de una aplicación hipermedia, que está formada de diferentes modelos (estructura, navegación, presentación, comportamiento, proceso y control de acceso). El modo de ensamblar las entidades de dichos modelos (nodos, contenidos, anclas, usuarios, procesos, etc.) puede ser diferente según el lenguaje de implementación final seleccionado a la vez que se pueden generar implementaciones de la aplicación en diferentes lenguajes. Este patrón nos ha permitido separar la implementación del prototipo de su representación conceptual, generando así diferentes PSMs.

Durante el proceso de generación de prototipos, las especificaciones declarativas realizadas durante la elaboración del proyecto de acuerdo a la especificación de ADM tienen que ser

compiladas para los lenguajes destinos seleccionados, incluyendo la resolución de los conflictos entre el modelado conceptual y la tecnología (p.e. los enlaces n -arios no están soportados por la tecnología Web), para los cuales se pedirá la intervención del usuario. Por lo tanto, la clase *AriadneReader* inicia el proceso de construcción desde las entidades fundamentales descritas durante el modelado detallado. La interfaz *Converter* especifica una interfaz abstracta para crear partes de un prototipo. Cada lenguaje destino construye y ensambla partes del prototipo implementando el interfaz *Converter*, produciendo así ficheros que contienen las entidades de bajo nivel codificadas y ensambladas entre ellas para cada tecnología. Hasta ahora, AriadneTool genera prototipos ligeros en SMIL, RDF/RDFS y XML. HTML es generado a partir del XML y un conjunto de hojas de estilo XSLT para su transformación.

La utilización del patrón *Builder* como esqueleto del módulo de generación de prototipos permite la inclusión nuevos lenguajes destino de manera modular e independiente.

6. CONCLUSIONES Y TRABAJO FUTURO

En este artículo hemos presentado cómo el proceso de desarrollo de ADM (Ariadne Development Method) plantea un enfoque MDD para los sistemas hipermedia y web. La diferencia de ADM con respecto a otras aproximaciones, es que además de cubrir las tradicionales vistas de navegación, estructura y presentación cuenta con productos para tratar con el diseño de composiciones multimedia (incluyendo restricciones espacio-temporales), el modelado de los usuarios del sistema y la especificación de reglas de acceso orientadas a definir personalización, adaptación y políticas de seguridad. Otra característica relevante del método consiste en contar con una actividad de Evaluación en la que se incluyen criterios específicos y que permite detectar mejoras y carencias relacionadas con la utilidad y usabilidad en cualquier momento del proceso del diseño. La herramienta de automatización, AriadneTool, utiliza ontologías para comprobar la completitud y la correctitud de los modelos de diseño realizados por el usuario con respecto a la semántica definida por ADM, y para realizar la transformación entre el modelo CIM a modelos PIM y de modelos PIM a un modelo PSM codificado en RDF/S desarrollando así aplicaciones orientadas a la Web Semántica [15].

Actualmente se está trabajando en la especificación de meta-metamodelo Labyrinth y de los modelos de ADM utilizando otros lenguajes formales, con objeto de comparar las ventajas e inconvenientes de cada aproximación. En concreto, se está llevando a cabo una especificación con gramáticas de grafos en la herramienta Atom3 [16] que ha hecho posible la inclusión de reglas de re-diseño que serían equivalentes a las reglas incluidas en la ontología y de métricas de diseño, donde valores extremos de estas métricas pueden disparar la aplicación de rediseños, proponiendo así mejoras al diseñador.

REFERENCIAS

- [1] "Model Driven Architecture", Document number ormsc/2001-07-01, Architecture

- Board ORMSC 2001
- [2] F. Afrati and L.D.Koutras. A Hypertext Model Supporting Query Mechanisms. Proceedings of the European Conference on Hypertext (Inria, Francia). 52-660, 1990
 - [3] J. Gómez, C. Cachero, y O. Pastor. Conceptual modeling of deviceindependent web applications. IEEE MultiMedia, 8(2):26–39, 2001.
 - [4] R. Hennicker y N. Koch. A UML-based methodology for hypermedia design. In UML 2000 - The Unified Modeling Language. Advancing the Standard. Third International Conference, York, UK, October 2000, Proceedings, LNCS, pages 410–424, 2000.
 - [5] S. Ceri, P. Fraternali y M. Matera. Conceptual Modeling of Data-Intensive Web Applications, IEEE Internet Computing 6(4), July/August 2002.
 - [6] P. Díaz, S. Montero, e I. Aedo. Modelling hypermedia and web applications: the Ariadne Development Method. Information Systems, Vol. 30, No. 8, pp. 649-673. 2005.
 - [7] P. Díaz, I. Aedo, e F. Panetsos. Labyrinth, an abstract model for hypermedia applications. description of its static components. Information Systems, 22(8):447–464, 1997.
 - [8] P. Díaz, I. Aedo, e F. Panetsos. Modelling the dynamic behaviour of hypermedia applications. IEEE Transactions on Software Engineering, 27(6):550–572, June 2001.
 - [9] S. Montero, P. Díaz, e I. Aedo. Ariadnetool: A design toolkit for hypermedia applications. Journal of Digital Information, 5(2), 2004.
 - [10] S. Montero, P. Díaz, e I. Aedo. Toward hypermedia design methods for the semantic web. In 14th International Workshop on Database and Expert Systems Applications (DEXA'03), pages 762–767. IEEE Computer Society, 2003.
 - [11] J. Preece, Y. Rogers, and H. Sharp. Interaction Design: Beyond Human-Computer Interaction. New York, NY: John Wiley & Sons, 2002.
 - [12] F. Garzotto, P. Paolini, D. Bolchini, and S. Valenti. Modeling-by-Patterns of web applications. In Advances in Conceptual Modeling: ER '99 Workshops on Evolution and Change in Data Management, Reverse Engineering in Information Systems, and the World Wide Web and Conceptual Modeling, pages 293–306, 1999.
 - [13] S. Montero, P. Díaz, e I. Aedo. From requirements to conceptual modeling of web applications through design patterns Conference Workshop on HCI patterns: Mapping User Needs into Interaction Design Solutions. INTERACT 2005.
 - [14] E. Gamma, R. Helm, R. Johnson, y J. Vlissides. Design Patterns, Elements of Reusable Object-Oriented Software. Addison-Wesley, 1994.
 - [15] L. Montells, S. Montero, P. Díaz e I. Aedo. Desarrollando aplicaciones hipermedia para la Web Semántica. Actas del Taller de Ingeniería Web y COTS. CEDI 2005.
 - [16] E. Guerra, P. Díaz y J. De Lara. Generación de entornos de modelado avanzados mediante técnicas de transformación de grafos. IEEE América Latina 4(2). 2006.