

Estudio Comparativo de Herramientas MDA para el Desarrollo de Sistemas Web

Magalí González, Luca Cernuzzi, Juan José Bareiro, Rodney Rodríguez

DEI - Universidad Católica “Nuestra Señora de la Asunción” – Asunción - Paraguay
{ mgonzalez, lcernuzz, juan_bareiro, fito_rodriguez}@uca.edu.py

Abstract. In recent years, were emerged a wide range of tools that support the MDA approach offering benefits to developers on several desirable features of the development process. This study compares MDA tools for the design and development of Web systems, focusing on the navigational perspective. We analyze both some traditional MDA tools as well as tools which support Web-based methodologies. The navigational perspective in the first class of tools is usually not considered as a first class abstraction in their models, while in the others it is considered very relevant and normally there exist specific models to cover it. The evaluation framework defines metrics for measurable attributes and applies the Logic Score of Preference (LSP) model to evaluate the most general characteristics.

Keywords: Herramientas MDA, Sistemas Web, Comparación, Navegación

1 Introducción

El desarrollo de Sistemas Web, trae consigo nuevas dimensiones con una serie de complejidades que anteriormente eran consideradas de manera parcial en el desarrollo de sistemas tradicionales. En particular, uno de los aspectos que principalmente diferencia a los Sistemas Web de los Sistemas tradicionales es la Navegación, considerado por diversos autores como un punto crítico en la Ingeniería Web [4]. Cabe mencionar que un esfuerzo importante en esta área se ha destinado a proponer diversas propuestas metodológicas [14] [18] [19], algunas de las cuales vienen acompañadas de herramientas que soportan el proceso de diseño y construcción.

Por otro lado, desde el surgimiento de MDA (Model Driven Architecture) [17] como arquitectura estándar que soporta el enfoque de desarrollo dirigido por modelos ha nacido una amplia gama de herramientas que adoptan dicha arquitectura [2] [3]. En general, las herramientas MDA actuales, no consideran a la navegación como elemento crítico o principal dentro de sus modelos. Para éstas, el aspecto navegacional se modela dentro de la capa de presentación, cosa que se contrapone con las diferentes propuestas metodológicas actuales para el desarrollo de aplicaciones Web, que consideran el aspecto navegacional en un modelo independiente y crítico, con diversos constructores que abstraen importantes aspectos relacionados con la navegación (como la orientación, mapas navegacionales, entre otros [8] [14] [18]).

Los avances que tuvo la Ingeniería de Software a partir de MDA son muy valiosos. En particular, las herramientas facilitan el desarrollo de aplicaciones y ofrecen ventajas en términos de mejorar la productividad, la reusabilidad, la portabilidad, interoperabilidad y mantenibilidad [6].

Tras el surgimiento de herramientas MDA, los estudios y comparaciones de las mismas se hicieron comunes en el ámbito de investigación de esta arquitectura [1] [16] [15]. Si bien el presente estudio toma como referencia a trabajos previos, interesa realizar una nueva comparación de herramientas, tomando esta vez como referencia el desarrollo de aplicaciones Web, considerando así aspectos no analizados en las demás comparaciones. Uno de los objetivos de este trabajo es considerar herramientas MDA que surgieron a partir de enfoques metodológicos para aplicaciones Web y confrontarlas con las herramientas MDA tradicionales, con el objeto de identificar puntos fuertes y débiles de ambos mundos, como por ejemplo aquellos directamente relacionados con el modelado y generación del aspecto navegacional, elemento considerado como crítico por la gran mayoría de las metodologías Web. Finalmente, esta comparación no evalúa a cada herramienta de forma general con un único valor, sino que identifica propiedades de la misma puntuables de por sí, sobre las que se realiza una evaluación independiente y posteriormente se obtiene un valor global, aplicando el modelo LSP (Logic Scoring of Preferente) [7] a estas propiedades.

El resto del trabajo se organiza de la siguiente manera: en la Sección 2, se presenta una introducción a las herramientas analizadas; en la Sección 3, se propone el marco comparativo con los elementos utilizados para el análisis y los resultados obtenidos; y por último la conclusión y trabajos futuros se presentan en la Sección 4.

2 Herramientas MDA y el Entorno Web

En la actualidad existe una amplia gama de herramientas que soportan la arquitectura MDA. En esta sección se presenta un resumen de algunas herramientas disponibles actualmente, utilizadas para este trabajo comparativo. Todas las herramientas estudiadas generan en mayor o menor medida una aplicación con una interfaz Web.

Debido a la gran cantidad de herramientas existentes, no es posible en este contexto realizar una comparación exhaustiva y por ende se ha seleccionado un subconjunto a partir de una serie de criterios generales:

- Seleccionar las herramientas con mayor reconocimiento en el ámbito MDA considerando ambientes diversificados, como el mundo open source y el ambiente comercial.
- Seleccionar herramientas definidas como soporte a propuestas metodológicas (ya que estas últimas proponen modelos específicos para la navegación).
- Dar soporte a por lo menos dos plataformas destino: .NET y J2EE

A partir de los mencionados criterios se han descartados herramientas muy interesantes como ser OptimalJ porque genera aplicaciones solamente para la plataforma J2EE. Así, se ha llegado finalmente a la selección de cuatro herramientas: dos del ámbito MDA y dos nacidas como soporte a métodos Web.

Entre las herramientas del ámbito MDA se han seleccionado AndroMDA [2] ya que corresponde a una de las herramientas Open Source con mayor reconocimiento en

el ámbito MDA [11], y ArcStyler [3] que es una de las herramientas MDA comerciales más extendida y ha sido objeto de varios estudios con resultados muy interesantes [9], [1], [6].

Las herramientas del ámbito Web seleccionadas han sido ArgoUWE [13] y OOWS-Suite [19], ya que ambas soportan métodos Web particularmente reconocidos; respectivamente los métodos UWE [14] y OOWS [8].

Por motivos de espacio no es posible presentar en detalle las herramientas seleccionadas. Sin embargo, se sintetizan las principales características en la tabla 1.

Tabla 1. Características resumidas de herramientas MDA

Característica	AndroMDA	ArcStyler	Argo-UWE	OOWS-Suite
Enfoque	Tradicional	Tradicional	Método Web (UWE)	Método Web (OOWS)
Licencia	Open Source	Comercial	Open Source	Propietario
Versión UML soportada	1.3, 1.4, 1.5, 2.0	1.3, 1.4, 1.5	2.0	2.0
Estándares MDA soportados	XMI, OCL, MOF, Perfiles UML	XMI, perfiles UML, OCL parcialmente	XMI, perfiles UML, MOF, OCL parcialmente	XMI, OCL
Extensibilidad	Cartuchos	Cartuchos-MDA	No disponible	No disponible
Modelo de Requisito	Casos de Uso	No modela	Casos de Uso y Actividades	Casos de Uso
Modelo Conceptual	Diagramas de Clases (entidades, VO)	Diagramas de Clases	Diagramas de Clases	Diagrama de Clases
Lógica de la Aplicación	Diagrama de Clases (Servicios)	Diagramas de Estado	Diagramas de Actividad	Diagramas de Estado, Diagramas de Secuencia, notación propia
Navegación	No presenta	Diagramas de componentes	Diagramas de Clase	Diagramas de paquetes
Presentación	Diagramas de Estados	Diagramas de Actividades y notación propia	Diagramas de clases	Diagrama de clases (contextos)
Plataformas destinos	Principalmente J2EE y .NET, pero abierto a otras	Principalmente J2EE y .NET	Principalmente java	Principalmente php

3 Análisis de Herramientas MDA

En esta sección se realiza el análisis de las herramientas seleccionadas a partir de un marco que permite obtener resultados numéricos facilitando así las comparaciones.

3.1 Marco para el Análisis

El marco para el análisis de las herramientas se resume en los siguientes puntos:

1. Definición del objetivo de la evaluación: misión global.
2. Definición de un árbol de atributos: corresponde a un árbol compuesto por una serie de características y sub-características, hasta llegar a un nivel de atributos medibles de una manera cuantitativa.
3. Medición de los atributos: para ello se define una métrica que será aplicada a cada atributo seleccionado para obtener un valor comprendido entre el -1 y 5.
4. Medición de las sub-características y características: usando el modelo LSP.
5. Análisis de los resultados obtenidos.

Este marco usa como referencia varios trabajos previos de comparativas [1], [10], [5], [11], [12], para la definición de pasos a seguir en el proceso de evaluación. A continuación se detalla cada uno de ellos.

Definición del Objetivo. Esta evaluación tiene como propósito el de calcular el “grado” con el que herramientas MDA soportan al proceso de desarrollo Web, dando particular énfasis a los elementos navegacionales utilizados por cada herramienta.

Definición del Árbol de Atributos. En la Tabla 2 se visualiza una lista de propiedades características, que toda herramienta MDA debería soportar. La identificación de las características y sub características, fue hecha a partir de lo siguiente:

1. Determinación de criterios generales identificados por la guía MDA [4].
2. Características deseables propuestas por otros autores [1], [5], [6], [12], [16].
3. Identificación de criterios particulares para el modelado de la navegación, a partir del análisis de las diferentes propuestas metodológicas [8], [14], [18], y casos reales de aplicaciones Web.

Como se puede notar en la Tabla 2, para la evaluación de la estructura navegacional, se identificaron 4 puntos críticos (1.3.1 al 1.3.4):

- Navegación Personalizada: es necesario analizar cómo se modela la personalización en la navegación, ya que hoy en día, la gran mayoría de las aplicaciones requieren de la definición de usuarios y acceso restringido a las páginas.
- Modelo de la Estructura Navegacional: representa un Mapa Navegacional fruto de la navegación del usuario. Se debería considerar el contexto así como los enlaces entre los nodos para poder proveer de orientación al usuario, en cuanto a la ubicación contextual y el recorrido.
- Navegación Parametrizada: normalmente los nodos son modelados para brindar un solo servicio mientras puede ser relevante que un nodo pueda ofrecer más de un servicio. Este atributo apunta a identificar si es posible y cómo se invoca solamente uno de estos servicios pasándole sus parámetros requeridos.
- Tipos de Navegación: podemos diferenciar dos tipos de navegación: la Navegación Intercontextual (la navegación convencional) y la Navegación Intracontextual (que ocurre dentro de un mismo contexto). Esta última se da por ejemplo al actualizarse una página o porciones de ella. Es de suma importancia realizar un análisis de cómo se modelan estos tipos de navegación y de qué forma afecta al código generado.

Tabla 2. Árbol de Atributos

Nº	Propiedad	Nº	Propiedad
1	Soporte PIM	3	Transformación y Generación
1.1	Estructura Estática	3.1	Enfoque de transformación utilizado
1.1.1	Diagrama de Clases	3.1.1	Usando Marcas
1.1.2	Diagrama de Objetos	3.1.2	Transformación de metamodelos
1.1.3	Diagrama de Componentes	3.1.3	Transformación de modelos
1.1.4	Diagrama de Despliegue	3.1.4	Utilizando patrones
1.2	Estructura Dinámica	3.2	Dirección de la Transformación
1.2.1	Diagrama de Casos de Usos	3.2.1	Transformación a Código
1.2.2	Diagrama de Secuencia	3.2.1.1	Transformación de PIM -> PSM -> Código
1.2.3	Diagrama de Actividad	3.2.1.2	Transformación PIM -> Código
1.2.4	Diagrama de Colaboración	3.2.2	Transformación de Modelo <- Código
1.2.5	Diagrama de Estados	4	Uso de Estándares
1.3	Estructura Navegacional	4.1	Soporte UML
1.3.1	Navegación Personalizada	4.1.1	Versión UML
1.3.2	Modelo de Estructura Navegacional	4.1.2	Definición de un Metamodelo
1.3.3	Navegación Parametrizada	4.1.2.1	Extensión del Metamodelo de UML (Perfiles)
1.3.4	Tipos de Navegación	4.1.2.2	Definición de un nuevo Lenguaje (MOF)
2	Soporte PSM	4.2	Soporte XMI
		4.3	Soporte OCL
		4.4	Soporte QVT

Definición de Métricas. Para la evaluación elemental todos los atributos fueron puntuados, basándose en los criterios de puntuación expresados en la Tabla 3.

Tabla 3. Criterios de Puntuación

Evaluación	Descripción	Puntos
Crea confusión	La forma en que está implementada crea confusiones o hacen que su uso sea dificultoso.	-1
No soporta	La propiedad no está soportada o no está documentada	0
Poco soporte	La propiedad es soportada en forma indirecta	1
Algún soporte	La propiedad aparece explícitamente dentro de la aplicación pero el soporte es moderado	2
Soporte fuerte	La propiedad está en los manuales y en la herramienta pero su uso depende exclusivamente de la experiencia del usuario	3
Soporte muy fuerte	Todos los aspectos son cubiertos por la herramienta y existen ayudas de cómo usarlo	4
Soporte completo	La propiedad es soportada en forma completa, todos los requerimientos son tomados en cuenta	5

Definición del mecanismo de medición de sub-características y características (LSP).

Como método de medición de propiedades se ha optado por LSP (Logic Scoring of Preference) [7], ya que para este estudio consideramos interesante contar con una serie de propiedades que éste modelo posee como identificar los requerimientos obligatorios otorgando una penalización alta a aquellos que no son cumplidos; identificar el grado de reemplazabilidad o simultaneidad para las características y sub-características; o bien, dar la capacidad de otorgar importancia relativa a través de pesos a cada preferencia de entrada.

El conjunto de propiedades representado en la Tabla 2, puede ser expresado como estructuras de agregación para LSP con sus respectivos pesos asociados y así obtener todos los resultados parciales y globales del árbol, es decir la calidad global.

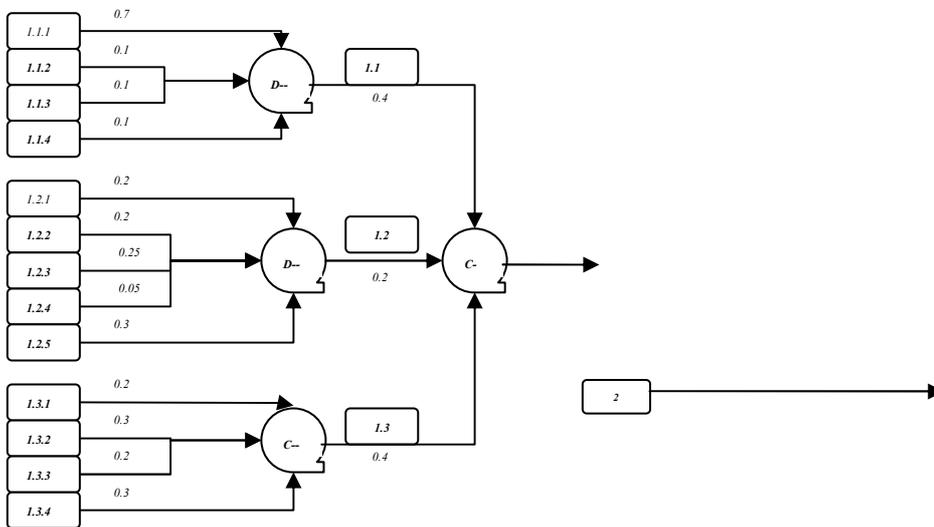


Fig. 1. Asignación de valores de agregación al Soporte PIM y PSM

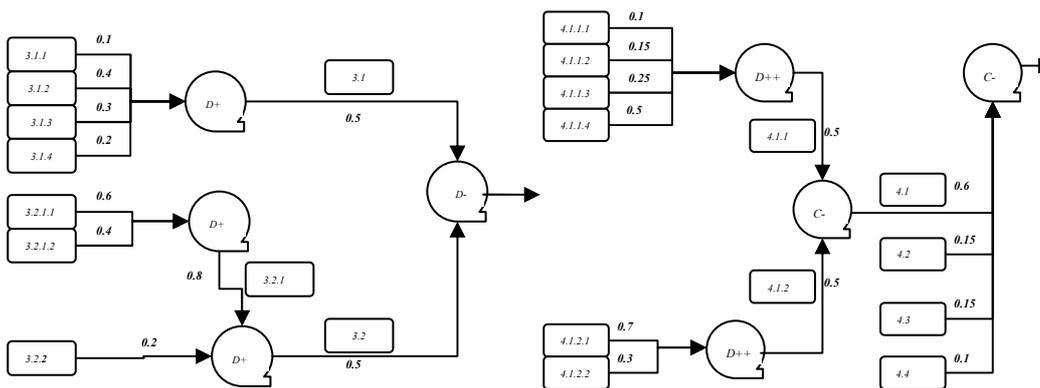


Fig. 2. Asignación de valores de agregación a las características de Transformaciones y soporte de Estándares

En la Figura 1 se muestran las estructuras de agregación de preferencias para las subpropiedades del nivel “1” (Soporte PIM) con los indicadores elementales (1.1.1 al 1.3.4), directamente medibles, clasificados en tres subpropiedades (1.1, 1.2, y 1.3). En general, los pesos asignados van de acuerdo al grado de influencia en MDA de cada propiedad analizada [1].

El valor global de la Estructura Estática (1.1) se obtiene aplicando el valor del operador lógico “D--” a sus distintos indicadores elementales. Se aplica el operador “D--” debido a que se considera que existe un nivel de reemplazabilidad débil entre las propiedades “1.1.1” al “1.1.4”. El diagrama de clases tiene asignado el mayor peso debido a su influencia por sobre el resto de los diagramas.

También para la Estructura Dinámica (1.2) se aplica el operador lógico “D--”. Finalmente, la Estructura Navegacional (1.3) es evaluada con un operador lógico “C--”, o conjunción débil. Los elementos de esta propiedad requieren un cierto grado de simultaneidad, pero la ausencia de alguno de ellos podría no afectar de manera significativa al resultado ya que los principales son el árbol navegacional y el diagrama del nodo. Esto se refleja en los pesos de las propiedades, considerandos más importantes aquellas que afectan directamente a la estructura navegacional (1.3.1 y 1.3.4).

De manera similar fueron definidos los operadores y pesos de las demás dimensiones (ver figura 2), que por cuestiones de espacio no se presentan en este documento.

3.2 Aplicación del Modelo de Análisis de Herramientas MDA

En la tabla 4 se muestra el resultado obtenido para cada una de las propiedades (reducido hasta el primer nivel de agregación por motivos de espacio) analizadas según la estructura de agregación mencionada anteriormente. La evaluación de las herramientas y sus propiedades específicas, fueron hechas en primera instancia a partir de un caso de estudio realizando el desarrollo de un sistema web académico. Sin embargo, hemos considerado además bibliografías, colaboración de algunos de sus creadores [8] [14], y tutoriales.

Tabla 4. Puntuación del Árbol de Atributos.

Nro.	Propiedad	OOWS-Suite	Argo UWE	Andro MDA	Arc Styler
1	Soporte PIM	59,71	55,59	60,47	49,61
1.1	Estructura Estática	63,70	63,70	79,62	86,71
1.2	Estructura Dinámica	51,82	48,51	64,77	47,03
1.3	Estructura Navegacional	60	54,60	43,60	27,34
2	Soporte PSM	40	20	40	40
3	Transformación y Generación	37,42	20,71	37,43	71,54
3.1	Enfoque de Transformación Utilizado	90,65	92,80	92,80	82,89
3.2	Dirección de la Transformación	30,91	17,01	30,90	64,95
4	Uso de Estándares	32,86	42,35	49,94	43,92
4.1	Soporte UML	76,72	87,71	99,15	97,88
4.2	Soporte XMI	100	100	100	80
4.3	Soporte OCL	20	60	100	60
4.4	Soporte QVT	0	0	0	0
	CALIDAD GLOBAL	47,68	44,58	47,78	56,80

3.3 Discusión de los Resultados

A partir de los resultados mostrados en la tabla 3 es posible realizar un análisis comparativo. Los valores resaltados en cada dimensión corresponden a los valores máximos de las preferencias de calidad para la característica respectiva.

Con respecto a la característica “Soporte PIM” se puede concluir que es AndroMDA (60.47) la que mejor comportamiento presenta. Es importante igualmente destacar que las demás herramientas tienen también un buen desempeño, excepto en alguna propiedad.

Al analizar más de cerca los resultados, se puede notar que para la sub-propiedad 1.1, ArcStyler es la que mayor puntaje obtuvo, ya que además de soportar los diagramas de clases también soporta a los diagramas de componentes, para modelar la Estructura Estática de un sitio de manera más completa. Para la sub-propiedad 1.2 AndroMDA modela la Estructura Dinámica de un sitio utilizando diagramas de estados y diagramas de casos de uso, lo cual ofrece ventajas en comparación con las demás herramientas. En la sub-propiedad 1.3 (que representa a los puntos críticos considerados en la navegación) OOWS-Suite obtuvo mejor puntaje. En este sentido OOWS provee un modelo propio para la navegación en donde se realiza una personalización asignando mapas navegacionales a los distintos agentes (o actores) del sistema. Por otro lado, también identifica dos tipos de navegación, el que se encuentra directamente relacionado con la estructura jerárquica y que permite identificar los menús y sub-menús accesibles, y aquella navegación que permite ser accedida a través de una secuencia de acceso. Si bien ArgoUWE también presenta un modelo navegacional, no cubre algunos aspectos que, como la identificación de tipos de navegación y la parametrización.

Se puede ver que el PSM no cuenta con suficiente soporte por parte de las herramientas analizadas. El puntaje más alto (40%) es compartido por 3 de las 4 herramientas estudiadas. La generación de un PSM intermedio entre el PIM y el Código, puede permitir al diseñador un mayor control del proceso de transformación, permitiéndole realizar un mapeo entre los elementos abstractos de un PIM y los elementos concretos de su PSM correspondiente. Todas mencionan que en versiones siguientes darán soporte completo a esta característica.

El puntaje más alto para la transformación y generación lo obtiene ArcStyler (71.54) en parte porque es la única herramienta analizada que brinda un mínimo soporte al “Reverse Engineering”. Una ventaja de AndroMDA con respecto a ArgoUWE y OOWS-Suite, es que es fácilmente extendible a través de los cartuchos, por lo que podría ser considerado más interesante si es necesario extender las plataformas destino.

Por otro lado, es importante aclarar que en el componente navegacional, aquellas que proponen modelos navegacionales generan estructuras más flexibles. En el caso de ArcStyler por ejemplo, la navegación y presentación es la capa que requiere mayores modificaciones, debido a que a nivel de modelado no es del todo flexible. En el caso de OOWS y ArgoUML el aspecto navegacional es fácilmente adaptable en el modelado permitiendo así una mayor flexibilidad en el código generado (por ejemplo, la inclusión de niveles de jerarquías de menú es perfectamente modelable con OOWS, cosa que debería ser modificada en forma manual si se utilizara ArcStyler).

Se puede notar que el Uso de Estándares está mejor soportado por AndroMDA (49.94). Todas soportan al menos alguna versión de UML, XMI, MOF; sin embargo, ninguna soporta QVT sino lenguajes de transformación como: ATL (Atlas Transformation Language), VTL (Velocity Template Language), entre otros. Esto se debe a que la primera versión estable de QVT es posterior a estas herramientas.

Consideraciones Generales

A pesar de que ArcStyler haya obtenido el puntaje global mayor, es importante aclarar que este estudio considera más significativos los puntajes obtenidos para cada una de las propiedades generales (Soporte PIM, Soporte PSM, Transformación y Generación, Uso de Estándares), ya que dependiendo del uso que se requiera, aportará mayor información para poder decidir qué herramienta utilizar.

Por ejemplo, a pesar de que OOWS-Suite y AndroMDA obtuvieron puntajes globales similares, si fuera necesario realizar definiciones de reglas de transformación, el uso de AndroMDA sería más adecuado, ya que esta herramienta permite extender los cartuchos para agregar nuevas reglas de transformación.

En cuanto a la dimensión navegacional, es posible notar que aquellas herramientas que proponen modelos específicos para la navegación, generan aplicaciones más flexibles y que no requieren de grandes modificaciones manuales para adecuarse a las necesidades particulares del problema a resolver.

4 Conclusiones y Trabajos Futuros

En el presente estudio se realiza un análisis comparativo de diversas herramientas que soportan el diseño y construcción de sistemas Web a través de un enfoque MDA. Los aportes con respecto a trabajos anteriores se centran en: I) La Especificidad del Dominio de la Aplicación (Sistemas Web), con la consecuente necesidad de considerar la Perspectiva Navegacional; II) La comparación de Herramientas (MDA tradicionales o bien de Soporte de Métodos para el Diseño de Sistemas Web); y III) la Adopción del Modelo Logic Scoring of Preference que permite un análisis ponderado teniendo en cuenta la “Simultaneidad” o “Reemplazabilidad” de las propiedades.

Por otro lado, con esta comparación, se tuvo en cuenta la relación existente entre las distintas propiedades, lo cual es de suma importancia para MDA, ya que sus fases están fuertemente relacionadas entre ellas. La comparación realizada sugiere que es ArcStyler la herramienta que ofrece un mejor comportamiento en cuanto a las propiedades evaluadas. Sin embargo, es importante aclarar que este estudio considera particularmente significativos los puntajes obtenidos para cada una de las propiedades, ya que dependiendo del uso que se requiera, aportará mayor información para decidir cual herramienta utilizar.

Un posible trabajo futuro prevé agregar otras herramientas a la comparación, como es el caso de “OptimalJ”, “Together Architect”, entre otras.

Referencias

1. Akhter, N., Tariq, N.: Comparison of Model Driven Architecture (MDA) based tools. Institute of Technology y Karolinska University Hospital. Estocolmo (2005)
2. AndromDA. <http://www.andromda.org/>
3. ArcStyler (Interactive Objets). <http://www.interactive-objets.com/products/arcstyler/>
4. Cachero, C., Koch, N. Conceptual Navigation Analysis: a Device and Platform Independent Navigation Specification. 2nd International Workshop on Web-oriented Software Technology (IWWOST'02). Málaga, España (2002)
5. Czarnecki K., Helsen S. Classification of Model Transformation Approaches. OOPSLA'03: Workshop on Generative Techniques in the Context of Model-Driven Architecture. University of Waterloo, Canada (2003)
6. Duarte, J., Gonzalez M., Cernuzzi, L., Pastor, O. IDEAS03: Evaluation of software development through an MDA tool: a case study. Latin America Transactions, IEEE (Revista IEEE America Latina). Volume: 6, Issue: 3. On page(s): 252-259 (2008)
7. Dujmovi , Jozo J., The LSP Method for Evaluation and Selection of Computer and Communication Equipment. Proceedings of MELECON'87, Mediterranean Electrotechnical Conference and 34th Conference on Electronics, IEEE/RIENA, pp. 251-254, Rome, Italy, (1987)
8. Fons, J. - Pelechado, V. - Pastor, O. - Valderas, P. - Torres, V., "Applying the OOWS Model- Driven Approach for Developing Web Applications: The Internet Movie Database (IMDB) Case Study" en Pastor, O. et al (eds.) Web Engineering: Modeling and Implementing Web Applications. Springer, Chapter 5, pp 65-108, London (2007)
9. García, J., Rodríguez, J., Menárguez, M., Ortín, M. J., Sánchez, J.: Un estudio comparativo de dos herramientas MDA: OptimalJ y ArcStyler. Universidad de Murcia. Murcia (2004)
10. González M., Cernuzzi L., Abrahao S., Fons J. Evaluación de métodos de diseño de aplicaciones Web: un análisis comparativo. Workshop Iberoamericano de Ingeniería de Requisitos y Ambientes Software (IDEAS) , ppg 304-309. La Habana, Cuba (2002)
11. Herrera J., Matteo A., Díaz I. "Una Caracterización de Herramientas MDA de Código Abierto". Primer Encuentro Venezolano sobre Tecnologías de la Información e Ingeniería del Software (EVETIS'07). Margarita (2007)
12. Jézéquel, J.M. Model Transformation Techniques. Universidad de Rennes, 2005. <http://www.irisa.fr/prive/jezequel/enseignement/ModelTransfo.pdf>
13. Knapp, A., Koch, N., Moser, F. and Zhang, G. ArgoUWE: A CASE Tool for Web Applications. Ludwig-Maximilians-Universität München, Germany (2003)
14. Koch, N. Knapp, A. Zhang, G. Baumeister, H., "UML-based Web Engineering, An Approach Based on Standards" en Pastor, O. et al. (eds.) Web Engineering: Modeling and Implementing Web Applications. Chapter 7, pp. 157-192. Springer, London (2007)
15. Koch, N., Meliá, S., Moreno, N., Pelechano, V., Sánchez, F., Vara, J.M. Ingeniería Web dirigida por modelos. Novática nº 192 (2008)
16. Modelbased.net. [Documento Electrónico]. http://www.modelbased.net/mda_tools.html
17. Object Management Group, MDA Guide Version 1.0.1. (2003)
18. Rossi, G. - Schwabe, D., "Modeling and Implementing Web Applications with OOHDm" en Pastor, O. - Olsina, L. et al (eds.) Web Engineering: Modeling and Implementing Web Applications. Springer, Chapter 6, pp 109-156, London, (2007)
19. Valverde, F., Valderas1, P., Fons1, J. OOWS Suite: Un Entorno de desarrollo para Aplicaciones Web basado en MDA. Workshop Iberoamericano de Ingeniería de Requisitos y Ambientes Software (IDEAS). Isla Margarita, pp. 253–269, Venezuela (2007)