

Tipo de artículo: Artículo de revisión
Temática: Ingeniería y gestión de software
Recibido :17/6/2011 | Aceptado : 3/8/2011 | Publicado : 28/9/2011

Una comparación de metodologías para el modelado de aplicaciones web

A Methodologies Comparison for the Web Applications Modeling

Rodolfo Villarroel Acevedo ^{1*}, Cristián Rioseco Reinoso ²

^{1*} Pontificia Universidad Católica de Valparaíso. Avenida Brasil 2241, Valparaíso, Chile. rodolfo.villarroel@ucv.cl

² Pontificia Universidad Católica de Valparaíso. c.rioseco.r@mail.ucv.cl

Resumen: Dentro de la ingeniería web se han desarrollado una serie de metodologías para el modelado de aplicaciones web que apuntan a resolver distintos problemas existentes en el desarrollo de este tipo de software. Tanto para los investigadores, como para desarrolladores de software no es trivial identificar y conocer las diferencias entre una y otra metodología por el gran número de propuestas existentes y la complejidad de ellas. En vista de esto, este artículo expone criterios transversales y criterios específicos que permiten la comparación de metodologías con el fin de reconocer las diferencias que hay entre ellas. Con estos criterios, se realizó la comparación entre las metodologías más reconocidas considerando el cumplimiento de los criterios expuestos y se analizaron los resultados obtenidos, lo que permitió considerar las características de cada tipo de propuesta. Con esta comparativa se pudo verificar que las metodologías más potentes tienden a abarcar distintos elementos de todas las clasificaciones de metodologías, pues así se toman en cuenta muchos más aspectos de la web que otras propuestas que se centran en brindar solución a un tipo de problema específico.

Palabras clave: Ingeniería web, metodologías, modelado web, criterios de comparación

Abstract: Within the web engineering a series of methodologies for web applications modeling has been developed that aim to solve different problems in the development of such software. For the software researchers and developers are not trivial to identify and understand the differences between the two methods for the large number of existing proposals and the complexity of them. In view of this, this paper presents cross-cutting and specific criteria to allow comparison of methodologies in order to recognize the differences that exist between them. With these criteria, a comparison was made between the recognized methodologies considering compliance with the criteria and analyzed the results obtained, which allowed us to consider the characteristics of each type of proposal. With this comparison it was observed that the most powerful methodologies tend to include different elements from all classifications of methodologies, as well as take into account many more aspects of the web that other proposals that focus on providing solution to a specific type of problem.

Keywords: Web engineering, methodologies, web modeling, comparison criteria

1. Introducción

En la actualidad existen diversas metodologías para el modelado de aplicaciones web, algunas muy parecidas, otras completamente diferentes, como también se pueden encontrar algunas que apuntan hacia resolver un solo tipo de los problemas presentes en el modelado de este tipo de aplicaciones, de la misma forma hay otras que buscan abarcar mucho más. Se puede decir que en el mundo del modelado, de ingeniería web, hay una gran heterogeneidad entre las metodologías existentes, dependiendo del enfoque de cada una. Además, existiendo tanta literatura tras este tema puede resultar engorroso reconocer las fortalezas y debilidades de las metodologías al verse enfrentadas ante una situación de elección, más aún si no se tienen grandes conocimientos de los avances logrados en esta área. Para realizar esta tarea, este artículo presenta criterios apropiados a las metodologías que permitan distinguir las características únicas y distintas del resto de las metodologías. Estos criterios permitirán la diferenciación de las metodologías según el tipo de problema que buscan resolver, es decir, son transversales, independiente de la clasificación que las metodologías reciban. Además, se presentan criterios específicos que permiten hacer la diferenciación entre las metodologías que persiguen objetivos comunes. Así se realiza una comparación completa, desde un nivel macro hasta un nivel micro.

La comparación de algunas metodologías existentes para el modelado de aplicaciones web se ha realizado en otros trabajos que han sido consultados para la realización de este artículo. Así en (Cachero, 2003) se exponen criterios, bien justificados, de alto nivel para poder comparar algunas metodologías. Al ser sólo una comparación general no se entrega la información necesaria para diferenciar las metodologías que pertenecen a una misma clasificación. Asunto que si fue tratado en (Melía, 2007), donde se realiza una comparación con criterios específicos para cada conjunto de metodologías, logrando un análisis más detallado para encontrar la diferenciación de metodologías envueltas en un mismo grupo de propuestas. A pesar de ello, los criterios allí expuestos no son justificados, ni explicados, simplemente se realiza la comparativa con dichos criterios.

A continuación se presentarán las siguientes secciones de este artículo. En la sección dos se realiza la comparativa, considerando la especificación de criterios transversales, criterios específicos y la comparación de las metodologías. En la sección tres se realiza el análisis de los resultados y discusión. Finalmente, en la sección cuatro se realizan las conclusiones pertinentes sobre el presente artículo.

2. Comparativa

En el último tiempo han surgido un gran número de metodologías de modelado desarrolladas especialmente para la creación de aplicaciones web, debido a la aparición de nuevos problemas que se agregan a los ya existentes en el desarrollo de software clásico. Así a la falta de trazabilidad entre el modelado y el resultado final, problemas de validación y pruebas, etc. que se encuentran en los desarrollos de software, se pueden encontrar problemas específicos de la plataforma web como por ejemplo: interdependencia entre dominio, navegación y presentación, falta de organización de los datos, diseño descuidado, etc. Este gran número de metodologías han sido agrupadas en tres conjuntos, dependiendo del enfoque de cada una: propuestas dirigidas por la arquitectura, las que se enfocan en toda la arquitectura que apoya la aplicación web; propuestas basadas en el desarrollo dirigido por modelos, tal como su nombre lo indica, este grupo se basa en la ingeniería dirigida por modelos, *Model-Driven Development* (MDD), llevado a la ingeniería web; y las propuestas funcionales, basadas en el cumplimiento de las tres vistas funcionales descritas por (Retschitzegger, 2000), dominio, navegación y presentación.

2.1 Especificación de criterios

Teniendo en cuenta los aportes realizados por Cachero (2003) y Melía (2007) se buscarán unir los criterios generales y específicos presentados en dichos trabajos de forma que no sean redundantes y, además, se agregarán otros criterios que se consideran relevantes para la realización de una comparativa que ayude a la identificación eficaz del conjunto de características que vuelve particular a cada una de las metodologías analizadas. Con ello se establecen dos tipos de criterios, transversales y específicos, los que son detallados a continuación.

2.1.1 Criterios Transversales

Los criterios transversales se basan en los presentados Cachero (2003), donde también se realiza una comparación general de metodologías. Por esto último, es que se han seleccionado los criterios más relevantes de ese trabajo y no todos, pues muchos de esos criterios podrían provocar algún tipo de redundancia o confusión al ser integrados con los criterios específicos, que se presentarán más adelante. Además, considerando el tiempo que ha pasado la realización del trabajo citado es que se han agregado otros criterios que pueden resultar interesantes al momento de llevar a cabo una comparativa, pues es un aspecto que hoy en día está muy en boga, como lo es la calidad. Cabe destacar que los tres primeros criterios, que serán presentados, buscan reconocer el cumplimiento de las dimensiones expuestas en Retschitzegger (2000). Este trabajo se considera de gran importancia en la literatura, pues las tres dimensiones que presenta son aspectos fundamentales para las aplicaciones web. Con este conjunto de criterios se podrá realizar una interesante comparación entre las metodologías de modelado, sin importar cuál es su objetivo o clasificación a la que pertenece, permitiendo conocer aspectos comunes entre aquellas que pertenecen a un mismo grupo y las diferencias existentes con las de otros grupos.

Los criterios a utilizar en la comparativa son los siguientes:

- Separación de niveles: reconocer si la metodología permite la separación de niveles en el modelado. Es fundamental para determinar la aproximación y flexibilidad que puede obtener del modelado al llevarlo hacia el desarrollo de la aplicación. Así se sabrá si la metodología presenta algún tipo de soporte para representar el dominio (contenido), la composición lógica de la información (hipertexto) y/o la visualización (presentación). Con este criterio se puede evaluar la dimensión de niveles.
- Elementos especiales de la web. Es necesario saber si la metodología permite modelar componentes propios de las aplicaciones web para tener en vista esto desde una fase temprana y no sobre la marcha. Esto es posible conociendo los elementos estructurales (forma visual) y el comportamiento que éstos tendrán ante la interacción con el usuario de la aplicación. Esto puede verse reflejado en el uso de patrones de diseño para la representación gráfica o para soluciones de usabilidad. Así se puede conocer si la metodología cumple con la dimensión de aspectos.
- Ciclo de vida – proceso: se requiere conocer qué etapas del ciclo de vida de la aplicación web son consideradas por la metodología de modelado, para conocer su alcance en el desarrollo. También se debe conocer si la metodología permite la utilización de algún proceso de desarrollo, sea conocido o de autoría propia de la metodología. Este criterio permite fijar la dimensión fases.

- Uso de estándares: el empleo de estándares reconocidos y aceptados para la definición de la semántica en los modelos facilita la utilización, el manejo y la comprensión que se puede tener sobre éstos, aún si son extensiones propias de la metodología, al ser reconocidos naturalmente por el equipo de trabajo.
- Herramientas para la implementación: para llevar a la práctica exitosamente la utilización de las metodologías se debe contar con alguna herramienta que proporcione soporte para ellas, ya sea gracias a extensiones para plataformas estándar que permita soportar la notación o mediante herramientas propias que se han creado en base a aplicaciones ya conocidas.
- Calidad: cualquier tipo de aplicación informática debe satisfacer a quienes la utilizan, cumpliendo con sus requerimientos y dando solución a sus necesidades. Esto es aún más importante en las aplicaciones web, pues el abanico de usuarios que podrían utilizar dicha aplicación es mucho más amplio y la gran parte de ellos, probablemente, no este considerado como un actor del sistema. Para cumplir con esto las metodologías de modelado deben permitir la inclusión de aspectos de calidad. En el presente se tomarán como aspectos de calidad presentados en la ISO/IEC 9126 Estos son: funcionalidad, fiabilidad, usabilidad, eficiencia, portabilidad y mantenibilidad.

2.1.2 Criterios Específicos

Pasando a un mayor nivel de detalle, los criterios específicos permitirán elaborar una comparación más detallada entre metodologías de modelado pertenecientes a una misma clasificación, a través del reconocimiento de los elementos característicos de cada propuesta que deben ser abarcados por las metodologías analizadas. Estos criterios están basados en los presentados por Melía(2007). Sin embargo, los criterios aquí presentados serán detallados y adaptados para complementar a los criterios transversales, para no caer en redundancias al momento de realizar la comparativa.

- Criterios para propuestas dirigidas por la arquitectura:
 - Vistas de arquitectura: la metodología debe poder modelar las cuatro vistas de la arquitectura presentadas por Kruchten (1995).
 - Patrones: los patrones son un elemento fundamental en el modelado de arquitecturas, por ello es necesario que la metodología soporte la incorporación de éstos al modelado de la aplicación.
 - Papel en el proceso de desarrollo: dada la poca heterogeneidad presente en las metodologías de esta clasificación es necesario reconocer la función que desempeña la metodología en el proceso de desarrollo de la aplicación web.
- Criterios para propuestas dirigidas por modelos:
 - Transformaciones de PIM a PSM: comprobar que la metodología permite la transformación de los modelos independientes de la plataforma hacia los específicos de la plataforma a través de los mapas y marcas de transformación según los descritos por Miller (2003).
 - Transformaciones de PSM a código: es muy importante saber si la metodología permitirá generar código ejecutable a través de la transformación de los modelos específicos de la plataforma.
 - Estándares MDA: es necesario que la metodología cumpla con los estándares MDA establecidos por la OMG (Miller, 2003).

- Criterios para propuestas funcionales
 - Notación del dominio: es necesario que la metodología cuente con herramientas para poder modelar los elementos propios del dominio de la aplicación.
 - Notación de navegación: debe ser posible representar la composición lógica y las estructuras de navegación de las páginas web.
 - Notación de presentación: para poder cumplir con los aspectos generales de las propuestas funcionales se requieren componentes que permitan modelar la representación grafica que tomará el hipertexto.

2.2 Comparación de Metodologías

La realización de la comparación se llevará a cabo en dos etapas. La primera aplicará los criterios transversales a seis metodologías, dos por cada clasificación. La segunda etapa, comparará cuatro metodologías a través de los criterios específicos que se han establecido para cada clasificación. Los resultados de las comparaciones se presentan a continuación.

Las Tablas 1 y 2 presentan los criterios transversales aplicados a: WAE (Conallen, 2002), REST (Fielding, 2000), MIDAS (Cáceres *et al.*, 2003), WebSA (Melía, 2007), OOHDM (Schwabe, 2005) y UWE (Koch y Kraus, 2002). La primera toma los criterios de: separación de niveles, elementos especiales de la web, ciclo de vida – proceso, uso de estándares y herramientas de implementación. La segunda tabla considera los seis aspectos de calidad.

A las metodologías ya presentadas, en las tablas 1 y 2, pertenecientes a las propuestas dirigidas por la arquitectura, WAE y REST, se agregan Architecture Recovery (Hassan y Hols, 2002) y WAM (Meinecke *et al.*, 2005) en la tabla 3 (sección a) para la comparación de propuestas dirigidas por la arquitectura usando criterios específicos.

Tabla 1: Comparativa de metodologías a través de criterios transversales

Metodología	Separación de niveles	Elementos especiales de la web	Ciclo de vida-proceso	Uso de estándares	Herramientas de implementación
WAE	NO	NO	Ciclo completo-RUP	UML	SI
REST	NO	NO	Diseño-ninguno	NO	SI
MIDAS	SI	SI	Análisis, diseño, implementación-ninguno	MDA	SI
WebSA	SI	SI	Ciclo completo-UP	MDA	SI
OOHDM	SI	SI	Análisis, diseño, implementación-iterativo	UML	SI
UWE	SI	SI	Ciclo completo-UP	UML	SI

Tabla 2: Comparativa de metodologías a través del criterio transversal Calidad

Metodología	Calidad					
	Funcionalidad	Fiabilidad	Usabilidad	Eficiencia	Portabilidad	Mantenibilidad
WAE	Adecuación Interoperabilidad Seguridad	Ninguna	Ninguna	Ninguna	Adaptabilidad Coexistencia	Ninguna
REST	Seguridad	Capacidad de recuperación	Ninguna	Utilización de recursos	Adaptabilidad Coexistencia	Ninguna
MIDAS	Adecuación Interoperabilidad	Ninguna	Ninguna	Ninguna	Adaptabilidad	Ninguna
WebSA	Adecuación Interoperabilidad Seguridad	Ninguna	Capacidad para ser operado	Ninguna	Adaptabilidad	Capacidad de ser cambiado y probado
OOHDM	Adecuación Interoperabilidad	Ninguna	Capacidad para ser entendido, aprendido y operado	Ninguna	Adaptabilidad	Ninguna
UWE	Adecuación Interoperabilidad	Ninguna	Capacidad para ser entendió, aprendido y operado	Ninguna	Adaptabilidad	Ninguna

Además de las metodologías pertenecientes al conjunto de aquellas dirigidas por modelos presentadas, MIDAS y WebSA, se agregan en la tabla 3 (sección b): TAI (Tai *et al.*, 2004) y MDD-UWE (Koch, 2006), para realizar la comparativa específica a este grupo de propuestas.

Tabla 3: Comparativa de metodologías a través de criterios específicos

(a) Comparación para Propuestas Dirigidas por Arquitectura			
Metodología	Vista de la Arquitectura	Patrones	Papel en el Proceso de Desarrollo
WAE	Lógica, procesos, física y desarrollo	SI	Dirigir el proceso
REST	Lógica, procesos y física	NO	Guiar el diseño
ARCHITECTURE RECOVERY	Lógica y procesos	NO	Documentar sistemas legados
WAM	Lógica, procesos y física	NO	Dirige el proceso
(b) Comparación para Propuestas Dirigidas por Modelos			
Metodología	Transformación de PIM a PSM	Transformación de PSM a código	Estándares de MDA
MIDAS	NO	SI	UML, MOF
WEBSA	SI	SI	UML, MOF, OCL, XML, MOFScript
TAI	NO	SI	UML
MDD-UWE	SI	PARCIAL	UML, MOF, OCL, QVT
(c) Comparación para Propuestas Funcionales			
Metodología	Notación del Dominio	Notación de Navegación	Notación de Presentación
OOHDM	Propia	Propia	Propia
UWE	UML	UML	UML
OOH	UML	UML y propia (DNS, DAN)	XML (DPA)
W2000	UML	UML	UML

Terminando con el ejercicio comparativo, a las propuestas funcionales expuestas en la comparación transversal, OOHDH y UWE, se agregan en la tabla 3 (sección c): OOH (Cachero, 2003) y W2000 (Miller, 2003) para realizar la comparativa específica a este grupo de propuestas.

3. Resultados y discusión

Para proporcionar un mejor entendimiento de las tablas recién presentadas se realizará un pequeño análisis sobre los resultados obtenidos de la comparativa.

En la Tabla 1 los criterios indican, de cierta forma, a qué categoría pertenece cada metodología. Así WAE y REST no presentan separación de niveles, ni elementos especiales de la web, debido a que están orientados hacia la arquitectura de la aplicación web, REST presenta un orientación mucho más fuerte que WAE en este aspecto, pues solamente cubre el diseño de la arquitectura. También se puede apreciar que las metodologías que apuntan hacia el desarrollo guiado por la transformación de modelos ocupan los estándares MDA y abarcan de mayor manera el ciclo de vida, además de poder integrar un proceso de desarrollo, como ocurre con MIDAS y WebSA. Las propuestas funcionales, OOHDH y UWE, son las más comunes dentro de las metodologías de desarrollo, así cumplen con la separación de niveles (esto es parte de la definición de las propuestas funcionales), elementos especiales de la web, ciclo de vida, proceso de desarrollo y usan como estándar UML. Cabe destacar que todas las metodologías expuestas a la comparación poseen herramientas CAWE.

Examinando la Tabla 2, en líneas generales se puede apreciar la evidente falta de elementos para modelar aspectos de calidad, esto podría explicarse debido a que estos aspectos son, en su mayoría, parte de requisitos no funcionales y, por lo tanto, difíciles de integrar a un modelo. Aún así, existen elementos comunes que se cumplen dentro de las metodologías analizadas. En aspectos de funcionalidad es claro que aquellas metodologías que utilizan UML pueden llevar a cabo el modelado de adecuación e interoperabilidad a través de los casos de uso, como también es claro que las metodologías enfocadas hacia la arquitectura, WAE y REST, presentan elementos para modelar la seguridad. Los atributos de fiabilidad son los más complicados de modelar, pues no existen herramientas para representar la tolerancia de fallos o la capacidad del software para evitar fallar. Sólo en REST se toma en cuenta la capacidad de recuperación. En cuanto a la usabilidad era de esperar que aquellas metodologías dirigidas hacia la arquitectura no tuviesen en consideración este punto, mientras que las dirigidas por los requisitos funcionales si tuviesen elementos para ello (mapas navegacionales, uso de patrones, etc.), fuera de estas dos metodologías, WebSA también presenta elementos para la utilización de patrones aplicables a interfaces web. Uno de los atributos que menor representatividad tiene en las metodologías es la eficiencia, donde solamente REST presenta elementos para acotar la utilización de recursos utilizados. Ya que se están estudiando metodologías de modelado, en los aspectos de portabilidad, la adaptabilidad es común para todas ellas, aún para las aquellas dirigidas por modelos pues la selección del lenguaje a que quieren ser transformados los modelos es parte de proceso que se debe llevar a cabo. Finalmente, los aspectos de mantenibilidad también son poco modelados, sólo WebSA considera la utilización de casos de uso para guiar las pruebas.

En la tabla 3 (sección a) se expone la comparativa entre las propuestas metodológicas dirigidas por la arquitectura. Así se puede derivar que WAE y WAM son las metodologías más robustas de este grupo, pues dirigen el proceso y cubren la mayor parte de la vista arquitectónica, sobre éstas WAE es la más completa al permitir uso de patrones y contemplar las 4 vistas de la arquitectura. Cabe destacar a *Architecture Recovery*, en comparación a las demás

metodologías presentadas, ya que puede parecer la más débil, pero esta metodología solamente busca recuperar la documentación de los sistemas, mientras las otras buscan cumplir con más objetivos.

De la comparación realizada en la tabla 3 (sección b), a las metodologías dirigidas por modelos, se aprecia el parecido que existe entre MIDAS y TAI, dado su origen común, donde TAI es la metodología que menos estándares MDA utiliza, mientras la más completa, en este aspecto, es WebSA. Así también, ésta última es la única metodología que presenta ambos tipos de transformaciones, MDD-UWE es quien le sigue brindando transformación PIM a PSM y una transformación parcial a código, mientras MIDAS y TAI no realizan el primer tipo de transformación.

Finalmente la comparación de metodologías funcionales, Tabla 3 (sección c), se observa la homogeneidad de la notación utilizada por las metodologías, gran parte UML, donde solamente OOHDMM utiliza notación propia. Dentro de todos los conjuntos de metodologías, las propuestas funcionales son las que presentan una mayor cantidad, gracias a la comparación se puede notar que las aquí analizadas son muy parecidas, donde UML es el factor principal para esta igualdad en las características estudiadas. Esto puede deberse al efecto positivo que presenta UML como estándar para el modelado, especialmente dirigido hacia la funcionalidad donde la correcta utilización de diagramas (como casos de uso) y el agregado de extensiones para incorporar elementos web puede entregar una herramienta potente para el modelado de este tipo de aplicaciones.

4. Conclusiones

En el presente artículo se han tomado como base trabajos anteriores que realizaban comparativas de metodologías de modelado incompletas, pero que en conjunto podrían ser de gran utilidad para el público interesado en el tema. Así se tomaron algunos de los criterios presentados en los trabajos base y fueron adaptados para que el ejercicio comparativo no entregase información redundante. Además, se integraron criterios que se consideran de importancia en el ambiente actual de desarrollo de cualquier tipo de software. Se realizó la comparación entre las metodologías más reconocidas, relevando sus igualdades y diferencias, a través de los criterios transversales y los específicos. Teniendo esto presente, se pueden tomar los aspectos estudiados para la mejora de las metodologías, incorporando elementos de otras metodologías o elementos nuevos que no han sido tomados en cuenta por ninguna propuesta. También es posible que desarrolladores busquen nuevas formas para la realización de un proyecto de desarrollo de alguna aplicación web y, ante esta comparativa, puedan dilucidar cuál metodología es más útil para sus objetivos. Finalmente, con esta comparativa podemos dilucidar que las metodologías más potentes, tienden a abarcar distintos elementos de todas las clasificaciones de metodologías, pues así se toman en cuenta muchos más aspectos de la web que otras propuestas que se centran en brindar solución a un tipo de problema específico.

Referencias

- BARESI, L., GARZOTTO, F., & PAOLINI, P. Extending UML Modeling Web Applications. *Proceedings of the 34th International Conference on System Sciences*. 2001.
- CÁCERES, P., MARCOS, E., & VELA, B. A MDA-Based Approach for Web Information System Development. *Workshop in Software Model Engineering*. 2003.
- CACHERO, C. *Tesis Doctoral: OO-H Una Extensión a los Métodos OO para el Modelado y Generación Automática de Interfaces Hipermediales*. Departamento de Lenguajes y Sistemas Informáticos, Universidad Alicante, 2003.

CONALLEN, J. *Building Web Applications with UML*. (2da). Addison Wesley Longman, 2002.

FIELDING, R. *Architectural Styles and the Design of Network-based Software Architectures*. 2000. [Disponible en: <http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>].

HASSAN, A., & HOLT, R. Architecture Recovery of Web Applications. *24th International Conference on Software Engineering*. 2002.

KOCH, N. *TRANSFORMATION Techniques in the Model-Driven Development Process of UWE*. *2nd International Workshop in Model-Driven Web Engineering 2006*. *6th International Conference on Web Engineering 2006*.

KOCH, N., & KRAUS, A. (2002). The Expressive Power of UML-based Web Engineering. *Workshop on Web-oriented Software Technology*. 2002.

KRUCHTEN, P. *4+1view-architecture*. IEEE Computer Society Press. 1995.

MEINECKE, J., GAEDKE, M., & NUSSBAUME, M. *A Web Engineering Approach to Model the Architecture of Inter-Organizational Applications*. COEA '05. 2005-

MELIÁ, S. *Tesis Doctoral: WebSA: Un Método de Desarrollo Dirigido por Modelos de Arquitectura para Aplicaciones Web*. Departamento de Lenguajes y Sistemas, Universidad Alicante, 2007.

MILLER, J., & Mukerji, J. *MDA Guide Version Document Number omg/2003-05-01*. 2003. <http://www.omg.com/mda>.

RETSCHITZEGGER, W., & SCHWINGER, W. Towards Modeling of DataWeb Applications - A Requirements Perspective. *Proceedings of the American Conference on Information Systems*, 2000..

SCHWABE, D. (2005). *Summary of OOHDM*. 2005. [Disponible en; <http://www.tecweb.inf.pucrio.br/oohdm/space/summary+of+OOHDM>].

TAI, H., MITSUI, K. I., NEROME, T., ABE, M., ONO, K., & HORI, M. Model-Driven Development of Large-Scale Web Applications. *IBM Journal. Research. & Development*. Vol. 48. 2004-