

# Una Aproximación de Metamodelado para la Evaluación de Calidad en Procesos de Desarrollo Web

Cristina Cachero  
DLSI, Universidad de Alicante  
Apartado de Correos 99. E-  
03080 Alicante, España  
ccachero@dlsi.ua.es

Emilio Insfran, Silvia Abrahão  
DSIC, Universidad Politécnica de  
Valencia  
Camino de Vera, s/n, 46022, Valencia,  
España  
{einsfran, sabrahao}@dsic.upv.es

Geert Poels  
Faculty of Economics and Business  
Administration, Ghent University  
Hoveniersberg 24, 9000 Ghent,  
Bélgica  
geert.poels@ugent.be

## Resumen

En los últimos años varios métodos de desarrollo Web han sido propuestos. Sin embargo, el uso sistemático de estos métodos no ha resuelto el problema de la industria. Las aplicaciones Web siguen siendo entregadas con errores significativos que causan altos costes de mantenimiento. Uno de los motivos es la falta de técnicas de evaluación de calidad que puedan ser integradas en las distintas fases del proceso de desarrollo Web. En este artículo, se introduce una aproximación para la evaluación de calidad en procesos de desarrollo Web dirigido por modelos. En particular, se presenta la evaluación de un Modelo de Navegación con respecto a un Modelo de Requisitos para determinar el grado de navegabilidad de una aplicación Web con respecto a los requisitos de usuario. La evaluación de calidad ha sido automatizada mediante la inclusión en el método de (1) modelos de medición que personalizan modelos de calidad genéricos y (2) transformaciones de calidad que actúan sobre los metamodelos de los artefactos que deben ser evaluados.

## 1. Introducción

En los últimos años se han propuesto varias definiciones de calidad. En líneas generales, estas definiciones dan soporte a distintas perspectivas [24]. Entre ellas, dos resultan especialmente relevantes para el presente trabajo:

- Conformidad a la especificación: calidad definida en base a las características medibles de los productos que deben satisfacer una especificación previa.
- Satisfacción de las necesidades de los usuarios: calidad definida como la capacidad

del producto para satisfacer las expectativas de los usuarios, independientemente de cualquier característica propia del producto.

Según el ISO/IEC 9126 [12] y el ISO/IEC 14598 [13] es esta segunda perspectiva, esto es, la satisfacción de las necesidades (explícitas o implícitas) de los usuarios, la que constituye el objetivo final de cualquier proceso de evaluación de calidad. Puesto que el usuario sólo tiene interés en los resultados del proceso (la aplicación Web final y/o sus variables asociadas, tales como coste total o tiempo de desarrollo total), el grado de cumplimiento de tales necesidades se debe establecer mediante una inspección del producto una vez que este se haya entregado. Sin embargo, se ha comprobado empíricamente que el coste de realizar esta detección y corrección de error tardía es más de 100 veces superior al coste de solventar los mismos errores en la fase de requisitos [18]. Por lo tanto, sería deseable que la evaluación de calidad se realizase en etapas más tempranas del proceso de desarrollo, para lo cual sería necesario adoptar la perspectiva de calidad como 'conformidad a la especificación'.

Afortunadamente, tanto el conjunto de estándares de calidad ISO como algunas investigaciones empíricas recientes establecen que existe una relación entre estas dos perspectivas, y que de hecho ciertas características del software pueden ser usadas como predictores de la satisfacción del usuario con la aplicación.

En este artículo, introducimos una aproximación para la evaluación de calidad en procesos de desarrollo Web dirigido por modelos [3]. Las actividades de evaluación de calidad se realizan en cada fase del proceso desde los requisitos hasta la obtención del producto software final. Esto implica un cambio de la

perspectiva tradicional de evaluación de calidad, que pasa de estar basada en la aplicación final a basarse en una evaluación de calidad sucesiva de los distintos modelos obtenidos como parte del proceso de desarrollo. De esta forma, el foco de la evaluación está en la prevención en lugar de la detección de errores.

Esta aproximación de evaluación de calidad requiere la conversión de modelos de calidad a distintos niveles de abstracción en modelos de medición, manipulables de manera automática. Para ello, estos modelos de medición se definen como instanciaciones de un metamodelo de medición de software soportado a su vez por una ontología [8]. El proceso de evaluación se realiza mediante transformación de modelos.

Aunque esta aproximación es independiente de proceso, en este artículo se ha utilizado el proceso de desarrollo de la metodología Web OO-H (*Object-Oriented Hypermedia*) [10]. En particular, se muestra la evaluación de calidad del modelo de navegación de OO-H con respecto a un Modelo de Requisitos. El objetivo de la evaluación es determinar el grado de navegabilidad de la aplicación Web con respecto a los requisitos de usuario.

Este artículo se estructura como sigue: la sección 2 presenta el estado del arte de la evaluación de calidad en la Web; la sección 3 presenta una visión general de la aproximación propuesta para la evaluación de calidad en procesos de desarrollo Web; la sección 4 presenta el proceso de desarrollo OO-H incorporando un Modelo de Requisitos; la sección 5 presenta la instanciación de la propuesta de evaluación de calidad a OO-H así como la automatización del proceso de evaluación; finalmente, la sección 6 presenta las conclusiones y trabajos futuros.

## 2. Evaluación de Calidad en la Web

Se han propuesto diversos modelos de calidad para la evaluación de productos software [16] [2] [12]. Debido al uso extendido de la familia de estándares ISO, varios de estos modelos son intentos, más o menos exitosos, de adaptar, refinar o mejorar dichos modelos de calidad ISO. Por ejemplo, Quint [22], Seffah [23] y Van Welie [25] proponen modelos de calidad que extienden el estándar ISO 9126 con características de calidad adicionales.

Además, la sensibilidad de los usuarios Web a los posibles errores de calidad en las aplicaciones ha motivado la aparición de propuestas de modelos de calidad específicos para la Web ([20] [5] [19] [6] [1]). El problema con estos modelos es que la mayor parte de ellos abordan la perspectiva, mencionada anteriormente, de *satisfacción de las necesidades de los usuarios*, y por tanto se centran en la aplicación Web una vez ésta ha sido implantada, descartando cualquier producto intermedio como candidato a la evaluación de calidad. Incluso WebQEM [20], una de las propuestas de evaluación de calidad Web más conocidas, asume un proceso de evaluación que se aplica únicamente a aplicaciones Web operativas. De hecho, sólo [6] y [1] consideran otros artefactos, aparte del código fuente, en el proceso de evaluación de calidad, aunque ninguna de las dos propuestas proporciona modelos de calidad específicos para estos artefactos. Por lo tanto, para sistematizar y aumentar el nivel de abstracción en el cual se mide la calidad, es necesaria la definición de marcos de calidad que puedan ser aplicados a los artefactos intermedios de un proceso de desarrollo Web. Este conjunto de artefactos intermedios incluye (aunque no está limitado a ellos) modelos de requisitos, de dominio, de navegación o de presentación.

En este contexto, trabajos recientes en calidad de modelos conceptuales [15] [18] proporcionan conceptos y marcos de alto nivel que pueden ser adaptados para tratar artefactos específicos de un proceso Web. Por ejemplo, los resultados empíricos de cómo la complejidad estructural de un diagrama de clases afecta su capacidad de mantenimiento [9] pueden ser usados a nivel de los modelos de navegación (que son diagramas de clase estereotipados).

Sin embargo, proporcionar estos marcos de calidad conceptual no es suficiente. Es necesario proporcionar además mecanismos específicos para su integración en el proceso de desarrollo Web de manera que la inclusión de aspectos de calidad no complique innecesariamente la labor del desarrollador. A continuación veremos cómo es posible plantear esta integración en el contexto de aproximaciones MDE (*Model Driven Engineering*) [14], con el objetivo de mantener el incremento de costes temporales y económicos asociados a la actividad de evaluación de calidad al mínimo.

### 3. Integración de Evaluación de Calidad en Procesos de Desarrollo Web

Un proceso de desarrollo Web basado en el paradigma MDE incluye, como mínimo, (1) un flujo de trabajo de modelado de requisitos, cuyo artefacto de salida es un Modelo de Requisitos, (2) un flujo de análisis, cuya salida es un Modelo de Dominio (un diagrama ER o un diagrama de clases UML), (3) un flujo de trabajo de diseño conceptual, cuya salida es un Modelo de Navegación y un Modelo de Presentación (expresados mediante *profiles* de UML o notaciones propietarias) (4) un flujo de diseño detallado que introduce aspectos específicos de plataforma y tecnología (normalmente J2EE o .NET) y (5) un flujo de implementación, que permite generar un interfaz de usuario que puede ser integrado con el resto de capas arquitecturales de la aplicación. Este proceso genérico se muestra gráficamente en la Figura 1A. Las variaciones más habituales de este proceso consisten en la adición de nuevos flujos y modelos independientes y/o dependientes de plataforma (modelos arquitecturales, modelos de proceso de negocio, soporte para distintas plataformas y/o lenguajes, etc.) que permiten enriquecer la especificación de la aplicación. Por otro lado, con el fin de adaptarse al paradigma MDE, la mayor parte de las metodologías Web promueven explícitamente la definición de metamodelos y el uso de transformaciones automáticas o semi-automáticas para agilizar el proceso y garantizar la trazabilidad entre conceptos y artefactos. La viabilidad de estos procesos automatizados ha sido confirmada por herramientas como WebRatio ([www.webratio.com](http://www.webratio.com)) y VisualWADE ([www.visualwade.com](http://www.visualwade.com)), que pueden generar aplicaciones Web a partir de modelos conceptuales. El uso de un proceso semi-automático como el presentado en la Figura 1A evita algunos problemas de desarrollo, como pueden ser inconsistencias entre modelos, falta de trazabilidad, etc. Sin embargo, esta automatización también puede causar la propagación de defectos de calidad a través de los distintos niveles de abstracción. Por ejemplo, un diseño pobre de caminos de navegación a través del espacio de información causa una pobre estructuración de la interfaz final generada, lo que probablemente disminuya el nivel de calidad en

uso percibido por el usuario de la aplicación, incluso si el resto de actividades (requisitos, dominio, presentación, implementación, etc.) del proceso se han realizado de manera correcta. Con el fin de evitar este problema, se propone un proceso de desarrollo Web que incorpora dos artefactos de calidad (ver Figura 1B):

- **Modelos de medición** que contienen la información concreta (medidas, criterios de decisión etc.) que permite evaluar los artefactos obtenidos en cada nivel de abstracción. Esta información es una adaptación de la contenida en el modelo de calidad asociado a ese nivel de abstracción, con el fin de adecuar el modelo de calidad genérico a la aplicación/dominio particular con los que se está trabajando.
- **Transformaciones de calidad**, que permiten preservar la naturaleza semi-automática del proceso de evaluación y por tanto no penalizan su productividad.

El modo en que se realiza la integración de actividades de desarrollo y de evaluación en el contexto de una aproximación MDE garantiza que cada problema es detectado y resuelto tan pronto como sea posible en el ciclo de vida de desarrollo sin penalizar la productividad del desarrollador. De este modo se persigue el objetivo de disminuir los costes y el tiempo de desarrollo de aplicaciones Web con un cierto nivel de calidad. La definición de modelos de calidad específicos para cada tipo de artefacto reconoce implícitamente la existencia de distintos propósitos de evaluación. Por ejemplo, durante la fase de requisitos el propósito de evaluación de calidad será comprobar que se han capturado de manera completa y correcta los requisitos de usuario. Durante la fase de modelado de navegación podremos por tanto asumir que los requisitos han sido capturados correctamente, y centrarnos en comprobar aspectos como por ejemplo que la topología de la interfaz no sea demasiado compleja o que no omita ningún vínculo entre conceptos que puede ser relevante para el usuario final. Esta evaluación de los productos intermedios tiene como objetivo no sólo predecir la calidad del producto final, sino también decidir cuándo un artefacto está listo para ser usado como entrada de la fase siguiente del proceso.

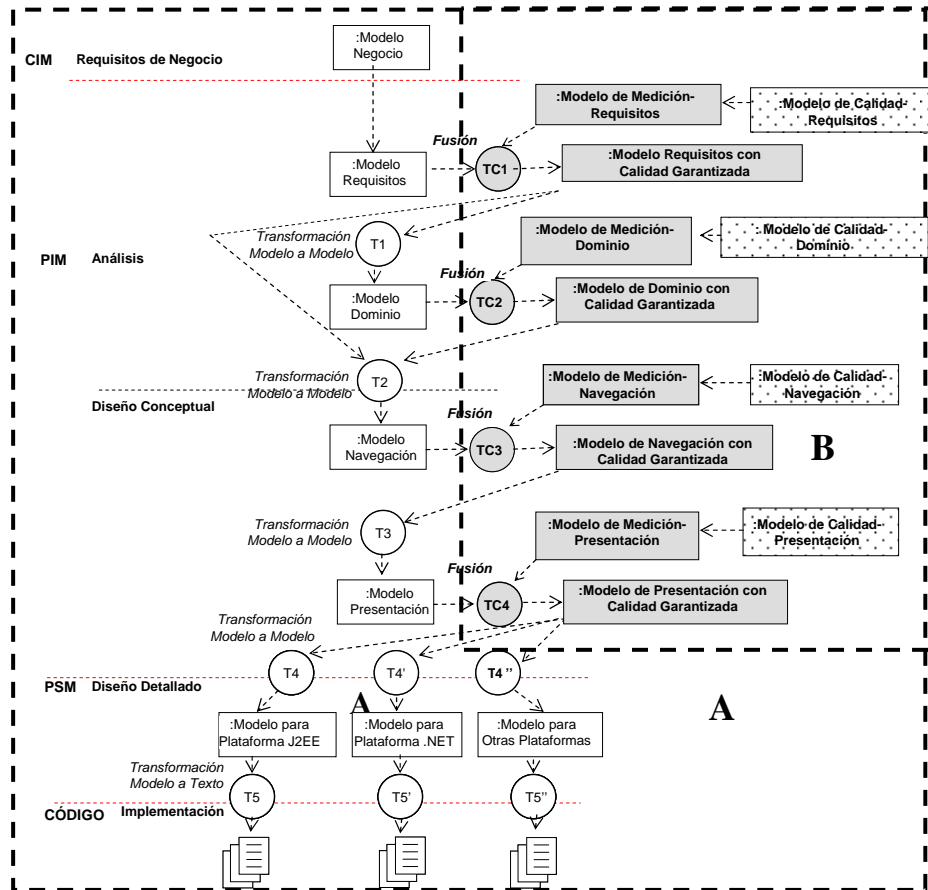


Figura 1. Proceso simplificado de Desarrollo de Interfaces Web. A engloba los artefactos y transformaciones relacionadas con los procesos de desarrollo de Ingeniería Web, mientras que B agrupa los elementos y el proceso de evaluación de la calidad.

#### 4. El proceso de desarrollo OO-H

Object-Oriented Hypermedia (OO-H) [10] es una metodología de ingeniería Web que provee a los diseñadores de un proceso y una notación para el desarrollo de interfaces de usuario Web y su conexión con módulos preexistentes de lógica de negocio. El desarrollo de aplicaciones Web con OO-H empieza con la fase de captura de requisitos para la construcción de un Modelo de Requisitos. La fase de análisis y diseño especifica la información estructural, de navegación y de presentación de la aplicación Web. La Figura 2

muestra una descripción general de la arquitectura de OO-H.

El Modelo de Requisitos [11] incluye las primitivas para la organización y especificación de los requisitos. Siguiendo una estrategia de descomposición *top-down*, los requisitos del sistema son especificados sin considerar los aspectos tecnológicos del sistema software a desarrollar. El modelo de requisitos está compuesto por la Misión del Sistema, un Árbol de Refinamiento de Funciones, un Modelo de Casos de Uso, un Modelo de Dominio de Requisitos y un Modelo de Interacción.

La Misión del Sistema describe el propósito del sistema en una o dos sentencias. Considerando en este momento el sistema como una caja negra,

las interacciones externas son identificadas como funciones. Estas funciones son jerárquicamente estructuradas y organizadas en un **Árbol de Refinamiento de Funciones (ARF)** donde la raíz de este árbol es la misión del sistema, los nodos intermedios son grupos de funciones y las hojas son las funciones elementales (interacciones externas). Este concepto de función es equivalente al de concepto de caso de uso. El Modelo de Casos de Uso (MCU) incluye todas las funciones identificadas, las interacciones externas que son los casos de uso, así como las relaciones de comunicación entre actores (E, S, E/S) y las relaciones estructurales entre actores (herencia) si las hubiera. Finalmente, la especificación de requisitos en OO-H se completa con el Modelo de Dominio de Requisitos (MDR) y el Modelo de Interacción. El Modelo de Dominio de Requisitos define el vocabulario del dominio incluyendo las entidades del dominio del problema (conceptos) y sus relaciones estructurales. Utiliza la misma notación que el diagrama de clases sin considerar atributos ni operaciones para las clases y sólo relaciones de asociación y herencia como relaciones estructurales.

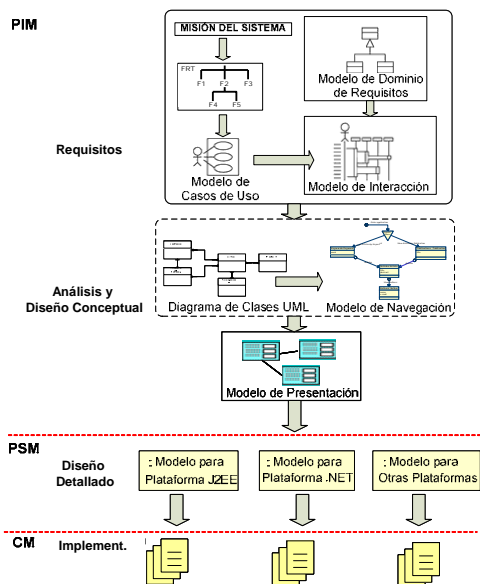


Figura 2. El proceso de desarrollo de OO-H

El Modelo de Interacción captura las interacciones entre objetos necesarias para realizar los casos de uso (debe haber al menos un diagrama de

interacción por cada caso de uso). Es en esta fase donde el analista indica de qué forma los objetos de las clases interactúan entre sí y de qué forma el caso de uso va a realizar el comportamiento esperado.

El flujo de trabajo de análisis y diseño conceptual produce como resultado tres vistas complementarias:

- Un Modelo de Dominio, que define la estructura estática de la aplicación usando la notación de diagramas de clases UML.
- Un Modelo de Navegación, que define los caminos que los usuarios pueden seguir a través del dominio de información para realizar sus objetivos. Está organizado con destinos de navegación (subsistemas) y enlaces. Los destinos de navegación encapsulan vistas definidas sobre clases del dominio y los enlaces son usados para interconectarlas.
- Un Modelo de Presentación, que define el aspecto de las páginas Web individuales.

Esta metodología está soportada por la herramienta VisualWADE ([www.visualwade.com](http://www.visualwade.com)) que incluye un compilador de modelos para la generación de la capa de datos, la lógica de los módulos y el interfaz del usuario en distintas plataformas.

## 5. Integración de la Evaluación de Calidad en OO-H

La evaluación de la calidad del Modelo de Navegación con respecto a su grado de cobertura del Modelo de Requisitos está conducida por la medida *Cobertura del Dominio del Modelo de Navegación (CDMN)* [4]. El fundamento de esta medida se basa en la suposición que los usuarios esperan encontrar en las aplicaciones Web las mismas relaciones entre conceptos que existen entre los conceptos del Modelo de Dominio del Modelo de Requisitos. Por tanto, no encontrar estos enlaces en la aplicación Web puede disminuir su satisfacción general durante el uso de la aplicación.

Para la evaluación automática de esta y otras medidas sobre el modelo de navegación, tal y como vimos en la Figura 1, es necesario partir de un *Modelo de Calidad de Navegación* como una

instancia del Metamodelo de Medición del Software para Ingeniería Web (WE-SMM), que es una simplificación del *Software Measurement Metamodel (SMM)* presentado en [7]. El WE-SMM asegura que el modelo de medición es definido de manera completa y consistente (de acuerdo con el significado semántico de sus elementos). Con el fin de facilitar su comprensión, en la Tabla 1 se presenta una breve descripción de los principales conceptos que componen este WE-SMM

Tabla 1. Definición de conceptos del WE-SMM

Término	Definición
Modelo de Calidad	Conjunto de conceptos medibles y relaciones entre ellos que proporcionan la base para especificar requisitos de calidad y evaluar la calidad de las entidades de una categoría de entidad determinada
Actor	Punto de vista que cubre el Modelo de Calidad
Necesidad de Información Global	Perspectiva global desde la que se manejan los objetivos, metas, riesgos y problemas
Necesidad de Información	Perspectivas parciales desde las que se manejan los objetivos, metas, riesgos y problemas
Concepto Medible	Relación abstracta entre atributos de entidades y necesidades de información
Entidad	Objeto que va a ser caracterizado mediante la medición de sus atributos
Categoría de Entidad	Colección de todas las entidades que satisfacen un predicado determinado
Atributo	Una propiedad de entidad medible, física o abstracta, que es compartida por todas las entidades de una categoría de entidad
Escala	Conjunto de valores con propiedades definidas
Tipo de Escala	Naturaleza de la relación entre valores en la escala
Unidad de Medida	Cantidad particular, definida y adoptada por convención, con la cual otras cantidades del mismo tipo son comparadas para expresar su magnitud en relación con esa cantidad.
Medida Base	Una medida de un atributo que no depende de ninguna otra medida, y que se mide mediante un método de medición
Medida	Medida que se deriva de otras

Derivada	medidas (bases o derivadas) mediante una función de medición
Indicador	Medida que se deriva de otras medidas mediante un modelo de análisis
Método de Medición	Secuencia lógica de operaciones, descritas de manera genérica, usadas para cuantificar un atributo con respecto a una escala específica
Función de Medida	Un algoritmo o cálculo realizado para combinar dos o más medidas base o derivadas.
Modelo de Análisis	Algoritmo o cálculo que combina una o más medidas con sus criterios de decisión asociados
Criterio de Decisión	Valores umbrales, objetivos o patrones usados para determinar la necesidad de más investigación, o para describir el nivel de confianza en un resultado dado

A continuación vamos a ejemplificar cómo este WE-SMM debe ser instanciado para reflejar la medida CDMN que hemos presentado anteriormente. Con el fin de evitar complicar innecesariamente el ejemplo, en este artículo supondremos que la medida CDMN es la única relevante para la aplicación a nivel de navegación. El modelo de medición que expresa, de una manera formal, dicha medida puede ser visto en la Figura 3. En este modelo, la medida provee información útil para satisfacer la **Necesidad de Información Global** “*Conocer la Navegabilidad de un Modelo de Navegación de una aplicación Web*” (*ConocerNavegabilidad*) del **Actor Usuario Final**. Esta medida está definida en el contexto de un **Modelo de Calidad** llamado **MCN (Modelo de Calidad de Navegación)**. Este modelo de calidad está definido para la **Categoría de Entidad Modelo Web** que es un modelo abstracto que compone todos los PIMs asociados con una aplicación Web. En concreto, incluye las **Categorías de Entidad: Modelo de Navegación (MN) y Modelo de Dominio de Requisitos (MDR)**.

El *Modelo de Calidad de Navegación (MCN)* evalúa el **Concepto Medible Navegabilidad**. Este concepto medible está relacionado con tres **Atributos: Cobertura de Dominio, Complejidad Estructural del MN y Complejidad Estructural del MDR**. De ellos, dos se relacionan con el Modelo de Navegación (Cobertura de Dominio y Complejidad estructural del MN), y el tercero con el Modelo de Dominio de Requisitos (**atributo Complejidad Estructural del MDR**).



Como resultado final, este Indicador nos indica si la Cobertura de Dominio del Modelo de Navegación es satisfactoria o no, esto es, qué nivel de navegabilidad tendrá nuestra aplicación Web con respecto a nuestros requisitos. De acuerdo con el criterio de decisión que hemos adoptado, al menos el 80% de las relaciones entre conceptos en el dominio del problema (especificado en el Modelo de Dominio de Requisitos) deben ser asociadas a pasos de navegación en el Modelo de Navegación para poder aceptarlo como válido.

La definición del modelo de medición como instancia del WE-SMM permite aplicar transformaciones para automatizar su aplicación a modelos de navegación concretos con el fin de evaluar dichos modelos. Siguiendo con nuestro ejemplo, a continuación vamos a mostrar una posible transformación de calidad a nivel de navegación (TC3, ver Figura 1) que permite verificar de manera automática el grado de cobertura que el Modelo de Navegación hace del Modelo de Dominio de Requisitos.

### 5.1. Definición de Transformaciones de Calidad en OO-H

La definición de nuestra propuesta como una

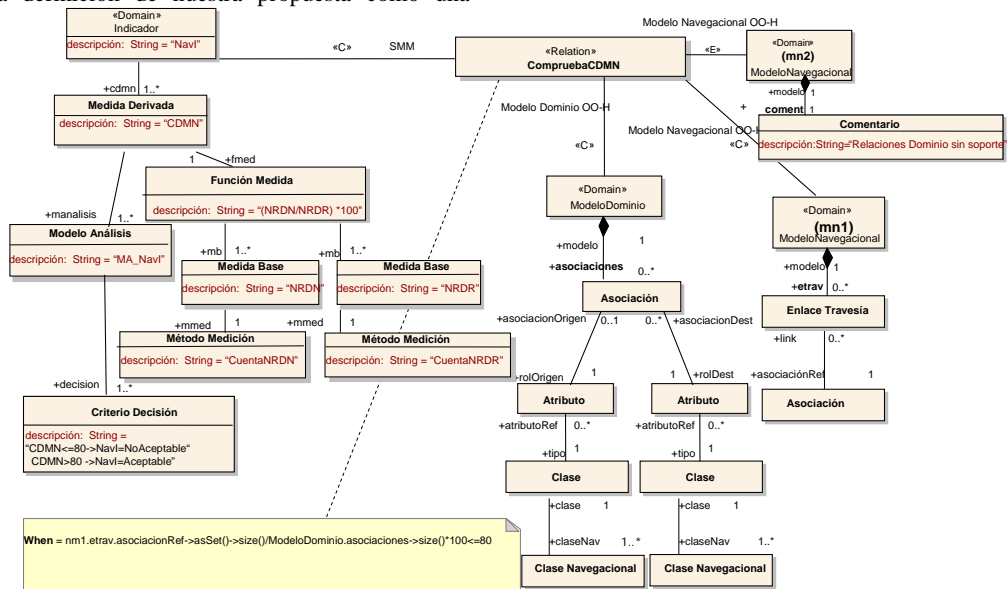


Figura 4. Regla de Evaluación *CheckDCNM*

aproximación MDE implica que la evaluación y evolución de los distintos modelos de la aplicación es formalizable y automatizable por medio de artefactos llamados transformaciones. Las transformaciones son modelos con las mismas capacidades que cualquier otro modelo del proceso de desarrollo. Un aspecto importante de estas transformaciones es que establecen claramente las relaciones de trazabilidad entre los conceptos que aparecen en los distintos modelos en las distintas fases del proceso de desarrollo.

Entre los lenguajes de transformación de modelos propuestos, el lenguaje Query-View-Transformations (QVT) [21], definido por la OMG, y en particular la modalidad QVT Relations, despierta un gran interés en el seno de la comunidad científica. Sin embargo, hasta la fecha no se dispone de herramientas que permitan ejecutar transformaciones expresadas en este lenguaje. Por esta razón, en el contexto de OO-H se ha optado por utilizar en su lugar UPT (UML Profile Transformation).

UPT es un lenguaje de transformación de modelos basado en QVT que define un metamodelo en conformidad con el estándar MOF de OMG, así como un perfil de UML. Este perfil extiende la semántica del diagrama de clases de UML.



UPT permite (i) una curva de aprendizaje baja para el modelado de transformaciones y (ii) el uso de herramientas de modelado UML para soportar la especificación de transformaciones. Como ventaja adicional, UPT está soportado por la herramienta WebSA [17]

Siguiendo con nuestro ejemplo, en la Figura 4 se muestra la especificación en UPT para la medida CDMN. Esta transformación automatiza el proceso de evaluación de la medida CDMN. Para ello, el primer paso es comprobar que la medida está presente en el modelo de medición, es decir, que es una medida relevante para la aplicación que está siendo evaluada (rama <<C>>-*check*-, ver parte izquierda de la Figura 4). Por otro lado, los diferentes componentes de la medida son calculados en las dos ramas de tipo *check* restantes. La función de medición y el criterio de decisión se codifica mediante OCL en la cláusula *When* de la transformación. Si el resultado de esta fórmula es verdadero (es decir, se viola el criterio de decisión establecido), la transformación modifica el Modelo de Navegación mediante la adición de una nota que indica que las relaciones de dominio no están suficientemente soportadas en el Modelo de Navegación ("*Relaciones Dominio sin soporte*", ver rama etiquetada con el estereotipo «E» (*enforce*)).

## 6. Conclusiones

En este artículo se ha presentado una aproximación para la evaluación de calidad que integra actividades de evaluación en las distintas fases de un proceso de desarrollo Web. La aproximación ha sido aplicada al proceso de desarrollo de OO-H para realizar la evaluación de calidad del modelo de navegación con respecto a un modelo de requisitos. La evaluación parte de un modelo de medición que, basado en un metamodelo, formaliza la definición de medidas, funciones de medida, indicadores, criterios de decisión, etc. que son aplicables a la aplicación que está siendo evaluada. La definición de modelos de calidad como instancias del metamodelo de medición del software tiene varias ventajas: i) proporciona un marco que determina cuál es la información que debe ser proporcionada para realizar la evaluación, ii) soporta el uso de módulos de evaluación puesto que la información se encuentra disponible de una manera homogénea, iii) soporta la reutilización de

módulos de evaluación y iv) soporta la estandarización de módulos de evaluación puesto que el formato está en conformidad a una ontología de medición.

Por otro lado, siguiendo el paradigma MDE, la evaluación se automatiza mediante reglas de transformación expresadas en el lenguaje de transformación de modelos UPT. Esta definición de un proceso de evaluación de calidad basado en el paradigma MDE también proporciona varias ventajas: i) facilita la automatización del proceso de evaluación de calidad, ii) reduce los costes y tiempos invertidos en tareas de evaluación. Finalmente, la integración de actividades de evaluación de calidad en los procesos de desarrollo Web dirigido por modelos permite un control gradual de la calidad de los artefactos intermedios facilitando la detección y corrección de errores en cada fase del proceso de manera sistemática y automatizable, contribuyendo al despliegue de aplicaciones Web de calidad.

Trabajos futuros incluyen la definición de modelos de calidad empíricamente contrastados que sirvan de base para la definición de modelos de medición aplicables a las distintas fases de un proceso de desarrollo Web. Otra línea de trabajo es la definición de un proceso de evaluación de calidad usando QVT-relations y su ejecución en el entorno de gestión de modelos MOMENT ([moment.dsic.upv.es](http://moment.dsic.upv.es)).

**Agradecimientos.** Este trabajo de investigación está financiado por el Ministerio de Educación y Ciencia, proyectos MEC-FEDER (TIN2004-03145), METASIGN (TIN2004-00779), DSDM (TIN2005-25866-E), CALIPSO (TIN2005-24055-E) y DESTINO (ref. TIN2004-03534). Esta investigación también forma parte del proyecto DADASMECA (GV05/220), financiado por la Comunidad Valenciana, y del proyecto DADS (PBC-05-012-2), financiado por Castilla-La Mancha.

## Referencias

- [1] Abrahão S., Insfran E. Early usability evaluation in Model-Driven Architecture Environments. In Proc. of the Sixth IEEE International Conference on Quality Software, IEEE Press, Wiley, Chichester, 2006.

- [2] Boehm B. W., Brown J. R., Kaspar H., Lipow M., McLeod G., and Merritt M. Characteristics of Software Quality, North Holland, 1978.
- [3] Cachero C., Poels G., Calero C. Towards a Quality-Aware Web Engineering Process. In Proceedings of the EMMSAD'07 Workshop, CAISE'07. June 2007. Norway. To appear.
- [4] Cachero C. Web Application Internal Navigability Measures. Available on-line at <http://www.dlsi.ua.es/~ccachero/pPublicaciones.htm>. Last updated November 2006.
- [5] Calero C., Ruiz J. and Piatinni M. A Web Metrics Survey Using WQM. In Proc. of the Fourth International Conference on Web Engineering, 147-160, Springer-Verlag, Heidelberg, LNCS 3140, 2004.
- [6] Comai S., Matera M. and Maurino A. A Model and an XSL Framework for Analyzing the Quality of WebML Conceptual Schemas. In Proc. of the Int. Workshop on Conceptual Modeling Quality, p 339-350. Springer-Verlag, Heidenberg, LNCS 2784, 2003.
- [7] Ferreira M., García F., Bertoa M., Calero C., Vallecillo A., Ruiz F., Piattini M. and Braga J. Software Measurement: Ontology and Metamodel. Tech. Report, University of Castilla-La Mancha, 2006.
- [8] Garcia F., Bertoa M., Calero C., Vallecillo A., Ruiz F., Piattini M., Genero M. Towards a consistent terminology for software measurement. Information and Software Technology 48(8), 631-644, 2005.
- [9] Genero, M. Olivas, J., Piattini M., Romero, F. Using metrics to predict OO information systems maintainability. CAISE 2001, Interlaken, Switzerland, 2001, 388-401.
- [10] Gomez J., Cachero C., Pastor O. Conceptual Modeling of Device-Independent Web Applications. IEEE MultiMedia 8(2), 26-39, 2001.
- [11] Insfran E., Pastor O. Wieringa R. Requirements Engineering-Based Conceptual Modelling. Journal of Requirements Engineering (RE), 7(2), 61-72, Springer-Verlag, 2002.
- [12] ISO/IEC 9126 Software engineering – Product quality – Part 1: Quality model. International Organization for Standardization, Geneva, 2001.
- [13] ISO/IEC 14598. Information technology – Software Product Evaluation. International Organization for Standardization. Geneva, 1999.
- [14] Kent S. Model-Driven Engineering. In Proc. of the Third International Conference on Integrated Formal Methods, p 286-298, Springer-Verlag, Heidelberg, LNCS 2335, 2002.
- [15] Lindland O.I., Sindre G. and Solvberg A. Understanding Quality in Conceptual Modeling. IEEE Software Vol 11(2) pp 42 – 49. IEEE Press. Alamitos, CA, USA, 1994.
- [16] McCall J.A., Richards P.K., Walters G.F., Factors in Software Quality, Nat'l Tech. Information Service, Vol 1, 2 and 3, 1977.
- [17] Meliá S and Gómez J. UPT: A Graphical Transformation Language based on a UML Profile. In Proc. of Second European Workshop on Milestones, Models and Mappings for Model-Driven Architecture, Springer-Verlag, 2006.
- [18] Moody D.L., Shanks G.G. Improving the quality of data models: empirical validation of a quality management framework. Information Systems 28: 619-650. Elsevier Science Ltd., 2003.
- [19] Moraga M., Calero C., Piattini M. Ontology Driven Definition of a Usability Model for Second Generation Portals. 1st Int. ICWE Workshop on Methods, Architectures & Technologies for e-Service Engineering (MATeS 2006), ACM Press. Vol. 155, 2006.
- [20] Olsina L., Rossi G. Measuring Web Application Quality with WebQEM, In IEEE Multimedia Magazine, Vol. 9, N° 4, pp. 20-29, 2002.
- [21] OMG-QVT. MOF QVT Final Adopted Specification. OMG doc. ptc/05-11-01, 2005.
- [22] QUINT2: The Extended ISO Model of Software Quality. Available on-line at: <http://www.serc.nl/quint-book/>
- [23] Seffah A., Donyaee M., Kline R.B. and Padda H.K. Usability measurement and metrics: a consolidated model. Software Quality Journal Vol 14. pp 159-178, 2006.
- [24] Software Quality Models and Philosophies. Chapter 1. [http://www.bth.se/tek/besq.nsf/\(WebFiles\)/CF1C3230DB425EDCC125706900317C44/\\$FILE/chapter\\_1.pdf](http://www.bth.se/tek/besq.nsf/(WebFiles)/CF1C3230DB425EDCC125706900317C44/$FILE/chapter_1.pdf)
- [25] Van Welie, M. van der Veer G. and Elins A. Breaking down Usability, In Proc. of Interact '99, Edinburgh, Scotland, 1999