

TRANSFORMACIÓN DE MODELOS NAVEGACIONALES DIRIGIDA POR CRITERIOS DE CALIDAD

C. Cachero¹, M. Genero², C. Calero² y S. Meliá¹

1: Grupo IWAD.

DLSI. Escuela Politécnica Superior. Universidad de Alicante
Ctra de San Vicente del Raspeig s.n. 03072 San Vicente del Raspeig. Alicante.
e-mail: {ccachero,smelia}@dlsi.ua.es

2: Grupo ALARCOS.

Escuela Superior de Informática de Ciudad Real. Universidad de Castilla-La Mancha
Paseo de la Universidad, 4. 13071 Ciudad Real
e-mail: {Marcela.Genero,Coral.Calero}@uclm.es

Palabras clave: Navegabilidad, Transformaciones QVT, Ingeniería Web

Resumen.

La Ingeniería Web ha reconocido desde sus inicios el papel preponderante que la navegabilidad juega en el diseño de aplicaciones Web, motivo por el cual se han propuesto distintas medidas para evaluarla. De entre ellas, las medidas definidas sobre modelos conceptuales son especialmente relevantes, ya que es bien sabido que la mejora de los artefactos producidos durante el proceso de desarrollo software tiene un gran impacto sobre la calidad y coste del producto final. Sin embargo, los modelos de medición en los que se engloban tales medidas, así como los métodos de medición asociados y las modificaciones de diseño que deben ser realizadas para mejorar sus valores, suelen ser totalmente dependientes de aproximaciones específicas de desarrollo hipermedial. En este artículo proponemos un modo de definir un modelo de medición para la evaluación de la navegabilidad de aplicaciones Web que es independiente de la metodología escogida. Para ello instanciamos un metamodelo que está basado en la Ontología de Medición de Software propuesta por García et al. Además presentamos el modo en que el modelo de medición puede integrarse con los modelos navegacionales de cualquier metodología concreta siguiendo una aproximación MDE. Esta integración se produce mediante la definición de un conjunto de reglas de transformación que, definidas en QVT, encapsulan el conocimiento específico de la metodología mediante expresiones que reflejan el método de medición y los criterios de decisión que se capturaron de manera abstracta en el modelo de medición. Además, las reglas de transformación también permiten automatizar las modificaciones que debe sufrir el modelo navegacional concreto si el resultado de la medición no satisface los criterios de decisión especificados.

1 Introducción

La creciente complejidad de las aplicaciones Web ha causado que el campo de la Ingeniería Web, definida como la aplicación sistemática, disciplinada y cuantificable de aproximaciones para el desarrollo y evolución eficiente de aplicaciones de alta calidad en la World Wide Web [16], se haya desarrollado de manera muy acelerada. Sin embargo, y a pesar de esta definición, la inclusión en las diferentes propuestas de Ingeniería Web de principios que contribuyan a garantizar que las aplicaciones resultantes cumplen con un conjunto de características de calidad deseables es aún un desafío para la disciplina. En efecto, hasta ahora no se ha demostrado empíricamente que exista una relación significativa entre la calidad de la aplicación Web y el hecho de que el diseñador haya seguido o no una metodología específica para su desarrollo.

Esta situación indudablemente contribuye al hecho de que, aún en nuestros días, el uso de aproximaciones "creativas" para el desarrollo de aplicaciones Web sea muy común. En este tipo de aproximaciones la calidad de la aplicación resultante depende de la pericia del diseñador, que suele centrarse en la mejora de la usabilidad de la interfaz mediante la aplicación de heurísticas que en su mayor parte (por encima del 60%) carecen de validación empírica [11]. Con este tipo de aproximaciones, la evaluación de la calidad debe hacerse sobre el producto final, a menudo con la ayuda de métodos de inspección específicos para la Web como el método de Evaluación de Usabilidad Sistemática (Systematic Usability Evaluation (SUE) [5]) o cualquiera de las múltiples herramientas de validación del contenido HTML de las páginas (véase e.g. [1, 19], por nombrar algunas). Aunque evaluar la calidad en las etapas finales del desarrollo puede resultar útil, es un hecho comúnmente aceptado que la detección temprana de problemas en los artefactos producidos en las fases iniciales del ciclo de vida puede ahorrar tiempo, reducir costes de producción, y lograr la calidad del producto deseada [10]. Consciente de tales beneficios, la Ingeniería Web ha comenzado recientemente a invertir esfuerzos en la inclusión de medidas que guíen la construcción de los diferentes modelos hipermediales que componen las distintas propuestas. En ellas, el modelo navegacional juega un papel preponderante, ya que es bien sabido que la navegabilidad de una aplicación web está en el núcleo de características de calidad como la usabilidad, la mantenibilidad, etc. [2, 21, 3, 4], y por lo tanto contribuye al éxito de implantación de la aplicación [3].

Las primeras propuestas en este nuevo campo han estado dirigidas hacia la promoción de un buen uso de los constructores de navegación conceptuales que provee cada aproximación. Por ejemplo, WebML define un Analizador de Calidad para WebML (WebML Quality Analyzer [12]) que permite analizar automáticamente la especificación XML de los esquemas conceptuales WebML, y verifica y mide algunos atributos internos, como la consistencia o compleción de los modelos. De manera similar, UWE [7] recomienda la integración de guías de diseño como parte del proceso de desarrollo de la aplicación, y su validación en el seno de las herramientas CAWE que soportan las distintas propuestas Web. Yendo un paso más allá, el trabajo de Abrahao *et al.* [2] propone la definición

formal de medidas que capturen algunas de las heurísticas existentes [17].

Sin embargo, aún queda un largo camino que recorrer. Según nuestro conocimiento, todas las medidas propuestas para modelos Web están fuertemente acopladas con respecto a las propuestas de Ingeniería Web para la que fueron definidas. Desafortunadamente, estas aproximaciones generalmente difieren no sólo en la notación sino también en la semántica asociada a los constructores, lo que dificulta la propagación de las medidas a diferentes entornos y/o metodologías. Además, una inmensa mayoría de las medidas se define de manera "ad-hoc", y el efecto de los valores de tales medidas sobre los modelos (es decir, las acciones que el diseñador debe realizar para mejorar el modelo) normalmente no se especifican.

Por lo tanto, en este artículo perseguimos tres objetivos. En primer lugar, pretendemos demostrar cómo la definición de metamodelos no sólo facilita el entendimiento de la semántica de las diferentes propuestas hipermediales sino que también sirve como la base sobre la cual se pueden definir y calcular medidas tempranas de una manera estándar. Para este propósito, en la Sección 2 presentamos una vista parcial del metamodelo navegacional de OO-H [15], y también cómo se expresan las medidas de navegabilidad sobre este metamodelo mediante expresiones OCL [25]. En segundo lugar, creemos que el campo de la Ingeniería Web puede beneficiarse de la definición de modelos de calidad Web débilmente acoplados, y que puedan además ser aplicados a una familia completa de aplicaciones Web (e.g. aplicaciones de comercio electrónico, aplicaciones de apoyo a la docencia, etc.), sin importar la metodología seleccionada. Afortunadamente contamos con una Ontología de la Medición (OMS) [14]. Esta ontología contribuye a la armonización de diferentes propuestas y estándares sobre la medición de software, proporcionando un conjunto coherente de conceptos comunes utilizados en este campo. Basándose en ella, se ha definido un metamodelo [13] que puede ser instanciado para definir de una manera estándar un modelo de medición Web genérico. En la Sección 3 ilustramos esta idea, y proporcionamos un ejemplo de instanciación del metamodelo. Por último, con el fin de automatizar la evaluación/evolución de los modelos navegacionales de acuerdo con el modelo de calidad Web seleccionado, en la Sección 4 proponemos el uso de reglas de transformación QVT [22] que, a partir del metamodelo navegacional y del metamodelo OMS, generan un nuevo modelo navegacional en el que se han comprobado los criterios de calidad expresados en la instanciación del metamodelo. Este modo de actuar es consistente con el paradigma *Model Driven Engineering* (MDE), que proporciona una serie de guías y principios para la automatización del proceso de producción del software. Para concluir, la Sección 5 resume las principales contribuciones de este artículo y algunas líneas de trabajo futuro.

2 El modelo navegacional de OO-H

Ya hemos comentado cómo el principal artefacto de la Ingeniería Web relacionado con la navegabilidad es el modelo navegacional. Este modelo recoge los caminos de navegación que el usuario puede seguir a través del espacio de información con el fin de conseguir sus

objetivos. El modelo navegacional se organiza normalmente en torno a subsistemas navegacionales, cada uno de los cuales encapsula tanto las vistas que el usuario necesita sobre los objetos de dominio como los modos de navegar de una vista a otra. Los nombres de los constructores, así como sus decoradores, normalmente difieren entre metodologías, reflejando de este modo los diferentes orígenes y objetivos para los que fueron desarrolladas. Debido a esta falta de acuerdo en cuanto a una ontología común que englobe conceptos conceptuales de navegación, la mayoría de las metodologías proporcionan un metamodelo que permite la definición sistemática y precisa de los conceptos involucrados y sus relaciones, y de este modo facilita la comunicación entre investigadores y/o profesionales.

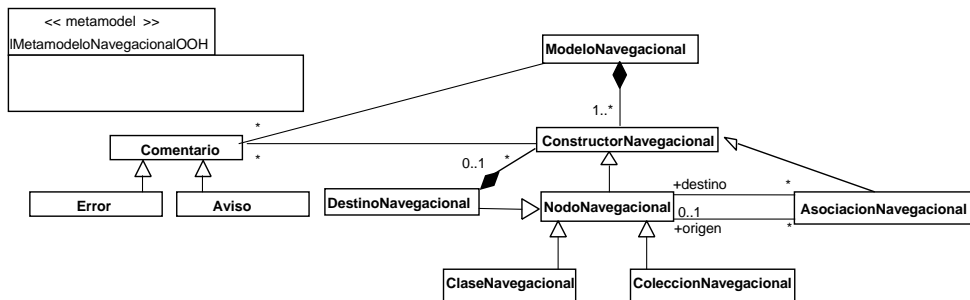


Figura 1: Vista parcial del paquete de navegación del metamodelo de OO-H

A modo de ejemplo, en la Fig. 1 podemos ver una vista simplificada del metamodelo de navegación de OO-H. En ella, podemos observar cómo, al igual que en otras aproximaciones hipermediales, los constructores principales son los *Destinos Navegacionales* (DN), las *Clases Navegacionales* (CN), las *Asociaciones Navegacionales* (AN) y las *Colecciones* (C). Brevemente, un DN es un mecanismo de agrupación que permite estructurar la navegación en subsistemas navegacionales. Las CN son vistas definidas sobre clases conceptuales, y reflejan la información que compone la vista de usuario, así como los servicios accesibles desde dicha vista. Las AN reflejan los caminos de navegación que un usuario puede recorrer a través del espacio de información. Por último, las C son mecanismos de acceso (menús) que permiten agrupar caminos de navegación. Con el fin de ilustrar su uso, imaginemos que deseamos modelar un sistema de venta de entradas por Internet. Un posible modelo navegacional de OO-H puede ser visto en la Fig. 2.

El modelo navegacional de primer nivel presentado en la Fig. 2 define una aplicación cuya navegación está organizada alrededor de distintos DN (e.g. *RegistroUsuario*, *Carrito-Compra*, *DetallesEspectáculo*, etc.), donde cada uno encapsula los caminos navegacionales disponibles para cubrir un subconjunto de requisitos de usuario. Los diferentes DN son accedidos mediante una Colección llamada *Entrada* que, dibujada como un triángulo invertido, representa el menú principal de la aplicación. Los DN y la C se conectan mediante AN que pueden tener uno o más *Filtros* asociados. Los filtros son expresiones OCL que permiten constreñir la navegación. En OO-H, existen filtros estructurales (Fe), que capturan la semántica de las relaciones de dominio y que pueden ser activados o no sobre cada

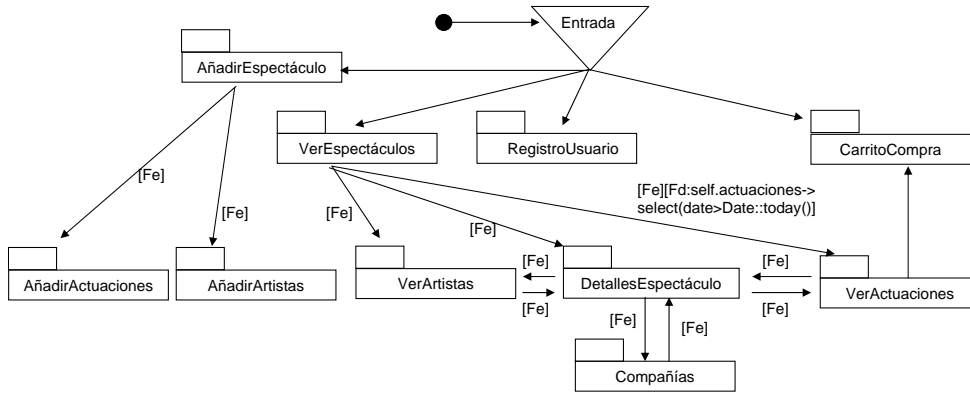


Figura 2: Modelo Navegacional OO-H correspondiente a un sistema de venta de entradas

AN cuando dicha asociación tiene el mismo origen y destino que la relación de dominio original. En nuestro ejemplo, la AN entre *VerEspectáculos* and *VerArtistas* es un ejemplo de asociación adornada con un Fe que se corresponde con la relación existente entre un objeto Espectáculo y sus Artistas correspondientes. Por otro lado, OO-H también permite definir filtros de usuario, que pueden restringir tanto la población origen del enlace como su población destino. Como ejemplo, el filtro *self.actuaciones->select(date>=Date::today())* que adorna la AN entre *VerEspectáculos* y *VerActuaciones* es un filtro de destino (Fd) que restringe el conjunto de objetos que son alcanzables al navegar por la asociación a aquellas actuaciones que todavía no han tenido lugar.

La Fig. 2 representa uno de los errores más comunes que los diseñadores web noveles realizan cuando comienzan a utilizar OO-H, y que es identificar nuevos DN para cada nuevo requisito de usuario, en lugar de realizar una tarea previa de agrupamiento. Esto causa un árbol navegacional pobremente estructurado, y aumenta la complejidad del modelo, tal y como se argumenta en [2]. Un modo de controlar esta proliferación de DN es mediante la definición de valores umbrales para el *Número de Destinos Navegacionales* (NDN). Esta medida ya fue definida y validada en [2] como *Número de Contextos Navegacionales*.

El uso de expresiones OCL nos ha permitido además integrar la definición de medidas en la herramienta CAWE *VisualWADE* [24] que da soporte a OO-H. De este modo se soporta la definición formal y no ambigua de medidas, al mismo tiempo que mejora su comprensibilidad [8, 9, 23]. Como ejemplo, la expresión que refleja la medida NDN sería la siguiente:

```
context ModeloNavegacional
def NDN:Integer=self.ConstructorNavegacional->select(oclIsTypeOf(DestinoNavegacional))->size()
```

De modo similar, en *VisualWADE* es posible definir en OCL valores umbrales para esa medida, i.e. el rango de valores deseable para ella. Con el fin de ilustrar este hecho, vamos a asumir que el número de DN de un modelo navegacional que evita que se comprometa la navegabilidad de una aplicación sigue la regla de Miller[20], que establece que el número de

conceptos (en nuestro caso DN) que una persona es capaz de recibir, procesar y recordar a la vez debe estar entre 5 y 9:

```
context NavigationalModel
inv noDemasiadosDN: NDN<=9
inv noDemasiadosPocosDN: NDN>=5
```

En la Fig. 2 podemos observar que la invariante noDemasiadosDN no se cumple, hecho que actualmente causa que se genere un aviso en nuestro entorno de desarrollo.

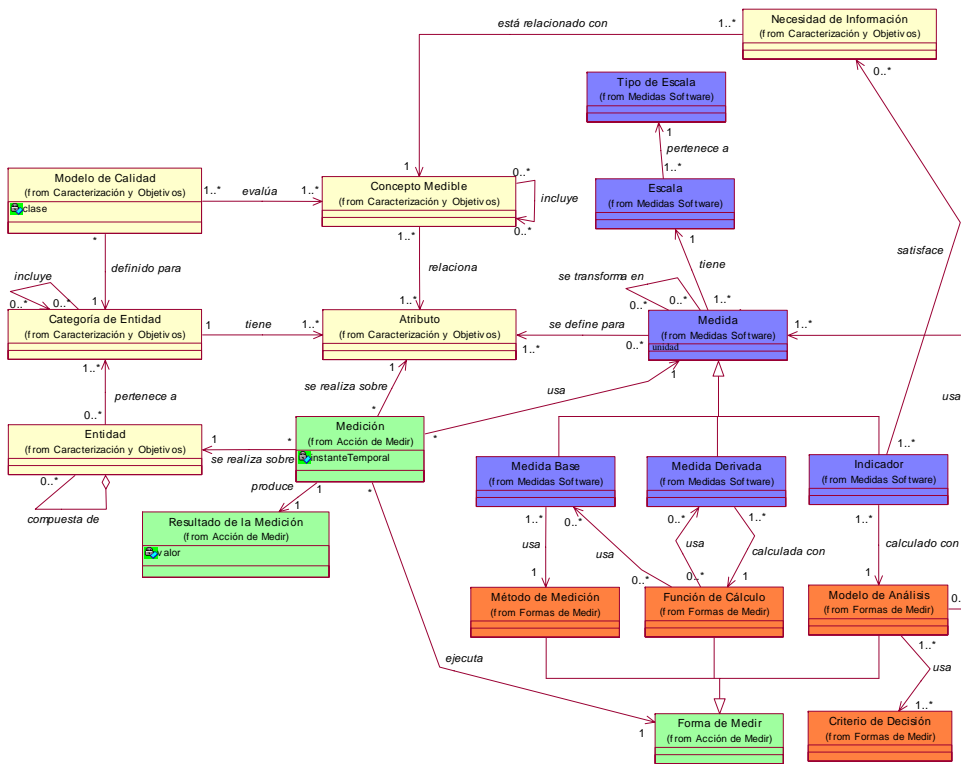


Figura 3: Diagrama UML del Metamodelo de la Medición OMS

Aunque esta forma de definir medidas de calidad funciona bien para aplicaciones sencillas, somos conscientes de que, a pesar de esto, presenta algunas desventajas. Por un lado, las reglas OCL asociadas con el metamodelo OO-H deben ser modificadas cada vez que cambian los criterios de decisión de las medidas. Esto suele ocurrir cada vez que cambia el tipo de aplicación que estamos modelando, ya que medidas y criterios que son relevantes para una cierta familia de aplicaciones puede no ser aplicable a otra. Más aún, esta forma de definir un modelo de calidad está fuertemente acoplado con los constructores de OO-H y su semántica, lo que hace que su adopción por parte de profesionales familiarizados con otras metodologías sea complicada, ya que se pueden encontrar con dificultades a la hora

de comprender y aplicar las medidas a sus propios modelos. Es por esto que a continuación proponemos un nuevo modo de definir medidas de navegabilidad que es independiente de la metodología elegida y de su metamodelo correspondiente, lo cual facilita la reusabilidad de la aproximación.

3 Modelo de medición para evaluar la navegabilidad de aplicaciones Web

Uno de los métodos más utilizados para homogeneizar y sistematizar conceptos que son relevantes para un determinado dominio es la definición de ontologías, es decir, especificaciones formales explícitas de una conceptualización compartida [14]. La definición de medidas utilizando una ontología estándar mejora la entendibilidad y comunicación de las medidas de software, ya que establece un vocabulario común para los diseñadores y desarrolladores, independientemente de la aproximación hipermedial elegida [6]. Es por ello que para nuestro trabajo hemos adoptado la OMS y su metamodelo asociado, presentados en [14, 13] (ver Fig. 3). La instanciación de este metamodelo nos permite definir un modelo de medición para evaluar la navegabilidad de cualquier modelo navegacional. Una característica importante de este modelo es que es reutilizable entre aplicaciones con requisitos de navegabilidad similares, que normalmente se agrupan en familias de aplicaciones.

El metamodelo de la medición de la Fig. 3 está dividido en cinco paquetes, un paquete básico y cuatro más que reflejan la división de la ontología en cuatro sub-ontologías. Estas sub-ontologías son: (1) la Caracterización y Objetivos de la Medición Software, (2) las Medidas Software, (3) las Formas de Medir, y (4) la Acción de Medir. El metamodelo puede ser fácilmente instanciado para definir el tipo de medidas en las que estemos interesados. A continuación mostramos cómo definir un modelo de medición concreto que incluye la medida NDN, que fue presentada en la sección 2.

3.1 Instanciación del metamodelo de la medición para la medida NDN

La definición de la medida NDN (ver Fig. 4) corresponde a la **Necesidad de Información** *conocer la navegabilidad* de un modelo navegacional de una aplicación Web. La **Categoría de Entidad** es un *Modelo Navegacional* genérico (podría ser el de cualquier metodología).

Con el objetivo de cubrir esta necesidad de información, nuestra medida es parte de un **Modelo de Calidad** (que hemos llamado Modelo de Calidad de la Navegación, *MCN*) cuyo objetivo por el momento se limita a evaluar el **Concepto Medible** *navegabilidad*. Este concepto medible se relaciona con un conjunto de **Atributos**, uno de los cuales es el atributo *complejidad navegacional*.

Para medir el atributo *complejidad navegacional* definimos la **Medida Base NDN**, cuya unidad de medición de esta medida es *numDN* (número de Destinos Navegacionales del diagrama), y que se captura mediante la propiedad **unidad** de la clase Medida. El valor de esta medida pertenece a la **Escala número natural**, englobada en la **Categoría**

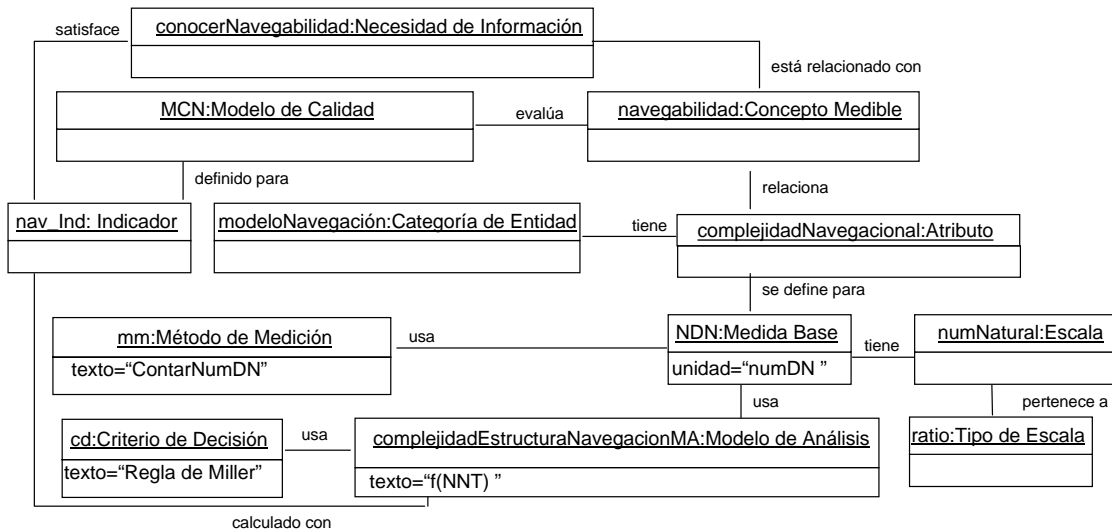


Figura 4: Instanciación parcial del metamodelo OMS con la medida NDN

de Escala *ratio*.

El **Método de Medición** de la medida NDN consiste en *contar el número de DN* que un modelo navegacional incluye en un mismo nivel. Esa medida base forma parte de un **Modelo de Análisis**, que usa un conjunto de **Criterios de Decisión** para evaluar la navegabilidad de la aplicación.

Volviendo a nuestro ejemplo, como sólo hemos definido una medida, el modelo de análisis simplemente explica cómo la complejidad navegacional de un modelo navegacional es función de la medida NDN, tal y como puede verse en la Fig. 4. Por su parte, el criterio de decisión que hemos asociado al modelo de análisis establece que un buen valor de la complejidad navegacional tiene que cumplir la regla de Miller, que como ya vimos implica que, para que el modelo navegacional no sea ni trivial ni demasiado complejo, el valor de NDN debe estar entre 5 y 9.

A continuación, una vez definida la medida NDN usando el metamodelo OMS, vamos a ilustrar cómo aplicarla de manera automática a un determinado modelo navegacional, como el que se presenta en la Fig. 2.

4 Evolución de modelos navegacionales basada en un modelo de calidad

Una de las ventajas principales del modelo de medición presentado en la Fig. 4 es su capacidad de reuso, ya que no asume ningún metamodelo navegacional concreto. Sin embargo, este conocimiento es necesario si pretendemos calcular el valor de la medida en un determinado modelo, ya que los nombres y semántica de los constructores navegacionales pueden diferir de una aproximación a otra. La conexión entre modelos particulares y el modelo de calidad genérico se puede establecer de manera sencilla si consideramos que un modelo navegacional en una aproximación concreta no es más que un subtipo de un

(hipotético) modelo navegacional genérico, es decir, un subtipo del concepto **Tipo de Entidad**. De manera similar, modelos navegacionales particulares (como el presentado en la Fig. 2) pueden ser vistos como instancia de la clase **Entidad**. Esta conexión abre el camino a la aplicación de una aproximación dirigida por modelos (MDE, Model Driven Engineering) [18] para la automatización del cálculo del concepto medible navegabilidad. Dicha automatización se produce mediante la definición de reglas de transformación que, definidas en QVT [22], permiten encapsular el conocimiento acerca de cómo un determinado valor se calcula en una aproximación de Ingeniería Web particular (en nuestro caso OO-H).

A partir del valor de las medidas que, a través de los atributos correspondientes, influyen en el concepto medible, es además posible decidir acerca de la conveniencia de realizar o no determinados cambios en el modelo navegacional sobre el que se ha calculado la medida, con el fin de acercar los valores de dicha medida a los valores considerados deseables para ese tipo de aplicación. De vuelta a nuestro ejemplo, en la Fig. 2 se puede observar a simple vista que el diagrama navegacional no cumple la regla de Miller. Sin embargo, ahora disponemos de un medio para automatizar la detección de este hecho, tal y como puede ser visto en la Fig. 5.

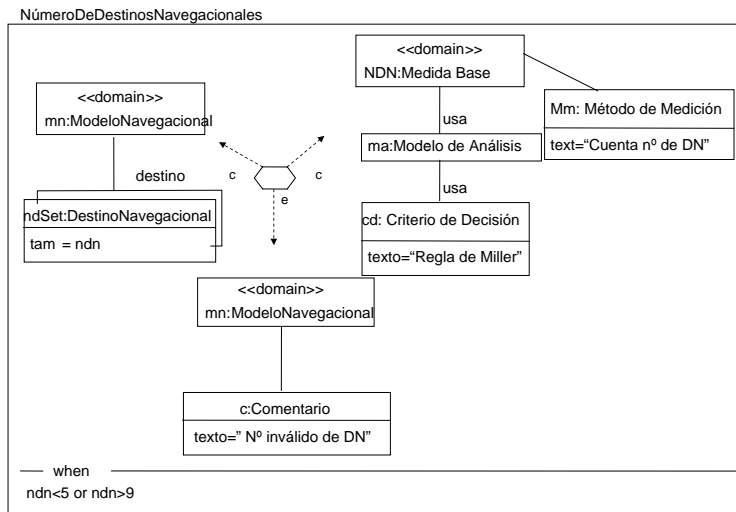


Figura 5: Regla de Transformación QVT que comprueba la violación de la regla de Miller sobre modelos navegacionales OO-H

En la Fig. 5 mostramos un ejemplo de regla de transformación que cuenta el número de DN de OO-H y es capaz de anotar el correspondiente modelo navegacional si el criterio de decisión asociado a la navegabilidad no se cumple.

Para ello se ha utilizado la notación gráfica de QVT para la relación *NúmeroDeDestinosNavegacionales*. Con esta notación es posible expresar cómo la transformación implica dos dominios de comprobación (*checkonly* (c)): el dominio *ModeloNavegacional* (raíz del metamodelo de

OO-H que presentamos en la Fig. 1) y el dominio *Medida Base* (definido en el contexto del metamodelo de medición OMS). A partir de estos dos dominios, la regla de transformación comprueba en primer lugar si existe una medida NDN en el modelo de calidad asociado a nuestra aplicación que presentamos en la Fig. 4 (esto es, si en el modelo de calidad existe un objeto de tipo Medida que se llame NDN, y si existe un objeto de tipo Criterio de Decisión cuyo texto sea 'Regla de Miller'). Si el resultado de la evaluación es verdadero, se procede a calcular el número de DN incluidos en el modelo navegacional considerado. En OO-H este valor se calcula contando simplemente el número de objetos de tipo 'Destino Navegacional' que existen en el modelo. La regla de transformación almacena este valor en la variable *ndn*.

La codificación de la la Regla de Miller en OO-H se establece en la cláusula *when* de la transformación. Esta cláusula indica, mediante una sintaxis compatible con OCL, que la regla de transformación debe ser activada sólo si se viola esta regla, es decir, sólo si el número de DN está fuera de los límites establecidos. En dicho caso, es necesario establecer la acción que deseamos que tenga lugar. Por simplicidad, en este ejemplo hemos decidido que la acción sea enriquecer el modelo navegacional de OO-H con un comentario, asociado al modelo completo, que avise al diseñador sobre la violación de la regla. No obstante es importante destacar que la acción podría ser mucho más compleja, e involucrar por ejemplo la reestructuración automática de los DN del modelo, dando lugar a un nuevo modelo navegacional. Sea cual sea la acción a desarrollar, ésta se debe especificar en un tercer dominio 'obligado' (*enforceable* (e)), que en este caso vuelve a ser el *ModeloNavegacional* de OO-H. La inclusión de nuevas medidas de navegabilidad implicaría la definición de nuevas reglas de transformación, al menos una para cada medida.

Por tanto los pasos de ejecución de cada regla de transformación serían:

- Comprobar en el modelo de medición si la medida para la que se ha definido la regla de transformación es relevante para el modelo navegacional de la aplicación actual
- Comprobar si se cumple la cláusula *when* que determina cuándo el modelo no se encuentra dentro del rango de valores considerado aceptable para la aplicación
- Si este es el caso, realizar las modificaciones necesarias sobre el modelo navegacional original para mejorar el valor de la medida

5 Conclusiones y trabajos futuros

Este artículo ha presentado un modo de definir modelos de medición para aplicaciones hipermediales que, basado en el metamodelo de medición OMS, permite integrarlos en cualquier aproximación Web. La definición de un modelo de medición a partir de un metamodelo permite además que dicho modelo participe en una aproximación MDE para la evaluación de la navegabilidad de modelos navegacionales. En dicha aproximación, el modelo de medición define los criterios de decisión que, en caso de ser necesario, son utilizados para la activación de una o más reglas de transformación. Estas reglas de

transformación actúan sobre el metamodelo específico de la aproximación que se trate, y permiten anotar y/o modificar los modelos navegacionales para convertirlos en modelos equivalentes respetuosos con los requisitos de navegabilidad impuestos por el modelo de calidad definido para esa familia de aplicaciones.

Así, las principales contribuciones de este trabajo pueden resumirse como sigue:

- Definición de un modelo de medición de la navegabilidad genérico, reutilizable entre metodologías dentro de una misma familia de aplicaciones.
- Automatización, siguiendo el paradigma MDE, del método de medición de las medidas expresadas en el modelo de medición.
- Automatización, siguiendo el paradigma MDE, de la anotación y/o transformación de modelos para facilitar su adecuación a los criterios de navegabilidad impuestos para la aplicación.
- Uso de ontologías y estándares (OMS, UML, OCL, QVT), que contribuyen a la mejora de la comprensión y reuso de nuestra propuesta.

En este momento se está trabajando en la definición de un repositorio de reglas de transformación para OO-H. Se prevé que las reglas contenidas en este repositorio no sólo anoten sino que también sean capaces de modificar la estructura del modelo navegacional de OO-H de una manera genérica. Para su elaboración, uno de los principales problemas que tendremos que abordar es la ausencia de trabajo empírico sobre valores umbrales para medidas sobre modelos navegacionales. Además, se está desarrollando una herramienta capaz de leer y aplicar reglas de transformación basadas en criterios de calidad para demostrar la viabilidad de nuestra propuesta.

6 Agradecimientos

Este trabajo ha sido parcialmente soportado por los proyectos METASIGN (TIN2004-00779), DSDM (TIN2005-25866-E) y TIC2003-07804-C05-03, del Ministerio de Ciencia y Tecnología, por el por el proyecto DADASMECA (GV05/220), de la Comunidad Valenciana, y por los proyectos DADS (PBC-05-012-2) y DIMENSIONS (PBC-05-012-1), financiados por el Ministerio de Educación de Castilla la Mancha.

REFERENCIAS

- [1] Web accessibility verifier. <http://aprompt.snow.utoronto.ca>.
- [2] S. Abrahao, N. Condory-Fernandez, L. Olsina, and O. Pastor. Defining and Validating Metrics for Navigation Models. En *Actas de METRICS'03*, páginas 200–210, 2003.
- [3] J. Almer. Designing for Web Site Usability. *IEEE Computer*, 35(7):102–103, 2002.

- [4] J. Almer. Web Site Usability, Design and Performance Metrics. *Information Systems Research*, 13(2):151–167, 06 2002.
- [5] A. De Angeli, M. Matera, M.F. Costabile, F. Garzotto, and P. Paolini. Validating the SUE inspection technique. En *Actas de Working Conference on Advanced Visual Interfaces*. ACM Press, 2000.
- [6] U. Assmann, S. Zschaler, and G. Wagner. *Ontologies En Software Engineering and Software Technology*, capítulo Ontologies, Meta-Models and the Model-Driven Paradigm. Springer, 2006 (pendiente de publicación).
- [7] R. Atterer, A. Schmidt, and H. Hussmann. Extending Web Engineering Models and Tools for Automatic Usability Validation. *Journal of Web Engineering*, 2005.
- [8] A. L. Baroni, S. Braz, and F. Brito e Abreu. Using OCL to Formalize Object-Oriented Design Metrics Definitions. En *Actas de 6th International ECOOP Workshop on Quantitative Approaches in Object-Oriented Software Engineering (QUAOOSE'02)*, 2002.
- [9] A. L. Baroni and F. Brito e Abreu. Formalizing Object-Oriented Design Metrics upon the UML Meta-Model. En *Brazilian Symposium on Software Engineering*, 2002.
- [10] L. Briand, S. Morasca, and V. R. Basili. Defining and Validating Measures for Object-Based High-Level Design. *IEEE Transactions on Software Engineering*, 25(5):722–743, 10 1999.
- [11] C. Calero, J. Ruiz, and M. Piattini. A Web Metrics Survey Using WQM. En *Actas de 4th International Conference on Web Engineering (ICWE'04)*. Springer, 2004.
- [12] S. Comai, M. Matera, and A. Maurino. A Model and an XSL Framework for Analyzing the Quality of WebML Conceptual Schemas. En *Actas de ER'02 International Workshop on Conceptual Modeling Quality*, páginas 339 – 350. Springer, 10 2002.
- [13] M. Ferreira, F. García, M. Bertoa, C. Calero, A. Vallecillo, F. Ruiz, M. Piattini, and J.L. Braga. Medición del software: Ontología y metamodelo (pendiente de publicación). Reporte Técnico, Universidad de Castilla-La Mancha, 2006.
- [14] F. García, M.F. Bertoa, C. Calero, A. Vallecillo, F. Ruiz, M. Piattini, and M. Genero. Towards a consistent terminology for software measurement. *Information and Software Technology*, páginas 1–14, 07 2005.
- [15] J. Gómez, C. Cachero, and O. Pastor. Conceptual Modelling of Device-Independent Web Applications. *IEEE Multimedia Special Issue on Web Engineering*, 8(2):20–32, 04 2001.
- [16] L. Heuser. The real world or Web Engineering? En *Actas de 4th International Conference on Web Engineering*, volume 3140, páginas 1–5. Springer, 06 2004.
- [17] M. Ivory. *Automated Web Site Evaluation*. Kluwer Academic Publishers, 2004.

- [18] S. Kent. The Expressive Power of UML-based Engineering. En *Actas de 3rd International Conference on Integrated Formal Methods*, volume 2335, page 286. Springer, 06 2002.
- [19] Knowledge-based web automatic reconfigurable evaluation with guidelines optimization. <http://www.isys.ucl.ac.be/bchi/research/Kwaresmi.htm>.
- [20] G. Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. En *The Psychological Review*, volume 63, páginas 81–97, 1956.
- [21] J. Nielsen. *Designing Web Usability*. New Riders, 2000.
- [22] MOF Query/Views/Transformations Final Adopted Specification. Documento OMG. [ptc/05-11-01. www.omg.org/docs/ptc/05-11-01.pdf](http://www.omg.org/docs/ptc/05-11-01.pdf).
- [23] L. Reynoso, M. Genero, and M. Piattini. OCL2: Using OCL in the Formal Definition of OCL Expression Measures. En *Actas de 1st International Workshop on Algebraic Foundations for OCL and Applications (WAFOCA'06)*, 2006.
- [24] VisualWADE: Web Application Development Environment. <http://www.visualwade.com/>.
- [25] J. Warmer and A. Kleppe. *The Object Constraint Language. 2nd Edition. Getting your models ready for MDA*. Addison Wesley, 2003.