Visualización de Propiedades del Historial de los Artículos de un Manejador de Contenidos Basado en Wiki Aplicando Ingeniería Dirigida por Modelos

Eugenio Scalise[1], Nancy Zambrano[1], Jean-Marie Favre[2]

1 Universidad Central de Venezuela, Facultad de Ciencias,
Escuela de Computación, Centro ISYS, Caracas, Venezuela

2 Universidad Joseph Fourier (Grenoble 1), Equipo Adele,
Laboratorio de Informática de Grenoble (LIG), Grenoble, Francia

eugenio.scalise@ciens.ucv.ve, nancy.zambrano@ciens.ucv.ve, Jean-Marie.Favre@imag.fr

Abstract. Este artículo describe un enfoque dirigido por modelos para el procesamiento del historial de los artículos que conforman un sistema para el manejo de contenido colaborativo basado en wiki. El historial constituye la secuencia de versiones que posee cada artículo y además de la información del contenido del artículo, contiene datos sobre la fecha y hora de modificación, el usuario que realizó el cambio, el tipo de cambio, entre otros. En este trabajo se describe cómo se pueden calcular propiedades alternas a las disponibles en este tipo de manejador de contenido y generar tanto reportes como visualizaciones gráficas de la información, todo mediante un enfoque dirigido por modelos, donde la composición de transformaciones es utilizada para generar escenarios de reportes y/o visualización. Los reportes y visualizaciones propuestos en este trabajo permiten tener una visión de la comunidad de usuarios que participan en un wiki, así como también la evolución de los artículos hospedados en el mismo.

Keywords: ingeniería dirigida por modelos, wiki, historial, metamodelos, transformaciones, espacios tecnológicos, visualización de información

1 Introducción

Un *wiki* es un manejador de contenidos cuyo principal propósito es el de permitir que los usuarios puedan crear y modificar el contenido publicado de una manera simple. En los últimos años se ha incrementado el uso de los wikis. El primer sitio con las características de un wiki fue introducido por Mard Cunningham en 1995 en el WikiWikiWeb (http://c2.com/cgi/wiki?WikiWikiWeb).

MediaWiki (http://mediawiki.org) es una implementación del concepto de wiki, de hecho, es el software utilizado por Wikipedia (http://wikipedia.org), el proyecto de enciclopedia abierta y multilenguaje soportado por la Wikipedia Foundation. Para abril de 2008 Wikipedia hospedaba aproximadamente 10 millones de artículos en 253

idiomas [9], siendo estos artículos escritos de manera colaborativa por voluntarios distribuidos en todo el planeta y cuyo único requisito es el acceso a internet.

La mayoría de las implementaciones de wikis mantienen un registro de los cambios realizados y guardan las distintas versiones de un artículo. Este historial es muy útil para conocer el progreso de cada artículo y para revertir cambios en caso de errores o vandalismo. Aunque algunos wikis soportan modificaciones por usuarios anónimos, cada versión de un artículo posee la información asociada al usuario que la realizó, el tipo de cambio, la fecha y hora, la sección modificada, entre otras.

En este artículo se presenta la aplicación de un enfoque dirigido por modelos para recuperar, procesar y visualizar información relativa al historial de los artículos que conforman un wiki basado en MediaWiki.

En la Sección 2 se plantea la problemática identificada y en la Sección 3 se detalla la propuesta de solución a esta problemática, esta solución es desglozada en varias partes: propuesta del metamodelo del dominio, extractor de los datos del wiki, cálculo de las propiedades del historial y visualización de estas propiedades mediante reportes o gráficos. En la Sección 4 se describen las conclusiones y aportes del trabajo, además de citar algunos trabajos relacionados.

2 Planteamiento del Problema

Los sistemas de wiki actuales permiten consultar el historial de cada página del wiki mediante una interfaz textual en la cual se observa la lista de cambios realizados sobre cada página en orden cronológico decreciente. Cada entrada de la lista representa una versión con toda la información básica: hora, fecha, usuario (incluyendo un enlace a sus contribuciones en el wiki y su página de discusión -*talk page*-), una indicación para cambios menores, el tamaño de la versión (en bytes) y la descripción del cambio. Adicionalmente se pueden realizar comparaciones entre versiones y si es necesario, revertir los cambios realizados en cualquier modificación.

A pesar que estas funcionalidades son muy útiles, los sistemas como MediaWiki no explotan propiedades presentes en la información gestionada, en particular con la evolución de cada artículo del wiki, comunidad de usuarios participantes, frecuencia de los cambios, entre otros. Además, no hay visualizaciones gráficas que pueden resultar de gran utilidad.

En este artículo se describe cómo el uso de técnicas de la ingeniería dirigida por modelos [3] pueden ser utilizadas para incorporar reportes alternos y visualizaciones sobre el historial de un wiki.

3 Solución del Problema

Como se indicó anteriormente, el objetivo principal de este artículo consiste en utilizar la información el historial de cambios (*history*) de las páginas de un wiki para visualizar propiedades de los datos. Esto se realiza mediante un enfoque dirigido por modelos utilizando un wiki basado en MediaWiki.

Para llevar a cabo este objetivo se requiere:

- Definir un metamodelo del dominio de interés, es decir un metamodelo que defina las entidades que conforman la información gestionada por un wiki basado en MediaWiki y las relaciones entre dichas entidades.
- Simplificar el metamodelo del dominio para considerar únicamente los aspectos relativos al problema (el historial y la información directamente relacionada), de manera que se pueda obtener un metamodelo a utilizar de manera activa en la implementación.
- Desarrollar un extractor que recupere la información de los artículos de un wiki y la represente mediante un modelo conforme con el metamodelo de implementación del historial de MediaWiki.
- Definir los metamodelos intermedios requeridos para la solución del problema utilizando un enfoque dirigido por modelos. En particular, serán necesarios metamodelos para las propiedades a calcular sobre el wiki y metamodelos para las visualizaciones a realizar.
- Definir transformaciones a nivel de los metamodelos para poder obtener modelos conformes a los metamodelos definidos, utilizando como entrada los datos de una instalación particular de un wiki.
- Generar los reportes y visualizaciones componiendo las transformaciones desarrolladas y utilizando modelos de prueba con datos reales.

En las próximas secciones se describe cada una de las actividades mencionadas arriba, que en conjunto representan una solución dirigida por los modelos para el problema planteado.

3.1 Definición del metamodelo del dominio

Para realizar el procesamiento de la información gestionada por un wiki es necesario definir el metamodelo asociado a este tipo de manejador de contenido.

Es importante aclarar que todos los metamodelos son presentados mediante diagramas de clase UML con el objetivo de facilitar su comprensión; sin embargo, una implementación mediante un enfoque dirigido por modelos requiere una representación de los metamodelos tal que puedan ser utilizados mediante herramientas de software. Todos los metamodelos utilizados en este caso de estudio tienen asociados su representación en KM3, que es la notación textual para definir metamodelos disponible en las herramientas de la plataforma AMMA (ATLAS Model Management Architecture) [1].

En el metamodelo conceptual de la Figura 1, se muestra que un wiki (clase `Wiki`) está conformado por varias páginas (`Page`). Hay varios tipos de páginas en un wiki, entre ellas destacan: páginas especiales con funcionalidades del wiki (`SpecialPage`), artículos del wiki (`Article`), categorías (`Category`), páginas para cada usuario del wiki (`UserPage`), páginas para dejar mensajes y establecer discusiones con usuarios del wiki (`UserTalkPage`).

Existe un tipo particular de páginas que pueden ser editadas por cualquier usuario (`EditablePage`); estas páginas son de especial interés para esta investigación debido a que ellas tienen asociado un historial de cambios (`History`), el cual es un conjunto ordenado de versiones. Una versión (`Version`) de una página editable contiene toda la información necesaria para saber quién realizó el cambio (atributo `modifiedBy`),

cuándo lo realizó (atributo `date`) y cuál es la versión resultante del cambio (atributo `text`). Esto permite restablecer cambios en caso de errores o vandalismo, e inclusive permite comparar versiones. Cuando un usuario modifica una página del wiki puede indicar si el cambio es menor (atributo `isMinorEdit`) y adicionalmente puede escribir una breve descripción sobre el cambio realizado (atributo `editSummary`).

Un wiki gestionado por MediaWiki puede ser configurado para ser modificado tanto por usuarios anónimos (representados por su dirección IP, clase `IPAddress` en el metamodelo) o por usuarios registrados (`RegisteredUser`).

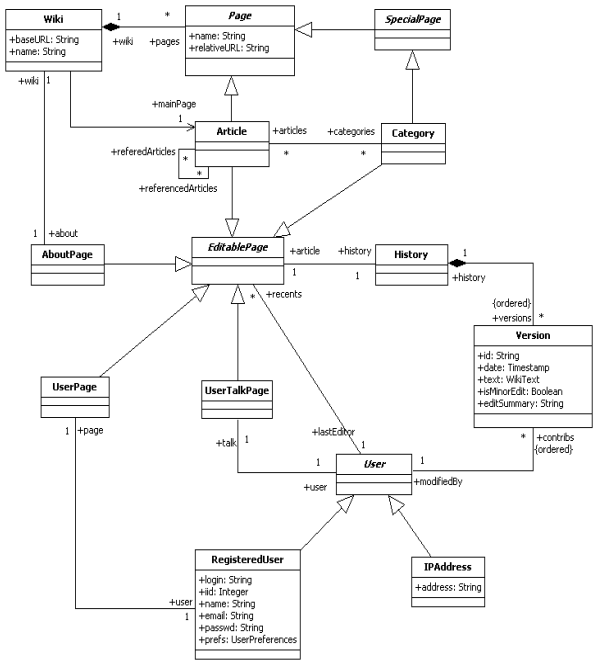

Fig. 1. Metamodelo de un wiki basado en MediaWiki

Aunque este *metamodelo conceptual* constituye una versión reducida del dominio de wikis basados en MediaWiki, éste debe ser simplificado de manera que se pueda utilizar en la implementación del caso de estudio descrito en este artículo, tal como se presenta en la Figura 2. Esta versión es un *metamodelo de implementación* que sólo considera las páginas editables (denomidadas por simplicidad *artículos*, clase `Article`), su historial y los usuarios. El metamodelo simplificado se puede obtener de manera heurística o aplicando primitivas para la refactorización orientada a objetos similares a las descritas en [4] y en [5].

Los criterios utilizados para la simplificación del metamodelo están relacionados con el nivel de detalle de los datos disponibles y con los datos que sean relevantes para la implementación a realizar. Los datos a utilizar en este caso de estudio se obtienen mediante un proceso de recuperación de información directamente del Web; es decir, directamente de las páginas del wiki. Esto generalmente se realiza mediante un proceso de recuperación o adquisición denominado *Web scraping*.

Esta actividad es realizada por un módulo extractor descrito en la Sección 3.2.

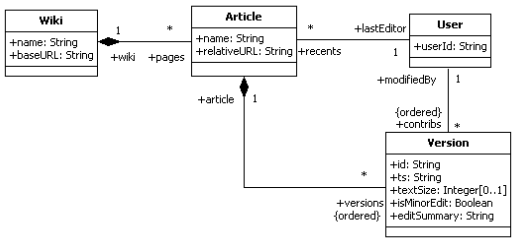

Fig. 2. Metamodelo de implementación de los artículos de un wiki y su historial

3.2 Desarollo del Extractor

Teniendo ya el metamodelo de la información de las páginas del wiki y su respectivo historial, se requiere un mecanismo para que los datos de un wiki en producción, generalmente almacenados en una base de datos y disponible para el usuario mediante páginas HTML, sean convertidos en un modelo conforme con el metamodelo de MediaWiki de la Figura 2.

Para esto se desarrolló un módulo denominado `MediaWikiHistoryExtractor` que recibe como entrada una especificación con los datos y artículos del wiki a extraer y produce como salida un modelo conforme al metamodelo de MediaWiki.

La arquitectura de los componentes que conforman el extractor es presentada en la Figura 3 y está conformada por:

- Un programa en Java (`MediaWikiHistoryScraper`) que recupera la información del Web y produce una representación XML con los datos del wiki (`MediaWikiScrap.xml`), los artículos y el historial de cada artículo. La especificación con la información de entrada (`MediaWikiScrapSpec.json`) se realizó mediante un DSL basado en JSON (JavaScript Object Notation) [6] en el cual se especifican los parámetros del escenario de recuperación (URL del wiki y conjunto de páginas a extraer).
- Un motor XSLT para ejecutar una transformación XSL que convierte los datos extraidos en un formato compatible con un DSL de MediaWiki (`ScrapInfo.txt`) que almacena la información del wiki y su historial en texto plano. Este formato intermedio es de especial importancia para automatizar el proceso de generación del modelo conforme al metamodelo de MediaWiki.
- El conjunto de herramientas que conforman TCS (Textual Concrete Syntax) [1], utilizado para implementar una transformación del tipo *text-to-model* (`MediaWikiHistory.tcs`) de los datos en el DSL de MediaWiki a un modelo conforme con el metamodelo de MediaWiki (`MWH_Model`).

El uso de un paso intermedio mediante una transformación XSL fue clave en las decisiones de diseño, debido a que es mucho más simple y extensible que la opción de incluir la transformación directamente en el extractor desarrollado en Java.

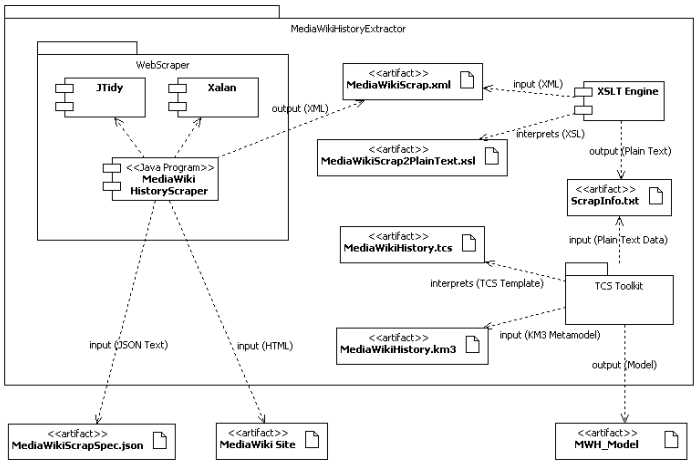

Fig. 3. Arquitectura del extractor de MediaWiki

Otro punto clave fue la inclusión de un DSL para la representación en texto plano del metamodelo de MediaWiki. La herramienta TCS permite establecer una correspondencia entre la sintaxis concreta (representación textual) de un metamodelo y la sintaxis abstracta (clases, atributos y relaciones). Mediante esta correspondencia, TCS es capaz de generar automáticamente artefactos de software que realicen las transformaciones del tipo *text-to-model* y *model-to-text*. La transformación utilizada en el extractor es del tipo *text-to-model* (DSL de MediaWiki a modelo conforme con el metamodelo de MediaWiki) y el modelo obtenido está expresado en el estándar de la Fundación Eclipse (Ecore) [2].

3.3 Cálculo de Propiedades del Historial

El modelo conforme con el metamodelo de MediaWiki, obtenido mediante el extractor descrito en la Sección 3.2, es el punto de partida para el desarrollo de transformaciones para el cálculo de propiedades sobre los datos del wiki, que luego pueden ser visualizadas de manera gráfica o como parte de reportes, no disponibles en los wikis basados en MediaWiki.

Por simplicidad a las propiedades se les denominó *métricas*, definiendo tres tipos de ellas: generales, con clasificador y específicas.

Las métricas generales constituyen propiedades numéricas (enteras o reales) calculadas para un artículo del wiki a partir de su información histórica. Como ejemplo se tienen: el número de versiones en el historial, el número de versiones marcadas como *menores*, la proporción de ediciones menores, la cantidad de usuarios que han participado en la edición del artículo, la proporción entre el número de editores con respecto al número de versiones, entre otras.

Las métricas con clasificador son similares a las métricas generales pero discriminan o agrupan las propiedades por una condición (denominada *clasificador*). Considerando como clasificador el usuario que realiza la versión, se tienen los

siguientes ejemplos para este tipo de métrica: el número de modificaciones realizadas sobre un artículo, el radio o proporción de modificaciones, el número de modificaciones menores, entre otras.

Las métricas específicas están orientadas a visualizaciones particulares o reportes de los datos gestionados por el wiki. En el marco de este trabajo se propuso una métrica específica, basada en la información utilizada por la visualización *history flow* propuesta en [7]. Para realizar dicha visualización se requiere calcular a partir del conjunto de versiones de cada artículo: el tiempo transcurrido entre dos versiones, el tamaño de la versión (en *bytes*), el usuario que realiza el cambio y la fecha del cambio. Estos datos son visualizados mediante una especialización de un diagrama de barras y puede ser muy útil para detectar patrones en los datos que permiten detectar conflictos, candidatos a vandalismo y en general, observar de manera gráfica la evolución de las páginas del wiki.

En general, para cada tipo de métrica se tiene su metamodelo y transformaciones que que expresen cómo se calculan los atributos de una instancia de métrica a partir de la información del historial de cada artículo del wiki, según la métrica a calcular.

En la Figura 4 se muestra el metamodelo asociado a las métricas generales, considerando sólo métricas numéricas. El metamodelo es suficientemente extensible para agregar tantas métricas concretas como se requiera. Una instancia de este metamodelo contiene un conjunto de métricas generales -pares (nombre,valor)- asociadas a cada item de estudio. En este contexto, cada item representa un artículo del wiki en estudio.

Para generar las instancias de métricas conformes al metamodelo de la Figura 4 se requiere una transformación que tenga como entrada una instancia del metamodelo de MediaWiki (Figura 2) y como salida una instancia del metamodelo de la Figura 4.

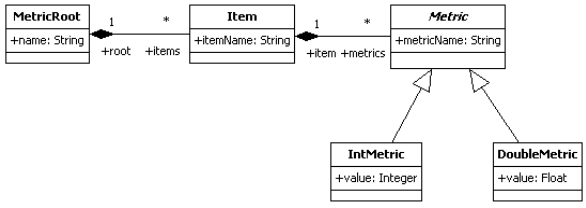

Fig. 4. Metamodelo de las métricas generales asociadas a un item (artículo de wiki)

Las transformaciones desarrolladas fueron escritas en el lenguaje ATL [1], que forma parte del ambiente AMMA. Por razones de espacio, las transformaciones no son mostradas, sin embargo la Figura 5 muestra el megamodelo (modelo donde cada elemento es a su vez un modelo) con los artefactos utilizados en esta solución dirigida por modelos.

La estructura de un escenario para las métricas con clasificador y específicas es similar al descrito para las métricas generales y por razones de espacio no es presentado.

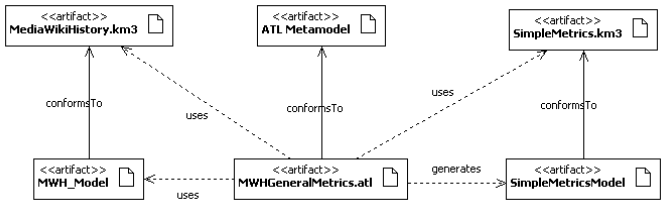

Fig. 5. Megamodelo correspondiente al escenario de generación de las métricas generales

3.4 Visualización de Propiedades del Historial

Una vez calculadas las propiedades a estudiar del historial de un wiki, sólo resta mostrar un reporte o representación gráfica con los datos calculados. Dependiendo del caso, este paso puede ser realizado mediante una transformación entre modelos (de una métrica a un gráfico/reporte) o mediante una transformación que genere directamente el gráfico/reporte en un formato apropiado (texto, html, csv u otro).

Para las métricas generales resultan apropiados los reportes mediante tablas o diagramas de barra, si se desea comparar los valores entre los distintos artículos del wiki.

Las métricas con clasificador pueden ser visualizadas mediante reportes textuales en forma de tablas. Si los valores de estas métricas son numéricos, pueden ser interesantes los diagramas de torta (*pie charts*) donde se puedan observar visualmente porcentajes, por ejemplo el porcentaje de contribución de cada usuario en la edición del artículo del wiki, entre otras.

El enfoque seguido para las métricas generales y con clasificador fue el de escoger el metamodelo de visualización apropiado y luego escribir la transformación para generar la instancia con los datos del historial del wiki. Estos resultados luego son importados hacia cualquier herramienta de visualización utilizando una transformación que genere el formato apropiado; a este tipo de transformación en el contexto de ATL se le denomina *query*.

Las métricas específicas no siguen un patrón general, cada una es particular. En la adaptación de la visualización *history flow*, se requiere un tipo particular de diagrama de barra donde: cada artículo del wiki tiene un diagrama asociado (su gráfico del historial), cada barra del diagrama corresponde a una versión, el ancho de cada barra corresponde a la distancia con respecto al cambio anterior, el alto de la barra representa el tamaño en bytes de la versión y el color representa el usuario que hizo el cambio. Este diagrama es muy útil para visualizar la evolución en el tiempo de un artículo y observar la frecuencia de cambios, los usuarios que participan y el crecimiento/decrecimiento del texto editado.

La estrategia seguida en este caso fue la de definir el metamodelo de nuestra adaptación del *history flow*, denominada *history graph* (ver metamodelo en la Figura 6) y escribir una transformación a partir de los datos del historial extraido de MediaWiki. Luego, se utilizó un *query* para generar una visualización del gráfico en HTML (ver ejemplo mostrado también el la Figura 6). El diagrama resultante utiliza

recursos interactivos de HTML para visualizar el autor de cada versión posicionando el ratón sobre la barra y cambiar la escala de visualización en el caso que se requiera mayor o menor detalle.

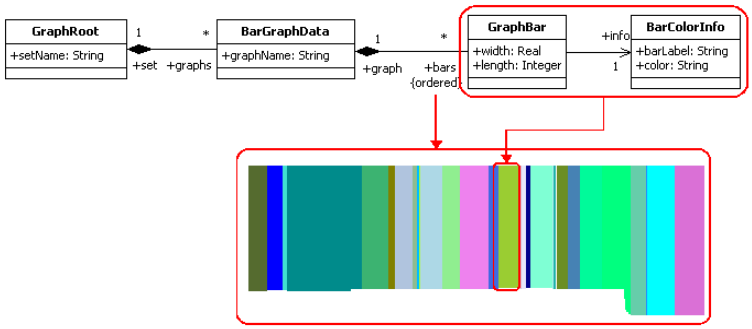

Fig. 6. Metamodelo de *History Graph* y relación con una visualización

Se desarrollaron dos versiones de transformaciones para generar *history graph*, una que considera todas las versiones del historial y otra que ignora las versiones marcadas como menores, lo que resulta de especial utilidad para historiales muy grandes de páginas que son modificadas con mucha frencuencia, pero con cambios poco significativos.

5 Conclusiones

Como se pudo observar a lo largo del artículo, el enfoque dirigido por modelos utilizado como estrategia de solución permitió dividir el proceso en tres actividades: extracción, cálculo de propiedades y visualización. En la extracción se utilizan varios espacios técnicos[1] (XML, grammarware y modelware) además de los lenguajes de transformación apropiados (XSL, TCS). El cálculo de propiedades se realiza totalmente con técnicas y métodos del espacio técnico modelware (modelos expresados en KM3/Ecore y transformaciones en ATL) y finalmente, la visualización mezcla técnicas del espacio técnico modelware y transformaciones específicas según el tipo de formato de salida. En resumen, el proceso completo consiste en componer transformaciones apropiadas, según el tipo de visualización que se quiere realizar, todo esto con una sola ejecución del proceso de extracción de los datos.

La solución propuesta para el desarrollo del extractor permite recuperar información del web siguiendo un enfoque flexible, basado en transformaciones simples que combinadas entre sí, producen como salida un modelo conforme a un metamodelo del dominio de los datos extraidos. Tradicionalmente el problema de recuperación de información es abordado mediante técnicas de procesamiento de texto, definición de robots web (también llamados *spiders* o *scrapers*), entre otras.

1 Un espacio técnico puede ser visto como un dominio tecnológico particular, donde se pueden caracterizar sus artefactos mediante relaciones similirares a las existentes entre los modelos y sus metamodelos de referencia, además de poseer tecnologías para la definición de transformaciones.

Estas alternativas tradicionales suelen ser medianamente complejas y sobre todo, muy dependientes del formato de los datos a recuperar y muy orientadas a programación en lenguajes de propósito general (Java, C/C++, Ruby, etc) o lenguajes especializados como Perl. El enfoque propuesto en esta investigación disminuye las dependencias de los programas con respecto a los aspectos de formato, utilizando plantillas XSL para extraer los datos relevantes mezclados en las etiquetas HTML. En caso de cambios de formato, sólo se requiere actualizar las plantillas XSL, permaneciendo sin cambios el resto de los componentes. Adicionalmente, el uso de TCS incorpora en el proceso de extracción los elementos propios del dominio de wiki.

El separar el cálculo de propiedades de la visualización o generación de reportes es de especial importancia para poder visualizar la misma información de distintas formas, e inclusive, para seleccionar la visualización más apropiada según los datos. Adicionalmente, el uso de un enfoque dirigido por los modelos permite que se puedan agregar nuevas métricas con sólo definir su metamodelo y la transformación para el cálculo de la misma, una situación similar ocurre con las visualizaciones.

El enfoque para utilizar los componentes desarrollados consiste en ensamblar un escenario, seleccionando las métricas y visualizaciones más apropiadas según los parámetros que se quiera estudiar del historial del wiki. Mediante las métricas alternas que se pueden calcular mediante este enfoque se ofrece una visión más organizada de la comunidad de usuarios que participan en el wiki, pudiendo observar cómo participan y en qué tópicos se especializan. Esta información está representada en el wiki, pero MediaWiki no las aprovecha.

Algunos grupos de investigación han realizado distintos tipos de visualizaciones de la información de Wikipedia, sin embargo, el enfoque seguido no es orientado por modelos como el que aquí se propone. Entre los trabajos disponibles destacan los desarrollados por el Visual Communication Lab de IBM Research que incluyen dos proyectos de visualización de datos de Wikipedia: History Flow (flujo de historial) [7] y Chromograms (cromogramas) [8]. Existen también algunas aplicaciones en línea que permiten visualizar y navegar los artículos de Wikipedia de manera alternativa, como por ejemplo WikiMindMap (http://wikimindmap.org).

Referencias

1. J. Bézivin, F. Jouault, I. Kurtev, P. Valduriez. Model-Based DSL Frameworks. 21st Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications, OOPSLA 2006, October 22-26, 2006, Portland, OR, USA. ACM, pages 602-616.
2. F. Budinsky, D. Steinberg, E. Merks, R. Ellersick, T. J. Grose. Eclipse Modeling Framework. ISBN 0131425420. Addison-Wesley, 2003.
3. J-M. Favre, J. Estublier, M. Blay-Fonmarino: L'ingénierie dirigée par les modèles: au-delá du MDA. Hermes Sciences. Février 2006.
4. M. Fowler. Refactoring: Improving the Design of Existing Code. Addison-Wesley, 1999.
5. K. Lano. Advanced Sytems Design with Java, UML and MDA. Elsevier Science & Technology Books, ISBN: 9780750664967, 2005.
6. D. Rubio. An Introduction to JSON. Dev2Dev, Bea Systems Inc, February 2007.
7. F. Viegas, M. Wattenberg, K. Dave. Studying Cooperation and Conflict between Authors with history flow Visualizations. Proc. of the SIGCHI.Vienna, Austria; pp 575 – 582; 2004.
8. M. Wattenberg, F. Viegas, K. Hollenbach. Visualizing Activity on Wikipedia with Chromograms. Proc. of Interact 2007. September 10-14, 2007. Rio de Janeiro, Brasil.
9. Wikipedia @ Wikipedia, the free encyclopedia, http://en.wikipedia.org/wiki/Wikipedia