**Capitulo 4 - Transiformación de modelo a texto (M2T) con Acceleo**

**4.1 Introducción**

En capitulo anterior se ha visto el metamodelo de contenido (*Content*) para la representación de la sintaxis abstracta de MoWebA con la nueva clasificación de elementos de interfaz, en la cual aparecen a diferencia de MoWebA tradicional que contempla elementos para la web 1.0, elementos enriquecidos que son comunes en las aplicaciones RIA. También junto al metamodelo de contenido, se presentó el metamodelo de posicionamiento (*Layout*), que permite definir la posición de los diferentes elementos del metamodelo de contenido, dentro de una página en particular. Al finalizar la presentación de los metamodelos, se dio pié a los perfiles para el modelado de los *PIM* de la aplicación, extendiendo a UML con las características nuevas propias del DSL MoWebA, específicamente a lo concerniente a contenido y posición.

En este capítulo, describiremos primeramente a la herramienta de transformación de modelo a texto *M2T* *Acceleo*, basado en plantillas, en la cual se describirán algunas características importantes de la herramienta, como así también, los pasos a seguir para llevar a cabo las transformaciones sobre los PIM de entrada. Seguidamente se presentará a la plantillas para llevar a cabo la transformación de los PIM correspondiente al contenido y a las plantillas de transformación para establecer la configuración de las posiciones de cada uno de los elementos del PIM.

Finalmente se concluirá el capítulo con un ejemplo de transformación M2T para PIM modelados con MoWebA.

**4.2 La herramienta de transformación M2T Acceleo.**

Acceleo es una herramienta de transformación M2T basada en los estándares propuestos por la OMG y que actualmente forma parte de la *Eclipse Foundation*. Acceleo es el resultado de varios años de investigación y desarrollo en el área de los lenguajes de transformación de modelos (MTL) .

**4.2.1 Algunas características importantes de Acceleo**

**Open Source**

Acceleo es un generador de código *open source*. Como tal es posible utilizarlo, bifurcarlo y contribuir con el el.

**Integrado con el Eclipse**

Acceleo está integrado con el *IDE* del *Eclipse*, un editor completo, con corrector de sintaxis, detección de errores en tiempo real, soluciones rápidas, refactorización y mucho más. También contiene vistas dedicadas que ayudan a navegar amigablemente por el generador de código

**Trazabilidad**

Con el generador de código, es fácil perderse en el código generado. De manera a manejar este inconveniente, Acceleo contiene un motor de trazabilidad que permiten encontrar fácilmente que elementos del modelo y que parte del generador (plantilla de transformación) ha sido utilizado para generar la pieza de código.

**Versatilidad**

Generadores de código son a menudo limitados a un conjunto de tecnologías. Con el enfoque basado en templates, Acceleo puede generar código para cualquier tipo de lenguajes. Si es posible escribirlo, Acceleo puede generarlo.

**Generación incremental**

En algún momento podría considerarse adecuado modificar manualmente el código generado por el *template* de transformación y mantener las modificaciones manuales realizadas, en caso que se desea regenerar el código de la aplicación. Acceleo permite definir *protected areas* en al cual se pueden llevar a cabo modificaciones al código generado de manera segura, protegiendo los cambios manuales hecho al código generado.

El lenguaje de plantilla de Acceleo ofrece diversos meta-marcadores que son llamados tags en Acceleo y que son comunes también en otros lenguajes de transfromación M2T.

Archivos Para generar código, los archivos deben ser abiertos, llenados y posteriormente cerrados. En Acceleo, existen un tag de archvio especial  *lo cual es utilizado*  para imprimirel contenido que es creado entrel el comienzo y el final del del tag file para un archivo en particular. La ruta y el nombre de archivo ambos son definidos por un atributo del tag.

Estructura de control: Existen tags para definir estructuras de control tales como loops (for tag) para iterrar entre colecciones de elementos, por ejemplo, son especialmente útiles para trabajar con referencias multi-valuadas para los resultados navegados en una colección de elementos y en ramas condicionales (if tag).

Consultas: Las consultas OCL pueden ser definidas por medio del tag *query*. Los queries pueden ser llamados a travez de todo el template y pueden ser utilizados para factorizar código que es recurrente.

Expresiones: Existen expreseiones generales para incluir los valores de las expresiones computadas en el texto generado de manera a producir las partes dinámicas del texto de salida. Las expresiones pueden también ser utilizadoas para llamar otros templates para inclir el código generadopaor el template llamado en el código producido por template llamante. Llamando a otros templates puede ser comparado a los métodos en el lenguaje Java.

Areas protegidas: Una característica importante de los lenguajes M2T es el soprote de proyectos donde solamente generación parcial de código es posible. En particular, soporte especial en necesario para guardar código manuelamente aderido en el código generado en ejecuciones subsecuentes del generador de código. Para estar tarea, un concepto especial llamado protected areas ha provado ser útil y es sportado por el Acceleo por medio del tag protected. Los protected areas, son utilizado para marcar secciones en el código generado que no deben ser sobreescritos de nuevo luego de una nueva ejecución del generador de código. Esta sección típicamente contiene código manuelmente escrito.

4.2.2 Entorno de trabajo con Acceleo.

4.2.3 Plantillas de transformación de modelo a texto (Templates).

4.2.4 Envolvedores de servicios (Service Wrappers).

4.2.5 Operaciones y expresiones OCL en Acceleo.

4.3 Integración de MoWebA con Acceleo.

4.3.1 Introducción.

4.3.2 Transformación de los modelos de MoWebA de MOF a EMF UML2 (v2.x) XMI.

4.3.3 Exportaciòn de los PIM's y perfiles MoWebA EMF UML2 (v2.x) XMI al Acceleo.

4.5 Integración de JQueryUI y JQwuery Validadte a Acceleo.

4.6 Generación de código para la interfaz de usuario de MoWebA con Acceleo.

4.6.1 Introducción

4.6.2 Plantilla de transformación para los elementos del perfil de contenido.

4.6.3 Plantilla de transformación para el posicionamiento de los elementos de contenido.

4.7 Conclusiones