

# Una plataforma para la gestión de modelos

Artur Boronat, José Á. Carsí, Isidro Ramos

Departamento de Sistemas Informáticos y Computación  
Universidad Politécnica de Valencia  
C/Camino de Vera s/n  
E-46071 Valencia- España  
{aboronat | pcarsi | iramos}@dsic.upv.es

**Resumen.** La gestión de modelos engloba todo tipo de tareas que permiten su representación y/o manipulación, entre ellas, la evolución de sistemas software, la migración de datos, la interoperabilidad, etc. En el trabajo se presenta una plataforma resultado de nuestra experiencia previa en varios proyectos industriales de migración de datos y de recuperación automática de aplicaciones legadas. La plataforma ha sido concebida y realizada en un contexto formal lo que permite deducción y verificación automática de propiedades de los modelos considerados. En nuestra aproximación, los modelos están representados por términos de un álgebra construidos según un determinado orden canónico. La manipulación de modelos se centra en el establecimiento de correspondencias entre los términos que representan dos o más modelos. Estas correspondencias son expresadas mediante morfismos que permiten generar nuevos modelos y establecer relaciones entre modelos existentes. El uso de herramientas declarativas para la creación de los modelos y de los morfismos proporciona una gran modularidad que favorece la utilización de técnicas de patronaje para realizar estas correspondencias y la generación automática de modelos mediante reescritura de términos.

**Palabras clave:** gestión de modelos, RDF, metainformación, mantenimiento de software.

## 1. Introducción

Según [9], “modelo” es una estructura compleja que representa un artefacto de diseño, como un esquema relacional, una interfaz orientada a objeto, un modelo UML, un DTD de XML, el esquema de una página web, una red semántica, un documento complejo, o una configuración de software. La gestión de modelos engloba todo tipo de tareas que permiten su representación y/o manipulación. El uso de modelos conlleva la gestión de cambios en los modelos y las transformaciones de información de un modelo a otro, que implican la representación explícita de correspondencias entre estos modelos.

Este trabajo se centra en la gestión de modelos fruto de la experiencia previa en dos proyectos industriales de migración de datos y de recuperación automática de aplicaciones legadas. El primero (ver [4]) presenta una solución para la migración de información entre esquemas conceptuales orientados a objeto cuya capa de

persistencia está formada por esquemas relacionales. Se aplica un algoritmo de comparación de esquemas cuyo resultado es analizado mediante patrones que generan un plan de migración relacional. Este plan de migración establece las correspondencias entre elementos de ambas bases de datos indicando cómo se realiza la copia de información. El segundo (ver [6]) presenta una solución y un método para la recuperación de bases de datos legadas procedentes de varios SGBD usando técnicas basadas en métodos formales y patrones. Los métodos formales (sistemas de reescritura de términos) se aplican en el proceso de ingeniería inversa de datos y las técnicas de patronaje se utilizan durante el proceso de recuperación de los datos de la base de datos legada.

Ambos proyectos coinciden en el tratamiento de modelos de información mediante técnicas formales como la reescritura de términos o la utilización de patrones para conseguir una mayor automatización del proceso y así reducir su coste. Mediante la plataforma de gestión de modelos se pretende definir una base formal que permita representar cualquier modelo y trabajar de forma genérica con estas técnicas. El objetivo de la plataforma no se restringe al campo de aplicación presentado por ambos proyectos anteriores sino que es la base que permitirá realizarlos de una forma integrada. A la vez, se dará soporte a proyectos con otros objetivos como la interoperabilidad, la gestión automática de recursos web, etc.

En esta sección se ha introducido el marco en el que se sitúa el trabajo. En el apartado 2 se proporciona una situación real en la que se necesita recuperar un sistema legado. En el apartado 3 se describe la estructura de la plataforma para gestión de modelos. En el apartado 4 se describe la aplicación de la plataforma para resolver el problema propuesto. Finalmente se describen trabajos relacionados, las conclusiones y los trabajos futuros.

## **2. Motivación: problema de la recuperación de un sistema legado**

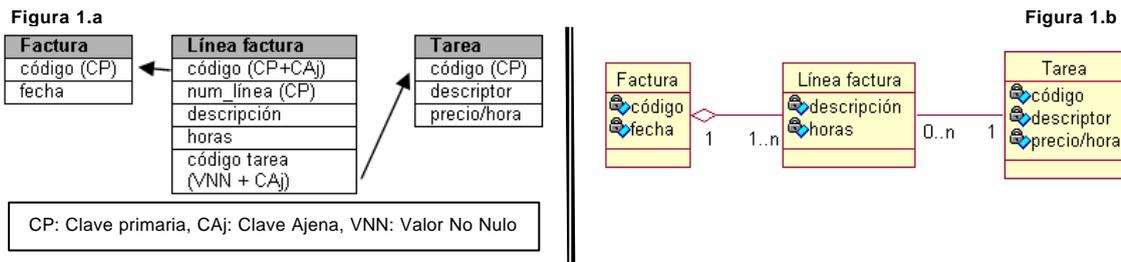
En esta sección se expone un ejemplo en el que se requiere la traducción de modelos de datos construidos mediante diferentes tecnologías. Una empresa dedicada al sector de la venta de inmuebles adquiere la empresa que administra los jardines de los inmuebles vendidos. La inmobiliaria gestiona su información mediante una aplicación orientada a objetos (OO), que ha sido modelada mediante OASIS y cuya información se almacena en una base de datos objeto relacional. Mientras que la empresa de jardinería utiliza una aplicación de gestión que almacena su información en una base de datos relacional. El objetivo de la empresa consiste en la aplicación de una política global sobre sus clientes mediante el aprovechamiento de la información de ambas aplicaciones. Considerando el esquema conceptual OASIS de la aplicación de gestión de la inmobiliaria y el esquema de la base de datos relacional de la empresa jardinera como modelos, para conseguir el objetivo propuesto se debe integrar el conocimiento inmerso en ambos modelos.

La empresa decide mantener la aplicación OO de la empresa inmobiliaria (en adelante llamaremos base de datos legada a la que utiliza la empresa de jardinería) y para integrar ambas aplicaciones se barajan dos posibilidades. La primera consiste en crear un wrapper que proporcione los mecanismos necesarios para acceder a la

información de la base de datos legada. Esta opción tiene un alto coste de implantación y de mantenimiento porque deben convivir dos tecnologías diferentes. Una segunda opción consiste en extender la aplicación OO con la información de la base de datos de la aplicación de administración de jardines. Esta opción facilita el mantenimiento de la futura aplicación pero sigue teniendo un alto coste de implantación, puesto que requiere la modificación de la aplicación OO y la migración de la información de la base de datos legada hacia la nueva base de datos.

La plataforma para la gestión de modelos puede aplicarse en esta segunda opción proporcionando la automatización del proceso de generación del nuevo modelo así como de la migración de información legada hacia la nueva base de datos. El trabajo manual se centraría en la incorporación de nueva funcionalidad a la aplicación web para realizar la gestión de jardines.

En la figura 1.a se presentan las tablas de una parte del esquema relacional de la base de datos legada, en la que se indica que una factura del trabajo realizado en un jardín recoge información sobre las diferentes tareas realizadas en él. Por cada tarea se recogen las horas de trabajo con el fin de calcular el importe de la factura. En la figura 1.b se presenta el esquema conceptual OO semánticamente equivalente al esquema relacional anterior que se obtiene para integrar el conocimiento de la base de datos legada al nuevo producto software.



**Fig. 1.** Esquema relacional de la base de datos legada (figura 1.a) y el modelo orientado a objetos equivalente (figura 1.b)

A continuación se describe la estructura de la plataforma de gestión de modelos y se ilustra la aplicación de la plataforma al problema expuesto.

### 3. Visión global

La plataforma permite la gestión y la manipulación de modelos de datos así como de la información que permiten almacenar (para una descripción más detallada de la plataforma ver [1]). La estructura básica utilizada para representar modelos de datos es la tripleta RDF sujeto-predicado-objeto (ver [2]), que permite describir las propiedades *objeto* que posee un elemento *sujeto* mediante relaciones *predicado*. La versatilidad de esta estructura se basa en la posibilidad de utilizar un elemento *sujeto* como propiedad *objeto* de otro elemento *sujeto* distinto. De esta forma, se pueden crear estructuras en forma de grafo que representen cualquier modelo de datos. La representación de modelos como términos de un álgebra permite la utilización de

técnicas formales tanto para la representación de un modelo como para establecer relaciones entre ellos. Esta estructura básica proporciona una gran modularidad que facilita la aplicación de técnicas de patronaje para realizar correspondencias entre modelos y reglas de reescritura para la generación automática de modelos.

La plataforma está formada por una jerarquía de cuatro niveles, donde cada nivel agrupa elementos con el mismo grado de abstracción. De esta manera, los elementos que constituyen un metamodelo son definidos en un nivel de abstracción. Los elementos que constituyen un esquema conceptual de ese metamodelo se sitúan en un nivel más concreto que el anterior. Por último, las instancias del esquema conceptual citado se ubican en un nivel aún más concreto. Los niveles que forman la plataforma son los siguientes, por orden de abstracción:

1. Nivel Gestión de Modelos: forma el núcleo de la plataforma de gestión de modelos y proporciona un mecanismo común de definición de metamodelos para diferentes tipos de modelos de datos.
2. Nivel Metamodelo: alberga los metamodelos que han sido definidos a partir del Nivel de Gestión de Modelos y permite definir modelos de datos en el nivel de Modelo.
3. Nivel Modelo: contiene los esquemas conceptuales que representan la información mediante estructuras de datos pertenecientes a un metamodelo determinado.
4. Nivel Información: en este nivel nos encontramos con la información que es tratada y almacenada en el modelo de datos.

Los elementos de un nivel intermedio pueden ser interpretados mediante el concepto de metaobjeto [3]. De esta manera pueden ser vistos como clases al describir elementos de niveles inferiores (más concretos) y como objetos al estar descritos por elementos de niveles superiores (más abstractos). Se utilizan dos mecanismos que permiten el cambio de abstracción de los elementos que pertenecen a un determinado nivel: la reflexión, que indica cómo un elemento específico del nivel  $N$  define un conjunto de elementos del nivel inmediatamente más concreto ( $N+1$ ); y la reificación, mediante la cuál se indica, para un determinado elemento situado en un nivel  $N$ , el elemento del nivel inmediatamente más abstracto ( $N-1$ ) que lo describe.

#### **4. Aplicación: solución al problema de la recuperación de un sistema legado**

En esta sección se describe la utilización de la plataforma para resolver el problema propuesto en el apartado 2. El proceso se realiza en 2 fases: la generación del modelo orientado a objetos y la migración de información.

En primer lugar, se definen el metamodelo relacional y el metamodelo OASIS en el nivel de metamodelo, como instancias del esquema de gestión de modelos del nivel superior. Entre los términos de ambos metamodelos se definen morfismos que permiten traducir cualquier esquema relacional a un esquema conceptual OASIS de una forma automatizada. De esta manera, el esquema relacional de la figura 1.a se define en el nivel de modelo como instancia del metamodelo relacional. A continuación se aplican los morfismos definidos sobre el metamodelo relacional para generar el esquema conceptual de la figura 1.b como instancia del metamodelo

OASIS. Cabe resaltar que únicamente se genera la parte estática del esquema conceptual.

En segundo lugar, se procede a migrar información desde la base de datos legada hacia la base de datos de la nueva aplicación. Partiendo de los morfismos de traducción se establecen correspondencias entre los elementos del esquema relacional y del modelo orientado a objetos recién generado. Estas correspondencias se plasman en un plan de migración al que se añade el orden en el que deben ser migrados la información del esquema relacional hacia la base de datos del modelo orientado a objetos. Por último, este plan de migración se compila a una serie de módulos que se conectan a la base de datos legada y a la destino y realizan la migración de información en el orden indicado, aplicando a los datos las transformaciones necesarias. La figura 2 ilustra el proceso descrito.

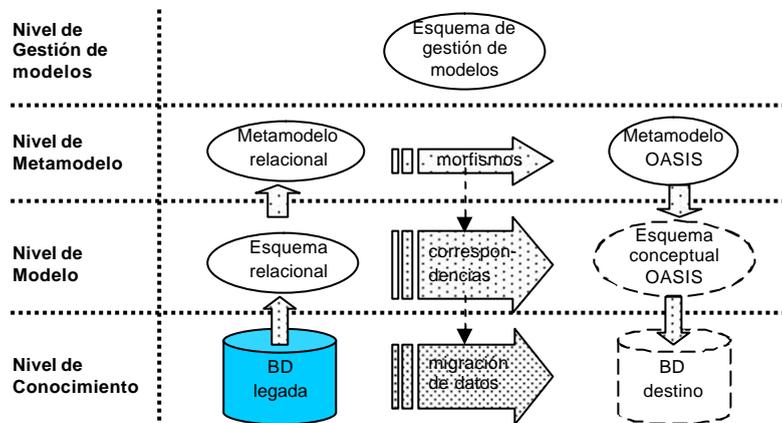


Fig. 2. Recuperación de sistemas de información legados mediante la plataforma de gestión de modelos

## 5. Trabajos relacionados

Existen herramientas que facilitan la gestión de modelos. Entre ellas destaca RONDO (ver [9]) que propone la representación de modelos mediante teoría de grafos y una serie de operadores de alto nivel que permiten manipular modelos y las correspondencias entre ellos. Los modelos se traducen a grafos mediante conversores específicos para cada tipo de modelo que deben tener en cuenta consideraciones sobre las operaciones de manipulación para que se puedan aplicar sobre los modelos resultantes.

Las propiedades semiforales que proporciona el lenguaje RDF facilitan la aplicación del uso de técnicas formales como los morfismos basados en reescritura de términos [6]. Este hecho aporta una aproximación automática y correcta a la hora de validar modelos y realizar correspondencias entre ellos, evitando el desarrollo de complejos algoritmos. Por otra parte, la plataforma permite definir tanto los metamodelos utilizados para gestionar modelos, como los modelos utilizados para

gestionar información, además de establecer correspondencias entre elementos de un mismo nivel de abstracción. De esta manera, se ofrece la posibilidad de definir nuevos metamodelos y relacionarlos con otros existentes abriendo la herramienta a todo aquél que desee utilizarla.

## 6. Conclusiones y trabajo futuro

Se ha presentado una plataforma semántica que facilita la gestión de modelos. Esta plataforma está formada por cuatro niveles de abstracción cuyos elementos se relacionan mediante mecanismos de reflexión y de reificación. La existencia de diferentes niveles de abstracción proporciona una gran versatilidad y potencia a la plataforma de gestión de modelos puesto que permite crear correspondencias entre esquemas conceptuales en diferentes niveles de abstracción. Además, las correspondencias entre los elementos de un esquema conceptual a un determinado nivel permite la generación automática de correspondencias en niveles más concretos. Este hecho se utiliza para la generación de esquemas conceptuales en el Nivel de Modelo a partir de las correspondencias definidas entre metamodelos, generación de planes de migración entre esquemas conceptuales a Nivel de Información partiendo de las correspondencias establecidas a Nivel de Modelo, etc.

En el ejemplo descrito se ha resuelto la recuperación de un sistema legado generando la parte estática de un esquema conceptual OO, pero no se ha tenido en cuenta la integración con el sistema de información existente (el de la inmobiliaria). Este es un problema interesante que abordaremos estudiando los conflictos que puedan aparecer al migrar información hacia una base de datos que ya está poblada. Entre ellos destacamos el problema de la preservación de la identificación de las instancias de las clases del esquema conceptual destino.

## 7. Bibliografía

1. Boronat A., Carsí J.A., Ramos I., *Una plataforma semántica para la gestión de modelos*, Informe Técnico DSIC-II/17/03, Univ. Politécnica de Valencia, Junio 2003.
2. Brickley D., Guha R.V. (Editors), *RDF Vocabulary Description Language 1.0: RDF Schema*, World Wide Web Consortium, November 2002.
3. Carsí J.A., *OASIS como marco conceptual para la evolución del software*, Tesis doctoral, DSIC, Univ. Politécnica de Valencia, ISBN 84-699-3372-8. 1999.
4. Carsí J.A., Ramos I., Silva J., Perez J., Anaya S., *Un generador automático de planes de migración de datos*, Electronic Journal (I+D Computación) on Computer Science Research and Development of the International Congress on Computer Science Research (CIICC), The National Academy of Computer Science (Mexico), ISSN: 1665-238X, Julio 2002.
5. Melnik S., Rahm E., Bernstein P. A., Rondo: A Programming Platform for Model Management, Proc. ACM SIGMOD 2003, San Diego, June 2003.
6. Perez J., Anaya V., Cubel J.M., Domínguez F., Boronat A., Ramos I., Carsí J.A., *Data Reverse Engineering of Legacy Databases to Object Oriented Conceptual Schemas*, SET 2002, Software Evolution Through Transformations: Towards uniform support throughout the software life-cycle, Barcelona - Spain, October 2002.