

MDE en la generación de aplicaciones para Repositorios Institucionales

Jose Texier^{1,2}, Marisa De Giusti³, Silvia Gordillo⁴

¹ Universidad Nacional de Chilecito, Argentina
jtexier@undec.edu.ar

² Universidad Nacional Experimental del Táchira, Venezuela
jtexier@unet.edu.ve

³ SEDICI, Universidad Nacional de La Plata, Argentina. Comisión de Investigaciones Científicas de la Provincia de Buenos Aires (CIC), Argentina
marisa.degiusti@sedici.unlp.edu.ar

⁴ Lifia, Universidad Nacional de La Plata, Argentina
gordillo@lifia.info.unlp.edu.ar

Resumen. En el 2012 el Repositorio Institucional de la Universidad Nacional de La Plata, SEDICI, realizó un proceso de migración de Celsius DL a DSpace, donde se evidenció el problema de la representación de recursos, problema recurrente estudiado por algunos autores, no obstante, los trabajos revisados abordan el tema en forma general, no se toma en cuenta el recurso como el eje central. El objetivo central fue dar una solución al problema de la representación de recursos en SEDICI. La solución se planteó en desarrollar un marco de referencia que permitió el desarrollo de aplicaciones, replicable a otros repositorios y bajo el paradigma Model Driven Engineering (MDE) para la implementación de la solución. El marco de referencia se estructuró en 5 módulos. Esta investigación dió respuesta al objetivo planteado y vinculó premisas devenidas de tres disciplinas: Ciencias de la Información, Ciencias Documentales y Ciencias de la Computación.

Palabras clave: MDE, repositorios institucionales, representación de recursos, DSpace.

1 Introducción

En el 2003, nace el Servicio de Difusión de la Creación Intelectual (SEDICI) como el Repositorio Institucional (RI) de la Universidad Nacional de La Plata (UNLP), con el objetivo prioritario de socializar el conocimiento generado en las diferentes áreas académicas de la Universidad, para devolver a la comunidad los esfuerzos destinados a la Universidad Pública, bajo políticas explícitas de contenidos, metadatos, datos, entre otras, que se aprecian en su portal web. SEDICI, como un portal de acceso libre, estuvo soportado por un desarrollo de software propio en PHP, MySQL y Java, llamado Celsius DL, que estaba adaptado a estándares

internacionales como XML, OAI-PMH, SOAP, y permitía el depósito y búsqueda de recursos. Celsius DL fue creciendo en funcionalidades por lo que el diseño, mantenimiento e implementación convirtieron a la plataforma de software de SEDICI en una herramienta compleja principalmente por dos razones: por una parte porque se requería de mayor tiempo y recurso humano y, por otra, porque se hacía necesaria una actualización de tecnologías de manera que no estuvieran en desventaja frente a nuevos desarrollos e investigaciones.

Por estos motivos para finales del 2011, se estudió la posibilidad de migrar a una plataforma que estuviera a la par de las nuevas tecnologías aplicadas al dominio y que fuera más amena para el usuario y para la gestión de recursos por parte del personal de SEDICI, así como a los fines de la preservación del material. La conclusión del estudio determinó que DSpace era la herramienta de software que mejor se adaptaba a las necesidades de SEDICI. El proceso de migración de Celsius DL a DSpace finalizó en el 2012 (contaba con aproximadamente 15.000 recursos). Esto permitió contar con nuevas funcionalidades y se logró una flexibilidad en los cambios estratégicos por parte de la gerencia del repositorio, modificaciones en la imagen institucional sin afectar la lógica del repositorio y ampliar la gestión de las tipologías de los recursos aceptados.

DSpace, al igual que otras plataformas, presenta limitaciones en diferentes ámbitos tales como: procesos de depósitos, manejo de estadísticas y de las comunidades-colecciones-items, vocabularios controlados centrados en los autores, representación de los recursos, etc. De estas limitaciones, la representación de los recursos -que consiste en el proceso de registrar en forma persistente un conjunto de datos como síntesis y reemplazo del objeto "real" para poder identificarlo, recuperarlo y distribuirlo- es uno de los principales problemas presentados en la transición de Celsius DL a DSpace, ya que se migraron muchos recursos en forma separada e incluso se realizaron adaptaciones por incompatibilidad de ambos sistemas porque la representación de distintos recursos era muy variada. Este proceso es llevado a cabo de diversas maneras en plataformas para repositorios. El trabajo de esta investigación se centró en solucionar el problema de la representación de recursos en SEDICI bajo el enfoque Model Driven Engineering (MDE).

La tipología de recursos de SEDICI a la que se limitó el trabajo fue: artículos de investigación, tesinas de grado y tesis de postgrado, libros, autores, instituciones, revistas y sus números, eventos y sus instancias, con la intención de ser replicado sobre cualquier otra tipología. Estos recursos usados cuentan con diferentes características que representan los metadatos de cada uno de ellos y agrupados en esquemas de metadatos. Los esquemas y los recursos se convierten en los elementos centrales para una representación de recursos básica dentro de los RI. A continuación se presentan algunas características a tomar en cuenta para una representación de recursos y que sirvieron de base para la propuesta desarrollada:

- Los recursos deben estar catalogados en más de un esquema de metadatos a fin de evitar pérdida de información.

- Debido a la gran diversidad en las estructuras (planas y jerárquicas) y restricciones de cada esquema de metadatos, es necesario plantear una solución de representación clara, flexible, factible, escalable, interoperable, mantenible y que no represente un desafío de configuración para los administradores, así como tampoco sea complicada su utilización por parte de los usuarios.
- Muchos campos presentes en los esquemas de metadatos necesitan ser estandarizados para garantizar una identificación y recuperación correcta, a fin de evitar la redundancia y garantizar la integridad de la información.
- En estos sistemas, frecuentemente, se encuentran elementos que poseen información descriptiva propia, conocidos como entidades abstractas que son representadas de forma separada. Por ello, es necesario permitir su reutilización en los diferentes procesos del repositorio y garantizar su representación.
- Las entidades abstractas pueden tener distintas representaciones según el esquema de metadatos en el que se represente el recurso.
- Todos los metadatos de los recursos de un repositorio institucional debe almacenarse de forma persistente. por ello, este almacenamiento debe ser configurable a cualquier tipo de paradigma de base de datos e independiente del modelo de representación de los recursos.
- Para una representación de los recursos básica, se deben tomar en cuenta los siguientes módulos: almacenamiento, catalogación, indexación, infraestructura de la plataforma de software y preservación.

2. Representación de recursos

La aparición de los Repositorios Institucionales (RI) ha generado un aumento en el uso de plataformas de software para RI y la necesidad de mantener dichas plataformas al mismo ritmo que otras tecnologías no específicas que rodean al repositorio. Por ello, la Representación de Recursos (RR) es uno de los principales problemas del mundo de los RI. Varios autores [4, 12, 20, 25] confirman tal situación y ponen en contexto del mundo científico el problema de la representación de recursos, problema que se ha complejizado poco a poco, principalmente, por la diversidad de plataformas de software y esquemas de metadatos existentes para representar los recursos de diferentes tipologías en los repositorios.

Algunas de las plataformas de software más usadas son DSpace, EPrints y Digital Commons [24, 28]. No obstante, se encuentran plataformas consolidadas de desarrollos propios que abarcan un gran porcentaje de los recursos esparcidos en el mundo como: arXiv, CiteSeerX, Social Science Research Network [7]. Se observa

que en estas diferentes plataformas se encuentran los recursos representados en diversos esquemas de metadatos. De hecho, muchos esquemas de metadatos se desarrollan de acuerdo con una variedad de usuarios y de disciplinas para facilitar la identificación, recuperación, utilización y/o gestión de recursos en los repositorios [6]. Por tanto, los esquemas de metadatos a través del conjunto de elementos diseñados en una estructura formal y relacionados con la semántica, la sintaxis y la obligatoriedad en sus valores, pasan a ser un elemento importante en la representación de recursos dentro de un RI.

Los esquemas de metadatos definidos como un conjunto de propiedades que contienen valores de acuerdo con un conjunto de reglas, han sido diseñados e implementados para permitir reusos de los registros e integración de los mismos. Estos metadatos proporcionan un nivel intermedio a través del cual puedan acceder a la información y se relacionan directamente con las características de un RI [11], independientemente de la plataforma de software elegida. Los esquemas de metadatos se aplican en muchos contextos, pero en el caso de este trabajo se aplican a los Recursos, los cuales son vistos como objetos físicos o digitales que se describen a partir de la enumeración de un conjunto de datos específicos (metadatos) que lo distinguen entre otros objetos [9, 16]. En algunas plataformas de repositorios, como DSpace, existen comunidades y colecciones, las primeras simbolizan a entidades administrativas de instituciones tales como departamentos, laboratorios, oficinas, centros de investigación, entre otros. En cambio, las colecciones simbolizan el lugar a donde pertenecen los recursos y no pueden contener otras categorías, estas deben encontrarse dentro de una comunidad.

El concepto de metadatos no es algo nuevo, antes de la aparición de Internet se usaba en la catalogación de libros y revistas para normalizar la información de manera que la misma se pudiera localizar y recuperar fácilmente. En el ámbito de las Ciencias de la Información, los metadatos se emplean para referirse a registros de recursos de información disponibles [16]. Los metadatos son datos que caracterizan a otros datos, es decir, es información estructurada que describe, explica y/o localiza un recurso de información para poder identificarlo, recuperarlo, utilizarlo, administrarlo o preservarlo de una manera más clara y sistemática. Para la representación de metadatos se han desarrollado distintos modelos, esquemas, formatos o estándares, que si bien en muchos casos comparten una sintaxis y estructura de la información en XML, difieren atendiendo a los propósitos de la información que describen [21]. A continuación se exponen los esquemas de metadatos más utilizados en el área de las Ciencias de la Información [8, 13]:

- Dublin Core (Dublin Core Metadata Initiative – DCMI).
- MARC 21 (MACHINE-Readable Cataloging - Century 21st - MARC).
- Metadata Object Description Schema (MODS).
- Metadata Authority Description Schema (MADS).

- Metadata Encoding and Transmission Standard (METS).
- Preservation Metadata Implementation Strategies (PREMIS).

Existen otros estándares de metadatos que se han desarrollado por distintos usuarios e instituciones y, que han ayudado a consolidar repositorios de datos en diferentes áreas. Algunos de los más conocidos son: Darwin Core, DDI (Data Documentation Initiative for Social and Behavioral Sciences Data), DIF (Directory Interchange Format for Scientific Data), EML (Ecological Metadata Language), NBII (National Biological Information Infrastructure), TEI (Text Encoding Initiative), ETD (Electronic Theses and Dissertations). Otros estudios presentan mapas visuales sobre la diversidad de estándares de metadatos para diferentes áreas, por ejemplo, Riley en el 2010 [27] realiza una investigación sobre el patrimonio cultural en el que se evidencian 105 estándares de metadatos y expone las relaciones sobre la base de cuatro categorías: comunidad, dominio, función y propósito.

El trabajo de Texier & De Giusti en el 2014 [30] describió un corpus que expone los problemas relacionados con la representación de recursos (RR), a saber: diversidad de soluciones tecnológicas, tratamiento de diferentes tipologías de recursos, esquemas de metadatos, recomendaciones de almacenamientos de recursos, la preservación de recursos, la recomendaciones del modelo OAIS y la aplicación de modelos conceptuales para solucionar el problema de la RR como un todo, tales como el modelo FRBR (Functional Requirements for Bibliographic Records), FRAD (Functional Requirements for Authority Data) y FRASAD (Functional Requirements for Subject Authority Data) [1, 3, 17, 22, 31, 33]. A partir del corpus se pudieron determinar los factores que inciden en una RR, los cuales se agruparon en los siguientes ejes de análisis: recursos, esquemas de metadatos, almacenamiento, catalogación, incorporación de recursos, gestión de los datos, acceso.

3. Model Driven Engineering (MDE)

En vista de la diversidad de paradigmas para la construcción de software, la comunidad de desarrolladores de software se ha sensibilizado respecto de la necesidad de formalizar la construcción de componentes de software y guiar tales desarrollos a través de modelos definidos. Tales modelos son un conjunto de elementos que sirven para demostrar la consistencia de una teoría, es decir, representan con detalle un sistema dado [32]. Este enfoque es conocido como Model-Driven, se define como una metodología de desarrollo de software que permite aplicar las ventajas del modelado en actividades de ingeniería de software [2].

El framework que propone el paradigma Model-Driven, permite una serie de beneficios que facilita la construcción de un marco de referencia que puede ser replicado en las diferentes plataformas de software existentes, ya que la representación de recursos debe ser general y a través de los metamodelos y modelos, se pueda adaptar a los diversos repositorios institucionales.

El MDE (Model-Driven Engineering) puede ser visto como el superconjunto de todas las variantes de la metodología Model-Driven, que son frecuentemente referidas como el acrónimo MD* (Model Driven star), entonces, cualquier enfoque MD*E podría estar bajo el paraguas MDE [2]. Por ejemplo: MDSE (Model Driven Software Engineering), MDPE (Model Driven Product Engineering), MDWE (Model Driven Web Engineering), entre otros. En MDE se puede aplicar en diferentes niveles de abstracción para poder proporcionar una visión integral en el desarrollo del sistema de software y tiene su base en Model-Driven Development (MDD) y Model-Driven Architecture (MDA) [18, 23, 26]. MDE busca soluciones de acuerdo con dos dimensiones (ver Figura 1): conceptualización (eje de las columnas) e implementación (eje de las filas). La implementación se observa con el mapeo de los modelos de funcionamiento de algunos sistemas existentes o futuros. De acuerdo con la Figura 2, se definen tres aspectos fundamentales [2]:

- El nivel de modelado: donde se definen los modelos.
- El nivel de automatización: donde se ponen las asignaciones de la modelización a los niveles de realización en su lugar.
- El nivel de realización: donde las soluciones se implementan a través de artefactos que se encuentran actualmente en uso dentro de los sistemas de funcionamiento.

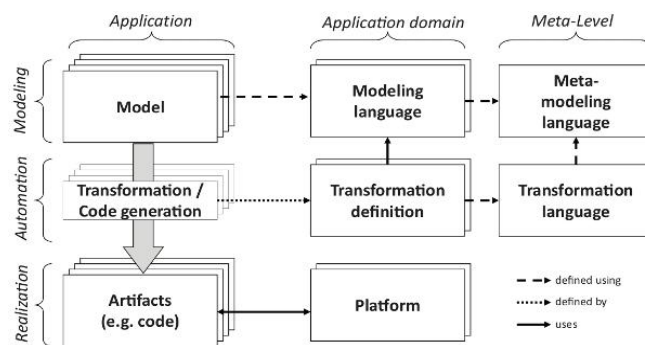


Fig 1. Visión general de MDE. Fuente: [2]

La conceptualización está orientada a la definición de los modelos conceptuales para describir la realidad. De acuerdo con la Figura 1, pueden aplicarse en tres niveles principales [2]:

- El nivel de aplicación: donde se definen los modelos de las aplicaciones, las reglas de transformación que son llevadas a cabo, y los componentes de funcionamiento que se generan.
- El nivel de dominio de aplicación: donde se definen el lenguaje de modelado, las transformaciones y las plataformas de aplicación para un dominio específico.

- El meta-nivel: donde la conceptualización de los modelos y de las transformaciones son definidas.

4. Marco de referencia

La solución propuesta en esta investigación tiene base en el enfoque MDE y se centró en el diseño de un modelo flexible e independiente de la plataforma para la representación de recursos bajo el paradigma MDE (ver Fig. 2), la cual se relaciona con los siguientes elementos de los repositorios: recursos, esquemas de metadatos, almacenamiento y catalogación, y que se basa en los siguientes procesos funcionales de acuerdo con el material depositado y la norma ISO 14721 [5]: carga (ingest), almacenamiento (storage), catalogación (cataloging), indización (indexing), búsqueda (search engine) y navegación (browsing). Para lograrlo, se hizo un relevamiento bibliográfico [29] para la construcción de metamodelos y se analizaron diferentes plataformas de repositorios que permitió comprender la diversidad de estándares y servicios que se proveen (a partir de sus modelos de datos y modelos conceptuales) para construir un marco de referencia que soporte el modelo propuesto, que sea independiente de la plataforma y adaptable a cualquier propuesta deseada. En este trabajo, el modelo desarrollado tuvo como base la norma ISO 14721 [5]. La propuesta se estructuró en 5 módulos o fases:

- DSL para el desarrollo del modelo flexible.
- Transformación del modelo a un modelo relacional.
- Transformación del modelo relacional a un script SQL para la creación de la base de datos.
- Mapping de los recursos de Dspace-SEDICI a la base de datos creada.
- Desarrollo de una aplicación en WebRatio para la visualización de los recursos que se encuentran en la base de datos propia.

Para cada una de estas fases se realizó: un análisis, una implementación y una descripción de sus beneficios, siempre sobre la base de una solución dirigida por modelos al problema de la representación de recursos.

En la Figura 2 se observa la propuesta de obtener un modelo que represente a los recursos a partir de un DSL gráfico, en cual debe ir transformándose en otro modelo hasta llegar a una base de datos relacional, en la cual se encontrará los diferentes recursos que representan el RI. Luego, a partir de esa base de datos se desarrollaran funcionalidades para el RI a través de WebRatio, gracias a los modelos que exige dicha herramienta.

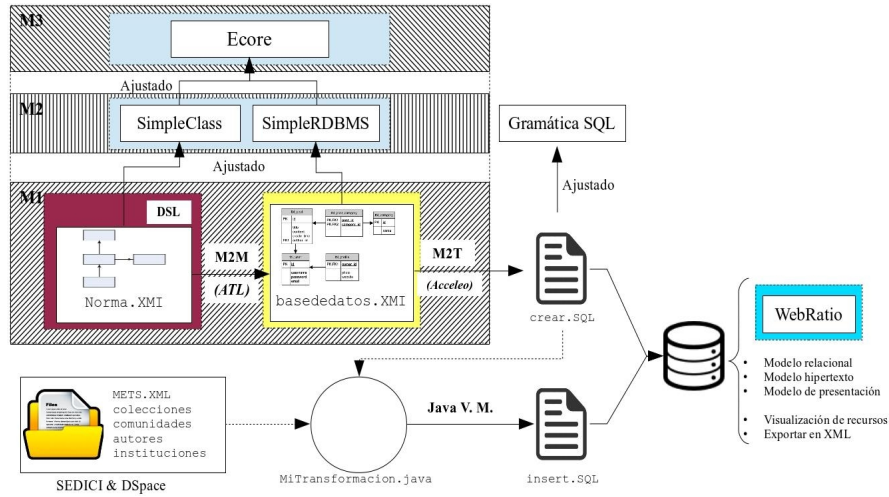


Fig. 2. Marco de referencia desarrollado

El ideal bajo el enfoque MDE para los actores involucrados (desarrolladores, dueños del negocio y expertos del dominio) en el diseño y desarrollo de un sistema para repositorios, es que exista un lenguaje neutral y de alto nivel que permita que las partes se pongan de acuerdo y que les sirva de apoyo para describir, discutir y negociar las funciones (destinados a recopilar, catalogar, almacenar, gestionar, acceder, difundir y preservar) que el repositorio debe ofrecer [29]. Este paradigma ofrece algunas ventajas como: incremento en la productividad (errores, costos, código), adaptación a cambios tecnológicos, reuso de software, mejora en la comunicación con usuarios y desarrolladores, asignación de roles, entre otros [26]. En la Tabla 1, se puede visualizar los artefactos de entrada y salida para las 5 etapas definidas:

Fases	Entrada	Salida	El desarrollo
1	* SimpleClass.ecore * Information Package ISO 14721	* Modelo PIM	* GMF (DSL) * Sintaxis abstracta y concreta
2	* Modelo PIM * SimpleRDBMS.ecore	* Modelo PSM	* ATL (M2M)
3	* Modelo PSM	* Script creación SQL	* Acceleo (M2T)
4	* Archivos exportados SEDIC-Dspace * Script creación SQL	* Script insert.SQL	* Java (T2T)
5	* Base de datos (creación & inserts)	* Aplicación Web	* WebRatio

Tabla 1. Artefactos de entrada y salida por fases

En la Fase 1, el objetivo consistió en proponer un DSL gráfico que permite diseñar un modelo flexible e independiente (PIM) que represente los recursos dentro de SEDICI como caso de estudio, y que sirva de modelo para otros sistemas de

repositorios, sobre la base de la recomendación de conceptos, relaciones y semántica de la norma ISO 14721, a través del modelo Information Package [5]. El desarrollar un DSL permite hacer ajustes según las partes involucradas del dominio. En este trabajo se crean entidades a partir de nodos en la barra de herramientas (frame de trabajo), completar los campos y relacionarlos respetando las siguientes reglas:

- Una entidad puede ser un contenido de información, una descripción de la información (metadatos) o una entidad abstracta.
- Sólo se permiten tipos de datos definidos como datos primitivos.
- Las relaciones sólo pueden corresponder entre las entidades.
- Cuando se transforme el modelo a un modelo relacional, las entidades que estén sin alguna relación no serán tomadas en cuenta.

Este proceso de la Fase 1 permite el modelado entre las partes involucradas en la administración del repositorio bajo la misma (terminología). Para la construcción de un editor de modelos (el DSL) se debe tener un metamodelo que soporte el diseño (ajustado al metamodelo Ecore) de acuerdo con las especificaciones de la plataforma Eclipse Modeling Project (EMP). El modelo que se obtiene esta basado en la sintaxis permitida por el metamodelo (representado por un archivo XMI) a partir de una sintaxis abstracta y al menos una sintaxis concreta. La sintaxis abstracta define los diferentes elementos del lenguaje y las reglas que establecen cómo pueden ser combinados y la sintaxis concreta (una o más) define cómo los elementos del lenguaje aparecen en una notación textual o gráfica. Cada sintaxis concreta necesita establecer una relación (mapping) con la sintaxis abstracta. En otras palabras, la sintaxis abstracta del DSL es el metamodelo y la sintaxis concreta es la notación textual o gráfica para lograr el modelo. Este editor se construyó bajo el componente de software GMF Tooling (Graphical Modeling Framework Tooling) de EMP que hace parte del proyecto GMP (Graphical Modeling Project).

Para la Fase 2, se requirió trasladar el modelo obtenido del DSL a un modelo relacional (por eso es un PSM) que luego permita representar en forma persistente los recursos de SEDICI diseñados en la fase anterior, el cual debe estar ajustado a un metamodelo relacional. Tanto el PIM como el PSM deben encontrarse bajo los lineamientos (léxico, sintaxis y semántica) de un lenguaje acorde con un metalenguaje. Para lograr la transformación se seleccionó el paquete de software ATL, que soporta procesos M2M y comprende metalenguajes como Ecore.

El objetivo de la Etapa 3 consistió en realizar una transformación del modelo relacional a texto (M2T) para crear la base de datos de la propuesta. Este script sirvió de base para establecer los requerimientos de incorporación de recursos a la base de datos y que son utilizados en la Fase 4. Esta transformación se logró con el software Aceleo, que permite interpretar un modelo de entrada y obtener un texto deseado.

En la Fase 4, se crea un script para conformar la base de datos y establecer los lineamientos de incorporación de registros que provienen de SEDICI-DSpace en la

propuesta. Este proceso se pudo diseñar bajo el enfoque Model-Driven, pero debido al nivel de detalle y múltiples entradas de archivos, se desarrolló una rutina en Java para leer la información de entrada (siete archivos de textos) y generar un script de salida que representa las diferentes tuplas o registros de recursos que provienen de SEDICI para el sistema relacional obtenido en fases anteriores. En caso, de aplicar este modelo de referencia para otros repositorios, es en esta fase donde se realiza la adaptación a partir de la información de los recursos exportada del RI. Por ello, la flexibilidad e independencia de la propuesta.

Después de obtener un modelo propio que representó los recursos de SEDICI, se generó una aplicación web (Fase 5) para visualizar los recursos obtenidos y exportarlos como paquetes de información del archivo (AIP) de acuerdo con las recomendaciones de la norma ISO 14721.

5. Transformaciones realizadas

En busca de una solución a la representación de recursos dentro de SEDICI, específicamente se desarrollaron dos transformaciones bajo el enfoque MDE: una transformación del modelo diseñado con el DSL a un modelo relacional (M2M) y otra transformación del modelo relacional a un script SQL para crear la base de datos (M2T). Adicionalmente se realizó una transformación de texto a texto (T2T) y se ejecutaron transformaciones para la aplicación web, proceso propio de WebRatio.

Para la transformación M2M se usó ATL, que provee una interfaz fácil de usar y dentro de la plataforma EMP. Se requiere definir cuáles son los metamodelos para los modelos de entrada y salida, y luego, definir la transformación según la sintaxis de ATL. La Figura 3 muestra el esquema general ATL aplicado. El código fuente del proyecto ATL, se encuentra en un proyecto GitHub como “Fase 2” [10]. Los dos metamodelos que permitieron la implementación de los modelos de entrada (`norma.XMI`) y salida (`basededatos.XMI`) son tomados como base de los diferentes tutoriales y ejemplos que ofrece la plataforma EMP, es decir, `SimpleClass.ecore` y `SimpleRDBMS.ecore` respectivamente.

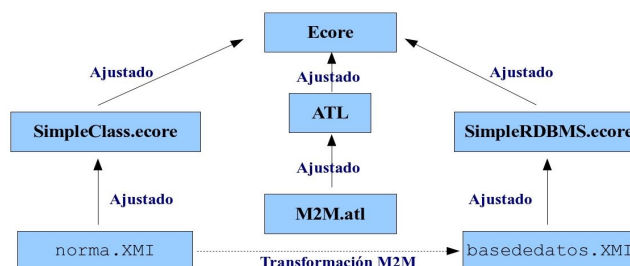


Fig. 3. Esquema de transformación M2M

La segunda transformación (M2T) se realizó con el paquete de software Acceleo y dentro del contexto de la plataforma EMP. Se definió cuál es el metamodelo que se ajusta el modelo de entrada de esta fase, además del código Acceleo (archivo con extensión .mtl) que permitió la transformación del modelo a texto. En la Figura 4 se observa el esquema general de la transformación M2T con Acceleo. Al igual que la anterior transformación, el código fuente de este módulo se encuentra en un proyecto GitHub “Fase 3” [10]. Es importante destacar, que primero se creó un proyecto Java, en el que se implementó un modelo generador del dominio para construir un editor en otra sesión de Eclipse, en la que se creó un proyecto Acceleo que reconoce el URI del metamodelo de entrada que se encuentra en el proyecto Java. Luego, a partir de ese proyecto se inició el proceso de ejecución de la transformación M2. Este proceso transforma el modelo de entrada (PSM), `basededatos.XMI`, en un archivo de texto llamado `creacion.SQL`, a partir del archivo `generate.mtl` en Acceleo.

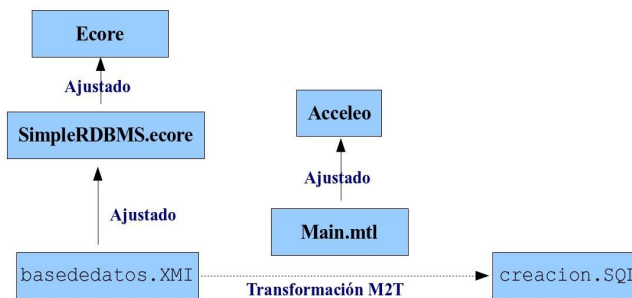


Fig. 4. Esquema de transformación M2T

La tercera transformación realizada (T2T) consistió en trasladar los recursos de SEDICI a la base de datos creada. De SEDICI proviene siete (7) archivos: recursos en formato AIP (en XML bajo el esquema de metadatos METS), información de las comunidades (desde una base de datos MySQL), las colecciones (MySQL), el listado de autores (MySQL), listado de las instituciones (MySQL), información de las revistas y sus números (MySQL). El programa en Java adaptó toda esa información al modelo relacional de acuerdo con los lineamientos recibidos del script de creación de la base de datos obtenida en la fase anterior. El resultado de este módulo es un script SQL de incorporación a la base de datos, llamado `insert.SQL`. El código fuente se encuentra en GitHub como “Fase 4” [10].

6. Funcionalidades desarrolladas

La evaluación del planteamiento realizado se hizo mediante la implementación de una aplicación que visualizó todos los recursos obtenidos y el modelo. Para ello, se seleccionó a WebRatio bajo el enfoque Model-Driven que permitió construir una aplicación web a partir de una base de datos existente, se sincronizó con la

plataforma de WebRatio. A continuación, se elaboró un modelo navegacional que sirve de base para generar la aplicación web deseada. El código fuente de todo este módulo se encuentra en un proyecto GitHub “Fase 5” [10]. Las aplicaciones web desarrolladas fueron dos: una, para visualizar todos los recursos importados desde DSpace-SEDICI adaptados al modelo propuesto, y, la otra aplicación, que permite exportar los recursos deseados en formato XML y ajustados a criterios similares (pero configurables) de las recomendaciones de la norma ISO 14721, es decir, exportar un paquete de información similar al AIP que sugiere la norma.

La metodología desarrollada en WebRatio se puede aplicar sobre cualquier repositorio, ya que es posible sincronizar y modificar el esquema de la base de datos del repositorio y realizar el diseño del modelo de hipertexto que después genera la aplicación web. Este módulo se centró en implementar dos funcionalidades a partir del modelo relacional. El modelo relacional obtenido podría integrarse en forma bidireccional con otros repositorios. Es importante destacar, las funcionalidades desarrolladas se limitaron al contexto de SEDICI con la plataforma DSpace, pero con un enfoque independiente y adaptable a cualquier otro caso de estudio.

7. Trabajos previos

En la publicación de Texier et al. en el 2014 [30] se analizaron trabajos que señalaban una relación entre el enfoque Model-Driven y el mundo LIS a partir de bases de datos de artículos y conferencias referentes del área. Se tomaron en cuenta 5 artículos resumidos en la línea de investigación de Malizia et al. [14, 15, 19, 20], y la línea de investigación de Paganelli & Pettenati en el 2006 [25] que se sintetizan.

En el trabajo de Paganelli & Pettenati se explica cómo los enfoques dirigidos por modelos pueden ayudar a los sistemas de gestión de documentos basados en modelos de información en XML. También describe las características de los documentos, los requerimientos para estos sistemas y un framework de desarrollo.

En el trabajo de Malizia et al. del 2010 describe un framework basado en un metamodelo y un lenguaje visual. Este framework, llamado por los autores CRADLE (Cooperative-Relational Approach to Digital Library Environments), es una definición de conceptos y servicios relacionados con el desarrollo de bibliotecas digitales compuesto por cinco entidades: actor, colección, servicios, estructura y documento. Este artículo reseña los antecedentes del enfoque dirigido por modelos en las Bibliotecas Digitales y describen la arquitectura y el metamodelo del framework. Las características principales del framework son: tener un lenguaje visual que ayuda a las partes interesadas integrando el modelado con la generación de código y en la definición de metadatos. Finalmente, estos autores presentan cómo es la generación de sistemas en este dominio y su evaluación.

Estos resultados reflejan un área de vacancia entre los modelos conceptuales para repositorios digitales y el desarrollo de software dirigido por modelos, es decir, el

Model-Driven no ha sido determinante en el diseño e implementación de sistemas de repositorios, ya sea por ser un paradigma nuevo o por no haber sido (aún) adoptado por la comunidad de informáticos, menos aún en la representación de recursos en los RI.

8. Conclusiones y trabajos futuros

Esta investigación permitió dar una solución al problema de la representación de recursos en SEDICI, donde se vincularon premisas devenidas de tres disciplinas: Ciencias de la Información, Ciencias Documentales y Ciencias de la Computación (LIS).

El desarrollar un modelo general, independiente de la plataforma y flexible que representó los recursos de un repositorio institucional bajo un enfoque Model-Driven mostró cómo los metadatos y las diferentes reglas de catalogación pueden convivir en esta propuesta de investigación en el tiempo, gracias al desarrollo de un DSL y las diferentes transformaciones realizadas. Para la solución se emplearon diversos lenguajes (Java, Aceleo, ATL, OCL, Ecore/MOF, WebRatio, SQL, XMI, XML) que hicieron de la propuesta una alternativa compleja y disímil bajo el enfoque MDE. A pesar de eso, se obtuvo un marco de referencia para diseñar un modelo para la representación de recursos. Se entiende que la propuesta se diseñó desde el punto de vista de los programadores, sin embargo, los dueños del negocio y los expertos del dominio pudieron participar en la primera fase de la propuesta (DSL), es decir, en el diseño del modelo general e independiente de la plataforma basado en las recomendaciones de la norma ISO 14721. Por tanto, la propuesta contó con dos puntos de vista: uno, desde los programadores que tienen una herramienta para la generación de aplicaciones en el dominio LIS garantizando como punto de partida la preservación de los recursos, y, el otro, desde la perspectiva del usuario final, que pueda hacer uso de dichos recursos a través de aplicaciones que nacen de este marco de referencia.

El enfoque Model-Driven fue transversal a toda la propuesta presentada sobre la base de la representación de recursos en los RI y permitió que los expertos del dominio y los dueños del negocio del dominio LIS se concentren en implementaciones de software más formales –consolidación de tales sistemas– cuyos principales beneficiarios serán los usuarios finales a través de los diferentes servicios que son y serán ofrecidos por estos sistemas.

Sobre la base de los resultados obtenidos surgen algunas líneas de trabajos futuro, tales como el marco de referencia desarrollado en esta investigación podría replicarse otras plataformas de software de repositorios como Eprints o FEDORA, incluso sobre otros repositorios en DSpace, de esta manera se puede realizar un análisis comparativo entre todos ellos. Adicionalmente el enfoque Model-Driven provee de muchas herramientas alternativas propietarias o con licencias abiertas, situación que

abre un espectro amplio de posibilidades de desarrollo de un marco de referencia como el que se propuso en este estudio, por ello, se puede realizar un estudio de las diferentes alternativas de herramientas de desarrollo de aplicaciones web siempre bajo el enfoque MDE.

Un área de estudio nueva en el enfoque Model-Driven es el desarrollo de transformaciones bidireccionales, que pueden ser de utilidad para ampliar la investigación presentada. Aunque esa bidireccionalidad esté sujeta a las personalizaciones de los sistemas de representación de recursos a los RI, situación que la hace más difícil y compleja.

Existen pruebas dirigidas por modelos que permiten validar y verificar el software desarrollado en esta propuesta, pero no hay mucho avance al respecto. Esas pruebas deben estar focalizadas en el proyecto interno (proyecto Model-Driven) que debe tener la participación de los usuarios finales para validar y verificar que se está desarrollando el software correcto. Existen otros tipos de pruebas que enfocadas al proyecto externo, que serán los dueños del negocio y expertos del dominio quienes validan y verifican las funcionalidades obtenidas.

Referencias

1. Altenhöner, R.: Data for the Future: the German Project “Co-operative Development of a Long-term Digital Information Archive” (kopal), <http://eprints.rclis.org/9199/>.
2. Brambilla, M. et al.: Model-Driven Software Engineering in Practice. Morgan & Claypool (2012).
3. Brindley, G. et al.: Provision of digital preservation metadata: a role for ONIX? Program Electron. Libr. Inf. Syst. 38, 4, 240–250 (2004).
4. Candela, L. et al.: Setting the Foundations of Digital Libraries. -Lib. 13, 3/4, (2007).
5. CCSDS: Reference Model for an Open Archival Information System (OAIS):ISO 14721. (2012).
6. Chan, L.M., Zeng, M.L.: Metadata Interoperability and Standardization - A Study of Methodology, Part I. -Lib Mag. 12, 6, 3– (2006).
7. CSIC: Ranking Web of Repositories, <http://repositories.webometrics.info/>.
8. Doderer, J.M. et al.: Metadata and Semantics Research: 6th Research Conference, MTSR 2012. Springer Publishing Company, Incorporated, Cadiz, Spain (2012).
9. Fox, E.A. et al.: Theoretical Foundations for Digital Libraries. The 5S Approach. Morgan & Claypool Publishers, North Carolina (2012).
10. GitHub, J.: Thesis Proposal, <https://github.com/dantexier/proposal>.
11. De Giusti, M. et al.: SEDICI – Desafíos y experiencias en la vida de un repositorio digital. RENATA. 1, 2, 16–33 (2011).
12. Gonçalves, M.A. et al.: Streams, structures, spaces, scenarios, societies (5s): A formal model for digital libraries. ACM Trans Inf Syst. 22, 2, 270–312 (2004).
13. Greenberg, J.: Understanding Metadata and Metadata Schemes. Cat. Classif. Q. 40, 3-4, 17–36 (2005).

14. Guerra, E. et al.: Model driven development of digital libraries-Validation, analysis and code generation. In: Webist 2007 - 3rd International Conference on Web Information Systems and Technologies, Proceedings. pp. 35–42 (2007).
15. Guerra, E. et al.: Supporting user-oriented analysis for multi-view domain-specific visual languages. *Inf. Softw. Technol.* 51, 4, 769–784 (2009).
16. Heery, R.: Review of metadata formats. *Program Electron. Libr. Inf. Syst.* 30, 4, 345–373 (1996).
17. Jeevan, V.K.J.: Digital library development: identifying sources of content for developing countries with special reference to India. *Int. Inf. Libr. Rev.* 36, 3, 185–197 (2004).
18. Little, S.W.: Model-Driven Software Development. In: Embley, D.W. and Thalheim, B. (eds.) *Handbook of Conceptual Modeling*. pp. 17–54 Springer Berlin Heidelberg (2011).
19. Malizia, A. et al.: A Cooperative-Relational Approach to Digital Libraries. In: Kovács, L. et al. (eds.) *Research and Advanced Technology for Digital Libraries*. pp. 75–86 Springer Berlin Heidelberg (2007).
20. Malizia, A. et al.: Generating Collaborative Systems for Digital Libraries: a Model-Driven Approach. *Inf. Technol. Libr.* 29, (2010).
21. Méndez, E.: La descripción de documentos electrónicos a través de metadatos : una visión para la Archivística desde la nueva e- Administración, <http://eprints.rclis.org/12684/>.
22. Mischo, W.H. et al.: The Growth of Electronic Journals in Libraries. *Sci. Technol. Libr.* 26, 3-4, 29–59 (2007).
23. Navarro, A. et al.: Architecture of a multiplatform virtual campus. *Softw. Pract. Exp.* (2011).
24. OpenDOAR: OpenDOAR - Home Page - Directory of Open Access Repositories, <http://www.opendoar.org/>.
25. Paganelli, F., Pettenati, M.C.: A Model-driven Method for the Design and Deployment of Web-based Document Management Systems. *J. Digit. Inf.* 6, 3, (2006).
26. Pons, C. et al.: *Desarrollo de Software Dirigido por Modelos*. Mc Graw Hill, La Plata, Argentina (2010).
27. Riley, J.: Seeing Standards, <http://www.dlib.indiana.edu/~jenlrile/metadatamap/>.
28. ROAR: Registry of Open Access Repositories (ROAR), <http://roar.eprints.org/>.
29. Texier, J.: La representación de recursos usando la metodología del Desarrollo Dirigido por Modelos en un repositorio institucional. Facultad de Informática (2015).
30. Texier, J., De Giusti, M.: Elements of Resource Representation in Institutional Repositories: a Bibliographic Review. *J. Inf. Organ. Sci.* 38, 1, (2014).
31. Weng, C., Mi, J.: Towards accessibility to digital cultural materials: a FRBRized approach. *OCLC Syst. Serv.* 22, 3, 217–232 (2006).
32. Whittle, J. et al. eds: *Model Driven Engineering Languages and Systems. MODELS 2011*, Wellington, New Zealand, (2011).
33. Zavalina, O.L.: Subject Access: Conceptual Models, Functional Requirements, and Empirical Data. *J. Libr. Metadata.* 12, 2-3, 140–163 (2012).