

Model-driven software development in the institutional repositories

El desarrollo de software dirigido por modelos en los repositorios institucionales

Jose Texier ^a, Marisa De Giusti ^b & Silvia Gordillo ^c

^a MSc. Profesor de la Universidad Nacional Experimental del Táchira, Venezuela jtexier@unet.edu.ve

^b Ing. Profesora de la Universidad Nacional de La Plata, Argentina marisa.degiusti@sedici.unlp.edu.ar

^c Dra. Profesora de la Universidad Nacional de La Plata, Argentina gordillo@lifia.info.unlp.edu.ar

Received: February 15th, 2013. Received in revised form: September 10th, 2013. Accepted: November 6th, 2013.

Abstract

The Institutional Repositories (IR) have been consolidated into the academy, the proof of this is the growth in number of records in existing directories made by various means, such as self-archiving by authors and the incorporation of material by librarians. In this paper, a bibliographic survey about the use of the Model-Driven Software Development approach is done (MDD) in the IR systems in order to establish a relationship between them. The MDD is a paradigm for building software that assigns a central role to the models and originates models that range from the most abstract to the most concrete. This paradigm also provides a framework that allows the interested ones to share their points of view and directly manipulate the representations of the domain entities from IR. In conclusion, the monitoring of the various researches, and what is expressed here, allow to encourage the implementation of software for IR.

Keywords: institutional repositories, Model-Driven Software Development, MDD, metamodels

Resumen

Los Repositorios Institucionales (RI) se han consolidado en la academia, prueba de ello es el crecimiento en número de registros en los directorios existentes realizado por diferentes vías: autoarchivo por parte de autores, la incorporación de material a cargo de bibliotecarios, entre otras. En este trabajo se hace un relevamiento bibliográfico sobre el uso del enfoque de Desarrollo de Software Dirigido por Modelos (MDD) en los sistemas de RI con el propósito de establecer una relación entre ellos. El MDD es un paradigma de construcción de software que asigna a los modelos un rol central y se derivan modelos que van desde los más abstractos a los más concretos. Este paradigma, además, proporciona un marco de trabajo que permite a los interesados compartir sus puntos de vista y manipular las representaciones de las entidades del dominio. En conclusión, el seguimiento de las diferentes investigaciones relevadas y lo aquí expuesto permiten incentivar implementaciones de software para los RI.

Palabras Clave: repositorios institucionales, desarrollo de software dirigido por modelos, MDD, metamodelos.

1. Introducción

Los Repositorios Digitales están teniendo un gran auge en las comunidades universitarias, de manera especial, por su capacidad para difundir y preservar las actividades académicas y científicas, y, normalmente, se registran en los llamados "directorios de repositorios", por ejemplo, OpenDOAR [1] o ROAR [2]. El crecimiento de uso de estas estructuras web interoperables queda evidenciado en los depósitos diarios de artículos o documentos realizados por varias vías, tales como el autoarchivo por parte de los usuarios registrados, las catalogaciones por parte de los bibliotecarios, entre otras. Los repositorios tienen como propósito recopilar, catalogar, gestionar, acceder, difundir y preservar los recursos científicos, académicos y administrativos, descritos por medio de un conjunto de datos específicos (metadatos) [3], [4]. Es necesario aclarar

que en este trabajo se utilizan indistintamente los términos Repositorios Institucionales (RI) y Bibliotecas Digitales (BD), ya que gracias a la evolución de los servicios e infraestructuras web, principalmente en los RI, se puede decir que un Repositorio Institucional es una Biblioteca Digital y una Biblioteca Digital es un Repositorio Institucional [5]–[7]. De igual forma, los RI también son conocidos como Repositorios Digitales [8].

El diseño y desarrollo de repositorios requiere que las diferentes partes interesadas: desarrolladores, dueños del negocio y expertos del dominio, se pongan de acuerdo sobre un lenguaje neutral y de alto nivel para describir, discutir y negociar los servicios que se pueden ofrecer, motivo por el cual es deseable que la construcción de sistemas o aplicaciones para el dominio de los repositorios se realice

bajo metodologías de construcción de software, con el fin de obtener productos de mayor calidad y reuso [9]. Una metodología en crecimiento y soportada por tecnologías abiertas es el Desarrollo de Software Dirigido por Modelos (MDD) [10].

Para noviembre de 2012, existían alrededor de 2900 repositorios en el mundo [1], los cuales se han implementado y desarrollado con base en modelos conceptuales creados en los últimos 25 años, a saber: DELOS, FRBR, Norma ISO 14721, entre otros [11], [12]. Lo ideal para consolidar los sistemas de repositorios es contar con un modelo general a partir de las virtudes de los modelos conceptuales más destacados –adicional al aporte propio– y formalizado bajo una metodología de desarrollo de software. Por ello, este trabajo se centra en el MDD en los RI, los cuales se han convertido en sistemas de información complejos que están basados en tecnologías y características de diferentes áreas tales como bibliotecas, sistemas de información, recuperación de información, representación de información e interacción persona-computador [13]; y, el MDD es una metodología para la construcción de software que permite involucrar diversas tecnologías, facilita el reuso del software (unión de esfuerzos) y está dirigido por modelos [9].

La interrogante que este artículo pretende responder es, ¿cuál es la relación entre el MDD y los RI? Para ello, se ha organizado el trabajo de la siguiente manera: en la segunda sección se expone el estado del arte del trabajo y un relevamiento bibliográfico del tema; en la tercera sección se describen los resultados y el aporte del trabajo presentado; y, finalmente, en la cuarta sección se presentan unas conclusiones y trabajos futuros derivados de la experiencia del Servicio de Difusión de la Creación Intelectual (SEDICI) [14], repositorio central de la Universidad Nacional de La Plata, Argentina.

2. Estado del arte

2.1. Modelos conceptuales en los repositorios

La comunidad de desarrolladores de software ha comenzado a tomar conciencia de la necesidad de formalizar la construcción de componentes de software y guiar tales desarrollos a través de modelos, definidos como un conjunto de elementos que sirven para demostrar la consistencia de una teoría, es decir, representan con detalle un sistema dado [15]. Los sistemas de repositorios están caracterizados por problemas originados por la representación de los recursos y por la diversidad de soluciones tecnológicas en distintos módulos, tales como: almacenamiento, catalogación, indexación, infraestructura de la plataforma de software y preservación de los recursos [11], [12], [16].

La propuesta de enfocar el desarrollo de software en los repositorios con la metodología MDD, hace necesario relevar el elemento principal de esa metodología para la

construcción del software: los modelos conceptuales de los repositorios [9]. Por tanto, se realizó una búsqueda de documentos (*article* o *review*) en la base de datos bibliográfica de Scopus desde 1990 hasta la fecha (25 de noviembre de 2012), con los términos de descriptores truncados “*Model of Digital Librar**” o “*Model for Digital Librar**” presentes en el título, resumen y/o palabras clave con el propósito de identificar los modelos conceptuales existentes de sistemas de repositorios. Se obtuvieron 20 resultados de los cuales se descartaron 18 trabajos por estar vinculados a la evaluación de las bibliotecas digitales, preservación, tecnologías grid, ontologías, entre otros, y no a modelos conceptuales en relación con el dominio tratado. Por ende, surgieron dos indagaciones relacionadas con el objeto de estudio, a saber:

1. El modelo formal propuesto por Goncalves *et al.* en el 2004, conocido como “*Streams, Structures, Spaces, Scenarios, Societies*” (5S) [11]. Se basa en cinco abstracciones fundamentales: Streams, Structures, Spaces, Scenarios and Societies, las cuales ayudan a definir, relacionar y unir conceptos de objetos digitales, metadatos, colecciones y servicios, requeridos para formalizar las bibliotecas digitales. Las relaciones de las 5S con los objetivos de una biblioteca digital son: *los flujos* se refieren a la comunicación y al consumo de información por parte de los usuarios; *las estructuras* apoyan la organización de la información en formas útiles y significativas; *los espacios* tratan de la presentación y del acceso a la información en formas útiles y eficaces; *los escenarios* proporcionan apoyo para la definición y diseño de diferentes tipos de servicios; y, *las sociedades* definen la forma de una biblioteca digital ayudando a satisfacer las necesidades de información de sus usuarios.
2. El modelo general DELOS [12], de Candela *et al.* en el 2007, proponen un modelo para las bibliotecas digitales donde se identifican los conceptos que pasan a ser la piedra angular de este dominio. Por ello, establecen un framework llamado Three Tier Framework, con tres sistemas: Digital Library (DL), Digital Library System (DLS) y Digital Library Management System (DLMS). El primero de los sistemas, DL, es la organización que recoge, gestiona y preserva a largo plazo los contenidos digitales y que también los ofrece a las distintas comunidades de usuario. El sistema DLS es básicamente el sistema de software, la arquitectura requerida para brindar la funcionalidad de la biblioteca. Finalmente, el DLMS es la plataforma en sí: sistema operativo, bases de datos, interfaz de usuario que brinda la funcionalidad básica y la integración con el software especializado. Mientras que el concepto de DL es abstracto, el DLS y el DLMS capturan realizaciones concretas de sistemas de software.

Por tanto, el hallazgo de dos trabajos permite afirmar que se cuenta con un corpus muy limitado de modelos conceptuales de repositorios que sirvan de base para soluciones en el área bajo el paradigma de construcción de

software MDD. Los trabajos reseñados dan cuenta de unos modelos que pueden servir de insumo para el diseño de modelos conceptuales a usar en MDD.

2.2. Desarrollo de software dirigido por modelos (MDD)

El desarrollo de software dirigido por modelos, conocido en inglés como *Model Driven Development* (MDD), es un paradigma de construcción de software cuyas motivaciones principales son la independencia de los productores de software a través de estandarizaciones y la portabilidad de los sistemas de software [9]. El objetivo de MDD es separar el diseño del sistema de la arquitectura de las tecnologías, para que puedan ser modificados independientemente. Para lograr esto, se asigna a los modelos un rol central y activo bajo el cual se derivan modelos que van desde los más abstractos a los concretos, este proceso se realiza a través de transformaciones sucesivas e iteraciones. La mayor importancia de este paradigma radica en que todo debe girar sobre la base de modelos, definidos a partir de metamodelos, que ayudan al computador a entenderlos y a transformarlos [17]. El Model Driven Architecture (MDA o Arquitectura Dirigida por Modelos) es una propuesta de MDD definida por Object Management Group (OMG). En ocasiones, el término MDA se intercambia con el de MDD, ya que MDA se refiere a las actividades que llevan a cabo los desarrolladores, mientras que MDD hace referencia a su definición formal [10], por tanto, bajo este contexto, es indistinto hacer referencia a MDA o a MDD.

2.2.1. Ciclo de vida del MDD

El ciclo de vida de desarrollo de software en MDD (Fig. 1) basa su funcionalidad en tres modelos [18]. Un **modelo computacional independiente** (Computation Independent Model o CIM) que es una vista del sistema que no muestra detalles de su estructura y se le puede conocer como el modelo del dominio; esto corresponde tradicionalmente a las etapas de captura de requisitos y análisis [9]. El **modelo independiente de la plataforma** (Platform Independent Model o PIM) definido a través de un lenguaje específico para el dominio en cuestión e independiente de cualquier tecnología. El modelo PIM puede traducirse a uno o más **modelos específicos de la plataforma** (Platform Specific Model o PSM). Las implementaciones de los PSM, pueden estar basadas en lenguajes específicos del dominio o lenguajes de propósito general como Java, C, base de datos relacionales (SQL), Python, etc. Luego de definir cada uno de estos modelos, se realiza la implementación del código fuente (Implementation Model o IM) a partir de cada uno de los PSM desarrollados. Es importante destacar que los PIM o PSM pueden contener varios modelos correspondientes a distintos puntos de vista del sistema [18].

Este paradigma de desarrollo de software, que plantea una metodología basada en la generación de modelos (PIM y PSM) que pueden transformarse en otros modelos

sucesivamente hasta obtener una representación final (usualmente código ejecutable), está soportado sobre un lenguaje que a su vez está definido a partir de un metalenguaje [10], lo cual genera la idea de sucesivas capas recursivas y estructuradas en 4 capas o niveles de modelado, de acuerdo con lo establecido en la OMG [18].

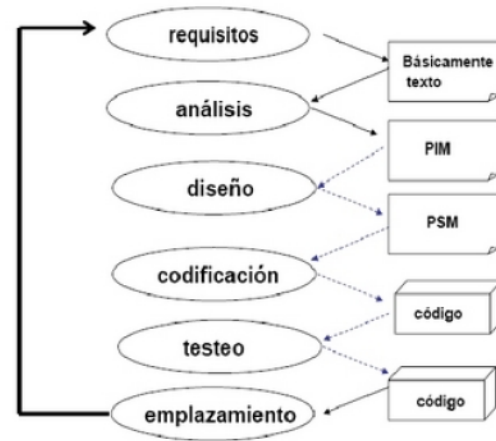


Figura 1. Ciclo de vida de MDD. Fuente: Kleppe et al. [19]

2.2.2. Antecedentes de MDD en los repositorios

Con el fin de conocer los *trabajos que existen sobre el dominio de los repositorios con el enfoque Model Driven* [20] se realizó una búsqueda en Scopus (1990-2012) con los términos de descriptores truncados: “*Model Driven* *” AND “*Institutional Repositor**”, o “*Model Driven* *” AND “*Digital Librar**”. Esta búsqueda se realizó sobre los campos título, resumen y palabras clave. Se encontraron 7 resultados, de los cuales 5 fueron tomados en cuenta porque muestran la línea de investigación de Malizia *et al.* [21]–[24], y la investigación de Paganelli *et al.* [25]. Los otros dos trabajos hallados fueron descartados porque se relacionan con transformaciones formales de modelos orientados a la preservación [26] y con la integración de funciones de modelos dirigidos por datos XML [27]. Tomando en cuenta que Scopus indexa la mayoría de los artículos presentes en las base de datos bibliográficas IEEE y Springer [28], [29], se realizó otra búsqueda con el fin de cruzar datos y verificar que los resultados coincidían y/o divergían. La búsqueda se realizó bajo los mismos criterios y se obtuvieron tres resultados: dos que coincidieron con la búsqueda anterior [21], [27] y un nuevo tercer trabajo que se descartó porque describe un framework para servicios web en las bibliotecas digitales [30]. Con el propósito de ampliar el corpus, se realizaron búsquedas en las conferencias más importantes sobre los RI y BD [31], dominio también conocido como LIS (Library and Information Sciences) [32]–[36], y en las conferencias referentes de la Ingeniería de Software, específicamente, en las áreas de generación automática de código y el enfoque dirigido por modelo

(Model Driven) [37]–[41]. En ambas búsquedas no se obtuvieron resultados.

Los trabajos hallados en el relevamiento descrito previamente se resumen en el de Paganelli *et al.* en el 2005 [25] y el de Malizia *et al.* en el 2010 [22]. El primero de ellos, 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. El trabajo de Malizia *et al.*, 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 BD, luego describen la arquitectura y el metamodelo del framework. Las características principales de este framework están en contar con 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. De igual forma, en el relevamiento anterior de Scopus, se encontraron 3 trabajos [21], [23], [24] que fueron investigaciones previas que están resumidas en el trabajo descrito de Malizia *et al.* [22].

Por tanto, la revisión de la literatura realizada permite concluir que el MDD 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.

En definitiva, los párrafos anteriores dan cuenta de dos relevamientos a partir de la búsqueda en bases de datos bibliográficas y de conferencias referentes del dominio LIS y de la Ingeniería de Software que se consideraron necesarios para conocer el estado del arte de un tema novedoso y con mucho campo en lo científico para la búsqueda de soluciones al dominio LIS, en beneficio de los usuarios principalmente. Asimismo, estos resultados reflejan un área de vacancia entre los modelos conceptuales para repositorios digitales y el desarrollo de software dirigido por modelos.

3. Resultados y aporte

3.1. MDD para el desarrollo de sistemas de repositorios

Lo ideal para consolidar el desarrollo de sistemas de repositorios es disponer de un modelo que represente a los recursos y diferentes servicios del repositorio para poder desarrollar aplicaciones de este dominio bajo metodologías dirigidas por modelos. Por ende, el paradigma MDD brinda un marco que permite a los interesados compartir sus puntos de vista y manipular directamente las representaciones de las entidades de este dominio [9]. A continuación, se

presenta una serie de beneficios para los tres actores presentes, del dominio de los repositorios, en el desarrollo de software que se pueden tomar tanto de la aplicación del paradigma como de su implementación en el contexto de los repositorios:

3.1.1. Los desarrolladores y/o diseñadores:

son los responsables de hacer realidad los requerimientos del sistema a desarrollar, cuyos beneficios son:

- Un menor número de líneas de código escritas, ya que los niveles de abstracción de MDD a través de los modelos y metamodelos diseñados fomentan el reuso del código y de los modelos.
- Alto nivel de abstracción para escribir aplicaciones y artefactos de software a través de la arquitectura de niveles del metamodelado y las capas de modelado de MDA. Este beneficio favorece diseñar una aplicación o artefactos de software partiendo de lo más general a lo más concreto, es decir, son independientes de la tecnología. Por ejemplo, los objetos del repositorio se pueden abstraer a un formato general propio de los RI, además de establecer relaciones entre ellos. Algunos ejemplos de objetos son: documentos, autores, instituciones, tesauros, sistemas de clasificación, áreas temáticas, etc.
- Especificación de requisitos de usuario a varios niveles obteniendo un sistema flexible a los cambios. La flexibilidad de estos sistemas se observa si la funcionalidad que se desea agregar es posible implementarla a través de los modelos correspondientes y evidentemente en el código generado.
- Evitar la adopción de una única tecnología de hardware particular gracias a los niveles de abstracción presentes sin generar un vínculo particular, por tanto, los metamodelos se convierten en el eje central, porque representan el modelo del sistema de manera independiente a la tecnología.
- Interoperabilidad entre los objetos en los sistemas de RI en un entorno multiplataforma. Esto se logra por los niveles de abstracción que permiten representar en un modelo general las relaciones entre objetos del RI en diversas plataformas.

3.1.2. Dueños del negocio:

son los encargados de coordinar y/o financiar el proyecto de desarrollo e implementación del sistema dentro de la organización o institución, y los beneficios los dueños del negocio pueden coordinar son:

- Desarrollo de componentes de software para los sistemas de repositorios. Este beneficio se evidencia por las fases del MDD, que van del CIM a la generación de código, ayudando a los desarrolladores a crear diseños de artefactos correctos sin causar problemas en el sistema general con el fin de incrementar la

productividad y calidad del mismo. Esos artefactos pueden ser indicadores para evaluar el uso del sistema, por ejemplo, recursos descargados y visitados, tiempo de permanencia en vistas de los objetos, etc.

- Preservación digital de los recursos y/o de los objetos generando estrategias que parten de los niveles altos de abstracción, como son los PIM. La importancia de la preservación radica en garantizar la perpetuidad en el tiempo del recurso almacenado.
- Generación de código para plataformas previamente especificadas a través de la funcionalidad que ofrece el paradigma, ya que los artefactos de software se generan en los PSM que luego se transformarán en el código deseado.
- Reducción de costes en el desarrollo de aplicaciones debido a la disminución del recurso humano requerido, de las horas hombre y del tiempo invertido en las diferentes actividades relacionadas.
- Documentación de todo el proceso de desarrollo de software, representado básicamente en el modelo PIM, documentación de alto nivel que se necesita para cualquier sistema de software.

3.1.3. Expertos del dominio:

representan a los especialistas presentes en las fases del mundo de los RI. Los beneficios en los que los expertos pueden interactuar son:

- Permitir la revisión de modelos por parte de los distintos expertos del dominio, a diferencia de los desarrolladores que se concentran en los detalles técnicos a través de transformaciones de los modelos partiendo desde los PIM hasta la generación de código.
- Generación de lenguajes específicos del dominio en las fases de la implementación de los RI bajo MDD, tales como: modelo de datos, modelo de la arquitectura, modelo de las entidades abstractas, interfaz de usuario, entre otros. Este beneficio se obtiene en el desarrollo de modelos correspondientes a diferentes puntos de vista para las fases principales del paradigma, PIM y PSM, al mismo tiempo la posibilidad de implementar lenguajes específicos del dominio para esos puntos de vista garantizando el reuso de los conceptos.
- Interoperabilidad entre los distintos modelos PSM principalmente, ya que pueden pertenecer a distintas tecnologías (plataformas). Esta interoperabilidad se logra a través de puentes construidos por las herramientas de transformación de modelos garantizando los conceptos definidos.

Estos beneficios ayudan a que todo el entorno del dominio de los RI se concentre en implementaciones de software más formales, generando una 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.

3.2. ¿Por qué MDD en los sistemas de repositorios?

El ideal 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 los servicios (destinados a recopilar, catalogar, gestionar, acceder, difundir y preservar) que el repositorio debe ofrecer. Por ello, el paradigma MDD brinda el marco que permite a los interesados compartir sus puntos de vista y manipular directamente las representaciones de las entidades de este dominio. Además, 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 [9].

La utilidad del metamodelo en MDD está centrada en definir lenguajes de modelado sin ambigüedades, en contar con herramientas de transformación para leer y entender los modelos, en tener reglas de transformación claras que describen cómo un modelo en un lenguaje fuente va a ser transformado a un modelo en un lenguaje destino y en el uso de definiciones formales obtenidas por la sintaxis de los lenguajes, facilitando su automatización.

La aplicación del paradigma MDD al complejo sistema de información de los Repositorios Institucionales permite trasladar las ventajas del MDD y la utilidad del metamodelado al desarrollo de los sistemas de repositorios, que involucran tecnologías y características de áreas como: bibliotecas, sistemas de información, recuperación de información, representación de información e interacción persona-computador [13]. Estos beneficios son capitalizados a su vez en los diferentes procesos presentes en los RI proporcionando a los desarrolladores conceptos del dominio sin ambigüedades.

4. Conclusiones y trabajos futuros

La idea de aplicar el Desarrollo de Software Dirigido por Modelos en sistemas y aplicaciones para RI que se ha expuesto en este trabajo, se tomaron en cuenta a partir del relevamiento bibliográfico realizado de los modelos conceptuales más importantes y de las experiencias de usar la metodología MDD en el mundo de los Repositorios. Los resultados obtenidos con el relevamiento evidencian la limitada literatura al respecto y demuestra cómo el MDD para RI/BD es un área de vacancia que requiere de más estudios y converge en un tema novedoso.

Por otra parte, se muestra en los modelos conceptuales cómo el concepto de los Repositorios ha evolucionado y se ha relacionado con el concepto de una Biblioteca Digital. Además, se destaca el gran auge que han tenido los RI en los últimos años de acuerdo con los directorios de repositorios existentes y con el afianzamiento de la filosofía

del acceso abierto en la comunidad de investigadores y académicos. Todas estas realidades incentivan el estudio de este dominio así como también el desarrollo de aplicaciones y componentes de software. De igual manera, se recomienda que todos los trabajos, investigaciones y desarrollos de metamodelos, modelos y artefactos de software, sean desarrollados bajo la filosofía del *open source* para que exista un crecimiento y una constante retroalimentación de los múltiples productos generados, de esta manera se podrá extender un poco más la poca bibliografía existente.

La construcción de software es aconsejable realizarla en el marco de alguna metodología, por ello, este trabajo se centró en el Desarrollo de Software Dirigido por Modelos. Se expuso una definición y una breve descripción de la metodología, de igual manera se encontraron trabajos que afirman la consolidación de enfoques dirigidos por modelos y de los diversos beneficios de la aplicación de este enfoque y de otros, que surgen de la implementación del paradigma en este contexto.

Evidentemente, se debe seguir ampliando el tema y profundizando en determinados aspectos, por ello, desde SEDICI se desea dar continuidad al trabajo presentado complementándolo con los siguientes tópicos:

- Análisis, evaluación y comparación de las diferentes metodologías dirigidas por modelos que pueden aplicarse para las aplicaciones y/o artefactos de software del dominio tratado en este trabajo.
- Diseño de un modelo flexible que mejore la representación de recursos dentro de los RI/BD y bajo el paradigma del Desarrollo de Software Dirigido por Modelos propuesto por OMG.

Referencias

- [1] OpenDOAR, OpenDOAR - Home Page - Directory of Open Access Repositories, 2013. [Online]. Available: <http://www.opendoar.org/>. [Accessed: 21-Feb-2013].
- [2] ROAR, Registry of Open Access Repositories (ROAR), 2013. [Online]. Available: <http://roar.eprints.org/>. [Accessed: 21-Feb-2013].
- [3] C. A. Lynch, Institutional Repositories: Essential Infrastructure for Scholarship in the Digital Age, ARL., Feb-2003. [Online]. Available: <http://www.arl.org/resources/pubs/br/br226/br226ir.shtml>. [Accessed: 28-Feb-2013].
- [4] H. Van de Sompel, S. Payette, J. Erickson, C. Lagoze, and S. Warner, Rethinking Scholarly Communication D-Lib Magazine, . 10. (9), Sep. 2004.
- [5] D. Waters, What Are Digital Libraries?, CLIR Issues, vol. 4, 1998.
- [6] C. L. Borgman, What Are Digital Libraries? Competing Visions., Information Processing & Management, 35. (3), pp. 227–243, 1999.
- [7] J. Xia and D. B. Opperman, Current Trends in Institutional Repositories for Institutions Offering Master's and Baccalaureate Degrees, Serials Review, 36, (1), pp. 10–18, Mar. 2010.
- [8] G. G. Chowdhury and S. Chowdhury, Digital library research: major issues and trends, Journal of Documentation, .55, (4), pp. 409–448, Jan. 1999.
- [9] C. Pons, R. Giandini, and G. Pérez, Desarrollo de Software Dirigido por Modelos. La Plata, Argentina: Mc Graw Hill, 2010.
- [10] A. Navarro, J. Cristóbal, C. Fernández-Chamizo, and A. Fernández-Valmayor, "Architecture of a multiplatform virtual campus," Software: Practice and Experience, Sep. 2011.
- [11] M. A. Gonçalves, E. A. Fox, L. T. Watson, and N. A. Kipp, Streams, structures, spaces, scenarios, societies (5s): A formal model for digital libraries, ACM Trans. Inf. Syst., 22, (2), pp. 270–312, Apr. 2004.
- [12] Leonardo Candela, Castelli, Y. Ioannidis, S. Ross, C. Thanos, P. Pagano, G. Koutrika, H.-J. Schek, and H. Schuldt, "Setting the Foundations of Digital Libraries," D-Lib, 13, (3/4), Mar. 2007.
- [13] L. Guo, "On construction of digital libraries in universities," in 2010 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT), vol. 1, pp. 452–456. 2010.
- [14] M. De Giusti, N. Oviedo, A. Lira, A. Sobrado, J. Martínez, and A. Pinto, SEDICI – Desafíos y experiencias en la vida de un repositorio digital, RENATA, 1, (2), pp. 16–33, Aug. 2011.
- [15] J. Whittle, T. Clark, and T. Kühne, Eds., Model Driven Engineering Languages and Systems 14th International Conference, MODELS 2011, Wellington, New Zealand, October pp.16-21, 2011. Proceedings. .
- [16] D. Bawden and I. Rowlands, Digital libraries: Developing a conceptual framework, New Review of Information Networking, 5, (1), pp. 71–89, 1999.
- [17] P. Gaona, C. Montenegro, J. Cueva, and O. Sanjuan, Aplicación de ingeniería dirigida por modelos (MDA), para la construcción de una herramienta de modelado de dominio específico (DSM) y la creación de módulos en sistemas de gestión de aprendizaje (LMS) independientes de la plataforma, DYNA, 169 (78), pp 43-52. 2011.
- [18] OMG, MDA Guide Version 1.0.1, 2003.
- [19] A. Kleppe, J. Warmer, and W. Bast, MDA Explained: The Model Driven Architecture(TM): Practice and Promise. Addison-Wesley Professional, 2003.
- [20] C. Challiol, Desarrollo dirigido por modelos de aplicaciones de hipermédia móvil, Tesis, Facultad de Informática, 2011.
- [21] A. Malizia, P. Bottoni, S. Levialdi, and F. Astorga-Paliza, A Cooperative-Relational Approach to Digital Libraries, in Research and Advanced Technology for Digital Libraries, L. Kovács, N. Fuhr, and C. Meghini, Eds. Springer Berlin Heidelberg, 2007, pp. 75–86.
- [22] A. Malizia, P. Bottoni, and S. Levialdi, Generating Collaborative Systems for Digital Libraries: a Model-Driven Approach, Information Technology & Libraries, vol. 29, Dec. 2010.
- [23] E. Guerra, J. D. Lara, and A. Malizia, Model driven development of digital libraries-Validation, analysis and code generation, in Webist 2007 - 3rd International Conference on Web Information Systems and Technologies, Proceedings, 2007, vol. WIA, pp. 35–42.
- [24] E. Guerra, J. de Lara, A. Malizia, and P. Díaz, Supporting user-oriented analysis for multi-view domain-specific visual languages, Information and Software Technology, 51,(4), pp. 769–784, Apr. 2009.
- [25] F. Paganelli and M. C. Pettenati, A Model-driven Method for the Design and Deployment of Web-based Document Management Systems, Journal of Digital Information, 6, (3), Jan. 2006.
- [26] T. Triebsees and U. M. Borghoff, A Theory for Model-Based Transformation Applied to Computer-Supported Preservation in Digital Archives, in Engineering of Computer-Based Systems, 2007. ECBS '07. 14th Annual IEEE International Conference and Workshops on the, pp. 359–370. 2007
- [27] M. A. Martínez-Prieto, P. de la Fuente, J. Vegas, and J. Adiego, Integrating Functionality and Appearance with the Electronic Work

- Logical Structure, in Web Conference, 2007. LA-WEB 2007. Latin American, pp. 84–91. 2007
- [28] J. Texier, Notas metodológicas para cubrir la etapa de documentar una investigación, Proyecto de Enlace de Bibliotecas (PrEBi), Aug. 2011.
- [29] CRL, Welcome to ADAT | Academic Database Assessment Tool. [Online]. Available: <http://adat.crl.edu/>. [Accessed: 01-Feb-2013].
- [30] H. Skogsrud, B. Benatallah, and F. Casati, A trust negotiation system for digital library Web services, *Int J Digit Libr*, 4, (3), pp. 185–207, Nov. 2004.
- [31] C. L. Liew, Digital library research 1997-2007: Organisational and people issues, *Journal of Documentation*, 65, (2), pp. 245–266, Jun. 2009.
- [32] JCDL, Joint Conference on Digital Library - Home. [Online]. Available: <http://www.jcdl.org/>. [Accessed: 14-Feb-2013].
- [33] ECDL, European Conference on Digital Libraries. [Online]. Available: <http://ecdconference.isti.cnr.it/>. [Accessed: 14-Feb-2013].
- [34] TPD, Theory and Practice of Digital Libraries. [Online]. Available: <http://www.tpd.eu/>. [Accessed: 14-Feb-2013].
- [35] ICADL, International Conference on Asian Digital Libraries. [Online]. Available: <http://www.icadl.org/>. [Accessed: 14-Feb-2013].
- [36] Open Repositories 2012. [Online]. Available: <http://or2012.ed.ac.uk/>. [Accessed: 14-Feb-2013].
- [37] Code Generation 2013. [Online]. Available: <http://www.codegeneration.net/cg2013/index.php>. [Accessed: 14-Feb-2013].
- [38] ICMT, International Conference in Model Transformation. [Online]. Available: <http://www.model-transformation.org/>. [Accessed: 14-Feb-2013].
- [39] ICWE, International Conference on Web Engineering. [Online]. Available: <http://icwe2013.webengineering.org/>. [Accessed: 14-Feb-2013].
- [40] MDWE, Model Driven Web Engineering. [Online]. Available: <http://mdwe2012.pst.ifi.lmu.de/>. [Accessed: 14-Feb-2013].
- [41] MODELS, Model Driven Engineering Languages and Systems. [Online]. Available: <http://www.modelsconference.org/>. [Accessed: 14-Feb-2013].